Role of Synaptic Stochasticity in Training Low-Precision Neural Networks

Carlo Baldassi,[1,2,3] Federica Gerace,[2,4] Hilbert J. Kappen,[5] Carlo Lucibello,[2,4] Luca Saglietti,[2,4]
Enzo Tartaglione,[2,4] and Riccardo Zecchina[1,2,6]

[1] *Bocconi Institute for Data Science and Analytics, Bocconi University, Milano 20136, Italy*
[2] *Italian Institute for Genomic Medicine, Torino 10126, Italy*
[3] *Istituto Nazionale di Fisica Nucleare, Sezione di Torino, Torino 10129, Italy*
[4] *Department of Applied Science and Technology, Politecnico di Torino, Torino 10129, Italy*
[5] *Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour 6525 EZ Nijmegen, Netherlands*
[6] *International Centre for Theoretical Physics, Trieste 34151, Italy*

(Received 27 October 2017; revised manuscript received 19 March 2018; published 29 June 2018)

Stochasticity and limited precision of synaptic weights in neural network models are key aspects of both biological and hardware modeling of learning processes. Here we show that a neural network model with stochastic binary weights naturally gives prominence to exponentially rare dense regions of solutions with a number of desirable properties such as robustness and good generalization performance, while typical solutions are isolated and hard to find. Binary solutions of the standard perceptron problem are obtained from a simple gradient descent procedure on a set of real values parametrizing a probability distribution over the binary synapses. Both analytical and numerical results are presented. An algorithmic extension that allows to train discrete deep neural networks is also investigated.

DOI: 10.1103/PhysRevLett.120.268103

Learning can be regarded as an optimization process over the connection weights of a neural network. In nature, synaptic weights are known to be plastic, low precision, and unreliable, and it is an interesting issue to understand if this stochasticity can help learning or if it is an obstacle. The debate about this issue has a long history and is still unresolved (see Ref. [1] and references therein). Here, we provide quantitative evidence that the stochasticity associated with noisy low-precision synapses can drive elementary supervised learning processes towards a particular type of solutions which, despite being rare, is robust to noise and generalizes well—two crucial features for learning processes.

In recent years, multilayer (*deep*) neural networks have gained prominence as powerful tools for tackling a large number of cognitive tasks [2]. In a K-class classification task, neural network architectures are typically trained as follows. For any input $x \in \mathcal{X}$ (the input space $\mathcal{X}$ typically being a tensor space) and for a given set of parameters W called *synaptic weights*, the network defines a probability density function $P(y|x,W)$ over the K possible outcomes. This is done through composition of affine transformations involving the synaptic weights W, elementwise nonlinear operators, and finally a softmax operator that turns the outcome of previous operations into a probability density function [3]. The weights W are adjusted in a supervised learning scenario using a training set $\mathcal{D}$ of M known input-output associations, $\mathcal{D} = \{(x^\mu, y^\mu)\}_{\mu=1}^{M}$. The learning problem is reframed into the problem of maximizing a log-likelihood $\tilde{\mathcal{L}}(W)$ over the synaptic weights W:

$$\max_W \tilde{\mathcal{L}}(W) \coloneqq \sum_{(x,y)\in\mathcal{D}} \log P(y|x,W). \tag{1}$$

The maximization problem is approximately solved using variants of the stochastic gradient descent (SGD) procedure over the loss function $-\tilde{\mathcal{L}}(W)$ [4]. In a Bayesian approach instead, one is interested in computing the posterior distribution $P(W|\mathcal{D}) \propto P(\mathcal{D}|W)P(W)$, where $P(W)$ is some prior over the weights W. In deep networks, unfortunately, the exact computation of $P(W|\mathcal{D})$ is typically infeasible, and various approximated approaches have been proposed [5–7].

Shallow neural network models, such as the perceptron model for binary classification, are amenable to analytic treatment while exposing a rich phenomenology. They have attracted great attention from the statistical physics community for many decades [8–16]. In the perceptron problem, we have binary outputs $y \in \{-1,+1\}$, while inputs x and weights W are N-component vectors. Under some statistical assumptions on the training set $\mathcal{D}$ and for large N, single variable marginal probabilities $P(W_i|\mathcal{D})$ can be computed efficiently using belief propagation [17–19]. The learning dynamics has also been analyzed, in particular, in the online learning setting [11,20]. In a slightly different perspective, the perceptron problem is often framed as the task of minimizing the error-counting Hamiltonian

$$\min_W \mathcal{H}(W) \coloneqq \sum_{(x,y)\in\mathcal{D}} \Theta\left(-y\sum_{i=1}^{N} W_i x_i\right), \tag{2}$$

where $\Theta(x)$ is the Heaviside step function, $\Theta(x) = 1$ if $x > 0$, and 0 otherwise. As a constraint satisfaction problem, it is said to be satisfiable (SAT) if zero energy [i.e., $\mathcal{H}(W) = 0$] configurations exist, and unsatisfiable (UNSAT) otherwise. We call *solutions* such configurations. Statistical physics analysis assuming random and uncorrelated $\mathcal{D}$ shows a sharp threshold at a certain $\alpha_c = M/N$, when N grows large, separating a SAT phase from an UNSAT one. Moreover, restricting the synaptic space to binary values $W_i = \pm 1$ leads to a more complex scenario: most solutions are essentially isolated and computationally hard to find [13,21]. Some efficient algorithms do exist though [12,22] and generally land in a region dense of solutions. This apparent inconsistency has been solved through a large deviation analysis which revealed the existence of subdominant and dense regions of solutions [14,23]. This analysis introduced the concept of local entropy [14], which subsequently led to other algorithmic developments [24–26] (see, also, Ref. [27] for a related analysis).

