ICCIMA'97

proceedings of

International Conference on Computational Intelligence and Multimedia Applications

10-12 February, 1997, GOLD COAST, AUSTRALIA

Edited by B.Verma and X. Yao

Sponsored and Hosted by

GRIFFITH UNIVERSITY
GOLD COAST CAMPUS

VIENA Classroom - a Collaborative Educational System with an Adaptive Multimodal Interface for Question Support
W. Winiwarter, O. Kagawa and Y. Kambayashi.....235

Improving Hypermedia Applications using Natural Language Processing Techniques
L. Colazzo and M. Costantino.....240

Multimedia Systems

Once Upon an Object...Computationally-Augmented Toys for Storytelling
J. W. Glos and M. Umaschi.....245

A Multi-User Storyboard-based Authoring System
T. M. Wong and K. W. Ng.....250

An Interactive Multi-Media Programme for Construction Safety Training
W. H. Fung, R.T Li and C. M. Tam.....255

A Review of Time Based Petri Net Models in Multimedia System
M. Chowdhury, S. Rahman, R. Bignall and S. Suthaharn.....261

A Model of Efficient Online Hypermedia Presentation Formats For Architecture Design Education
H.Lee, J. Choi, K. Lee, H.Lee.....267

A Multimedia Software Artifacts Library for Collaborative Software Reuse
S. M. Rahman and K. N. Tsoi.....271

Neural Network Applications

Network Intrusion Detection Using Neural Networks
A. M. Cansian, E. S. Moreira and A. de Carvalho.....276

An On-Line Complex Radial Basis Function Network and Its Application to Equalisation of Nonlinear Satellite Channels
B. M. Yoon and Y. Shin.....281

A Feature Based Neural Network Segmenter for Handwritten Words
B. Eastwood, A. Jennings and A. Harvey.....286

Using Neural Networks to Predict AFL Game Outcomes
E. Ong and A. Flitman.....291

Neural Networks and Intelligent Control

A Neural Model Reference Adaptive Controller
X. Yu.....296

Temporal Multidirectional Associative Memory Generating Spatial Trajectories
A. R. Araújo and M. Vieira.....301

Codebook learning in vector quantisation for image data compression
V. Delport.....306

The Quantum Behaviour Technique (QBT) to Intelligent Control
L. S. Yew.....311

Neural Network Learning

Comparison of ANN Training Algorithms based on Nonlinear Linearization and Linear Problems
P. Wen, J. Cao and D. Zhang....317

Hybrid Learning with Dynamic Bounds
R. Suna and J. Gizzi....322

Visualisation of Learning in Neural Networks using Principal Component Analysis
M. Gallagher and T. Downs....327

Eliminating Oscillatory Behaviours without Neuronal Hysteresis in Solving Optimisation Problems with Binary Neural Networks
L. Wang....332

Neuro-Fuzzy Systems

Neural Fuzzy Search Agents
S. Mitaim and B. Kosko....336

A Fuzzy Neural Approach to Speed Control of an Elastic Two-Mass System
Y. Frayman and L. Wang....341

A Rapid Learning Algorithm for a Neuro-Fuzzy Controller
Z. Zhang, P. Anderson and B. Bignall....346

Neuro Fuzzy Control of Cache Memories in Pipelined Processors
O. Hammami....351

Reasoning

Reasoning based on contectual implications
T. Makino and T. Iida....356

Patterns of Scientific Explanation as Nonmonotonic Reasoning under Uncertain Evidence
C. Delrieux....361

Transformations of Two Dimensional Uncertainty in Distributed Expert Systems: A Case Study
X. Luo and C. Zhang....366

The Favoring-Opposing Cause Model for Uncertain Reasoning
X. Luo and C. Zhang....373

Teaching Intelligent Information Management
G. Antoniou and M. A. Williams....380

Text/Image Generation/Retrieval

The Domain Knowledge and Automatic Indexing for the Korean Text Retrieval
K. S. Kim and P. K. Kim....384

An Experimental Study of an area based method for shape representation and similarity measure
A. Sajjanhar and G. Lu....388

5. References

[1] Brockett RW. Finite Dimensional Linear Systems, John Wiley & Sons, Inc. 1970.

