Storing Sparse Messages in Networks of Neural Cliques

Behrooz Kamary Aliabadi, *Student Member, IEEE*, Claude Berrou, *Fellow, IEEE*, Vincent Gripon, *Member, IEEE*, and Xiaoran Jiang, *Student Member, IEEE*

***Abstract*— An extension to a recently introduced binary neural network is proposed to allow the storage of sparse messages, in large numbers and with high memory efficiency. This new network is justified both in biological and informational terms. The storage and retrieval rules are detailed and illustrated by various simulation results.**

***Index Terms*— Associative memory, error correcting code, machine learning, parsimony, recurrent neural network, sparse coding.**

I. Introduction

THE brain stores information with high concern for parsimony. For obvious reasons of both restricted available resource and energy limitations, the pieces of information memorized by the biological neural network result from a strong compression of the physical stimuli stemming from the "richly detailed world" [1]. For the same reasons, storage and retrieving operations involve a few cerebral regions and a few neurons at each time. The way the brain recruits and organizes these small populations of neurons to perform the so-called "sparse coding" of mental information [2]–[5] has still to be discovered.

On the other hand, in healthy brains, mental information is robust and durable, therefore must be redundant. Without redundancy, mental information would be too frail facing the physicochemical aggressions that the brain constituents suffer continuously and for so many years.

In those terms, the situation is very similar to the well-known source coding/channel coding scheme of modern telecommunication systems: first, information is cleared of useless components and then intelligent redundancy is added to allow error correction at the receiver side [6]. This rationale has recently led to the proposal of a new neural architecture combining recurrent binary networks and error correcting codes [7]. Actually, it was demonstrated in this latter paper that no error-correcting code had to be artificially added. Indeed, any graph, whatever the support, biological or artificial, may contain highly redundant codewords when they are assimilated to specific graph patterns, namely cliques. Exploiting this very beneficial property, multipartite clique-based networks have been proposed in [7] to store messages with large diversity (the maximum number of messages that can be stored) and capacity (the maximum amount of stored binary information), as well as strong robustness toward erasures or errors. However, these networks were devised in such a way that all the clusters resulting from multipartition are used in the memorization procedure. Therefore, they do not correspond directly to the sparse coding vision of mental information. Moreover, the diversity of these networks is proportional to the square of the number of nodes in each cluster, and not to the square of the total number of nodes. To lift these restrictions, the clique-based networks have to be reconsidered and reassessed with respect to the storage of sparse messages, that is, messages that do not call for the complete network but only for parts of it. The storage cost (i.e., the required binary resource) for sparse messages being significantly reduced compared with that of nonsparse ones, the improvement in diversity is appreciable.

The acquisition and storage of sparse messages have been an important research topic in various fields; among those are compressed sensing [8], [9], sparse regression [10] and information theory [11]–[13]. Regarding the possible applications of the network presented in this paper, the most obvious one could be the implementation of dictionaries for the sparse acquisition and representation of data, such as speech, images or semantics [14]–[16]. In this kind of applications, the network would acquire and store numerous dictionary codewords materialized by small cliques and the concomitant activation of some of these cliques would materialize a particular element of knowledge. The dictionary may be prefixed (e.g., using wavelets as codewords) or may evolve with the coming of new elements. The advantages offered by the network for the implementation of such dictionaries would be high capacity, full parallelism, and error correction capability. The rest of this paper, which does not deal with such advanced applications but deepens the informational properties of the proposed network, is organized in seven sections. Section II recalls the principles and notations of the clique-based networks and also proposes a slight improvement of the retrieving algorithm introduced in [7]. Some considerations about the biological plausibility of the architecture are propounded as well. In Section III, the storage and retrieving algorithms of sparse messages are described. Sections IV and V provide some theoretical analysis and simulation results for the basic applications of associative memory and set implementation. Section VI gives

Manuscript received August 16, 2012; revised August 17, 2013; accepted October 1, 2013. Date of publication November 8, 2013; date of current version April 10, 2014. This work was supported by the European Research Council under Grant ERC-AdG2011 290901 NEUCOD.

The authors are with the Electronics Department, Brest 29238, France, and also with the Laboratory for Science and Technologies of Information, Communication and Knowledge, Brest 29238, France (e-mail: behrooz.kamaryaliabadi@telecom-bretagne.eu; claude.berrou@telecom-bretagne.eu; vincent.gripon@telecom-bretagne.eu; xiaoran.jiang@telecom-bretagne.eu).

This paper includes multimedia material available online at http://ieeexplore.ieee.org (File size: 27 Kbytes).

Digital Object Identifier 10.1109/TNNLS.2013.2285253

an illustration of the network performance, in terms of error correction for classification. In section VII, the question of sparse messages with variable degrees of sparsity is taken up. Finally, some comments about the openings of this new kind of neural networks are proposed in the conclusion.

II. Networks of Neural Cliques

A. Summary

Consider a network with n binary nodes linked by binary edges (that is, each edge exists with weight 1 or does not exist). This network can then be described by a nonweighted, nonoriented graph whose nodes may be activated or not, with respective values 1 and 0. The network is split into c clusters, each containing $l = n/c$ nodes. For reasons that will be given later, these nodes are called fanals. Though any alphabet with cardinality l could be considered in the representation of information stored by the network, we focus on binary messages to allow classic computations or estimations of storage properties. Therefore, l is taken to be a power of 2: $l = 2^\kappa$. With an input binary message m of length $k = c\kappa$ bits to store, is associated a unique set of fanals, one per cluster, using the mapping: $C : m = (m_1, \ldots, m_i, \ldots, m_c) \rightarrow (f(m_1), \ldots, f(m_i), \ldots, f(m_c))$ where m_i of length κ bits is the submessage or character associated with the ith cluster, $f(.)$ is the function that maps each submessage to a unique fanal in the corresponding cluster.

Thus, the network stores a given message by selecting one fanal per cluster and connecting these c fanals to build a fully interconnected subgraph, that is, a clique. In other words, storing the particular message m is equivalent to storing the pattern $C(m)$. If $\mathcal{M}$ is the set of messages stored by the network, $\mathcal{W}(m)$ the set of edges that has to be created to store particular message m, the ensemble $\mathcal{W}$ of existing edges resulting from the storage of $\mathcal{M}$ is simply given by

$$\mathcal{W} = \bigcup_{m \in \mathcal{M}} \mathcal{W}(m). \tag{1}$$

This result does not depend on the order in which messages are presented, and storing a new message can be done at any moment. This very simple storage rule leads to a completely binary network. Note that no connection is established within a cluster.

