Neural Networks Built from Unreliable Components

Amin Karbasi[1], Amir Hesam Salavati[1], Amin Shokrollahi[1], and Lav R. Varshney[2]
[1]School of Computer and Communication Sciences, Ecole Polyetechnique Federale de Lausanne, Switzerland
[2]IBM Thomas J. Watson Research Center, USA

***Abstract*—Recent advances in associative memory design through strutured pattern sets and graph-based inference algorithms have allowed the reliable learning and retrieval of an exponential number of patterns. Both these and classical associative memories, however, have assumed internally noiseless computational nodes. This paper considers the setting when internal computations are also noisy. Even if all components are noisy, the final error probability in recall can often be made exceedingly small, as we characterize. There is a threshold phenomenon. We also show how to optimize inference algorithm parameters when knowing statistical properties of internal noise.**

I. Introduction

Associative memories are neural networks that have great promise for their ability to learn patterns from presented inputs, store a large number of patterns, and retrieve them reliably in the face of noisy queries [1]–[3]. In particular, associative memories are designed to memorize a set of given patterns, so that later, corrupted versions of the memorized patterns may be presented and the correct memorized pattern retrieved.

Although information storage and retrieval systems are just communication systems between the present and the future and seemingly fall naturally into the information-theoretic framework where an exponential number of messages can be communicated reliably using a linear number of symbols [4], classical associative memories could only store a linear number of patterns with a linear number of symbols [2].

A primary shortcoming of classical associative memories had been their requirement of memorizing a set of randomly chosen patterns. By enforcing structure and redundancy in the possible set of memorizable patterns—much like natural stimuli [5] and like codewords in error-control codes—new advances in associative memory design have allowed storage of an exponential number of patterns with a linear number of symbols [6], [7], just like in communication systems.

Since people have strong abilities in learning, storing, and reliably retrieving patterns [8], one might wonder if the human brain is operating close to information-theoretic limits and whether it uses associative memory. Indeed, both information-theoretic and associative memory models of storage have been used in neuroscience to predict experimentally measurable properties of synapses in the mammalian brain [9], [10].

Contrary to the fact that noise is present in computational operations of the brain [11], both classical and modern associative memory models assume no internal noise in the computational nodes [1], [7]. The purpose of the present paper is to include internal noise into models of associative memories and study whether they are still able to operate reliably.

We revisit a multi-level, graph code-based, associative memory model [7] and find that even if all components are noisy, the final error probability in recall can be made exceedingly small, as we characterize. There is a threshold phenomenon. We also show how to optimize algorithm parameters when knowing statistical properties of internal noise.

Reliably storing information in memory systems constructed completely from unreliable components is a classical problem in fault-tolerant computing [12]–[14], where achievability schemes have essentially used random access memory architectures with sequential correcting networks. Although direct comparison is difficult since notions of circuit complexity are slightly different, our work also demonstrates that associative memory architectures can store information reliably despite being constructed from unreliable components.

II. Problem Setting and Notation

In our model, a neural associative memory is represented by a weighted bipartite graph G with n pattern nodes, $x_1, x_2, \dots, x_n$, and m constraint nodes, $y_1, y_2, \dots, y_m$. Graph G is composed of L clusters $G^{(1)}, G^{(2)}, \dots, G^{(L)}$, each of which is again a bipartite graph. More specifically, cluster $G^{(i)}$ consists of n_i pattern nodes $x_1^{(i)}, x_2^{(i)}, \dots, x_{n_i}^{(i)}$, and m_i constraint nodes $y_1^{(i)}, y_2^{(i)}, \dots, y_{m_i}^{(i)}$. The edge weight matrix W of the graph G is chosen such that

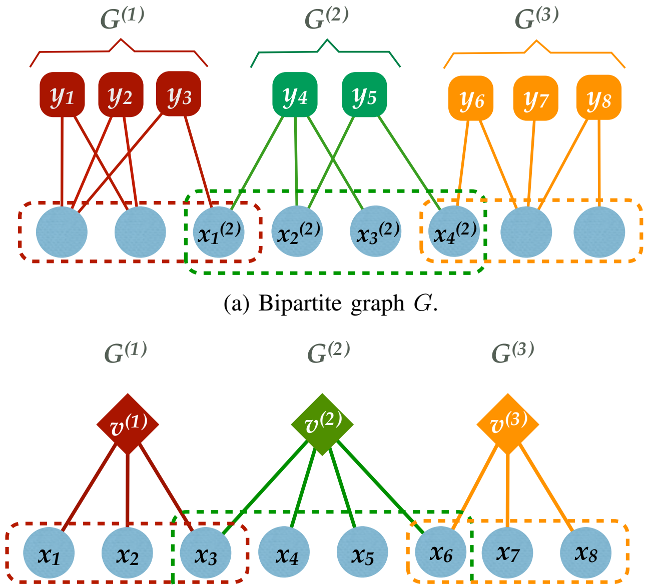
$$W \cdot x = 0, \text{ for all } x \in \mathcal{X}, \tag{1}$$

