# Hierarchical Associative Memories: The Neural Network for Prediction in Spatial Maps

Jana Štanclová and Filip Zavoral

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

**Abstract.** Techniques for prediction in spatial maps can be based on associative neural network models. Unfortunately, the performance of standard associative memories depends on the number of training patterns stored in the memory; moreover it is very sensitive to mutual correlations of the stored patterns. In order to overcome limitations imposed by processing of a large number of mutually correlated spatial patterns, we have designed the Hierarchical Associative Memory model which consists of arbitrary number of associative memories hierarchically grouped into several layers. In order to further improve its recall abilities, we have proposed new modification of our model. In this paper, we also present experimental results focused on recall ability of designed model and their analysis by means of mathematical statistics.

## 1 Introduction

The strategies for prediction in spatial maps can be based on ideas of Fukushima [2]. Let us e.g. imagine a situation when we walk through a real place known to us. In such a case, we usually see only a scenery close around us. However, we are often able to recall the scenery that we do not see yet but shall appear soon in the direction of our next movement. Triggered by the newly recalled image, we can also recall another scenery further ahead of us. Thus, we can in principle imagine the scenery of a wide area by a chain of recall processes. This ability helps us to ensure a quick and safe movement through a known environment. This process can be used e.g. in autonomous driving.

Within the framework of our previous research, we studied the approach to the problem of prediction in spatial maps introduced by Fukushima [2]. The performance of associative memories is limited by number of patterns which can be stored in the model. During our cooperation with Iveta Mrázová, we have proposed the model of the so-called Hierarchical Associative Memories (HAM) which was developed with a stressed necessity to work with huge amounts of data. We expect that the HAM-model will allow a reliable and quick storage and recall of larger amounts of spatial patterns with respect to the problem of prediction in spatial maps.

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 786 – 793, 2005.

## 2 The Associative Memories

The associative memory is a neural network, for which all its neurons are input and output neurons simultaneously and oriented interconnections are among all neurons. (Basic notions and characteristics of this memory can be found e.g. in [4]). All its weights are symmetric and each neuron is connected to all other neurons except itself. An output of the associative memory is the vector of the outputs of all the neurons in the associative memory. A weight matrix W of the associative memory with n (n>0) neurons is a n×n matrix $W=(w_{ij})$ where $w_{ij}$ denotes the weight between the neuron i and the neuron j.

For training an associative memory, the Hebbian rule can be applied. According to this rule, the training pattern $\mathbf{x^k} = (x_1^k,..., x_n^k)$ can be stored in the associative memory with the weights $w_{ij}$ (i,j=1,...,n) by adjusting the respective weight values $w_{ij}$: $w_{ij} \leftarrow w_{ij} + x_i^k x_j^k$ for i, j = 1, ... , n and $i \neq j$. We assume that the weight values $w_{ij}$ (i, j = 1,..., n) are initialized to zero. Hence, the weight matrix $W=(w_{ij})$ corresponds to the auto-correlation matrix. For unlearning the pattern $\mathbf{x^k}$, we adjust back the respective weight values $w_{ij}$: $w_{ij} \leftarrow w_{ij} - x_i^k x_j^k$ for i, j = 1, ... , n and $i \neq j$. During the iterative recall, individual neurons preserve their output until they are selected for a new update. It can be shown that the associative memory with an asynchronous dynamics - each neuron is selected to update (according to the sign of its potential value $\xi$ to +1 or -1) randomly and independently - converges to a local minimum of the energy function.

Associative memories represent a basic model applicable to image processing and pattern recognition. They can recall reliably even “damaged” patterns but their storage capacity is relatively small (approximately 0.15n where n is the dimension of the stored patterns [4]). Moreover, the stored patterns should be orthogonal or close to 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].

