A new bidirectional heteroassociative memory encompassing correlational, competitive and topological properties

Sylvain Chartier\textsuperscript{a,}\textsuperscript{*}, Gyslain Giguère\textsuperscript{b,1}, Dominic Langlois\textsuperscript{a,2}

\textsuperscript{a} University of Ottawa, School of Psychology, 200 Lees E217, 125 University Street, Ottawa, ON K1N 6N5, Canada
\textsuperscript{b} The University of Texas at Austin, Department of Psychology, 1 University Station, A8000, Austin, TX 78712-0187, USA

\textbf{A R T I C L E I N F O}

\textbf{Article history:}
Received 6 May 2009
Received in revised form 26 May 2009
Accepted 25 June 2009

\textbf{Keywords:}
Bidirectional heteroassociative memory
Competitive learning
Topological architecture
Unsupervised learning
Categorization
Self-organizing map

\textbf{A B S T R A C T}

In this paper, we present a new recurrent bidirectional model that encompasses correlational, competitive and topological model properties. The simultaneous use of many classes of network behaviors allows for the unsupervised learning/categorization of perceptual patterns (through input compression) and the concurrent encoding of proximities in a multidimensional space. All of these operations are achieved within a common learning operation, and using a single set of defining properties. Moreover, because the model is recurrent, it can reconstruct perfect outputs from incomplete and noisy patterns. Empirical exploration of the model’s properties and performances show that its ability for adequate clustering stems from: (1) properly distributing connection weights, and (2) producing a weight space with a low dispersion level (or higher density). In addition, since the model uses a sparse representation ($k$-winners), the size of topological neighborhood can be fixed, and no longer requires a decrease through time as was the case with classic self-organizing feature maps. Since the model’s learning and transmission parameters are independent from learning trials, the model can develop stable fixed points in a constrained topological architecture, while being flexible enough to learn novel patterns.

\textcopyright 2009 Elsevier Ltd. All rights reserved.

\section{1. Introduction}

\subsection{1.1. Cognitive science background}

Every day, humans are exposed to situations in which they are required to either differentiate or regroup perceptual patterns (such as objects presented to the visual system). Their perceptual/cognitive system achieves these operations in order to produce appropriate responses, upon the identity and properties of the encountered stimuli. To accomplish these perceptual tasks, the system must create and enrich context-dependent memory representations, which are adapted to different environments, but can be shared between these environments through generalization. This general process, known as perceptual learning, mainly consists in the implicit abstraction of previously unavailable information, leading to semi-permanent changes at the memory structure level (Gibson & Gibson, 1955). Most perceptual learning processes can be achieved autonomously, through the associative abstraction of the environment’s statistical structures (as some neural networks do: Goldstone, 1998; Hall, 1991).

One of the system’s main goals in defining mental representations is cognitive economy (Goldstone & Kersten, 2003). Invariance is a quality that leads to economy by reducing the quantity of information that the system must take into account in a given situation, thus accelerating cognitive processes (or more precisely, retrieval of memory traces). Hence, in order to be efficient, the human perceptual/cognitive system must create representations including relevant statistical properties leading to quick differentiation, identification and recognition (Goldstone, 1998). These representations will be useful in future situations, when the need for a decision based on a new or repeating perceptual stimulation arises. Strictly memorizing invariant information lessens the computational burden on the cognitive system, and allows it to follow an information reduction strategy. At the object level, this strategy would be found in the form of a dimensional reduction applied to inputs (Edelman & Intrator, 1997).

Another strategy used by the system to reduce information and increase processing speed is the representation of similar perceptual stimulations by a single higher-level entity, created according to the common perceptual properties of the member objects. This process is called perceptual category learning (Murphy, 2002). Cognitive scientists have historically argued over the fact...
that the human cognitive system either uses generic abstractions (such as prototypes) or very specific perceptual stimulations (such as complete exemplars) to achieve category learning and further classification (Komatsu, 1992). Prototype supporters believe that categories possess a special representation status within the cognitive system. Hence, every exemplar set is linked to a more generic, summary representation that can be retrieved independently, and for which the memory trace is stronger and more durable than for any exemplar (Posner & Keele, 1968, 1970). These representations help the system in quickly deciding on a stimulus’ category membership, and are enhanced according to new incoming information. To achieve this, categories must be easy to differentiate; this is why the natural world is composed of cohesive categories, in which associated exemplars are similar to each other (Rosch, 1973).

Following the work of Knowlton and Squire (1993) and Knowlton, Mangels, and Squire (1996), we now know that the prototype theory is valid in categorization. Neuropsychological dissociation studies have led to the conclusion that while single object representations must be memorized in order to achieve recognition, identification and discrimination, categorical representations are completely separate in the system, and operate according to a prototype principle.

1.2. Modeling background

1.2.1. Recurrent/Bidirectional associative memories

When trying to achieve autonomous (or unsupervised) learning and categorization, many neural network options are available. A class of networks known to achieve these types of tasks is that of recurrent autoassociative memories (RAMs). In psychology, AI and engineering, autoassociative memory models are widely used to store correlated patterns. One characteristic of RAMs is the use of a feedback loop, which allows for generalization to new patterns, noise filtering and pattern completion, among other uses (Hopfield, 1982). Feedback enables a given network to shift progressively from an initial pattern towards an invariable state (namely, an attractor).