The retrieving algorithm is a two step, possibly iterative, procedure. First, at the global scale and from what is known of the stimulus (that is, some of the submessages m_i), the corresponding fanals send unitary signals toward the network through established connections and then, the contributions are added at each node. After this message passing step, at the local scale of each cluster, a winner-take-all rule is performed. Noting $v(n_{ij})$ the value of the jth fanal in the cluster with index i ($1 \leq i \leq c$; $1 \leq j \leq l$) and $w_{(i'j')(ij)}$ the weight (0 or 1) of the edge between the fanals $n_{i'j'}$ and n_{ij}, the global decoding equation is

$$v(n_{ij}) \leftarrow \sum_{i'=1}^{c} \max_{1 \leq j' \leq l} (w_{(i'j')(ij)} v(n_{i'j'})) + \gamma\, v(n_{ij}). \tag{2}$$

A memory effect with parameter γ is added to the message passing procedure. This relation is slightly different from the one proposed in [7], as the summation on node n_{ij} of the signals stemming from the same cluster with index i' is replaced with a selection of its maximum. The reason why equation (2) is now preferred is detailed in [17]. Briefly, the max function is justified by the following argument: when several fanals are active within the same cluster, that is, in the presence of ambiguity, this very cluster must not impact on the rest of the network more than in the case of only one active fanal. In other words, ambiguity is tolerated at the local level but not favored at the global scale. As for the local winner-take-all selection, the relations are the same as in [7], precisely

$$\forall i,\ 1 \leq i \leq c : v_{\max,i} \leftarrow \max_{1 \leq j \leq l} (v(n_{ij}))$$

$$\forall i \text{ and } j,\ 1 \leq i \leq c, 1 \leq j \leq l :$$

$$v(n_{ij}) = \begin{cases} 1 & \text{if } v(n_{ij}) = v_{\max,i} \text{ and } v_{\max,i} \geq \sigma, \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

After these operations, all fanals have value of 1 or 0, which explains why this network is said to be binary, even if transitory fanal values may be larger than 1. σ is a threshold, which is quite comparable with that of the McCulloch–Pitts model of neuron [18] and which may be used as an additional level of control. In normal conditions and classical applications, all the $v_{\max,i}$ computed in the c clusters are equal and can then be reduced to a single maximal score $v_{\max}$. A counter-example would be, for instance, a network in which some established connections have disappeared, due to some physical flaw; in these conditions, some signals may be missing in the computation described by (2), preventing one or more fanal values from reaching $v_{\max}$.

After the calculations formulated by (3), more than one fanal may remain activated within the same cluster. In this ambiguous situation, a repetition of (2) and (3) may be profitable and the process may need several iterations to converge toward a fixed point.

B. Biological Considerations

As already pointed out in the introduction, parsimony is a prominent characteristic of the brain organization and functioning. The amount of data which are continually conveyed by the nervous system, the visual cortex for instance, is gigantic. Yet, what is retained by the brain for a possible later exploitation (e.g., the description of a flower) requires much less information than the original physical stimuli. These stimuli undertake several filtering operations, from the sensory levels (e.g., points, lines, shapes and colors) to the most cognitive ones (names, concepts, etc.), each level adapting the elements of knowledge to its own mode of representation and storage [19].

It is now surmised [3], [20] that these different modes of representation of mental information, either sensory or cognitive, correspond to the activation of small populations of neurons in dedicated parts of the cortex. Under some strongly reductionist hypotheses, it is possible to describe this biological network by a recurrent graph in which the activation

of some nodes may form a pattern, which could be considered as the material representation of a stored piece of information. Actually, from the point of view of informational organization, the fundamental processing unit acting as a repetitive node in the biological graph is likely not a neuron alone, but the microcolumn (also called the minicolumn) [21]–[23]. This very heterogeneous group of about 100 neurons repeats itself quasi-uniformly over the about 20 square decimeters of the human gray matter. Therefore, the microcolumn, as an "identical repeating unit", can be considered as a node in a graph, able to receive and send many signals from and toward the rest of the network, with the same informational abilities everywhere in the cortex. As written in [23]: "current data on the microcolumn indicate that the neurons within the microcolumn receive common inputs, have common outputs, are interconnected, and may well constitute a fundamental computational unit of the cerebral cortex."

These microcolumns are grouped into columns whose populations are various, from some tens to some hundreds. These columns are believed to contain microcolumns that react to the same family of stimuli (e.g., the value of an angle in the visual cortex). Rising in the neural hierarchy, columns gather in macrocolumns and then several macrocolumns together constitute the so-called functional areas of the brain.

We propose to liken the fanals of the clique-based networks to microcolumns, the clusters to columns, and finally the network itself to a macrocolumn. For the sake of simplicity, all the clusters have same cardinality l in this paper, but as already stated, biological columns are of various sizes. The term fanal has been adopted to represent a node in the clique-based networks for two reasons. First, it expresses the reality of a cluster in which only one fanal can be lighted during the storage process; second, it makes clear that a node of the graph is not a single neuron, but a group of neurons, namely a microcolumn.

III. Processing Sparse Messages

Consider a message of length χ composed of characters taken in an alphabet denoted by $\mathcal{A}$, the neutral character 0 being one of its elements. According to the classical definition, a sparse message is a message containing few nonzero characters. Such messages are the subject of many current studies in the field of information theory, especially for the compressed sensing application [8], [9], or also for classification [24]. We have here to change the definition and say that a message is sparse when it contains few significant characters, that is, a limited number of informative characters, in specific locations, the others being of no concern. To take a simple example, if $\mathcal{A}$ is the ensemble of positive or null integers less than eight, a sparse message of length $\chi = 24$, according to the conventional definition, could be: (000500060010000030000000), whereas we consider here messages such as: (− − −5 − − − 6− −1− − − − −3− − − − − − −), each dash meaning "blank". The blank characters do not need to be stored by the neural network and do not require any fanal to be recruited in the storage of the message, while a classical sparse message would need the writing of the 0s. This vision of sparse messages is much more in accordance with the way the brain learns its knowledge elements, with the concern for parsimony.

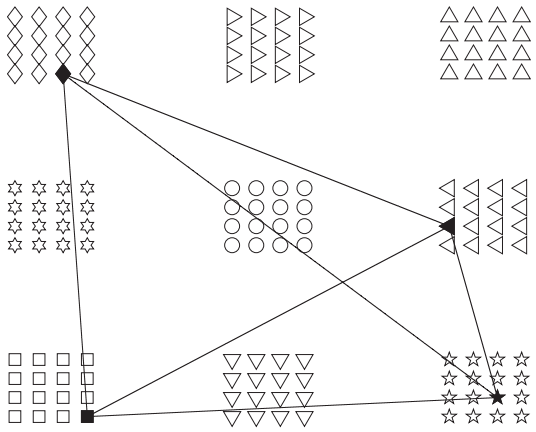


Fig. 1. Network composed of nine clusters of 16 fanals each. A first clique with four vertices has been formed in the network.

