Using Tags to Improve Diversity of Sparse Associative Memories

Stephen Larroque, Ehsan Sedgh Gooya, Vincent Gripon and Dominique Pastor

Electronics Department
Télécom-Bretagne (Institut Mines-Télécom)
Email: firstname.lastname@telecom-bretagne.eu

***Abstract*—Associative memories, a classical model for brain long-term memory, face interferences between old and new memories. Usually, the only remedy is to enlarge the network so as to retain more memories without collisions: this is the network's size–diversity trade-off. We propose a novel way of representing data in these networks to provide another mean to extend diversity without resizing the network. We show from our analysis and simulations that this method is a viable alternative, which can perfectly fit cases where network's size is constrained, such as neuromorphic FPGA boards implementing associative memories.**

***Keywords*—neural coding; associative memory; neural network; information theory; graph theory; sparse coding; clique; computational neuroscience.**

I. Introduction

Studying the inner workings of brain memory has increasingly become a major challenge for modern neuroscience, since memory is likely a fundamental building block for higher cognitive functions, such as language, reasoning, creativity and consciousness [1].

Associative memories are a branch of now classical computational models for brain memory. Contrary to the *von Neumann* computing architecture [2], [3], where memory is indexed by attributing a unique address for each data, an associative memory change the representation of data in a way that allows to recover an entry only using an incomplete or noisy portion of that data. Furthermore, these models emphasize greater biological plausibility by satisfying the metabolic constraints the organic brain has to face [4], [5].

However, since transformed data can overlap in associative memories, they suffer from interference: there is a tradeoff between network's size and data diversity (number of different entries possibly stored) [6].

We propose a novel way of representing data in these networks by adding a pairing meta-information among edges, thus relaxing the above mentioned tradeoff by providing another way to extend the network's data diversity.

For this purpose, we will first introduce briefly a classical model in Section 2, then in Section 3 we will extend this model with the pairing strategy. The specific dynamics of this extended model will then be analyzed in Section 4 and simulated in Section 5. Finally, an opening to biological hypotheses will be offered to the reader in Section 6 and this work will be concluded alongside a description of a few future avenues in Section 7.

II. Classical model

We will extend the *clique neural network*, a neural network based auto-associative memory introduced by Gripon et al. [7]. Since this is an associative memory, messages are stored such that it is possible to retrieve them from noisy or partially erased input.

Formally, we call *message* a finite sequence of characters of length χ over the alphabet $[\ell]$ where $[\ell]$ denotes the set of integers between 0 and ℓ, with 0 being a special symbol representing emptiness (this is a *non-value*), and c the number of significant, nonzero symbols in a message. This empty 0 symbol allows to construct sparse messages, because this character has no explicit representation in the network.

Consider a set $\mathcal{M}$ of sparse messages. To store them, Aliabadi et al. [8] propose to use a neural network with χ parts each containing ℓ units. They index each part from 1 to χ to correspond to each character of a message, and in each part they index each unit from 1 to ℓ to correspond to the possible values for that character. By notation abuse, we will make no distinction between a unit and its associated pair of indices. Then, they define a mapping associating any sparse message $m = (m_1, m_2, \ldots, m_\chi)$ with a subset of units in the network:

$$\mu = f(m) = \{(i, m_i), 1 \leq i \leq \chi, m_i \neq 0\}\ . \tag{1}$$

Rather than storing a message m, Aliabadi et al. [8] propose to store μ. To do so, they connect together all units in μ, embodying a clique into the neural network and effectively learning in one-shot. This process is depicted in Figure 1. This implies that the network is binary: an edge exists or it doesn't.

Storing a set of messages $\mathcal{M}$ in the network is simply the union of every messages' cliques. Because cliques can be overlapping (by sharing at least two units), this representation of information is lossy [8].

The process of retrieving a previously stored message, also called *decoding*, given a partial and/or erroneous input then consists in iterating two steps [9], as shown in Figure 2.

