2016 9th International Symposium on Turbo Codes & Iterative Information Processing

Distributed coding and synaptic pruning

Eliott Coyac, Vincent Gripon, Charlotte Langlais, and Claude Berrou

Electronics Department, Telecom Bretagne, name.lastname@telecom-bretagne.eu

***Abstract*—This paper deals with the modelization of synaptic pruning in the developing brain, at the informational level. Relying on the hypotheses of clique-based memory and Hebbian learning, we consider several scenarios to try to understand how reliable point-to-point communication may be achieved in the cortex whereas at birth, neurons are connected to a multitude of other neurons, similar to a connection of broadcast type. It is shown that quasi-perfect transfer of information can be obtained in a plausible way using simple rules if a mechanism of synaptic normalization is implemented at the receiver side.**

I. INTRODUCTION

At the informational level, the mature brain may be seen as a network composed of several hundred modules according to a small-world organization [1]. This arrangement as well as the setting-up of communication channels between the modules is the result of both neurogenesis (genetic programming) and cerebral plasticity all along the lifetime. At birth, almost all the neurons that will be used during the life are already present but the pre-established connections have no cognitive signification as learning has not really started out. The brain will acquire and memorize myriads of pieces of information thanks to the modification of these connections which will be reinforced or weakened. Recent experiments [2] have shown that synaptic strengths, which are the intensity of connections between neurons, may be spread out by a factor of 10^5 in mammalian brains.

Whereas many studies have been done about the way information is materialized and stored in the brain, so far little work has been devoted to the problem of communication between modules. Many questions arise when trying to understand this communication: the kind of encoding and modulation, the length of messages, the role of oscillations and time [3] (for instance, is spiking modelization necessary to explain information transfer or is it sufficient to consider binary signaling?), the type and level of noise, etc. [4]. In particular, if we assume that cortical modules actually exchange messages in a similar way to telecommunication networks, the transmission of a given message must be reproducible, that is, the same neurons must be involved in both sides of the channel each time it is transmitted. In the first years of childhood, the receiving neurons are not fixed and stabilized within a module. The supernumerary connections make the situation rather broadcast-like than point-to-point transmission towards this module and therefore many connections have to be discarded to ensure reliable communication.

In this paper we are interested in understanding how content stored in a local module of the brain, through assemblies of neurons, can be mapped to that of another module, despite the fact that long distance connections are initally broad. The removing or weakening of useless connections, so-called *pruning* in the neuroscience field (see [5], [6] for instance), may be analyzed with very simple hypotheses when considered at the informational level. Relying on clustered clique-based representation of information [7] and binary signaling, we show that pruning can be efficiently implemented using three fundamental rules: Winner-Takes-All, Hebbian rule and normalization. As already shown in [8] with a more complex model, synaptic normalization is essential to the efficacy of pruning.

We first present the neural clique network model in section II, which we consider to be the local memory model in each module. Then we attack the problem in itself in section III: how to structure connections between two such modules in order to be able to transfer information with maximum efficiency (later defined), while using simple rules to be biologically plausible? Finally we offer a conclusion which sums up how reliable transmission of data between two cortical areas or modules can be achieved, thanks to synaptic pruning.

II. NEURAL CLIQUE NETWORKS

Associative memories are devices in which a stored piece of information can be addressed from part of it, as opposed to classical index-based memories in which an explicit address is required. Because of their ability to associate contents, they are considered a realistic model for brain memory. The most prominent model of associative memories was introduced by John Hopfield in 1982 [9], despite the fact that other models [10] are known to provide better efficiencies [11].

Recently, a novel family of associative memories has been proposed [12], [13], termed neural clique networks. This model improves existing ones by making use of error correcting techniques inspired by the functioning of the neocortex.

Formally, consider a finite alphabet $\mathcal{A}$. We call *message* of length c over $\mathcal{A}$ a vector $\mathbf{m} \in \mathcal{A}^c$. We denote by m_i the i-th symbol of $\mathbf{m}$. Neural clique networks are able to store messages then retrieve them from part of messages with high probability as long as there are not too many stored in the network. The retrieval process can be seen as a decoding process.