A. Storing

Until Section VII, we consider the storage of messages of length χ with the same number c of a few significant characters that we call indifferently the clique or message order in the sequel. Alphabet $\mathcal{A}$ contains $l = 2^\kappa$ nonblank elements and, in the same way as proposed in [7], the network is split into χ clusters, each containing l fanals. Therefore, the network contains $n = \chi l$ fanals and $\chi(\chi-1)l^2/2$ potential connections, that is, a binary resource of

$$Q = \frac{\chi(\chi-1)l^2}{2} \quad \text{[bits]}. \tag{4}$$

Starting from an initial state with no connection at all, the storage of a first message will recruit c fanals and establish $c(c-1)/2$ connections to buildup a clique (see an example in Fig. 1 for $\chi = 9$, $l = 16$, and $c = 4$ and one established clique with $4 \times 3/2 = 6$ connections). The probability that any particular connection does not belong to this clique is supposed to be: $1 - c(c-1)/\chi(\chi-1)l^2$ (this is not rigorously exact because the connections created by the clique are correlated, but the effect of this correlation is negligible when a large number of cliques have been formed). So, after the storage of M independent identically distributed (i.i.d.) random messages (each character being drawn at random), this probability, which we assume to give directly the expected connection density, is

$$d = 1 - \left(1 - \frac{c(c-1)}{\chi(\chi-1)l^2}\right)^M. \tag{5}$$

Reciprocally, given a density d, the number M of i.i.d. stored messages is

$$M = \frac{\log(1-d)}{\log\left(1 - \frac{c(c-1)}{\chi(\chi-1)l^2}\right)}. \tag{6}$$

Fig. 2 shows the evolution of d as a function of M, for $\chi = 100$, $l = 64$ and various values of c. For low values of d, we have

$$d \approx \frac{c(c-1)M}{\chi(\chi-1)l^2} \tag{7}$$

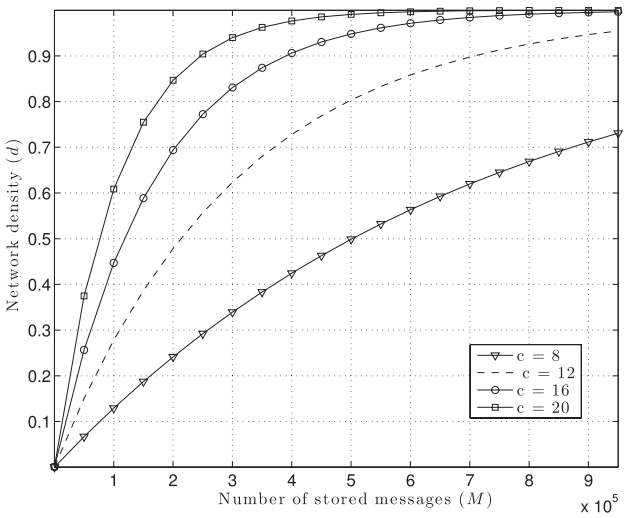


Fig. 2. Density of the network connections as a function of the number M of stored messages, for $\chi = 100$, $l = 64$ and four values of c.

and reciprocally

$$M \approx \frac{\chi(\chi-1)l^2}{c(c-1)}d \approx \frac{\chi^2 l^2}{c(c-1)}d \approx \frac{n^2 d}{c(c-1)} \tag{8}$$

for $\chi \gg 1$. This very simple result shows that, for a given density d and a particular value of the clique order, the number of messages that the network is able to store is proportional to the square of the number of nodes, or fanals. This quadratic law has for instance to be compared with the well-known sublinear law of Hopfield networks (see [7] for details) or other comparable laws obtained with networks based on weighted connections.

We can express the amount of binary information B stored by the network after the memorization of M messages as

$$B = M\left[\log_2\left(\binom{\chi}{c}\right) + c\kappa\right]. \tag{9}$$

The first term between the parentheses accounts for the choice of the clusters in the storage of one message, with c significant characters among χ. The second term represents the binary content of each message, where $\kappa = \log_2(l)$. Finally, we define efficiency η as the ratio of B and Q:

$$\eta = \frac{B}{Q} = \frac{2M\left[\log_2\left(\binom{\chi}{c}\right) + c\log_2(l)\right]}{\chi(\chi-1)l^2}. \tag{10}$$

For efficiency equal to 1, this formula leads to an upper bound for M, called the efficiency-1 diversity of the network[1]

$$M_{\max} = \frac{\chi(\chi-1)l^2}{2\left[\log_2\left(\binom{\chi}{c}\right) + c\log_2(l)\right]}. \tag{11}$$

For instance, with $\chi = 100$, $l = 64$, and $c = 16$, $M_{\max}$ is around 1.30×10^5 messages of 156 bits each. Density calculated from (5) is then about 0.54.

From (11), we observe that the largest values of $M_{\max}$ are obtained for the lowest clique orders c. This is quite natural: for a given binary resource, the shorter the messages are, the more numerous they can be. However, the smaller a clique is, the less robust it is facing possible flaws (e.g., errors, erasures or over storage). The choice of c is thus the result of a tradeoff between the density and robustness. The optimal values for the applications of associative memory will be discussed in Section IV.

B. Retrieving

The basic equations for the retrieving of stored messages are still (2) and (3), but in which some adjustments have to be made to consider sparsity

$$v(n_{ij}) \leftarrow \sum_{i'=1}^{\chi} \max_{1\le j'\le l}\left(w_{(i'j')(ij)}\ v(n_{i'j'})\right) + \gamma\, v(n_{ij}) \tag{12}$$

$$\forall i,\ 1 \le i \le \chi : v_{\max,i} \leftarrow \max_{1\le j\le l}(v(n_{ij}))$$

$$v_{\max} \leftarrow \max_{1\le i\le\chi}\left(v_{\max,i}\right)$$

$\forall i$ and j, $1 \le i \le \chi, 1 \le j \le l$:

$$v(n_{ij}) = \begin{cases} 1 & \text{if } v(n_{ij}) = v_{\max} \text{ and } v_{\max} \ge \sigma_i \\ 0 & \text{otherwise} \end{cases}. \tag{13}$$