Different ways of operating these two steps (called *retrieval rules*) have been extensively studied by Aboudib et al. [9]. In this work, we choose to define the score of a unit to be the number of activated units it is connected to, called the *Sum-Of-Sum rule* [8]. We then select the units that reach the

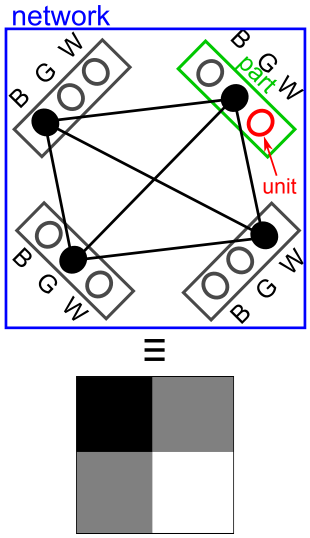

Figure 1. Process of storing an image composed of the pixels sequence {B, G, G, W} (from left to right and from top to down) in the clique network. In this scenario, the parameters of the network are $\chi = c = 4$ and $\ell = 3$ (pixel intensities range of 3 values: B for black, G for gray and W for white).

For each input query:

1) Activate units corresponding to the input query
2) Until stop criterion (number of iterations or convergence criterion):
   a) Propagation: Compute a score s for each unit in the network. This score represents the unnormalized likelihood that a unit is part of the target message.
   b) Filtering: Use a selection operator to choose whether to activate units or not based on their score.

Figure 2. Clique network iterative retrieval (decoding) process

maximum score in the network: this is the *Global Winner-Take-All rule* [10].

III. Proposed Model

Although the clique network is binary, brain synapses do not function in such a fashion: they emit an action potential of variable intensity. Illustrious models for associative memories [11]–[14], as well as most other non-associative neural networks, take account of this variable intensity by affecting a weight on the edges. However, this results in poor performance in terms of memory efficiency, constrained by a sub-linear law [6], [15].

Contrariwise to this approach, we propose to assign a color, or *tag*, to each edge instead of a weight, with the goal of pairing together the edges from the same clique. Indeed, this edge meta-information now represents a pairing cue instead of a synaptic potential intensity modulation. Thus, this meta-information does not affect information processing, but only helps in disambiguating. The tags can be seen as a modified Hebbian rule: *Neurons that fire together, wire together*, and with a strong affinity. In this sense, the tags can be related to the neurobiological mechanism called synaptic discrete states [16]–[18].

More specifically, let us now suppose that connections in the network are not binary but can take up to g distinct, discrete, values. This results in a colored graph, where each connection has its own color. We modify the storage process as follows:

1) First we associate each message to store with a tag,
2) When storing the clique equivalent of a message, we assign the corresponding tag to the clique's edges, replacing any previous tag if the edge already exists.

As a result, a recently stored message, which now corresponds to a clique with a **unique** tag in the network, can overwrite parts of an older message they share by changing the tag of those shared connections.

Another, more visual, formulation using colored graph theory is that tags can be seen as different overlays or colors of the same network, each defining a sub-graph containing edges of only one color. By focusing on one color, it's easy to decode the clique without ambiguity, which is not possible with the original clique network. Thus, those colored layers can be separated easily: newer messages can be retrieved flawlessly, while older messages can still be retrieved without ambiguity, as illustrated by Figure 3.

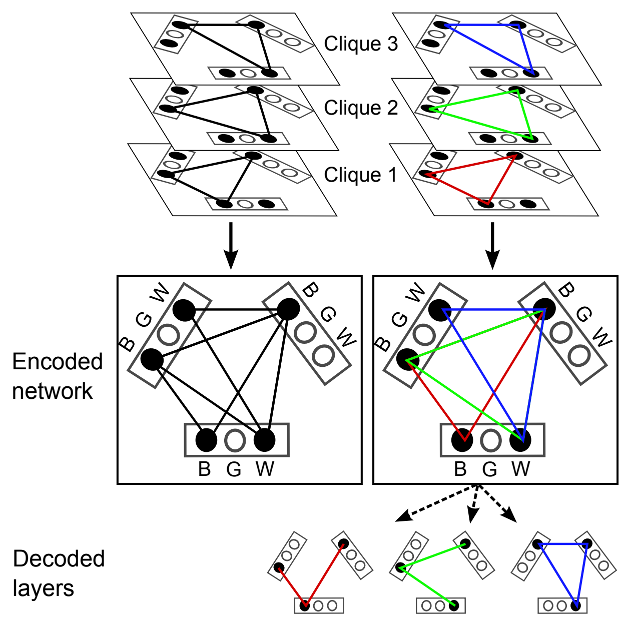

Figure 3. Comparison of the clique network (left) and tagged network (right), with tags represented as colored layers.

