Efficiency of Local Learning Rules in Threshold-Linear Associative Networks

Francesca Schönsberg,[1,*] Yasser Roudi,[2] and Alessandro Treves[1,2]
[1] *SISSA, Scuola Internazionale Superiore di Studi Avanzati, Trieste, Italy*
[2] *Kavli Institute for Systems Neuroscience & Centre for Neural Computation, NTNU, Trondheim, Norway*

(Received 30 June 2020; revised 10 December 2020; accepted 11 December 2020; published 6 January 2021)

We derive the Gardner storage capacity for associative networks of threshold linear units, and show that with Hebbian learning they can operate closer to such Gardner bound than binary networks, and even surpass it. This is largely achieved through a sparsification of the retrieved patterns, which we analyze for theoretical and empirical distributions of activity. As reaching the optimal capacity via nonlocal learning rules like back propagation requires slow and neurally implausible training procedures, our results indicate that one-shot self-organized Hebbian learning can be just as efficient.

DOI: 10.1103/PhysRevLett.126.018301

Introduction.—Learning in neuronal networks is believed to happen largely through changes in the weights of the synaptic connections between neurons. Local learning rules, those that self-organize through weight changes depending solely on the activity of pre- and postsynaptic neurons, are generally considered to be more biologically plausible than nonlocal ones [1]. But how effective are local learning rules? Quite ineffective, has been the received wisdom since the 1980s, when nonlocal iterative algorithms came to the fore. However, this wisdom, when it comes to memory storage and retrieval, is largely based on analyzing networks of binary neurons [2–5], while neurons in the brain are not binary.

A better, but still mathematically simple description of neuronal input-output transformation is through threshold-linear (TL) activation function [6,7], also predominantly adopted in recent deep learning applications (called ReLU in that context) [8]. Therefore, one may ask if the results from the 1980s highlighting the contrast between the effective, iterative procedures used in machine learning and the self-organized, one-shot, perhaps computationally ineffective local learning rules are valid beyond binary units [9].

The Hopfield model, a most studied model of memory, is a fully connected network of N binary units endowed with a local, *Hebbian* learning rule [2,3]: the weight between two units increases if they have the same activity in a memory pattern; otherwise it decreases. The network can retrieve only up to $p_{\max} \simeq 0.14N$ patterns, while, in comparison, Elizabeth Gardner showed [5] that with C connections per unit, the optimal capacity that such a network can attain is $p_{\max} = 2C$, about 14 times higher; the bound can be approached through iterative procedures like backpropagation that progressively reduce the difference between current and desired output. This consolidated the impression that unsupervised, Hebbian plasticity may well be of biological interest, but is rather inefficient for memory storage. In the fully connected Hopfield model, the transition to no retrieval is discontinuous: right below the storage capacity, $\sim$1.5% of units in a retrieved pattern are misaligned with the stored pattern, but 50%, i.e., chance level, just above the capacity [3]. This rather low error certainly contributes to the low capacity. However, the negative characterization of Hebbian learning in binary networks persisted even when more errors occur: in the more biologically relevant highly diluted networks the error smoothly goes to 50% [10], but the capacity is still a factor of 3 away [4], *approaching* the bound only when the fraction of active unit in each pattern is $f \ll 1$ [11].

What about TL units? Are they more efficient in the unsupervised learning of memory patterns? Here, we study the optimal pattern capacity *à la* Gardner in networks of TL units. Past work discussed above [11] had suggested that the distribution of activity (along with the connectivity) may play a role in how efficient Hebbian learning is, but, back then, this only meant changing f. Besides being a better model of neuronal input-output transformation, by allowing nonbinary patterns, TL units permit a better understanding of the interplay between the retrieval properties of recurrent networks and the distribution of the activity stored in the network. In fact, we show that while for binary patterns the Gardner bound is larger than the Hebbian capacity no matter how sparse the code, this does not, in general, hold for nonbinary stored patterns: the Hebbian capacity can even surpass the bound. This perhaps surprising violation of the bound is because the Gardner calculation imposes an infinite output precision [12], while Hebbian learning exploits its loose precision to *sparsify* the retrieved pattern. In other words, with TL units, Hebbian capacity can get much closer to the optimal capacity or even surpass it, by retrieving a sparser version of the stored pattern. We find that experimentally observed distributions from the inferior-temporal visual cortex [13], which can be taken as patterns to be stored, would be sparsified about 50% by Hebbian learning, and would reach about 50%−80% of the Gardner bound.

