Learning by message passing in networks of discrete synapses

Learning by Message Passing in Networks of Discrete Synapses

Alfredo Braunstein[1] and Riccardo Zecchina[1]
[1]*ICTP, Strada Costiera 11, I-34100 Trieste, Italy*
(Received 8 November 2005; published 25 January 2006)

We show that a message-passing process allows us to store in binary ''material'' synapses a number of random patterns which almost saturate the information theoretic bounds. We apply the learning algorithm to networks characterized by a wide range of different connection topologies and of size comparable with that of biological systems (e.g., $n \simeq 10^5$–10^6). The algorithm can be turned into an online—fault tolerant—learning protocol of potential interest in modeling aspects of synaptic plasticity and in building neuromorphic devices.

DOI: 10.1103/PhysRevLett.96.030201 PACS numbers: 87.19.La, 02.50.−r, 05.45.−a, 75.10.Nr

Learning and memory are implemented in neural systems mostly through distributed changes of synaptic efficacy [1]. The learning problem in neural networks (NNs) asks whether one can find values for the synaptic efficacies such that a set of p patterns are stored simultaneously. Depending on the structure of the network—feed-forward or recurrent—the storage problem is either seen as a classification problem (input patterns are classified according to the output of the network) or as an attractor dynamics problem (patterns are the external stimuli which drive the dynamics of the network to the closest attractor) [2]. In any case, understanding the mechanisms underlying synaptic changes constitutes a crucial step for modeling real neural circuits (e.g., the Purkinje cells in the cerebellum [3]). On the purely theoretical side many basic results have been derived, ranging from information theoretic bounds [4,5] and statistical physics analysis of learning capabilities [6] in model NNs to concrete algorithms, like artificial pattern recognition systems. Still there exist many open conceptual problems that are related to the need of satisfying realistic constraints [7]. Modeling material synapses is possibly one of the most basic ones, the discrete case (and specifically the switchlike binary one) being of particular experimental [8] and technological interest [9]: recent experiments—at the single synapse resolution level—have shown that some synapses undergo potentiation or depression between a restricted number of discrete stable states through switchlike unitary events [8]. It is has been known for many years that the discreteness of synaptic efficacies makes the learning problem extraordinarily difficult [10]: even the task of finding binary synaptic weights for a single layer network (the binary perceptron) which classifies in two classes a given set of patterns is both NP-complete and computationally hard on average (as observed in classical numerical experiments). In spite of the fact that binary networks can, in principle, classify correctly an extensive number $p = \alpha n$ of random patterns with n binary synapses [11], in practice, there exists no known algorithm which is able to store exactly more than just a logarithmic number [12,13] as soon as a subexponential cut is put on their running time.

Here we present a distributed *message-passing* algorithm of statistical physics origin which is able to store efficiently an extensive number ($p = \alpha n$ with $\alpha > 0$) of random patterns in binary NN characterized by a wide range of different topologies. We consider single and multilayer networks with local connectivities of the neurons ranging from finite to extensive. The typical computational complexity of the algorithm is shown to scale roughly as $O(n^2 \log(n))$, that is almost *linearly* on the size of the input for an extensive number of patterns. This fact together with the parallel nature of the algorithm allows to easily find optimal synaptic weights for systems as large as $n = 10^6$ with α relatively close the critical value α_c above which perfect learning is no longer possible. From the algorithmic viewpoint, our solution to the binary learning problem should be seen as an example of solution of constraint satisfaction problems over dense *factor graphs* (a graphical representation of combinatorial constraints used in information theory [14,15]). As such, our results show how the recent progress in combinatorial optimization by statistical physics and message-passing techniques which have allowed to solve efficiently famous combinatorial problems like random K-satisfiability [16] or random graph Q-coloring [17], can be extended to other classes of problems in which constraints involve an extensive number of variables.

