

Performing Complex Associations Using a Feature-Extracting Bidirectional Associative Memory

Craig Leth-Steensen¹, Sylvain Chartier², Dominic Langlois² & Marie-France Hébert²

¹Department of Psychology
Carleton University

²School of Psychology
University of Ottawa

craig leth steensen@carleton.ca, schartie@uottawa.ca, dlang074@uottawa.ca, mhebe060@uottawa.ca

Abstract

Learning in Bidirectional Associative Memory (BAM) is typically based on Hebbian type learning. Since Kosko's paper on BAM in late 80s many improvements have been proposed. However, none of the proposed modifications allowed BAM to perform complex associative tasks that combine many to one with one to many associations. Even though BAMs are often deemed more plausible biologically, if they are not able to solve such mappings they will have difficulties establishing themselves as good models of cognition. This paper presents a BAM that can perform complex associations using only covariance matrices. It will be demonstrated that this network can be trained to learn both the 2 and 3 bit parity problem. The conditions that provide optimal learning performance within this latter network framework are then explored along with some of its dynamical properties. Results show that contrary to other associative memory models, the proposed neural network is able to perform parity tasks while maintaining a basic property of BAMs, namely, its pattern reconstruction abilities.

Introduction

The learning of associations is one of the most basic and important processes in cognition. It is a phenomenon which has been philosophized about for 100s of years and studied empirically for just over the last 100 years. The essence of associationism is characterized within the neural network or connectionist modelling framework in terms of Hopfield networks (Hopfield 1982) and their generalization into bidirectional associative memories (BAMs; Kosko 1988), where Hopfield-type networks are used mainly for auto-association and BAMs for hetero-association. A key feature of BAM is that it can be used to associate two sets of patterns. In the vast majority of cases, BAM uses a one-to-one association type (e.g., a name is associated with a picture; Figure 1a). Within this model,

Copyright © 2010, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

patterns of activation (\mathbf{x}) across a set of x -layer units become associated with patterns of activation (\mathbf{y}) across a set of y -layer units through the application of a "one-shot" Hebbian-based learning rule used to derive the values of the connection weights (\mathbf{W}) between the two sets of units; illustrated in Equation 1.

$$(1) \quad \mathbf{W} = \mathbf{Y}\mathbf{X}^T$$

After learning, the presentation of any particular x -layer activation pattern to the network will serve to reinstate the particular y -layer activation pattern that it has been associated with (and vice versa) as described by Equation 2.

$$(2a) \quad \mathbf{y} = f(\mathbf{W}\mathbf{x})$$

$$(2b) \quad \mathbf{x} = f(\mathbf{W}^T\mathbf{y})$$

Although the learning of associations within such a BAM is always perfect for orthogonal pattern sets, the learning of correlated patterns sets requires the use of both nonlinearity in the output function of the units and recurrency whereby correct pattern association requires that the network first settles into the appropriate attractor state. Over the years, several variants have been proposed to overcome the original model's limited storage capacities and improve its noise sensitivity, and most of today's BAM models can store and recall many different types of pattern sets (e.g., Arik 2005; Du et al. 2005; Leung 1994; Shen and Cruz 2005; Wang 1996).

BAM is generally deemed to be more biologically plausible and more dynamically complex than many other classes of network models. However, they do have difficulties learning more complex types of associations. For example, although learning many-to-one (e.g., different pictures of tables are associated with the word "table") or one-to-many associations (e.g., storing two series patterns that contains similar items) are possible, very few are able (e.g., Chartier and Boukadoum 2006a). Moreover, complex associations (depicted in Figure 1b)

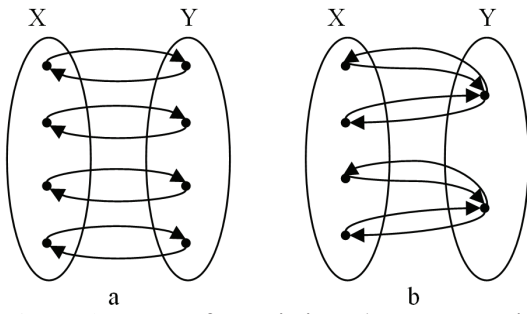


Figure 1. Types of association: a) one-to-one; b) complex (many-to-one combined with a one-to many).