© 2021 American Physical Society

Model description.—We consider a network of N units and p patterns of activity, $\{\eta_i^\mu\}_{i=1,\ldots,N}^{\mu=1,\ldots,p}$ each representing one memory stored in the connection weights via some procedure. Each η_i^μ is drawn independently for each unit i and each memory μ from a common distribution $\Pr(\eta)$. The activity of unit i is denoted by v_i and is determined by the activity of the C units feeding to it as

$$v_i = g[h_i - \vartheta]^+, \tag{1a}$$

$$h_i\{v_i\} = \frac{1}{\sqrt{C}} \sum_j J_{ij} v_j, \tag{1b}$$

where $[x]^+ = x$ for $x > 0$ and $= 0$ otherwise; and both the gain g and threshold ϑ are fixed parameters. The storage capacity, or capacity for short, is defined as $\alpha_c \equiv p_{\max}/C$, with $p_{\max}$ the maximal number of memories that can be stored and individually retrieved. The synaptic weights J_{ij} are taken to satisfy the spherical normalization condition for all i

$$\sum_{j \neq i} J_{ij}^2 = C. \tag{2}$$

We are interested in finding the set of J_{ij} that satisfy Eq. (2), such that patterns $\{\eta_i^\mu\}_{i=1,\ldots,N}^{\mu=1,\ldots,p}$ are self-consistent solutions of Eqs. (1), namely that for all i and μ we have, $h_i^\mu = \vartheta + \eta_i^\mu/g$ if $\eta_i^\mu > 0$ and $h_i^\mu \leq \vartheta$ if $\eta_i^\mu = 0$.

Replica analysis.—Adapting the procedure introduced in [5] for binary units to our network, we evaluate the fractional volume of the space of the interactions J_{ij} which satisfy Eqs. (1) and (2), using the replica trick and the replica symmetry ansatz, we obtain the standard order parameters $m = (1/\sqrt{C}) \sum_j J_{ij}$ and $q = (1/C) \sum_j J_{ij}^a J_{ij}^b$ corresponding, respectively, to the average of the weights within each replica and to their overlap between two replicas a and b (Supplemental Material [14], Sec. A). Increasing p, for $C \to \infty$, shrinks the volume of the compatible weights, eventually to a single point, i.e., when there is only a unique solution and the storage capacity is reached. This corresponds to the case where all the replicated weights are equal $q \to 1$, implying that only one configuration satisfying all the equations exists. Adding a further memory pattern would make it impos-sible, in general, to satisfy them all. At the end, we obtain the following equations for α_c:

$$0 = -f\left(x + \frac{d_1}{g\sqrt{d_3}}\right) + (1-f)\int_x^\infty Dt(t-x),$$
$$\frac{1}{\alpha_c} = f\left[x^2 + \frac{d_2}{g^2 d_3} + \frac{2xd_1}{g\sqrt{d_3}} + 1\right] + (1-f)\int_x^\infty Dt(t-x)^2, \tag{3}$$

where we have introduced the averages over $\Pr(\eta)$: $d_1 \equiv \langle \eta_i^\mu \rangle$, $d_2 \equiv \langle (\eta_i^\mu)^2 \rangle$ and $d_3 \equiv d_2 - d_1^2$; $x = (\vartheta - d_1 m)/\sqrt{d_3}$ is the normalized difference between the threshold and the mean input, while $f = \Pr(\eta > 0)$ is the fraction of active units and $Dt \equiv dt \exp(-t^2/2)/\sqrt{2\pi}$. The two equations yield x and α_c. Both equations can be understood as averages over units, respectively, of the actual input and of the square input, which determine the amount of quenched noise and hence the storage capacity.