The idea is to use a neural network made of c parts (or clusters) containing $\ell = |\mathcal{A}|$[1], units each. Therefore it is possible to index clusters from 1 to c and units in each cluster

This work was funded in part by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement n° 290901, and by Future & Rupture of the Fondation Télécom.

---

[1] The cardinal of $\mathcal{A}$.

978-1-5090-3401-7/16/.00 ©2016 IEEE

from 1 to ℓ such that each unit is uniquely determined by a pair (i, j) where $1 \leq i \leq c$ and $1 \leq j \leq \ell$. We call *representation* of a message $\mathbf{m}$ in the neural network the set of units $\{(i, m_i), 1 \leq i \leq c\}$. In order to *store* a message, all pairs of units in its representation are connected, embodying a clique in the neural network. An already existing connection in the network is not strengthened by the accumulation of messages: connections are binary. Figure 1 depicts the storage procedure in a toy network made of 4 clusters containing 16 units each.

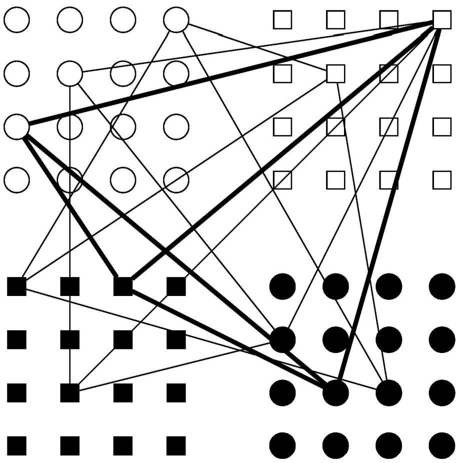

Figure 1. Storing procedure illustration. The pattern to store (with thick edges) connects units from 4 clusters of 16 units each (filled circles, filled rectangles, rectangles and circles).

Once multiple messages have been stored, the authors propose an iterative procedure that aims at retrieving them from part of their content. Formally, let us consider a stored message $\mathbf{m}$, from which part of its symbols are missing. The units in the representation of $\mathbf{m}$ that correspond to the non-missing symbols are *activated*. The iterative procedure iterates two steps [14]:

1) Computing a score for each unit in the network. Typically the score of a unit is the number of activated units it is connected to.
2) Selecting which units to activate for the next step depending on their score. Typically selected units are those that achieve the maximum score in their cluster.

Inhibiting neurons other than the highest scoring one amongst a group of neurons is also referred to as Winner-Take-All (WTA) [15]. This process iterates until a stopping criterion is reached (in practice a few iterations are often enough) [13].

As an example, a network storing messages of 8 bytes would be made of 8 clusters of 256 units each, for a total of 2048 units. To each byte value and position in the message would correspond a single unit. Storing a message of 8 bytes would create complete interconnections between the 8 corresponding units in the network. Then it is possible to test if a message is stored by checking if the corresponding units are fully interconnected, or to recover a message from part of its bytes using the decoding algorithm.

It is possible to estimate the error probability in successfully retrieving previously stored messages when those are uniformly and independently distributed. First, the probability that a connection between units (i, j) and (i', j') $(i \neq i')$ exists, called density, is [13]:

$$d = 1 - \left(1 - \ell^{-2}\right)^{M} ,$$

where M is the number of stored messages. Then, considering connections to be independent (proved bounds have been introduced in [16]), what the authors argue to be a reasonable assumption as supported by simulations, this density leads to the error probability of retrieving a message when c_e of its symbols are missing [13]:

$$P_e = 1 - \left(1 - \left[1 - \left(1 - \ell^{-2}\right)^{M}\right]^{c-c_e}\right)^{(\ell-1)c_e} .$$

From this equation, it is seen that for a given c_e and P_e, with c and ℓ tending to infinity, the number M of messages stored grows quadratically with ℓ if ℓ is small compared to d^{-c}.

Neural clique networks are known to perform better than other existing models [16]. Noticeable improvements have been proposed in order to allow storage of sparse messages [17] and temporal sequences [18].

