

Learning and Recalling Arbitrary Lists of Overlapping Exemplars in a Recurrent Artificial Neural Network

Damien Rolon-Mérette (drolo083@uottawa.ca)

School of Psychology, 136 Jean-Jacques Lussier, Vanier Hall
Ottawa, ON, K1N 6N5, CAN

Thaddé Rolon-Mérette (trololo068@uottawa.ca)

School of Psychology, 136 Jean-Jacques Lussier, Vanier Hall
Ottawa, ON, K1N 6N5, CAN

Sylvain Chartier (sylvain.chartier@uottawa.ca)

School of Psychology, 136 Jean-Jacques Lussier, Vanier Hall
Ottawa, ON, K1N 6N5, CAN

Abstract

The mechanisms behind the ability to retrieve all exemplars in a class when presented a specific contextual cue, have puzzled the world of cognition. Various approaches have been used to better understand this concept, especially in the field of artificial neural networks. That being said, very few models can enumerate all exemplars associated to multiple lists in a cognitively plausible way. This is mostly due to the problem of multiple One-to-Many Associations (OMAs) where various exemplars can belong to different lists. To resolve this issue, different approaches have been used; from deep learning and natural language processing to time delayed contextual units. However, none of them is satisfying for a biologically based computational model of cognition. A promising solution is using the class label as context and associates it with each exemplar of the corresponding class. This allows each input to be unique and the problem becomes a standard association one. This strategy has been implemented within the neurodynamic perspective using a bidirectional associative memory. The simulations consisted of learning three arbitrary sequences of various lengths containing multiple intertwined exemplars. Results showed that it was possible to enumerate all associated exemplars from a class simply by presenting the corresponding contextual label. These findings are an important step towards developing cognitively plausible neural implementation of multi-step patterns as well as semantic networks in order to develop generalized artificial intelligence.

Keywords: Cognition; Class enumeration; One-to-many associations; Learning; Memory; Bidirectional associative memory; Feature extraction bidirectional associative memory

Introduction

When presented with a class label, the brain has no difficulty in enumerating all associated exemplar. This cognitive ability is remarkable since learned exemplars are rarely exclusive to a single class; they can have multiple associations (also referred as One-to-Many Associations; OMAs). A simple example to illustrate this would be to enumerate all actions (exemplars) needed to score in a specific sport. In soccer a sequence may resemble, kick off, pass, dribble, run, and kick; while in hockey: face off, pass, dribble, skate and slap; and finally, in basketball: jump ball, pass, dribble, run and throw. Depending on the class of the sport (soccer, hockey or

basketball), different exemplars (kick, slap and throw) and identical ones can be found (pass, dribble). Thus, in such a case, when enumerating exemplars from a class, it is easily seen that these exemplars can be associated to a single or multiple classes (OMA). Furthermore, when enumerating a list, it may always follow a specific sequence (ex. opening a door) or, in the case of semantic memory, may not (ex: free association task; Nelson, McEvoy & Schreiber, 2004).

Many formal models in cognitive sciences have been proposed over the years that can accomplish this listing task. Models, such as the Semantic memory models, are known to predict human performances accurately (Jones, Willits, Dennis, & Jones, 2015) but remain limited for neural implementation.

Artificial neural networks have also been used to perform this listing task with most using the multi-layer Perceptron approach (Collobert et al., 2011; Elman, 1990; Jordan, 1997; Neville, 2008). Although these are all interesting models, they are limited in meeting the requirements to be considered biologically based computational models of cognition (O'Reilly, 1998). One such class of models that fills these requirements are the Recurrent Associative Memory (RAMs) that belong to the neurodynamic approach (Haykin, 2009). Associative memory consists of learning and storing pairs of identical (auto-association) or different (hetero-association) exemplars. This has been popularized by Hopfield (1982) for auto-association and generalized to Bidirectional Associative Memory (BAM) by Kosko (1988) for hetero-association. Since then, BAMs have seen many modifications allowing them to perform various tasks with better performances; see Acevedo-Mosqueda, Yanez-Marquez & Acevedo-Mosqueda (2013) for a review. Previous studies have shown that RAMs are able to enumerate simple independent lists of exemplars (Chartier & Boukadoum, 2006). However, in the presence of overlapping list of exemplars, like the initial example, they are not able to accomplish the task. This is due to the fact that they must deal with several OMAs. In other words, there are dealing with a relationship instead of a function.