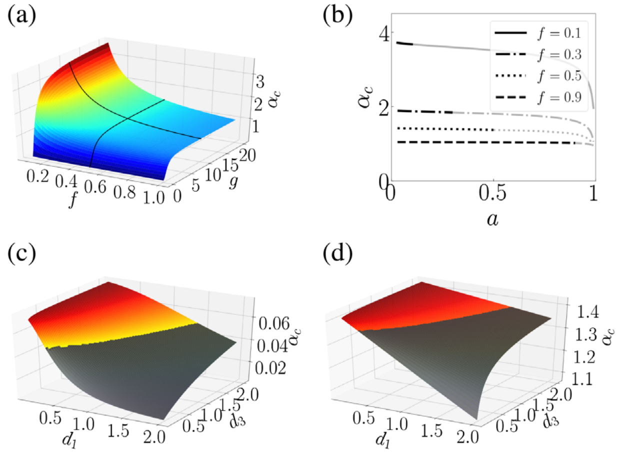

FIG. 1. Dependence of the Gardner capacity α_c on different parameters: in (a) as a function of g and f ($d_1 = 1.1$, $d_2 = 2$), in (b) as a function of $a = d_1^2/d_2$ for different values of f ($g = 10$, $d_1 = 1.1$), in (c) and (d) as a function of d_1 and d_3 for $g = 0.2$ and $g = 10$, respectively ($f = 0.5$). Note that fixing f restricts the available range of a, as a cannot be larger than f; the inaccessible ranges are shadowed in (b)–(d).

The capacity α_c then depends on the proportion f of active units, but also on the gain g, and on the cumulants d_1 and d_3. Figure 1(a) shows that at fixed g, α_c increases as more and more units remain below threshold, ceasing to contribute to the quenched noise. In fact, α_c diverges as $[2f \ln(1/\sqrt{2\pi}f)]^{-1}$, for $f \to 0$; see Supplemental Material [14], Sec. B. At fixed f, there is an initially fast increase with g followed by a plateau dependence for larger values of g. One can show that $\alpha_c \to (g^2/g^2 + 1)$ as $f \to 1$, i.e., when all the units in the memory patterns are above threshold, it is always $\alpha_c < 1$ for any finite g. At first sight this may seem absurd: a linear system of N^2 independent equations and N^2 variables always has an inverse solution, which would lead to α_c being (at least) one. Similar to what was already noted in [12], however, the inverse solution does not generally satisfy the spherical constraint in Eq. (2); but it does, in our case, in the limit $g \to \infty$ and this can also be understood as the reason why α_c is highest when g is very large. In practice, Fig. 1 indicates that over a broad range of f values, α_c approaches its $g \to \infty$ limit already for moderate values of g; while the dependence on d_1 and d_3 is only noticeable for small g, as can be seen by comparing Figs. 1(c) and 1(d). For $g \to \infty$, one sees that Eqs. (3) depend on $\Pr(\eta)$ only through f.

Equations (3), at $g \to \infty$, have been verified by explicitly training a threshold linear perceptron with binary patterns,

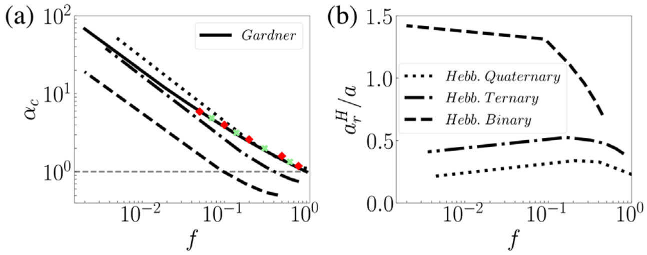

FIG. 2. Hebbian vs Gardner capacity. (a) α_c^H vs f for different sample distribution of stored patterns compared to the analytically calculated universal α_c^G; the red diamonds and green crosses are reached using perceptron training for binary and ternary patterns, respectively. (b) the sparsification of the stored patterns at retrieval, for Hebbian networks at their capacity.

evaluating α_c numerically as the maximal load which can be retrieved with no errors; see Supplemental Material [14], Sec. C for details. Estimated values of α_c are depicted by red diamonds in Fig. 2, and they follow the profile of the solid line describing the $g \to \infty$ limit of Eqs. (3).