A problem with these models is that contrary to what is found in real-world situations, they store learned information using noise-free versions of the input patterns. In comparison, to overcome the possibly infinite number of stimuli stemming, for instance, from multiple perceptual events, humans must regroup these unique inputs into a finite number of stimuli or categories. Also, while RAMs can perfectly reconstruct learned patterns through an iterative process, they cannot associate many distinct inputs with a single representation, that is they cannot categorize.

Direct generalization of RAM models is the development of bidirectional associative memory (BAM) models (Kosko, 1988). BAMs can associate any two data vectors of equal or different lengths (representing for example, a visual input and a prototype or category). These networks possess the advantage of being both autoassociative and heteroassociative memories (Kosko, 1988) and therefore encompass both unsupervised and supervised learning. A BAM model would be able to develop prototype representations linked to different exemplars, but only in a supervised fashion. Nowadays, many RAM/BAM models can display both stability and plasticity for a data set of exemplars (e.g. Davey and Hunt (2000) and Du, Chen, Yuan, and Zhang (2005)).

1.2.2. Competitive networks

Overall, category formation in a perceptual framework is often seen as a process akin to classic clustering techniques, which involve partitioning stimulus spaces in a number of finite sets, or clusters (Ashby & Waldron, 1999). In cognitive modeling, competitive networks (e.g. Grossberg (1988) and Kohonen (1982)) are known for their capacity to achieve clustering behavior. In fact, they constitute local, dynamic versions of clustering algorithms. In these models, each output unit represents a specific cluster. When taking decisions, the association between an exemplar and its determined cluster unit in the output layer is strengthened. In winner-take-all (WTA) networks (Grossberg, 1988; Kohonen, 1982), exemplars may only be associated with one cluster (i.e. only one output unit at a time can be activated).

An example of one such hard competitive framework is that of the Adaptive Resonance Theory (ART: (Grossberg, 1988)). ART networks possess the advantage of being able to deal effectively with the exemplars/prototype scheme, while solving the stability/plasticity dilemma. These unsupervised models achieve the desired behavior through the use of a novelty detector (using vigilance). Various degrees of generalization can be achieved by this procedure: low vigilance parameter values lead to the creation of broad categories, while high values lead to narrow categories, with the network ultimately performing exemplar learning.

Another example of this framework is the self-organizing feature map (SOFM: Kohonen, 1982), which, in addition to showing WTA behaviors, uses a topological representation of inputs and outputs. Although SOFMs only consider one active output unit at a time, the learning algorithm also allows for physically close neighboring units to update their connection weights. In a SOFM, an exemplar may thus, for instance, be geometrically positioned between two clusters, and possess various degrees of membership.

An extension of the WTA principle selects the k largest outputs from the total n outputs (Majani, Erlanson, & Abu-Mostafa, 1989). This k-winners-take-all (kWTA) rule is thus a more general case of the WTA principle, within which exemplars may be associated with many clusters at differing degrees. This procedure provides a more distributed classification.

While extremely useful for object and category processing, competitive networks do not achieve recurrent behavior. They are thus generally sensitive to input noise during recall. ART networks can deal with noise through a novelty detection procedure, but do not encompass topological properties. In comparison, SOFMs show topological properties, but are not resistant to noise and do not show plasticity.

1.3. Goals and presentation

In the present paper, we propose a new bidirectional heteroassociative memory (BHM), named BHM + SOFM, which encompasses kWTA and SOFM properties. This modification will enable a known BHM model (Chartier & Boukadoum, 2006a) to increase its clustering capacity; hence, higher network performance and readability should follow. In addition, this kWTA–SOFM version of a BHM will allow for sparse coding, which is a distributed representation principle supported by neuropsychological findings (Olshauser & Field, 2004). The network will also inherit properties from its recurrent memory status, such as attractor development and noise tolerance (Hassoun, 1989). Finally, using sparse coding, we will propose a modification to the original network (Chartier, Giguère, Langlois, & Sioufi, in press) that will enable it to solve a simple version (exemplar data set) of the stability–plasticity problem (Grossberg, 1987). Therefore, the goal of this study is to propose a general model based on BAMs, that shows properties similar to those found in other network types, such as competitive behavior in SOFMs.
2. Theoretical background: SOFMs and other general competitive models

Competitive models are based on a linear output function defined by:

\[ y = Wx \]  

(1)

where \( x \) is the original input vector, \( W \) is the weight matrix, and \( y \) is the output. Weight connection modifications follow an Hebbian learning principle with forgetting term \( g(y_j)w_j \), where \( g(y_j) \) must satisfy the following requirement:

\[ g(y_j) = 0 \quad \text{for} \quad y_j = 0. \]  

(2)

Therefore, the learning rule can be expressed by

\[ w_j(p + 1) = w_j(p) + \eta(p)y_jx - g(y_j)w_j(p) \]  

(3)

where \( w_j \) represents neuron \( j \)'s weight vector, \( \eta \) represents a learning parameter, and \( p \) is the learning trial number. The specific choice of function \( g(y_j) \) can lead to two different unsupervised model rules. Eq. (3) will lead to Kohonen (1982)'s competitive learning rule if:

\[ g(y_j) = \eta y_j \]  

(4)

and to a PCA learning rule (Oja, 1982) if:

\[ g(y_j) = y_j^2 \quad \text{and} \quad \eta(p) = \eta. \]  

(5)

Hence, both PCA and competitive neural networks constitute special cases of Hebbian learning using a forgetting term.

