# A Study of Retrieval Algorithms of Sparse Messages in Networks of Neural Cliques

Ala Aboudib, Vincent Gripon, Xiaoran Jiang
Electronics Department
Télécom Bretagne (Institut Mines-Télécom)
Brest cedex 3, France
e-mail: name.surname@telecom-bretagne.eu

The authors are also with the Laboratory for Science and Technologies of Information, Communication and Knowledge, CNRS Lab-STICC, Brest, France.

***Abstract*** **– Associative memories are data structures addressed using part of the content rather than an index. They offer good fault reliability and biological plausibility. Among different families of associative memories, sparse ones are known to offer the best efficiency (ratio of the amount of bits stored to that of bits used by the network itself). Their retrieval process performance has been shown to benefit from the use of iterations. In this paper, we introduce several rules to enhance the performance of the retrieval process in recently proposed sparse associative memories based on binary neural networks. We show that these rules provide better performance than existing techniques. We also analyze the required number of iterations and derive corresponding curves.**

***Keywords – associative memory; sparse coding; parsimony; iterative retrieval; threshold control***

## I. Introduction

Associative memories are alternatives to classical index-based memories where content is retrieved using a part of it rather than an explicit address. Consider for example accessing a website using a search engine instead of a uniform resource locator (URL). This mechanism is analogous to human memory [*1*] and has inspired many neural-networks-based solutions such as [*2*] [*3*].

A new artificial neural network model was proposed recently by Gripon and Berrou [*4*]. It employs principles from information theory and error correcting codes and aims at explaining the long-term associative memory functionality of the neocortex. This model was proved to outperform the celebrated Hopfield neural network [*3*] in terms of diversity (the number of messages the network can store) and efficiency (the ratio of the amount of useful bits stored to that of bits used to represent the network itself) [*5*]. It was later extended in [*6*] to a sparser version based on the Willshaw-Palm associative memory model [*2*] [*7*].

The key difference between the models proposed in [*6*] and [*2*] is the use of specific structures in the network. This is done by grouping neurons into clusters within which connections are not authorized (multi-partite graph). These clusters are considered analogous to cortical columns [*4*] of mammalian brains within which nodes are likened to micro-columns. This is supported by Mountcastle [*8*] who suggests that a micro-column is the computational building block of the cerebral neo-cortex. In addition, here are some reasons to motivate the use of clusters:

- It is believed that micro-columns in each cortical column react to similar inputs. The concept of clustering is meant to imitate this stimulus-similarity-based grouping. A consequence is the possibility to use this network for retrieving messages from inaccurate observations. This type of retrieval is addressed by Gripon and Jiang in [*9*].
- Clusters allow for simple and natural mapping between non-sparse input messages and sparse patterns representing them in the associative memory. In the case where clusters are all of size 1 each, a model equivalent to the classical Willshaw-Palm networks is obtained, where input messages have to be sparse.
- It was observed that micro-columns usually have many short inhibitory connections with their neighbors [*10*] [*8*], which means that the activation of one micro-column causes all of its near neighbors to be deactivated. This is due to the locally limited energy supply of the brain. This mechanism is represented by the local winner-takes-all rule introduced in [*4*].
- Using clusters allows for introducing guided data recovery in which a prior knowledge of the location of clusters containing the desired data can significantly enhance performance. A detailed study of this type of data retrieval is available in [*6*].

In this paper, we consider the extended version of the model proposed in [*6*]. We introduce several retrieval rules including adaptations of those proposed by Willshaw [*2*], Palm [*11*], Schwenker [*12*] and Gripon and Berrou [*4*]. We also propose new ones and make a comparison of these regarding performance and number of iterations.

The rest of this paper is organized as follows: in Section II, we describe the general architecture of the network model we use. Section III introduces a generic formulation of the retrieval algorithm on such structures. Then, the following five sections develop each step of this algorithm. For each step, different rules are reviewed if available. Some of these rules have been proposed in previous papers, and others we introduce here for the first time. In Section IX, performance comparisons of several combinations of retrieval rules are presented. Section X is a conclusion.

## II. Network Topology and Storing Messages

This section focuses on the neural-network-based auto-associative memory introduced by Gripon and Berrou in [*4*]. It is dedicated to defining this network and describing how it can be extended to store variable-length messages.

### A. *Architecture*

The network can be viewed as a graph consisting of $n$ vertices or units initially not connected (zero adjacency matrix) organized in $\chi$ parts called clusters with each vertex belonging to one and only one cluster. Clusters are not necessarily equal in size but for simplicity, they will be all considered of size $l$ throughout this work. Each cluster is given a unique integer label between 1 and $\chi$, and within each cluster, every vertex is given a unique label between 1 and $l$. Following from this, each vertex in the network can be referred to by a pair $(i, j)$, where $i$ is its cluster label, and $j$ is the vertex label within cluster $i$. For biologically-inspired reasons [*13*] [*14*], and as explained in [*6*], a unit in this model is chosen to represent a cortical micro-column instead of a single biological neuron which is why we shall not use the term "neuron".