Comparison with a Hebbian rule: Theoretical analysis.—With highly diluted connectivity and nonsparse patterns a binary network can get to $1/\pi$ of the bound, even if with vanishing overlaps, much closer than in the fully connected case. This is intuitively because the quenched noise is diminished as J_{ij} and J_{ji} become effectively independent. Besides its biological relevance, with TL units, the fair comparison to the capacity *à la* Gardner is thus that of a Hebbian network with *highly* diluted connectivity. In what follows, we indicate the Gardner capacity as calculated in the previous section and the Hebbian capacity, by α_c^G and α_c^H, respectively, and use similar superscript notations for other quantities.

The capacity of the TL network with diluted connectivity was evaluated analytically in [16]; see Supplemental Material [14], Sec. D for a recap. Whereas for $g \to \infty$ the Gardner capacity depends on $\Pr(\eta)$ only via f, for Hebbian networks it does depend on the distribution, and most importantly on a, the *sparsity*

$$a = \langle \eta_i^\mu \rangle^2 / \langle (\eta_i^\mu)^2 \rangle \tag{4}$$

whose relation to f depends on the distribution [16].

Figure 2 shows the results for three examples of binary, ternary, and quaternary distributions for which f and a are related through $f = a$, $9a/5$, and $9a/4$, respectively, see Supplemental Material [14], Sec. E; the Hebbian and the Gardner capacities diverge in the sparse coding limit.

When attention is restricted to binary patterns in Fig. 2(a), the Gardner capacity α_c^G *seems* to provide an upper bound to the capacity reached with Hebbian learning; more structured distributions of activity, however, dispel such a false impression: the quaternary example already shows higher capacity for sufficiently sparse patterns. The bound, in fact, would only apply to perfect errorless retrieval, whereas Hebbian learning creates attractors which are, up to the Hebbian capacity limit, correlated but not identical to the stored patterns; in particular, we notice that when considering TL units and Hebbian learning, in order to reach close to the capacity limit, the threshold has to be such as to produce sparser patterns at retrieval, in which only the units with the strongest inputs get activated. Figure 2(b) shows the ratio of the sparsity of the retrieved pattern produced by Hebbian learning, $a_r^H = \langle v_i^\mu \rangle^2 / \langle (v_i^\mu)^2 \rangle$ (estimated as described in Supplemental Material [14]) to that of the stored pattern a, vs f: except for the binary patterns at low f, the retrieved patterns, at the storage capacity, are always sparser than the stored ones. The largest sparsification happens for quaternary patterns, for which the Hebbian capacity overtakes the Gardner bound, at low f. Sparser patterns emerge as, to reach close to α_c^H, ϑ has to be such as to inactivate most of the units with intermediate activity levels in the stored pattern. Of course, the perspective is different if α_c^H is considered as a function of a_r instead of a, in which case the Gardner capacity remains unchanged, as it implies retrieval with $a_r = a$, and is above α_c^H for each of the three sample distributions; see Fig. 1 of Supplemental Material [14].

Comparison with a Hebbian rule: Experimental data.—Having established that the Hebbian capacity of TL networks can surpass the Gardner bound for some distributions, we ask what would happen with distributions of firing rates naturally occurring in the brain. We considered published distributions of single neurons in the inferior-temporal cortex in response to short naturalistic movies [13]. Such distributions can be taken as examples of patterns elicited by visual stimuli, to be stored with Hebbian learning, given appropriate conditions, and later retrieved using attractor dynamics, triggered by a partial cue [17,18]. How many such patterns can be stored, and with what accompanying sparsification?