Biologically, each *unit* of a neural clique network is considered to be a *minicolumn* [17] which is a group of dozens of neurons with a few axons going outside the network [19]–[21]. As such, in the following of the paper we refer to the *units* of the neural networks as neurons. In the neural clique network model, they correspond to one of the few neurons of the minicolumn with connections outgoing from the network or incoming from outside the network.

III. SYNAPTIC PRUNING

In this section we offer a biologically plausible way to achieve one-to-one communication between neurons of two neural networks, in order to be able to transmit data efficiently. There are several ideas involved including Hebbian Learning, Winner-Take-All and synaptic normalization, which are gradually introduced in order to show their influence. During the process, the connection weights between the two networks naturally become binary from their initial floating value.

We study this process in the context of neural clique networks, but it is similarly applicable to any two neural networks, or layers of neural networks, as long as the Winner-Take-All operation on the target network is available during the process. The motivation is to be able to transmit information as-is from a local module of a brain area to that of another, and enable communication between those two modules. This is helpful in architectures composed of multiple neural networks, as each group of networks can communicate efficiently with others and separate long-distance communication from local computation.

A. Communication Model

Consider two modules $\mathcal{A}$ and $\mathcal{B}$, both individually behaving as neural clique networks. They are comprised of c *clusters* containing ℓ *neurons* each, behaving in the same way as individual clusters from neural clique networks in the sense that they provide a *Winner-Take-All* mechanism, referenced later in this paper as *WTA*. WTA allows only *one* neuron per cluster to be active at any given time, the neuron that is stimulated the most.

Let us denote $c_a, a \in \{1, ..., c\}$ the clusters of $\mathcal{A}$ and $c'_b, b \in \{1, ..., c\}$ the clusters of $\mathcal{B}$.

A neuron $u_{a,i}, i \in \{1, ..., \ell\}$ of c_a is originally connected to multiple neurons of c'_b, the corresponding cluster in $\mathcal{B}$. It isn't connected to other clusters of $\mathcal{B}$, as the biological counterpart's *axon* has a spreading limit. The connections have random weights between 0 and 1 following a truncated Gaussian distribution centered in 0.5.

When considering a single pair of corresponding clusters of $\mathcal{A}$ and $\mathcal{B}$, the neurons from those clusters are denoted $u_i, i \in \{1, ..., \ell\}$ and $v_j, j \in \{1, ..., \ell\}$. The connection weight between those neurons is denoted $w_{i,j}$.

Our goal is, from those initial conditions, to be able to transfer cliques from $\mathcal{A}$ to $\mathcal{B}$ with minimal loss in performance. We will measure performance by seeing whether cliques transferred from $\mathcal{A}$ to $\mathcal{B}$ offer the same erasure-recovering properties, and to which extent.

In order to achieve our goal, we want to transform the initially varied, broadcast connections between clusters into point-to-point connections where one neuron of a cluster of $\mathcal{A}$ is only connected to one neuron of the corresponding cluster of $\mathcal{B}$. Then, data can be transferred or copied from $\mathcal{A}$ to $\mathcal{B}$ more efficiently.

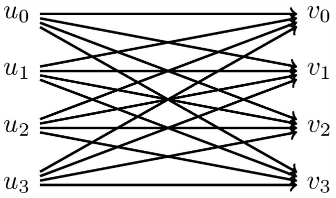

Figure 2. Original state: multiple point to point connections. The figure only shows 4 neurons of a pair of corresponding clusters c_1 and c'_1. The connections have each different random weights between 0 and 1.

B. Same size clusters

We use the principle of Hebbian Learning [22] by which connections that are used are reinforced and connections that prove not useful have their weight decreased. We also take a pair of clusters, c_1 and c'_1 as examples, but we can generalize the same operations to all clusters.

We activate a neuron u_i from c_1. All neurons from c'_1 are stimulated depending on the weight of their connection to u_i. Let's call v_j the neuron from c'_1 that has the strongest connection to u_i. Because we apply WTA to c'_1, only v_j is activated while the other neurons of c'_1 stay silent. Thus, because of Hebbian Learning the connection between u_i and v_j is strengthened while the connections between u_i and the other neurons of c'_1 are weakened. By repeating this a number of times, u_i is only connected to v_j, and by extending this to other neurons of c_1 each neuron from c_1 is associated with one neuron from c'_1.

