Performance of Neural Clique Networks Subject to Synaptic Noise

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

Electronics Department
IMT Atlantique
Brest, France
email: name.lastname@telecom-bretagne.eu

***Abstract*—Artificial neural networks are so-called because they are supposed to be inspired from the brain and from the ways the neurons work. While some networks are used purely for computational purpose and do not endeavor to be a plausible representation of what happens in the brain, such as deep learning neural networks, others do. However, the question of the noise in the brain and its impact on the functioning of those networks has been little-studied. For example, it is widely known that synapses misfire with a significant probability. We model this noise and study its impact on associative memories powered by neural networks: neural clique networks and Hopfield networks as a reference point. We show that synaptic noise can in fact slightly improve the performance of the decoding process of neural clique networks by avoiding local minima.**

***Keywords*–associative memories; neural clique networks; synaptic noise**

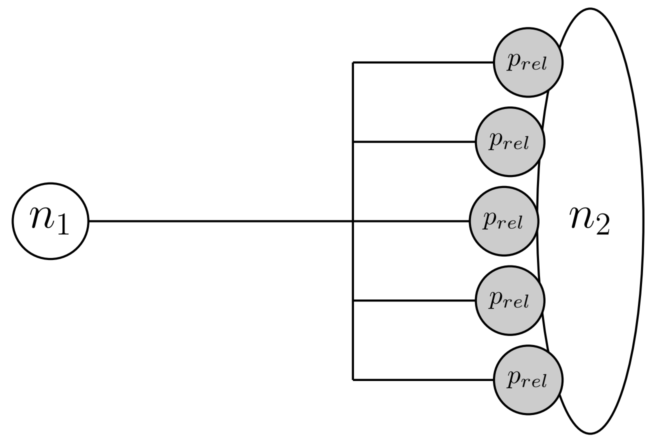

Figure 1. Example of neuronal transmission from one neuron n_1 to another n_2 with $\mathbf{n_{syn} = 5}$. Each synapse has a probability $\mathbf{p_{rel} = 0.5}$ of stimulating n_2 when n_1 is activated.

I. INTRODUCTION

There are multiple sources of noise in the brain. Indeed, they can be molecular [1], [2] or due to external neurons [2]. Other factors include synaptic noise, the intermittent failure of synapses, which seem to have a role outside of being just noise [1], [2]. In this paper, we explore this particular type of noise in details.

There are a lot of models of neural networks, which either aim at modeling what happens in the brain or simply focus on efficiency at their specific purpose. The study of the impact of noise on such artificial neural networks focusing on performance is only relevant in the context of electronical components, but that is obviously not the case when considering neural networks that strive to be biologically plausible. Such neural networks should not react adversely to noise to be considered biologically plausible. In this paper, we consider artificial neural networks that aim both at providing efficient solutions to real-world problems but also try to remain plausible as a possible way the brain works, and study the impact biological noise has on them. We focus on neural networks working as associative memories [3]–[6], and study how noise impacts their inner workings and performance. Studies have already been conducted on the impact of noise when implementing such neural networks on unreliable hardware circuits [7], where the noise is caused by unreliable components.

In this paper, we consider noise internal to the network, and more specifically synaptic noise. We show how it can be seen as an higher abstraction level than molecular noise and that it can be easily modelled. The impact of synaptic noise has been theoreticized in biological neural networks [8], but never studied with regard to artificial neural networks that are used for practical applications in computer science.

The outline of the paper is as follows. We first study how synaptic noise can be represented in Section II. In Section III, we introduce neural clique networks and discuss their biological plausibility and applications. Finally, in Section IV, we study the impact of synaptic noise on neural clique networks, both theoretically and by running simulations. We also briefly depict the impact of synaptic noise on Hopfield networks, a classical form of neural network behaving as an associative memory, for reference.

II. SYNAPTIC NOISE IN THE BRAIN

In the brain, each neuron has numerous inputs from other neurons and a single axon, which then branches to reach a multitude of other *target* neurons. Even then, there is not a single point of contact between the neuron and a target neuron, but several. The axon not only branches to reach multiple neurons, it also branches off in several synapses reaching the same target neuron.