[2] Colina-Morles E, Mort N. Neural network based adaptive control design. Journal of Systems Engineering, 1993; 1: 9–14.

[3] Kuschewski JG, Hui S, Zak SH. Application of feedforward networks to dynamic system identification and control. IEEE Transactions on Control Systems Technology, 1993; 1: 37–49.

[4] Rumelhart DE, McClelland JL, the PDP Research Group. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol.1:Foundation, Cambridge, MA:MIT Press, 1986.

[5] Sira-Ramirez H, Zak SH. The adaptation of perceptrons with applications to inverse dynamic identification of unknown dynamic systems. IEEE Transactions on Systems, Man, and Cybernetics, 1991; 21: 634–643.

[6] Sira-Ramirez H, Colina-Morles E. Adaptive learning in perceptrons: a sliding mode control approach. Pure Mathematics and Applications, 1993, 4: 99–133.

[7] Slotine JJE, Li W. Applied Nonlinear Control, Prentice-Hall, Inc, 1991.

[8] Stotsky AA. Combined adaptive and variable structure control, in *Variable Structure and Lyapunov Control*, ed. A. S. I. Zinober, chapter 14, Springer-Verlag, London, 1994.

[9] Utkin VI. Sliding Modes in Control Optimization, Springer-Verlag Berlin, Heidelberg, 1992.

[10] Widrow B, Winter R. Neural nets for adaptive filtering and adaptive pattern recognition, Computer, 1988; 12: 25–39.

[11] Yoo DS, Chung MJ. A variable structure control with simple adaptation laws for upper bounds on the norm of the uncertainties. IEEE Trans. Automatic Control, 1992; 37:860–865.

[12] Yu X. A perceptron adaptation algorithm for neural networks and its application, in Artificial Intelligence - Sowing the Seeds for the Future, eds. C. Zhang, J. Debenham, D. Lukose, World Scientific, pp. 411-417, 1994.

[13] Yu, X., Man, Z. and Rahman, S.M.M., Adaptive sliding mode approach for learning in a feedforward neural network, submitted for publication in Neural Computing and Application, 1996.

[14] Zinober ASI. Deterministic Control of Uncertain Systems. Peter Peregrinus Ltd. London, 1990.

Temporal Multidirectional Associative Memory Generating Spatial Trajectories

Aluizio F. R. Araújo Marcelo Vieira
Universidade de São Paulo - Depto. de Engenharia Elétrica
C.P. 359, São Carlos, SP Brazil
aluizioa@lasi01.sel.eesc.sc.usp.br vieira@lasi01.sel.eesc.sc.usp.br

ABSTRACT - This paper introduces a new version of the Multidirectional Associative Memory (MAM) in which the state of activation of the processing units ranges from -1 to +1 and each unit has one-to-one self-connections. Furthermore, the correlation matrices, generated by Hebbian rules, are trained by Widrow-Hoff rule. The model aims at reproducing temporal sequences. The results of the experiments suggest that the model has a fast training stage, improves the learning capacity of MAM, reproduces trained temporal sequences, interpolates and extrapolates states in a trained sequence, and improves the accuracy of the recall with these inclusions of states.

Keywords: Neural networks, associative memory, multidirectional associative memory, temporal sequences, Widrow-Hoff rule.

1. Introduction

An associative neural memory model [1] is understood as a class of artificial neural networks which stores information as stable attractors through Hebbian rules. When a perfect, partial, or noisy version of a trained piece of information is presented to the network, it responds to the stimulus with the closest of the stored memories. The outcome is reached through a self-relaxation process. This paper focus attention on a number of models characterized by a topology with a set of layers densely interconnected, construction of Hebbian correlation matrices, discrete representation, and self-relaxation dynamics. A particular approach, among the elected ones, is chosen and modified in order to improve its performance and increase its range of applications.