More formally, let us consider that messages to store are assigned a tag k from 1 to g. The storing process can be defined as constructing the adjacency matrix A of the colored graph, but instead of assigning 0 or 1 to assert an edge existence, we assign the latest, highest tag k attached to each edge:

$$A_{(ij)(i'j')} = \{max(k) \big| \exists m \in \mathcal{M}, \quad tag(m) = k \wedge m_i = j \wedge m_{i'} = j'\} , \quad \text{for } 1 \leq i,\, i' \leq \chi \text{ and } 1 \leq j,\, j' \leq \ell. \tag{2}$$

Note that k is to be defined by an assignment function, which purpose is to generate a tag for each message. We will later in this section discuss about several possible strategies.

We thus obtain an adjacency matrix where entries are valued from 0 (edge nonexistence) to g (existence + tag membership).

In order to benefit from this added material, the iterative retrieval process is adapted, as shown in Figure 4.

1) Decode just like before: propagate using Sum-Of-Sum rule and filter using Global Winners-Take-All rule.
2) Disambiguation post-processing step:
   For each message:
   a) Find major tag (= compute mode) among edges.
   b) Delete every edges possessing a different tag than the major.
   c) Delete isolated units.

Figure 4. Global-Vote-Local-Elimination retrieval rule

This results in a collaborative decision between units, which will favor likely tagged edges, and remove units that don't share at least one edge of the correct, major tag.

This combination of a local elimination based on a global, cooperative decision is the most successful strategy, which we call the *Global-Vote-Local-Elimination rule*. We also tried several other variants like a local vote (compute mode per each node) and global elimination (filter out all nodes which local major tag isn't the global major tag) but they all produced significantly lower performance. The voting strategy to find the most likely tag is probably optimal, since we simulated a tagged network with tag guiding (the tag for each clique is known, hence there's no uncertainty), and it provided no difference in performances.

Of course, assigning a tag per clique is optimal, since the tag is then unique. However, $\log_2(g)$ more resources than the clique network are needed to store the tags information. Hence, it's possible to tradeoff with the number of tags g compared to the total number M of messages in the set $\mathcal{M}$:

- $g = 1$ will output the same result as the non-tagged clique network, since all edges will have the same tag, the tags disambiguation step will just have no effect.
- $1 < g < M$ defines a limited set of tags to use among all cliques. This produces a trade-off between network's capacity and the amount of resources required to represent the tags. In practice, since there is a limited set of tags available, they will be recycled among the cliques, thus rendering tags non-unique and producing more ambiguities. Any surjective function can be used to map the tags onto the cliques. In practice, it seems reasonable to use a uniform distribution, which will distribute randomly the tags almost uniformly among messages, and also allow for *online learning* (learning new messages over time).
- $g = M$ will assign one unique tag per clique. In this case, we get as many tags as there are cliques. Performance is then optimal, but more resources are consumed. In such case we consider that the set of messages to store are ordered from lowest to highest tag, such that a message with a high tag number is said to have been stored recently and a message with a low tag is said to be old.

IV. Analysis

A. Density

Since the network still relies on cliques and edges existence to store and retrieve information, the theoretical density d – defined as the ratio of used edges to that of possible ones– is just the same as in the classical clique network [8]:

$$d = 1 - \left(1 - \frac{c(c-1)}{\chi(\chi-1)\ell^2}\right)^M . \tag{3}$$

B. Efficiency

The network is split into χ parts with ℓ units in each. Thus, the network possess $n = \chi\,\ell$ total units and $\frac{\chi(\chi-1)\ell^2}{2}$ total possible edges. Furthermore, edges aren't binary anymore, but store their tag, thus an edge can now store a value between 0 and g, and therefore the tagged network representation amounts to a binary resource Q of:

$$Q = \frac{\chi(\chi-1)\ell^2}{2}\,\log_2(g+1) . \tag{4}$$

The entropy per message b and total entropy B, or amount of binary information B learned by the network after storing all message M, does not change from the classical model [8]:

$$b = \log_2\binom{\chi}{c} + c\,\log_2(\ell) . \tag{5}$$

$$B = bM = M\left(\log_2\binom{\chi}{c} + c\,\log_2(\ell)\right) . \tag{6}$$

We can then easily derive the network's efficiency η, that is the efficient usage of available network's resources:

$$\eta = \frac{B}{Q} = \frac{2M\left(\log_2\binom{\chi}{c} + c\,\log_2(\ell)\right)}{\chi(\chi-1)\,\ell^2 \cdot \log_2(g+1)} . \tag{7}$$

which leads to the network's *efficiency-1 diversity*, an upper bound of the optimal number of M messages to store to maximize the network's efficiency:

$$M_{max} = \frac{Q}{b} = \frac{\chi(\chi-1)\ell^2 \cdot \log_2(g+1)}{2\left(\log_2\binom{\chi}{c} + c\,\log_2(\ell)\right)} . \tag{8}$$

C. Error rate

Since we are doing an associative task, that is we are trying to recover a full corrected message from a query, a *retrieval error* is defined as the network converging to a different, *spurious clique* than the correct clique from which the (partial, noisy or complete) input query was generated from.