The NN models that we consider are composed of simple threshold units connected by binary weights $w_{j,k} = \pm 1$. For the sake of simplicity we consider two-layer networks with one output unit and with weights of the output layer that are fixed $w_{\ell,\mathrm{out}} = 1$ (see Fig. 1). Each of the K internal units is connected to c_ℓ inputs in either a treelike structure or in an overlapping way. We consider NN with connectivities ranging from finite to extensive; i.e., take $c_\ell = O(n^\epsilon)$ where $\epsilon \in [0, 1]$. In order to keep extensive the overall number of synapses we chose $K \propto \langle c \rangle^{-1}$, where $\langle c \rangle$ is the average connectivity. Under these conditions, the information theoretic bounds on the maximum number of bits which can be stored in the binary synapses are compatible with the exact storage of an extensive number of patterns ($p = \alpha n$, $\alpha > 0$) [6]. The output τ_ℓ of each inter-

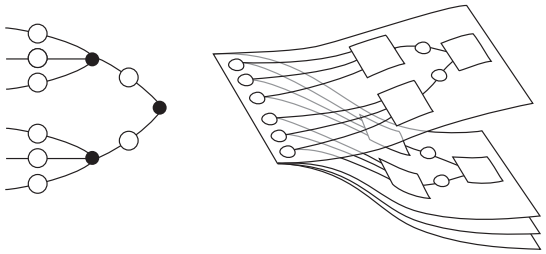

FIG. 1. A nonoverlapping two-layer network with six synapses (empty circles) and three threshold units (filled dots), and its corresponding factor graph for four patterns (right). The factor graph is composed of *variable* nodes (circles; indices *i, j, o* in the text) and *function* nodes (squares; indices *a, b* in the text); messages "travel" over the edges of the factor graph in both directions. Note that while synaptic weights have a unique corresponding variable node on the factor graph, each of the two auxiliary variable computing a partial threshold (*hidden units*), being pattern-dependent, must be replicated for every pattern on the factor graph.

nal unit is just the sign of the weighted sum of its inputs ξ_j minus some threshold, $\tau_\ell = \mathrm{sgn}(\sum_{j\in V(\ell)} w_{j,\ell}\xi_j - \gamma_\ell)$ where $V(\ell)$ is the set of inputs connected to unit ℓ. The overall output σ of the network is given by $\sigma(\vec{\xi}) = \mathrm{sgn}(\sum_{k=1}^{K} \tau_k - \gamma_{\mathrm{out}})$.

For $K = 1$ and $c = n$ we recover the binary perceptron, which is the elementary building block of many NN models. In the case of random input patterns, statistical mechanics and rigorous methods [4–6,11] have allowed us to study the typical behavior of this type of systems in the limit of large n. For instance, the storage capacity α_c has been computed for different finite values of K. Interestingly enough, the general scenario for binary networks is that while the storage capacity is indeed extensive the geometric structure of the space of solutions in the *satisfiable* region $\alpha < \alpha_c$ is rather complex [18]. Optimal synaptic configurations are typically far apart in Hamming distance and coexist with an exponential number of suboptimal configurations in which an extensive number of errors are made. Suboptimal states act as dynamical traps for learning algorithms [13]. Here we first show how the so-called *belief propagation* (BP) equations [14,15] (a variant of the Bethe approximation in statistical physics) can be applied on single problem instances, providing useful information such as the entropy of solutions, agreeing with statistical physics results in the large n limit [11]. Next we modify the equations by introducing a local reinforcement term which forces the system to polarize to a single optimal configuration of synaptic weights, effectively turning BP into a solver for this problem.