However, as seen in Figure 3, a neuron of c'_1 can be associated with multiple neurons of c_1, and some neurons of c'_1 are associated with no neurons of c_1 at all. The probability for a neuron of c'_1 to *not* be associated with any neuron of c_1 is $\left(1 - \frac{1}{\ell}\right)^{\ell}$, and as such only approximately $\ell\left(1 - \left(1 - \frac{1}{\ell}\right)^{\ell}\right)$ neurons of c'_1 are connected to neurons of c_1.

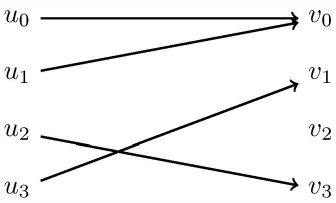

Figure 3. Example of non-optimal association between c_1 and c'_1, where multiple neurons of c'_1 may be associated with one neuron in c_1.

For example with $\ell = 256$, there are only 63% of neurons used in c'_1, while the other neurons have no functionality whatsoever.

As the capacity of neural clique networks is quadratic compared to the number of neurons in a cluster, there is a great loss of capacity in $\mathcal{B}$ if it only learns data from clusters of $\mathcal{A}$. As such in a network with 256 neurons per cluster, there is a capacity loss of more than 60% for $\mathcal{B}$ when learning cliques from $\mathcal{A}$[2]. Moreover, in addition to there being less neurons available, there is also correlation introduced by neurons from c'_1 sharing a neuron in c_1.

C. Larger target clusters

We try another model where the clusters from $\mathcal{B}$ are originally of size $r \cdot \ell$ with $r \in \mathbb{R}, r \geq 1$, as opposed to being the same size as clusters from $\mathcal{A}$.

The probability for a neuron of c'_1 to not be associated with any neuron of c_1 is $\left(1 - \frac{1}{r\ell}\right)^{\ell}$. In Figure 4 the average number of neurons in c'_1 connected to at least one neuron in c_1 for $\ell = 256$ is shown as a function of r. This figure shows that relatively small values of r lead to one-to-one mappings for a large proportion of input neurons, but obtaining it for all neurons is not achieved even for large values of r. This fact motivates for improved strategies in the following subsections.

After the association is done, we can remove or use the non-connected neurons from c'_1 for something else, thus alleviating the cost of having more neurons. This is similar to biology where neurons that are not used are pruned or reallocated. That way, although the clusters from $\mathcal{B}$ initially contain more material than those of $\mathcal{A}$, in the end approximately the same amount of material (neurons) is used in both modules.

D. Cliques performance

In the previous subsection we focused on measuring how close to one-to-one the obtained mappings were. Regarding our motivation, it is more interesting to look at the impact of this process on information retrieval. Specifically, we stress

---

[2] As $0.63^2 = 0.3969$, the maximum capacity of $\mathcal{B}$ compared to $\mathcal{A}$ is a bit less than 40%.

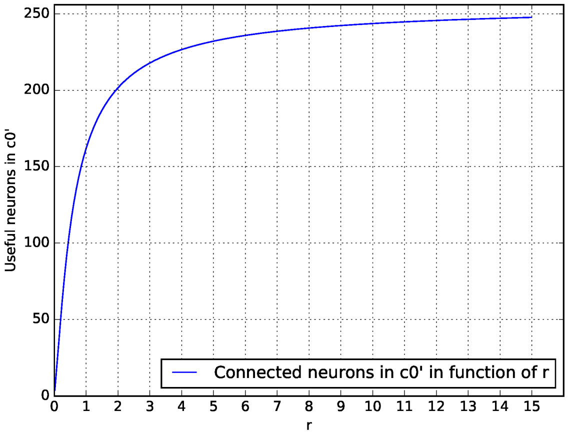

Figure 4. For $\ell = 256$, the average final number of neurons of c_1' connected to c_1 depending on the initial ratio r of the size of c_1' compared to that of c_1.

the probability to retrieve a clique in $\mathcal{B}$, considering these cliques have been transferred from $\mathcal{A}$ through our proposed lossy communication scheme.