Let us now suppose that we set $g = M$. As described in the previous section, at the disambiguation step of the decoding process, only nodes without any edge of the major tag will be filtered out. This implies that even if an old edge from an old clique can get its tag overwritten by a new clique, the two units, which the edge is linked to, are still retrievable without a hitch as long as they each possess at least one other edge with the proper tag. This means that for a clique to be irretrievable anymore, it has to lose at least one unit, and to lose one unit is for this unit to be shared by so many other, new cliques that *all edges of this unit got overwritten*.

This kind of error, very specific to the tagged network, is what we call the *lost unit error* P_{lost}, and is the most significant factor contributing to retrieval error. It is also quite interesting for the fact that it's only defined by the learning process (new cliques overwriting tags of old cliques' edges), without any influence by the decoding dynamics.

Since the network store cliques of size (number of edges) $\frac{c(c-1)}{2}$, and if we consider that cliques' edges are generated randomly uniformly among all possible network's edges, then the probability to overwrite one edge of an old unit when storing a new clique is $\frac{2}{\chi(\chi-1)\ell^2}$, and we can define P_{lost} as follows:

$$P_{lost} = \left(1 - \left(1 - \frac{2}{\chi(\chi-1)\ell^2}\right)^{(M-1)\frac{c(c-1)}{2}}\right)^c . \quad (9)$$

This formula can be understood as the probability that an edge gets overwritten by any edge from every learnt messages, and then this probability is powered to c because for a unit to be lost, every one of its edges belonging to the original clique must be overwritten. This formula is but an approximation of the lost unit error because of our assumption that messages are *i.i.d.*, which is of course not the case.

Beside the lost unit error, other factors may contribute to a wrong retrieval, such as when the vote to compute the major tag leads to a wrong tag (*tag vote error*), or when the propagation/filtering steps lead to a wrong clique (leading to a spurious clique, just like with the classical clique network) before the tag disambiguation step (*decoding error*). Yet, after simulating the tagged network's dynamics with erasures, we have found that the lost unit error is prevalent, and is a very good approximation of the overall error, as shown in Figure 5. However, this is only true when $g = M$, if we use only a limited definite number of tags lower than M, other types of error will have increasingly more effect.

V. SIMULATIONS

We analyze the network's performance with an erasure scenario, which is the substitution of one or several nonzero symbols in a message m by 0. The simulations were done by learning uniformly random messages and then each point was generated by sampling a subset of the learnt messages and erase half of the nonzero symbols. To avoid random fluctuations, each point has been averaged over 10k trials (200 messages per 50 different networks).

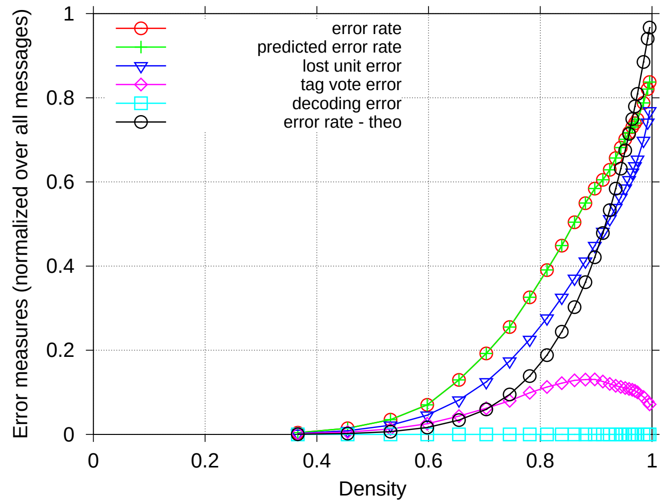

Figure 5. Impact of various types of errors on message retrieval in a sparse tagged network. The errors are computed as the ratio of total messages suffering from this particular error type over all learnt messages. Predicted error rate is the sum of all error types to check that they are good predictors of the real error rate. As can be seen, the lost clique error is a very close predictor of the real error rate, with a small difference due to the impact of the tag vote error.