For simplicity let us fix a threshold value γ and first consider a perceptron with binary weights $w_i \in \{-1, 1\}$ for $i = 1, \ldots, n$. Given an input pattern ξ, the binary perceptron is an elementary device which just computes the function $f_{\mathbf{w}}(\xi) = \mathrm{sgn}(\sum_i w_i \xi_i - \gamma) \in \{-1, 1\}$. Patterns ξ are then classified by this perceptron by its output into the two preimage sets of the function $f_{\mathbf{w}}$. Given two sets of random patterns $\Xi_\pm$ we want to find vector of synaptic weights $\mathbf{w}$ such that $f_{\mathbf{w}}(\Xi_\pm) = \pm 1$. Consider the uniform probability space over the set W of all optimal assignments. We are interested in single marginals, that is, the probabilities $P(w_i = \pm 1)$ that the single synapses take a certain binary value. Under some weak correlations assumption, it is possible to write a close set of equations for these quantities. Such BP equations provide results which are believed to be exact in certain classes of problems defined over sparse factor graphs in which the size of loops tends to infinity with the problem size (e.g., in low density parity check codes [15]). In the case of problems corresponding to highly connected factor graphs (like the learning problem we discuss here) the validity of the BP approach relies on an apparently stronger condition, the so-called *clustering hypothesis*, in which the weak correlations condition arises from the weak effective interactions among variables. Until recently no algorithmic approach existed that allowed us to study the properties of a given problem instance of this type. Previous attempts in this direction were based on iterations of the mean-field Thouless-Anderson-Palmers (TAP) equations [19], which turn out to diverge in most cases. Recently BP has been used to study some densely connected problems on which it was shown that BP equations converge while TAP equations do not, even though the fixed point of the two is the same [20].

At variance with statistical mechanics results where the average over the patterns and the limit $n \to \infty$ are done, here we are interested in single problem instances. Thanks to the concentration of measure of the error-energy function, the so-called self-averaging property, we expect the quantities estimated by the equations on single problem instances to match the typical case as n gets large enough. Despite the fact that the approximations behind BP become exact only as n gets large, also at finite n the results provide very good approximations which can be used for algorithmic purposes (see Fig. 2). A large n expansion of the BP equations for the $K = 1$ and $\gamma = 0$ network learning problem read

$$m^t_{i\to a} = \tanh(h^t_{i\to a}), \tag{1}$$

$$u^t_{b\to i} = f\Big(\frac{1}{\sqrt{n}}\sum_{k\neq i}\xi^b_k m^t_{k\to b}, \frac{1}{n}\sum_{k\neq i}(m^t_{k\to b})^2\Big), \tag{2}$$

$$h^{t+1}_{i\to a} = \frac{1}{\sqrt{n}}\sum_{b\neq a}\xi^b_i u^t_{b\to i}, \tag{3}$$

where $f(a, b) = (\int_0^{+\infty} \exp(-\frac{(x-a)^2-a^2}{2(1-b)})dx)^{-1}$. At the fixed point $m_{i\to a}$ represents the mean value of w_i over the set of $W^{(a)}$ of synaptic weight configurations satisfying all patterns except pattern ξ^a. The quantity $h_{i\to a}$ is referred to as a *local field* that synapse i feels in absence of pattern a. The fixed point of these equations provides the information we

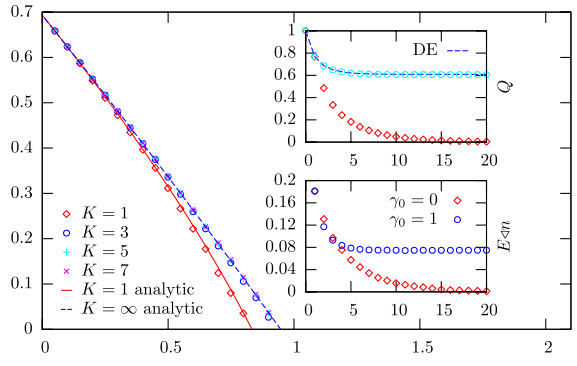

FIG. 2 (color online). BP entropy vs α for single problem instances of size $n = 3465$ for $K = 1, 3, 5, 7$. The analytic result for $K = 1$ and $K \gg 1$ for $n \to \infty$ are also plotted for comparison. The upper inset shows Q^t vs t of the analytical DE prediction (dashed line) vs simulations over a system of size $10^5 + 1$ at $\alpha = 0.6$ without reinforcement (data in perfect agreement to the prediction) and with reinforcement ($\gamma_0 = 0$). The bottom inset shows the fraction of errors E/n vs t for both cases. In the latter case we can see that $Q^t \to 0$ as the solution is reached.