In a WTA perspective, the largest output must be selected. This allows for the detection of the winning unit, and of the topological neighborhood center's position for SOFMs. This is equivalent to allowing the detection of the winning unit, and of the topological neighborhood center's position for SOFMs.

3. Model properties

Like any artificial neural network, the proposed model, BHM + SOFM, can be entirely described by its architecture, as well as its transmission function and learning rule.

3.1. Architecture

The BHM + SOFM architecture (Fig. 1) is closely linked to that of the Bidirectional Heteroassociative Memory (BHM) proposed by Chartier and Boukadoum (2006a). It extends a previous effort by Chartier and Giguère (2008); see also: Chartier, Giguère, Renaud, Proulx, and Lina (2007); Giguère, Chartier, Proulx, and Lina (2007a) and Giguère, Chartier, Proulx, and Lina (2007b)). The architecture consists of two Hopfield-like neural networks interconnected in head-to-toe fashion (Hassoun, 1989). When connected, these networks allow a recurrent flow of information that is processed bidirectionally. As in a standard BAM, both layers serve as a “teacher” for the other layer, and the connections are explicitly depicted in the model.

The model thus includes two layers, \( x \) and \( y \). The \( x \) layer serves as an initial network input, as well as for input reconstruction. Its size is equal to the number of the original input images. The \( y \) layer serves as an information compression layer; it applies and memorizes a dimensional reduction mapping (Chartier et al., 2007). Here, the compression is in fact a topological transformation into a 2D space. The compression level depends on the number of winning units used for layer \( y \) (Chartier & Giguère, 2008).

As with the BHM, two separate weight matrices are used, in order to represent the possibility of asymmetric connections, and thus creating two distinct memories. Matrix \( W \) (the compression matrix) links layer \( x \) to layer \( y \), and is used to determine a mapping allowing the transformation of inputs into their topologically-compressed version. Matrix \( V \) (the reconstruction matrix) links back layer \( y \) to layer \( x \) and allows the network to reconstruct the original pattern using a nonlinear combination of the learned components (or filters). Here, the main difference between the present model and Chartier and Boukadoum (2006a)'s BHM lies in the absence of an “external” input for layer \( y \). In general, a BAM (such as the BHM) is used to associate two sets of known vectors; for each vector \( x \), a pre-associated vector \( y \) must also be provided. In the present case, there is no initial input for layer \( y \). This layer’s content will be created during the original compression process, in which an original pattern will be associated with its own topologically reduced version.

Here, the network’s compression layer \( y \) is further constrained using a rectangular 2D topology; this enables the network to behave like a SOFM. A SOFM not only categorizes the input data, ...
but also recognizes which input patterns are nearby each other in stimulus space. In order to achieve this, the SOFM's original topological neighborhood function was extended into a “Mexican-hat” function (Fig. 2). This function is well-suited to bipolar stimuli and is described by:

$$h_{i,j}(p) = \frac{1}{2} \left( 3e^{-\frac{y_i^2(x_i)}{\sigma^2(p)}} - e^{-\frac{y_i^2(x_i)}{4\sigma^2(p)}} \right); \quad d_{i,j}^2 = \|r_j - r_i\|^2$$  \hspace{1cm} (9)

where $r_j$ defines the position of neuron $j$, $r_i$ defines the position of winning neuron $i$, $p$ represents the learning trial number, and $\sigma$ represents neighborhood size. The size of the topological neighborhood on trial $p$ is also required to exponentially decrease with time:

$$\sigma(p) = \sigma_0 \exp \left(-\frac{p}{\tau} \right)$$  \hspace{1cm} (10)

where $\tau$ represents a time constant, and $\sigma_0$ represents the initial value for $\sigma$.

In order to emulate kWTA processes, the compressed ($y$) layer’s architecture must be constrained using inhibitory connections from each compression unit to all units from that same layer. As more and more $y$ units become active (i.e. going from a WTA to a kWTA-type setting), the network will evolve towards a more distributed (or “sparse”) type of representation (Chartier & Giguère, 2008).

The learning process used by the new model can thus be described as follows: the network must first compute the product between $W$ and initial input $x(0)$, and transform the result according to the transmission function (described in the next section). This allows the model to determine the initial activation of each unit in layer $y$. Then, the $k$ most activated $y$ units are chosen through the Mexican-hat inhibitory process (Eq. (9)), and the initial compression $y(0)$ is computed. This topologically-compressed version goes through matrix $V$, in order to produce final reconstruction $x(1)$. Finally, the reconstruction goes through $W$ once again, producing final compression $y(1)$. The $W$ weight matrix will be used to memorize the mappings between the input and the associated constrained representation, while the $V$ connection matrix will be used to reconstruct inputs using compressed and constrained versions.