To study the influence of tags on the error rate, we simulated a sparse tagged network with various numbers of tags, against a classical sparse clique network with similar parameters as described by Aliabadi et al. [8]. As can be seen in Figure 6, tags greatly impact on network's performance, significantly lowering the error rate even with a small finite set of tags such as 5, but the maximal gain is of course obtained by using M tags (one unique tag per message).

Empirically, we found that to maximize the tagged network's performance, some key parameters need to be set, in particular: the network must be sparse ($\chi > c$) ; there must be more than one unit per part ($\ell > 1$) ; and the γ memory effect [8] should be set to 1.

Thus, tags significantly enhance the retrieval process, but to be fair, we have to consider the added resources we use in the network to account for those tags. Therefore, we have done a similar simulation in Figure 7, but we here compared the clique network's efficiency with the tagged network's, and to set a comparable frame we plot the efficiencies with respect to the error rate since the tagged network can run at far higher density regimes than the clique network.

This shows that tags are a less efficient mean to extend a network's diversity than by resizing the network. However, these two means are not exclusive, and thus can be used concurrently to extend a network: first by size up to a limit, and then tags can be used to extend further. This can be a

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-390-2

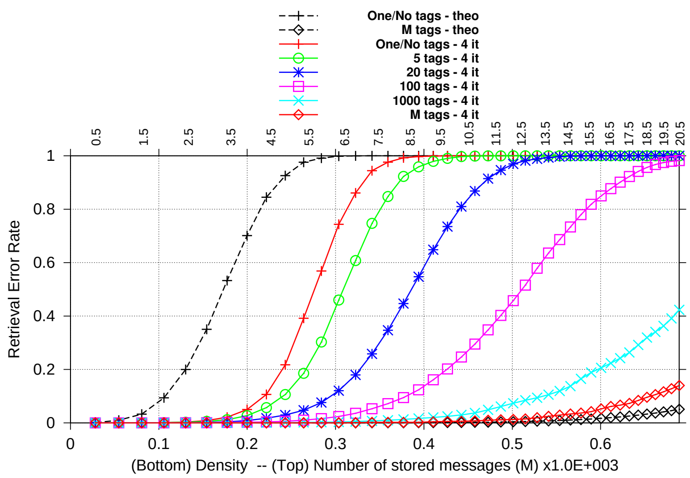

Figure 6. Error rate evolution with respect to the number of maximum tags allowed to be assigned to learned cliques. "One tag" curve corresponds to the sparse clique network [8] and serves as a reference, while "M tags" is when a unique tag is assigned per message. Network's parameters are $\chi = 16$, $c = 8$, $\ell = 64$, erasure rate $\alpha = 0.5$ (half of the c units are erased from input query) and 4 decoding iterations. The plot has two axes: the main one at the bottom defines the network's density (how much the network is full of messages), which eases comparison between different figures because the density isn't influenced by network's size, while the second axis at the top is the number of learnt messages this density corresponds to.

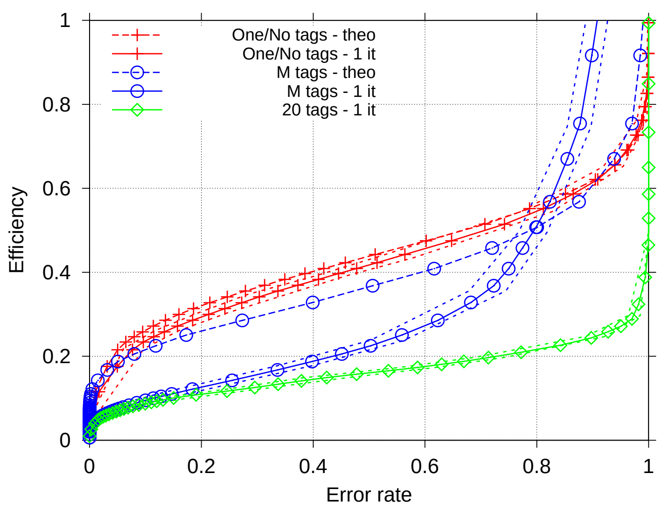

Figure 7. Efficiency with respect to the error rate for the sparse tagged network (green) compared to the clique network (red). Standard deviations are shown in dotted lines and theoretical error rates in dashed lines. Network's parameters are the same as in Figure 6 except that we here use only 1 decoding iteration. Note that the curves remain identical whatever the network's size is.

very interesting alternative for devices with a finite, static set of units, such as VLSI (Very-Large-Scale Integration) [19]–[21], ASIC (Application-Specific Integrated Circuit) [22] or FPGA (Field-Programmable Gate Array) [23] based neuromorphic and neuromemristive [24] hardware devices, where it is certainly easier to add a tag counter than to resolder the board in order to resize the network.