Let us first point out that, despite the fact that a stored message uses a restricted number c of clusters at each time, these clusters vary from one message to another. Therefore, in common situations, the decoding procedure has to address the whole network uniformly. To achieve this, c is replaced with χ as the upper index of the summation in (12). Second, the winner-take-all rule expressed by (3) for each cluster has now to be extended to the whole network to find the appropriate clusters in search of a particular message, if their indices are not known. To achieve this, (13) retains the maximum value $v_{\max}$ of maximum scores obtained in all clusters and assigns $v_{\max}$ as the condition to reach for all fanals to be active. Finally and in general cases, threshold σ is now cluster dependent and thus is denoted σ_i. Indeed, in some applications, all the clusters may not be involved in a specific task. For instance, if the network is asked to recognize, or reject, a particular message with $c < \chi$ significant characters, only the corresponding clusters have to be processed. The threshold of the other clusters is then set at a high unreachable value. In another situation if the network has to store not only binary messages, but also some kind of cognitive features, the choice of distinct values for σ_i may orientate the network decoding toward constrained solutions. A high value for the threshold of a particular cluster would mean that this has no relevance in the current process.[2]

IV. Associative Memory

Suppose that a network with parameters χ and l has stored M i.i.d. random messages with order c, that is, M cliques with c vertices. We want to know what the error probability is in the recovery of one stored message when $c_e < c$ clusters

[1]However, because messages are not ordered in an associative memory and therefore require less resource than ordered messages, efficiency larger than 1 is not inconceivable [7].

[2]A C++ software, performing the storage and retrieval of sparse messages, in the way described in this section, is available as a multimedia content in IEEEXplore.

are not provided with information. Two extreme cases have to be considered. Either the indices of the c_e missing clusters are completely unknown—we call this most severe case *blind recovery*—or they are totally known—it is then called *guided recovery*.

A. Blind Recovery

The first and most obvious reason why the retrieving of a previously stored message may fail is because at least one other valid message shares the same known characters. For i.i.d. messages and large enough values of $c - c_e$ and l, this probability is negligible. The second reason is the possible existence of spurious cliques (i.e., nonvalid cliques resting inopportunely on the edges of valid cliques) that would contain the known fanals and then interfere in the retrieving process.

Because any subset of a clique is a clique, the probability that no such spurious clique, of any order, exists is the probability that there is no spurious clique with $c - c_e + 1$ vertices, that is only one more than the known characters. Therefore, to determine the probability of nonexistence of spurious cliques of any order, it is sufficient to calculate the probability P' of not having a spurious clique with $c - c_e + 1$ vertices. Given the density d of the network connections, the probability that a particular fanal does not form a clique with the $c - c_e$ known fanals is $1 - d^{c-c_e}$. The number of fanals available to form such a spurious clique is $c_e(l-1)+(\chi-c)l$. Therefore, the probability that no such clique exists is

$$P' = \left(1 - d^{c-c_e}\right)^{c_e(l-1)+l(\chi-c)}. \tag{14}$$

Because of the aforementioned arguments, the probability of nonexistence of spurious cliques, which we assimilate to the probability P_r of retrieving the correct message, is

$$P_r = \left(1 - d^{c-c_e}\right)^{c_e(l-1)+l(\chi-c)}. \tag{15}$$

Performing several iterations does not help the recovery in the presence of spurious cliques. The appendix gives an example of why such a spurious clique makes the decoding fail, even in an iterative process. The probability of recovery error P_e is then given by

$$P_e = 1 - P_r = 1 - \left(1 - d^{c-c_e}\right)^{c_e(l-1)+l(\chi-c)} \tag{16}$$

which can be approximated, for small values of d, as

$$P_e \approx (c_e(l-1) + (\chi - c)l)d^{(c-c_e)}. \tag{17}$$

Note that, even without erasure ($c_e = 0$), the error probability in blind recovery is not zero. This is because a given stored clique of order c may belong to a spurious clique with order $c+1$, with approximated probability $(\chi - c)ld^c$.

The curve (a) of Fig. 3 shows the simulated error rate in the blind recovery of messages, as a function of their number M, for $\chi = 100$, $l = 64$, $c = 12$, and $c_e = 3$, after one iteration of (12) and (13). All thresholds σ_i are equal to zero in the decoding process and memory parameter γ is equal to 1. Values given by (16) are also indicated, showing good correspondence between theory and simulation.

From relations (5) and (16), it is possible to link M and c, for a predetermined value of $P_e = P_0$. To do this easily, we use

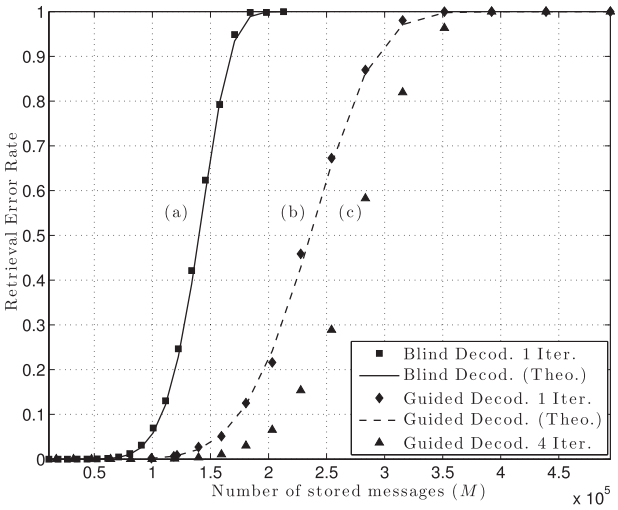


Fig. 3. Error rate in both blind and guided recovery of M i.i.d. messages with order $c = 12$ in a network composed of $\chi = 100$ clusters of $l = 64$ fanals each. $c_e = 3$ clusters have no initial information.

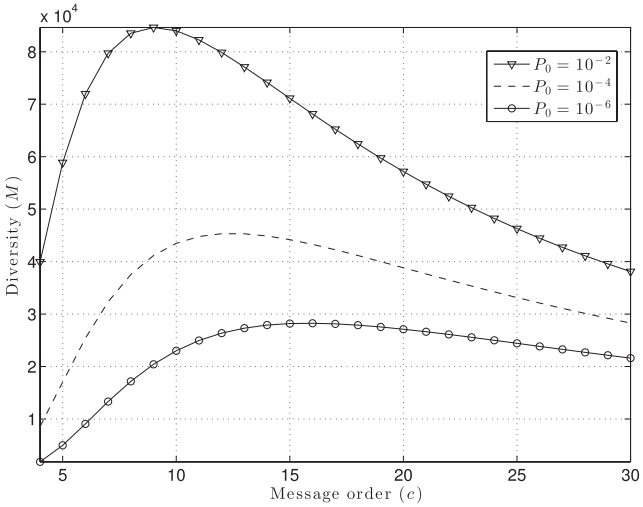


Fig. 4. Diversity of the network composed of $\chi = 100$ clusters of $l = 64$ fanals each, as a function of the message order c, when a quarter of the c clusters have no information at the initialization step [relation (18) with $\alpha = 0.25$] and for a given error probability P_0 in blind recovery.

