Bidirectional Associative Memory and Learning of Nonlinearly Separable Tasks

Christophe Tremblay (ctrem040@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier
Ottawa, ON K1N 6N5 Canada

Kaia Myers-Stewart (kmyer080@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier
Ottawa, ON K1N 6N5 Canada

Laurence Morissette (lmori096@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier
Ottawa, ON K1N 6N5 Canada

Sylvain Chartier (sylvain.chartier@uottawa.ca)
School of Psychology, 136 Jean-Jacques Lussier
Ottawa, ON K1N 6N5 Canada

Abstract
Research has shown that human beings are able to perform various associations, whether linearly separable or nonlinearly separable, with little effort. While bidirectional associative memory (BAM) models show great promise in modeling various types of associations the humans perform, they still have difficulties with solving various types of nonlinearly separable problems. The present study introduces a modification of the architecture of a given type of BAM by adding an unsupervised pathway to the original BAM structure. Results showed that the modification allows the network to perform nonlinearly separable associations such as the n-bit parity task and the double-moon problem. The network is able to associate more difficult types of problems while keeping the same learning and transmission function. This study could lead to enhanced cognitive models capable of modeling a wider set of associations.

Keywords: Artificial intelligence; Connectionist models; Bidirectional associative memory; Nonlinearly separable problems.

Introduction
Humans are exposed to situations in which they must discriminate, identify, classify and categorize perceptual patterns on a daily basis, and this is done with little effort. In the field of cognition, much research has been done on non-linearly separable tasks. The ease with which participants learned linearly separable or nonlinearly separable categories is a function of the knowledge structures applied to the task that create differences in integration strategies, making human beings naturally able to perform both types of tasks (Wattenmaker, Dewey, Murphy & Medin, 1986; Wattenmaker, 1995; Nadler, Minda & Lin, 2008).

This type of separation is easily accomplished by many machine learning techniques such as in multi-layer Perceptrons, Radial Basis Function, Cascade Correlation, Support Vector Machine, Deep Belief Network, etc. Although those networks are able to achieve the desired tasks, their internal properties make them ill suited as good cognitive models. For example, a good cognitive model should adapt its weight connections locally, without homunculus knowledge, backpropagation of the error, complex learning procedure or a highly nonlinear transmission function (O'Reilly, 1998).

Bidirectional Associative Memories (BAMs) have been proposed as models of neurodynamics. As such, they are able to develop attractors that correspond to desired associations. This allows them to be noise tolerant and able to perform pattern completion. Over the years, many studies have proven BAMs to be capable of performing multi-step patterns recognition, learning real-valued correlated patterns, having increased storage capacities (Chartier & Boukadoum, 2006, 2011) and extracting features from given inputs (Chartier, Giguère, Renaud, Lina & Proulx, 2007).

Despite the advancements made since the introduction of BAM models by Kosko (1988), BAM models have difficulties learning nonlinear separable tasks. The problem comes from the fact that these networks do not have a hidden layer or a highly nonlinear transmission function. Although BAMs could be modified to use backpropagation of the error through descent of gradient (Zheng, Zhang & Tang, 2011) in order to solve nonlinearly separable problems, such modification would make these models lose their property of being models of human categorization. Over the years few attempts were made to enable a BAM to solve nonlinear separable tasks without modifying their underlying properties. In Chartier, Boukadoum & Amiri (2009), a BAM model was able to solve a simple nonlinear separable task (XOR) by decomposing it into linear tasks first. However, not all nonlinear separable tasks can be divided as such, which imposes some limitation on the applicability of the model to more complex tasks. Another solution was to use a modified BAM architecture, the Feature Extracting Bidirectional Associative Memory (FEBAM) (Chartier, & al., 2007). In this model, external
connections of one layer were removed making the model's learning process unsupervised. By connecting two FEBAM models in a head-to-head fashion the network were able to supervise each layer through the sharing of a hidden layer (Chartier, Leth-Steensen & Hébert, 2012). The learning and transmission functions remained constant (i.e, hebbian-type and sigmoid-type respectively). The resulting network could perform more difficult tasks, such as the 3, 4 and 5-bit parity problems. However, the model was unable to perform real valued inputs problems such as the double-moon task; described in the simulation section.