are seeking for. Solving the equations by iteration proved itself to be an efficient technique, fully distributed, which is known as a *message-passing* method (the components of the vectors u and h can be thought of as messages running along edges of the factor graph; see Fig. 1). From the fixed point we may compute the list of all probability marginals $P(w_i = \pm 1)$ together with global quantities of interest such as the entropy (normalized logarithm of the size of the set W). As expected from the statistical mechanics results [11], the entropy monotonically decreases with α and vanishes at $\alpha_c \sim 0.833$ for large enough n. Similar results can be derived for multilayer networks as shown in Fig. 2. The BP equations can be adapted in a straightforward way to networks of arbitrary topology, even if the notation is slightly more encumbered. In general, this network is formed by connecting several perceptron subunits. The corresponding factor graph can be recovered trivially as in Fig. 1, by just replicating every perceptron for each pattern, and adding a set of auxiliary units to represent the output of every perceptron subunit of the network. It will suffice then to derive a set of slightly more general BP equations for the perceptron which we omit for the sake of brevity. We have studied analytically the dynamical behavior of the BP algorithm in the large n limit by the so-called *density evolution* (DE) technique (see, e.g., Ref. [20] for details on DE). In the upper inset of Fig. 2 we can see the comparison of numerical simulations of large single instances with the analytical prediction of the quantity $Q = 1 - \frac{1}{\alpha n^2} \sum_i \sum_a m_{i \to a}^2$ at every iteration step. In the spirit of Ref. [16], a way of using the information provided by BP is to "decimate" the problem. This approach is indeed feasible and leads to optimal assignments. However, here we focus on a much more efficient and fully distributed version [21] of the algorithm. The idea is to introduce an extra term into Eqs. (1)–(3) enforcing $h_i = \pm\infty$ at a fixed point, and use $w_i = \mathrm{sgn}(h_i)$ as a solution. This term is introduced stochastically (with probability 0 at the first iteration and probability 1 at $t = \infty$) to improve convergence. We replace Eq. (3) with Eqs. (4) and (5):

$$h_i^{t+1} = \frac{1}{\sqrt{n}} \sum_b \xi_i^b u_{b \to i}^t + \begin{cases} 0 & \text{w.p. } \gamma_t \\ h_i^t & \text{w.p. } 1 - \gamma_t \end{cases}, \tag{4}$$

$$h_{i \to a}^{t+1} = h_i^{t+1} - \frac{1}{\sqrt{n}} \xi_i^a u_{a \to i}^t. \tag{5}$$

We use $\gamma_t = \gamma_0^t$ for $0 \le \gamma_0 \le 1$ (though other choices are also possible). Choosing $\gamma_0 = 1$ clearly gives back the original BP set of equations, Eqs. (1)–(3). We note that a similar inertia term γh_i^t (constant γ) was introduced in Ref. [22], which would correspond to the average of the one in Eq. (4). Note also that the extra term for $\gamma_t = 0$ corresponds to adding an *external field* equal to the local field computed in the last step. Remembering that "fixing" a variable as in the standard decimation procedure is equivalent to adding an external field of infinite intensity, one can think of this procedure as a sort of *smooth decimation* in which *all* variables (not only the most polarized ones) get an external field, but the intensity is proportional to their polarization. Numerical experiments of learning randomly generated patterns have been carried out on systems of various sizes (up to $n = 10^6$), with different choices of K and with different topologies (overlapping and treelike). Some are reported in Fig. 3. An easy to use version of the code is made available at α[23]. It is not hard to think how the same algorithm could also be made effective in the presence of faulty contacts and heterogeneous discrete synaptic values (which need not to be identified *a priori* as the message-passing procedure, distributed over the same graph, could incorporate defects by