Generally, the connection between two neurons is comprised of 5 to 25 synapses [9]. One may ask why there are so many synapses for a simple connection between two neurons. Having a few is understandable for redundancy, but there can be several tens of synapses. In fact, synapses are not reliable [9], [10], and the probability of them working typically ranges from 0.2 to 0.8 [9]. Such a configuration of synapses can help functioning when stressed under high frequency of neuronal activation by spreading the load over the different synapses [11], [12].

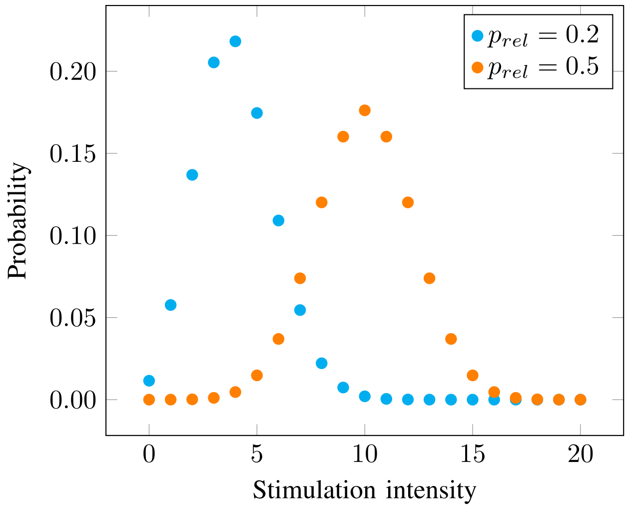

Figure 2. Probabilities for different stimulation instensities with $n_{syn} = 20$. Values for n_{syn} and p_{rel} fall within the model and are discussed at the end of the paper.

In this paper, we consider the way failing synapses affect the connection between two neurons, and then we show how it translates to artificial neural networks. To make things simpler, we consider that each connection between two neurons has n_{syn} synapses of the same strength, normalized to 1, and that they each have the same probability p_{rel} of working, independently one from another. p_{rel} is named as such as it represents the probability of releasing neuro-transmitters when stimulated by the axon. The connection between two neurons n_1 and n_2 is represented in Fig. 1. With this model, the stimulation a neuron receives from another follows a binomial law $B(tries, p_{success})$ with the parameters n_{syn} and p_{rel} as shown in Fig. 2,.

III. NEURAL CLIQUE NETWORKS

Neural clique networks are associative memories with error-correcting properties. They store binary patterns and use binary connections. They have a high capacity, comparable to binary storage. They also strive to achieve biological plausibility.

A. Minicolumns and clusters

The smallest unit in a neural clique network is a *fanal* and is based on a cortical minicolumn [13], which is a pattern comprising around 100 neurons. This pattern has been observed in humans and several other species. These fanals are organized in *clusters*, and only one fanal can be active at the same time in the same cluster, replicating the widely-used *winner-takes-all* law and alleviating concerns of energy efficiency.

A neural clique network is made of χ clusters containing ℓ fanals each.

B. Storage

A message is stored as a group of several fanals belonging to different clusters. A connection is established between two fanals if they both belong to the same message, following Hebb's law [14]. A fanal can belong to a multitude of messages. It is the connections between the fanals that define the messages and contain the information. Thus, each message is represented by a fully interconnected group of fanals, called a *clique*. In full neural clique networks, the number of fanals c making up a clique is equal to χ, the number of clusters in the network. As such, each clique contains one fanal from each cluster. As an example, a full neural clique network containing 3 messages is shown in Fig. 3.

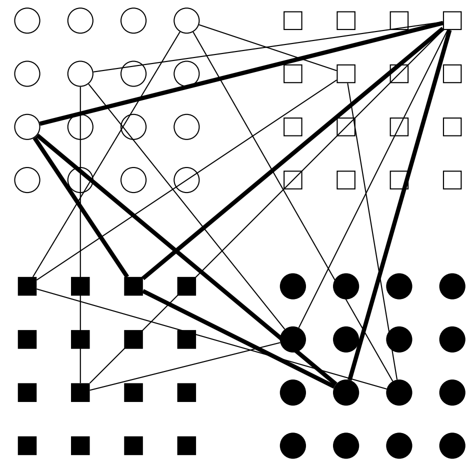

Figure 3. Storing procedure illustration. The pattern to store (with thick edges) connects units from 4 clusters of 16 units each (filled circles, filled rectangles, rectangles and circles).