Despite the improvements mentioned previously, no BAM model is yet capable of solving a wide range of non-linearly separable tasks (binary and real-valued). The present study introduces a modification of the BAM by the inclusion of an unsupervised layer (FEBAM) acting as a hidden layer. The architecture is therefore modified through the addition of a pathway between the BAM of Chartier & Boukadoum (2006; 2011) and the FEBAM (Chartier & al., 2007), allowing the two networks to work together. The remainder of the paper is divided as follows. Section II describes the architecture of the network, while section III presents simulation results regarding the performance of the network on non-linearly separable tasks: 1) the n-bit parity task; and 2) the double-moon task. Section IV discusses the results and provides a conclusion of the study.

**Background**

Associative memories are artificial neural networks that model the recognition and recall of patterns of various natures and different contexts. These brain inspired recurrent associative memories offer the ability to develop attractors for each pattern through feedback connections such as the Hopfield model (Hopfield, 1982). Kosko (1988) later generalized the Hopfield model to perform heteroassociation, thereby creating a new class of neural network models, the bidirectional associative memories (BAMs). BAMs display attractor-like behaviour and have the advantage of acting as an autoassociative and heteroassociative memories. BAM models are generally deemed to be more biologically plausible and dynamically complex than many other types of neural network models. Over the years, numerous BAM models have been developed since Kosko's (e.g. Eom, Choi & Lee, 2002; Shi, Zhao & Zhuang, 1998; Xu, Leung & He, 1994; Zhuang, Huang & Chen, 1993). For a review of BAM models, see Acevedo-Mosqueda, Yanez-Marquez & Acevedo-Mosqueda (2013).

**Bidirectional Heteroassociative Memory**

One model proposed by Chartier & Boukadoum (2006; 2011) was the introduction of a unique matrix for each layer. This Bidirectional Heteroassociative Memory (BHM) is able to learn correlated patterns for bipolar patterns as well as for real-valued patterns.

**Architecture**

The network is made of two Hopfield-like neural networks interconnected in a head-to-tail fashion, providing a recurrent flow of information that is processed bidirectionally. The network's architecture is shown in Fig.1 where \( x(0) \) and \( y(0) \) represent the initial vectors-states, \( W \) and \( V \) are the weight matrices and \( t \) is the current iteration number.

**Transmission Function**

The transmission function is defined by the following equations:

\[
\forall i, \ldots, N, y_{l+1} = f(a_{l+1}^i) = \begin{cases} 
1, & \text{if } a_{l+1}^i > 1 \\
-1, & \text{if } a_{l+1}^i < -1 \\
(\delta + 1)a_{l+1}^i - \delta a_{l+1}^3, & \text{else}
\end{cases}
\]

and

\[
\forall i, \ldots, M, x_{l+1} = f(b_{l+1}^i) = \begin{cases} 
1, & \text{if } b_{l+1}^i > 1 \\
-1, & \text{if } b_{l+1}^i < -1 \\
(\delta + 1)b_{l+1}^i - \delta b_{l+1}^3, & \text{else}
\end{cases}
\]

where \( N \) and \( M \) are the number of units in each layer, \( i \) is the unit index, \( \delta \) is a general transmission parameter and \( a \) and \( b \) are the activation. These activations are obtained the usual way: \( a(t) = Wx(t) \) and \( b(t) = Vy(t) \). Figure 2 illustrates the shape of the transmission function for \( \delta = 0.2 \). In short, the equation consists of a cubic function with two hard limits added at 1 and -1. Contrary to sigmoid-type function, there are no asymptotic behaviors in the transmission. This
function has the advantage of exhibiting grey-level attractor behaviour which contrasts with other BAMs that can only develop bipolar attractors (Chartier & Boukadoum, 2006).