the approximations given by (7) and (17), with the additional and rough hypothesis: $\chi \gg c \gg 1$. We also set c_e as a fraction α of c: $c_e = \alpha c$. All this results in

$$M \approx \left(\frac{\chi l}{c}\right)^2 \left(\frac{P_0}{\chi l}\right)^{\frac{1}{(1-\alpha)c}}. \tag{18}$$

Fig. 4 shows the variation of diversity M, as a function of c, for $\chi = 100$, $l = 64$, $\alpha = 0.25$ and three values of P_0. These curves show that there exists a value of c that maximizes the diversity of the network, for a given value of P_0. This value, denoted c_{opt}, is estimated by deriving (18) (through the logarithm) with respect to c, and finding the condition for extremum. The computation gives

$$c_{\text{opt}} \approx \frac{\log\left(\frac{\chi l}{P_0}\right)}{2(1-\alpha)}. \tag{19}$$

For instance, with $\chi = 100$, $l = 64$, $\alpha = 0.25$, we have c_{opt} equal to 9 and 15 for $P_0 = 10^{-2}$ and 10^{-6}, respectively. The corresponding diversities given by (18) are around 70 000 and 25 000 and efficiencies, as formulated by (10), are 33% and 18%. As can be surmised, material efficiency (η) and effectiveness in restoring the messages (P_e) are conflicting, but not so sharply.

B. Guided Recovery

In this favorable case, some characters of the message are missing, but their supporting clusters are known. The thresholds σ of irrelevant clusters are then set to an unreachable value. The situation comes down exactly to that described in [7, Section VI] in which all clusters are systematically known to retrieve a message. The error probability, after one iteration, is given by

$$P_e = 1 - \left(1 - d^{c-c_e}\right)^{(l-1)c_e}. \tag{20}$$

The simulated error rate, after several iterations, may be notably less than P_e. The curves (b) and (c) of Fig. 3 show the evolution of the error rate as a function of M, after one and four iterations, respectively, and with the same parameters as curve (a). The theoretical values given by (20) are also indicated.

The gap in performance between the blind and guided recovery is not considerable in terms of diversity. When guided, instead of blind, decoding is performed and for a given error rate, diversity M [which is roughly proportional to density d, through approximation (7)] is higher by about 50%, after one iteration and 100% after four iterations.

V. Set Implementation

As in [7, Section V], we consider a simple set implementation application. The network with parameters χ and l stores M i.i.d. messages with same order $c < \chi$. The decoding procedure is still described by relations (12) and (13) with $\gamma = 1$, but thresholds σ are now equal to c for the clusters under test (those which have a fanal activated) and a higher unreachable value for the others, to prevent them from interfering. The network is then asked whether a message is known by it or not, that is, practically whether the decoding procedure will accept the stimulus without modification or not. The recognition of stored messages is always successful, since all the activated fanals will obtain the maximal score (i.e., c), as soon as the first iteration has finished and no other one can be the winner within the clusters under test. Therefore, the type I error is naught:

$$P_{\text{type I error}} = 0. \tag{21}$$

Because the thresholds of the irrelevant clusters are set to unreachable values, the only possibility for the network to accept a wrong message, that is, a type II error, is the existence of a spurious clique of order c in the clusters under test. Assuming again that not only messages, but also connections are i.i.d., the probability of having such a spurious clique is

$$P_{\text{type II error}} = d^{\frac{c(c-1)}{2}} = \left(1 - \left(1 - \frac{c(c-1)}{\chi(\chi-1)l^2}\right)^M\right)^{\frac{c(c-1)}{2}}. \tag{22}$$

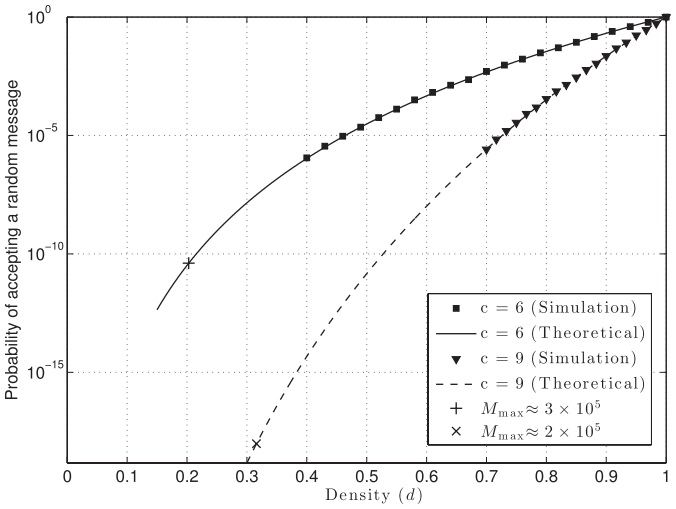


Fig. 5. Type II error rate as a function of the density of the network composed of $\chi = 100$ clusters of $l = 64$ fanals each, with cliques of order $c = 6$ or 9. Densities corresponding to efficiency-1 diversities ($M_{\max}$) are also indicated.

Fig. 5 shows the type II error rate obtained from simulation, on the one hand, and relation (22), on the other hand, as a function of density d. The parameters are $\chi = 100$, $l = 64$ and $c = 6$ or 9. This figure also shows the values of efficiency-1 diversity $M_{\max}$, as calculated from (11), showing that low error rates may be obtained even for efficiencies higher than 1. For instance, with $c = 9$, diversities more than three times $M_{\max}$ [with $d = 0.68$ according to (5)] can be attained while keeping the error rate below 10^{-6}.

VI. Blurred Messages

Because the proposed network uses cliques as support of information and since cliques are redundant structures, the error correction of distorted or blurred incident messages is possible. To illustrate this property, we consider a network with parameters $\chi = 100$, $l = 64$, and $c = 12$. M messages are stored using c contiguous clusters, the first one being randomly located at a position between 1 and $\chi - c + 1 = 89$. After the storage phase, messages are presented to the network after systematic permutation of two consecutive characters, in a cyclic way. For instance, if the messages were, among other things, 12 letter English words, the stored word "intelligence" may be received either as "nietllgineec" or "etnleilegcni". If the network is able to recognize "intelligence" from these pairwise permuted versions, it would also be able to correct less disturbed words, such as "intleligecne" or "intellgienec" (a task that the human brain is also reputed to be capable of [25]).