3.2. Transmission function

The transmission function is expressed by the following equations:

$$\forall i, \ldots, N, \quad y_i(t + 1) = \begin{cases} 1, & \text{if } Wx_i(t) > 1 \\ -1, & \text{if } Wx_i(t) < -1 \\ (\delta + 1)Wx_i(t) - \delta(Wx_i(t))^3, & \text{else} \end{cases} \hspace{1cm} (11)$$

and

$$\forall i, \ldots, M, \quad x_i(t + 1) = \begin{cases} 1, & \text{if } Vy_i(t) > 1 \\ -1, & \text{if } Vy_i(t) < -1 \\ (\delta + 1)Vy_i(t) - \delta(Vy_i(t))^3, & \text{else} \end{cases} \hspace{1cm} (12)$$

where $N$ and $M$ are the number of units in layers $x$ and $y$ respectively, $i$ is the index of the respective vector element, $y(t)$ and $x(t)$ represent layer contents at iteration cycle $t$, $W$ is the compression matrix, $V$ is the reconstruction matrix, and $\delta$ is a general positive transmission parameter that should be fixed at a value lower than 0.5 to assure fixed-point behavior (Chartier & Proulx, 2005; Hélie, 2008). This function possesses the advantage of exhibiting continuous-valued (gray-level) attractor behavior (for a detailed example see Chartier and Boukadoum (2006a)). Such properties contrast with networks using a standard nonlinear output function, which can only exhibit bipolar attractor behavior (e.g. Kosko (1988)).

3.3. Learning rule

Learning follows correlational principles. It is based on time-difference Hebbian association (Chartier & Boukadoum, 2006a; Chartier, Renaud, & Boukadoum, 2008; Chartier & Proulx, 2005; Kosko, 1990; Oja, 1989; Sutton, 1988), and is formally expressed by the following equations:

$$W(p + 1) = W(p) + \eta(y(0) - y(t))(x(0) + x(t))^T$$  \hspace{1cm} (13)

$$V(p + 1) = V(p) + \eta(x(0) - x(t))(y(0) + y(t))^T$$  \hspace{1cm} (14)

where $\eta$ represents a learning parameter, $y(0)$ and $x(0)$, the initial patterns at $t = 0$, $y(t)$ and $x(t)$, the state vectors after $t$ iterations through the network, and $p$ the learning trial. The learning rule is thus very simple, and can be shown to constitute a generalization of Hebbian/anti-Hebbian correlation in its autoassociative memory version (Chartier & Boukadoum, 2006a; Chartier & Proulx, 2005). For weight convergence to occur, $\eta$ must be set according to the following condition (Chartier et al., 2008):

$$\eta < \frac{1}{2(1 - 2\delta)\text{Max}[N, M]}, \quad \delta \neq 1/2.$$  \hspace{1cm} (15)

Eqs. (13) and (14) show that the weights can only converge when the content of layers is identical to the initial inputs (that is, $y(t) = y(0)$ and $x(t) = x(0)$). The function therefore correlates directly with network outputs, instead of activations. As a result, the learning rule is dynamically linked to the network’s output (unlike most BAMs).

4. Simulations: General learning behavior

This first empirical section is concerned with the more “cognitive” simulations, that is, those related to the model’s learning and categorization behavior. We will be interested in determining if we can successfully implement the kWTA and topological properties.

4.1. Simulation: Prototype-based categorization

Categorization, as already argued, is a crucial cognitive task. Adding competitive and topological properties to the original BHM will enable it to emulate this process, by creating a categorical landscape based on a topological 2D map. A first simulation was conducted in order to show the network’s ability to cluster exemplars into categories. During recall, whenever the network associated two or more inputs with the exact same $x$ layer reconstruction, these inputs were considered as being categorized together.
4.1.1. Methodology

For this simulation, artificial pixel-based stimuli (Fig. 3(a)) were created. Four category prototypes were produced by generating bipolar-valued, 36-position vectors, for which each value (or “feature”) followed a random discrete uniform distribution. The presence of a feature (black pixel) was represented by a value of +1, and absence (white pixel) by a value of −1. Five exemplars were generated using each prototype, by randomly “flipping” the value of one to four features (Fig. 3(a)). The average intra-category correlation was $r = 0.72$ (average inter-category correlation: $r = 0.08$).

Learning was conducted according to the following procedure:

1. Random weight initializations;
2. Random selection of a given exemplar ($x(0)$);
3. Computation of the activation according to Eq. (11);
4. Computation of the initial compression ($y(0)$) according to Eq. (9);
5. Computation of the reconstructed input ($\hat{x}(1)$) according to Eq. (12);
6. Computation of the compressed output ($\hat{y}(1)$) according to Eq. (11);
7. Weight updates according to Eqs. (13) and (14);
8. Repetition of 1 to 7 for all exemplars;
9. Repetition of 1 to 8 until the mean error ($\|y(0) - \hat{y}(1)\|/g$), where $g$ is the number of exemplars in the learning set, is sufficiently low ($<0.1$).

Since the BHM + SOFM is a recurrent network, the following iterative recall process, using the converged weight matrices, was used:

1. Selection of a given exemplar;
2. Iteration through the network (with selection of $k$-winners during each iteration) until both the contents from each layer have converged, that is: $x(t) = x(t+1)$, and $y(t) = y(t+1)$;
3. Repetition of 1 and 2 for all exemplars;
4. Determination of category landscape: if the final compression $y$ for two or more exemplars are identical, then these exemplars are members of the same category.

For this simulation, parameters were set to the following values: $\eta = 0.01$; $\delta = 0.1$; $\sigma_0 = 10$; and $\tau = 100$. The number of $y$ units was set to 100 ($10 \times 10$ grid) and the number of winning units was set to 15.