In sparse neural networks, we have $1 < c < \chi$. Stored messages do not use all avalaible clusters.

C. Retrieval and performance

There are two forms of message retrieval. The network can be asked if a message exists already, which is a simple matter of testing if the fanals representing the message are fully interconnected. If a message was stored, the network will always say so, but false positives can be generated if two many messages sharing fanals in common overlap.

The second form of message retrieval is providing a partial message to the network with erasures (and possibly errors) and retrieving the full message. The algorithm for retrieving the full message simply consists of finding the fanals with the most connections to the known fanals, at most one per cluster.

As far as performance goes, the network needs a binary storage capacity of $(\chi \cdot \ell)^2/2$ bits to store the connections between the different fanals, the adjacency matrix of the graph representing the network. As shown in the examples further in this paper, a full network of 8 clusters of 256 fanals can retrieve 15000 half-erased messages with an error rate of less than 2%. The binary storage needed for storing all the messages without any error correcting mechanism would be 120 kB, and the neural network uses 260 kB of storage for the adjacency matrix, so the storage efficiency compared to raw binary is 46%, but with large resistance to message erasures.

D. Decoding algorithm

The algorithm that we use is suitable for retrieving partially erased messages. Other algorithms can be applied first to filter out irrelevant inputs or with other purposes. However, that is not our concern here.

 ISBN: 978-1-61208-531-9

1) Full networks: Each fanal gets a score, which is the number of other activated fanals connected to it. The fanals with the highest score in each cluster are activated for the next iteration, all the others are deactivated. Known fanals from the partially erased message are already provided with a high score at each iteration so they are always the only fanal activated in their own cluster. If there are several fanals with the same highest score in a cluster, they all are activated for the next iteration. The algorithm stops after several iterations, which are enough to reach a stable state with the *memory effect* introduced below. When the algorithm stops, one activated fanal is picked from each cluster, at random if there are several fanals activated.

A memory effect γ can be added, which increases performance in the case of noise-free networks. It consists of adding γ to the score of a fanal if it was activated the previous iteration. Generally, $\gamma = 1$ is used, simulating each fanal being connected to themselves. It is interpreted as an already active fanal being easier to activate than an inactive fanal of the same cluster.

2) Sparse networks: Sparse networks follow the same principle as full networks. The difference is that only a few clusters are to have a fanal activated for the next iteration amongst all the clusters. There are two algorithms in order to choose those activated clusters, the *c-global-winners-take-all* and *global-winners-take-all*. The *c-global-winners-take-all* takes the clusters with the highest score (determined by the score of the fanal with the highest score inside the cluster) until c clusters are chosen, with c being the size of the message. Then it takes all the clusters that tie the chosen cluster with the lowest score. All the chosen clusters are activated for the next iteration.

The *global-winners-take-all* just stops at taking c clusters. If several clusters have the same score, it only takes as many as it needs to have c clusters, at random. The *c-global-winners-take-all* is known to have better performance.

E. Applications

Neural clique networks have been used in various application cases. In electronics, Boguslawski et al. [15] use them to handle power management in multicore processors, showing significant improvements in energy usage compared to existing solutions. In [16], the authors propose to use neural clique networks to accelerate search in databases. They provide a fully hardware implementation of their solution using memristors, and obtain reduced energy consumption and delays compared to classical solutions.

In [17], the authors propose to use neural clique networks in combination with product quantization in order to accelerate search in visual descriptors of images. The result is a gain of a factor of about 100 in comparison to exhaustive search. Similar work has been proposed in [18].