We first proceed with Hebbian learning to structure the connections between $\mathcal{A}$ and $\mathcal{B}$. We then transfer cliques onto $\mathcal{B}$ using the point-to-point connections between the clusters of both modules.

Specifically, for each clique, we activate it in $\mathcal{A}$, which will then activate the corresponding neurons in $\mathcal{B}$ depending on the connections between $\mathcal{A}$ and $\mathcal{B}$. We maintain that state in $\mathcal{A}$ for several iterations. Plasticity, and more specifically Hebbian learning, creates the local connections in $\mathcal{B}$, thus providing it with the same error-correcting properties as $\mathcal{A}$.

We then perform a test: we only activate part of the neurons of a previously learned clique in $\mathcal{B}$, and see if the clique can be recovered in the network. Multiple values of r are considered. The test is also performed in $\mathcal{A}$ as reference.

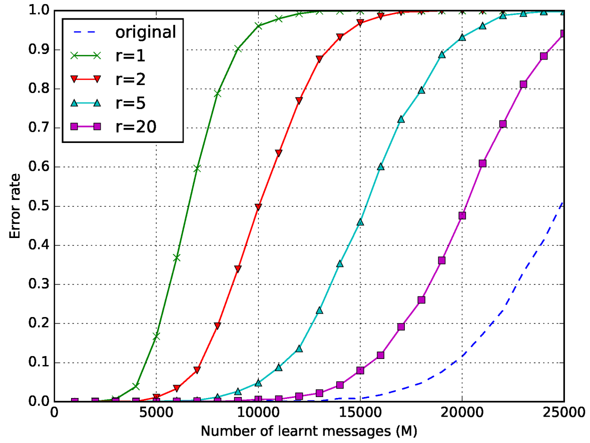

Figure 5. For $\ell = 256$ and $c = 8$, the error rate in retrieving half-erased messages from cliques learned from $\mathcal{A}$ in $\mathcal{B}$. *original* shows the error rate in $\mathcal{A}$.

From Figure 5, it is seen that to obtain acceptable performance in $\mathcal{B}$ compared to $\mathcal{A}$ at least 20 times the original material is needed, which is a lot to ask for. Despite that, there is still a significant loss of performance. We can surmise that there is a more *guided* behavior than mere random Hebbian Learning where biology is concerned. It is worth noting that the rate of Hebbian learning has no influence on the results in the present conditions, as for any neuron of $\mathcal{A}$, the neuron of $\mathcal{B}$ with the strongest connection to it will never change.

E. Limited spreading

We introduce a modification to our model: instead of initially having each neuron from one cluster connected to each neuron from the corresponding cluster, we introduce a parameter $s \in \mathbb{N}^*$ which is the number of neurons of $\mathcal{B}$ a neuron from $\mathcal{A}$ is initially connected to.

Limiting the number of connections to a reasonable initial value reduces the material used and the complexity of the problem. As such, the initial condition is that a neuron of $\mathcal{A}$ is connected to s random neurons of $\mathcal{B}$ with weights following a (truncated) Gaussian distribution between 0 and 1, centered in 0.5. Results previously obtained, including Figure 5, do not change by introducing this parameter.

F. Normalization

We introduce normalization [8] as well as a learning rate ε, which we now use for both the Hebbian learning process and the normalization process.

The Hebbian learning process is as such: Let u_i be an active neuron from $\mathcal{A}$ and v_j the corresponding neuron in $\mathcal{B}$ with the strongest connection to u_i. Then, $w_{i,j} = w_{i,j} + \varepsilon(1 - w_{i,j})$ and $\forall k \neq j, w_{i,k} = w_{i,k}(1 - \varepsilon)$.

The normalization process is symmetrical to the Hebbian learning process: $\forall k \neq i, w_{k,i} = w_{k,i}(1 - \varepsilon)$. It can be described biologically as the receiving neuron favouring one synapse in particular, the one just activated, by reducing the weights of the other synapses according to the learning rate ε.