4.1.2. Results

After 236 learning blocks, the error ($\|y(0) - \hat{y}(1)\|/g$) value was lower than the criterion, and therefore learning stopped. Fig. 4 shows the associations between category membership and $y$ layer units. Each category is represented by a different color. In order to determine category association, each exemplar was passed through the transmission function (Eq. (11)) once, using the $W$ matrix in its current state, at different stages of learning (i.e., after $p$ blocks). For each exemplar, the activation for each unit of layer $y$ was computed. The category containing the exemplar with the highest level of activation was selected for each unit. Each panel represents the “categorical landscape” after $p$ learning blocks. It can be seen that as the number of learning blocks increases, category sensitivity is adequately represented topologically in the matrix. The network creates 4 well-separated categories. Moreover, each category occupies approximately 25% of the total space. Therefore, the model seemingly acts in a way similar to Kohonen’s SOFM model (see Chartier et al. (in press)).

The BHM + SOFM is a recurrent model, and should therefore preserve autoassociative memory properties (Hassoun, 1989). During the recall phase, when the initial inputs (Fig. 3(a); $x(0)$) are iterated until convergence ($\hat{x}(c)$), the network is lead to an invariant state (or attractor). Fig. 3(b) shows the converged recall output for the $x$ layer ($\hat{x}(c)$). The network develops four attractors, one for each category. Each separate exemplar from a category reaches a common attractor, and groupings respect predetermined category membership. The network has extracted prototypes from their related exemplars. In this case, because the within-category correlations are quite high and the number of output units (100) is low, the task can be achieved without supervision. As shown in the next simulation, if one is to store specific exemplars using unsupervised learning, then the number of output units must be increased.5

4.2. Simulation: Learning pixel-matrix letters

4.2.1. Methodology

The previous simulation was replicated using stimuli with no predetermined categorical (or cluster) membership, thus making it an exemplar learning (or identification) task. Here, input patterns must not be clustered (such as in the categorization task), but rather be differentiated; each stimulus must “form its own cluster”. We were interested once again in studying the topological properties, but we would also like to explore the simultaneous reconstruction capacities of the network, under normal and noisy

5 Of course, there are other ways to encode the desired number of exemplars in a more efficient way. For instance, a vigilance parameter could be used (Grossberg, 1988).
conditions (recurrent network properties). The patterns used for the simulation were 7 × 7 pixel matrices representing a letter of the alphabet (Fig. 5). White and black pixels were respectively assigned corresponding values of −1 and +1. Correlations (in absolute values) between the patterns varied from 0.02 to 0.84. Because the network had to discriminate between a larger number of “classes” than in the previous simulation (10 stimuli vs. 4 categories), the size of the output grid was increased to 400 units (20 × 20). The learning parameter η was set to 0.001 and the number of winning units was increased to 20 (5% of the output units’ resources). Learning was conducted using the procedure described in the previous section. However, in this case, the learning error is now based on correct reconstruction of the x layer, and therefore was defined as (∥x(0) − x(1)∥/g). The stopping criterion was changed to 0.001.

In addition to regular recall, an additional recall phase was conducted. The same recall procedure was used. However, the original exemplars were initially contaminated with various types of noise (random pixels flipped, addition of normally-distributed random noise, specific portion of a given pattern removed).

Fig. 4. 2D topology in function of p learning blocks.

Fig. 5. Set of patterns used for training.

Fig. 6. 2D topology for weight connections. Each letter represents the output’s stable solution. Blank boxes indicate unused weight connections.

Fig. 7. Noisy recall of different patterns after 20 cycles through the network. The first pattern is the letter “A” for which 6 pixels have been randomly flipped. For the second pattern, a random noise vector N ∼ (0, 0.5) has been added to the letter “F”. The third pattern is the letter “B”, for which the upper portion pixels have been set to zero. The fourth pattern is the letter “J” for which the right portion pixels have been set to zero.

4.2.2. Results

4.2.2.1. SOFM properties. The weights have converged after approximately 4500 learning blocks. Fig. 6 shows the best matched pattern for each unit of the y layer, once all 20 winning output units have stabilized. Once again, for each unit, the most activated pattern was selected. Each pattern is well-separated into a closed, well-delimited region, and we can see through the topological representation how common features lead the network to represent some letters closer in space. For example, letters E and F, which share a left vertical bar, as well as two horizontal bars, are represented as closer to each other than to letter I, which does not share any of these features. Letters C and G, who are highly similar, are also closer in 2D space. It is clear that only a small portion of the 2D map is used at one time. It is thus apparent that the BHM can encompass topological and k-WTA properties successfully, in a more general learning situation.

4.2.2.2. Autoassociative properties. The model was able to perfectly reconstruct the original patterns from Fig. 5. Because the architecture is bidirectional, these reconstructed vectors also are invariant states in the network (Hassoun, 1989). This allows the network to filter noise and complete patterns just like any recurrent associative memory (e.g., Hopfield (1982)). Fig. 7 illustrates such properties through a noisy recall task, where different examples of input patterns are contaminated with various types of noise (random pixels flipped, addition of normally-distributed random noise, specific portion of a given pattern removed). As shown, the contamination does not prevent the network from recalling one of the original patterns.

We compared the performance of the new model, BHM + SOFM, with the original BHM in a more systematical fashion; random pixels were flipped during recall. Fig. 8 shows the recall performance of both networks when the value of 1 to 15 pixels was changed (from 1 to −1 or vice versa). The performance level of the new model is generally slightly lower than that of the original BHM when pixels are flipped. However, the original BHM is not an unsupervised model per se, and does not display any SOFM properties. Also, the new model’s performance is still higher than that of the original BAM (Kosko, 1988).
4.3. Discussion

In this section, we have shown that it can be advantageous to reunite principles stemming from competitive and topological networks with bidirectional Hebbian (correlational) properties. The resulting model can achieve topological categorization and identification, while preserving autoassociative memory behaviors, such as attractor development and resistance to noise.