An early solution to solve OMA following findings in cognition (Clarke, 2017; Spillers & Unsworth, 2011; Stoet & Snyder, 2007) was the use of time delayed (context) units

(Elman, 1990). This method integrated previous output(s) with the current input in order to accurately predict the next exemplar in the list (ex. Collobert et al. 2011). Therefore, the one-to-many association was transformed into a one-to-one association. Unfortunately, this solution of delay units (or surrounding context) requires the global knowledge of the number of contextual units prior to learning. Furthermore, it does not really help towards the original task itself; time delayed units (context) are not representative of the class but only of the previous exemplars.

A more interesting solution in machine learning was introduced by Jordan (1997) which used context as a label to modify each exemplar for the enumeration of a given class. Therefore, this contextual label also modifies each exemplar to make them unique without any prior global knowledge.

A second problem may also arise in RAMs if the OMAs are overlapping. In this case there is the possibility that the task becomes a non-linearly separable one. Unfortunately, standard BAMs are not able to solve this unless the model is complexified with a wide range of arbitrary parameters, thus losing its simplistic nature. However, recent studies have shown that an unsupervised version of the BAM can be used to increase the dimensionality of the inputs and therefore, a linear solution can be found when combined with the BAM (ex. Tremblay, Myers-Stewart, Morissette & Chartier, 2013).

Following recent progress in using contextual labels (Rolon-Mérette, Rolon-Mérette & Chartier, 2018a) it is thus proposed to use the class label to make each exemplar unique using a combination of supervised and unsupervised BAMs. This will increase the BAM's versatility and help in learning any number of overlapping OMAs of any length, where exemplars can have any level of correlation and where a non-linear solution is required.

The remainder of the paper is divided as follows: the next section gives brief background of the BAM used in the study; This is then followed by Simulation I, where context is used to show the feasibility of enumerating exemplars from a class and the limits when facing with overlapping OMAs; Simulation II is then presented with a brief background of the unsupervised BAM and how its interaction with the BAM allows the network to perform the desired task; Finally, a short discussion ends this paper.

Bidirectional associative memory

Model description

The model is a modified version of the BAM. Like any neural network, it is defined by an architecture, transmission and learning functions.

Architecture

The BAM's architecture is illustrated in Figure 1. The supervised model has two layers of interconnected units in a bidirectional fashion, where the \mathbf{W} and \mathbf{V} layers return information to each other (both acting as a teacher to one another); where M and N represents the number of units in each layer. The initial patterns are represented by $\mathbf{x}(0)$ and

$\mathbf{y}(0)$ while the outputs of the network are $\mathbf{x}(t)$ and $\mathbf{y}(t)$ after t cycles.

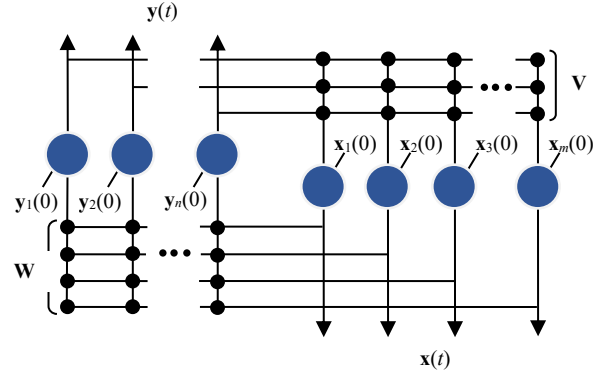


Figure 1: Architecture of the BAM

Output function

The transmission function is defined by equation 1a and 1b:

$$(1a) \forall i, \dots, N, y_{i(t+1)} = f(a_{i(t)}) = \begin{cases} 1, & \text{if } a_{i(t)} > 1 \\ -1, & \text{if } a_{i(t)} < -1 \\ (\delta + 1)a_{i(t)} - \delta a_{i(t)}^3, & \text{Else} \end{cases}$$

$$(1b) \forall i, \dots, M, x_{i(t+1)} = f(b_{i(t)}) = \begin{cases} 1, & \text{if } b_{i(t)} > 1 \\ -1, & \text{if } b_{i(t)} < -1 \\ (\delta + 1)b_{i(t)} - \delta b_{i(t)}^3, & \text{Else} \end{cases}$$

Where i is the index unit, δ the general transmission parameter and a and b the activations. These activations are obtained the usual way: $\mathbf{a}(t) = \mathbf{W}\mathbf{x}(t)$ and $\mathbf{b}(t) = \mathbf{V}\mathbf{y}(t)$.