A rate of $\varepsilon = 1$ corresponds to a fast, or instant selection of the strongest connection, discarding the other connections from both sides and creating a bilateral, unique point-to-point connection.

As the normalization process is gradual and neurons from a same cluster in $\mathcal{A}$ can now affect each other's connections as they are connected to common neurons in $\mathcal{B}$, we do the normalization and Hebbian learning process by activating cliques from $\mathcal{A}$ one after the other. When the connections are stabilized, which doesn't need multiple iterations over the whole number of cliques, we then transfer the cliques from $\mathcal{A}$ to $\mathcal{B}$.

Table I shows the impact of different learning rates when $r = 1$ for a standard neural clique network, the results have been obtained by performing 40000 tests for each value of the learning rate ε. The number of useful neurons is the average number of neurons in $\mathcal{B}$ per cluster that participate in learning cliques from $\mathcal{A}$. We see that the learning rate has a decisive impact on the performance, with higher learning rates offering much better performance.

ε	Useful neurons in $\mathcal{B}$ (per cluster)	Error rate for 15000 cliques
0.1	173.28	0.999
0.2	202.63	0.916
0.3	222.59	0.552
0.5	244.57	0.070
0.99	251.68	0.024

Table I

IMPACT OF THE LEARNING RATE ε ON PERFORMANCE FOR NETWORKS OF 8 CLUSTERS OF 256 NEURONS EACH, WITH $s = 20$ AND $r = 1$.

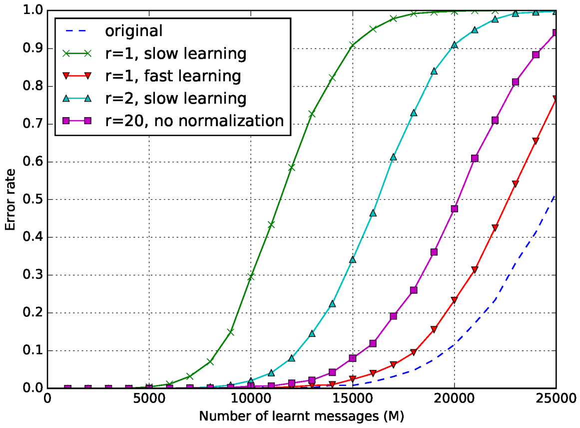

Figure 6. For $\ell = 256$ and $c = 8$, shows the error rate in retrieving half-erased messages from cliques learned from $\mathcal{A}$ in $\mathcal{B}$. *original* shows the error rate in $\mathcal{A}$. Slow learning corresponds to $\varepsilon = 0.2$. Fast learning corresponds to $\varepsilon = 0.99$. Each neuron of $\mathcal{A}$ is initially connected to $s = 20$ random chosen neurons of $\mathcal{B}$ from the corresponding cluster.

Simulations with normalization and connections initially reduced in range are performed and shown in Figure 6, where $\mathcal{A}$ is a neural clique network with 8 clusters of 256 neurons each. As seen, normalization achieves much better results than simple Hebbian learning which is shown in Figure 5. With slow normalization and $r = 5$ or fast normalization and $r = 2$, the goal is achieved as the performance is the same as in the original network, which is denoted by the dashed curve.

Limited spreading helps to reduce complexity and be more plausible biologically but the parameter $s = 20$ in fact reduces performance, slightly better performance would be obtained with initially fully interconnected neurons ($s = r\ell$).

IV. CONCLUSION

We have studied how reliable communication can be established in the brain between two cortical areas, and shown how two neural networks can stabilize point-to-point connections from one network to another. Non-trivial remote neuron pairing is attained by the use of simple local rules such as Hebbian learning [22], normalization [8] and clusterized *Winner-Take-All*. They are used to transform initially broad connections to focused, point-to-point connections. Those three rules achieve reliable synaptic pruning, and with an initially slightly more dense destination cortical area which can later be pruned of its useless neurons, perfect transmission of data can be achieved between two similar neural networks. The impact of using different normalization processes such as Sinkhorn's [23] on performance can be studied in a future work.