Figures 3(a) and 3(b) show the analysis of two sample distributions from [13]. The observed distributions, in blue, labeled "Gardner," are those we assume could be stored and retrieved, exactly as they were, with a suitable training procedure bound by the Gardner capacity. In orange, we plot the distribution that would be retrieved following Hebbian learning operating at its capacity, see Supplemental Material [14], Sec. I for the estimation of the retrieved distribution. Note that the absolute scale of the retrieved firing rate is arbitrary; what is fixed is only the shape of the distribution, which is sparser (as clear already from the higher bar at zero). The pattern in Fig. 3(a), which has $a < 0.5$, could also be fitted with an exponential distribution having $f = 2a$ (see Supplemental Material [14], Sec. F). In that panel we also show the values of $\alpha_c^{H_{\mathrm{exp}}}$ and $a_r^{H_{\mathrm{exp}}}$, calculated assuming the exponential fit, along with values from the observed discrete distribution ($\alpha_c^{H_{\mathrm{naive}}}$ and $a_r^{H_{\mathrm{naive}}}$). Figure 3(c) shows both α_c^G and $\alpha_c^{H_{\mathrm{exp}}}$ versus f; we have indicated by diamonds the Hebbian

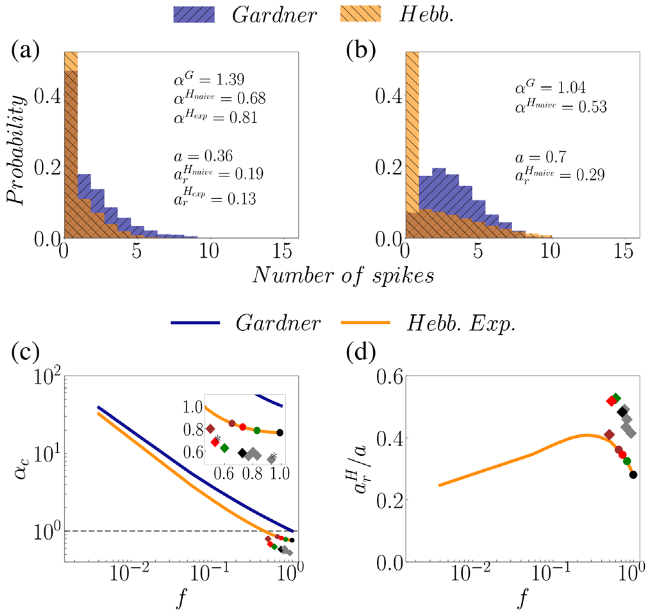

FIG. 3. Hebbian vs Gardner capacity for experimental data. (a), (b) histograms of two experimentally recorded spike counts (blue) and the retrieved distributions, if the patterns were stored using Hebbian learning (orange). Note that the retrieved distributions *à la* Gardner would be the same as the stored patterns. (c) Analytically calculated Gardner capacity α_c^G (blue), compared to $\alpha_c^{H_{\rm exp}}$ for the Hebbian learning of an exponential distribution (orange, circles). $\alpha_c^{H_{\rm naive}}$ is shown by diamonds. The asterisks mark the two cells whose distribution is plotted in (a) and (b). (d) Sparsification of the retrieved patterns, for Hebbian learning.

capacities for the nine empirical distributions in [13] and by circles the fitted values for those which could be fitted to an exponential. In the Supplemental Material [14], Sec. G we also discuss the fit to a log-normal, which is better at reproducing experimental distributions with a mode above zero [19], as in Fig. 3(b).

There are three conclusions that we can draw from these data. First, the Hebbian capacity from the empirical distributions is about 80% of that of the exponential fit, when available. Second, in general for distributions like those of these neurons, the capacity achieved by Hebbian learning is about 50%–80% of the Gardner capacity, depending on the neuron and whether we take its discrete distribution *as is*, or fit it to an exponential (or, e.g., to a log-normal) shape. Third, with Hebbian learning retrieved patterns tend to be 2–3 times sparser than the stored ones, again depending on the particular distribution, empirical or exponential fit (as for nonsparse distributions, which could be better fit by a log-normal, see Supplemental Material [14], Sec. G). As illustrated in Fig. 3(d), the empirical distributions achieve a lower capacity than that of their exponential fit, as the latter leads to further sparsification at retrieval.

