Noise-Enhanced Associative Memories

Amin Karbasi
Swiss Federal Institute of Technology Zurich
amin.karbasi@inf.ethz.ch

Amir Hesam Salavati
Ecole Polytechnique Federale de Lausanne
hesam.salavati@epfl.ch

Amin Shokrollahi
Ecole Polytechnique Federale de Lausanne
amin.shokrollahi@epfl.ch

Lav R. Varshney
IBM Thomas J. Watson Research Center
varshney@alum.mit.edu

Abstract

Recent advances in associative memory design through structured pattern sets and graph-based inference algorithms allow reliable learning and recall of exponential numbers of patterns. Though these designs correct external errors in recall, they assume neurons compute noiselessly, in contrast to highly variable neurons in hippocampus and olfactory cortex. Here we consider associative memories with noisy internal computations and analytically characterize performance. As long as internal noise is less than a specified threshold, error probability in the recall phase can be made exceedingly small. More surprisingly, we show internal noise actually improves performance of the recall phase. Computational experiments lend additional support to our theoretical analysis. This work suggests a functional benefit to noisy neurons in biological neuronal networks.

1 Introduction

Hippocampus, olfactory cortex, and other brain regions are thought to operate as associative memories [1,2], having the ability to learn patterns from presented inputs, store a large number of patterns, and retrieve them reliably in the face of noisy or corrupted queries [3–5]. Associative memory models are designed to have these properties.

Although such information storage and recall seemingly falls into the information-theoretic framework, where an exponential number of messages can be communicated reliably with a linear number of symbols, classical associative memory models could only store a linear number of patterns [4]. A primary reason is classical models require memorizing a randomly chosen set of patterns. By enforcing structure and redundancy in the possible set of memorizable patterns—like natural stimuli [6], internal neural representations [7], and error-control codewords—advances in associative memory design allow storage of an exponential number of patterns [8,9], just like in communication systems.

Information-theoretic and associative memory models of storage have been used to predict experimentally measurable properties of synapses in the mammalian brain [10, 11]. But contrary to the fact noise is present in computational operations of the brain [12, 13], associative memory models with exponential capacities have assumed no internal noise in the computational nodes. The purpose here is to model internal noise and study whether such associative memories still operate reliably. Surprisingly, we find internal noise actually enhances recall performance, suggesting a functional role for variability in the brain.

In particular we consider a multi-level, graph code-based, associative memory model [9] and find that even if all components are noisy, the final error probability in recall can be made exceedingly small. We characterize a threshold phenomenon and show how to optimize algorithm parameters when knowing statistical properties of internal noise. Rather counterintuitively the performance of the memory model *improves* in the presence of internal neural noise, as observed previously as *stochastic resonance* [13, 14]. There are mathematical connections to perturbed simplex algorithms for linear programing [15], where internal noise pushes the algorithm out of local minima.

The benefit of internal noise has been noted previously in associative memory models with stochastic update rules, cf. [16]. However, our framework differs from previous approaches in three key aspects. First, our memory model is different, which makes extension of previous analysis nontrivial. Second, and perhaps most importantly, pattern retrieval capacity in previous approaches *decreases* with internal noise, cf. [16, Fig. 6.1], in that increasing internal noise helps correct more external errors, but also reduces the number of memorizable patterns. In our framework, internal noise does not affect pattern retrieval capacity (up to a threshold) but improves recall performance. Finally, our noise model has bounded rather than Gaussian noise, and so a suitable network may achieve *perfect* recall despite internal noise.

Reliably storing information in memory systems constructed completely from unreliable components is a classical problem in fault-tolerant computing [17–19], where models have used random access architectures with sequential correcting networks. Although direct comparison is difficult since notions of circuit complexity are different, our work also demonstrates that associative memory architectures constructed from unreliable components can store information reliably.

Building on the idea of structured pattern sets [20], our associative memory model [9] relies on the fact all patterns to be learned lie in a low-dimensional subspace. Learning features of a low-dimensional space is very similar to autoencoders [21] and has structural similarities to Deep Belief Networks (DBNs), particularly Convolutional Neural Networks [22].

2 Associative Memory Model

Notation and basic structure: In our model, a neuron can assume an integer-valued state from the set $\mathcal{S} = \{0, \dots, S-1\}$, interpreted as the short term firing rate of neurons. A neuron updates its state based on the states of its neighbor $\{s_i\}_{i=1}^n$ as follows. It first computes a weighted sum $h = \sum_{i=1}^n w_i s_i + \zeta$, where w_i is the weight of the link from s_i and ζ is the *internal noise*, and then applies nonlinear function $f : \mathbb{R} \to \mathcal{S}$ to h.

An associative memory is represented by a weighted bipartite graph, G, with pattern neurons and constraint neurons. Each pattern $x = (x_1, \dots, x_n)$ is a vector of length n, where $x_i \in \mathcal{S}, i = 1, \dots, n$. Following [9], the focus is on recalling patterns with strong *local correlation* among entries. Hence, we divide entries of each pattern x into L *overlapping* sub-patterns of lengths $n_1, \dots, n_L$. Due to overlaps, a pattern neuron can be a member of multiple subpatterns, as in Fig. 1a. The ith subpattern is denoted $x^{(i)} = (x_1^{(i)}, \dots, x_{n_i}^{(i)})$, and local correlations are assumed to be in the form of subspaces, i.e. the subpatterns $x^{(i)}$ form a subspace of dimension $k_i < n_i$.

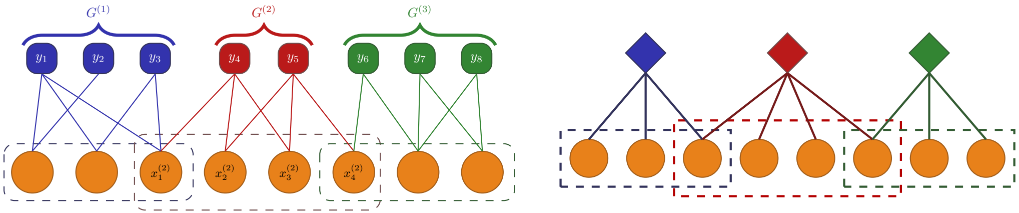
We capture the local correlations by learning a set of linear constraints over each subspace corresponding to the dual vectors orthogonal to that subspace. More specifically, let $\{w_1^{(i)}, \dots, w_{m_i}^{(i)}\}$ be a set of dual vectors orthogonal to all subpatterns $x^{(i)}$ of cluster i. Then:

$$y_j^{(i)} = (w_j^{(i)})^T \cdot x^{(i)} = 0, \quad \text{for all } j \in \{1, \dots, m_i\} \text{ and for all } i \in \{1, \dots, L\}. \tag{1}$$

Eq. (1) can be rewritten as $W^{(i)} \cdot x^{(i)} = 0$ where $W^{(i)} = [w_1^{(i)} | w_2^{(i)} | \dots | w_{m_i}^{(i)}]^T$ is the matrix of dual vectors. Now we use a bipartite graph with connectivity matrix determined by $W^{(i)}$ to represent the subspace constraints learned from subpattern $x^{(i)}$; this graph is called *cluster* i. We developed an efficient way of learning $W^{(i)}$ in [9], also used here. Briefly, in each iteration of learning:

1. Pick a pattern x at random from the dataset;
2. Adjust weight vectors $w_j^{(i)}$ for $j = \{1, \dots, m_i\}$ and $i = \{1, \dots, L\}$ such that the projection of x onto $w_j^{(i)}$ is reduced. Apply a sparsity penalty to favor sparse solutions.

This process repeats until all weights are orthogonal to the patterns in the dataset or the maximum iteration limit is reached. The learning rule allows us to assume the weight matrices $W^{(i)}$ are known and satisfy $W^{(i)} \cdot x^{(i)} = 0$ for all patterns x in the dataset $\mathcal{X}$, in this paper.



(a) Bipartite graph G.

(b) Contraction graph $\widetilde{G}$.

Figure 1: The proposed neural associative memory with overlapping clusters.

For the forthcoming asymptotic analysis, we need to define a *contracted graph* $\widetilde{G}$ whose connectivity matrix is denoted $\tilde{W}$ and has size $L \times n$. This is a bipartite graph in which constraints in each cluster are represented by a single neuron. Thus, if pattern neuron x_j is connected to cluster i, $\tilde{W}_{ij} = 1$; otherwise $\tilde{W}_{ij} = 0$. We also define the degree distribution from an *edge perspective* over $\widetilde{G}$, using $\widetilde{\lambda}(z) = \sum_j \widetilde{\lambda}_j z^j$ and $\widetilde{\rho}(z) = \sum_j \widetilde{\rho}_j z^{j-1}$ where $\widetilde{\lambda}_j$ (resp., $\widetilde{\rho}_j$) equals the fraction of edges that connect to pattern (resp., cluster) nodes of degree j.