where $\mathcal{X}$ is the database of $\mathcal{C}$ patterns x of length n. The matrix W can, for instance, be determined by applying a learning rule to $\mathcal{X}$, cf. [7]. Equation (1) can be written equivalently as $W^{(i)} \cdot x^{(i)} = 0$, where $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_{n_i}^{(i)})$ denotes the ith subpattern and $W^{(i)}$ denotes the weight matrix of cluster $G^{(i)}$. Note that due to overlaps, a pattern node can be a member of multiple subpatterns, as shown in Figure 1a. This property, together with the constraints imposed by (1), helps in recalling the memorized patterns $\mathcal{X}$ even in the presence of noise.

We assume that pattern elements are non-negative integers as they simply represent the firing rates of neurons.

The goal of each cluster is to be able to correct one input error. To this end, an iterative decoding procedure is performed. In contrast to message-passing decoding of LDPC codes, messages on outgoing links of a pattern/constraint node are all the same: the same message is broadcast to all neighbors since neurons do not distinguish between their neighbors.

With slight abuse of notation, let us denote the messages transmitted by pattern node i and constraint node j at round t with $x_i(t)$ and $y_j(t)$, respectively. In round 0, the pattern



(a) Bipartite graph G.

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

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

nodes are initialized by a pattern $\hat{x}$ sampled from the dataset $\mathcal{X}$, plus a noise vector z, i.e., $x(0) = \hat{x} + z$. We further define $x^{(\ell)}(0) = \hat{x}^{(\ell)} + z^{(\ell)}$, where $z^{(\ell)}$ is the realization of noise on subpattern $x^{(\ell)}$. In this work, we restrict $z_i \in \{-1, 0, 1\}$.

In round t, the pattern and constraint neurons update their states based on feedback from neighbors. However, neuronal computation is faulty and so neuron decisions are not always reliable. The decision making criteria for pattern node i in cluster ℓ is

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

where φ is the update threshold and g_i is given by

$$g_i^{(\ell)}(t) = \frac{\left((\text{sign}(W^{(\ell)})^\top \cdot y^{(\ell)}(t)\right)_i}{d_i} + u_i. \tag{3}$$

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. Herein, we consider a bounded noise model such that u_i is uniformly distributed in the interval $[-\upsilon, \upsilon]$, for some $\upsilon < 1$.

On the constraint side, the update rule is

$$y_i^{(\ell)}(t) = f(h_i^{(\ell)}(t), \psi) = \begin{cases} +1, & \text{if } h_i^{(\ell)}(t) \geq \psi \\ 0, & \text{if } -\psi \leq h_i^{(\ell)}(t) \leq \psi \\ -1, & \text{o.w.,} \end{cases} \tag{4}$$

where ψ is the update threshold and $h_i^{(\ell)}$ is defined as

$$h_i^{(\ell)}(t) = \left(W^{(\ell)} \cdot x^{(\ell)}(t)\right)_i + v_i, \tag{5}$$

in which $x^{(\ell)}(t) = [x_1^{(\ell)}(t), \dots, x_{n_\ell}^{(\ell)}(t)]$ 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's such they are uniformly distributed in the interval $[-\nu, \nu]$ for some $\nu < 1$.

Algorithm 1 Intra-Module Error Correction

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

For our asymptotic analysis, we need to define the degree distribution associated with a bipartite graph from an *edge perspective*. To this end, we define $\lambda(z) = \sum_j \lambda_j z^{j-1}$ and $\rho(z) = \sum_j \rho_j z^{j-1}$ where λ_j (resp., ρ_j) equals the fraction of edges that connect to pattern (resp., constraint) nodes of degree j. Similarly, denote by $\lambda^{(i)}$ and $\rho^{(i)}$ the pattern/constraint degree distributions of cluster $G^{(i)}$ from the edge perspective.

III. Main Results

Building on the (noisy) update rules presented in the previous section, we use a combination of Alg. 1 and Alg. 2 to deal with the internal and external noise in recall; these algorithms are modified from [7] to account for unreliable computations.

Alg. 1 aims at canceling the effect of internal noise and correcting a single external error within a cluster by a series of backward and forward iterations. The messages transmitted by pattern neuron j and cluster neuron i in cluster ℓ are represented by $y_i^{(\ell)}$ and $x_j^{(\ell)}$, respectively. We let P_c represent the average probability that a cluster can successfully correct one external error.