To allow the network to cope with the upset order of characters, the initialization of the retrieving process has to be modified: when a fanal is activated in a cluster, then the equivalent fanals (i.e., fanals with the same index values j) of the adjacent clusters have also to be activated. Therefore, at the beginning of the retrieving process, each cluster contains three active fanals. The stored clique intelligence being contained in the activated subgraph composed of the $12 \times 3 = 36$ nodes, the decoding procedure described by (12) and (13) will hopefully, after several iterations, switch

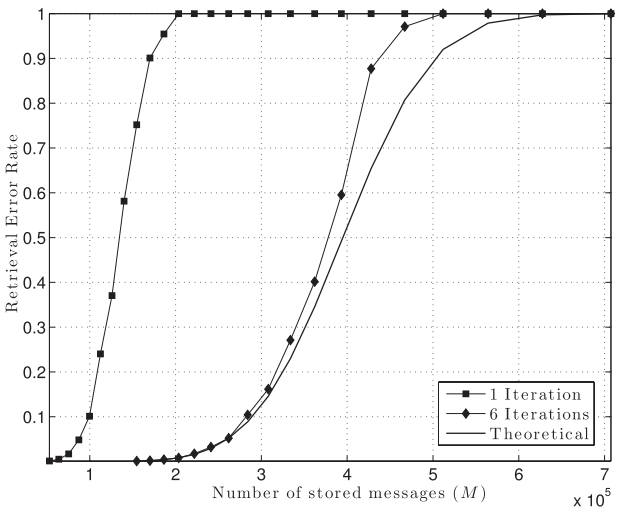


Fig. 6. Error rate in the retrieving of messages distorted by permutations of contiguous characters. The network parameters are $\chi = 100$ and $l = 64$, and messages have order $c = 12$.

off the irrelevant fanals and display the word with the right order.

The probability for the decoder to fail is again related to the existence of spurious cliques. The most likely are those which share $c-1$ vertices with the stored clique and use one wrong fanal in the remaining cluster as the last vertex. For a given set of $c-1$ correct vertices, the probability that such a spurious clique exists is d^{c-1}. Because there are $2c$ possible patterns having $c-1$ correct vertices and a wrong one (of two possible) in the last cluster, the probability that a false clique may exist, giving directly the error probability in the retrieving process, is

$$P_e = 1 - \left(1 - d^{c-1}\right)^{2c} \approx 2cd^{c-1} \quad \text{for } d \ll 1. \tag{23}$$

Fig. 6 shows the error rate obtained by simulation after one and six iterations, as well as the theoretical curve deduced from (23). Because messages that are presented to the network are strongly distorted, several iterations are required to approach the theoretical performance. By comparing abscissas in Figs. 3 and 6, we note that the network is able to correct wrong messages as easily as, even a little better than, erasures. If full anagrams, instead of contiguous permutations, were considered, c fanals would have to be activated in each cluster at the initialization step, instead of only 3. The error probability in deciphering these anagrams would then be given by

$$P_e = 1 - \left(1 - d^{c-1}\right)^{c(c-1)} \approx c(c-1)d^{c-1} \quad \text{for } d \ll 1. \tag{24}$$

VII. Variable Order Messages

We want now to assess the ability of the network with parameters χ and l to store and retrieve messages with various values of c. From Fig. 4, we can notice that a relatively large set of orders c around c_{opt} may be employed without much deteriorating the retrieving performance of the network. For instance, with a targeted error rate P_0 equal to 10^{-4} and for this particular network with $\chi = 100$ and $l = 64$, values of c chosen between 8 and 20 would decrease the diversity by about 10% or less from the optimal value obtained with $c_{\text{opt}} = 12$.

To formalize the general case, let us consider clique orders distributed between $c_{\min}$ and $c_{\max}$, such that $1 < c_{\min} \leq c_{\max} < \chi$. Following the same rationale that led to relation (5), we can express the density of the network as

$$d = 1 - \prod_{c=c_{\min}}^{c_{\max}} \left(1 - \frac{c(c-1)}{\chi(\chi-1)l^2}\right)^{M_c} \tag{25}$$

where M_c is the number of messages stored with order c. If c is uniformly distributed between $c_{\min}$ and $c_{\max}$, M_c is equal to M/λ, M still being the total number of i.i.d. messages and $\lambda = c_{\max} - c_{\min} + 1$.

The storage and recovery rules are still those given in Section III, with threshold values σ_i, all equal or not, depending on the application. The amount of binary information borne by a particular message $m \in \mathcal{M}$, materialized by a clique with order c_m, is now

$$\log_2\left(\binom{\chi}{c_m}\right) + c_m \log_2(l).$$

To write this formula, we have considered that the value of c_m does not result from a choice but is imposed by message m. Therefore, the value of c_m does not bring any information about this particular message.

Efficiency η is given by

$$\eta = \frac{2\sum_{m\in\mathcal{M}} \left(\log_2\left(\binom{\chi}{c_m}\right) + c_m \log_2(l)\right)}{\chi(\chi-1)l^2}. \tag{26}$$

It is not easy to exploit this formula in the general case. For a uniform distribution of orders c between $c_{\min}$ and $c_{\max}$, this amounts to a more convenient formula:

$$\eta = \frac{2M\sum_{c=c_{\min}}^{c_{\max}} \left(\log_2\left(\binom{\chi}{c}\right) + c \log_2(l)\right)}{\lambda\chi(\chi-1)l^2}. \tag{27}$$

As in Section III-A, it is then possible to find the efficiency-1 diversity by setting $\eta = 1$. This gives

$$M_{\max} = \frac{\lambda\chi(\chi-1)l^2}{2\sum_{c=c_{\min}}^{c_{\max}} \left(\log_2\left(\binom{\chi}{c}\right) + c \log_2(l)\right)}. \tag{28}$$

For instance, taking $c_{\min} = 12$ and $c_{\max} = 20$ gives $M_{\max} = 1.27 \times 10^5$, a value very close to that obtained for c constant and equal to 16, as observed in Section III-A.

Finally, to finish this survey of sparse messages stored in network of neural cliques, we consider a network with parameters χ and l storing M i.i.d. messages of variable and uniformly distributed orders c and the blind recovery which is done without the knowledge of $c_e = \alpha c$ submessages. Relation (16) being applicable to each of the subset of the messages

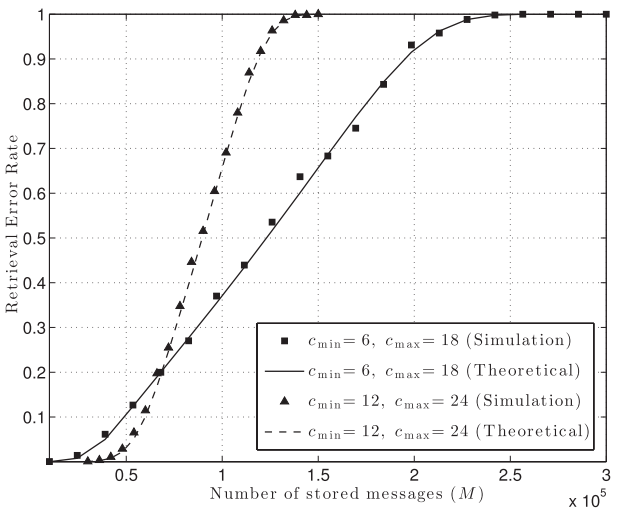


Fig. 7. Blind recovery error rate of messages with variable order between $c_{\min}$ and $c_{\max}$ in a network with parameters $\chi = 100$ and $l = 64$, and when $c_e = c/4$ clusters (on average) have no initial information. One iteration.