Noise model: There are two types of noise in our model: *external errors* and *internal noise*. As mentioned earlier, a neural network should be able to retrieve memorized pattern x from its corrupted version y due to external errors. We assume the external error is an additive vector of size n, denoted by z satisfying $y = x + z$, whose entries assume values independently from $\{-1, 0, +1\}$[1] with corresponding probabilities $p_{-1} = p_{+1} = \epsilon/2$ and $p_0 = 1 - \epsilon$. The realization of the external error on subpattern $x^{(i)}$ is denoted $z^{(i)}$. Note that the subspace assumption implies $W \cdot y = W \cdot z$ and $W^{(i)} \cdot y^{(i)} = W^{(i)} \cdot z^{(i)}$ for all i. Neurons also suffer from internal noise. We consider a bounded noise model, i.e. a random number uniformly distributed in the intervals $[-\upsilon, \upsilon]$ and $[-\nu, \nu]$ for the pattern and constraint neurons, respectively ($\upsilon, \nu < 1$).

The goal of recall is to filter the external error z to obtain the desired pattern x as the true states of the pattern neurons. When neurons compute noiselessly, this task may be achieved by exploiting the fact the set of patterns $x \in \mathcal{X}$ to satisfy the set of constraints $W^{(i)} \cdot x^{(i)} = 0$. However, it is not clear how to accomplish this objective when the neural computations are noisy. Rather surprisingly, we show that eliminating external errors is not only possible in the presence of internal noise, but that neural networks with moderate internal noise demonstrate better external noise resilience.

Recall algorithms: To efficiently deal with external errors, we use a combination of Alg. 1 and Alg. 2. The role of Alg. 1 is to correct at least a single external error in each cluster. Without overlaps between clusters, the error resilience of the network is limited. Alg. 2 exploits the overlaps: it helps clusters with external errors recover their correct states by using the reliable information from clusters that do not have external errors. The error resilience of the resulting combination thereby drastically improves. Now we describe the details of Alg. 1 and Alg. 2 more precisely.

Alg. 1 performs a series of forward and backward iterations in each cluster $G^{(l)}$ to remove (at least) one external error from its input domain. At each iteration, the pattern neurons locally decide whether to update their current state: if the amount of feedback received by a pattern neuron exceeds a threshold, the neuron updates its state, and otherwise remains as is. With abuse of notation, let us denote messages transmitted by pattern node i and constraint node j at round t by $x_i(t)$ and $y_j(t)$, respectively. In round 0, pattern nodes are initialized by a pattern $\hat{x}$, sampled from dataset $\mathcal{X}$, perturbed by external errors z, i.e., $x(0) = \hat{x} + z$. Thus, for cluster ℓ we have $x^{(\ell)}(0) = \hat{x}^{(\ell)} + z^{(\ell)}$, where $z^{(\ell)}$ is the realization of errors on subpattern $x^{(\ell)}$.

In round t, the pattern and constraint neurons update their states using feedback from neighbors. However since neural computations are faulty, decisions made by neurons may not be reliable. To minimize effects of internal noise, we use the following update rule for pattern node i in cluster ℓ:

$$x_i^{(\ell)}(t+1) = \begin{cases} x_i^{(\ell)}(t) - \text{sign}(g_i^{(\ell)}(t)), & \text{if } |g_i^{(\ell)}(t)| \geq \varphi \\ x_i^{(\ell)}(t), & \text{otherwise,} \end{cases} \tag{2}$$

[1] Note that the proposed algorithms also work with larger noise values, i.e. from a set $\{-a, \dots, a\}$ for some $a \in \mathbb{N}$, see [23]; the ±1 noise model is presented here for simplicity.

Algorithm 1 Intra-Module Error Correction

```
Input: Training set $\mathcal{X}$, thresholds $\varphi, \psi$, iteration $t_{\max}$
Output: $x_1^{(\ell)}, x_2^{(\ell)}, \dots, x_{n_\ell}^{(\ell)}$
1: for $t = 1 \to t_{\max}$ do
2:     Forward iteration: Calculate the input $h_i^{(\ell)} = \sum_{j=1}^{n_\ell} W_{ij}^{(\ell)} x_j^{(\ell)} + v_i$, for each neuron $y_i^{(\ell)}$ and set $y_i^{(\ell)} = f(h_i^{(\ell)}, \psi)$.
3:     Backward iteration: Each neuron $x_j^{(\ell)}$ computes
           $g_j^{(\ell)} = \frac{\sum_{i=1}^{m_\ell} \text{sign}(W_{ij}^{(\ell)}) y_i^{(\ell)}}{\sum_{i=1}^{m_\ell} \text{sign}(|W_{ij}^{(\ell)}|)} + u_i.$
4:     Update the state of each pattern neuron $j$ according to $x_j^{(\ell)} = x_j^{(\ell)} - \text{sign}(g_j^{(\ell)})$ only if $|g_j^{(\ell)}| > \varphi$.
5: end for
```

Algorithm 2 Sequential Peeling Algorithm

```
Input: $\widetilde{G}, G^{(1)}, G^{(2)}, \dots, G^{(L)}$.
Output: $x_1, x_2, \dots, x_n$
1: while there is an unsatisfied $v^{(\ell)}$ do
2:     for $\ell = 1 \to L$ do
3:         If $v^{(\ell)}$ is unsatisfied, apply Alg. 1 to cluster $G^{(l)}$.
4:         If $v^{(\ell)}$ remained unsatisfied, revert the state of pattern neurons connected to $v^{(\ell)}$ to their initial state. Otherwise, keep their current states.
5:     end for
6: end while
7: Declare $x_1, x_2, \dots, x_n$ if all $v^{(\ell)}$'s are satisfied. Otherwise, declare failure.
```

where φ is the update threshold and $g_i^{(\ell)}(t) = \big((\text{sign}(W^{(\ell)})^\top \cdot y^{(\ell)}(t)\big)_i / d_i + u_i$.[2] Here, d_i is the degree of pattern node i, $y^{(\ell)}(t) = [y_1^{(\ell)}(t), \dots, y_{m_\ell}^{(\ell)}(t)]$ is the vector of messages transmitted by the constraint neurons in cluster ℓ, and u_i is the random noise affecting pattern node i. Basically, the term $g_i^{(\ell)}(t)$ reflects the (average) belief of constraint nodes connected to pattern neuron i about its correct value. If $g_i^{(\ell)}(t)$ is larger than a specified threshold φ it means most of the connected constraints suggest the current state $x_i^{(\ell)}(t)$ is not correct, hence, a change should be made. Note this average belief is diluted by the internal noise of neuron i. As mentioned earlier, u_i is uniformly distributed in the interval $[-\upsilon, \upsilon]$, for some $\upsilon < 1$. On the constraint side, the update rule is:

$$y_i^{(\ell)}(t) = f(h_i^{(\ell)}(t), \psi) = \begin{cases} +1, & \text{if } h_i^{(\ell)}(t) \geq \psi \\ 0, & \text{if } -\psi \leq h_i^{(\ell)}(t) \leq \psi \\ -1, & \text{otherwise,} \end{cases} \tag{3}$$

where ψ is the update threshold and $h_i^{(\ell)}(t) = \big(W^{(\ell)} \cdot x^{(\ell)}(t)\big)_i + v_i$. Here, $x^{(\ell)}(t) = [x_1^{(\ell)}(t), \dots, x_{n_\ell}^{(\ell)}(t)]$ is the vector of messages transmitted by the pattern neurons and v_i is the random noise affecting node i. As before, we consider a bounded noise model for v_i, i.e., it is uniformly distributed in the interval $[-\nu, \nu]$ for some $\nu < 1$.[3]

The error correction ability of Alg. 1 is fairly limited, as determined analytically and through simulations [23]. In essence, Alg. 1 can correct one external error with high probability, but degrades terribly against two or more external errors. Working independently, clusters cannot correct more than a few external errors, but their combined performance is much better. As clusters overlap, they help each other in resolving external errors: a cluster whose pattern neurons are in their correct states can *always* provide truthful information to neighboring clusters. This property is exploited in Alg. 2 by applying Alg. 1 in a round-robin fashion to each cluster. Clusters either eliminate their internal noise in which case they keep their new states and can now help other clusters, or revert back to their original states. Note that by such a scheduling scheme, neurons can only change their states towards correct values. This scheduling technique is similar in spirit to the peeling algorithm [24].

3 Recall Performance Analysis

Now let us analyze recall error performance. The following lemma shows that if φ and ψ are chosen properly, then in the absence of external errors the constraints remain satisfied and internal noise cannot result in violations. This is a crucial property for Alg. 2, as it allows one to determine whether a cluster has successfully eliminated external errors (Step 4 of algorithm) by merely checking the satisfaction of all constraint nodes.

[2] Note that $x_i^{(\ell)}(t+1)$ is further mapped to the interval $[0, S-1]$ by saturating the values below 0 and above $S-1$ to 0 and $S-1$ respectively. The corresponding equations are omitted for brevity.

[3] Note that although the values of $y_i^{(\ell)}(t)$ can be shifted to $0, 1, 2$, instead of $-1, 0, 1$ to match our assumption that neural states are non-negative, we leave them as such to simplify later analysis.