**Learning Rule** The weight connections are modified following a Hebbian/anti-Hebbian approach (Storkey & Valabregue, 1999; Bégin & Proulx, 1996):

\[
\begin{align*}
W_{(k+1)} &= W_{(k)} + \eta(y_{(0)} - y_{(t)})x_{(0)} \quad \text{for the BHM layer} \\
V_{(k+1)} &= V_{(k)} + \eta(x_{(0)} - x_{(t)})y_{(0)} + y_{(t)} \quad \text{for the FEBAM layer}
\end{align*}
\]

where \(x_{(0)}\) and \(y_{(0)}\) are the initial inputs to be associated, \(\eta\) is the learning parameter, and \(k\) is the learning trial number. Equation (2) shows that the weight matrices will converge only when \(x_{(0)} = x_{(t)}\) or \(y_{(0)} = y_{(t)}\). Thus, each weight matrix converges when the feedback is equal to the initial inputs. In the BHM, the network convergence is guaranteed if the learning parameter \(\eta\) is smaller than the threshold found with:

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

where \(M\) and \(N\) are respectively the dimensionality of the input and its association (Chartier & Boukadoum, 2006). The \(\eta\) parameter was set to a lower value than the threshold found in (3) for every simulation performed.

**Architecture Modifications Through the Inclusion of the FEBAM-Layer** In order to perform nonlinearly separable associations, a hidden layer was added to the original BHM. Figure 3 presents the architecture of the resulting network. The hidden layer acts like a FEBAM network, (Chartier, & al., 2007). The FEBAM is identical to the BHM except that the external connections of the y-layer have been removed. Therefore, the initial output is obtained by iterating once through the network as illustrated in figure 4.

**Simulations**

The simulations are divided in two sections. The first set of simulations assessed the capability of the new architecture to solve the classic XOR problem and more difficult \(n\)-bit parity problems. The second set of simulations evaluated the capacity of the network to classify the double-moon problem at various difficulty levels. Contrary to the \(n\)-bit parity problem, the input patterns for the double moon problem are real valued and thus present a challenge for BAMs.

**Simulation 1: Solving the \(n\)-bit Parity Problem**

This first set of simulations aimed to replicate the performance of the previous BAM-types on a binary (bipolar) classification task. The task is the classic XOR and its generalization, the \(n\)-bit parity task.

Table 1: The 2-bit parity problem.

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

**Methodology** The architecture used is described in figure 4. Table 1 illustrates the patterns to be associated for the 2-bit (XOR) situation. Learning was carried out according to the following procedure:

1) Random selection of a pair of patterns \((x(0)\) and \(y(0))\).
2) Computation of \(x(t)\) and \(y(t)\) according to the transmission function (1).
3) Computation of the weight matrices update according to (2).
4) Repetition of steps 1 to 3) until all of the pairs have been presented.
5) Repetition of steps 1 to 4) for a given number of trials, to be set prior to training.

The transmission function parameter \(\delta\) was set to 0.2 throughout the simulations and the number of iterations to perform by the network before the weight matrices are updated was set to \(t = 1\). The initial weight activation was set to 0 for the BHM layer and randomly activated between -2 and 2 for the FEBAM layer. The number of units of the FEBAM module was varied from 2 to 50. The network was tested 200 times for a given problem and average performances are reported. Finally, the number of bit tested varied from 2 to 5.

Following the learning phase, the network was tested on a recall task according to the following procedure:
1) Selection of an input pattern \( x(0) \).
2) Computation of \( y(1) \) according to the transmission function (1).
3) Comparison with the target value \( y(0) \)
4) Repetition of steps 1) to 3) until all of the patterns have been presented.