IV. Neural Clique Networks with unreliable connections

We study the impact of unreliable connections due to synaptic noise on neural clique networks. Assuming messages are independently and uniformly distributed, we propose a full mathematical analysis of the retrieval of a partially erased pattern after one iteration, which corresponds to the second form of message retrieval previously discussed. For multiple iterations, due to the complexity of the problem and the multitude of variables we can only provide simulations, which show us the retrieval rates with or without synaptic noise.

A. One iteration

We consider the probability of finding the correct version of a partially erased message after one iteration.

1) Full network: Let's consider a full neural clique network (each cluster is used for each message). Let M be the number of messages in the network. When we try to recover a message in the network, let c_k the number of known clusters and c_e ($c_e = c - c_k$) the number of erased clusters. The density of the network, that is the probablity of a connection existing between any two fanals, is [7]:

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

Let's consider an erased cluster. Let s_0 be the correct fanal, n_{s_0} its score, s be an incorrect fanal, and n_s its score. The score of a fanal is the number of synapses connected to it that released neurotransmitters. So a fanal connected to i other fanals can get a score between 0 and $i \cdot n_{syn}$. The correct fanal s_0 is obviously connected to the other c_k known correct fanals. So for x from 0 to $n_{syn} \cdot c_k$ we have

$$\begin{aligned} P(n_{s_0} = x) &= P\left(B(n_{syn} \cdot c_k, p_{rel}) = x\right) \qquad (2)\\ &= pmf(x, n_{syn} \cdot c_k, p_{rel}) \qquad (3)\end{aligned}$$

We noted pmf the probability mass function of the binomial law B. The incorrect fanals of the erased cluster can have between 0 and c_k connections to the known correct fanals. First, we need to determine $P_E(i)$, the probability that an incorrect fanal has i connections to known correct fanals. In theory, existence of connections are not independent events, which may lead to difficult mathematical analysis [19]. In order to simplify the proofs, we make the assumption they are independent, which has been reported to be a fair approximation [6]. We find

$$P_E(i) = \binom{c_k}{i} d^i (1-d)^{c_k - i}. \tag{4}$$

Indeed, the probability of not being connected to any of the known fanals is $(1-d)^{c_k}$ and the probability of being connected to all the known fanals is d^{c_k}. The probability of being connected to only a specific known fanal is $(1-d)^{c_k-1}d$, and to be solely connected to any one of the known fanals is $c_k \cdot (1-d)^{c_k-1}d$.

And with that, we can deduce the probability of an incorrect fanal getting a score x_0 for any $0 \le x_0 \le n_{syn} \cdot c_k$:

$$P(n_s = x_0) = \sum_{i=0}^{c_k} P_E(i)\ pmf(x_0, n_{syn} \cdot i, p_{rel}) \tag{5}$$

$$P(n_s \le x_0) = \sum_{x=0}^{x_0} \sum_{i=0}^{c_k} P_E(i)\ pmf(x, n_{syn} \cdot i, p_{rel}) \tag{6}$$

Now that we have this, we can write the probability that the correct fanal is amongst the fanals with the highest scores:

$$P_{succ}(s_0) = \sum_{x_0=0}^{n_{syn}\cdot c_k} P(n_{s_0} = x_0) P(n_s \leq x_0)^{\ell-1}. \qquad (7)$$

The global probability of success, i.e., the probability that in all erased nodes the correct node is amongst the winner is $P_{succ} = P_{succ}(s_0)^{c_e}$. The error rate is $1 - P_{succ}$.

That approach is too lax, however. In practice, when looking for a message of size c, we want c symbols as the output of the network, not a set of size s ($s \geq c$) containing the c correct symbols. This means that if we have several fanals with the highest score in the same cluster, we need to pick only one of them. We then have a chance $\frac{1}{k+1}$ of picking the correct fanal in ambiguous cases, where k is the number of incorrect fanals sharing the highest score with the correct fanal.