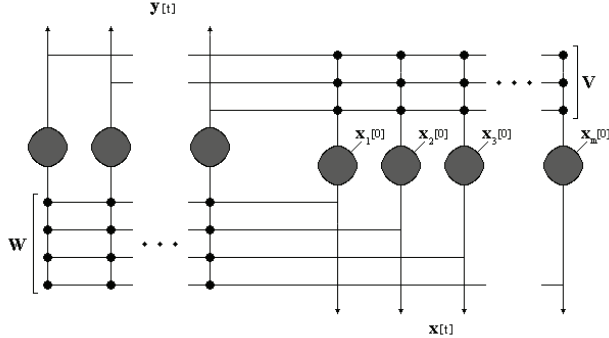


Figure 2. Architecture of the FEBAM model.

that incorporate both many-to-one with one-to-many associations (such as those exemplified by non-linearly separable associative mappings) are presently *impossible* to learn for this class of model. In this paper, it will be demonstrated that a recently proposed BAM model is indeed able to perform such complex associations under some conditions. The remainder of this paper is divided as follows. First, a brief introduction of FEBAM will be presented followed by simulations of two tasks that vary in their level of difficulty. Then, an extension of the model will be proposed, followed by a general discussion.

FEBAM

A variant of BAM, called feature-extracting bidirectional associative memory (FEBAM), has recently been proposed by Chartier et al. (2007). This model is able to perform nonlinear principal component analysis, clustering, as well as learning in noisy environments (Chartier et al. 2007; Giguère et al. 2007a; 2007b).

Architecture. FEBAM's original architecture is illustrated in Figure 2. This architecture is nearly identical to that of the BAM model proposed by (Chartier & Boukadoum, 2006b). It consists of two Hopfield-like neural networks interconnected in head-to-toe fashion. When connected, these networks allow a recurrent flow of information that is processed bidirectionally. As shown in Figure 2, the \mathbf{W} weights send information to the y -layer and the \mathbf{V} weights send information back to the x -layer, in a kind of “top-down/bottom-up process” fashion. As in a standard BAM, both layers serve as a teacher for the other layer.

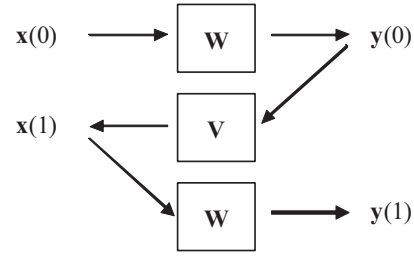


Figure 3. Output iterative process used for updates on learning trials.

The main difference between FEBAM and standard BAM lies in the absence of an “external” input to the y -layer. Typically, a BAM is used to associate two sets of known vectors where for each vector \mathbf{x} , a pre-associated vector \mathbf{y} must also be provided. In the present FEBAM case, there is no initial input to the y -layer as there is for $\mathbf{x}(0)$. Thus, $\mathbf{y}(0)$ is not obtained externally, but is instead acquired by iterating once through the network as illustrated in Figure 3.

Output Function. First the activation is computed using the following cubic function:

$$(3a) \quad \mathbf{a}(t) = (\delta + 1)\mathbf{W}\mathbf{x}(t) - \delta(\mathbf{W}\mathbf{x})^3(t)$$

$$(3b) \quad \mathbf{b}(t) = (\delta + 1)\mathbf{V}\mathbf{y}(t) - \delta(\mathbf{V}\mathbf{y})^3(t)$$