At a given moment, a binary state $v_{ij}$ is associated with each unit $(i, j)$ in the network. It is given the value 1 if $(i, j)$ is active and 0 if it is idle. Initially, all units are supposed to be idle. The adjacency matrix (also called the weight matrix $w$) for this graph is a binary symmetric square matrix whose elements take values in $\{0,1\}$. In this representation, a zero means an absence of a connection while a 1 indicates that an undirected (or a symmetric) connection is present. Note that despite the fact that physiological neural networks are known to be asymmetric, we argue that units in the proposed model represent populations of tens of neurons, and therefore can be mutually connected.

Row and column indexes of the weight matrix are $(i, j)$ pairs. So in order to indicated that two units $(i, j)$ and $(k, s)$ are connected, we write $w_{(i,j)(k,s)} = 1$. All connection combinations are allowed except those among units belonging to the same cluster, resulting in a χ-partite undirected graph. When the memory is empty, $w$ is a zero matrix.

## B. *Message Storing Procedure*

We now describe how to store sparse messages using this network. This methodology has been first introduced in [*6*]. Suppose that each message consists of χ submessages or segments. Some of these segments are empty, i.e., they contain no value that need to be stored, while the rest has integer values in $\{1, .., l\}$. For the sake of simplicity, let us consider that all messages contain the same number of submessages $c$. Only those nonempty submessages are to be stored while empty ones are ignored. For example, in a network with $\chi = 6$ and $l = 12$, a message $m = \{,10,7,,12,11\}$ with $c = 4$ has two empty segments (the 1st and the 4th) while the remaining ones have values that need to be stored. In order to store $m$, each nonempty segment position $i$ within this message is interpreted as a cluster label, and the segment value $j$ is interpreted as a unit label within the cluster $i$. Thus, each nonempty segment is associated with a unique unit $(i, j)$. So the message $m$ maps to the 10th unit of the 2nd cluster, the 7th unit of the 3rd cluster, the 12th unit of the 5th cluster and the 11th unit of the 6th (last) cluster. A single message is not allowed to use more than one unit within the same cluster because, by definition, connections are not allowed within a cluster.

Then, given these $c$ elected units in $c$ distinct clusters, the weight matrix of the network is updated according to (1) so that a fully connected subgraph (clique) is formed of these selected units.

$$w_{(i,j)(k,s)} = 1 \quad \text{if} \quad i \neq k \quad \text{and} \quad m_i \,, m_k \neq 0 \tag{1}$$

where $w_{(i,j)(k,s)}$ refers to the undirected connection between $(i, j)$ and $(k, s)$ which are two units associated to message segments $m_i$ and $m_k$, respectively. $i$ and $k$ are cluster indices while $j$ and $s$ are unit indices.

The value of the parameter $c$ can be unified for all stored messages, or it can be variable. A description of how to choose an optimal value of $c$ is provided in [*6*] where $c$ is considered identical for all messages.

It is important to note that if one wishes to store another message $m'$ that overlaps with $m$, i.e., the clique corresponding to $m'$ shares one or more connections with that of $m$, the value of these connections, which is 1, should not be modified. Such a property is called the nondestructivity of the storing process. As a direct consequence, the network's connection map is the union of all cliques corresponding to stored messages.

It is worth noting that when $l = 1$, the structure of this network becomes equivalent to the Willshaw-Palm model.

# III. THE RETRIEVAL PROCESS

The goal of the retrieval process introduced in this paper is to recover an already stored message (by finding its corresponding clique) from an input message that has undergone partial erasure. A message is erased partially by eliminating some of its nonempty segments. For example, if $m = \{,1,8,,10,12\}$ is a stored message, a possible input for the network is $\overline{m} = \{,,,,10,12\}$.

The core of the retrieval process can be viewed as an iterative twofold procedure composed of a dynamic rule and an activation rule. Figure 1 depicts the steps of the retrieval process:

```
Insert an Input Message.
Apply a dynamic rule.
Phase 1:
   Apply an activation rule.
   Apply a dynamic rule.
Phase 2:
  While (stopping criterion is not attained) {
      Apply an activation rule.
      Apply a dynamic rule.
   }.
output ← active units.
```

Figure 1. The generic algorithm for the retrieval process.

Each step of the algorithm of Figure 1 is described in detail in the next sections.