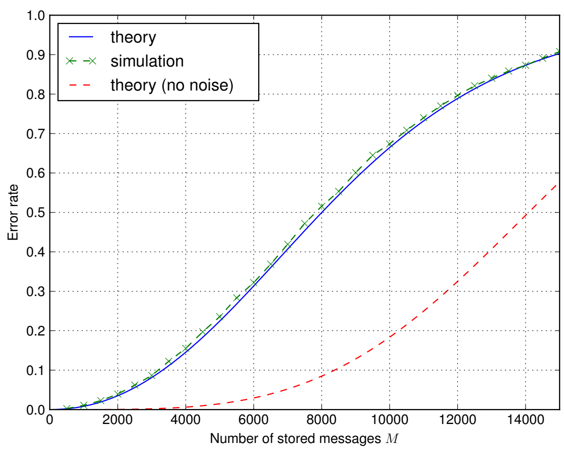

Figure 4. Analytical and simulated results with $c = 8, c_k = 4, \ell = 256, n_{syn} = 10, p_{rel} = 0.5$.

To take that into account, $P_{succ}(s_0)$ is rewritten. First, we create the probability of success for the correct fanal if its score is x_0:

$$P_{succ}(s_0, n_{s_0} = x_0) = \sum_{k=0}^{\ell-1} \frac{1}{k+1} \binom{l-1}{k} P(n_s = x_0)^k P(n_s < x_0)^{(\ell-1-k)} \qquad (8)$$

and

$$P_{succ}(s_0) = \sum_{x_0=0}^{n_{syn}\cdot c_k} P(n_{s_0} = x_0) P_{succ}(s_0, n_{s_0} = x_0) \qquad (9)$$

The results are shown on Fig. 4.

2) Sparse networks: Sparse networks are harder to tackle, due to the problem of spurious clusters. While what happens in each cluster with a correct fanal doesn't concern the other clusters, all the correct fanals belonging to the erased clusters must have a score higher than all the fanals of the erased clusters and incorrect clusters.

It is possible to formalize this second relationship, a.k.a. the lowest score of the correct fanals must be higher than the highest score of the incorrect fanals. If we consider χ the total number of clusters and c the size of a message, if $\chi >> c$ then we only need to consider that second relationship. As if each correct fanal has a score higher than the highest fanals of the $\chi - c$ incorrect clusters, then the probability that they have the highest score in their own cluster is close to 1.

Using the same logic as before, taking into account the c_e correct fanals and the $(\chi - c)\ell$ fanals belonging to incorrect clusters, we obtain the formula. First, we change $P_{succ}(s_0, n_{s_0} = x_0)$ into $P_{succ}(\ell_c, \ell_i | n_{s_0} = x_0)$, being the probability that given that l_c correct fanals have a score x_0 and the other correct fanals a higher score, all the correct fanals are chosen. We denote $\ell_i = (\chi - c)\ell$.

We get

$$P_{succ}(\ell_c, \ell_i | n_{s_0} = x_0) = \sum_{k=0}^{\ell_i} \frac{1}{\binom{k+\ell_c}{\ell_c}} \binom{\ell_i}{k} P(n_s = x_0)^k P(n_s < x_0)^{\ell_i - k} \qquad (10)$$

and

$$P_{succ} = \sum_{x_0=0}^{n_{syn}\cdot c_k} \sum_{j=1}^{c_e} \binom{c_e}{j} P(n_{s_0} = x_0)^j P(n_{s_0} > x_0)^{c_e - j} P_{succ}(j, n_{s_0} = x_0) \qquad (11)$$

B. Multiple iterations

In the case of multiple iterations, we are unable to provide a detailed mathematical analysis. But we can use simulations to study the impact of synaptic noise.

1) Parameters: The first question we have to answer is how do we know we have a solution. There's no perfect stable state due to the noise, so how do we determine when we stop the algorithm? We chose to keep a maximum of 100 iterations, in order to limit the execution time. Then, we stop if the result is stable after n_{it} iterations, n_{it} being a parameter that varies. Fig. 5 shows such a test on a full network, with n_{it} ranging from 2 to 4. From that graph we chose $n_{it} = 3$ as the best iteration number. For very low error rates, $n_{it} = 4$ is better, then $n_{it} = 3$ until an error rate of around 20%, then $n_{it} = 2$. The reason $n_{it} = 2$ becomes a better solution for higher error rates is probably because it is harder to keep a stable state then, making the algorithm reach 100 iterations before reaching a stable state with $n_{it} = 3$ or $n_{it} = 4$.