Then, the output can be obtained using a piecewise function defined by the following equations:

$$(4a) \quad \forall i, \dots, N, \mathbf{y}_i(t+1) = \begin{cases} 1, & \text{If } \mathbf{a}_i(t) > 1 \\ -1, & \text{If } \mathbf{a}_i(t) < -1 \\ \mathbf{a}_i(t), & \text{Else} \end{cases}$$

$$(4b) \quad \forall i, \dots, M, \mathbf{x}_i(t+1) = \begin{cases} 1, & \text{If } \mathbf{b}_i(t) > 1 \\ -1, & \text{If } \mathbf{b}_i(t) < -1 \\ \mathbf{b}_i(t), & \text{Else} \end{cases}$$

where N and M are the number of units in each layer, i is the index of the respective vector element, $\mathbf{y}(t+1)$ and $\mathbf{x}(t+1)$ represent the layers' contents at time $t+1$, and δ is a general output parameter. This parameter should be fixed at $\delta < 0.5$ to assure fixed-point behavior (Chartier, Renaud and Boukadoum 2008).

Learning. Learning is based on time-difference Hebbian association (Chartier, Boukadoum & Amiri, 2010), and is formally expressed by the following equations:

$$(5a) \quad \mathbf{W}(k+1) = \mathbf{W}(k) + \eta(\mathbf{y}(0) - \mathbf{y}(t))(\mathbf{x}(0) + \mathbf{x}(t))^T$$

$$(5b) \quad \mathbf{V}(k+1) = \mathbf{V}(k) + \eta(\mathbf{x}(0) - \mathbf{x}(t))(\mathbf{y}(0) + \mathbf{y}(t))^T$$

where η is a learning parameter, $\mathbf{y}(0)$ and $\mathbf{x}(0)$, are the initial patterns at $t = 0$, $\mathbf{y}(t)$ and $\mathbf{x}(t)$ the state vectors after t iterations through the network (where typically $t = 1$), and k is the learning trial. The learning rule is thus very simple and constitutes a generalization of Hebbian/anti-Hebbian correlation in its auto-associative memory version (Chartier & Boukadoum 2006b). For weight convergence

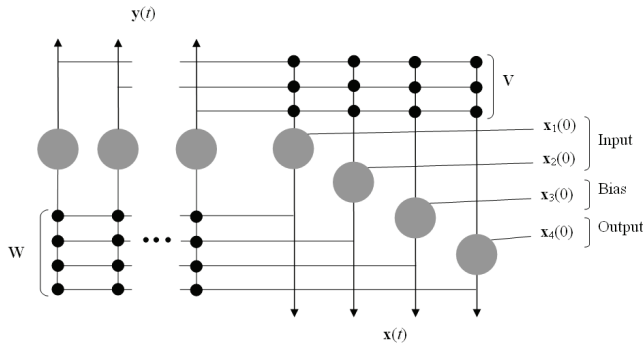


Figure 4. Architecture of the model in the context of learning a complex association task.

to occur, η must be set according to the following condition (Chartier, Renaud and Boukadoum 2008):

$$(6) \quad \eta < \frac{1}{2(1-2\delta)\text{Max}[N, M]}, \quad \delta \neq \frac{1}{2}$$

Equations 5a and 5b show that the weights can only converge when a layer's content at $t = 1$ is identical to the initial inputs, that is, $y(1) = y(0)$ and $x(1) = x(0)$. As a result, the learning rule is dynamically linked to the network's output. Of course, just like any BAM, for each stored pattern the model also stores its associated complement (i.e., $-y(0)$ and $-x(0)$).

FEBAM Architecture for Learning Complex Associations

The key to setting up a FEBAM network to learn complex associations is to structure it in a manner such that one (or more) of the x -layer units can be activated with desired target output values. After learning, the network should then be able to re-instantiate the target values on that unit whenever the appropriate partial activation pattern occurs on the other x -layer units (by using the y -layer units to remap sets of one-to-many relations into one-to-one associations). The FEBAM architecture that allows for the learning of such complex associations is shown in Figure 4. In that figure, the y -layer units act as “hidden” units, whereas three of the four units at the x -layer act as the “input” units, with one acting as a “bias” unit, and the fourth one as the “output” unit. If more complex associations are desired then the number of output and input units can be increased and the architecture adjusted as needed.

Simulations

First, the model (with the architecture depicted in Figure 4) will be trained to learn the *simplest* type of complex association, namely, the 2-bit parity problem illustrated in Table 1. Next, the model will then be trained on a more complex learning task (i.e., the 3-bit parity problem).