# IV. INPUT MESSAGE INSERTION

An input message should be fed to the network in order to trigger the retrieval process. For example, suppose that we have a stored message $m = \{7,1,5,11,,\}$. Suppose now that we wish to retrieve $m$ from the partially erased input $\overline{m} = \{,,5,11,,\}$. In order to do that, all units corresponding to nonempty segments should be activated. That is, a unit $(i, j)$ associated with the segment of $\overline{m}$ at position $i$ whose value is $j$ is activated by setting $v_{ij} = 1$. So, $\overline{m}$ activates two units: (3,5) and (4,11). Having a number of active units, a dynamic rule should be applied.

# V. DYNAMIC RULES

A dynamic rule is defined as the rule according to which unit scores are calculated. We will denote the score of a unit $(i, j)$ by $\lambda_{ij}$. Calculating units' scores is crucial to deciding which ones are to be activated. A score is a way of estimating the chance that a unit belongs to a bigger clique within the set of active units and thus the chance that it belongs to the message we are trying to recover. In principle, the higher the score the higher this chance is. Two dynamic rules have been already introduced, namely, the Sum-of-Sum [*4*] and the Sum-of-Max [*15*] rules. We also introduce for the first time what we call the Normalization rule.

## A. *The Sum-of-Sum Rule (SoS)*

The Sum-of-Sum is the original rule. It states that the score of a unit $(i, j)$ denoted by $\lambda_{ij}$ is simply the number of active units connected to $(i, j)$ plus a predefined memory effect γ which is only added if $(i, j)$ is active. Scores should be calculated for all of the units in the network. This Sum-of-Sum rule can be formalized by the following equation:

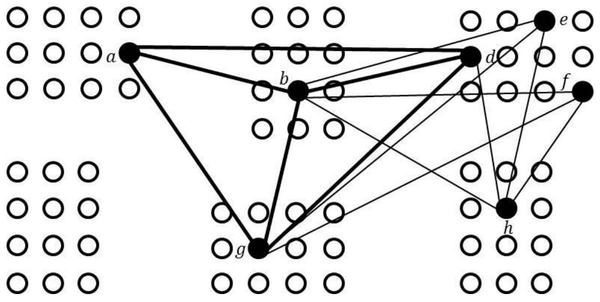


Figure 2. Dynamic rule application phase. Black-filled units are active.

$$\forall i \text{ and } j, 1 \leq i \leq \chi, 1 \leq j \leq l: \quad \lambda_{ij} = \gamma v_{ij} + \sum_{k=1}^{\chi} \sum_{s=1}^{l} w_{(i,j)(k,s)} v_{ks}. \tag{2}$$

Although intuitive, this rule has a major problem which is that in some cases, the scores give a false estimate of the chance that a given unit belongs to a bigger clique within the set of active units. To clarify this point we consider the example of Figure 2 where black circles represent active units at an iteration $t > 1$. The clique we wish to restore is $abdg$ which is maximal in size. Now, we will see what happens when we calculate the scores of units $a$ and $h$ given a memory effect $\gamma = 1$ where $a$ is part of the searched message while $h$ is not. According to the Sum-of-Sum rule, unit $a$ has a score of 4 while unit $h$ has a score of 5. This result indicates that the latter unit is more likely to belong to a bigger clique than the former because it has a higher score. This observation is not true since most of the active units connected to $h$ belong to the same cluster and by conception, a message can only contain at most one unit per cluster. In order to solve this problem, the Sum-of-Max and the Normalization dynamic rules can be applied.

## B. *The Normalization Rule (Norm)*

In the Normalization rule that we introduce here, units' scores are calculated using the following equation:

$$\lambda_{ij} = \gamma v_{ij} + \sum_{k=1}^{\chi} \frac{1}{|v_k|} \sum_{s=1}^{l} w_{(i,j)(k,s)} v_{ks}. \tag{3}$$

where $|v_k|$ is the number of active units in cluster $k$. Equation (3) states that the contribution of a unit $(k,s)$ to the score of another unit connected to it is normalized by the number of active units in cluster $k$. That is, if the cluster $k$ contains $x$ active units, then the contribution of the unit $(k,s)$ becomes $1/x$. So, by applying this rule to the network of Figure 2, unit $h$ gets a score of 3 and unit $a$ gets a score of $3\frac{1}{3}$ which is a result that privileges the activation of the latter unit and thus solves the Sum-of-Sum rule problem.