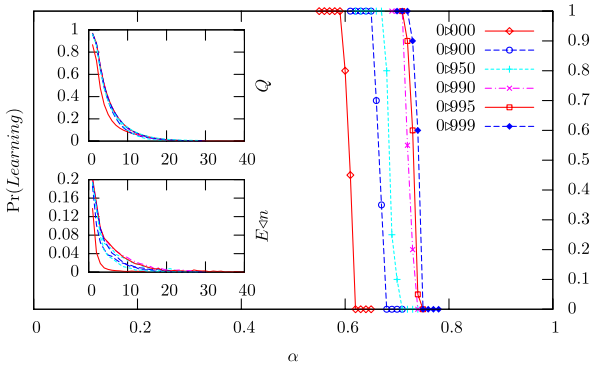

FIG. 3 (color online). Learning of αn pseudorandom patterns curves for the binary perceptron for different values of γ_0 ($n = 10^4 + 1$, 20 samples). The running time scales with γ_0 roughly as $1/(1 - \gamma_0)$. Inset: evolution of Q^t and E^t vs time t for various kinds of two-layer network topologies, i.e., $n = 3^7$, $\alpha = 0.5$, and $K \in \{3^0, 3^1, \ldots, 3^6\}$. Note that the number of errors E goes to 0 in all cases.

modifying accordingly the messages). Even for the limit case of continuous synapses the process converges to optimal solutions in a wide range of α.

Experiments have been performed using an improved version of Eqs. (1)–(3): Using further linearizations like in Ref. [20] one can obtain a new set of equations that are equivalent to Eqs. (1)–(3) up to an error of $O(n^{-1/2})$, having two main implementation advantages: memory requirements of just $O(n)$ (in addition to the set of patterns which amounts to αn^2 bits), and needing just $O(n)$ (slow) hyperbolic function computations in addition to $O(n^2)$ elementary (fast) floating point operations. BP equations can also be simplified by approximating $m_{k\to b}$ by m_k in Eqs. (1)–(3) (without correction terms), giving a simple closed expression in the quantities $\{m_i^t\}$. The resulting equation is not asymptotically equivalent to BP anymore [although the approximation itself has an error of $O(n^{-1/2})$ it participates in a sum of n terms], but nonetheless gives comparable (just slightly worse) algorithmic performances. Of particular interest are the corresponding equations for $\gamma_0 = 0$ (full reinforcement) which take a simple additive form if written in terms of the local fields h_i^t:

$$h_i^{t+1} = \sum_{t' \le t} \sum_b \frac{\xi_i^b}{\sqrt{n}} u_b^t \sim h_i^{\tau+1} = h_i^\tau + \frac{\xi_i^b}{\sqrt{n}} u_{b_\tau}^\tau \qquad (6)$$

where $u_b^s = f(\sum_{k\neq i} \frac{\xi_k^b}{\sqrt{n}} \tanh h_k^s, \frac{1}{n}\sum_{k\neq i} \tanh^2 h_k^s)$ and t scales as $\alpha n \tau$. By choosing at time τ one pattern ξ_{b_τ} from the set Ξ, Eq. (6) implements a sequential learning protocol, still leading to an extensive memory capacity (around $\alpha_{\max} \simeq .5$ for the binary perceptron). The simplicity of Eq. (6) represents a proof of concept of how highly nontrivial learning can take place by message passing between simple devices disposed over the network itself. This fact could shed some light on the biological treatment of information in neural systems [24].

This work has been supported by the EC, MTR 2002-00319 "STIPCO" and FP6 IST consortium "EVERGROW." We thank the ISI Foundation for hospitality.

[1] D. O. Hebb, *Organization of Behavior* (Wiley, New York, 1949); P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems* (MIT Press, Cambridge, MA, 2001).

[2] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood City, 1991).

[3] N. Brunel, V. Hakim, P. Isope, J.-P. Nadal, and B. Barbour, Neuron **43**, 745 (2004).

[4] G. J. Mitchison and R. M. Durbin, Biol. Cybern. **60**, 345 (1989).

[5] J. H. Kim and J. R. Roche, J. Comput. Syst. Sci. **56**, 223 (1998).