Lemma 1. *In the absence of external errors, the probability that a constraint neuron (resp. pattern neuron) in cluster ℓ makes a wrong decision due to its internal noise is given by* $\pi_0^{(\ell)} = \max\left(0, \frac{\nu-\psi}{\nu}\right)$ *(resp.* $P_0^{(\ell)} = \max\left(0, \frac{\upsilon-\varphi}{\upsilon}\right)$*).*

Proof is given in [23]. In the sequel, we assume $\varphi > \upsilon$ and $\psi > \nu$ so that $\pi_0^{(\ell)} = 0$ and $P_0^{(\ell)} = 0$. However, an external error combined with internal noise may still push neurons to an incorrect state.

Given the above lemma and our neural architecture, we can prove the following surprising result: in the asymptotic regime of increasing number of iterations of Alg. 2, a neural network with internal noise outperforms one without. Let us define the fraction of errors corrected by the noiseless and noisy neural network (parametrized by υ and ν) after T iterations of Alg. 2 by $\Lambda(T)$ and $\Lambda_{\upsilon,\nu}(T)$, respectively. Note that both $\Lambda(T) \leq 1$ and $\Lambda_{\upsilon,\nu}(T) \leq 1$ are non-decreasing sequences of T. Hence, their limiting values are well defined: $\lim_{T\to\infty} \Lambda(T) = \Lambda^*$ and $\lim_{T\to\infty} \Lambda_{\upsilon,\nu}(T) = \Lambda^*_{\upsilon,\nu}$.

Theorem 2. *Let us choose φ and ψ so that $\pi_0^{(\ell)} = 0$ and $P_0^{(\ell)} = 0$ for all $\ell \in \{1, \ldots, L\}$. For the same realization of external errors, we have $\Lambda^*_{\upsilon,\nu} \geq \Lambda^*$.*

Proof is given in [23]. The high level idea why a noisy network outperforms a noiseless one comes from understanding stopping sets. These are realizations of external errors where the iterative Alg. 2 cannot correct all of them. We show that the stopping set shrinks as we add internal noise. In other words, we show that in the limit of $T \to \infty$ the noisy network can correct any error pattern that can be corrected by the noiseless version and it can also get out of stopping sets that cause the noiseless network to fail. Thus, the supposedly harmful internal noise will help Alg. 2 to avoid stopping sets.

Thm. 2 suggests the only possible downside with using a noisy network is its possible running time in eliminating external errors: the noisy neural network may need more iterations to achieve the same error correction performance. Interestingly, our empirical experiments show that in certain scenarios, even the running time improves when using a noisy network.

Thm. 2 indicates that noisy neural networks (under our model) outperform noiseless ones, but does not specify the level of errors that such networks can correct. Now we derive a theoretical upper bound on error correction performance. To this end, let P_{c_i} be the average probability that a cluster can correct i external errors in its domain. The following theorem gives a simple condition under which Alg. 2 can correct a linear fraction of external errors (in terms of n) with high probability. The condition involves $\tilde{\lambda}$ and $\tilde{\rho}$, the degree distributions of the contracted graph $\tilde{G}$.

Theorem 3. *Under the assumptions that graph $\widetilde{G}$ grows large and it is chosen randomly with degree distributions given by $\widetilde{\lambda}$ and $\widetilde{\rho}$, Alg. 2 is successful if*

$$\epsilon\widetilde{\lambda}\left(1 - \sum_{i\geq 1} P_{c_i} \frac{z^{i-1}}{i!} \cdot \frac{d^{i-1}\widetilde{\rho}(1-z)}{dz^{i-1}}\right) < z, \ for \ z \in [0, \epsilon]. \tag{4}$$

Proof is given in [23] and is based on the density evolution technique [25]. Thm. 3 states that for any fraction of errors $\Lambda_{\upsilon,\nu} \leq \Lambda^*_{\upsilon,\nu}$ that satisfies the above recursive formula, Alg. 2 will be successful with probability close to one. Note that the first fixed point of the above recursive equation dictates the maximum fraction of errors $\Lambda^*_{\upsilon,\nu}$ that our model can correct. For the special case of $P_{c_1} = 1$ and $P_{c_i} = 0, \forall i > 1$, we obtain $\epsilon\widetilde{\lambda}1 - \widetilde{\rho}(1-z)) < z$, the same condition given in [9]. Thm. 3 takes into account the contribution of all P_{c_i} terms and as we will see, their values change as we incorporate the effect of internal noise υ and ν. We will see in the next section that the maximum value of P_{c_i} does not occur when the internal noise is equal to zero, i.e. $\upsilon = \nu = 0$, but instead when the neurons are contaminated with internal noise! This also suggests that even individual clusters are able to correct more errors in the presence of internal noise. A typical trend of P_{c_i}'s in terms of ν is shown in Fig. 2 (note that maximum values are not at $\upsilon = 0$).

3.1 Simulations

Now we consider simulation results for a finite system. To learn the subspace constraints (1) for each cluster $G^{(\ell)}$ we use the learning algorithm in [9]. Henceforth, we assume that the weight matrix W

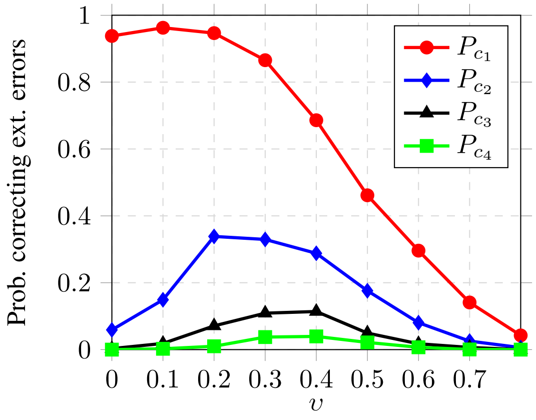


Figure 2: The value of P_{c_i} as a function of pattern neurons noise υ for $i = 1, \dots, 4$. The noise at constraint neurons is assumed to be zero ($\nu = 0$).

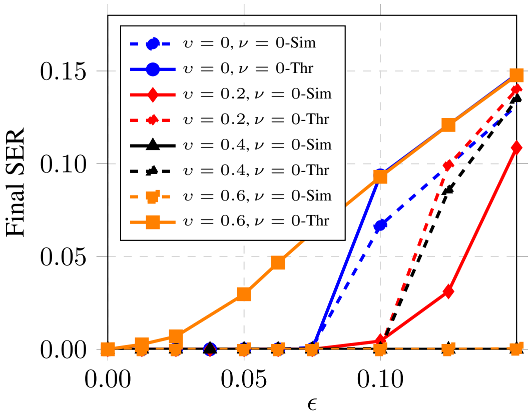
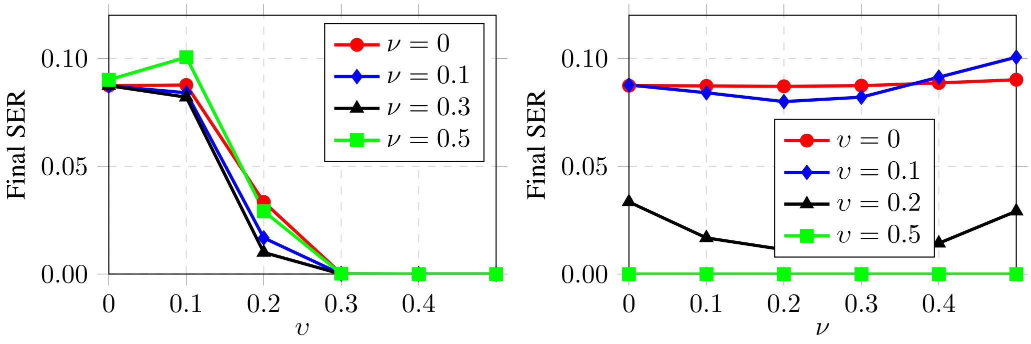


Figure 3: The final SER for a network with $n = 400$, $L = 50$ cf. [9]. The blue curves correspond to the noiseless neural network.



(a) Final SER as function of υ for $\epsilon = 0.125$.

(b) The effect of ν on the final SER for $\epsilon = 0.125$

Figure 4: The final SER vs. internal noise parameters at pattern and constraint neurons for $\epsilon = 0.125$

is known and given. In our setup, we consider a network of size $n = 400$ with $L = 50$ clusters. We have 40 pattern nodes and 20 constraint nodes in each cluster, on average. External error is modeled by randomly generated vectors z with entries ± 1 with probability ϵ and 0 otherwise. Vector z is added to the correct patterns, which satisfy (1). For recall, Alg. 2 is used and results are reported in terms of Symbol Error Rate (SER) as the level of external error (ϵ) or internal noise (υ, ν) is changed; this involves counting positions where the output of Alg. 2 differs from the correct pattern.