Learning rule

The connection weights for the model are modified following a hebbian/anti-hebbian rule (Chartier & Boukadoum, 2006).

$$(2a) \mathbf{W}(k+1) = \mathbf{W}(k) + \eta(\mathbf{y}(0) - \mathbf{y}(t))(\mathbf{x}(0) + \mathbf{x}(t))^T$$

$$(2b) \mathbf{V}(k+1) = \mathbf{V}(k) + \eta(\mathbf{x}(0) - \mathbf{x}(t))(\mathbf{y}(0) + \mathbf{y}(t))^T$$

Where $\mathbf{x}(0)$ and $\mathbf{y}(0)$ are the initial inputs, η is the learning parameter and k is a given learning trial. Equation 2a and 2b shows that the matrix weights will converge when $\mathbf{x}(0) = \mathbf{x}(t)$ or $\mathbf{y}(0) = \mathbf{y}(t)$. To reduce the simulation time, the number of cycles performed according to equation 1 is usually set to $t = 1$. It is guaranteed that the learning will converge if the learning parameter (η) is smaller than the following value (Chartier & Boukadoum, 2006):

$$(3) \quad \eta < \frac{1}{2(1-2\delta)\text{Max}[M,N]}, \delta \neq \frac{1}{2}$$

Simulation I: BAM

The general task is illustrated in Figure 2. In order to recreate the task of enumerating exemplars from a class by only

presenting its class label, three overlapping list of arbitrary patterns were used. The general goal was to learn all overlapping lists. Two simulation (conditions) were created to better understand the complexity of the task and the feasibility of using a single BAM. The first condition was to establish if labels can be used to solve a simple OMA and enumerate all the exemplars of a class and while the second was to show its limitation with overlapping OMAs. Both conditions are illustrated in Figure 3.

Methodology

Arbitrary alphabetic patterns were used to test the network. Each pattern was a 49-dimensional pixel base pattern where black pixels represent the value of +1 and white pixels -1. Those patterns have the property of showing various levels of correlation (between 0.02 and 0.92). Moreover, they can be naturally partitioned in multiple overlapping classes and are easily recognized by experimenters. Of course, any other arbitrary patterns could have been used without any modification in the results. Two conditions were created.

In condition 1, two sequences of three exemplars (class label followed by two letters) were used to generate an OMA scenario. For both sequences the class label was concatenated to each exemplar of the list allowing exemplar modification and transform the OMA into a one-to-one association (Figure 3a). In condition 2, multiple intertwined lists were used (Figure 3b). The number of lists was set to three for proof of concept while avoiding the simplicity of having a binomial solution. Furthermore, contrary to condition 1, each list was of different lengths and contained multiple OMAs. The first list contained the class label “L” followed by the 26 letters of the alphabet in lowercase. The second sequence was a subset of the first, containing the class label “V” followed by all the vowels in lowercase. Finally, the third sequence was a different subset of the first list containing the class label “C” followed by all the consonants in lowercase. For both conditions, each list was ended by an auto-association on the last exemplar (final attractor).

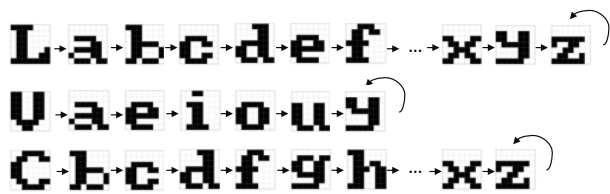


Figure 2: Sequences with class labels (“L”, “V” and “C”) at the beginning of each class for the overall task.

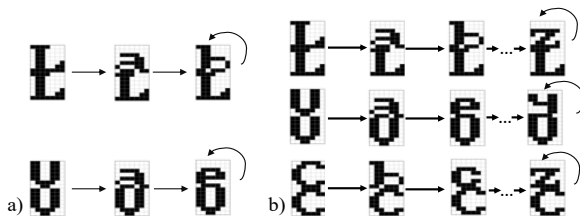


Figure 3: Class labels followed by modified exemplars for condition 1(a) and condition 2 (b).

