Map Recall based on Hierarchical Associative Memories

Jana Štanclová, David Obdržálek

Department of Software Engineering, Faculty of Mathematics and Physics,
Charles University, Malostranské nám. 23, 118 00, Praha 1, Czech Republic
{jana.stanclova,david.obdrzalek}@mff.cuni.cz

Abstract

During recent years, artificial neural networks turned to be quite popular even in areas like cartography or navigation where processing of huge amounts of high-dimensional spatial data is needed. In this context, the data may represent geographical maps, plans of buildings, etc., which lead us straight to use similar ideas for autonomous devices operation and control.

When a traveller moves along a scenery, he can usually see only a very close surrounding. Based on his previous knowledge about the whole area, he might be able to recall also some part of environment which he does not see yet, but will see soon as he moves. We call such a recall process "path prediction". If we could formalize and algorithmize this process, we might nicely use it when building an autonomous robot. It would help a lot with robot localization and thus its better overall control in usage tasks where repetitive action in known environment is needed, e.g. an autonomous "warehouse keeper", a postman in a factory, automatic hoover etc.

In our paper, we show how strategies for path prediction in spatial maps can be based on associative memories, for which purpose a model of the so-called Hierarchical Associative Memory model (HAM) was designed. The HAM-model comprises an arbitrary number of associative memories grouped hierarchically in several layers. A suitable strategy applied during training the networks by dynamical adding of "local" associative memories to existing layers can allow a reliable storage and recall of larger amounts of spatial data. At the same time it enables storage of mutually highly correlated data, in contrast with standard associative memory models which work well with orthogonal data but easily fail when the data loses orthogonality.

1. INTRODUCTION

Let us imagine a situation when we walk through a real place known to us. In such a case, we usually see only a scenery quite 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 part of the 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.

For simple pattern recall, associative memories can be used. However, this method is not suitable for path prediction mentioned above, as it cannot cope with the need for high number of stored patterns and the fact the patterns are not orthogonal. In this paper, we show how the recall process can be implemented using proposed Hierarchical Associative Memory. This recall process can then be used while building an autonomous device, a robot, as a possible way how to implement one of the processes which help the robot to stay oriented in the real world environment.

The rest of this paper is organized as follows: Chapter 2 gives a motivation how map prediction can be applied in mobile robotics. In Chapter 3, the basic concept of associative memory is given.

Chapter 4 shows ideas of associative memory map prediction, and Chapter 5 defines Hierarchical Associative Memory model which enables implementation of the prediction. In Chapter 6 we give some experiment results and Chapter 7 concludes the paper with short evaluation of the method and some ideas about future work.

2. MOTIVATION

When designing an autonomous robot control, different information sources are usually taken into consideration: stimulae from bumpers and other sensors, odometry information, videocamera signal, beacon signals and many others. Most of the methods used to handle incoming measured data are burdened with errors and cumulative errors and correct localization information usually cannot be maintained for longer time with sufficient precision using only a single information source. For example, the odometry information originating form a car-like robot wheels cannot cope with wheel slides on a slippery surface or with robot position change if external forces shift the robot sideways. Thus, to localize the robot in its virtual world, overwhelming majority of implementations use a combination of more methods of processing the source data. Our proposed map recall can be used to widen the palette of such methods, which enable the control mechanisms of the robot to determine the robot location in the real world well.

Map recall results can be used for example to compare or match against previously acquired (or learnt) “world map” and can help eliminate or at least decrease the error coming from other localization algorithms. Let us imagine a control and planning mechanism which has to guide a robot from one room in a building to another one. As the robot moves along the corridors, it has to turn on the junctions to follow the journey plan. Before it can actually turn, the robot must detect a junction and decide which junction exactly it is at. If we implement the proposed map recall to process data originally coming from a camera attached to the robot, we could acquire an output which is in fact a prediction what the robot will see in a near future (“behind a corner”). This in fact means we could provide quite significant information to the robot control mechanism *before* it actually comes to the junction or the turn (and can detect it using regular sensors). At first, the control mechanism can have more time to compute its decision, which is helpful especially when computing power of the decision unit is not big and regular processes would lead to a robot which stops every now and then to decide its next steps. The control mechanism can also start manoeuvring before it detects the junction via other sensors, which can make its move smoother or even possible at all (e.g. a big robot in a narrow place). The input for the map recall does not necessarily have to be a camera pictures, it can be any information source from any kind of sensor provided that we are able to virtually create the overall “map” of sensor data for the whole environment.

3. THE ASSOCIATIVE MEMORIES

The associative memory is a neural network, for which all neurons are input and output neurons simultaneously and oriented interconnections exist between all neurons (basic notions and characteristics of this memory can be found 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 $n\times n$ matrix $W=(w_{ij})$ where w_{ij} denotes the weight between the neuron i and the neuron j.