Discussion.—While instrumental in conceptualizing memory storage [20], Hebbian learning has been widely considered a poor man's option, relative to more powerful machine learning algorithms that could reach the Gardner bound for binary units and patterns. No binary or quasibinary pattern of activity has ever been observed in the cerebral cortex, however. A few studies have considered TL units, showing them to be less susceptible to memory mix-up effects [21] or perturbations in the weights and inputs values [22] but, in the framework of *à la* Gardner calculations, they have focused on issues other than associative networks storing sparse representations. For instance, a replica analysis was carried out in [12] with a generic gain function, but then discussed only in a quasibinary regime. Others considered monotonically increasing activation functions under the constraint of nonnegative weights [23]. Here, we report the analytical derivation of the Gardner capacity for TL networks, validate it via perceptron training, and compare it with Hebbian learning. We find that the bound can be reached or even surpassed, and that retrieval leads to sparsification. For sample experimental distributions, we find that one-shot Hebbian learning can utilize 50%–80% of the available "errorless" capacity if retrieving sparser activity, compatible with recent observations [18].

In deriving the Gardner bound, we assumed errorless retrieval and it remains to be seen how much allowing errors increases this bound for TL units and neurally plausible distributions. For the binary case of [10], as already mentioned, this errorless bound is still above the Hebbian capacity of the highly diluted regime, with its continuous (second order) transition, i.e., with vanishing overlap at storage capacity [10]. How does the overlap behave in the TL case? For highly diluted TL networks with Hebbian learning, in fact, except for special cases, the transition at capacity is discontinuous: the overlap drops to zero from a nonzero value that depends on the distribution of stored neural activity but can be small [24]. It is worth noting, though, that while in the binary case the natural measure of error is simply the fraction of units misaligned at retrieval, in the TL case error can be quantified in other ways. In the extreme in which only the most active cells remain active at retrieval, those retrieved memories cannot be regarded as the full pattern, with its entire information content, but more as a pointer, effective perhaps as a mechanism only to distinguish between different possible patterns or to address the full memory elsewhere, as posited in *index* theories of two-stage memory retrieval [25]. Further understanding would also derive from comparing the maximal information content per synapse for TL units, with Hebbian or iterative learning, as previously studied for binary networks [26]. Using nonbinary patterns might also afford a solution to the low storage capacity observed in balanced memory networks storing binary patterns [27].

Our focus here has been on memory storage in associative neural networks, with the overarching conclusion that the relative efficiency of Hebbian learning is much higher when units have a similar transfer function to real cortical neurons. The efficiency of local learning rules had also been challenged by their comparatively weaker performance in other (machine learning) settings [28], while results to the contrary are also reported [29,30]. It may therefore be argued that the efficiency of local learning in these settings might also be fundamentally dependent on both the types of units used and the data, observations consistent with the findings in [30,28], respectively. In evaluating a learning rule, it may therefore be crucial to consider whether it is suited to the transfer function and data representation it operates on.

We thank Rémi Monasson and Aldo Battista for useful comments and discussions. This research received funding from the EU Marie Skłodowska-Curie Training Network 765549 M-Gate, and partial support from Human Frontier Science Program RGP0057/2016, the Research Council of Norway (Centre for Neural Computation, Grant No. 223262; NORBRAIN1, Grant No. 197467), and the Kavli Foundation.

Y. R. and A. T. contributed equally to this work.

[*]Corresponding author.
francesca.schonsberg@sissa.it

[1] T. H. Brown, E. W. Kairiss, and C. L. Keenan, Annu. Rev. Neurosci. **13**, 475 (1990); D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory* (J. Wiley, Chapman & Hall, New York, 1949); D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks* (Cambridge University Press, Cambridge, England, 1992).

[2] J. J. Hopfield, Proc. Natl. Acad. Sci. U.S.A. **79**, 2554 (1982).

[3] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Ann. Phys. (N.Y.) **173**, 30 (1987).

[4] B. Derrida, E. Gardner, and A. Zippelius, Europhys. Lett. **4**, 167 (1987).

[5] E. Gardner, J. Phys. A **21**, 257 (1988).