Since clusters overlap, they can help each other in resolving external errors. This is done by Alg. 2. The following theorem gives a simple condition under which Alg. 2 can correct a linear fraction of external errors (in terms of n) with an exceedingly small error probability. The condition involves $\tilde{\lambda}$ and $\tilde{\rho}$, the degree distributions of the contracted graph $\widetilde{G}$ defined as follows. For each cluster $G^{(\ell)}$ we contract its set of constraint nodes into a single node $v^{(\ell)}$ (see Fig. 1b).

Theorem 1: 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 $\epsilon\widetilde{\lambda}(1 - P_c \cdot \widetilde{\rho}(1 - z)) < z$ for $z \in (0, \epsilon)$.

Algorithm 2 Sequential Peeling Algorithm [7]

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

Proof: The complete proof can be found in the Appendix. In short, let a cluster receives an error message from its neighboring pattern nodes with probability z. Consider a given *noisy* pattern neuron that is connected to cluster $v^{(\ell)}$. Moreover, let $\Pi^{(\ell)}(t)$ denote the probability that the cluster node $v^{(\ell)}$ with degree $\widetilde{d}_\ell$ sends an error message in iteration t of Alg. 2.

This happens if either the node $v^{(\ell)}$ is connected to more than one noisy pattern neuron (in which case it sends out an error message with probability one), or the node $v^{(\ell)}$ does not receive any error message from its neighbors (in which case it sends out an error message with probability at most $P_1^{(\ell)} = 1 - P_c$). Hence, $\Pi^{(\ell)}(t) = 1 - P_c(1 - z(t))^{\widetilde{d}_\ell - 1}$.

Now, let $\Pi(t)$ represent the average probability a cluster node sends a message declaring at least one of its constraint neurons is violated. Thus we get $\Pi(t) = 1 - P_c \cdot \widetilde{\rho}(1 - z(t))$. Then, a given pattern neuron x_i with degree d_i will remain noisy in iteration $t+1$ of Alg. 2 if it was noisy in the first place; in iteration $t+1$ all of its neighbors among constraint neurons will send a violation message. Therefore, the probability of this node being noisy will be $z(0)\pi(t)^{d_i}$.

As a result, noting that $z(0) = \epsilon$, the average probability that a pattern neurons remains noisy will be

$$z(t+1) = \epsilon \sum_i \widetilde{\lambda}_i (\Pi(t))^i = \epsilon \widetilde{\lambda}(1 - P_c \widetilde{\rho}(1 - z(t))) \qquad (6)$$

Therefore, the decoding operation will be successful if $z(t+1) < z(t)$, $\forall t$. As a result, we must look for the maximum ϵ such that $\epsilon \widetilde{\lambda}(1 - P_c \widetilde{\rho}(1 - z)) < z$ for $z \in [0, \epsilon]$. ■

Thm. 1 provides insight on the role of P_c: if it is equal to 1, we get the same result as the noise-free case [7]. However, as P_c moves towards 0, the value of $z(t)$ grows towards ϵ, which means we can not correct any input error.

Further note that once $P_c < 1$, $z(t)$ will be bounded away from 0 because $\widetilde{\lambda}(x)$ is an increasing function of x. Hence, for $1 - P_c \widetilde{\rho}(1 - z) \geq 1 - P_c$ we have $z(t+1) \geq \widetilde{\lambda}(1 - P_c)$. Fig. 2 illustrates how $z - \epsilon \widetilde{\lambda}(1 - P_c \widetilde{\rho}(1 - z))$ behaves as a function of z for different values of P_c. Reliable storage occurs when the expression is negative.

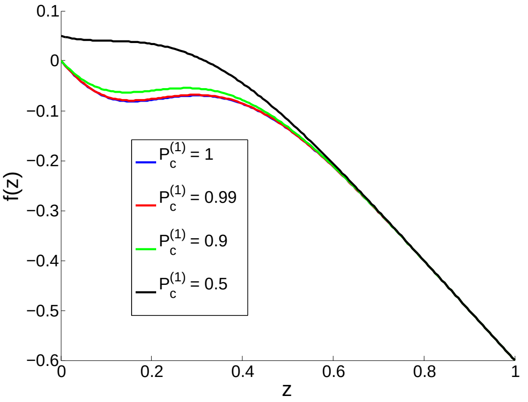


Fig. 2: The behavior of $f(z) = z - \epsilon \widetilde{\lambda}(1 - P_c \widetilde{\rho}(1 - z))$ as a function of z and for different values of P_c. In all cases, $\epsilon = 0.1$.