Procedure

For all simulations, the transmission parameter (δ) was set to 0.2 and the learning parameter (η) respected equation 3. The M and N layers were set to 98 units each, which represents the dimensionality of the combined exemplar (49) and the class label (49). Learning was stopped when the mean squared error (MSE) was lower than 10^{-15} or when 5000 learning trials were reached.

Learning

1. Selection of a list containing both the exemplars and the combined context (Figure 3).
2. Random selection of a pair ($\mathbf{x}(0)$ and $\mathbf{y}(0)$).
3. Computation of $\mathbf{x}(1)$ and $\mathbf{y}(1)$ according to the transmission function (equation 1a and 1b).
4. Computation of the weights according to the learning function (equation 2a and 2b)
5. Repeat step 2) and 4) until all the pairs are selected.
6. Repeat step 2) to 5) until the desired MSE or the maximum learning trial are reached.

Recall

1. Selection of an initial contextual label of a given list, $\mathbf{x}(0)$.
2. Compute $\mathbf{y}(t)$ in accordance to the transmission function (equation 1a and 1b) until convergence; end of the list.
3. Comparison of the outputted exemplars with the correct ones.
4. Repetition of step 1) to 4) for each contextual label (“L”, “V” and “C”).

Results

Figure 4 shows the output for each of the conditions when presented with the class label. In condition 1, the output is a clear representation of the desired solution for both associated outputs. By using this approach, it was possible to solve a simple OMA. In condition 2, results showed that appropriate retrieval was unobtainable in the situation of many OMAs. This is not surprising because in such a scenario the task becomes non-linear as well. Therefore, a single BAM will not be able to perform this task. However, by combining the BAM with its unsupervised version, it is possible to overcome this limit. Therefore, in the next simulation, we show such an implementation while keeping the contextual encoding strategy to allow the network to achieve the desired behaviour shown in Figure 2.

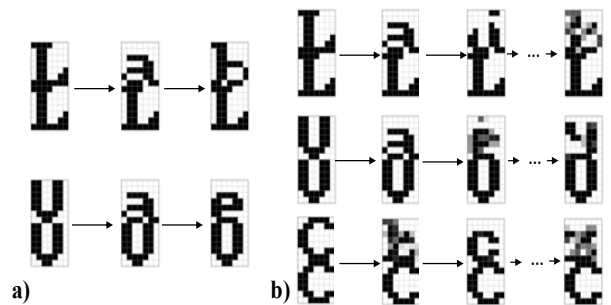


Figure 4: Recall outputs for condition 1(a) and 2 (b)

Simulation II: FEBAM-BAM

In order to use the context to discriminate identical exemplars and to solve non-linear classification, the BAM network is modified to take into account information from its unsupervised version; the Feature Extracting Bidirectional Associative Memory (FEBAM). The FEBAM generates a representation that when combined with the initial input increases the dimensionality and makes the classification problem into a linear one (Tremblay et al., 2013). By still having the same learning and transmission functions and the same general bidirectional architecture, this contributes towards increasing the internal consistency of the overall model.

FEBAM model description

The FEBAM is the unsupervised version of the BAM previously described. The only notable difference between the two is the absence of external ($y(0)$) connections. Consequently, there is no teacher and the model must rely only on one set of inputs, ($x(0)$). Therefore, the goal of this model is to generate the best representation that allows optimal reconstruction of the inputs. It is a process akin to feature extraction (Chartier et al., 2007).

Architecture

The FEBAM's architecture is illustrated in Figure 5. Like the BAM, this model has two layers of interconnected units in a bidirectional fashion, where the W and V layers return information to each other. As mentioned, there is only one explicit set of connections, $x(0)$, used to store information.