## C. *The Sum-of-Max Rule (SoM)*

According to the Sum-of-Max rule, the score of a unit $(i,j)$ is the number of clusters in which there is at least one active unit $(k,s)$ connected to $(i,j)$ plus the memory effect $\gamma$ if $(i,j)$ is active:

$$\lambda_{ij} = \gamma v_{ij} + \sum_{k=1}^{\chi} \max_{1 \leq s \leq l} \left( w_{(i,j)(k,s)} v_{ks} \right). \tag{4}$$

So referring back to Figure 2, and according to (4), unit $a$ has a score of 4 whereas unit $h$ has a score of 3. This is a more satisfying result than the one obtained by the Sum-of-Sum rule since it indicates that the latter unit, although connected to more active units, is less likely to belong to a bigger clique within the set of active units than unit $a$.

Moreover, it has been shown in [*15*] that for the particular case, when $c = \chi$, the Sum-of-Max rule guarantees that the retrieved massage is always either correct or ambiguous but not wrong. An ambiguous output message means that in some clusters more than one unit might be activated among which one is the correct unit.

# VI. Activation Rules

The activation rule is applied for electing the units to be activated based on their scores after the application of a dynamic rule. So basically, a unit $(i,j)$ is activated if its score $\lambda_{ij}$ satisfies two conditions:

- $\lambda_{ij}$ is greater or equal than a global threshold that may be chosen differently for each activation rule.
- $\lambda_{ij} \geq \sigma_{ij}$ where $\sigma_{ij}$ is the activation threshold proper to unit $(i,j)$. [*16*]

The difference between the two thresholds defined above is that $\sigma_{ij}$ could be set differently for each unit, so it can be used to control a unit's sensitivity to activation. For a very large value of $\sigma_{ij}$, $(i,j)$ is inhibited. This is helpful for excluding a group of units from the search of a certain message in order to save time. The global threshold has a unique value independent of any individual unit. So it is used to elect units to be activated in a competitive activation process. For example, in a winner-take-all competitive process, this threshold could be dynamically set to the value of the highest score in the network in order to activate only units with the highest score.

The activation rule should be able to find two unknowns: The subset of clusters to which the message we are trying to recover belongs, and the exact units within these clusters representing the submessages. Two activation rules are introduced in this paper: the Global Winners Take All rule (GWsTA) which is a generalization of the Global Winner Take All (GWTA) rule, and an enhanced version of the Global Losers Kicked Out (GLsKO) rule initially presented in [*17*] .

## A. *The GWsTA Rule*

The GWTA rule introduced in [*6*] activates only units with the network-wide maximal score. The problem with this rule is that it supposes that units belonging to the message we are looking for have equal scores. It also supposes that this unified score is maximal which is not necessarily the case. It has been shown in [*6*] that spurious cliques, i.e., cliques that share one or more edges with the clique we are searching, might appear and render the scores of the shared units of the searched clique higher than others'.

For example, in the network of Figure 2, if the searched clique is $abdg$, then $bdh$ is an example of a spurious one. Now, by applying the Sum-of-Max rule on the black units which are supposed to be active, and considering $\gamma = 1$, we get the scores: $\lambda_a = 4$, $\lambda_b = 5$, $\lambda_d = 5$, $\lambda_e = 4$, $\lambda_f = 4$, $\lambda_g = 4$, $\lambda_h = 3$. Thus, according to the GWTA rule, only units $b$ and $d$ will be kept active and the clique $abdg$ is lost. This is caused by the spurious clique $bdh$ which increases the scores of $b$ and $d$.

The generalization of the GWTA rule we propose is meant to account for this problem.

The behavior of the Global Winners Take All rule is the same in both phases of the retrieval process. It elects a subset of units with maximal and near-maximal scores to be activated. In other words, it defines a global threshold $\theta$ at each iteration, and only units that have at least this threshold are activated.

In order to calculate this threshold at a given iteration, we first fix an integer parameter $\alpha$. Then we make a list compromising the $\alpha$ highest scores in the network including scores that appear more than once. For example, if the units scores in a network with a total number of units $n = 10$ are

$\{25, 18, 25, 23, 23, 19, 18, 19, 18, 17\}$ and $\alpha = 7$, then our list becomes $\{25, 25, 23, 23, 19, 19, 18\}$. The minimum score in this list which is 18 becomes the threshold $\theta$. Then we apply the equation:

$$\forall i \text{ and } j, 1 \leq i \leq \chi, 1 \leq j \leq l:$$
$$v_{ij} = \begin{cases} 1, & \lambda_{ij} \geq \theta \text{ and } \theta \geq \sigma_{ij} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

It is worth noting that this activation rule is equivalent to the retrieval rule proposed by Willshaw [*2*] in that units are activated by comparing their scores to a fixed threshold $\theta$.

One problem with this rule is that the choice of an optimal $\alpha$ for a certain message size would not be adapted for other message sizes. This limits the possibility of using this rule for retrieving messages of variable sizes. Moreover, this rule always activates a subset of units with maximal and near-maximal scores. But in some cases, when the number of stored messages reaches a high level, the units with the near-maximal score do not necessarily belong to the searched message.