A. Estimating P_c

To bound P_c, 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 $P_c^{(\ell)} = 1 - P_1^{(\ell)}$.

The following lemma, with proof in the Appendix, shows that when update thresholds φ and ψ are chosen properly, the probability of making a mistake in the absence of external noise tends to zero.

Lemma 2: In absence of external noise, the probability that a constraint neuron (resp. pattern neuron) makes a wrong decision due to its internal noise is given by

$$\pi_0^{(\ell)} = \max\left(0, \frac{\nu - \psi}{\nu}\right), \qquad (7)$$

$$P_0^{(\ell)} = \max\left(0, \frac{\upsilon - \varphi}{\upsilon}\right), \qquad (8)$$

which will be 0 for $\psi \geq \nu$ (resp. $\varphi \geq \upsilon$).

Next, we derive an upper bound on the probability a constraint node makes a mistake in the presence of one external error; proof is given in the Appendix.

Lemma 3: In 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 $\eta \geq \psi$.[1]

[1] 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 $\psi \geq \nu$.

Finally, we obtain an upper bound on $P_1^{(\ell)}$. For brevity, we leave details to the Appendix. Briefly, let us assume w.l.o.g. that the first node $x_1^{(\ell)}$ is the one corrupted with noise $+1$.

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

Out of these b common neighbors, b_c will send ± 1 messages and the others will mistakenly send nothing. Let o_j denote $\left(\text{sign}(W^{(\ell)})^\top \cdot y^{(\ell)}\right)_j$. Then, we have

$$q_1^{(\ell)}(o_j) = \begin{cases} +1, & \text{if } |o_j| \geq (\upsilon + \varphi)d_j \\ \max(0, \frac{\upsilon-\varphi}{\upsilon}), & \text{if } |o_j| \leq |\upsilon - \varphi| d_j \\ \frac{\upsilon - (\varphi - o_j/d_j)}{2\upsilon}, & \text{if } |o_j - \varphi d_j| \leq \upsilon d_j \\ \frac{\upsilon - (\varphi + o_j/d_j)}{2\upsilon}, & \text{if } |o_j + \varphi d_j| \leq \upsilon d_j. \end{cases} \tag{9}$$

We now average the above equation over o_j, b_c and b, yielding:

$$\bar{q}_1^{(\ell)} = \sum_{b=0}^{d_j} p_b \sum_{b_c=0}^{b} p_{b_c} \sum_{e=0}^{b_c} \binom{b_c}{e} (1/2)^{b_c} q_1^{(\ell)}(2e - b_c), \tag{10}$$

where $q_1^{(\ell)}(2e - b_c)$ is given by (9), p_b is the probability of having b common neighbors and is estimated by $\binom{d_j}{b}(1 - \bar{d}^{(\ell)}/m_\ell)^{d_j - a}(\bar{d}^{(\ell)}/m_\ell)^b$, 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, $p_{b_c} = \binom{b}{b_c}(\pi_1^{(\ell)})^{a-b_c}(1 - \pi_1^{(\ell)})^{b_c}$.

Now we turn our attention to the probability the corrupted node x_1 makes a mistake: either not updating or updating in the wrong direction. Recall we had assumed the external noise for x_1 is $+1$, and so the wrong direction for node x_1 is increasing its current value instead of decreasing it. Furthermore, we had assumed that out of d_1 neighbors of x_1, j of them have made mistakes and will not send any messages to x_1. Thus, the decision parameter of x_1 will be $g_1^{(\ell)} = u + (d_1 - j)/d_1$. Letting the probability of making a mistake in this situation be $q_2^{(\ell)}$,

$$q_2^{(\ell)} = \Pr\left\{\frac{d_1 - j}{d_1} + u < \varphi\right\}, \tag{11}$$

which can be simplified to:

$$q_2^{(\ell)}(j) = \begin{cases} +1, & \text{if } |j| \geq (1 + \upsilon - \varphi)d_1 \\ \max(0, \frac{\upsilon-\varphi}{\upsilon}), & \text{if } |j| \leq (1 - \upsilon - \varphi)d_1 \\ \frac{\upsilon + \varphi - (d_1 - j)/d_1}{2\upsilon}, & \text{if } |\varphi d_1 - (d_1 - j)| \leq \upsilon d_1. \end{cases} \tag{12}$$

Noting 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

$$\bar{q}_2^{(\ell)} = \sum_{j=0}^{d_1} \binom{d_1}{j} (\pi_1^{(\ell)})^j (1 - \pi_1^{(\ell)})^{d_1 - j} q_2^{(\ell)}(j), \tag{13}$$

where $q_2^{(\ell)}(j)$ is given by (12).