The models above are classified, according to the nature of the associations they are able to perform, as autoassociative or heteroassociative. The autoassociative models [2], [3], [4], are formed by a single layer in which all processing units are totally connected by feedback links. These models store binary or bipolar patterns and may operate synchronously or sequentially. The convergence process of these models may end at a desired point, a spurious attractor or a limit cycle. The main range of application of these models involves retrieving of noisy, distorted or incomplete versions of the stored patterns. The heteroassociative models [5], [6], [7], are taken as an extension of the former models. They are formed by more than one layer and each layer is fully connected with all the others. These models construct mappings between couples of layers. The models store binary or bipolar patterns and operate synchronously. They are tolerant to noise and incomplete stimuli.

The Bidirectional Associative Memory (BAM) [6] is likely to be the most used among the heteroassociative models. This is a two-layer memory model which associates pairs of patterns. The correlation matrix in BAM is constructed by a Hebbian rule, the network always converges to stable attractors, and it is relatively easy to be implemented in hardware. However, BAM can simply perform one-to-one mapping, presents small storage capacity and does not guarantee correct recall. Such limitations motivated the introduction of models dealing with multiple associations, the Multidirectional Associative Memory (MAM) [7], and training strategies to improve the storage capacity of BAM [8].

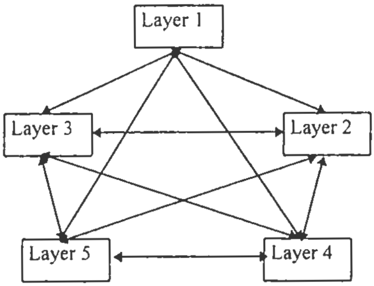
This papers aims at introducing a modified version of MAM which allows real and limited representation, autoassociative connections, and a simple and fast training stage to improve the learning capacity of MAM. The new model will be tested to store and recall temporal sequences. The proposed model does not use hidden layers and backpropagation to train the network as most of the works dealing with temporal sequences [9].

This paper discusses the main features of MAM in Section 2, introduces the modified model in Section 3, and reports on tests to evaluate the capacity of the model to deal with temporal sequences in Section 4. The main results are discussed in Section 5.

2. The Multidirectional Associative Memory

The Multidirectional Associative Memory (MAM) is a generalization of BAM. This model correlates the mutual existence of n-tuples of patterns. The associations are embodied in $nl(nl-1)/2$ correlation matrices, for a network with nl layers. The correlation matrix W_{lk} between the layers l and k is formed by the summation of the outer product of each pair of patterns which are presented to such layers. The matrix W_{lk} is involved in the information flow from layer l to layer k and W_{lk}^T is concerned with inverse flow. In this paper, the description of MAM will be restricted to a five dimensional model because this option will be used later in the simulations. However, all the descriptions are valid to models with any number of layers.

Let the network be a version of MAM with five layers ($nl = 5$) in which np bipolar 5-tuples will be stored. The activation state of two different layers due to a pattern p are represented by $(1Xn)$ and $(1Xm)$ vectors X^p and Y^p. The weighted sum of the layer l for a time step t is denoted by $\mathrm{NET}(t)_l$. The topology of the network and the equations determining its behavior are sketched in Figure 1.



(a)

The correlation matrix between layers l and k is:

$$W_{lk} = \sum_{p=1}^{np} (X_l^p)^T Y_k^p \qquad (1)$$

The propagation rule for the layer l is:

$$NET_l(t) = \sum_{\substack{k=1 \\ l \neq k}}^{nl} W_{lk} Y_k^T (t-1) \qquad (2)$$

The activation rule for the layer l is:

$$X_l(t) = \mathrm{sgn}(NET_l(t)) \qquad (3)$$

where, $\mathrm{sgn}(v) = \begin{cases} +1 & \text{if } v > 0 \\ -1 & \text{if } v \leq 0 \end{cases}$

(b)

Figure 1: (a) The topology of a five-layer MAM. (b) The equations describing MAM.