[6] A. Engel and C. van den Broeck, *Statistical Mechanics of Learning* (Cambridge University Press, Cambridge, England, 2001).

[7] S. B. Laughlin and T. J. Sejnowski, Science **301**, 1870 (2003); A. M. Zador and L. E. Dobrunz, Neuron **19**, 1 (1997).

[8] D. H. O'Connor, G. M. Wittenberg, and S. S.-H. Wang, Proc. Natl. Acad. Sci. U.S.A. **102**, 9679 (2005); C. C. H. Petersen, R. C. Malenka, R. A. Nicoll, and J. J. Hopfield, Proc. Natl. Acad. Sci. U.S.A. **95**, 4732 (1998).

[9] Ö. Türel, Lee J. Hoon, X. Ma, and K. Likharev, Int. J. Circuit Theory Appl. **32**, 277 (2004).

[10] A. L. Blum and R. L. Rivest, Neural Networks **5**, 117 (1992); E. Amaldi, *On the Complexity of Training Perceptrons*, edited by T. Kohonen, K. Makisara, O. Simula, and J. Kangas, in Artificial Neural Networks Vol. 1 (Elsevier Science Publisher, North-Holland, Amsterdam, 1991), pp. 55–60.

[11] W. Krauth and M. Mézard, J. Phys. (France) **50**, 3057 (1989); E. Barkai, D. Hansel, and H. Sompolinsky, Phys. Rev. A **45**, 4146 (1992); M. Mezard, J. Phys. A **22**, 2181 (1989); H. S. Seung, H. Sompolinsky, and N. Tishby, Phys. Rev. A **45**, 6056 (1992); E. Barkai and I. Kanter, Europhys. Lett.. **14**, 107 (1991); R. Penney and D. Sherrington, J. Phys. A **26**, 6173 (1993).

[12] M. Tsodyks, Mod. Phys. Lett. B **4**, 713 (1990); D. J. Amit and S. Fusi, Network **3**, 443 (1992); D. J. Amit, S. Fusi, Neural Comput. **6**, 957 (1994).

[13] H. Horner, Z. Phys. B **86**, 291 (1992).

[14] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, Cambridge, England 2003).

[15] J. S. Yedidia, W. T. Freeman, and Y. Weiss, IEEE Trans. Inf. Theory **51**, 2282 (2005).

[16] R. Monasson and R. Zecchina, Phys. Rev. Lett. **76**, 3881 (1996); M. Mézard and R. Zecchina, Phys. Rev. E **66**, 056126 (2002); M. Mézard, G. Parisi, and R. Zecchina, Science **297**, 812 (2002); A. Braunstein, M. Mézard, and R. Zecchina, Random Struct. Algorithms **27**, 201 (2005).

[17] R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina, Phys. Rev. Lett. **89**, 268701 (2002); A. Braunstein, R. Mulet, A. Pagnani, M. Weigt, and R. Zecchina, Phys. Rev. E **68**, 036702 (2003).

[18] S. Cocco, R. Monasson, and R. Zecchina, Phys. Rev. E **54**, 717 (1996).

[19] D. J. Thouless, P. W. Anderson, and R. G. Palmers, Philos. Mag. **35**, 593 (1977).

[20] Y. Kabashima, J. Phys. Soc. Jpn. **72**, 1645 (2003); Y. Kabashima, J. Phys. A **36**, 11 111 (2003); J. P. Neirotti and D. Saad, Europhys. Lett. **71**, 866 (2005).

[21] J. Chavas, C. Furtlehner, M. Mezard, and R. Zecchina, J. Stat. Mech. (2005) P11016.

[22] T. Murayama, Phys. Rev. E **69**, 035105(R) (2004); T. Hosaka and Y. Kabashima, cs.IT/0509086.

[23] http://www.ictp.it/~zecchina/SP.

[24] C. Baldassi, A. Braunstein, N. Brunel, C. Grosso, and R. Zecchina, (to be published).