## B. *The GLsKO Rule*

As we have seen, The GWsTA rule needs a prior knowledge of the value of the message size $c$ in order to retrieve a message. This means that if $c$ is not available at retrieval, the rule may not be able to correctly retrieve information. The Global Losers Kicked Out (GLsKO) rule is designed to address this problem by being independent of any prior information about $c$ which should also enable it to retrieve variable-sized messages more efficiently than the GWsTA rule. In order to achieve this, the GLsKO rule has a behavior in *phase 1* of the retrieval process that differs from that of *phase 2* as follows:

***Phase 1:***
*Apply the GWTA rule.*
***Phase 2:***
*Kick losers out.*

In *phase 1*, the GWTA rule is applied resulting in the activation of a subset of units to which the searched message is guaranteed to belong. After this, the activation thresholds of inactive units are set to infinity because we are no more interested in searching outside the already activated units.

In *phase 2*, the rule changes behavior, so, at each iteration, it makes a list compromising the $\beta$ lowest nonzero scores of the active units only. For example, if the set $\{25, 18, 25, 23, 23, 19, 18, 19, 17, 17\}$ represents the scores of active units in a network with a total number of units $n = 10$ and we fix $\beta = 3$, then the list of lowest scores becomes $\{18,19,18,19,17,17\}$. After that, a threshold $\theta$ equal to the maximum score in the latter list is set, and only units with scores greater than $\theta$ are kept active. This can be described by the following equation:

$$\forall i \text{ and } j, 1 \leq i \leq \chi, 1 \leq j \leq l:$$
$$v_{ij} = \begin{cases} 1, & \lambda_{ij} > \theta \text{ and } \theta \geq \sigma_{ij} \\ 0 \text{ and } \sigma_{ij} \to \infty, & \text{otherwise} \end{cases} \quad (6)$$

The reason why $\sigma_{ij}$ is set to an infinitely large value is that after the first phase of the algorithm, a subset of units is activated. The clique corresponding to the message we are looking for is guaranteed to exist in this subset given that we are dealing with partially erased messages. So, setting $\sigma_{ij}$ in this fashion ensures that units that have failed to be active upon the first phase would be out of the search scope throughout the retrieval process.

We propose to enhance the performance of the GLsKO rule by controlling the number of units $\mu$ to be deactivated. This is only interesting when $\beta = 1$. For example, if we set $\beta = 1$ in the network example of the previous paragraph, we get the following list of scores {17,17}. If $\mu$ is not specified, all losers are deactivated. But by setting $\mu = 1$, only one of these two units is randomly chosen to be deactivated. This may be useful if we wish to exclude losers one at a time and thus reduce incautious quick decisions.

# VII. STOPPING CRITERIA

Since the retrieval process is iterative, a stopping criterion should be used in order to put this process to an end. In parts *A* and *B* of this section we review the two classic criteria that are already in use. In parts *C* and *D*, we propose two new ones that are supposed to enhance performance.

## A. *A Fixed Number of Iterations (Iter)*

A stopping criterion can be defined as a predefined number of iterations of the retrieval process. So dynamic and activation rules are applied iteratively, and when a counter announces that the desired number of iterations is attained, the retrieval process terminates and the activated units are taken for the retrieved message. The problem with this approach is that the stopping criterion which is a simple iteration counter is independent of the nature of retrieved message. That is, the activated units after the last iteration are not guaranteed to form a clique corresponding to an already stored message. This stopping criterion is only interesting with the GWsTA rule.

## B. *The Convergence Criterion (Conv)*

This criterion states that if the set of active units at iteration $t + 1$ is the same as that of iteration $t$, the retrieval process is taken as converged so it terminates and the result is output. The convergence criterion is only compatible with the GWsTA rule. In the case of the GLsKO rule, one or more active units are deactivated in each iteration. So it is not possible to have the same set of active units throughout two subsequent iterations.

## C. *The Equal Scores Criterion (EqSc)*

The idea we propose here is that when all scores of active units are equal, the retrieval process terminates and the result is output.

## D. *The Clique Criterion (Clq)*

This new criterion depends on the relationship between the number of activated units and their scores. If the activate units form a clique the retrieval process terminates. Thus, the retrieved message is more likely to make sense though it is not necessarily the correct result. In order to check if the activated units form a clique, we define the set of active units as $\{a_i \mid i = 1,2, \ldots, |A|\}$, $\lambda(a_i)$ as the score of the active unit $a_i$ and $\rho$ as an integer, then we apply the procedure shown in Figure 3:

$\forall\ 1 \leq i, j \leq |A|$.
If
  $\lambda(a_i) = \lambda(a_j) = \rho$ and $|A| = \rho - (\gamma - 1)$.
Then
  output the result.
terminate the retrieval process.