The dynamics of MAM is described as follows. Given an input pattern represented by a 5-nuple $(V_1', V_2', V_3', V_4', V_5')_{ini}$ which may be an incomplete or noisy version of a desired pattern $(V_1, V_2, V_3, V_4, V_5)$, the network changes states as shown below:

$$(V_1', V_2', V_3', V_4', V_5')_{ini} \rightarrow W_{lk} \rightarrow (V_1', V_2', V_3', V_4', V_5')_1 \rightarrow \ldots \rightarrow (V_1', V_2', V_3', V_4', V_5')_{fin} \rightarrow W_{lk} \rightarrow (V_1', V_2', V_3', V_4', V_5')_{fin} \quad (4)$$

The presented 5-tuple generates a new state through the correlation matrices. Each new state originates another one. This dynamic process ends when the state of the network stops changing. Ideally, the final state of the network $(V_1', V_2', V_3', V_4', V_5')_{fin}$ is the desired one $(V_1, V_2, V_3, V_4, V_5)$. However, very often, this does not occur.

In summary, this model has the following characteristics:

- The equilibrium state corresponds to a local minimum point;
- The training stage is simply the construction of the correlation matrices;
- The correct recall of a particular pattern is not guaranteed;
- The storage capacity of the model is very low;
- The discrete representation of the model limits its range of application;

A modified version of MAM is described below to deal with the last three undesired features.

3. The Temporal Multidirectional Associative Memory

This proposed model is treated as a dynamic system in a high-dimensional and continuous state space. Such a model preserves the original architecture of MAM in terms of connections between layers, however, it adds a set of autoassociative all-to-all connections in each layer. Then, the various connections associate a particular state of activation of each processing unit with both activation state of the units in the other layers and the activation of the units in its own layer, all of them delayed in one time step. The correlation matrix between layers l and

k are built employing equation (1) whereas the correlation matrix considering the connections between units in layer l is constructed as follows:

$$W_{ll} = \sum_{p=1}^{np} (X_l^p)^T X_l^p \qquad (5)$$

As a consequence the new propagation rule is:

$$NET_l(t) = \sum_{\substack{k=1 \\ l \neq k}}^{nl} W_{lk} Y_k^T (t-1) + W_{ll} X_l (t-1) \qquad (6)$$

The second modification concerns the continuous representation, thus the activation value of each processing units ranges from -1 to $+1$. The activation rule becomes:

$$X_l(t) = \tanh(NET_l(t)) \qquad (7)$$

where, $$\tanh(v) = \frac{1-\exp(-v)}{1+\exp(-v)}$$

This model is conceived to deal with continuous processes. Thus, the model aims at storing the associations between continuous patterns and recalling these patterns when required. The first experiments with this model showed poor performance on retrieving the stored patterns. The difference between the desired and the obtained outputs were significant. Thus, a training phase was introduced to diminish this error.

3.1. Two Learning Stages

The temporal MAM has two learning stages. In the first one, derived from the original MAM, all correlation matrices are built. In the second learning phase, a supervised training occurs.

The correlation matrices are constructed using equations (1) and (5). In order to store temporal sequences the patterns are presented to the network as follows:

- Let $nstate$ be the number of states V_t of a given temporal sequence;
- Let nl be the number of layers in the network in which all layers have the same number of units (n);
- The number of patterns (np) to be stored in the network is given by:

$$np = nstate - nl + 1 \qquad (8)$$

- Generate (np) patterns (P_p) varying $1 \leq p \leq np$ as follows:

$$t = p$$
$$P_p = (V(t), V(t+1), \ldots V(t+nl-1)) = (V_t, V_{t+1}, \ldots, V_{t+nl-1}) \qquad (9)$$

The training stage occurs following the construction of the correlation matrices. A previous attempt to train MAM was proposed in [10]. Such a proposal was based on the PRLAB [11], an algorithm that increases successfully the storage capacity of BAM through an adaptation of a relaxation method to solve a system of linear equations. Nevertheless, the Widrow-Hoff rule [12] was chosen to train the correlation matrices since such an option works better than PRLAB for BAM [13].