REFERENCES

[1] D. Meunier, R. Lambiotte, and E. T. Bullmore, "Modular and hierarchically modular organization of brain networks," *Frontiers in neuroscience*, vol. 4, p. 200, 2010.

[2] N. Markov, P. Misery, A. Falchier, C. Lamy, J. Vezoli, R. Quilodran, M. Gariel, P. Giroud, M. Ercsey-Ravasz, L. Pilaz *et al.*, "Weight consistency specifies regularities of macaque cortical networks," *Cerebral Cortex*, vol. 21, no. 6, pp. 1254–1272, 2011.

[3] A. J. Watrous, J. Fell, A. D. Ekstrom, and N. Axmacher, "More than spikes: common oscillatory mechanisms for content specific neural representations during perception and memory," *Current opinion in neurobiology*, vol. 31, pp. 33–39, 2015.

[4] C. Berrou, O. Dufor, V. Gripon, and X. Jiang, "Information, noise, coding, modulation: What about the brain?" in *Turbo Codes and Iterative Information Processing (ISTC), 2014 8th International Symposium on*. IEEE, 2014, pp. 167–172.

[5] G. Chechik, I. Meilijson, and E. Ruppin, "Neuronal regulation: A mechanism for synaptic pruning during brain maturation," *Neural Computation*, vol. 11, no. 8, pp. 2061–2080, 1999.

[6] J. Iglesias, J. Eriksson, F. Grize, M. Tomassini, and A. E. Villa, "Dynamics of pruning in simulated large-scale spiking neural networks," *Biosystems*, vol. 79, no. 1, pp. 11–20, 2005.

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

[8] T. Elliott and N. R. Shadbolt, "Multiplicative synaptic normalization and a nonlinear hebb rule underlie a neurotrophic model of competitive synaptic plasticity," *Neural computation*, vol. 14, no. 6, pp. 1311–1322, 2002.

[9] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[10] D. J. Willshaw, O. P. Buneman, and H. C. Longuet-Higgins, "Non-holographic associative memory." *Nature*, 1969.

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

[12] V. Gripon and C. Berrou, "A simple and efficient way to store many messages using neural cliques," in *Proceedings of IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain*, Paris, France, April 2011, pp. 54–58.

[13] ——, "Sparse neural networks with large learning diversity," *IEEE Transactions on Neural Networks*, vol. 22, no. 7, pp. 1087–1096, July 2011.

[14] A. Aboudib, V. Gripon, and X. Jiang, "A study of retrieval algorithms of sparse messages in networks of neural cliques," in *Proceedings of Cognitive 2014*, May 2014, to appear.

[15] S. Grossberg, "Contour enhancement, short term memory, and constancies in reverberating neural networks," in *Studies of mind and brain*. Springer, 1982, pp. 332–378.

[16] V. Gripon, J. Heusel, M. Löwe, and F. Vermet, "A comparative study of sparse associative memories," *arXiv preprint arXiv:1512.08892*, 2015.

[17] B. K. Aliabadi, C. Berrou, V. Gripon, and X. Jiang, "Storing sparse messages in networks of neural cliques," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, pp. 980–989, 2014.

[18] X. Jiang, V. Gripon, and C. Berrou, "Learning long sequences in binary neural networks," in *Proceedings of Cognitive 2012*, Nice, France, July 2012, pp. 165–170.

[19] E. G. Jones, "Microcolumns in the cerebral cortex," *Proceedings of the National Academy of Sciences*, vol. 97, no. 10, pp. 5019–5021, 2000.

[20] C. Johansson and A. Lansner, "Towards cortex sized artificial neural systems," *Neural Networks*, vol. 20, no. 1, pp. 48–61, 2007.

[21] L. Cruz, S. V. Buldyrev, S. Peng, D. L. Roe, B. Urbanc, H. Stanley, and D. L. Rosene, "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, no. 2, pp. 321–332, 2005.

[22] D. O. Hebb, *The organization of behavior: A neuropsychological approach*. John Wiley & Sons, 1949.

[23] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The annals of mathematical statistics*, vol. 35, no. 2, pp. 876–879, 1964.