Table 1. Complex associations (2-bit parity problem).

X_1	X_2	X_3	X_4
-1	-1	1	-1
1	-1	1	1
-1	1	1	1
1	1	1	-1

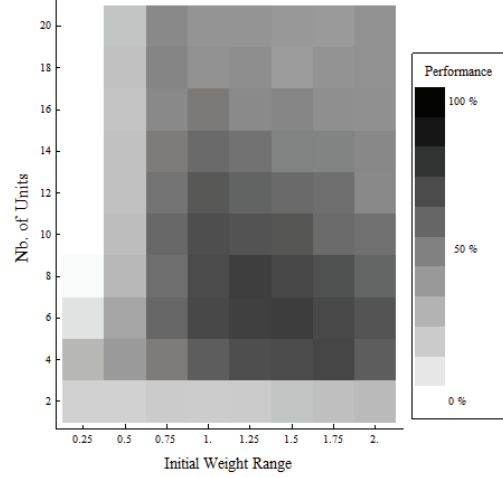


Figure 5. Performance as a function of the range of initial weight values and the number of units on the y -layer.

Complex Associations

An example of a complex association like the one depicted in Figure 1b is the 2-bit parity problem described in Table 1. Although this type of problem may be considered as the easiest of such associations, nevertheless no BAM (or Hopfield-type model) is currently able to perform it using solely covariance matrices (i.e., Hebbian learning). Moreover, solving this problem requires the model to take into account two variable values simultaneously. Such classification is analogous to that examined recently by Smith et al. (2004) who had humans and rhesus monkeys perform six types of category-learning tasks involving both linearly and non-linearly separable categories.

Methodology. The simulations were run using the following general procedure. All four bipolar input patterns in Table 1 were randomly presented once an epoch and the network was trained. The procedure for each learning trial within an epoch involved (a) setting the activations of the units on the x -layer of FEBAM to their desired values (i.e., the input, bias, and target output values) at $t = 0$, (b) running the network through one full iteration (as shown in Figure 3) to eventually obtain $y(0)$, $x(1)$, and $y(1)$, and then (c) applying the learning Equations 3a and 3b.

After each epoch, each of the four patterns was tested by setting the activations of the first three units on the x -layer of FEBAM to correspond to the input and bias values at $t = 0$ with the activation of the fourth x -layer unit now set to 0. Then it was determined whether the reconstructed

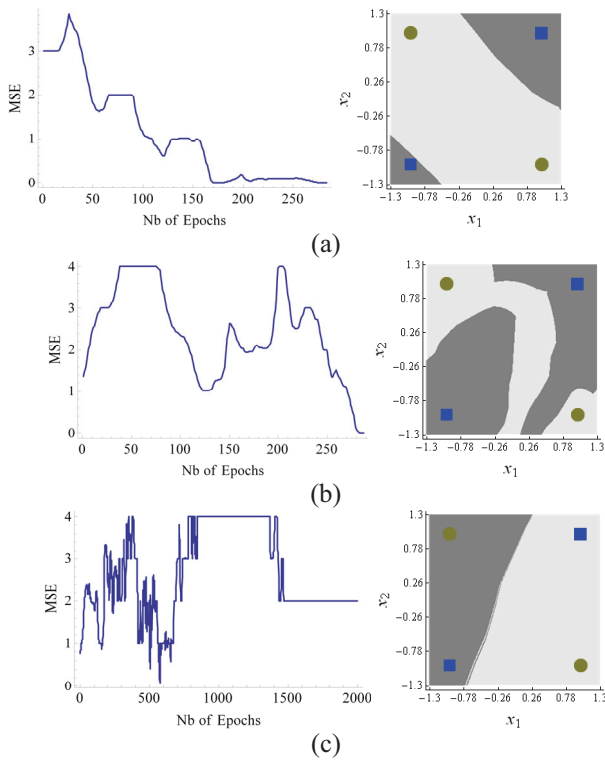


Figure 6. Three examples of MSE curves as a function of the number of epochs: (a) and (b) where the network was able to correctly accomplish the 2-bit parity association task, (c) where the network failed to accomplish the task.

activation on the x_4 unit at $t = 1$ was within .1 of the target value with an MSE very close to 0. Learning was deemed successful (and stopped) whenever the network was able to respond correctly to all four test patterns within 2,000 learning epochs. The learning parameter (η) and the transmission (δ) parameter were set to 0.005 and 0.1, respectively which satisfy the requirement of Equation 4.

