Robust super-polynomial pattern storage in Hopfield networks

Christopher Hillar, Redwood Center for Theoretical Neuroscience, UC Berkeley, CA, USA
Ngoc M. Tran, University of Texas, Austin, TX, USA

Summary. The Hopfield recurrent neural network is an auto-associative distributed model of memory. This architecture is able to store collections of generic binary patterns as *robust attractors*; i.e., fixed-points of the network dynamics having large basins of attraction. However, the number of (randomly generated) storable memories scales at most linearly in the number of neurons, and it has been a long-standing question whether robust super-polynomial storage is possible in recurrent networks of linear threshold elements. Here, we design sparsely-connected Hopfield networks on n-nodes having $\frac{2^{\sqrt{2n}+\frac{1}{4}}}{n^{1/4}\sqrt{\pi}}$ graph cliques as robust memories by analytically minimizing the probability flow objective function over these patterns [HT14]. Our methods also provide a biologically plausible convex learning algorithm that efficiently discovers these networks from training on very few sample memories. Our networks also have applications to coding theory and its relation to statistical physics [VSK02] as they provide a novel family of (nonlinear) error-correcting codes that are simple to implement, parallelizable, and tolerate symmetric binary channel corruption of up to 50% (i.e., achieving the Shannon bound for low density codes).

Background. The n-node Hopfield model [Hop82] is a symmetric recurrent network of linear threshold McCulloch-Pitts neurons that robustly stores $n/(4 \ln n)$ generic binary patterns as distributed memories using the so-called *outer-product learning rule* (OPR). While several aspects of the Hopfield network appeared earlier, the model connects ideas of neural computing with statistical mechanics. Since then memory efforts have mainly focused on storing sets of random patterns X using OPR, which adjusts couplings between network nodes to be a simple correlation over X. Independent of the method, the number of randomly generated dense patterns storable in a Hopfield network with n nodes is at most $2n$. Interestingly, if the binary patterns to memorize have few nonzero entries, then it is sometimes possible to store nearly a quadratic number of them. Despite this potential for super-linear capacity, it has been a basic open question whether a Hopfield network exists with an identified exponential number of memories, each one a robust attractor. Even so, there are several examples of large storage in Hopfield networks. For instance, it is known that a random (independent standard normal couplings) n-node Hopfield network has $\approx 1.22^n$ memories, but these patterns are difficult to determine from the network and have narrow basins of attraction. Here, we construct (and learn in a local, neurologically plausible manner) families of sparsely-connected Hopfield networks with robust exponential memory. **Short Python code**: www.msri.org/people/members/chillar/files/local_mpf_rule.py

References

[Hop82] J.J. Hopfield. *Proc. Nat. Acad. Sci. U.S.A.*, 79(8):2554, 1982.

[HT14] C. Hillar and N. M. Tran. *ArXiv e-prints: nlin.AO 1411.4625*, 2014.

[VSK02] Renato Vicente, David Saad, and Yoshiyuki Kabashima. *Advances in Imaging and Electron Physics*, 125:232–355, 2002.

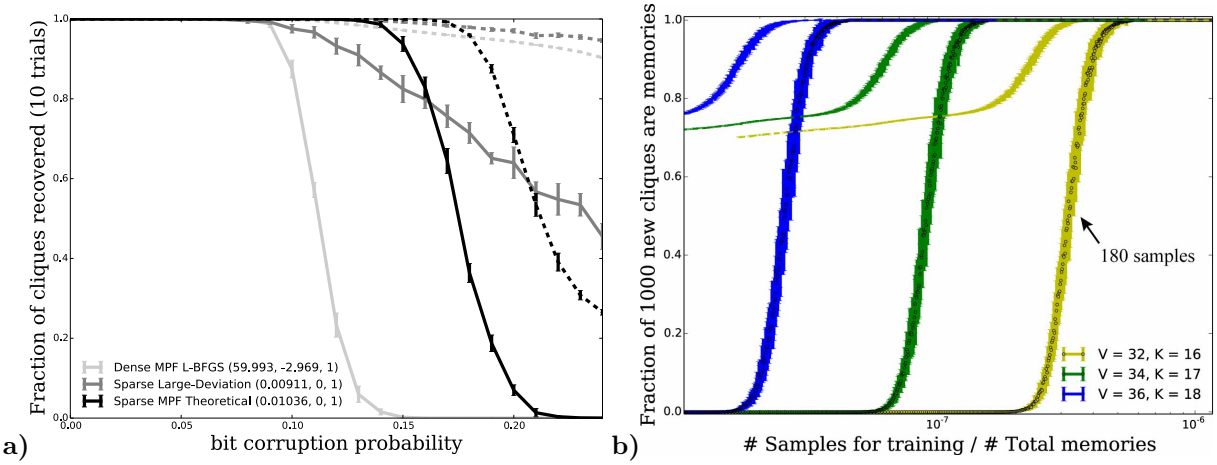

Figure 1: **a) Exponential memories, exponential attraction basins [HT14]**. Denoising performance of Hopfield networks storing all 64-cliques in $v = 128$ vertex graphs using a dense 8128-bit network minimizing probability flow on 50000 random cliques (light gray line), sparsely-connected $(x, 0, 1)$ network with x as in Large Deviation theory argument and $p = 1/4$ (gray), or MPF theoretical optimum (black). 200 cliques chosen at random were p-corrupted for different p and then dynamics were converged initialized at noisy cliques. The plot shows the fraction of cliques correctly recovered as a function of the pattern corruption p. Dotted lines are average bits retrieved correctly. **b) Number of samples before storing all cliques is small.** For $v = 32, 34, 36$ ($k = 16, 17, 18$) with 50 trials each, the percent of 1000 random k-cliques that are memories vs. the fraction of training samples to total number $\binom{v}{k}$ of k-cliques; dotted lines are average percentage of correct bits after converging dynamics.