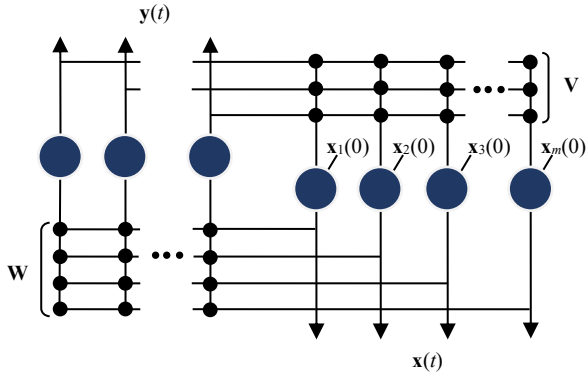


Figure 5: Architecture of the FEBAM

Transmission and learning functions

Both transmission (equation 1a and 1b) and learning functions (equation 2a and 2b) remained the same. However, since $y(0)$ is not explicitly given, the information has to circulate a little longer in the network in order to get all needed inputs.

As shown in Figure 6, $y(0)$ is obtained by iterating $x(0)$ through its corresponding weight connections W using the transmission function. Subsequently, $x(1)$ is obtained from $y(0)$ and finally, $y(1)$ from $x(1)$. Through weight updates, each $x(1)$ and $y(1)$ will converge to a solution that will try to best reconstruct its associated initial pattern $x(0)$ and/or its

representation $y(0)$. Thus, in the case where it is impossible for $x(1)$ to equal $x(0)$ (ex. information compression), weight convergence will only be guaranteed by $y(1)$ and $y(0)$. The number of units in the y -layer determines the dimensionality (level of compression) of the generated representations. The more units there are, the better the reconstruction will be (Giguère et al., 2009).

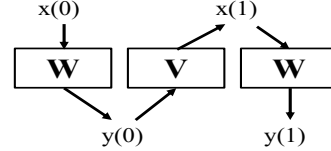


Figure 6: Iterative process used to gather inputs for learning.

FEBAM-BAM Model

Figure 7 illustrates the overall network to accomplish the task where the FEBAM is used to generate features (context).

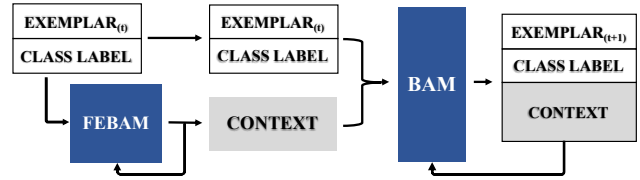


Figure 7: Overall architecture of the FEBAM-BAM

Methodology

The task consisted of learning the same three sequences from simulation I's condition 2 (Figure 3b). This time, exemplars with their class label were fed to the FEBAM first. This allowed the FEBAM to generate features which acted as a unique "signature" for the current exemplar. This representation was then concatenated to the initial input and fed to the BAM for learning. The number of y units in the FEBAM was fixed at a dimension of 98. This was determined in order to increase the probability of success (Rolon-Merette et al., 2018b). The number of y -units can be lower than the number of x -units but for the scope of this study it was not investigated. Finally, in addition to recalling condition 2's lists, a simple noisy (pixel flip) recall task was performed, where the class labels were distorted prior being presented to the FEBAM-BAM. For the noisy recall task, pixel flip ranged from 0 to 50% of the original class label to show the FEBAM-BAM's ability to deal with noise.

FEBAM Learning

The inputs were presented to the FEBAM. To maintain internal consistency, the transmission parameter (δ) and the learning parameter (η) were not change from simulation I. The weights were randomly initialized with values between -0.1 and 0.1 and were updated after one cycle ($t = 1$). Learning stopped when the network achieved a mean squared error (MSE) of less than 10^{-15} or when 5000 learning trials were reached. The learning procedure can be described as follow:

1. Selection of a list containing all three sequences of modified exemplar as seen in Figure 3b).
2. Random selection of a given exemplar from the list to obtain $x(0)$.
3. Iteration through the network (as illustrated in Figure 6) using the output function (equation 1a and 1b) to obtain $y(0)$, $x(1)$ and $y(1)$.
4. Computation of weight updates according to the learning rule (equation 2a and 2b).
5. Repetition of steps 2) to 4) until the minimum mean squared error between $y(0)$ and $y(1)$ or max trials is reached.

Each output was then concatenated to its associated input before being presented to the BAM for learning. The same learning and recall procedure from simulation I was used for the BAM except for the M and N layers, they were increased to 196 units due to the concatenation.

Results

All three sequences (Figure 2) were successfully learned by the combined FEBAM-BAM model (Figure 7). Furthermore, contrary to condition 2 in simulation I, every exemplar for each sequence is retrieved correctly without any distortion. These results are similar to ones obtained in machine learning (Collobert et al., 2011; Jordan, 1997; Neville, 2008).