These results are reproducible via the complete source-code in MatLab/Octave, which is freely available online [25].

VI. Discussions

The biological mesoscopic mechanisms of learning and memory storing are extremely complex and are still a mystery. The mechanisms of forgetting are even further from grasp. This simple extension provides an elegant way to implicitly implement a forgetting mechanism with variable effect (stronger when the number of tags is low), and as such this provides a continuum to transition from a long-term memory model (clique network), where there's no overwriting nor "forgetting" of old memories, to a palimpsestic working memory [17], [26], [27]. However, even long term memory can benefit from forgetting, as this fundamental regulating mechanism seems to be tightly coupled with the retention of memories in order to mitigate the overfitting (lack of generalization) phenomenon [28]–[31]. An interesting side effect is that refreshing tags on access (i.e., when a clique is accessed, the computed major tag is assigned to each of its edges, thus "refreshing" the clique) could account for the spacing effect in learning [32]–[34], and future work in this direction may yield interesting results.

On a biological side, there is currently no observation of such an implementation of tags, and we don't argue that tags are physically embodied as-is in a biological brain. However, the tags model a concept that is far more general: affinity between synapses. Biological mechanisms behind such affinities are still merely assumptions, yet they are not implausible: synaptic discrete states [16], resonance, alike morphologies, synapse's conductance rate using variable myelination [35], [36], cascading biochemical signature [37], or a sensory modality cue. This is not as far-fetched as it sounds, as it is currently thought, according to the synaptic tagging model [38]–[42], that the very process of memory creation uses some kind of chemical tagging to convert recent, short-term and weak, memories into long-term, long-lasting and resilient, memories.

The following is merely a hypothesis, but if we suppose that the brain is ruled by stochastic processes, then if a set of synapses get created at the same moment – which may certainly be the case if synaptogenesis can be triggered in a synchronized way by glial cells just like they can trigger the synchronization of synaptic communications over wide areas [36], [43] – these synapses may get the same identical set of specific parameters (since they were created at the same moment of the stochastic process ruling the parameters of synaptic parameters), whether those parameters are a similar chemical signature, a similar myelination profile (which is very heterogeneous, probably unique, for each synapse over the brain), a similar activation threshold, a similar set of neurotransmitters, or just a similar morphology. Sharing a similar set of attributes may allow these synapses to mutually sustain, to resonate, when one or more of these synapses are activated at the same moment later in time, as some kind of reminder that they also were created together. This could be seen as some sort of evolutionary collaboration: these synapses were created at the same moment, and thus probably embody some sort of co-occurrence in information, and an affinity may perfectly encode that.

Let's now discuss about how a tagged network could technically and efficiently be implemented in neuromorphic hardwares. We mentioned in the previous section a few neuromorphic technologies that could benefit from tags, but a specific type of component could be the most efficient way to achieve a very low-energy neuromorphic device based on tags: memristors, and in particular compound memristors [24], could be a great fit for tags since they can store finite precision integers, while retaining the very interesting feature of any memristor, that is to use almost no current to maintain their state. The compound memristors could thus be used to store edges tags at a very low energy cost, which is sufficient and enough to define the storage of a whole neural network based on tags.

VII. Conclusion and future work

We have presented a new generic method to extend an associative memory network's diversity by tags, which provides an alternative to the network's size versus messages diversity trade-off. We based this method on an efficient associative memory model called clique neural network, and we provided the algorithmics underpinning this extension, which we called the *tagged neural network*. We then analyzed the network's dynamics and simulated a retrieval scenario with partially erased queries in order to study the impact on performance and efficiency of this extension, which demonstrate that tags can be used as a viable alternative, although a bit less efficient, to extend an associative memory neural network's capacity when the network's size is constrained.

Future work on this approach should focus on the analysis in a noisy scenario, where tags would not be reliable indicators anymore. In this scenario, it may be advisable to adapt the retrieval process to account for this uncertainty of the tag indicator. Another interesting avenue is the fact that the biggest source of decoding error in this network resides in the tag overwriting of old cliques by new cliques, resulting in the lost units error we described. This source of error may potentially be reduced by adapting, interestingly, the learning process, and not the decoding process, since losing units happens at the learning stage, without any influence of the decoding stage. To be more explicit: the biggest source of error is structurally encoded in the network at the learning stage, thus, optimization effort should focus on the learning process.