For sparse networks, experimentation has shown that $n_{it} = 2$ is a better choice.

There is also the question of the *memory effect*. The *memory effect* is known to be beneficial for networks when

 ISBN: 978-1-61208-531-9

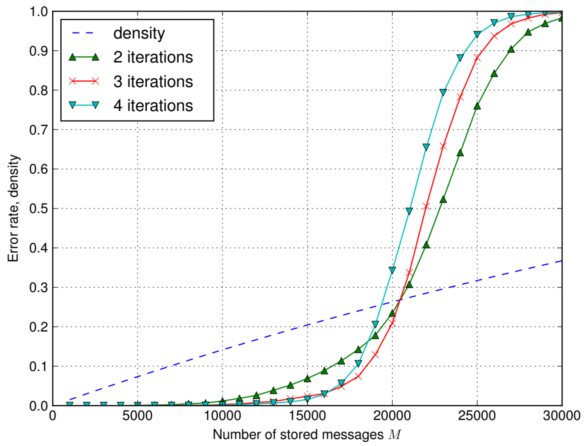

Figure 5. Error rate for a full neural clique network of parameters $c = 8$ and $\ell = 256$, with $n_{syn} = 10$, $c_k = 4$ and $p_{rel} = 0.5$. The decoding process is stopped after 2, 3 or 4 stable iterations.

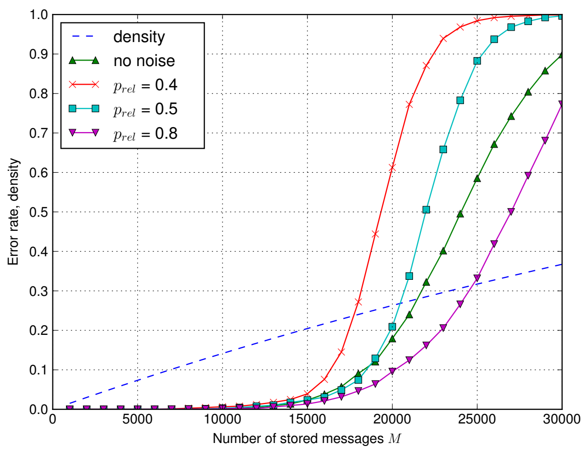

Figure 6. Error rate for a full neural clique network of parameters $c = 8$ and $\ell = 256$, with $n_{syn} = 10$, $c_k = 4$ and p_{rel} ranging from 0.4 to 1.

noise is not involved, but as noise is involved it is more beneficial to not have any memory effect. As shown on Fig. 4, the error rate after one iteration in a noisy network is important even for a low number of messages, and the memory effect would carry those mistakes onto further iterations. In order to avoid that, a memory effect of $\gamma = 0$ is chosen for noisy networks.

2) Results: As can be seen from Fig. 6 and Fig. 7, the binomial noise seems to have more beneficial effects on full networks than on sparse networks. For $p_{rel} = 0.5$, there's even a better capacity than with no noise for error rates inferior to 10%. For $p_{rel} = 0.8$, the capacity seems better for virtually all error rates.

Concerning sparse networks, a significant degradation of performance is observed for $p_{rel} \leq 0.5$ compared to when there is no noise. We observe a reduction of the capacity of the network of approximately 20% to 30% for error rates ranging from 2% to 10%, which is still a good result considering the

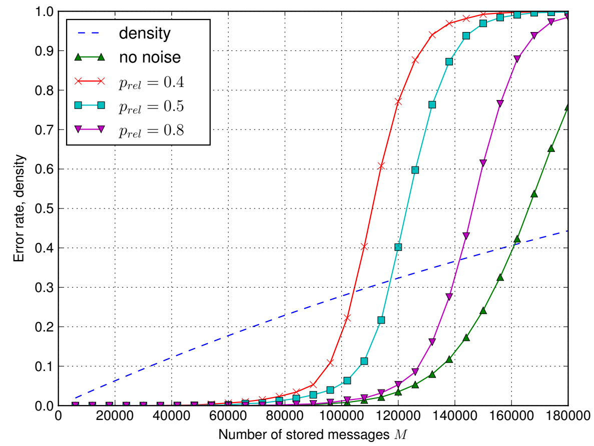

Figure 7. Error rate for a sparse neural clique network of parameters $\chi = 100$ and $c = 12$, with $n_{syn} = 10$, $c_k = 9$ and p_{rel} varying.

unreliability of the network.