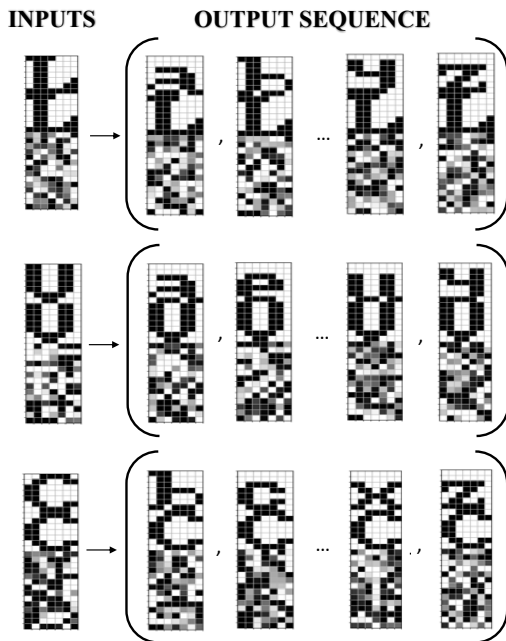


Figure 9: Recall of the learned three sequences.

Likewise, during the pixel flip recall task, correct retrieval was possible for distortion between 0 and 25 %. Figure 10 shows results for a pixel flip of 10% (10 pixels) of the original class label 'L'. This "cleaning" by the FEBAM portion of the FEBAM-BAM allowed to obtain the same retrieval results (Figure 9) while dealing with noisy class labels (inputs).

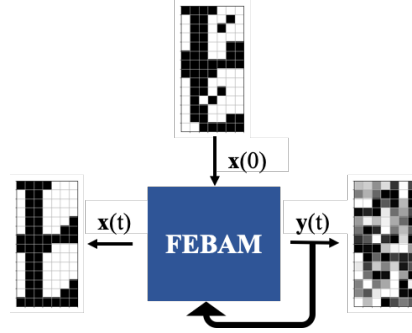


Figure 10: Noisy recall (10 % pixel flip) of class label 'L'

Discussion

In Simulation I, the goal was to learn a simple one-to-many association task (condition 1) by using the context to discriminate the exemplar in a BAM. Results showed that joining the input with fixed contextual information allows the network to solve a simple OMA task using a cognitively plausible neuronal implementation. However, condition 2 showed that this strategy alone is not sufficient if the task contains multiple OMAs. In this last case, a non-linear classification is then required.

To remedy this problem, in simulation II, the BAM was employed in combination with the FEBAM. This addition allowed the network to create its own generated features and when combined with the initial input, allowed to solve non-linear task. The network was able to achieve a perfect learning and recall while maintaining internal consistency; the same transmission, learning functions and the same general bidirectional architecture were used. Furthermore, when faced with distorted exemplars, the model was able to "clean" noisy inputs and reconstruct the appropriate class label while retrieving the associated generated feature. This allowed the FEBAM-BAM to solve the enumeration task despite being presented noisy inputs. This is an important feature towards having a model deal with real world stimuli.

This combination (FEBAM-BAM) is an interesting solution because it avoids the current task specific problem (Marcus, 2018). In approaches where context is given through time delay units (Chartier & Boukadoum, 2006; Collobert et al., 2011; Elman, 1990) the network must know beforehand how many of those units will be necessary for the task, limiting its versatility and plausibility.

That being said, in this model, the proposed mechanism is sequence specific. In other words, although the sequences themselves were arbitrary and could be replaced by any sequences of exemplars, the network outputs will always be in the same order. This is accurate in the case of learning multi-step patterns like motor outputs. However, a future desired property would be the inclusion of more flexibility where the order of outputs is determined from the frequency of occurrence or the success rate of past experience using reinforcement learning. Furthermore, it would be interesting to follow up on the inherent characteristic task (Hattori & Hagiwara, 1998) while using the FEBAM-BAM's ability to