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. 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].

4. MAP PREDICTION

In the Fukushima model [2], the chain process of predicting (recalling) the scenery of a given place far ahead is simulated using a 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 centre of the pattern to be recalled). As the scenery image shifts following the movement of the body, a vacant region appears in the "still not seen" part of the pattern. This area is partially filled by already known pattern from previous position and partially is left empty to represent the "not yet seen" part of scenery. We are trying to recall the empty part of the pattern. During the recall, a pattern with a vacant "not yet seen" region (the so-called "incomplete pattern") is presented to the correlation matrix and the recalled pattern is expected to be filled (see Figure 1).

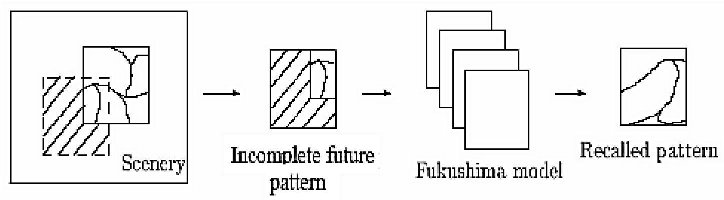

Figure 1. Map prediction inspired by the Fukushima model

Unfortunately, it is necessary to place the pattern presented to the correlation matrix exactly at the same location as one of the memorized patterns. Therefore, 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 centre of the next region.

The vacant part of the shifted pattern is topped up by the auto-associative matrix memory. Although the recall process sometimes fails, it usually does not harm too much because the model contains a 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 is moved to another location.

5. THE HIERARCHICAL ASSOCIATIVE MEMORY

Standard associative memories are able to reliably recall damaged or incomplete images as long as the number of stored patterns is relatively small and the patterns are almost orthogonal. Hence, the performance of associative memories is limited by number of patterns which can be stored in the model and the fact the patterns have to be 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 have designed the Hierarchical Associative Memory model (HAM-model; for details see the design in Mrázová [5]). This model is based on the ideas of a Cascade Associative Memory (CASM) of Hirahara et al. [3] which allows dealing with a special type of correlated patterns. 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.

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 (the so-called local 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 Figure 2.

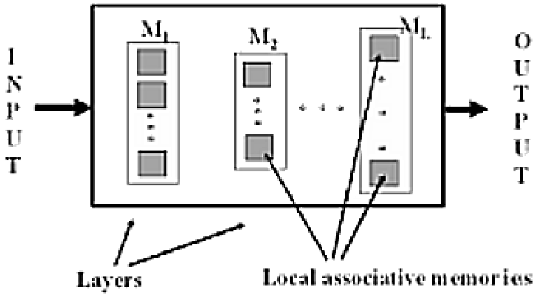

Figure 2. Structure of the Hierarchical Associative Memory

For training the HAM-model H, we have designed the "DLT-algorithm" – dynamical layer training algorithm whose main ideas follow: Each of the layers M_k (k=1,...,L) is trained separately (this can be done for all layers in parallel). 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 (where the pattern is stored and recalled correctly) is found and the training pattern is stored in it. If there is no suitable local associative memory, a new local associative memory is created and added to the layer and the pattern is stored in this newly created local associative memory.

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 ($1 \le k \le L$), 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. ${}^{k+1}\mathbf{x} = {}^{k}\mathbf{y}$, $1 \le k < L$). 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 Figure 3.

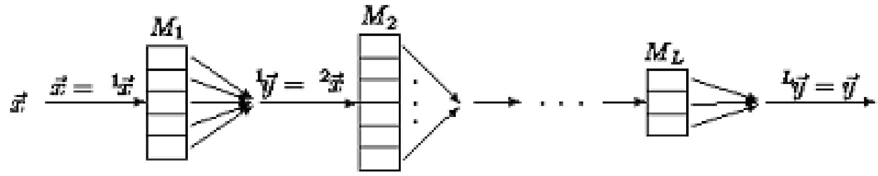

Figure 3. Recall process in the HAM-model

During recall in the layer M_k, input pattern ${}^{k}\mathbf{x}$ is presented to the layer M_k. Afterwards, each local 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 ${}^{k}x$ (in our tests, we use Hamming distance, but other metrics may be considered too).

Nevertheless, we should keep in mind that however the above-sketched heuristic for storing patterns in the dynamically trained HAM-model is quick, simple and easy to implement, it is not optimal. Considering the DLT-algorithm, a pattern remains stored in a local associative memory where the pattern is correctly recalled. If there is no such local associative memory, a new local associative memory is created for storing the pattern. The problem is that when 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.