## 3 Prediction Inspired by the Fukushima Model

In the Fukushima model [2], the chain process of predicting (recalling) the scenery of a given place far ahead is simulated using the correlation matrix memory similar to the associative memory. A “geographic map” is divided into spatial patterns overlapping each other. These fragmentary patterns are memorized in the correlation matrix memory. The actual scenery is represented in the form of a spatial pattern with an egocentric coordinate system. When we “move”, the actual area “becomes shifted” relatively to our previous position in the direction of the “move” (in order to keep our body always in the center of the “scenery” pattern to be recalled). If the “scenery image” shifts following the movement of the body, a vacant region appears in the “still not seen scenery” pattern. This area is filled partially by already known pattern from previous position and partially by a vacant region from the “will not seen” part of scenery. We are trying to recall the rest of the pattern. During the recall, a pattern with a vacant “not yet seen” region (the so-called “incomplete future” pattern) is presented to the correlation matrix and the recalled pattern should fill its missing part (see Fig. 1).

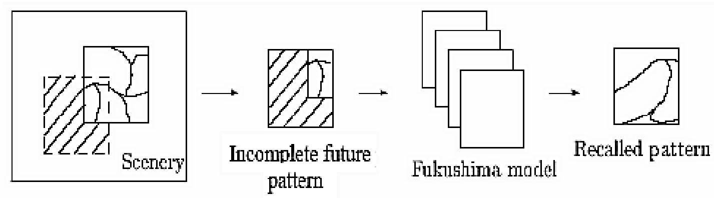


**Fig. 1.** Prediction inspired by the Fukushima model

Unfortunately, it is necessary to "place" the pattern presented to the correlation matrix exactly at the same location as one of the memorized patterns. The pattern to be recalled is shifted in such a way that the non-vacant region coincides with one of the memorized patterns. In order to speed-up the evaluation of the region-matching criteria, the Fukushima model incorporates the concept of the piled pattern. The point yielding the maximum correlation between the "seen scenery" and the corresponding part of the piled pattern should become the center of the next region.

The vacant part of the shifted pattern is filled, i.e. recalled by the auto-associative matrix memory. Although the recall process sometimes fails, it usually does not harm too much because the model contains the so-called monitoring circuit that detects the failure. If a failure is detected, the recalled pattern is simply discarded and recall is repeated after some time when the "body" was moved to another location.

## 4 The Hierarchical Associative Memory

Standard associative memories are able to recall reliably "damaged" or "incomplete" images if the number of stored patterns is relatively small and the patterns are almost orthogonal. But real patterns (and spatial maps in particular) tend to be correlated. This greatly reduces the possibility to apply standard associative memories in practice. To avoid (at least to a certain extent) these limitations, we designed (with cooperation with Iveta Mrázová) the so-called Hierarchical Associative Memory model (HAM-model). This model is based on the ideas of a Cascade Associative Memory (CASM) of Hirahara et al. [3] which allows to deal with a special type of correlated patterns. But our goal is to use the basic CASM-model more efficiently by allowing an arbitrary number of layers with more networks grouped in each layer.

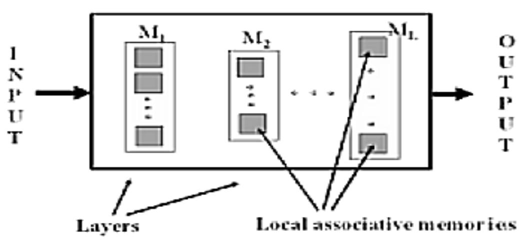


**Fig. 2.** Structure of the Hierarchical Associative Memory

A Hierarchical associative memory H with L layers (L>0) is an ordered L-tuple $H=(M_1,...,M_L)$ where $M_1,...,M_L$ are finite non-empty sets of associative memories; each of the memories having the same number of neurons n (n>0). A set $M_k$ (k=1,...,L) is called layer of the memory H. $|M_k|$ denotes the number of local associative memories in the layer $M_k$ (k=1,...,L). A training tuple T of H is an ordered L-tuple $T=(T_1,..., T_L)$ where $T_k$ (k=1,...,L) is a finite non-empty set of training patterns for the layer $M_k$. The structure of the HAM-model is shown in Fig.2.

Training of the HAM-model H lies in training each of its layer $M_k$ (k=1,...,L) separately (the so-called layer-training) and it can be done for all layers in parallel. In this way, the training patterns from the set $T_k$ will be 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. For each training pattern, "the most suitable" local associative memory in the layer $M_k$ is found and the training pattern is stored in it. If there is no "suitable" local associative memory, the new local associative memory is created and added to the layer. The pattern is stored in the newly created local memory. Now, we describe the so-called DLT-algorithm (dynamical layer training algorithm) for layer-training in formal way.