While the model performs well, it still retains one of the major downsides from the original SOFM. In order for this model to learn adequately, the size of the topological neighborhood and the value of the learning parameter are required to decrease with time (Eqs. (6) and (7)). This leads to stability in the model, but prevents the model from learning novel stimuli without having to start over the learning process, using new weights and an extended data set.

Although the BHM + SOFM can use a constant learning parameter value (Eqs. (13) and (14)), variance for the Gaussian function must still decrease with time (Eqs. (9) and (10)), making it impossible to learn novel stimuli after a finalized learning phase. Hence, just as the SOFM, the BHM + SOFM model shows stability, but not plasticity (Grossberg, 1988). In a topologically-constrained network using a WTA learning algorithm (such as the SOFM), there are no solutions to achieve simultaneous stability and plasticity. However, if one were to allow the use of several winning units (kWTA situation), then, under some conditions, it is possible to preserve essential topological properties, while achieving novel learning.

The principle is straightforward. Fig. 9(a) illustrates a single y unit using a Mexican-Hat function (generally used in a WTA case for bipolar patterns, as well as in our model so far). In a WTA situation, if unit A wins the competition, then, in function of its topological neighborhood’s size (solid line), connections for another unit C will either be positively reinforced if unit C is included in unit A’s neighborhood (Fig. 9(a)), or negatively reinforced (inhibited) if it is outside of that neighborhood (Fig. 9(b)). If two units, both using Mexican-Hat functions, are allowed to win (kWTA situation), then, in function of the distance between winning units A and B, unit C may be negatively reinforced by both units if it is positioned outside of their neighborhood (Fig. 9(c); dashed line), slightly positively reinforced if it is part of both units’ neighborhoods (Fig. 9(d)), and may even receive more combined positive reinforcement than any of the two winning units if these units’ neighborhoods show a large overlap (Fig. 9(e)).

This last situation illustrates how the combination of excitation and inhibition from many winning units could “bring the units together” to form a single cluster while keeping the topological size constant. Therefore, in a kWTA setting, instead of having the size of topological neighborhood function that decreases over time, the distance between the winners has to decrease over time. This can be accomplished when using a converging learning algorithm that uses time-difference association (e.g., Eqs. (13) and (14)).

To sum up, the size of the topological neighborhood (standard deviation in Eq. (9)) will from now on be independent from the number of learning trials. Eq. (10) thus becomes:

$$h_{j,i(x)} = \frac{1}{2} \left( 3e^{-\frac{d_{i,j}^2}{\sigma^2}} - e^{-\frac{d_{j}^2}{4\sigma^2}} \right)$$

$$d_{j,i(x)} = \| r_j - r_i \|^2$$

(16)

where $\sigma (>0)$ is a general free parameter. The value for $\sigma$ should be selected in function of the number of desired winners and the size of the grid. A higher number of winners, or a larger grid should lead to a higher $\sigma$ value being used.

5. Simulations: General properties

The following subsections will explore some of the more general properties of the model, which now uses a new constant definition of neighborhood size. We will use results from the next set of simulations in order to determine, in a following study, if the change in defining neighborhood size allows for simultaneous stability and plasticity.

5.1. Simulation: Number of winning units

The network replicated the categorization task used in Section 4.1. We studied the effect of varying the number of winning units used in a given simulation (from 1 to 15 units). Two performance indicators were used. We first measured the distance between the actual and optimal (uniform) weight distributions. Given that four categories were used, the optimal proportion of weights related to each category should be equal to 25% (or 0.25). In addition, a measure of dispersion index was used. Here, for each category, the distance between each unit in the grid and all the other units that belong to the same category are computed, averaged and standardized. The process is repeated for all categories,
Fig. 9. Effect of combining 2 winners using the “Mexican-hat” transmission. In a WTA setting, the reinforcement of unit C will be a function of the size of the topological neighborhood for winning unit A. The reinforcement can be positive (a) or negative (b). In a kWTA setting, reinforcement of unit C will be a function of the distance between winning units A and B. If the distance is too large, unit C will be negatively reinforced (c). If the distance is small, then unit C can be positively reinforced (d) as in a WTA situation (a), or even be more reinforced than the two winning units (e) to form a single cluster.

and then globally averaged. More formally, the distance $d$ for a given category $c$ is given by

$$d_c = \frac{\sum_i \sum_j \sum_k \sum_l |z_{ij} - z_{kl}|}{n_c}$$

(17)

where $z$ represents the position of a particular unit from category $c$ and $n_c$ the total units of category $c$. The global standardized value of the dispersion index is given by

$$\text{dispIndex} = \frac{\bar{d}_c - \text{MinDisp}}{\text{MaxDisp} - \text{MinDisp}}$$

(18)

where, $\bar{d}_c$ represents the average distance for all categories, and $\text{MinDisp}$ and $\text{MaxDisp}$ respectively represent the minimum and maximum theoretical dispersion. Both the minimum and maximum will vary according to the size of the topological grid. The dispersion index measures within-cluster variability, and thus establishes how dense the developed clusters are. Values range from 0 (maximally sparse) to 1 (maximally dense).