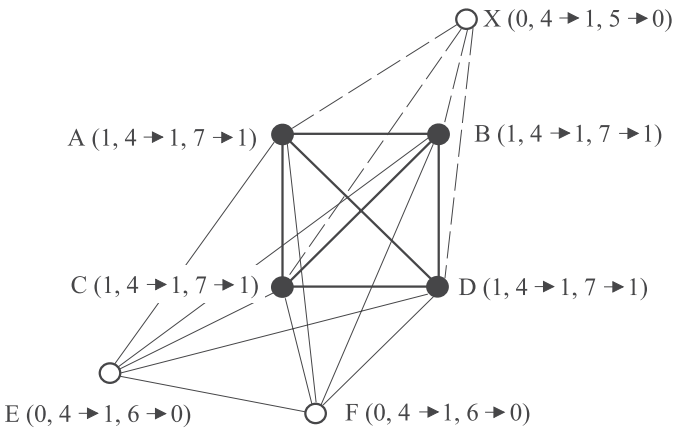


Fig. 8. Example of failure in the retrieving of a stored clique (A-B-C-D-E-F) due to a spurious clique of shorter order (A-B-C-D-X).

with particular order c, we can write the average retrieving error rate as

$$\bar{P}_e = \frac{1}{\lambda} \sum_{c=c_{\min}}^{c_{\max}} \left(1 - \left(1 - d^{(1-\alpha)c}\right)^{\alpha c(l-1)+(\chi-c)l}\right). \qquad (29)$$

Combined with (25) for $M_c = M/\lambda$, it is then possible to estimate the performance of the network in terms of error rate versus diversity when variable order messages are stored. Fig. 7 shows this performance in two cases: $c_{\min} = 6$ and $c_{\max} = 18$ on the one hand, $c_{\min} = 12$ and $c_{\max} = 24$ on the other hand, and for $\alpha = 0.25$ (on average when c is not a multiple of four). Both simulated and theoretical values deduced from (29) are also displayed. As in the case of c constant, performing multiple iterations does not reduce the error rate and then only one iteration is considered. The curves of Fig. 7 have to be compared with that obtained for c constant and equal to 12 [Fig. 3(a)]. Unsurprisingly, the performance is somewhat lower for variable order messages than for c constant and close to c_{opt}. For low error rates, reduction in diversity is more pronounced for small values of c. However, in all cases, provided that c remains small compared with χ, the order of magnitude for diversity remains the same.

VIII. Conclusion

We have demonstrated and assessed the ability of binary networks to store and retrieve a large number of sparse messages, of constant or variable orders. These messages are stored as cliques whose vertices belong to distinct clusters, the number of which (c) is small compared with the total number (χ). The stored messages may be retrieved in the presence of erasures and even after some kind of distortion, provided that the decoding algorithm is adapted to the particular problematic.

To speak in terms of nonlinear systems, we have shown that an appropriately organized binary recurrent graph may contain a large number of attractors. These stable and distinguishable patterns may be considered as codewords of a distributed code whose local codes are constant weight-1 codes [17] and the global code is a clique-based code. The big difference and advantage of this neural code over a classical distributed code, such as turbo [26] or low-density parity-check [27] code, is of course its ability to store independent codewords, that is, words that are not linked by some linear coding relation.

Everywhere in this paper, we have considered messages as i.i.d. to make probabilistic analysis simple. Indeed, identically distributed patterns result in a homogeneous network, that is, constant expected density. This hypothesis may not reflect real applications for which messages to store are more or less correlated (e.g., words of a lexicon having the same root). Unequal density impairs the performance of the associative memory because the most connected fanals may become less discriminating in the search for the appropriate pattern, in case of erasures or distortions. The effects of correlation have been widely studied in the specific case of Hopfield networks (see [28] for instance) and need also to be assessed for clique-based associative memories, to find efficient compensation techniques.

The construction of the proposed network was inspired by the hierarchical organization of the neocortex: microcolumns (fanals) grouped in columns (clusters), which gather in macrocolumns (networks). We have observed that the number of messages that a network composed of $\chi = 100$ clusters having $l = 64$ fanals each is able to store and retrieve correctly is around 100 000. If we extrapolate this result to what could offer the resource of the human cortex with its billion microcolumns (roughly), the quadratic law expressed by (8) leads to $10^5(10^9/6400)^2 \approx 10^{15}$ messages. This order of magnitude is certainly exaggerated as the small world organization of the neocortex [29] does not allow extending the quadratic law to the whole scale.

The concept of neural clique is familiar to neuroscientists [30], [31] but to our knowledge, the storage and retrieving properties of clique-based sparse messages had never been studied to the point of formalization we have developed in this paper. In a recent communication [32], such cluster-based networks were also demonstrated to be suited to the storage of sequences, and not only to atemporal messages. To allow this, cliques are replaced with tournaments, that

is, cliques where arrows substitute for edges. Efficiencies in the range of 20% are achievable with still good properties of robustness, tournaments being structures almost as redundant as cliques. The kind of neural networks that we have analyzed in this paper has very simple storage and retrieving rules and offers a large amount of storage capacity as well as attractive correction properties. This may be considered as a good candidate for modeling the cerebral long term memory and also an interesting starting point for the design of machines able to store a lot of messages/situations/sequences and to combine them using some cognitive principles yet to be defined. For instance, as already said in Section I, this kind of network may be considered as an interesting model for sparse coding dictionaries implementation. This work is currently undertaken within the framework of the Neural Coding Project funded by the European Research Council.

APPENDIX

Fig. 8 shows a stored clique of order six with vertices A, B, C, D, E, and F, all belonging to distinct clusters. Only vertices A, B, C, and D are known at the beginning of the retrieving process. A spurious clique of order five with vertices A, B, C, D, and X exists, due to connections established by other stored cliques (not represented).

Therefore, at the beginning, A, B, C, and D have initial values 1 whereas the others have value 0. Then unitary signals are sent through the network and (11) with memory parameter $\gamma = 1$ fixes all values to 4. Spurious vertex X cannot be eliminated using relation (12); all values are then positioned to 1. If a second iteration is carried out, the scores become $A = B = C = D = 7, E = F = 6$, and $X = 5$. Again, because E and F do not obtain the maximal score which is 7, only nodes A, B, C, and D are assigned value 1 and the clique cannot be recovered. Further iterations will repeat the same cycle.