**The DLT-algorithm (for the layer $M_k$)**

1. The weight matrices of all local associative memories in $M_k$ are set to zero.
2. A training pattern **x** from $T_k$ is presented to the layer $M_k$.
3. The pattern **x** is stored in all local associative memories in the layer $M_k$ (according to the Hebbian training rule).
4. The pattern **x** is recalled by all local associative memories from $M_k$. Let us denote $\mathbf{y}^i$ the output of the i-th local associative memory in the layer $M_k$.
5. The Hamming distance $d_i$ of the pattern **x** and the output $\mathbf{y}^i$ is computed for each recalled output $\mathbf{y}^i$ ($i=1,.., |M_k|$).
6. The minimum Hamming distance $d_{min}$ is found ($d_{min} = \min \{d_i\}$, $i=1,..., |M_k|$). min is set to the index of the local associative memory with satisfying $d_{min}$. If there exist more local associative memories in $M_k$ with the same minimum Hamming distance $d_{min}$, min will be set to the lowest index of the local memory satisfying $d_{min}$.
7. The pattern **x** is unlearnt from local associative memories i in the layer $M_k$ where ($d_i \neq 0$ or $i \neq$ min).
8. If the pattern **x** is unlearnt from all local associative memories in $M_k$, a new local associative memory is created and added to the layer $M_k$. The pattern **x** is stored in the newly created local memory.
9. If there is any other training pattern in $T_k$, Step 2.

During recall, an input pattern **x** is presented to the HAM-model. The input pattern **x** represents an input for the first layer $M_1$. At every time step k ($1 \leq k \leq L$), the corresponding layer $M_k$ produces its output ${}^k\mathbf{y}$ which is used as the input for the "next" layer $M_{k+1}$ (i.e. ${}^{k+1}\mathbf{x} = {}^k\mathbf{y}$, $1 \leq k < L$). The output **y** of the HAM H is the output ${}^L\mathbf{y}$ of the "last" layer $M_L$. The recall process of the HAM-model is illustrated in Fig. 3.

Now, we focus on the recall process in one layer more precisely. During recall in the layer $M_k$, input pattern ${}^k\mathbf{x}$ is presented to the layer $M_k$. Afterwards, each local

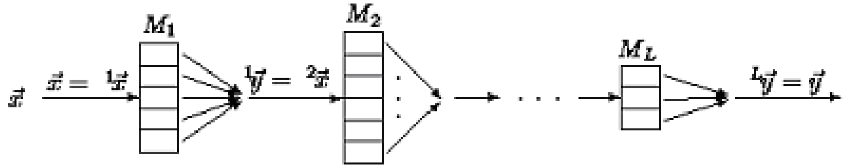

**Fig. 3.** Recall process in the HAM-model

associative memory in the layer $M_k$ produces the corresponding recalled output ${}^k\mathbf{y}^i$ (i=1,..., $|M_k|$). The output ${}^k\mathbf{y}$ of the layer $M_k$ is a recalled output which is "the most similar" to the input pattern ${}^k\mathbf{x}$. Now, we describe the so-called LR-algorithm (layer recall algorithm) for recall in one layer in formal way.

**The LR-algorithm (for the layer $M_k$):**

1. The input pattern ${}^k\mathbf{x}$ is presented to the layer $M_k$.
2. The pattern ${}^k\mathbf{x}$ is recalled by all local associative memories in the layer $M_k$. Let us denote ${}^k\mathbf{y}^i$ the output of the i-the local associative memory in the layer $M_k$.
3. The Hamming distance ${}^kd_i$ of the input ${}^k\mathbf{x}$ and the output ${}^k\mathbf{y}^i$ is computed for each i=1,..., $|M_k|$.
4. The minimum Hamming distance ${}^kd_{min}$ is found (${}^kd_{min} = \min \{{}^kd_i\}$, i=1,..., $|M_k|$). min is set to the index of the local associative memory satisfying ${}^kd_{min}$. If there exist more local associative memories with the same minimum Hamming distance ${}^kd_{min}$, min will be set to the lowest index of the local associative memory in the $M_k$ satisfying ${}^kd_{min}$.
5. The output ${}^k\mathbf{y}$ of the layer $M_k$ is the output ${}^k\mathbf{y}^{min}$ (i.e. ${}^k\mathbf{y} = {}^k\mathbf{y}^{min}$).