5.1.1. Methodology

For each separate simulation, the learning and recall procedures were identical to those of Section 4.1. In order to obtain a robust estimate of the model’s performance, for each different variable value (1 to 15 units), learning and recall were repeated 50 times, each time with a different set of exemplars (stemming from randomly-generated prototypes).

5.1.2. Results

Fig. 10 shows that the dispersion index (and thus within-cluster variability) decreases as the number of winners increases. This
means that cluster density increases as the number of winning units increases from 1 to 11, after which it stabilizes. Fig. 10 also shows that the weight distribution was fairly optimal (low average distance between the observed and theoretical proportion) for all conditions. Therefore, if the number of winning units is too low (10 or less), the network will use about 25% of the available units to encode the patterns (low distance), but these units will be scattered all over the map (as shown by the high dispersion index). With approximately 11 units or more, the network will encode the four categories with the same optimal weight proportion and will create well-defined clusters (as shown by the low dispersion index).

5.2. Simulation: Size of the topological grid

The previous task was replicated using a different variable, namely the size of the 2D topological grid (from $4 \times 4$ (16 units) to $8 \times 8$ (64 units)). Following results from the previous simulation, in order to obtained well-distributed weights and dense clusters, the number of winning units was kept constant at 15. For this simulation, performance was compared to that of the original SOFM (Haykin, 1999; Kohonen, 1982).

5.2.1. Methodology

For each separate simulation, the learning and recall procedures were identical to those of Section 4.1, except of course for the size of the topological function $\sigma$, which remained constant throughout ($\sigma = 4$). For each different variable value, learning and test were repeated 50 times, each time using a different set of patterns. The performance indicators from the preceding simulation were used.

5.2.2. Results

Fig. 11 illustrates the performance for the BHM + SOFM and the SOFM as it relates to distance from the optimal weight proportions (Fig. 11(a)) and the dispersion index (Fig. 11(b)). In general, the value of both performance indicators for the SOFM remained relatively unchanged when varying grid size. In contrast, because the BHM + SOFM model uses several winning units, its performance is a function of the grid’s dimensionality. Using a grid size of at least $8 \times 8$ with 15 winning units enables the BHM + SOFM to show better performance than the original SOFM for both measures.

6. Simulations: Simultaneous stability and plasticity

Now that certain constants have been established, we can determine the usefulness of the constant neighborhood size parameter, by testing if the network can learn novel patterns without losing memory for a previously topologically-encoded pattern set.

6.1. Simulation: Learning pixel-matrix letters

This simulation aims at determining if the model can increase its memory for novel patterns. Two learning phases are defined. During the first phase, the network must learn four letter patterns (Fig. 5(a); letters A to D). Once learning is complete, the network enters a second learning phase, where the number of patterns to learn is increased from four to six (the four original patterns plus two new ones; Fig. 5(a); letters A to F).

6.1.1. Methodology

Patterns used in both phases are identical to those illustrated in Fig. 5(a). For each phase, the learning and recall procedures were identical to those described in Section 4.1, using the error measure defined as $\left(\frac{\|x(0) - x(1)\|}{g}\right)$, and a stopping criterion set to 0.001. During the first phase, pixel versions of the uppercase letters A, B, C and D were presented to the network. During the second phase, the letters E and F were added to the original pattern set. Because a total of only 6 patterns will be learned, a $15 \times 15$ topological grid was used. The number of winning units was set to 20; the size of the topological function was set to $\sigma = 4$ and remained constant throughout learning. Learning was achieved in a blocked fashion.

6.1.2. Results

Fig. 12 shows the error in function of the number of learning blocks or epochs. Following completion of the first learning phase, the error is minimal, as expected. When the second phase begins, the error value increases suddenly, but remains much lower than the error measured on the first block of the initial learning phase. The sudden increase is clearly attributable to the new patterns
being learned; if the initial patterns had been forgotten, the error value would have been similar to that measured at the beginning of the initial learning phase. This result thus suggests that the network has preserved its memory for the original patterns, even though new patterns are inserted in the learning set.

Another way to explore this behavior is by examining the topological grid at certain stages of learning. Fig. 13(a) shows the $15 \times 15$ topological grids for the first four patterns (letters A to D) after only one block of training (Phase 1). It can be seen that no cluster has been formed in stimulus space. The 20 winning units are scattered all over the bidimensional map. After completing the first learning phase, clusters have been formed, and the network has correctly learned the initial pattern set. During the subsequent phase, two additional patterns are added to the learning set. Fig. 13(b) shows the grids after one block of learning in the second phase. It can be seen that while the new patterns have not been learned yet, adding new patterns to the learning set has absolutely no effect on the grids for the initial four patterns. Once the second learning phase is complete, all six patterns have been learned correctly (Fig. 13(c)). The output now forms six well-defined clusters. These results therefore show that the network is able to increase its memory for patterns without losing what it had already stored. This behavior cannot be achieved by a strict SOFM.

### 7. General discussion

In this paper, we have proposed a new model encompassing correlational, competitive and topological properties. The new model, BHM + SOFM, primarily stems from Chartier and Boukadoum's (2006a) BHM. The first modification to the original BHM concerns the architecture. While the BHM uses two inputs (supervised model), the BHM + SOFM only needs one (unsupervised model). Also, a kWTA setting, a topological grid, and a Mexican-Hat function were added to the original model.