In addition, the performance of the network as a function of the free parameters was assessed. More precisely, the number of units at the y -layer (i.e., the hidden units) was varied from 2 to 20 and the range of initial random weight values was varied from ± 0.25 to ± 2.0 .

Results. As depicted in the Figure 5, the maximum performance is obtained when 6 or 8 units of the y -layer are used and the range of initial random weight values is set to ± 1.25 . At this level, the network is successful about 75% of the time. Figure 6 illustrates some of the different behaviors expressed by the network. More precisely, in Figures 6a and 6b the network is able to find the correct association (MSE lower than 10^{-4}). However, the classification border found by the network (also given in Figure 6 for each solution) will vary from simulation to simulation. In contrast, Figure 6c illustrates an example of an unsuccessful simulation such that, even after 2000 epochs, the network was not able to find the correct solution (MSE stabilized at a value of 2).

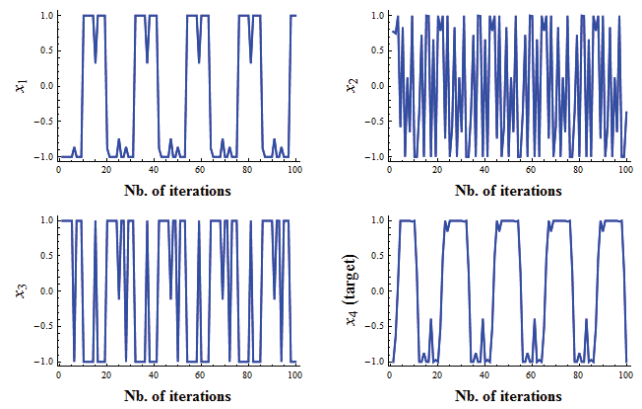


Figure 7. Example of long-term behavior of the model. In this example the input is $x_1=1$; $x_2=1$; $x_3=1$ and $x_4=0$. The network correctly outputs the desired association ($x_4=-1$) at $t=1$.

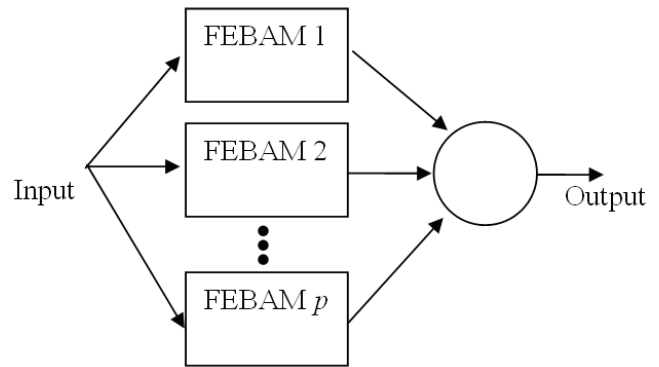


Figure 8. Block diagram of a committee machine.

Until now, the behavior of model has been studied after only one cycle through the network. However, because the model has a feedback loop, the network should converge, like any BAM, to steady states if the initial weight values are set to zero. If they are not initialized at values of zero, then a fixed point is not guaranteed. Since, the network performs well when its weights are initialized at some random value greater than zero (Figure 5) dynamic behavior must be studied. Thus, if the input of the network is allowed to iterate through the network until convergence ($t = c$), most of time the network will not stabilize to a fixed-point but rather will demonstrate a cyclic behavior as depicted in Figure 7. Therefore, the model is able to solve the one-to-many problem by modifying the inputs ($x(1)$) through the feedback given by the y -layer. In other words, the desired solution is a transient state rather than a fixed point.