[6] H. K. Hartline and F. Ratliff, J. Gen. Physiol. **40**, 357 (1957); J. Gen. Physiol. **41**, 1049 (1958).

[7] A. Treves, J. Phys. A **23**, 2631 (1990).

[8] V. Nair and G. E. Hinton, in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (2010), pp. 807–814; A. L. Maas, A. Y. Hannun, and A. Y. Ng, in *Proceedings of ICML*, 1 (2013), p. 3; K. He, X. Zhang, S. Ren, and J. Sun, in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1026–1034; I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).

[9] U. Pereira and N. Brunel, Neuron **99**, 227 (2018).

[10] E. Gardner, S. Mertens, and A. Zippelius, J. Phys. A **22**, 2009 (1989).

[11] M. V. Tsodyks and M. V. Feigel'man, Europhys. Lett. **6**, 101 (1988).

[12] D. Bollé, R. Kuhn, and J. Van Mourik, J. Phys. A **26**, 3149 (1993).

[13] A. Treves, S. Panzeri, E. T. Rolls, M. Booth, and E. A. Wakeman, Neural Comput. **11**, 601 (1999).

[14] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.126.018301 for analytical derivations, their numerical validation, and further comparisons with experimental data, which includes Ref. [15].

[15] A. Treves, Phys. Rev. A **42**, 2418 (1990); Y. Roudi and A. Treves, Phys. Rev. E **73**, 061904 (2006).

[16] A. Treves, J. Phys. A **24**, 327 (1991); A. Treves and E. T. Rolls, Netw., Comput. Neural Syst. **2**, 371 (1991).

[17] J. M. Fuster and J. P. Jervey, Science **212**, 952 (1981); Y. Miyashita, Nature (London) **335**, 817 (1988); K. Nakamura and K. Kubota, J. Neurophysiol. **74**, 162 (1995); D. J. Amit, S. Fusi, and V. Yakovlev, Neural Comput. **9**, 1071 (1997); E. T. Rolls and A. Treves, *Neural Networks and Brain Function* (Oxford University Press, Oxford, 1998).

[18] S. Lim, J. L. McKee, L. Woloszyn, Y. Amit, D. J. Freedman, D. L. Sheinberg, and N. Brunel, Nat. Neurosci. **18**, 1804 (2015).

[19] G. Buzsáki and K. Mizuseki, Nat. Rev. Neurosci. **15**, 264 (2014).

[20] D. J. Amit and N. Brunel, Netw., Comput. Neural Syst. **8**, 373 (1997); F. P. Battaglia and A. Treves, Neural Comput. **10**, 431 (1998); K. Yoon, M. A. Buice, C. Barry, R. Hayman, N. Burgess, and I. R. Fiete, Nat. Neurosci. **16**, 1077 (2013).

[21] A. Treves, J. Phys. A **24**, 2645 (1991); Y. Roudi and A. Treves, Phys. Rev. E **67**, 041906 (2003).

[22] C. Baldassi, E. M. Malatesta, and R. Zecchina, Phys. Rev. Lett. **123**, 170602 (2019).

[23] C. Clopath and N. Brunel, PLoS Comput. Biol. **9** (2013).

[24] F. Schönsberg, Y. Roudi, and A. Treves (to be published).

[25] T. J. Teyler and P. DiScenna, Behavioral Neuroscience **100**, 147 (1986).

[26] J.-P. Nadal and G. Toulouse, Netw., Comput. Neural Syst. **1**, 61 (1990).

[27] Y. Roudi and P. E. Latham, PloS Comput. Biol. **3**, e141 (2007).

[28] S. Bartunov, A. Santoro, B. Richards, L. Marris, G. E. Hinton, and T. Lillicrap, in *Advances in Neural Information Processing Systems 31*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc., New York, 2018), pp. 9368–9378.

[29] Y. Amit, Front. Comput. Neurosci. **13**, 18 (2019).

[30] D. Krotov and J. J. Hopfield, Proc. Natl. Acad. Sci. U.S.A. **116**, 7723 (2019).