Those results can be attributed to avoiding local mimima while still keeping a low deviation, with a principle loosely similar to simulated annealing. The reason full networks give better results than sparse networks would be that even if the noise can send the decoding algorithm off track, it still keeps to the same clusters in full networks.

C. Impact on Hopfield Networks

Hopfield networks [3] are artificial recurrent networks functioning as associative memories. They are made up of N neurons and can store binary messages of N bits, but the connection weights are not binary. The number of messages they can store is $\mathcal{O}(N/\log(N))$ [20]. Each pattern stored is an attractor, and when inputting data it shifts to the closest pattern stored.

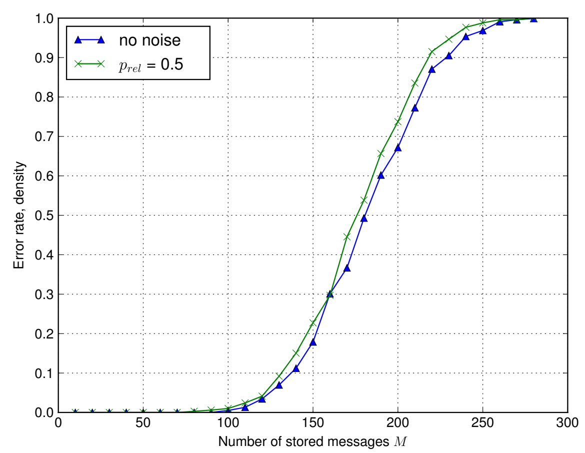

Figure 8. Error rate for an Hopfield network of 2048 neurons, with or without noise, and $\frac{1}{4}$ of the input erased.

We ran a simulation on Hopfield networks to see how such a model with precise synaptic weights would react to the large fluctuations introduced by the unreliable connections.

 ISBN: 978-1-61208-531-9

Fig. 8 shows the behavior of an Hopfield network in similar conditions to before, with $\frac{1}{4}$ of the input erased, 2048 neurons, and 10 synaptic contacts at each connection with each a probability of release of 0.5. The simulation stops when a stable state over two iterations is reached.

We can see that there is only a minor increase in the error rate. We can surmise that this is due to the high number of nodes active at the same time, averaging the effects of the binomial noise. Compared to full neural clique networks, which can take advantage of the noise, Hopfield networks seem to suffer a little decrease in performance.

V. Discussion

The analysis in this paper was based on the supposition of having 10 synaptic contacts per connection and a probability of release of the neurotransmitters of exactly 0.5 for each synaptic contact at each iteration, independently of the previous iterations.

The number of synaptic contacts per connection was chosen to be $n_{syn} = 10$, but simulations show a much lower error rate after one iteration for $n_{syn} = 20$, which is also a realistic number.

As said in [9], synaptic contacts adapt with the help of feedback, so it is wise to consider whether the release probability could exceed 0.5 where strong connections are concerned. Moreover, it is difficult to imagine that in the case of repeated stimulation of a synaptic contact, the probability of release of neurotransmitters each time is independent from the previous occurences. It makes sense that it is more probable for a synapse to release neurotransmitters if it did not with the previous stimulation, as it would be more ready. As such, the variance of the probalistic law governing the stimuli would be reduced, making the biological architecture — the brain — more reliable.

VI. Conclusion

The contribution of this paper is twofold. First, we show the significance of the noise generated by unreliable synapses, which we refer to as *synaptic noise*, and model that noise. We see it introduces randomness with high variance in the stimulation a neuron receives from another neuron, which can be represented by a binomial law depending on the number of synapses n_{syn} the connection is made of and the neurotransmitter release probability p_{rel}. We then study the impact of this noise on associative memories that strive on biological plausibility, to show if the models we have of neural networks in the brain survive scrutiny. In particular, we show the impact of synaptic noise on neural clique networks and Hopfield networks.