The training process starts with one local associative memory in each layer. Other local associative memories are added to the HAM-model during training according to the incoming patterns. Hence, the number of local associative memories in the HAM model depends only on the structure of training data.

Nevertheless, we should keep in mind that the above-sketched heuristic for storing patterns in the dynamically trained HAM-model is quick, simple and easy to implement but it is not optimal. Considering the DLT-algorithm, a pattern remains stored in such a local associative memory where the pattern is correctly recalled (Step 7 of the DLT-algorithm). If there is no such a local associative memory, a new local associative memory is created for storing the pattern. However, using this method for choosing the "most suitable" local associative memory, we cannot predict anything about recalling previously stored patterns. Some previously stored patterns can be recalled incorrectly (after storing some other patterns) or can even become lost.

## 5 Experimental Simulations and Analysis

In the previous paper [5], we presented preliminary experiments for the HAM-model and we compared the performance of the designed HAM-model with Fukushima model [2]. At this paper, we focus on robust ability of the HAM-model. In application of the HAM-model for spatial maps prediction, the problem requires a robust recall of presented patterns, often unknown in some parts of their surface. Due to such requirements, we have proposed the following two restrictions to the HAM-model in our simulations.

1. The number of patterns which can be stored in each local associative memory of the HAM-model is limited to 0.05n, where n is the dimension of stored patterns. It corresponds approximately to 30% of the capacity for standard associative memory.

2. A pattern remains stored in that local associative memory where even its "noisy" pattern (i.e. pattern where certain number of randomly selected elements change their value to the opposite one) is recalled correctly. If there is no such local memory, a new one is created to store this pattern. Hence, it is a modification of Step 7 of the DLT-algorithm.

Anyway, the above two modifications lead in general to an increased number of local associative memories. On the other hand, the experimental results show that the second restriction does not cause rapid increase of the number of the local associative memories in the HAM-model.

The experimental simulations are restricted to a two-level hierarchy of the HAM-model. Therefore, we can call the first- and second-level patterns to be ancestors and descendants, respectively. For experiments and their further statistical analysis, we have generated 100 sets of 100 randomly generated bipolar patterns (each of size 15x15). In a bipolar pattern, every elements take the value +1 or -1. For each set of patterns, 1/4 of the patterns with the smallest cumulative correlation between the respective pattern were chosen to be the ancestors and the remaining patterns were used to form the descendants. During our experiments, we have tested "relatively small" patterns as we needed to do huge number of experiments to use statistical methods. We have performed also experiments with "bigger" data (approx. 100x100) and the results were very similar (or even a bit better).

Every experiment is run on its set of patterns independently on other sets. Experiments are repeated for every data set. During the training process, ancestors and descendants are stored in the HAM-model according to the training algorithm. During recall process, we test the HAM-model recall ability of stored patterns and their corresponding "incomplete future" patterns (of different level). A pattern is recalled correctly if it coincides with its original in "known" part. A pattern is recalled with error k if it varies with its original in k elements. For each set, we observe distribution of patterns which are correctly recalled and patterns which are recalled with error including error-rate.

First, we focus the HAM-model ability to recall stored patterns. We denote random variable Y which corresponds to ratio of correctly recalled patterns from a set of patterns. Suppose that Y has binomical distribution (number of correctly recalled patterns does not depend on other HAM-networks). For a large number of patterns, it is possible to approximate binomical distribution by a normal distribution with the same parameters.

Our experimental data leads to a null-hypothesis H: EY = 0.998 in favor of alternative hypothesis A: EY <> 0.998 at confidence level $\alpha = 0{,}05$ (EY denotes mean value of variable Y). Using statistical hypothesis testing, the hypothesis is rejected if $T \geq t_{m-1,\alpha}$ where $T = |\hat{Y} - y| / (\rho / \sqrt{n})$. According measured data, we can assess the truth of null-hypothesis H: EY=0,998 at confidence level $\alpha = 0{,}05$. Hence, we can say that the number of correctly recalled patterns is 99.8%.