3.1.1 Symbol Error Rate as a function of Internal Noise

Fig. 3 illustrates the final SER of our algorithm for different values of υ and ν. Recall that υ and ν quantify the level of noise in pattern and constraint neurons, respectively. Dashed lines in Fig. 3 are simulation results whereas solid lines are theoretical upper bounds provided in this paper. As evident, there is a threshold phenomenon such that SER is negligible for $\epsilon \leq \epsilon^*$ and grows beyond this threshold. As expected, simulation results are better than the theoretical bounds. In particular, the gap is relatively large as υ moves towards one.

A more interesting trend in Fig. 3 is the fact internal noise helps in achieving better performance, as predicted by theoretical analysis (Thm. 2). Notice how ϵ^* moves towards one as ν increases.

This phenomenon is examined more closely in Figs. 4a and 4b where ϵ is fixed to 0.125 while υ and ν vary. As we see, a moderate amount of internal noise at both pattern and constraint neurons improves performance. There is an optimum point (υ^*, ν^*) for which the SER reaches its minimum. Fig. 4b indicates for instance that $\nu^* \approx 0.25$, beyond which SER deteriorates.

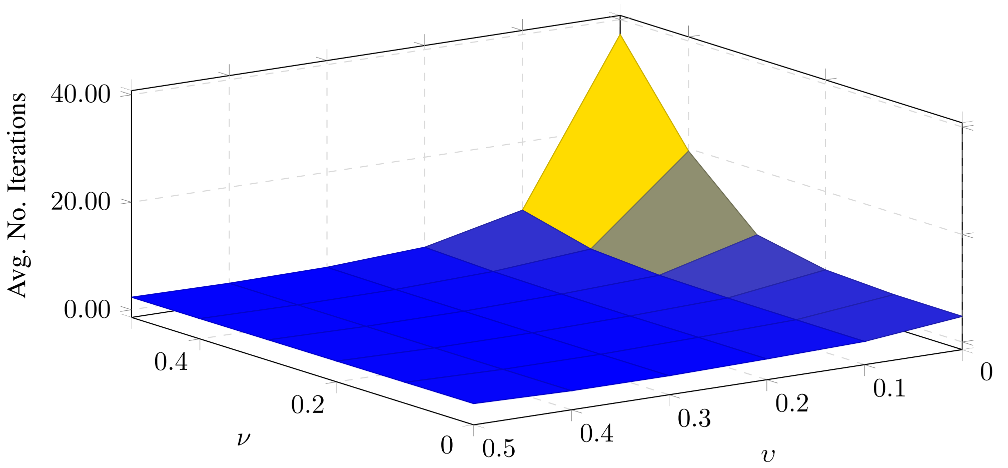
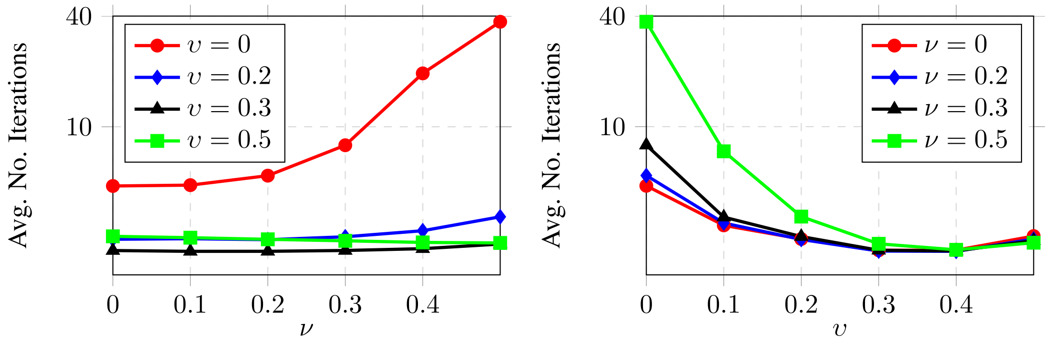


Figure 5: The effect of internal noise on the number of iterations of Alg. 2 when $\epsilon = 0.075$.



(a) Effect of internal noise at constraint neurons side.

(b) Effect of internal noise at pattern neurons side.

Figure 6: The effect of internal noise on the number of iterations performed by Alg. 2, for different values of υ and ν with $\epsilon = 0.075$. The average iteration number of 40 indicate the failure of Alg. 2.

3.2 Recall Time as a function of Internal Noise

Fig. 5 illustrates the number of iterations performed by Alg. 2 for correcting the external errors when ϵ is fixed to 0.075. We stop whenever the algorithm corrects all external errors or declare a recall error if all errors were not corrected in 40 iterations. Thus, the corresponding areas in the figure where the number of iterations reaches 40 indicates decoding failure. Figs. 6b and 6a are projected versions of Fig. 5 and show the average number of iterations as a function of υ and ν, respectively.

The amount of internal noise drastically affects the speed of Alg. 2. First, from Fig. 5 and 6a observe that running time is more sensitive to noise at constraint neurons than pattern neurons and that the algorithms become slower as noise at constraint neurons is increased. In contrast, note that internal noise at the pattern neurons may improve the running time, as seen in Fig. 6b.

Note that the results presented here are for the case where the noiseless decoder succeeds as well and its average number of iterations is pretty close to the optimal value (see Fig. 5). In [23], we provide additional results corresponding to $\epsilon = 0.125$, where the noiseless decoder encounters stopping sets while the noisy decoder is still capable of correcting external errors; there we see that the optimal running time occurs when the neurons have a fair amount of internal noise.

In [23] we also provide results of a study for a slightly modified scenario where there is only internal noise and no external errors. Furthermore, $\varphi < \upsilon$. Thus, the internal noise can now cause neurons to make wrong decisions, even in the absence of external errors. There, we witness the more familiar phenomenon where increasing the amount of internal noise results in a worse performance. This finding emphasizes the importance of choosing update threshold φ and ψ according to Lem. 1.

4 Pattern Retrieval Capacity

For completeness, we review pattern retrieval capacity results from [9] to show that the proposed model is capable of memorizing an exponentially large number of patterns. First, note that since the patterns from a subspace, the number of patterns C does not have any effect on the learning or recall algorithms (except for its obvious influence on the learning time). Thus, in order to show that the pattern retrieval capacity is exponential in n, all we need to demonstrate is that there exists a training set $\mathcal{X}$ with C patterns of length n for which $C \propto a^{rn}$, for some $a > 1$ and $0 < r$.

Theorem 4 ([9]). *Let $\mathcal{X}$ be a $\mathcal{C} \times n$ matrix, formed by $\mathcal{C}$ vectors of length n with entries from the set $\mathcal{Q}$. Furthermore, let $k = rn$ for some $0 < r < 1$. Then, there exists a set of vectors for which $\mathcal{C} = a^{rn}$, with $a > 1$, and rank$(\mathcal{X}) = k < n$.*

The proof is constructive: we create a dataset $\mathcal{X}$ such that it can be memorized by the proposed neural network and satisfies the required properties, i.e. the subpatterns form a subspace and pattern entries are integer values from the set $\mathcal{S} = \{0, \ldots, S-1\}$. The complete proof can be found in [9].

5 Discussion

We have demonstrated that associative memories with exponential capacity still work reliably even when built from unreliable hardware, addressing a major problem in fault-tolerant computing and further arguing for the viability of associative memory models for the (noisy) mammalian brain. After all, brain regions modeled as associative memories, such as the hippocampus and the olfactory cortex, certainly do display internal noise [12, 13, 26]. The linear-nonlinear computations of Alg. 1 are certainly biologically plausible, but implementing the state reversion computation of Alg. 2 in a biologically plausible way remains an open question.

We found a threshold phenomenon for reliable operation, which manifests the tradeoff between the amount of internal noise and the amount of external noise that the system can handle. In fact, we showed that internal noise actually improves the performance of the network in dealing with external errors, up to some optimal value. This is a manifestation of the *stochastic facilitation* [13] or *noise enhancement* [14] phenomenon that has been observed in other neuronal and signal processing systems, providing a functional benefit to variability in the operation of neural systems.

The associative memory design developed herein uses thresholding operations in the message-passing algorithm for recall; as part of our investigation, we optimized these neural firing thresholds based on the statistics of the internal noise. As noted by Sarpeshkar in describing the properties of analog and digital computing circuits, "In a cascade of analog stages, noise starts to accumulate. Thus, complex systems with many stages are difficult to build. [In digital systems] Round-off error does not accumulate significantly for many computations. Thus, complex systems with many stages are easy to build" [27]. One key to our result is capturing this benefit of digital processing (thresholding to prevent the build up of errors due to internal noise) as well as a modular architecture which allows us to correct a linear number of external errors (in terms of the pattern length).

This paper focused on recall, however learning is the other critical stage of associative memory operation. Indeed, information storage in nervous systems is said to be subject to storage (or learning) noise, *in situ* noise, and retrieval (or recall) noise [11, Fig. 1]. It should be noted, however, there is no essential loss by combining learning noise and *in situ* noise into what we have called external error herein, cf. [19, Fn. 1 and Prop. 1]. Thus our basic qualitative result extends to the setting where the learning and stored phases are also performed with noisy hardware.