Suprisingly, we see that with the correct parameters such synaptic noise can in fact increase the retrieval rate of partially erased messages in neural clique networks. It is due to the noise allowing the network to overcome the local minima in its decoding process. Regarding Hopfield networks, on which a simulation is run as a reference, synaptic noise decreases performance only very slightly. As such, both associative memories sustain the test of synaptic noise and neural clique networks even benefit from it. As a future work, it would be interesting to see the impact of this kind of noise on feedforward neural networks, as they emulate the way the visual cortex works.

Acknowledgement

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. It was also funded in part by the CominLabs project Neural Communications and the Future & Rupture program.

References

[1] J. A. White, J. T. Rubinstein, and A. R. Kay, "Channel noise in neurons," Trends in neurosciences, vol. 23, no. 3, pp. 131–137, 2000.

[2] A. A. Faisal, L. P. Selen, and D. M. Wolpert, "Noise in the nervous system," Nature reviews neuroscience, vol. 9, no. 4, pp. 292–303, 2008.

[3] 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.

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

[5] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," Cognitive science, vol. 9, no. 1, pp. 147–169, 1985.

[6] 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.

[7] F. Leduc-Primeau, V. Gripon, M. Rabbat, and W. Gross, "Cluster-based associative memories built from unreliable storage," in ICASSP, pp. 8370–8374, May 2014.

[8] A. Zador, "Impact of synaptic unreliability on the information transmitted by spiking neurons," Journal of Neurophysiology, vol. 79, no. 3, pp. 1219–1229, 1998.

[9] T. Branco and K. Staras, "The probability of neurotransmitter release: variability and feedback control at single synapses," Nature Reviews Neuroscience, vol. 10, no. 5, pp. 373–383, 2009.

[10] C. Allen and C. F. Stevens, "An evaluation of causes for unreliability of synaptic transmission," Proceedings of the National Academy of Sciences, vol. 91, no. 22, pp. 10 380–10 383, 1994.

[11] M. Fauth, F. Wörgötter, and C. Tetzlaff, "The formation of multi-synaptic connections by the interaction of synaptic and structural plasticity and their functional consequences," PLoS Comput Biol, vol. 11, no. 1, pp. e1 004 031, 2015.

[12] A. Figurov et al., "Regulation of synaptic responses to high-frequency stimulation and ltp by neurotrophins in the hippocampus," Nature, vol. 381, no. 6584, pp. 706–709, 1996.

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

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

[15] B. Boguslawski, V. Gripon, F. Seguin, and F. Heitzmann, "Twin neurons for efficient real-world data distribution in networks of neural cliques: Applications in power management in electronic circuits," IEEE transactions on neural networks and learning systems, vol. 27, no. 2, pp. 375–387, 2016.

[16] H. Jarollahi et al., "A non-volatile associative memory-based context-driven search engine using 90 nm cmos mtj-hybrid logic-in-memory architecture," Journal on Emerging and Selected Topics in Circuits and Systems, vol. 4, pp. 460–474, 2014.

[17] D. Ferro, V. Gripon, and X. Jiang, "Nearest neighbour search using binary neural networks," in Neural Networks (IJCNN), 2016 International Joint Conference on. IEEE, pp. 5106–5112, 2016.

[18] C. Yu, V. Gripon, X. Jiang, and H. Jégou, "Neural associative memories as accelerators for binary vector search," in COGNITIVE 2015: 7th International Conference on Advanced Cognitive Technologies and Applications, pp. 85–89, 2015.

[19] V. Gripon, J. Heusel, M. Löwe, and F. Vermet, "A comparative study of sparse associative memories," Journal of Statistical Physics, pp. 1–25, 2015.

[20] R. McEliece, E. Posner, E. Rodemich, and S. Venkatesh, "The capacity of the hopfield associative memory," IEEE Transactions on Information Theory, vol. 33, no. 4, pp. 461–482, 1987.

 ISBN: 978-1-61208-531-9