6 EXPERIMENTAL SIMULATIONS AND ANALYSIS

One of the key points in neural network implementation is the recall ability. In application of the HAM-model for map prediction in an autonomous device, we require a robust recall of presented patterns, often unknown in some parts of their surface.

Due to these requirements, we have proposed following restriction to the HAM-model in our simulations: a pattern remains stored in that local associative memory where even its "noisy" pattern (i.e. pattern where certain number of randomly selected elements changed their value) is recalled correctly. If there is no such local memory, a new one is created to store this pattern.

Our 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 basic experiments, we have generated 100 sets of 100 randomly generated bipolar patterns (each of size 15*15 points). For each set of patterns, 1/4 of the patterns with the smallest cumulative correlation between the respective patterns were chosen to be the ancestors and the remaining patterns were used to form the descendants.

In preliminary experiments [5], we have tested the recall ability of the HAM-model. In theory, the storage capacity of an associative memory is approximately 0.15n where n is the dimension of the stored patterns [4]. From our tests, we have found that the proposed HAM-model was not able to recall correctly all its training patterns. Hence, we decided to artificially restrict the storage capacity of every local associative memory.

Let us define the capacity coefficient c which reduces the maximum number of patterns stored in the memory. Then, the maximum storage capacity q of local associative memory is given by formula $q=0.15 \cdot n \cdot c$. The capacity coefficient c takes the value of 1 for standard associative memory. For decreasing c the maximum storage capacity is reduced and the recall ability rises. The results are summarized in the Table 1.

c	anc	desc	recalled correctly
1	1,15	3,68	56,71%
0,9	1,15	3,71	56,83%
0,8	1,15	3,89	82,73%
0,7	2	4,39	89,05%
0,6	2	4,32	92,80%
0,5	2	5,01	97,15%
0,4	2	6	99,20%
0,3	3	9	99,95%
0,2	5	13,2	100,00%

Table 1. Number of local associative memories and the number of patterns recalled correctly with respect to capacity coefficient c

The columns "anc" and "desc" correspond to the average number of local associative memories in the first and second layer, respectively. A pattern is recalled correctly if it coincides fully with its original (for incomplete patterns we measure only elements in the known part). A pattern is recalled with error k if it differs from its original in k elements.

Now, let us focus on the HAM-model ability to recall incomplete patterns. According to the above mentioned results, we decided to perform recall experiments with capacity level c=0.5. We tested three groups of incomplete patterns – the first group contained 13% of unknown elements, the second one contained 25% of unknown elements and the third one contained 36% of unknown elements, which corresponds to a diagonal shift of the pattern by 1, 2, or 3 pixels respectively. The recall results are showed in Figure 4. In this histogram, the X axis corresponds to the number of error elements in one recalled pattern and the Y axis corresponds to the number of recalled patterns with given error in relation to the total pattern number (patterns with recall error of 0 are not shown).

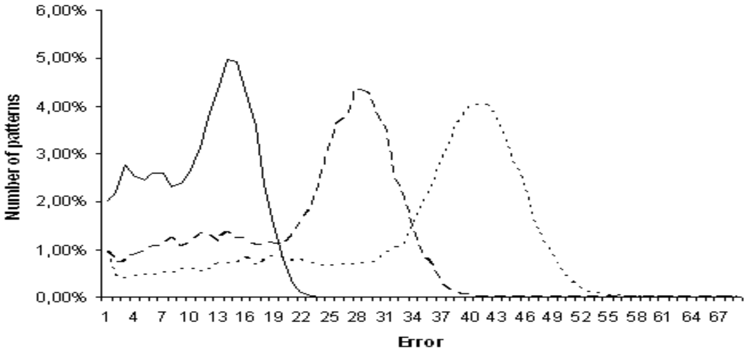

Figure 4. Histogram of incomplete patterns recall error in the HAM-model with capacity coefficient c=0.5.

7 CONCLUSIONS AND FUTURE WORK

We have focused on applications of associative memories for prediction in spatial maps. Their performance depends on the number of training patterns stored in the memory, and is very sensitive to mutual correlations of the stored patterns. We have therefore proposed the Hierarchically Associative Memory model (HAM), which improves the storage ability of standard associative memories and allows handling large number of correlated patterns. The experiments performed so far show promising results for the path prediction problem in respect to the use for mobile autonomous devices. The recall ability of HAM-model can be further improved by applying an artificial limit on storage capacity of the neural network. The experiments carried out have confirmed the legitimacy of these restrictions.

In the training learning phase, the right choice of a local associative memory for storing the presented pattern represents an important point of a successful overall performance of the model. For the first implementation, we have used a basic straightforward method, which we are now trying to improve. We hope it will directly increase the robustness of the HAM-model. As the work proceeds, we also 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.