Going forward, it is of interest to investigate other neural information processing models that explicitly incorporate internal noise and see whether they provide insight into observed empirical phenomena. As an example, we might be able to understand the threshold phenomenon observed in the SER of human telegraph operators under heat stress [28, Fig. 2], by invoking a thermal internal noise explanation.

References

[1] A. Treves and E. T. Rolls, "Computational analysis of the role of the hippocampus in memory," *Hippocampus*, vol. 4, pp. 374–391, Jun. 1994.

[2] D. A. Wilson and R. M. Sullivan, "Cortical processing of odor objects," *Neuron*, vol. 72, pp. 506–519, Nov. 2011.

[3] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 79, pp. 2554–2558, Apr. 1982.

[4] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, "The capacity of the Hopfield associative memory," *IEEE Trans. Inf. Theory*, vol. IT-33, pp. 461–482, 1987.

[5] D. J. Amit and S. Fusi, "Learning in neural networks with material synapses," *Neural Comput.*, vol. 6, pp. 957–982, Sep. 1994.

[6] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Curr. Opin. Neurobiol.*, vol. 14, pp. 481–487, Aug. 2004.

[7] A. A. Koulakov and D. Rinberg, "Sparse incomplete representations: A potential role of olfactory granule cells," *Neuron*, vol. 72, pp. 124–136, Oct. 2011.

[8] A. H. Salavati and A. Karbasi, "Multi-level error-resilient neural networks," in *Proc. 2012 IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 1064–1068.

[9] A. Karbasi, A. H. Salavati, and A. Shokrollahi, "Iterative learning and denoising in convolutional neural associative memories," in *Proc. 30th Int. Conf. Mach. Learn. (ICML 2013)*, Jun. 2013, pp. 445–453.

[10] N. Brunel, V. Hakim, P. Isope, J.-P. Nadal, and B. Barbour, "Optimal information storage and the distribution of synaptic weights: Perceptron versus Purkinje cell," *Neuron*, vol. 43, pp. 745–757, 2004.

[11] L. R. Varshney, P. J. Sjöström, and D. B. Chklovskii, "Optimal information storage in noisy synapses under resource constraints," *Neuron*, vol. 52, pp. 409–423, Nov. 2006.

[12] C. Koch, *Biophysics of Computation*. New York: Oxford University Press, 1999.

[13] M. D. McDonnell and L. M. Ward, "The benefits of noise in neural systems: bridging theory and experiment," *Nat. Rev. Neurosci.*, vol. 12, pp. 415–426, Jul. 2011.

[14] H. Chen, P. K. Varshney, S. M. Kay, and J. H. Michels, "Theory of the stochastic resonance effect in signal detection: Part I–fixed detectors," *IEEE Trans. Signal Process.*, vol. 55, pp. 3172–3184, Jul. 2007.

[15] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *J. ACM*, vol. 51, pp. 385–463, May 2004.

[16] D. J. Amit, *Modeling Brain Function*. Cambridge: Cambridge University Press, 1992.

[17] M. G. Taylor, "Reliable information storage in memories designed from unreliable components," *Bell Syst. Tech. J.*, vol. 47, pp. 2299–2337, Dec. 1968.

[18] A. V. Kuznetsov, "Information storage in a memory assembled from unreliable components," *Probl. Inf. Transm.*, vol. 9, pp. 100–114, July-Sept. 1973.

[19] L. R. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Trans. Inf. Theory*, vol. 57, pp. 4427–4444, Jul. 2011.

[20] V. Gripon and C. Berrou, "Sparse neural networks with large learning diversity," *IEEE Trans. Neural Netw.*, vol. 22, pp. 1087–1096, Jul. 2011.

[21] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML 2008)*, Jul. 2008, pp. 1096–1103.

[22] Q. V. Le, J. Ngiam, Z. Chen, D. Chia, P. W. Koh, and A. Y. Ng, "Tiled convolutional neural networks," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Cambridge, MA: MIT Press, 2010, pp. 1279–1287.

[23] A. Karbasi, A. H. Salavati, A. Shokrollahi, and L. R. Varshney, "Noise-enhanced associative memories," arXiv, 2013.

[24] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, pp. 569–584, Feb. 2001.

[25] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge: Cambridge University Press, 2008.

[26] M. Yoshida, H. Hayashi, K. Tateno, and S. Ishizuka, "Stochastic resonance in the hippocampal CA3–CA1 model: a possible memory recall mechanism," *Neural Netw.*, vol. 15, pp. 1171–1183, Dec. 2002.

[27] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology," *Neural Comput.*, vol. 10, pp. 1601–1638, Oct. 1998.

[28] N. H. Mackworth, "Effects of heat on wireless telegraphy operators hearing and recording Morse messages," *Br. J. Ind. Med.*, vol. 3, pp. 143–158, Jul. 1946.

6 Proof of Lemma 1

To calculate the probability that a constraint node makes a mistake when there is no external noise, consider constraint node i whose decision parameter will be

$$h_i^{(\ell)} = \left(W^{(\ell)} \cdot x^{(\ell)}\right)_i + v_i = v_i.$$

Therefore, the probability of making a mistake will be

$$\begin{aligned} \pi_0^{(\ell)} &= \Pr\{|v_i| > \psi\} \\ &= \max\left(0, \frac{\nu - \psi}{\nu}\right). \end{aligned} \tag{5}$$

Thus, to make $\pi_0^{(\ell)} = 0$ we will select $\psi > \nu$.[4] So from now on, we assume

$$\pi^{(0)} = 0. \tag{6}$$

Now knowing that the constraint will not send any non-zero messages in absence of external noise, we focus on the pattern neurons in the same circumstance. A given pattern node $x_j^{(\ell)}$ will receive a zero from all its neighbors among the constraint nodes. Therefore, its decision parameter will be $g_j^{(\ell)} = u_j$. As a result, a mistake could happen if $|u_j| \geq \varphi$. The probability of this event is given by

$$\begin{aligned} P_0^{(\ell)} &= \Pr\{|u_i| > \varphi\} \\ &= \max\left(0, \frac{\upsilon - \varphi}{\varphi}\right). \end{aligned} \tag{7}$$

Therefore, to make $P_0^{(\ell)}$ go to zero, we must select $\varphi \geq \upsilon$. As our numerical analysis in Appendix 12 shows, this choice is in harmony with our goal to minimize $P_1^{(\ell)}$ as well.

7 Proof of Theorem 2

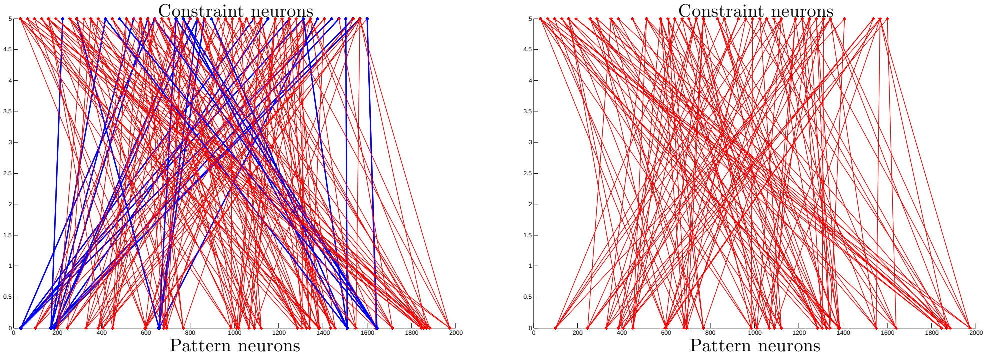
We first show that the noisy neural network can correct any external error pattern that the noiseless counterpart can correct in the limit of $T \to \infty$. The idea is that if the noiseless decoder succeeds, then there is a non-zero probability P that the noisy decoder succeeds in a given round as well (corresponding to the case that noise values are rather small). Since we do not introduce new errors during the application of Alg. 2, the number of errors in the new rounds are smaller than or equal to the previous round, hence, the probability of success is lower bounded by P. If we apply Alg. 2 T times, then the probability of correcting the external errors at the end of round T is $P + P(1-P) + \cdots + P(1-P)^{T-1} = 1 - (1-P)^T$. Since $P > 0$, for $T \to \infty$ this probability tends to 1.

Now, we turn our attention to the cases where the noiseless network fails in eliminating external errors and show that there exist external error patterns, called *stopping sets*, for which the noisy decoder is capable of eliminating them while the noiseless network has failed. Assuming that each cluster can eliminate i external errors in their domain and in absence of internal noise[5], stopping sets correspond to noise pattern in which each cluster has more than i errors. Then Alg. 2 can not proceed any further. However, in the noisy network, there is a chance that in one of the rounds, the noise acts in our favor and the cluster could correct more than i errors. This is reflected in Fig. 2 as well, where the value of P_{c_i} is larger when the network is noisy. In this case, if the probability of getting out of the stopping set is P in each round, for some $P > 0$, then a similar argument to the previous case shows that $P \to 1$ when $T \to \infty$. This concludes the proof.