This training phase, in a 5-layer modified MAM, works as follows:

- An initial pattern $(V_1, V_2, V_3, V_4, V_5)_{ini}$ is presented to the network;
- The network produces a spurious attractor $(S_1, S_2, S_3, S_4, S_5)_{ini}$ as output;
- Each correlation matrix is adapted according to learning rule:

$$\Delta W_{lk}(t) = (\lambda / (n+1)) * \left((X_l^p - S_l^p)^T * Y_k^p \right) \qquad (10)$$

where: $\lambda/(n+1)$ is the learning rate;

X_l^p, Y_l^p are the desired result;

S_l^p is a spurious attractor.

The training stage stops when the error reaches a value smaller than maximum allowed error.

4. Experimental Results

In these tests the task is to store and recall n-tuples representing three dimensional positions and the instant of time associated with each spatial position. These tuples form a spatial movement of a particular point in the 3-D space. Each spatial position may vary from -1 to +1 which is represented by the activation state of each processing unit. The model has five layers of four processing units each. The three proposed sequences are formed by six, eight, and eight states respectively. The training patterns are generated following equations (8) and (9). The correlation matrices are constructed according to equations (1) and (5), and these matrices are trained following equation (10).

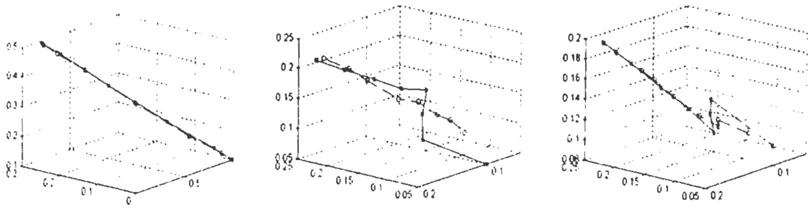


Figure 2: The recall of the trained points in the temporal sequences.

Three different trajectories are tested. The level of complexity to recall a particular sequence increases with the variations of the trajectory directions. The sequences took 2421, 736, and 212 epochs to be trained reaching an final error equal to 0.0606, 0.0528, 0.1446. In figures 2 - 3 the obtained trajectories are represented by solid lines whilst the desired sequences are sketched by dotted lines.

The first set of experiments tests the capacity of the network to reproduce the trained sequences. In this case, the first spatial point of each sequence and the trained time steps are given in order to recall the whole trajectory. The results, sketched in Figure 2, suggest that the retrieval capacity of the model diminishes as the complexity of the task increases. Therefore, the points that change direction abruptly are poorer recalled than those of smooth direction variations.

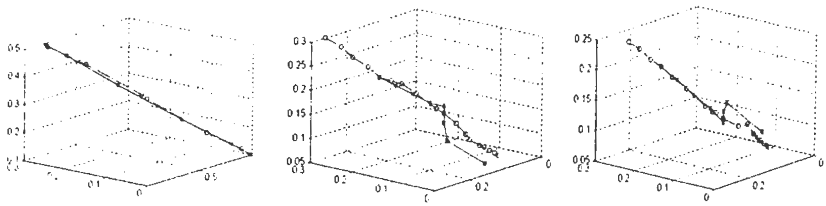


Figure 3: The recall of the trained, interpolated, and extrapolated points in the temporal sequences.

The second set of experiments aims at evaluating the capacity of the network to interpolate and extrapolating points in the sequences. This is made by presenting to the network four trained points. The point to be included, in any position of the sequence, is initially set to the desired time step and either the value of its preceding state or the value of its following state. The results (Figure 3) suggest that the model improves performance while includes points in trajectory.