Also, tags are flexible indicators, whose underlying representation is totally dependent on the designer's conception. This flexibility of representation can be used for various purposes and applications beyond neuromorphic hardware, for example, by using tags as semantic cues: a tag can be seen as a label representing an identity/class of the clique pattern. Hence, a tagged network may not only increase storage diversity but also be seen as a clear identification system for specific kinds of patterns. Thus, the same tag could be used to regroup patterns that are semantically similar, or which originate from the same sensory modality (e.g., using the same tag to regroup all patterns originating from vision, another tag for audio, another one for taste, etc.). If efficient enough, this semantic use of tags could be applied successfully to a wide array of applications where we need to semantically disambiguate, such as objects class recognition in a scene.

Acknowledgment

This work was partially funded as part of the NEUCOD project by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n° 290901.

References

[1] C. Frith and R. Dolan, "The role of the prefrontal cortex in higher cognitive functions," Cognitive brain research, vol. 5, no. 1, 1996, pp. 175–181.

[2] J. Von Neumann, The computer and the brain. Yale University Press, 1974.

[3] W. Aspray, John von Neumann and the origins of modern computing. Mit Press Cambridge, MA, 1990, vol. 191.

[4] L. C. Aiello and P. Wheeler, "The expensive-tissue hypothesis: the brain and the digestive system in human and primate evolution," Current anthropology, 1995, pp. 199–221.

[5] C. W. Kuzawa et al., "Metabolic costs and evolutionary implications of human brain development," Proceedings of the National Academy of Sciences, Aug. 2014, p. 201323099.

[6] A. Knoblauch, G. Palm, and F. T. Sommer, "Memory capacities for synaptic and structural plasticity," Neural Computation, vol. 22, no. 2, 2010, pp. 289–341.

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

[8] B. K. Aliabadi, C. Berrou, V. Gripon, and J. Xiaoran, "Storing sparse messages in networks of neural cliques," IEEE transactions on neural networks and learning systems, vol. 25, no. 5, 2014, pp. 980–989.

[9] A. Aboudib, V. Gripon, and X. Jiang, "A study of retrieval algorithms of sparse messages in networks of neural cliques," in COGNITIVE 2014, The Sixth International Conference on Advanced Cognitive Technologies and Applications, 2014, pp. 140–146.

[10] S. Kurt et al., "Auditory cortical contrast enhancing by global winner-take-all inhibitory interactions," PLoS ONE, vol. 3, no. 3, Mar. 2008, p. e1735.

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

[12] D. J. Willshaw, O. P. Buneman, and H. C. Longuet-Higgins, "Non-holographic associative memory," Nature, vol. 222(5197), June 1969, pp. 960–962.

[13] T. Kojima, H. Nonaka, and T. Da-Te, "Capacity of the associative memory using the boltzmann machine learning," in Neural Networks, 1995. Proceedings., IEEE International Conference on, vol. 5, Nov 1995, pp. 2572–2577 vol.5.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, 1997, pp. 1735–1780.

[15] V. Gripon, "Networks of neural cliques," Ph.D. dissertation, Télécom-Bretagne, Jul. 2011.

[16] J. M. Montgomery and D. V. Madison, "Discrete synaptic states define a major mechanism of synapse plasticity," Trends in Neurosciences, vol. 27, no. 12, 2004, pp. 744–750.

[17] J. Sacramento and A. Wichert, "Binary willshaw learning yields high synaptic capacity for long-term familiarity memory," Biological cybernetics, vol. 106, no. 2, 2012, pp. 123–133.

[18] A. M. Dubreuil, Y. Amit, and N. Brunel, "Memory capacity of networks with stochastic binary synapses," PLoS computational biology, vol. 10, no. 8, 2014.

[19] H. University. Brainscales - neuromorphic processors. [Online]. Available: http://www.artificialbrains.com/brainscales [retrieved: 03, 2015]

[20] D. Brüderle et al., "A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems," Biological cybernetics, vol. 104, no. 4-5, 2011, pp. 263–296.