If the amount of internal or external noise is too high, the noisy architecture will eventually get stuck just like the noiseless network.

[4] Note that this might not be possible in all cases since, as we will see later, the minimum absolute value of network weights should be at least ψ. Therefore, if ψ is too large we might not be able to find a proper set of weights.

[5] From Fig. 2, $i = 2$ for our case.



(a) $t = 0$

(b) $t \to \infty$

Figure 7: An external noise pattern that contains a stopping set in a noiseless neural circuit. Left figure shows tha original pattern and the right figure illustrates the result of the decoding algorithm after sufficient number of iterations where the algorithm gets stuck. Blue pattern nodes are those that are connected to at least one cluster with a single external error. Obviously, the stopping set on the right does not have any blue nodes.

Fig. 7a illustrates an example of a stopping set over the graph $\widetilde{G}$ in our empirical studies. In the figure, only the nodes corrupted with external noise are shown for clarity. Pattern neurons that are connected to at least one cluster with a single error are colored blue and other pattern neurons are colored red. Figure 7b illustrates the same network but after a sufficient number of decoding iterations which results in the algorithm to get stuck. Obviously, we have a stopping set in which no cluster has a single error and the algorithm could not proceed further since $P_{c_i} \simeq 0$ for $i > 1$ in a noiseless architecture. Thus, the external error can not get corrected.

As evident from Fig. 7, the stopping set is the result of clusters not being able to correct more than one external errors. And this is the place where internal noise might come to rescue. Interestingly, an "unreliable" neural circuit in which $\upsilon = .6$ could easily get out of the stopping set shown in Fig. 7b and correct all of the external errors. The reason might be because we try several times to correct errors in a cluster (and overall in the network) while making sure that the algorithm does not introduce new errors itself. Thus, the noise might act in our favor in one of these attempts and the algorithm might be able to avoid stopping set as depicted in Fig. 7.

8 Proof of Theorem 3

The proof is based on the density evolution technique [25]. Without loss of generality, assume that we have P_{c_1}, P_{c_2} and P_{c_3} but the approach can easily be extended if we have P_{c_i} for $i > 3$. Let $\Pi(t)$ be the average probability that a cluster node sends a failure message, i.e. that it can not correct external errors lying in its domain. Then, the probability that a pattern neuron with degree d_i can not be corrected will be the probability that none of its neighboring clusters can correct the error, i.e.

$$P_i = (\Pi(t))^{d_i}$$

Averaging over d_i we find that the average probability of error in iteration t will be

$$z(t+1) = \lambda(\Pi(t)) \tag{8}$$

Now consider a cluster ℓ that contains d_ℓ pattern neurons. This cluster will *not* send a failure message to a noisy pattern neuron in its domain with probability:

1. P_{c_1}, if it is not connected to any other noisy neuron.
2. P_{c_2}, if it is connected to exactly one other constraint neuron.
3. P_{c_3}, if it is connected to exactly two other constraint neurons.
4. 0, if it is connected to more than two other constraint neuron.

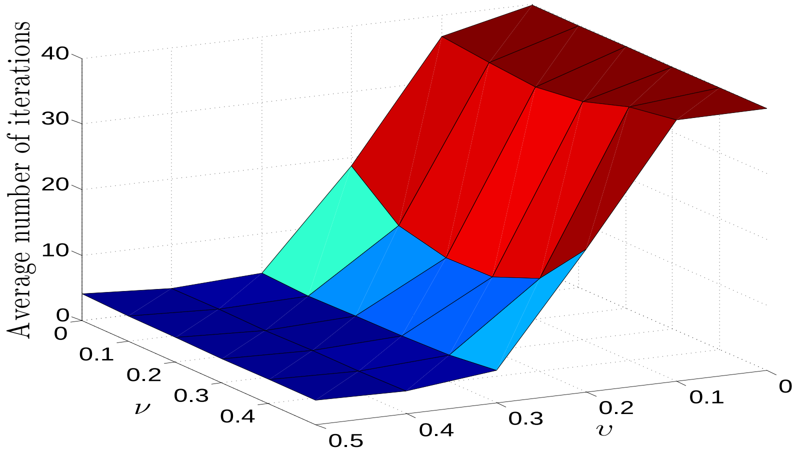


Figure 8: The effect of internal noise on the number of iterations performed by Alg. 2, for different values of υ and ν with $\epsilon = 0.125$.

Thus, we obtain

$$\Pi^{(\ell)}(t) = 1 - P_{c_1}(1-z)^{d_\ell - 1} - P_{c_2}\binom{d_\ell - 1}{1} z(1-z)^{d_\ell - 2} - P_{c_3}\binom{d_\ell - 1}{2} z^2(1-z)^{d_\ell - 3}.$$

Averaging over d_ℓ we obtain

$$\Pi(t) = \mathbb{E}_{d_\ell}\left(\Pi^{(\ell)}(t)\right) = 1 - P_{c_1}\rho(1-z(t)) - P_{c_2}z\rho'(1-z(t)) - 0.5P_{c_2}z^2\rho"(1-z(t)), \quad (9)$$

where $\rho'(x)$ and $\rho''(x)$ are derivatives of the function $\rho(x)$ with respect to x.

Equations (8) and (9) will give us the value of $z(t+1)$ as a function of $z(t)$. We can then calculate the final BER as $z(\infty)$. For $z(\infty) \to 0$, it is sufficient to have $z(t+1) < z(t)$, which proves the theorem.

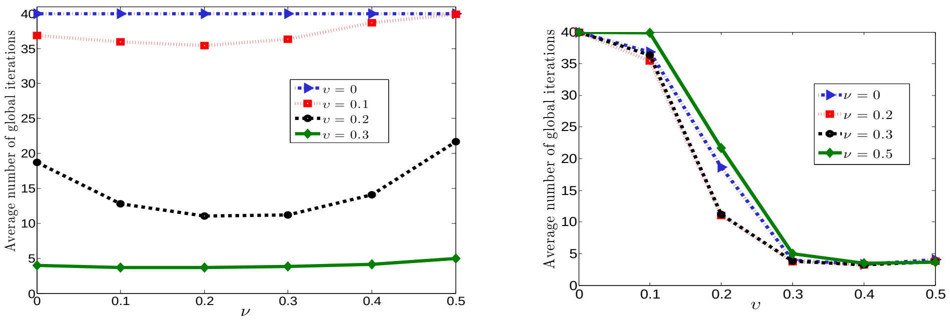
9 More Results on Recall Time as a function of Internal Noise

Fig. 8 illustrates the number of iterations performed by Alg. 2 for correcting the external errors when ϵ is fixed to 0.125. We choose not to run the algorithm beyond 40 iterations, as the algorithm can typically correctly denoise the pattern with less than 40 iterations or it fails. Thus, the corresponding areas in the figure where the number of iterations reaches 40 indicate decoding failure. Figs. 9b and 9a are projected versions of Fig. 8 and show the average number of iterations as a function of υ and ν, respectively.

Surprisingly, the amount of internal noise drastically affects the speed of Alg. 2. Internal noise in pattern neurons is necessary for Alg. 2 to be successful, as shown in Fig. 9a. Notice the curve with $\upsilon = 0$ which indicates that Alg. 2 is never successful in denoising the pattern while by increasing υ the denoising process is completed in less than 40 iterations . The same figure also shows that there is an optimal value $\nu^* = 0.25$ for which the number of iterations is minimized. Furthermore, a close inspection of Fig. 9b reveals the same phenomenon, i.e., around $\upsilon^* = 0.4$, the number of iterations reaches its minimum.

10 Effect of internal noise on the performance of the neural network in absence of external noise

At this point, we are interested in investigating the effects of internal noise on a particular network without external noise. More specifically, we assume $\epsilon = 0$ but $\upsilon > 0$, $\nu > 0$. Furthermore, in contrast to what we have considered so far, we select $\varphi < \upsilon$. Therefore, even if there is no



(a) Effect of internal noise at constraint neurons side. (b) Effect of internal noise at pattern neurons side.

Figure 9: The effect of internal noise on the number of iterations performed by Alg. 2, for different values of υ and ν with $\epsilon = 0.125$. The average iteration number of 40 indicate the failure of Alg. 2.