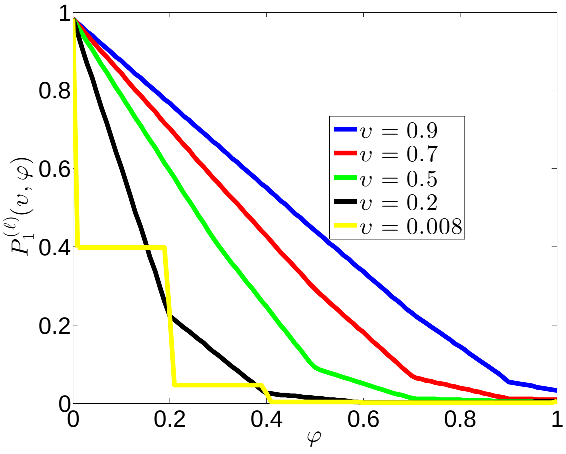


Fig. 3: The behavior of $P_1^{(\ell)}$ as a function of φ for different values of noise parameter, υ. Here, we have $\pi_1^{(\ell)} = 0.01$.

Putting things together, the overall probability a pattern neuron makes a mistake with one bit of external noise is:

$$P_1^{(\ell)} = \frac{1}{n^{(\ell)}} \bar{q}_2^{(\ell)} + \frac{n^{(\ell)} - 1}{n^{(\ell)}} \bar{q}_1^{(\ell)}. \tag{14}$$

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

B. *Choosing the best φ*

We use numerical methods applied to (14) to find the best φ, ensuring tight results. Loose bounds on $P_1^{(\ell)}$ allow an analytical approximation to the best φ, as given in the Appendix. Fig. 3 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 $P_0^{(\ell)} = 0$.

IV. Simulations

To investigate the performance of the proposed algorithm in dealing with external noise, we have used the a modified version of the learning algorithm proposed in [7], in order to account for the internal noise affecting the neurons.

We have considered a network of $n = 400$ pattern neurons with $L = 50$ clusters and on average 40 pattern and 20 constraint nodes in each cluster. Similar to [7], the external noise is modelled by randomly choosing a pattern node with probability p_e and corrupting it with an additive ± 1 noise. At this point, Alg. 2 is utilized to eliminate the external noise. Once finished, we calculate the bit error rate (BER) by counting the number of places the output of the algorithm is different from the correct version.

Fig. 4 illustrates the final error rate of the proposed algorithm for different values of υ (noise parameter for pattern nodes) and ν (noise parameter for constraint nodes). The dashed lines correspond to the simulation results and the solid lines are the theoretical upper bounds. As evident from the figure, we witness a threshold phenomenon, i.e. the BER is negligible for $\epsilon \leq \epsilon^*$, and it grows as we move beyond

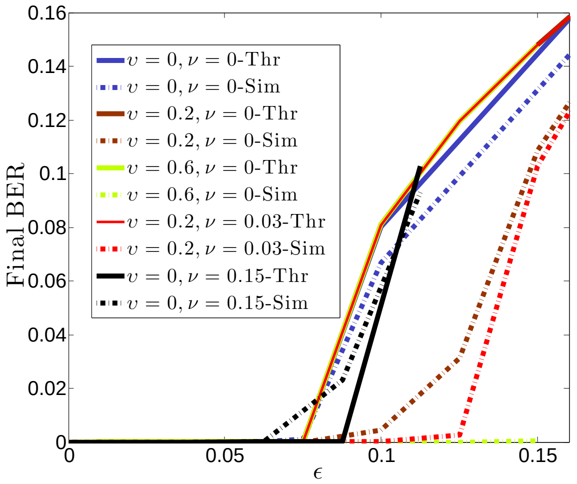


Fig. 4: The final pattern error probability for the a network with $n = 400$, $L = 50$, and on average 40 and 20 pattern and constraint nodes per cluster cf. [7]. The blue curves correspond to the noiseless neural network.