A first set of simulations showed that uniting different classes of network properties allows the model to categorize and learn patterns in a topological fashion, while retaining the BHM’s basic properties, such as attractor development and noise tolerance during recall. The recurrent architecture allows for the simultaneous development of prototypes in one layer, and of clusters in the other layer. All the tasks are achieved using a single architecture, as well as the same learning and transmission rules.

A slight modification to the model allows it to show both stability and plasticity, properties that have not before been shown simultaneously in neural networks based on topological architectures. This is a very interesting breakthrough, since as humans, we do not forget as soon as we learn something new. However, the variance of Gaussian is fixed, making convergence to a proper clustered solution a function of the number of winning units and the size of grid. As more and more units are allowed, the network will produce representations that are less sparse. A solution would be to increase the value for parameter $\sigma$. However, if this value is too high, the network will not be able to differentiate between categories. In comparison, if $\sigma$ is small, then it is hypothesized that the network will regroup the outputs on more than one cluster, but less than the number of winning units. Therefore, further studies should evaluate the possibility of optimizing the value of $\sigma$ in function of the size of the grid and the number of winning units. The type and number of clusters could then be evaluated, in order to provide an estimation of grouping (shared clusters) between exemplars and categories. In addition, although the type of stability/plasticity shown by the model is the identical to the one achieved by RAM/BAMs, its flexibility could be increased if a novelty detector (vigilance) was implemented (Grossberg, 1988).

The new set of properties presented in this paper adds to the many other BHM framework properties, such as bidirectional association of gray-level patterns, aperiodic recall, many-to-one association, multi-step pattern recognition, perceptual feature extraction, input compression and signal separation (Chartier & Boukadoum, 2006a, 2006b; Chartier & Giguère, 2008; Chartier et al., 2007, 2008; Giguère et al., 2007a, 2007b). In short, the BHM framework can replicate various behaviors shown by recurrent associative memories, principal component analysis and self-organizing maps while keeping the same learning and output functions, as well as its general bidirectional heteroassociative architecture. This result constitutes a step toward unifying various classes within a general architecture.

Finally, while it was shown that the addition of competitive and topological properties was beneficial to the network, some of the desirable properties of the model may stem from the
simple modification made to the architecture. Work is already
under way to explore the basic features of a fully distributed
version of the network, which then becomes totally equivalent
to a dynamic heteroassociative memory (Hassoun, 1989). Further
studies will also be focused on the extent to which this framework
can reproduce other kinds of human learning behaviors.

References


Perceptual learning: The psychology of learning and motivation: Vol. 36. Learning
as extraction of low-dimensional representations (pp. 353–380). San Diego, CA:
Academic Press.

for binary and grey-level patterns. IEEE Transactions on Neural Networks, 17,
385–396.

for multi-step pattern recognition and one-to-many association. IEEE Transactions
on Neural Networks, 17, 59–68.

Chartier, S., & Giguère, G. (2008). Autonomous perceptual feature extraction in a
topology-constrained architecture. In B. C. Love, K. McRae, & V. M. Sloutsky
(Eds.), Proceedings of the 30th Annual Conference of the Cognitive Science Society

Chartier, S., Giguère, G., Langlois, D., & Sioufi, R. Bidirectional associative memories,
self-organizing maps and k-winners-take-all: Uniting feature extraction and
topological principles. In Proceedings of the 2009 International Joint Conference
on Neural Networks (in press).

International Joint Conference on Neural Networks (pp. 1679–1687).


memory for learning bipolar and nonbipolar correlated patterns. IEEE Transactions
on Neural Networks, 16, 1393–1400.

on Neural Computation (NC’2000) (pp. 55–61).


Gibson, J. J., & Gibson, E. (1955). Perceptual learning: Differentiation or enrichment?
Psychological Review, 62, 32–41.

Giguère, G., Chartier, S., Proulx, R., & Lina, J.M. (2007a). Category development and
reorganization using a bidirectional associative memory-inspired architecture.
In R. L. Lewis, T. A. Poll, & J.E. Laird (Eds.), Proceedings of the 8th International
Conference on Cognitive Modeling, (pp. 97-102). Ann Arbor, MI: University of
Michigan.

features using a BAM-inspired architecture. In D. S. McNamara, & J. G. Trafton
(Eds.), Proceedings of the 29th Annual Conference of the Cognitive Science Society

585–612.

W. Proctor (Eds.), Experimental psychology: Vol. 4. Comprehensive handbook of
psychology (pp. 591–621). New York: Wiley.


Press.


Networks, 2, 275–287.


associative memory. Neural Networks, 21, 1041–1044.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective
computational abilities. Proceedings of the National Academy of Sciences of the
USA, 9, 2554–2558.


systems for item memory and category knowledge. Science, 262, 1747–1749.

Biological Cybernetics, 43, 59–69.

500–526.

Man and Cybernetics, 18, 49–60.

Networks, 1, 44–57.

network. In D. Touretsky (Ed.), Advances in neural information processing systems
(pp. 634–642). San Mateo, CA: Morgan Kaufmann.


Mathematical Biology, 15, 267–273.

Journal of Neural Systems, 1, 61–68.

Opinions in Neurobiology, 14, 481–487.


Experimental Psychology, 81, 304–308.


Machine Learning, 3, 9–44.