Increasing Performance by Using a Committee Machine

There are several ways to increase the performance by modifying the architecture of the network. One possibility would be to put several FEBAMs in parallel (Figure 8) and make them work as a committee machine (Haykin, 1999).

Therefore, if the probability of one FEBAM finding the solution is $s = 0.75$, the overall committee machine probability of finding the solution is:

$$(7) \quad perf(p) = 1 - (1 - s)^p$$

where p represents the number of FEBAM networks that take part in the committee machine. For example, if $p = 4$ networks are linked together, the probability is 99.6% that at least one of FEBAM will solve the task. However, this will involve a more complex architecture with a controller that must allow only the FEBAM that has found the solution to give the output. In addition, this solution only works if a given model is able to perform the task. Although it works well for the previous task, we need to evaluate if it can be applied to a more complex problem.

More Complex Associations

Although the previous 2-bit parity task cannot be performed by any other BAMs, the task is still at the most basic level regarding complexity. Therefore, the model was also trained on the 3-bit parity task in order to see if it could learn that as well. As illustrated in Table 2, the 3-bit parity task is a composition of 2-bit parity classification.

Methodology. The simulations were run using the same general methodology just described for the 2-bit parity task. However, the size of the x -layer of the network was increased by one unit to allow for a third input unit. The performance of the network as a function of the free parameters was also assessed within the same range of parameter values.

Results. As depicted in Figure 9, the maximum performance is obtained when four y -units are used and the range of initial random weight values is set to ± 1.5 . At this level, the network is successful about 45% of the time. Therefore, the overall committee machine probability of finding the solution can be determined by Equation 7 with $s = 0.45$. To achieve the same probability of (near) perfect performance of 99.6%, the number of parallel FEBAMs must be increased from $p = 4$ to 9.

Discussion

Learning a complex association task is something that is difficult to do for BAM, given that it has some constraints not met by any other network models. First, the learning has to be performed using Hebbian-type rules only. In addition, the weight connections have to be updated online. Finally, both the inputs and the desired output have to explicitly be available to the network. Nonetheless, the present results have shown that complex association such as the 2- and 3-bit parity problem can indeed be learned within a BAM. Of course, feedforward neural networks (e.g., MLP, SVM) will typically perform better than FEBAM on such tasks because they do not have the same constraints. The fact that FEBAM can perform such

Table 2. Complex association (3-bit parity).

X_1	X_2	X_3	X_4	X_5
-1	-1	-1	1	-1
-1	-1	1	1	1
-1	1	-1	1	1
-1	1	1	1	-1
1	-1	-1	1	1
1	-1	1	1	-1
1	1	-1	1	-1
1	1	1	1	1

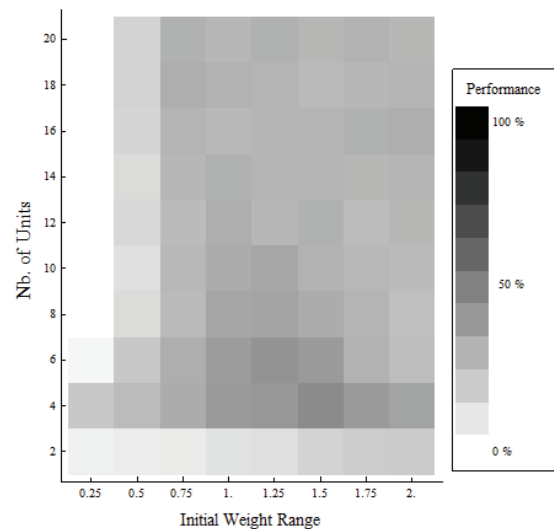


Figure 9. Performance of the model on the 3-bit parity task as a function of the range of initial weight values and the number of units on the y -layer.

associations represents what we believe to be an important step that now potentially opens the door for almost any kind of learning by BAM-type models.