Figure 3. The Clique stopping criterion (Clq).

To make sense of this stopping criterion, we take an intuitive situation when $\gamma = 1$. In this case, the stopping criterion is that when all active units have an equal score which is equal to the number of these units, a clique is recognized, so the process terminates and the result is output.

It is worth noting that when using the GWsTA rule, it is always preferable to combine any stopping criterion with the Iter criterion so that when any one of them is satisfied the process terminates which prevents infinite looping.

## VIII. RESULTS

We have seen that there are many possible combinations of dynamic, activation rules and stopping criteria in order to construct a retrieval algorithm. In this section we will demonstrate the performance of some of these combinations. All messages used for the following tests are randomly generated from a uniform distribution over all possible message values. Reported retrieval error rates for a given number of stored messages are averaged over 100 trials. However, no significant difference was found between average error rates and error rates resulting from single trials. All the tests were written in C++, compiled with g++ and executed on a Fedora Linux operating system.

### A. *Comparing Dynamic Rules*

Figure 4 shows that both the SoM and the SoS dynamic rules give a similar performance. The Norm rule was found to give the same results also, but it is not shown in the figure for clarity. This is not the case with the original network introduced in [*4*] where the Sum-of-Max rule was proved to give better results [*15*]. This is an interesting phenomenon that is worth studying. It may indicate that the major source of retrieval errors in this sparse version of the network is not related to the different methods of calculating units' scores. This renders the differences in performance due to the use of different dynamic rules insignificant.

### B. *Comparing Retrieval Strategies*

We notice in Figure 5 that the GWsTA ($\alpha = 12$) rule gives a better performance than the GWTA (equivalent to GWsTA with $\alpha = 1$) rule used with the Conv stopping criterion with 30 iterations allowed at most. This is due to the fact that the former rule has a better immunity to the phenomenon of spurious cliques described in Section VI.A. We also notice that the GWsTA ($\alpha = 12$) rule gives even a lower error rate when the memory effect $\gamma$ is set to a large value such as 1000. This is because

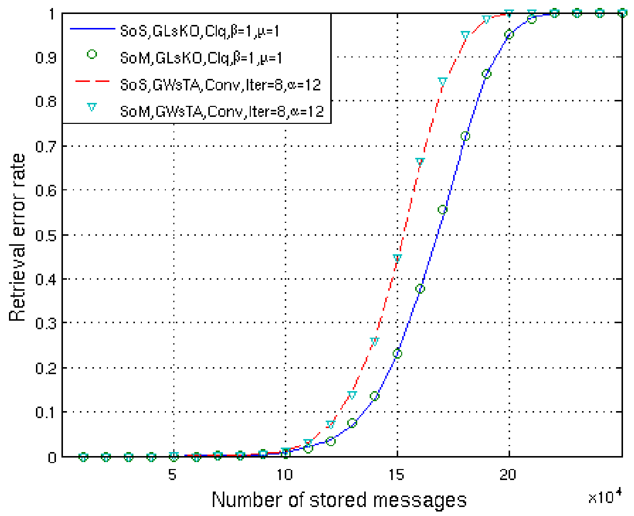


Figure 4. Influence of dynamic rules on retrieval error rates in a network with $\chi = 100, l = 64, c = 12, \gamma = 1, \sigma_{ij} = 0$ initially, with 3 segments of partial erasure in input messages.

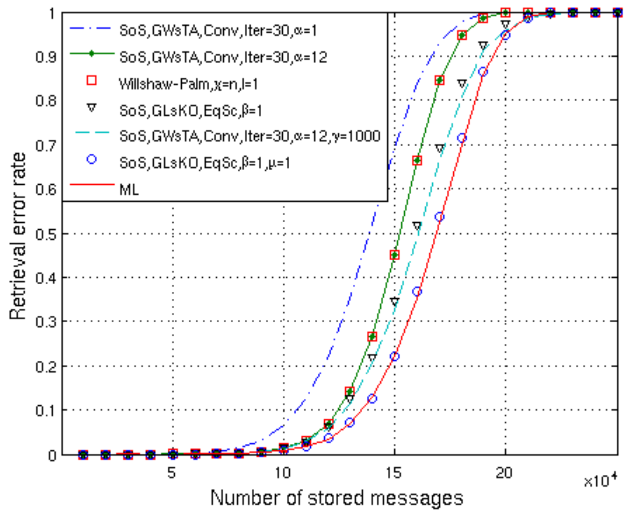


Figure 5. Influence of activation rules on retrieval error rates in a network with $\chi = 100, l = 64, c = 12, \gamma = 1$ if not stated otherwise, $\sigma_{ij} = 0$ initially, with 3 segments of partial erasure in input messages.