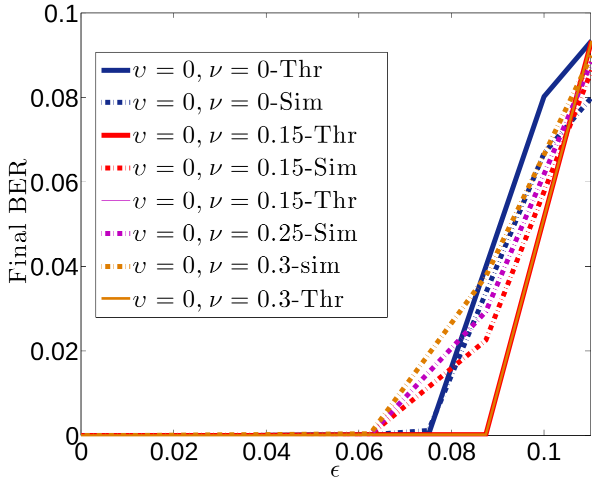


Fig. 5: The effect of ν on the final pattern error probability for the a network with $n = 400$, $L = 50$, and on average 40 and 20 pattern and constraint nodes per cluster cf. [7]. The blue curves correspond to the noiseless neural network. In all cases, the pattern neurons noise parameter, υ, is equal to zero.

this threshold. Furthermore, except for a few cases near the threshold, the simulations results are better than the theoretical upper bounds, a gap that might be attributed to the small network size used in our simulations.

Another interesting trend in Fig. 4 is the fact that the internal noise sometimes helps the network to achieve a better performance. This phenomenon, also known as *stochastic resonance* in the literature [15], [16], is indeed very similar to the one observed in genetic algorithms where limited amount of noise can help the network not get stuck in local minimums. To see why, note that Alg. 2 introduces no new errors. However, in each iteration, it might simply happen that the internal noise of neurons acts in our favor and helps clusters eliminate the external noise of their own, and those of neighbouring clusters. As a result, a small amount of deviations introduced by the internal noise might be enough for Alg. 2 to avoid places where the ”noiseless” architecture inevitably gets stuck. Nevertheless, as noise becomes too much, the performance deteriorates (the black curves in Fig. 4).

Figure 5 illustrates the effect of constraint neurons noise parameter ν on the overall BER. As evident from the figure, increasing ν from 0 to 0.15 reduces the overall BER considerably. However, beyond this point, if we continue increasing ν, we will end up with a higher BER. This behavior is in line with our conjecture that a limited amount of noise helps the network in reducing the recall error rate. However, as the noise is increased beyond some threshold, the recall probability of error increases, resulting is worse results.

V. Discussion

We have demonstrated that associative memories still work reliably even when built from unreliable hardware, addressing a major problem in fault-tolerant computing and arguing for the viability of associative memory models for the (noisy) mammalian brain. Further, 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.

The associative memory design we have proposed uses thresholding operations in the message-passing algorithm for recall; as part of our investigation, we optimized these neuron 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” [17]. The key to our result is capturing this benefit of digital processing: thresholding to prevent the build up of errors due to internal noise.

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 [10, 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 noise herein, cf. [14, 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.

References

[1] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.

[2] 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, no. 4, pp. 461–482, Jul. 1987.

[3] D. J. Amit and S. Fusi, "Learning in neural networks with material synapses," *Neural Comput.*, vol. 6, no. 5, pp. 957–982, Sep. 1994.

[4] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, July/Oct. 1948.

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

[6] 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.

[7] 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, to appear.

[8] T. F. Brady, T. Konkle, G. A. Alvarez, and A. Oliva, "Visual long-term memory has a massive storage capacity for object details," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 105, no. 38, pp. 14 325–14 329, Sep. 2008.

[9] 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, no. 5, pp. 745–757, Sep. 2004.

[10] L. R. Varshney, P. J. Sjöström, and D. B. Chklovskii, "Optimal information storage in noisy synapses under resource constraints," *Neuron*, vol. 52, no. 3, pp. 409–423, Nov. 2006.

[11] C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*. New York: Oxford University Press, 1999.

[12] M. G. Taylor, "Reliable information storage in memories designed from unreliable components," *Bell Syst. Tech. J.*, vol. 47, no. 10, pp. 2299–2337, Dec. 1968.

[13] A. V. Kuznetsov, "Information storage in a memory assembled from unreliable components," *Probl. Inf. Transm.*, vol. 9, no. 3, pp. 100–114, July-Sept. 1973.

[14] L. R. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4427–4444, Jul. 2011.

[15] 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, no. 7, pp. 3172–3184, Jul. 2007.

[16] K. Wiesenfeld and F. Moss, "Stochastic resonance and the benefits of noise: from ice ages to crayfish and SQUIDs," *Nature*, vol. 373, no. 6509, pp. 33–36, Jan. 1995.

[17] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology," *Neural Comput.*, vol. 10, no. 7, pp. 1601–1638, Oct. 1998.