The performance is obtained by computing the number of correct classifications over the total number of patterns to classify.

**Results** Figure 5 illustrates the relationship between the number of units in the FEBAM layer and the quality of the output for the 2-bit parity task. The 5 hidden units network was able to solve the task. However, with more units (50), the solution is found in a smoother fashion and the output map closely matches the outputs found with multilayer Perceptrons. Adding more units (150) on the hidden layer do not have a noticeable impact on the convergence of the network or the quality of separation.

Figure 6 shows the results of the performances in function of the number of units on the FEBAM layer. The network is able to achieve perfect performance regardless of the difficulty of the task. However, in order to solve more complex associations, the network will need more units in the FEBAM module. In comparison to the results found in Chartier, Leth-Stensen & Hebert (2011), the new architecture requires about half the number of units to reach similar performances.

**Simulation 2: Solving the double-moon problem**

In this set of simulations the network must perform the double-moon classification problem (Haykin, 2009). This type of classification is different from the \( n \)-bit parity since the inputs are continuous instead of being binary (bipolar). Currently, there are no BAM models able to solve the double-moon problem.

**Methodology** The general methodology followed the one described in the previous section. The double-moon classification task is illustrated in Figure 7. The network was tested on different level of difficulty, \( d \) from 1 to 3; the lower the value of \( d \), the easier the task is. If the value of \( d \) is set equal or lower to 0, then the problem is linear. The network was tested 200 times for each level of difficulty. Finally, the number of FEBAM units was varied from 5 to 25.
Results Figure 8 presents the convergence and the mapping of the final separation made by the new architecture on a double-moon task with the $d$ parameter set to 2 for various numbers of units on the hidden layer (FEBAM units). Contrary to the $n$-bit parity task the error curve is much more rough, and this holds up even as we increase the number of FEBAM units. Moreover, the resulting separations made by the network seem to soften slightly which could lead to reduced performances as the $d$ parameter is increased, i.e. the difficulty of the problem.

Figure 9 presents the performances of the new architecture on the double-moon problem. The network was generally able to accomplish the task perfectly without any errors. However, on some occasions, the separation was not perfect, which lead to reduced performances. As the figure shows, in every condition the network was able to separate more than 95% of the input patterns. As would be expected, more difficult problems lead to lower performances. However, unlike in the previous set of simulations, more units on the hidden layer do not lead to increased performances. In fact, the performances decrease as the number of units is increased.

Discussion and Conclusion

The results showed that the modified architecture is capable of performing nonlinearly separable associations with both binary and real-valued inputs. For the 2-bit parity task, the present model had similar results than other models (Chartier, Boukadoum & Amiri, 2009, where the network was capable of solving the XOR problem in addition to more difficult $n$-bit problems. Moreover, for the $n$-bit parity task, similar results were obtained (Chartier, Leth-Steensen & Hébert, 2012). However, the current hybrid model outperforms both models in that it is able to adequately classify nonlinearly separable problems that comprise real-valued inputs.

The modified BHM shows great performances on the $n$-bit parity problem, where it could consistently separate all of the input patterns, no matter the difficulty of the task, given enough units on the hidden layer. However, the network could not be as consistent on the double-moon problem where in no condition could the network consistently separate all of the input units. A potential explanation could be that the network needs more hidden layers, as opposed to more hidden units, in order to solve more difficult problems (Fahlman & Lebiere 1990). Future research should also evaluate the performance of the modified BHM on other nonlinear tasks such as $n$-label problems and regression problems.

This study presents a biologically based computational model (O’Reilly, 1998) capable of solving nonlinearly separable problems. By presenting a new BAM architecture capable of solving nonlinearly separable problems, this study showed that BAM models are well suited to perform a wide range of problems and could lead to better cognitive models capable of modeling a wider set of associations.

Acknowledgments

This research was partly supported by the Natural Sciences and Engineering Research Council of Canada.

References