REFERENCES

[1] R. A. Resnik, "The dynamic representation of scenes," *Vis. Cognit.*, vol. 7, pp. 17–42, Jul. 2000.

[2] W. E. Vinje and J. L. Gallant, "Sparse coding and decorrelation in primary visual cortex during natural vision," *Science*, vol. 287, pp. 1273–1276, Feb. 2000.

[3] P. Földiàk, "Sparse coding in the primate cortex," *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 2003, pp. 1064–1068.

[4] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Current Opinion Neurobiol.*, vol. 14, no. 4, pp. 481–487, Aug. 2004.

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

[6] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, Jul. 1948.

[7] V. Gripon and C. Berrou, "Sparse neural networks with large learning diversity," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1087–1096, Jul. 2011.

[8] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[9] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[10] O. A. Maillard and R. Munos, "Compressed least-squares regression," in *Advances in Neural Information Processing Systems*, Vancouver, BC, Canada: MIT Press, Dec. 2009.

[11] M. J. Wainwright, "Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting," *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5728–5741, Dec. 2009.

[12] S. Aeron, V. Saligrama, and M. Zhao, "Information theoretic bounds for compressed sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5111–5130, Oct. 2010.

[13] M. Akcakaya and V. Tarokh, "Shannon-theoretic limits on noisy compressive sampling," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 492–504, Jan. 2010.

[14] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1553–1564, Mar. 2010.

[15] M. G. Jafari and M. D. Plumbley, "Fast dictionary learning for sparse representations of speech signals," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 1025–1031, Sep. 2011.

[16] K. Jia, X. Wang, and X. Tang, "Image transformation based on learning dictionaries across image spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 367–380, Feb. 2013.

[17] V. Gripon and C. Berrou, "Nearly-optimal associative memories based on distributed constant weight codes," in *Proc. Workshop ITA*, San Diego, CA, USA, Feb. 2012, pp. 269–273.

[18] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biol.*, vol. 5, no. 4, pp. 115–133, 1943.

[19] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[20] C. A. Perfetti and D. J. Bolger, "The brain might read that way," *Sci. Stud. Read.*, vol. 8, no. 3, pp. 293–304, Jul. 2004.

[21] E. G. Jones, "Microcolumns in the cerebral cortex," *Proc. Nat. Acad. Sci.*, vol. 97, no. 10, pp. 5019–5021, May 2000.

[22] C. Johansson and A. Lansner, "Towards cortex sized artificial neural systems," *Neural Netw.*, vol. 20, pp. 48–61, Jan. 2007.

[23] L. Cruz, S. V. Buldyrev, S. Peng, D. L. Roe, B. Urbanc, H. E. Stanley, *et al.*, "A statistically based density map method for identification and quantification of regional differences in microcolumnarity in the monkey brain," *J. Neurosci. Methods*, vol. 141, no. 2, pp. 321–332, May 2005.

[24] E. Elhamifar and R. Vidal, "Robust classification using structured sparse representation," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 1873–1879.

[25] J. Grainger and C. Whitney, "Does the huamn mnid raed wrods as a wlohe?" *Trends Cognit. Sci.*, vol. 8, no. 2, pp. 58–59, 2004.

[26] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[27] R. Gallager, "Low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[28] M. Löwe, "On the storage capacity of Hopfield models with correlated patterns," *Ann. Appl. Probab.*, vol. 8, no. 4, pp. 1216–1250, 1998.

[29] O. Sporns and J. D. Zwi, "The small world of the cerebral cortex," *Neuroinformatics*, vol. 2, no. 2, pp. 145–162, 2004.

[30] L. Lin, R. Osan, and J. Z. Tsien, "Organizing principles of real-time memory encoding: Neural clique assemblies and universal neural codes," *Trends Neurosci.*, vol. 29, no. 1, pp. 48–57, Jan. 2006.

[31] L. Lin, R. Osan, S. Shoham, W. Jin, W. Zuo, and J. Z. Tsien, "Identification of network-level coding units for real-time representation of episodic experiences in the hippocampus," *Proc. Nat. Acad. Sci.*, vol. 102, no. 17, pp. 6125–6130, Apr. 2005.

[32] X. Jiang, V. Gripon, and C. Berrou, "Learning long sequences in binary neural networks," in *Proc. Cognit.*, Nice, France, Jul. 2012, pp. 165–170.

Behrooz Kamary Aliabadi (S'09) received the B.Sc. degree in telecommunications from Azad University, Tehran, Iran, and the M.Sc. degree in wireless communications from Lund University, Lund, Sweden. He is currently pursuing the Ph.D. degree with Télécom Bretagne, Institut Mines-Télécom, France.

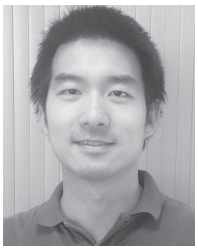
His current research interests include information theory, signal processing, and neural networks.

Claude Berrou (M'86–F'09) is a Professor with the Electronics Department, Télécom Bretagne, Institut Mines-Télécom, France. In 1990, in collaboration with Prof. A. Glavieux, he introduced the concept of probabilistic feedback into error-correcting decoders and developed a new family of quasioptimal error correcting codes, which he nicknamed turbo codes. He pioneered the extension of the turbo principle to joint detection and decoding processing, known today as turbo detection and turbo equalization. His current research interests include computational intelligence in the light of information theory.

Mr. Berrou has received several distinctions, including the IEEE Information Theory Golden Jubilee Award for Technological Innovation in 1998, the IEEE Richard W. Hamming Medal in 2003, the Grand Prix France Télécom de l'Académie des Sciences in 2003, and the Marconi Prize in 2005. He was elected as a member of the French Academy of Sciences in 2007.

Vincent Gripon (S'10–M'12) received the Ph.D. degree from Télécom Bretagne, Institut Mines-Télécom, France.

He is currently a Post-Doctoral with McGill University, Montreal, QC, Canada. He is the co-creator and organizer of a programming contest named TaupIC, which targets French top undergraduate students. His current research interests include information theory, error-correcting codes and cognitive science, and links between neural networks and distributed error correcting codes.



Xiaoran Jiang (S'11) was born in Hangzhou, China, in 1987. He received the Engineer degree in telecommunication from Télécom Bretagne, Institut Mines-Télécom, France, in 2010. He is currently pursuing the Ph.D. degree with the Electronics Department, Télécom Bretagne.

His current research interests include information theory, sparse coding, cognitive science, and, especially, sequence learning in neural networks.