[18] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

APPENDIX

PROOF OF LEMMA 2

To calculate the probability that a constraint node makes a mistake when there are 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

$$\begin{aligned} \pi_0^{(\ell)} &= \Pr\{|v_i| > \psi\} \\ &= \max\left(0, \frac{\nu - \psi}{\nu}\right). \end{aligned} \tag{15}$$

Thus, to make $\pi_0^{(\ell)} = 0$ we will select $\psi > \nu$.[2] So from now on, we assume

$$\pi^{(0)} = 0 \tag{16}$$

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

$$\begin{aligned} P_0^{(\ell)} &= \Pr\{|u_i| > \varphi\} \\ &= \max\left(0, \frac{\upsilon - \varphi}{\varphi}\right). \end{aligned} \tag{17}$$

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

PROOF OF THEOREM 1

The proof is similar to the proof of Theorem 3.50 in [18]. Each cluster node receives an error message from its neighboring pattern nodes with probability z. Now consider a given *noisy* pattern neuron which is connected to a given cluster $v^{(\ell)}$. Let $\Pi^{(\ell)}(t)$ be the probability that the cluster node $v^{(\ell)}$ with degree $\widetilde{d}_\ell$ sends an error message during iteration t of Algorithm 2. This event happens if

1) the cluster node $v^{(\ell)}$ receives at least one error message from its other neighbors among pattern neurons along its input edges, i.e. if it is connected to more than one noisy pattern neuron. Then, with probability one it send an error message.
2) the cluster node $v^{(\ell)}$ does not receive any other error messages from its other neighbors. In this case, it will send an error message with probability at most $P_1^{(\ell)} = 1 - P_c$.

Therefore,

$$\Pi^{(\ell)}(t) = 1 - P_c(1 - z(t))^{\widetilde{d}_\ell - 1} \tag{18}$$

As a result, if $\Pi(t)$ shows the average probability that a cluster node sends a message declaring the violation of at least one of its constraint neurons, we will have,

$$\begin{aligned} \Pi(t) &= \mathbb{E}_{\widetilde{d}_\ell}\{\Pi^{(\ell)}(t)\} = \sum_i \widetilde{\rho}_i (1 - P_c(1 - z(t))^{\widetilde{d}_\ell - 1}) \\ &= 1 - P_c \cdot \widetilde{\rho}(1 - z(t)) \end{aligned} \tag{19}$$

Now consider a given pattern neuron x_i which is connected to d_i clusters. This node will remain noisy in iteration $t+1$ of Algorithm 2 if it was noisy in the first place and in iteration $t+1$ all of its neighbors among constraint neurons send a violation message. Therefore, the probability of this node being noisy will be $z(0)(\Pi(t))^{d_i}$. As a result, noting that $z(0) = \epsilon$, the average probability that a pattern neurons remains noisy will be

$$z(t+1) = \epsilon \sum_i \widetilde{\lambda}_i (\Pi(t))^i = \epsilon \widetilde{\lambda}(\Pi(t)) = \epsilon \widetilde{\lambda}(1 - P_c \widetilde{\rho}(1 - z(t))) \tag{20}$$

Therefore, the decoding operation will be successful if $z(t+1) < z(t)$, $\forall t$. As a result, we must look for the maximum ϵ such that we will have $\epsilon \widetilde{\lambda}(1 - P_c \widetilde{\rho}(1 - z)) < z$ for $z \in [0, \epsilon]$.

[2] 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.

Proof of Lemma 3

It's now time to consider the situation in which we have one external error. 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

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

We consider two error events:

1) A constraint node j makes a mistake and do 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 send 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):

$$\begin{aligned} \pi_2^{(1)} &= \Pr\{w_{ji} + v_j < -\psi\} \\ &= \max\left(0, \frac{\nu - (\psi + w_{j1})}{2\nu}\right). \end{aligned} \tag{21}$$

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 remains to do is to calculate the probability that they remain silent by mistake.

To this end, we will have

$$\begin{aligned} \pi_1^{(1)} &= \Pr\{|w_{ji} + v_j| < \psi\} \\ &= \max\left(0, \frac{\nu + \min(\psi - w_{j1}, \nu)}{2\nu}\right). \end{aligned} \tag{22}$$

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

$$\pi_1^{(1)} \leq \max\left(0, \frac{\nu - (\eta - \psi)}{2\nu}\right). \tag{23}$$

Putting the above equations together, we will obtain

$$\pi^{(1)} \leq \max\left(0, \frac{\nu - (\eta - \psi)}{2\nu}\right). \tag{24}$$

In case $\eta - \psi > \nu$, we could even manage to make this probability equal to zero. However, we will leave it as it is and use equation (24) to calculate $P_1^{(\ell)}$.