[21] G. Indiveri, "Neuromorphic bistable vlsi synapses with spike-timing-dependent plasticity," in NIPS, 2002, pp. 1091–1098.
[22] L. S. Smith, "Neuromorphic systems: past, present and future," in Brain Inspired Cognitive Systems 2008. Springer, 2010, pp. 167–182.
[23] A. Cassidy, A. G. Andreou, and J. Georgiou, "Design of a one million neuron single fpga neuromorphic system for real-time multimodal scene analysis," in Information Sciences and Systems (CISS), 2011 45th Annual Conference on. IEEE, 2011, pp. 1–6.
[24] J. Bill and R. Legenstein, "A compound memristive synapse model for statistical learning through stdp in spiking neural networks," Frontiers in Neuroscience, vol. 8, no. 412, 2014, pp. 1–18. [Online]. Available: http://www.frontiersin.org/neuromorphic_engineering/10.3389/fnins.2014.00412/abstract
[25] S. Larroque. Clique network (gbnn) implementation in octave/matlab. (doi: http://dx.doi.org/10.5281/zenodo.15788). [Online]. Available: https://github.com/lrq3000/gbnn-matlab [retrieved: 03, 2015]
[26] G. Parisi, "A memory which forgets," Journal of Physics A: Mathematical and General, vol. 19, no. 10, 1986, p. L617.
[27] D. J. Amit and S. Fusi, "Learning in neural networks with material synapses," Neural Computation, vol. 6, no. 5, 1994, pp. 957–982.
[28] A. M. Jasnow, P. K. Cullen, and D. C. Riccio, "Remembering another aspect of forgetting," Frontiers in psychology, vol. 3, 2012, p. 175.
[29] C. O'Donnell and T. J. Sejnowski, "Selective memory generalization by spatial patterning of protein synthesis," Neuron, vol. 82, no. 2, 2014, pp. 398–412.
[30] A. Dovgopoly and E. Mercado III, "A connectionist model of category learning by individuals with high-functioning autism spectrum disorder," Cognitive, Affective, & Behavioral Neuroscience, vol. 13, no. 2, 2013, pp. 371–389.
[31] R. Spencer, "Neurophysiological basis of sleep's function on memory and cognition," ISRN Physiology, vol. 2013, 2013, p. 17.
[32] H. Ebbinghaus, "Memory: A contribution to experimental psychology," Annals of Neurosciences, vol. 20, no. 4, 10 2013, pp. 155–156, [Über das gedächtnis: untersuchungen zur experimentellen psychologie. Duncker & Humblot, 1885].
[33] F. N. Dempster, "The spacing effect: A case study in the failure to apply the results of psychological research." American Psychologist, vol. 43, no. 8, 1988, p. 627.
[34] R. A. Bjork, "Assessing our own competence: Heuristics and illusions." 1999.
[35] G. S. Tomassy et al., "Distinct profiles of myelin distribution along single axons of pyramidal neurons in the neocortex," Science, vol. 344, no. 6181, 2014, pp. 319–324.
[36] N. Levine-Small, K. Mueller, R. Guebeli, B. Chow, W. Weber, and U. Egert, "Selective stimulation of astrocytes modulates activity states in neuronal networks," 2014.
[37] S. Fusi, P. J. Drew, and L. Abbott, "Cascade models of synaptically stored memories," Neuron, vol. 4, no. 45, 2005, p. 45.
[38] U. Frey and R. G. Morris, "Synaptic tagging and long-term potentiation," Nature, vol. 385, no. 6616, 1997, pp. 533–536.
[39] C. Clopath, L. Ziegler, E. Vasilaki, L. Büsing, and W. Gerstner, "Tag-trigger-consolidation: A model of early and late long-term-potentiation and depression," PLoS Computational Biology, vol. 4, no. 12, 12 2008, p. e1000248.
[40] C. Clopath, "Synaptic consolidation: an approach to long-term learning," Cognitive neurodynamics, vol. 6, no. 3, 2012, pp. 251–257.
[41] S. Sajikumar, S. Navakkode, and J. U. Frey, "Identification of compartment-and process-specific molecules required for "synaptic tagging" during long-term potentiation and long-term depression in hippocampal ca1," The Journal of neuroscience, vol. 27, no. 19, 2007, pp. 5068–5080.
[42] S. Frey and J. U. Frey, "'synaptic tagging'and 'cross-tagging'and related associative reinforcement processes of functional plasticity as the cellular basis for memory formation," Progress in brain research, vol. 169, 2008, pp. 117–143.
[43] N. Levine-Small et al., "Astrocytes drive neural network synchrony," in MEA Meeting, 2012, p. 30.