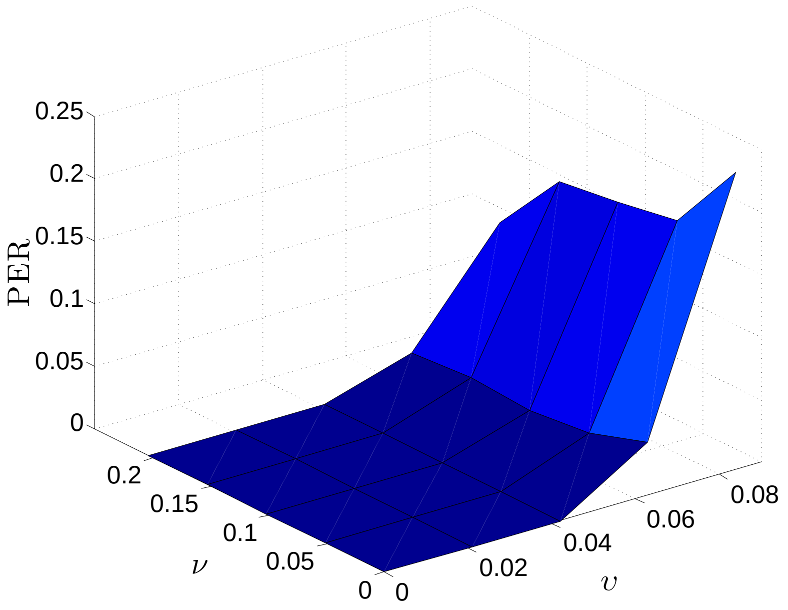
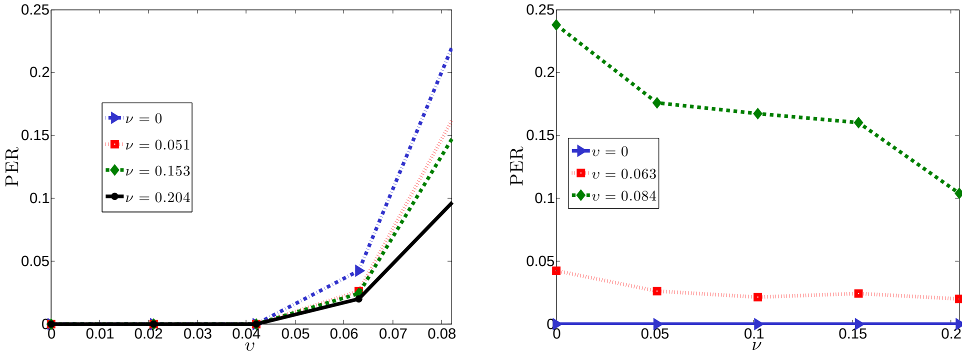


Figure 10: The effect of the internal noise on final PER as a function of υ and ν in absence of external noise.

external noise, the pattern neurons could be corrupted due to internal noise parameter υ. With abuse of notation, we assume the pattern neurons to be corrupted with a ± 1 noise added to them with probability υ. The rest of the scenario is the same as before. Fig. 10 illustrates the effect of the internal noise as a function of υ and ν, the noise parameters at the pattern and constraint nodes, respectively.

As evident from the figure, the higher the noise is on the pattern side, the higher the pattern error rate (PER) will be, i.e. the performance deteriorates as a function of internal noise. This behavior is shown in Figs. 11a and 11b for better inspection.

Interestingly, this behavior is very similar to the effect of heat stress on the performance of wireless telegraphy operators [28, Fig. 2]. The two phenomena might be related because external heat will translate into neurons with more internal thermal noise.



(a) Effect of internal noise at pattern neurons side. (b) Effect of internal noise at constraint neurons side.

Figure 11: The effect of the internal noise on final PER as a function of υ and ν in absence of external noise.

11 Estimating P_{c_1} Theoretically

To bound P_{c_1}, consider four event probabilities for a cluster:

- $\pi_0^{(\ell)}$ (resp. $P_0^{(\ell)}$): The probability that a constraint neuron (resp. pattern neuron) in cluster ℓ makes a wrong decision due to its internal noise when there is no external noise introduced to cluster ℓ, i.e. $\|z^{(\ell)}\|_0 = 0$.
- $\pi_1^{(\ell)}$ (resp. $P_1^{(\ell)}$): The probability that a constraint neuron (resp. pattern neuron) in cluster ℓ makes a wrong decision due to its internal noise when one input error (external noise) is introduced, i.e. $\|z^{(\ell)}\|_0 = \|z^{(\ell)}\|_1 = 1$.

Notice $P_{c_1}^{(\ell)} = 1 - P_1^{(\ell)}$.

We derive an upper bound on the probability a constraint node makes a mistake in the presence of one external error.

Lemma 5. *In the presence of a single external error, the probability that a constraint neuron makes a wrong decision due to its internal noise is given by*

$$\pi_1^{(\ell)} \leq \max\left(0, \frac{\nu - (\eta - \psi)}{2\nu}\right),$$

where $\eta = \min_{i,j,W_{ij}\neq 0}(|W_{ij}|)$ is the minimum absolute value of the non-zero weights in the neural graph and is chosen such that $\eta \geq \psi$.[6]

Proof. Without loss of generality, we assume it is the first pattern node, $x_1^{(\ell)}$, that is corrupted with noise whose value is $+1$. Now we would like to calculate the probability that a constraint node makes a mistake in such circumstances. Furthermore, we will only the constraint neurons that are connected to $x_1^{(\ell)}$. Because for the other constraint neurons, the situation is the same as the previous cases where there were no external noise.

for a constraint neuron j that is connected to $x_1^{(\ell)}$, the decision parameter is

$$\begin{aligned} h_j^{(\ell)} &= \left(W^{(\ell)}.(x^{(\ell)} + z^{(\ell)})\right)_j + v_j \\ &= 0 + \left(W^{(\ell)}.z^{(\ell)}\right)_j + v_j \\ &= w_{j1} + v_j. \end{aligned}$$

[6] This condition can be enforced during simulations as long as ψ is not too large, which itself is determined by the level of constraint neuron internal noise, ν, as we must have $\psi \geq \nu$.

We consider two error events:

1. A constraint node j makes a mistake and does not send a message at all. The probability of this event is denoted by $\pi_1^{(1)}$.
2. A constraint node j makes a mistake and sends a message with the opposite sign. The probability of this event is denoted by $\pi_2^{(1)}$.

We first calculate the probability of $\pi_2^{(1)}$. Without loss of generality, assume the $w_{j1} > 0$ so that the probability of an error of type two is as follows (the case for $w_{j1} < 0$ is exactly the same):

$$\begin{aligned}\pi_2^{(1)} &= \text{Pr}\{w_{ji} + v_j < -\psi\} \\ &= \max\left(0, \frac{\nu - (\psi + w_{j1})}{2\nu}\right). \end{aligned} \tag{10}$$

However, since $\psi > \nu$ and $w_{j1} > 0$, then $\nu - (\psi + w_{j1}) < 0$ and $\pi_2^{(1)} = 0$. Therefore, the constraint neurons will never send a message that has an opposite sign to what it should have. All that remains to do is to calculate the probability that they remain silent by mistake.

To this end, we will have

$$\begin{aligned}\pi_1^{(1)} &= \text{Pr}\{|w_{ji} + v_j| < \psi\} \\ &= \max\left(0, \frac{\nu + \min(\psi - w_{j1}, \nu)}{2\nu}\right). \end{aligned} \tag{11}$$

The above equation can be simplified if we assume that the absolute value of all weights in the network is bigger than a constant $\eta > \psi$. Then, the above equation will simplify to

$$\pi_1^{(1)} \leq \max\left(0, \frac{\nu - (\eta - \psi)}{2\nu}\right). \tag{12}$$

Putting the above equations together, we obtain:

$$\pi^{(1)} \leq \max\left(0, \frac{\nu - (\eta - \psi)}{2\nu}\right). \tag{13}$$

□

In the case $\eta - \psi > \nu$, we could even manage to make this probability equal to zero. However, we will leave it as is and use (13) to calculate $P_1^{(\ell)}$.

11.1 Calculating $P_1^{(\ell)}$

We start by first calculating the probability that a non-corrupted pattern node $x_j^{(\ell)}$ makes a mistake, which is to change its state in round 1. Let us denote this probability by $q_1^{(\ell)}$. Now to calculate $q_1^{(\ell)}$ assume $x_j^{(\ell)}$ has degree d_j and it has b common neighbors with $x_1^{(\ell)}$, the corrupted pattern node.

Out of these b common neighbors, b_c will send ± 1 messages and the others will, mistakenly, send nothing. Thus, the decision making parameter of pattern node j, $g_j^{(\ell)}$, will be bounded by

$$g_j^{(\ell)} = \frac{\left(\text{sign}(W^{(\ell)})^\top \cdot y^{(\ell)}\right)_j}{d_j} + u_j. \leq \frac{b_c}{d_j} + u_j.$$

We will denote $\left(\text{sign}(W^{(\ell)})^\top \cdot y^{(\ell)}\right)_j$ by o_j for brevity from this point on.

In this circumstances, a mistake happens when $|g_j^{(\ell)}| \geq \varphi$. Thus

$$\begin{aligned}q_1^{(\ell)} &= \text{Pr}\{|g_j^{(\ell)}| \geq \varphi | \deg(a_j) = d_j \& |\mathcal{N}(x_1) \cap \mathcal{N}(a_j)| = a\} \\ &= \text{Pr}\{\frac{o_j}{d_j} + u_j \geq \varphi\} + \text{Pr}\{\frac{o_j}{d_j} + u_j \leq -\varphi\}, \end{aligned} \tag{14}$$

where $\mathcal{N}(a_i)$ represents the neighborhood of pattern node a_i among constraint nodes.