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

$$\begin{aligned} q_1^{(\ell)} &= \Pr\{|g_j^{(\ell)}| \geq \varphi | \deg(b_j) = d_j \& |\mathcal{N}(x_1) \cap \mathcal{N}(b_j)| = a\} \\ &= \Pr\{\frac{o_j}{d_j} + u_j \geq \varphi\} + \Pr\{\frac{o_j}{d_j} + u_j \leq -\varphi\}, \end{aligned} \tag{25}$$

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

By simplifying equation (25) we will 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 should now average the above equation over o_j, b_c, b and d_j. To start, suppose out of the b_c non-zero messages the node b_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

$$\bar{q}_1^{(\ell)} = \sum_{e=0}^{b_c} \binom{b_c}{e} (1/2)^{b_c} q_1^{(\ell)}(2e - b_c). \tag{26}$$

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

$$\bar{q}_1^{(\ell)} = \sum_{b_c=0}^{b} \binom{b}{b_c} (\pi_1^{(\ell)})^{a-b_c} (1 - \pi_1^{(\ell)})^{b_c} \sum_{e=0}^{b_c} \binom{b_c}{e} (1/2)^{b_c} q_1^{(\ell)}(2e - b_c). \tag{27}$$

Finally, the probability that b_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, and again with some abuse of notation, we will obtain

$$\bar{q}_1^{(\ell)} = \sum_{b=0}^{d_j} p_b \sum_{b_c=0}^{b} p_{b_c} \sum_{e=0}^{b_c} \binom{b_c}{e} (1/2)^{b_c} q_1^{(\ell)}(2e - b_c), \tag{28}$$

where $q_1^{(\ell)}(2e-b_c)$ is given by (9), p_b is the probability of having b common neighbors and is estimated by $\binom{d_j}{b}(1-\bar{d}^{(\ell)}/m_\ell)^{d_j-a}(\bar{d}^{(\ell)}/m_\ell)^b$, 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, $p_{b_c}=\binom{b}{b_c}(\pi_1^{(\ell)})^{a-b_c}(1-\pi_1^{(\ell)})^{b_c}$. 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

$$\begin{aligned} q_2^{(\ell)} &= \Pr\{g_1^{(\ell)}\leq\varphi|\deg(x_1)=d_1\&j \text{ errors in constraints}\} \\ &= \Pr\{\frac{d_1-j}{d_1}+u<\varphi\}, \end{aligned} \tag{29}$$

which simplifies to

$$q_2^{(\ell)}(j)=\begin{cases} +1, & \text{if } |j|\geq(1+\upsilon-\varphi)d_1 \\ \max(0,\frac{\upsilon-\varphi}{\upsilon}), & \text{if } |j|\leq(1-\upsilon-\varphi)d_1 \\ \frac{\upsilon+\varphi-(d_1-j)/d_1}{2\upsilon}, & \text{if } |\varphi d_1-(d_1-j)|\leq\upsilon d_1. \end{cases} \tag{30}$$

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 will get

$$\bar{q}_2^{(\ell)}=\sum_{j=0}^{d_1}\binom{d_1}{j}(\pi_1^{(\ell)})^j(1-\pi_1^{(\ell)})^{d_1-j}q_2^{(\ell)}(j), \tag{31}$$

where $q_2^{(\ell)}(j)$ is given by equation (30).

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

$$P_1^{(\ell)}=\frac{1}{n^{(\ell)}}\bar{q}_2^{(\ell)}+\frac{n^{(\ell)}-1}{n^{(\ell)}}\bar{q}_1^{(\ell)} \tag{32}$$

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

Investigating the effect of choosing proper φ

We now apply numerical methods to equation (32) in order 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, $P_1^{(\ell)}$ as well for comparison.

Figure 3 illustrates the behavior of the error probability as a function of φ for different values of υ and for $\pi_1^{(\ell)}=0.02$.

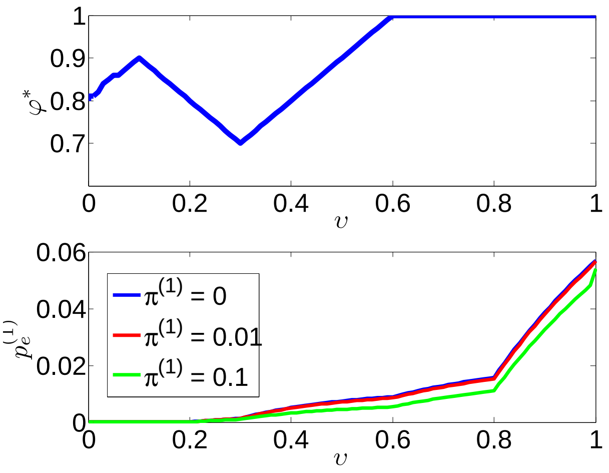


Fig. 6: The behavior of $P_1^{(\ell)}$ as a function of φ^* for different values of noise parameter, υ and

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.