modify the exemplars with pseudo-contextual compartments (Clarke, 2017; Spillers & Unsworth, 2011; Stoet & Snyder, 2007). This would open the door towards a cognitively plausible artificial neural-network capable of combining knowledge acquisition and knowledge transfer, increasing even further the model's versatility. Additionally, it is known that the number of y-units must be greater or equal to the number of unique exemplars for feature extraction in the FEBAM (Tremblay et al., 2013). That being said, it would be advantageous to investigate the probability of success for this multi-OMA task while controlling for the dimensionality of the generated context (FEBAM y-unit). This could determine if the number of exemplars in a list or the number of intertwined exemplars have an impact on the number of y-units needed for the non-linearly separable OMA task. Finally, it would be interesting to account for exemplars in a list representing a single exemplar or a whole category in itself. This would allow the model to perform an important semantic memory task while being a simple neuronal model (free association task; Nelson et al., 2004)

In sum, it was shown that a simple bidirectional recurrent associative memory with a hebbian/anti-hebbian learning algorithm is sufficient to solve a complex task requiring the enumeration of all associated arbitrary exemplars from a class by the sole presentation of a class label. These findings are an important step towards developing a neural implementation of semantic networks in order to shift from narrow intelligence to artificial general intelligence (Bengio et al. 2015; Marcus, 2018).

References

- Bengio, Y., Lee, D. H., Bornschein, J., Mesnard, T., & Lin, Z. (2015). Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*.
- Chartier, S., & Boukadoum, M. (2006). A sequential dynamic heteroassociative memory for multistep pattern recognition and one-to-many association. *IEEE Transactions on Neural Networks*, 17(1), 59-68.
- Chartier, S., Giguère, G., Renaud, P., Lina, J. M., & Proulx, R. (2007, August). FEBAM: A feature-extracting bidirectional associative memory. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on* (pp. 1679-1684). IEEE.
- Clarke, A. (2017). Top-down cognitive influences on object recognition: a commentary on Perry and Lupyan (2016). *Language, Cognition and Neuroscience*, 32(8), 947-949.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(aug), 2493-2537.
- Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179-211.
- Giguère, G., Chartier, S., Proulx, R., & Lina, J. M. (2007, January). Creating perceptual features using a BAM-inspired architecture. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 29, No. 29).
- Hattori, M., & Hagiwara, M. (1998). Multimodule associative memory for many-to-many associations. *Neurocomputing*, 19(1-3), 99-119.
- Haykin, S. (2009). *Neural networks and learning machines*, 3rd ed., Copyright by Pearson Education. Inc., Upper Saddle River, New Jersey, 7458.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 2554-2558.
- Jones, M. N., Willits, J., Dennis, S., & Jones, M. (2015). Models of semantic memory. *Oxford handbook of mathematical and computational psychology*, 232-254.
- Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. In *Advances in psychology* (Vol. 121, pp. 471-495). North-Holland.
- Kosko, B. (1988). Bidirectional associative memories. *IEEE Transactions on Systems, man, and Cybernetics*, 18(1), 49-60.
- Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.
- Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (2004). The University of South Florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3), 402-407.
- Neville, R. (2008). Third-order generalization: A new approach to categorizing higher-order generalization. *Neurocomputing*, 71(7-9), 1477-1499.
- O'Reilly, R. C. (1998). Six principles for biologically based computational models of cortical cognition. *Trends in cognitive sciences*, 2(11), 455-462.
- Rolon-Mérette, D., Rolon-Mérette, T., & Chartier, S. (2018a, July). Distinguishing Highly Correlated Patterns using a Context Based Approach in Bidirectional Associative Memory. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.
- Rolon-Mérette, T., Rolon-Mérette, D., & Chartier, S. (2018b). Generating Cognitive Context with Feature-Extracting Bidirectional Associative Memory. *Procedia Computer Science*, 145, 428-436.
- Spillers, G. J., & Unsworth, N. (2011). Variation in working memory capacity and temporal-contextual retrieval from episodic memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 37(6), 1532.
- Stoet, G., & Snyder, L. H. (2007). Task-switching in human and non-human primates: Understanding rule encoding and control from behavior to single neurons. *The neuroscience of rule-guided behavior*, 227-254.
- Tremblay, C., Myers-Stewart, K., Morissette, L., & Chartier, S. (2013, July). Bidirectional Associative Memory and Learning of Nonlinearly Separable Tasks. In R. West & T. Stewart (Eds.), *Proceedings of the 12th International Conference on Cognitive Modeling*, Ottawa, Canada, pp. 420-425