Now, we focus on the HAM-model ability to recall "incomplete future" patterns. Let level k of a "incomplete future" pattern is a number of rows/columns of "incomplete" L-shaped area. During simulations, "incomplete future" of level 1, 2 and 3 are tested. Hence, a "incomplete future" pattern of the level 1 contains 13% (= 29 unknown elements / 225 total elements) "unknown" elements, the second one 25% (=56/225) "unknown" elements and the third type 36% (=81/225) "unknown" elements. The recall results are summarized in Table 1.

**Table 1.** The table shows a number of correctly and incorrectly recalled "incomplete future" patterns of the level 1, 2 and 3

| | Level 1 | Level 2 | Level 3 |
|---|---|---|---|
| Recalled correctly | 4993 | 4310 | 3884 |
| Recalled incorrectly | 5007 | 5690 | 6116 |

Moreover, the distribution of incorrectly recalled "incomplete future" patterns can be analyzed by means of an error function (i.e. number of incorrectly recalled elements in one pattern). The results are shown in Fig. 4.

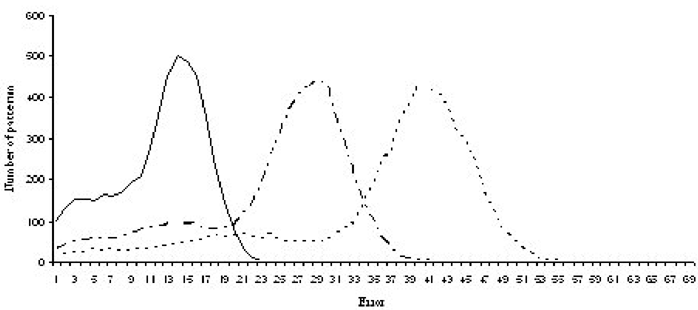


**Fig. 4.** Histogram of error in "incomplete future" patterns recalled incorrectly

In the figure, the axis X corresponds to a number of error in one recalled pattern and the axis Y corresponds to a number of recalled patterns with given error. The straight line depicts recall of "incomplete future" patterns of the level 1. The dash-and-dot line corresponds to recall of "incomplete future" patterns of the level 2. The dotted line denotes recall of "incomplete future" patterns of the level 3. When the "unknown" area is small, the model is able to recall such patterns quite well (the error does not exceed 10%). As the "unknown" area grows, the number of incorrectly recalled patterns is increased.

## 6 Conclusions

Our current research in the area of associative memories is focused on applications of associative memories for prediction in spatial maps. Unfortunately, the performance of standard associative memories depends on the number of training patterns stored in the memory, and is very sensitive to mutual correlations of the stored patterns. In order to overcome these limitations, we have designed the Hierarchical Associative Memory model. In this paper, we presents experimental results focused on recall ability of this model.

The HAM-model improves storage ability of standard associative memories to allow to deal with large number of mutually correlated patterns. For practical application with respect to prediction problem, it is necessary to further improve robustness of the HAM-model to recall correctly (or at least with small error) patterns with larger "unknown" area. The right choice of the ancestor patterns represents an important point of a successful application of the model. We are in the process of developing more sophisticated methods - based on self-organization - for choosing "the most suitable" ancestor patterns. This could improve the robustness of the HAM-model with respect to recall patterns contained in the "unknown" area.

In the future, we plan to analyze the time- and space-complexity of the HAM-model.

## References


1. D. J. Amit, H. Gutfreund, H. Sompolinsky: Information storage in neural networks with low levels of activity, in: Physical Review A, 35, 2293-2303,1987
2. K. Fukushima, Y. Yamaguchi, M. Okada: Neural Network Model of Spatial Memory: Associative Recall of Maps, in: Neural Network, Vol. 10, No. 6, pp. 971-979, 1997.
3. M. Hirahara, N. Oka, T. Kindo: A cascade associative memory model with a hierarchical memory structure, in: Neural Networks, Vol. 13, No. 1, pp. 41-50, 2000.
4. J. J. Hopfield: Neural Networks and physical system with emergent collective computational abilities, Proc. Natl. Acad. Sci. USA, 79: 2554-2558, 1982.
5. I. Mrázová, J. Tesková: Hierarchical Associative Memories for Storing Spatial Patterns, Proceedings of ITAT 2002, pp. 143-154, 2002.