setting $\gamma$ to a very large value restrains the search to only a limited region of the network where the target message is thought to exist. This is due to the fact that a large value of $\gamma$ guarantees that active units always get higher scores than other ones. So, in subsequent iterations, the set of active units would most often be the same or a subset of the previous active set. In all cases, the GLsKO ($\alpha = 1$, $\mu = 1$) rule using the EqSc or the Clq (not shown on the figure for clarity) stopping criterion has the lowest error rate which almost achieves the performance of the brute force Maximum Likelihood retrieval algorithm (ML) (which is a simple exhaustive search for a maximum clique) for 3 erased input submessages out 12. This is because the GLsKO rule configured with such parameter values searches for the output in a limited subset of units resulting from *phase 1* and excludes only one unit at a time before testing for the stopping criterion. This is proved by the degraded performance shown in Figure 6 of this same rule but without specifying a value of $\mu$ which results in the exclusion of more than one unit at a time rendering the retrieval process less prudent and more susceptible to bad exclusions.

We also notice that when a Willshaw-Palm network with $n = 6400$ units is used with the GWsTA ($\alpha = 12, \gamma = 1$) rule, the same performance as in a clustered network is obtained.

### C. *The Number of Iterations*

Figure 6 shows that the average number of iterations required to retrieve a message is relatively constant for all rules up to 140000 messages learnt. Beyond this, the number of iterations required for the GLsKO and the GWsTA rules with $\gamma = 1$ begins to increase rapidly. It is worth emphasizing that the maximum number of iterations we allowed for the GWsTA rule is 30 so the constant level reached by the curve representing this rule with $\gamma = 1$ in Figure 6 is just a result of that constraint. However, the number of iterations for the GWsTA rule with $\gamma = 1000$ increases only slightly approaching an average of 3.3 up to 250000 messages stored.

The reason for this explosion of the number of iterations in the case of the GLsKO rule is that the number of units activated after the first phase increases with the number of stored messages. So more iterations would then be needed in order to exclude losers and thus shrink the set of active units. In the case of the GWsTA rule with $\gamma = 1$, all units in the network are concerned with the search for a message in each iteration. So when the number of stored messages increases, the connection density in the network gets higher and it then would be more likely that new winners appear in each iteration violating the Conv criterion.

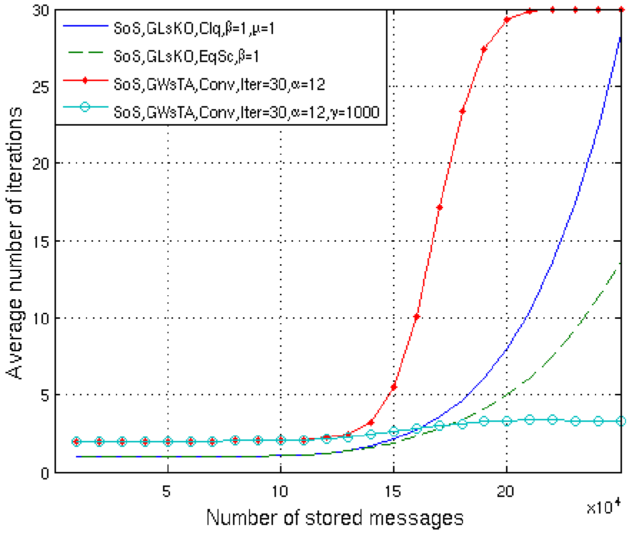


Figure 6. Average number of iterations for different scenarios in a network with $\chi = 100, l = 64, c = 12, \gamma = 1$ if not stated wise, $\sigma_{ij} = 0$ initially, with 3 segments of partial erasure in input messages.

Setting γ to 1000 limits the possibility of the apparition of new winners in each iteration and consequently decreases the number of iterations needed before satisfying the Conv criterion.

## IX. Conclusion and Future Work

In this paper, we analyzed the performance of retrieval algorithms on extensions of a recently proposed sparse associative memory model. We demonstrated and compared the performance of these algorithms when using partially erased messages as inputs. We also provided comparisons between the model proposed in this paper and the Willshaw-Palm model where we proved that the clustering constraint applied to the model we use which decreases the number of available connections does not necessarily affect performance when comparing with the Willshaw-Palm model.