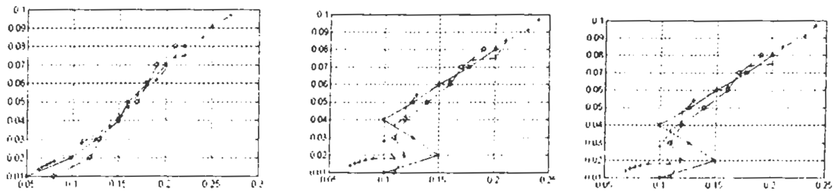


Figure 4: Variation of axes (a) x, (b) y, (c) z as function of time. The solid lines represent the trained sequences, the dotted lines with circles (signal plus) represent the obtained trajectories in the first (second) experiments.

In order to see the learning progress between the first and the second experiments, the variations of each dimension are plotted as function of the time in Figure 4. Thus, the results illustrate that the obtained trajectories get closer to the desired one as more points are interpolated and/or extrapolated in the trajectory.

5. Conclusions

This paper proposes a modified version of MAM: The Temporal MAM. This is an adaptable model with continuous representation which does not require long training processes, hidden layers, and backpropagation training. The model is tested to reproduce temporal sequences.

The results suggest that the model is able to reproduce a trained temporal sequence, to interpolate and extrapolate points in the sequence. Furthermore, the initial poor recollection of more complex trajectories may be ameliorated with the interpolation and extrapolation in the trained trajectories.

Acknowledgments

I would like to thank FAPESP, for providing financial support for this research.

References

[1] M. H. Hassoun (Ed): *Associative Neural Memories: Theory and Implementations*, Oxford University Press, New York, 1993.
[2] K. Nakano: "Associatron - a model of associative memory". *IEEE Transactions on Systems, Man and Cybernetics*, vol. 2, no. 3, pp. 380 - 388, 1972.
[3] S. Amari: "Learning patterns and pattern sequences by self-organizing nets of threshold elements", *IEEE Transactions on Computers*, vol. C-21, No.11, pp. 1197-1206, 1972.
[4] J. J. Hopfield: "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences*, vol. 79, pp. 2554-2558, 1982.
[5] K. Okajima, S. Tanaka, and S. Fujiwara: "A heteroassociative memory network with feedback connection", *IEEE First International Conference on Neural Networks*, vol. 2, pp. 711-178, 1987.
[6] B. Kosko: "Bidirectional associative memories", *IEEE Transactions on Systems Man and Cybernetics*, vol. 18, pp. 49-60, 1988.
[7] M. Hagiwara: "Multidirectional associative memory", *International Joint Conference on Neural Networks*, vol. 1, pp. 3-6, 1990.
[8] A. F. R. Araújo: "Memory, Emotions, and Neural Networks: Associative Learning and Memory Recall Influenced by Affective Evaluation and Task Difficulty". *Ph.D. Thesis*. School of Cognitive and Computing Sciences, University of Sussex, Brighton, UK, 1994.
[9] J. Hertz, A. Krogh, and R. G. Palmer: *Introduction to the Theory of Neural Computing*, Addison-Wesley, Redwood City, CA, 1991.
[10] M. Hattori, and M. Hagiwara: "Quick learning for multidirectional associative memories", *IEEE International Conference on Neural Networks*, pp.1949-1954, 1995.
[11] H. Oh, and S. C. Kothari: "Adaptation of the relaxation method for learning in bidirectional associative memory", *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp.576-583, 1994.
[12] B. Widrow and M. E. Hoff: "Adaptative switching circuits", *IRE WESCON Convention Record*, pp. 96-104, 1960.
[13] A. F. R. Araújo and G. M. Haga: "Two Simple Strategies to Improve Bidirectional Associative Memory's Performance: Unlearning and Delta Rule", Brazilian Symposium on Neural Network, pp. 39-46, Recife - PE, Brazil, 1996.