However, as a model of cognition, the kind of properties displayed by any neural network is an important issue. In addition to more complex supervised learning, FEBAM is also able to perform blind source separation, perceptual feature extraction, input compression, signal separation, as well as demonstrate attractor-like behavior, such as prototype development and learning in noisy environments, and noisy recall. Furthermore, it can also create flexible clusters or categories and reorganize them through time. All these other properties are not shared by feedforward neural networks. Moreover, because FEBAM is a special case of BAM, it inherits many of its interesting properties such as aperiodic recall, many-to-one association, as well as multi-step pattern recognition, among others. Therefore, it seems that this type of network could indeed represent a good candidate for learning in larger-scale cognitive tasks. Hence, further studies with the FEBAM model presented here will be focused on the extent to which this model can reproduce other kinds of complex human learning behavior.

Acknowledgement

This research was supported in part by Natural Sciences and Engineering Research Council (NSERC) of Canada grants to both C. Leth-Steensen and S. Chartier.

References

- Arik, S. 2005. Global Asymptotic Stability Analysis of Bidirectional Associative Memory Neural Networks with Time Delays. *IEEE Transactions on Neural Networks* 16: 580-586.
- Chartier, S. and Boukadoum, M. 2006a. A sequential dynamic heteroassociative memory for multistep pattern recognition and one-to-many association. *IEEE Transactions on Neural Networks* 17: 59-68.
- Chartier, S. and Boukadoum M. 2006b. A bidirectional heteroassociative memory for binary and grey-level patterns. *IEEE Transactions on Neural Networks* 17: 385-396.
- Chartier, S. Boukadoum M. & Amiri, M. 2010. BAM Learning of Nonlinearly Separable Tasks by Using an Asymmetrical Output Function and Reinforcement Learning. *IEEE Transactions on Neural Networks* 20: 1281-1292.
- Chartier, S., Giguère, G., Renaud, P., Proulx, R. and Lina, J.-M. 2007. FEBAM: A feature-extracting bidirectional associative memory. In *Proceeding of the International Joint Conference on Neural Networks*, 1679-1684. IEEE.
- Chartier, S., Renaud, P. and Boukadoum, M. 2008. A nonlinear dynamic artificial neural network model of memory. *New Ideas in Psychology* 26: 252-277.
- Du, S., Chen, Z., Yuan, Z. and Zhang, X. 2005. Sensitivity to noise in bidirectional associative memory (BAM). *IEEE Transactions on Neural Networks* 16: 887-898.
- Giguère, G., Chartier, S., Proulx, R. and Lina, J.-M. 2007a. Creating perceptual features using a BAM-inspired architecture. In *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, 1025-1030. D. S. McNamara and J. G. Trafton, Eds. Austin, TX: Cognitive Science Society.
- Giguère, G., Chartier, S., Proulx, R. and Lina, J.-M. 2007b. Category development and reorganization using a bidirectional associative memory-inspired architecture. In *Proceedings of the 8th International Conference on Cognitive Modeling*, 97-102. R. L. Lewis, T. A. Polk, and J. E. Laird, Eds. Ann Arbor, MI: University of Michigan.
- Haykin, S. 1999. *Neural Networks: a comprehensive foundation*. New-Jersey: Prentice-Hall.
- Hopfield, J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the Natural Academy of Sciences (U.S.A.)* 79: 2554-2558.
- Kosko, B. 1988. Bidirectional Associative Memories. *IEEE Transactions on Systems, Man and Cybernetics* 18: 49-60.
- Leung, C.-S. 1994. Optimum learning for bidirectional associative memory in the sense of capacity. *IEEE Transactions on Systems, Man and Cybernetics* 24: 791-795.
- Shen, D. and Cruz Jr, J. B. 2005. Encoding Strategy for Maximum Noise Tolerance Bidirectional Associative Memory. *IEEE Transactions on Neural Networks* 16: 293-300.
- Smith, J. D., Minda, J. P., & Washburn, D. A. 2004. Category learning in rhesus monkeys: a study of the Shepard, Hovland, and Jenkins (1961) tasks. *Journal of Experimental Psychology: General* 133: 398-414.
- Wang, Z. 1996. A bidirectional associative memory based on optimal linear associative memory. *IEEE Transactions on Computers* 45: 1171-1179.