By simplifying (14) we get

$$q_1^{(\ell)}(o_j) = \begin{cases} +1, & \text{if } |o_j| \geq (\upsilon + \varphi)d_j \\ \max(0, \frac{\upsilon-\varphi}{\upsilon}), & \text{if } |o_j| \leq |\upsilon - \varphi|d_j \\ \frac{\upsilon-(\varphi-o_j/d_j)}{2\upsilon}, & \text{if } |o_j - \varphi d_j| \leq \upsilon d_j \\ \frac{\upsilon-(\varphi+o_j/d_j)}{2\upsilon}, & \text{if } |o_j + \varphi d_j| \leq \upsilon d_j. \end{cases}$$

We now average this equation over o_j, b_c, b and d_j. To start, suppose out of the b_c non-zero messages the node a_j receives, e of them have the same sign as the link they are being transmitted over. Thus, we will have $o_j = e - (b_c - e) = 2e - b_c$. Assuming the probability of having the same sign for each message is $1/2$, the probability of having e equal signs out of b_c elements will be $\binom{b_c}{e}(1/2)^{b_c}$. Thus, we will get

$$\bar{q}_1^{(\ell)} = \sum_{e=0}^{b_c} \binom{b_c}{e} (1/2)^{b_c} q_1^{(\ell)}(2e - b_c). \tag{15}$$

Now note that the probability of having $a - b_c$ mistakes from the constraint side is given by $\binom{b}{b_c}(\pi_1^{(\ell)})^{b-b_c}(1-\pi_1^{(\ell)})^{b_c}$. With some abuse of notations we get:

$$\bar{q}_1^{(\ell)} = \sum_{b_c=0}^{b} \binom{b}{b_c} (\pi_1^{(\ell)})^{b-b_c}(1-\pi_1^{(\ell)})^{b_c} \sum_{e=0}^{b_c} \binom{b_c}{e} (1/2)^{b_c} q_1^{(\ell)}(2e - b_c). \tag{16}$$

Finally, the probability that a_j and x_1 have b common neighbors can be approximated by $\binom{d_j}{b}(1 - \bar{d}^{(\ell)}/m_\ell)^{d_j-b}(\bar{d}^{(\ell)}/m_\ell)^b$, where $\bar{d}^{(\ell)}$ is the average degree of pattern nodes. Thus (again abusing some notation), we obtain:

$$\bar{q}_1^{(\ell)} = \sum_{b=0}^{d_j} p_b \sum_{b_c=0}^{b} p_{b_c} \sum_{e=0}^{b_c} \binom{b_c}{e} (1/2)^{b_c} q_1^{(\ell)}(2e - b_c), \tag{17}$$

where $q_1^{(\ell)}(2e - b_c)$ is given by (14), p_b is the probability of having b common neighbors and is estimated by $\binom{d_j}{b}(1 - \bar{d}^{(\ell)}/m_\ell)^{d_j-b}(\bar{d}^{(\ell)}/m_\ell)^b$, with $\bar{d}^{(\ell)}$ being the average degree of pattern nodes in cluster ℓ. Furthermore, p_{b_c} is the probability of having $b - b_c$ out of these b nodes making mistakes. Hence, $p_{b_c} = \binom{b}{b_c}(\pi_1^{(\ell)})^{b-b_c}(1-\pi_1^{(\ell)})^{b_c}$. We will not simplify the above equation any further and use it as it is in our numerical analysis in order to obtain the best parameter φ.

Now we will turn our attention to the probability that the corrupted node, x_1, makes a mistake, which is either not to update at all or update its itself in the wrong direction. Recalling that we have assume the external noise term in x_1 to be a $+1$ noise, the wrong direction would be for node x_1 to increase its current value instead of decreasing it. Furthermore, we assume that out of d_1 neighbors of x_1, some j of them have made a mistake and will not send any messages to x_1. Thus, the decision parameter of x_1, will be $g_1^{(\ell)} = u + (d_1 - j)/d_1$. Denoting the probability of making a mistake at x_1 by $q_2^{(\ell)}$ we will get

$$\begin{aligned} q_2^{(\ell)} &= \Pr\{g_1^{(\ell)} \leq \varphi | \deg(x_1) = d_1 \text{ and } j \text{ errors in constraints}\} \\ &= \Pr\{\frac{d_1 - j}{d_1} + u < \varphi\}, \end{aligned} \tag{18}$$

which simplifies to

$$q_2^{(\ell)}(j) = \begin{cases} +1, & \text{if } |j| \geq (1 + \upsilon - \varphi)d_1 \\ \max(0, \frac{\upsilon-\varphi}{\upsilon}), & \text{if } |j| \leq (1 - \upsilon - \varphi)d_1 \\ \frac{\upsilon+\varphi-(d_1-j)/d_1}{2\upsilon}, & \text{if } |\varphi d_1 - (d_1 - j)| \leq \upsilon d_1. \end{cases} \tag{19}$$

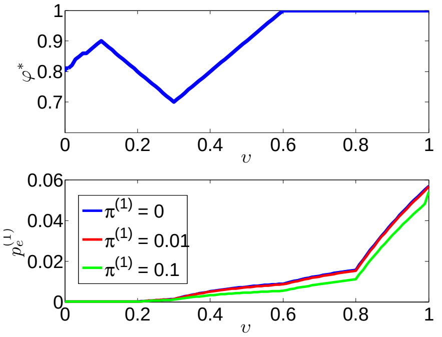


Figure 12: The behavior of $P_1^{(\ell)}$ as a function of φ^* for different values of noise parameter, υ

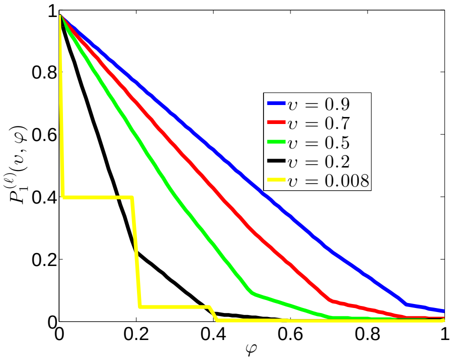


Figure 13: The behavior of $P_1^{(\ell)}$ as a function of φ for different values of noise parameter, υ. Here, we have $\pi_1^{(\ell)} = 0.01$.

Noting that the probability of making j mistakes on the constraint side is $\binom{d_1}{j}(\pi_1^{(\ell)})^j(1-\pi_1^{(\ell)})^{d_1-j}$, we get

$$\bar{q}_2^{(\ell)} = \sum_{j=0}^{d_1} \binom{d_1}{j} (\pi_1^{(\ell)})^j (1-\pi_1^{(\ell)})^{d_1-j} q_2^{(\ell)}(j), \tag{20}$$

where $q_2^{(\ell)}(j)$ is given by (19).

Putting the above results together, the overall probability of making a mistake on the side of pattern neurons when we have one bit of external noise is given by

$$P_1^{(\ell)} = \frac{1}{n^{(\ell)}} \bar{q}_2^{(\ell)} + \frac{n^{(\ell)} - 1}{n^{(\ell)}} \bar{q}_1^{(\ell)}. \tag{21}$$

Finally, the probability that cluster ℓ could correct one error is that all neurons take the correct decision, i.e.

$$P_{c_1}^{(\ell)} = (1 - P_1^{(\ell)})^{n^{(\ell)}}$$

and the average probability that clusters could correct one error is simply

$$P_{c_1} = \mathbb{E}_\ell(P_{c_1}^{(\ell)}). \tag{22}$$

We will use this equation in order to find the best update threshold φ.

12 Choosing proper φ

We now apply numerical methods to (21) to find the best φ for different values of noise parameter, υ. The following figures show the best choice for the parameter φ. The update threshold on the constraint side is chosen such that $\psi > \nu$. In each figure, we have illustrated the final probability of making a mistake, $P_1^{(\ell)}$, for comparison.

Fig. 13 illustrates the behavior of the error probability as a function of φ for different values of υ and for $\pi_1^{(\ell)} = 0.02$. The interesting trend here is that in all cases, φ^*, the update threshold that gives the best result, is chosen such that it is quite large. This actually is in line with our expectation because a small φ will results in non-corrupted nodes to update their states more frequently. On the other hand, a very large φ will prevent the corrupted nodes to correct their states, specially if there are some mistakes made on the constraint side, i.e. $\pi_1^{(\ell)} > 0$. Therefore, since we have much more non-corrupted nodes to the corrupted nodes, it is best to choose a rather high φ but not too high.

Please also note that when $\pi_1^{(\ell)}$ is very high, there are no values of υ for which error-free storage is possible.

Fig. 13 illustrates the behavior of the error probability as a function of φ for different values of υ and for $\pi_1^{(\ell)} = 0.01$. As evident from the figure, choosing a larger φ results in smaller error probability. Moreover, in all cases we have $\varphi^* > \upsilon$. We use this choice, which also makes $P_0^{(\ell)} = 0$.