We found that our modified version of the GLsKO activation rule combined with the equal scores or the clique stopping criteria gives the best results in terms of retrieval error rate but with a rapidly increasing number of iterations. Actually, the second phase of the GLsKO rule along with the clique criterion can be viewed as an operation equivalent to searching the maximum clique among active units. This is a famous NP-complete problem. However, many suboptimal solutions were suggested for this problem (or equivalently, the minimum vertex cover problem) such as [*18*] [*19*] and many more. We believe that such suboptimal solutions are adaptable to our problem and can be integrated in our retrieval algorithm in the future in order to give a better performance with a more reasonable number of iterations.

Finally, the retrieval algorithms presented in this work are all synchronous in the sense that, at each iteration, dynamic and activation rules are always applied to all clusters. In future work, we will consider asynchronous methods which can take into account the fact that some clusters may reach their final state before others, so application of dynamic and activation rules could then be limited to only a subset of clusters. We also aim at extending the scope of the algorithms presented in this paper to deal with other types of input messages, such as distorted ones in which some submessages have slightly modified values from their origin.


## Acknowledgment

The authors would like to thank Zhe Yao and Michael Rabbat at McGill University for interesting discussions that helped leading to the proposal of the new retrieval techniques introduced in this paper. We also acknowledge funding from the NEUCOD project at the electronics department of Télécom Bretagne and support from the European Research Council (ERC-AdG2011 290901 NEUCOD).


## References


[1] J. R. Anderson and G. H. Bower, Human associative memory.: Psychology press, 2013.

[2] D. J. Willshaw, O. P. Buneman, and H. C. Longuet-Higgins, "Non-holographic associative memory," Nature, vol. 222, pp. 960-962, 1969.

[3] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in Proceedings of the National Academy of Science, USA, vol. 79, 1982, pp. 2554-2558.

[4] V. Gripon and C. Berrou, "Sparse neural networks with large learning diversity," IEEE Transactions on Neural Networks, vol. 22, no. 7, pp. 1087-1096, 2011.

[5] V. Gripon and M. Rabbat, "Maximum Likelihood Associative Memories," in Proceedings of the IEEE Information Theory Workshop (ITW), 2013, pp. 1-5.

[6] B. K. Aliabadi, C. Berrou, V. Gripon, and X. Jiang, "Learning sparse messages in networks of neural cliques," IEEE Transactions on Neural Networks and Learning Systems, August 2012, in press.

[7] G. Palm, "Neural associative memories and sparse coding," Neural Networks., vol. 37, pp. 165-171, 2013.

[8] V. B. Mountcastle, "The columnar organization of the neocortex," Brain, vol. 120, no. 4, pp. 701-722, 1997.

[9] V. Gripon and X. Jiang, "Mémoires associatives pour observations floues," in Proceedings of XXIV-th Gretsi seminar, September 2013, in press.

[10] D. P. Buxhoeveden and M. F. Casanova, "The minicolumn hypothesis in neuroscience," Brain, vol. 125, no. 5, pp. 935-951, 2002.

[11] F. T. Sommer and G. Palm, "Improved bidirectional retrieval of sparse patterns stored by Hebbian learning," Neural Networks, vol. 12, no. 2, pp. 281-297, 1999.

[12] F. Schwenker, F. T. Sommer, and G. Palm, "Iterative retrieval of sparsely coded associative memory patterns," Neural Networks, vol. 9, no. 3, pp. 445-455, 1996.

[13] L. Cruz et al., "A statistically based density map method for identification and quantification of regional differences in microcolumnarity in the monkey brain," Journal of Neuroscience Methods, vol. 141, pp. 321-332, 2005.

[14] E. G. Jones, "Microcolumns in the cerebral cortex," PNAS, vol. 97, no. 10, pp. 5019-5021, 2000.

[15] V. Gripon and C. Berrou, "Nearly-optimal associative memories based on distributed constant weight codes," in Proceedings of Information Theory and Applications Workshop, USA, 2012, pp. 269-273.

[16] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," The Bulletin of Mathematical Biophysics, vol. 5, no. 4, pp. 115-133, 1943.

[17] X. Jiang, V. Gripon, and C. Berrou, "Learning long sequences in binary neural networks," in The Fourth International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE 2012)

IARIA, 2012, pp. 165-170.

[18] X. Xu and J. Ma, "An efficient simulated annealing algorithm for the minimum vertex cover problem," Neurocomputing, vol. 69, no. 7, pp. 913-916, 2006.

[19] X. Geng, J. Xu, J. Xiao, and L. Pan, "A simple simulated annealing algorithm for the maximum clique problem," Information Sciences, vol. 177, no. 22, pp. 5064-5071, 2007.

[20] Z. Yao, V. Gripon, and M. G. Rabbat, "A massively parallel associative memory based on sparse neural networks," Submitted to IEEE Transactions on Neural Networks and Learning Systems, arXiv:1303.7032v1 [cs.AI].