In the generalization perspective, solutions within a dense region may be loosely considered as representative of the entire region itself and, therefore, act as better pointwise predictors than isolated solutions, since the optimal Bayesian predictor is obtained averaging all solutions [14].

Here, we propose a method to solve the binary perceptron problem (2) through a relaxation to a distributional space. We introduce a perceptron problem with stochastic discrete weights and show how the learning process is naturally driven towards dense regions of solutions, even in the regime in which they are exponentially rare compared to the isolated ones. In perspective, the same approach can be extended to the general learning problem (1), as we will show.

Denote with $Q_\theta(W)$ a family of probability distributions over W parametrized by a set of variables θ. Consider the following problem:

$$\max_\theta \mathcal{L}(\theta) \coloneqq \sum_{(x,y)\in\mathcal{D}} \log \mathbb{E}_{W\sim Q_\theta} P(y|x,W). \tag{3}$$

Here, $\mathcal{L}(\theta)$ is the log-likelihood of a model where for each training example $(x, y) \in \mathcal{D}$ the synaptic weights are independently sampled according to $Q_\theta(W)$. Within this scheme, two class predictors can be devised for any input x: $\hat{y}_1(x) = \mathrm{argmax}_y P(y|x, \hat{W})$, where $\hat{W} = \mathrm{argmax}_W Q_\theta(W)$, and $\hat{y}_2(x) = \mathrm{argmax}_y \int dW P(y|x, W) Q_\theta(W)$. In this Letter, we will analyze the quality of the training error given by the first predictor. Generally, dealing with problem (3) is more difficult than dealing with problem (1), since it retains some of the difficulties of the computation of $P(W|\mathcal{D})$. Also, notice that for any maximizer $W^\star$ of problem (1) we have that $\delta(W - W^\star)$ is a maximizer of problem (3) provided that it belongs to the parametric family, as can be shown using Jensen's inequality. Problem (3) is a "distributional" relaxation of problem (1).

Optimizing $\mathcal{L}(\theta)$ instead of $\tilde{\mathcal{L}}(W)$ may seem an unnecessary complication. In this Letter, we argue that there are two reasons for dealing with this kind of task. First, when the configuration space of each synapse is restricted to discrete values, the network cannot be trained with SGD procedures. The problem, while being very important for computational efficiency and memory gains, has been tackled only very recently [5,28]. Since variables θ typically lie in a continuous manifold instead, standard continuous optimization tools can be applied to $\mathcal{L}(\theta)$. Also, the learning dynamics on $\mathcal{L}(\theta)$ enjoys some additional properties when compared to the dynamics on $\tilde{\mathcal{L}}(W)$. In the latter case, additional regularizers, such as dropout and L_2 norm, are commonly used to improve generalization properties [4]. The SGD in the θ space instead already incorporates the kind of natural regularization intrinsic in the Bayesian approach and the robustness associated to high local entropy [14]. Here, we make a case for these arguments by a numerical and analytical study of the proposed approach for the binary perceptron. We also present promising preliminary numerical results on deeper networks.

Learning for the stochastic perceptron.—Following the above discussion, we now introduce our binary stochastic perceptron model. For each input x presented, N synaptic weights $W = (W_1, \ldots, W_N)$, $W_i \in \{-1, +1\}$ are randomly extracted according to the distribution

$$Q_m(W) = \prod_{i=1}^N \left(\frac{1+m_i}{2} \delta_{W_i,+1} + \frac{1-m_i}{2} \delta_{W_i,-1} \right), \tag{4}$$

where $\delta_{a,b}$ is the Kronecker delta symbol. We will refer to the set $m = (m_i)_i$, where $m_i \in [-1, 1] \quad \forall\, i$, as the magnetizations or the control parameters. We choose the probability $P(y|x, W)$ on the class $y \in \{-1, +1\}$ for a given input x as follows:

$$P(y|x, W) = \Theta\left(y \sum_{i=1}^N W_i x_i \right). \tag{5}$$

While other possibilities for $P(y|x, W)$ could be considered, this particular choice is directly related to the form of the Hamiltonian in problem (2), which we ultimately aim to solve. Given a training set $\mathcal{D} = \{(x^\mu, y^\mu)\}_{\mu=1}^M$, we can then compute the log-likelihood function of Eq. (3), with the additional assumption that N is large and the central limit theorem applies. It reads

$$\mathcal{L}(m) = \sum_{(x,y)\in\mathcal{D}} \log H\left(-\frac{y \sum_i m_i x_i}{\sqrt{\sum_i (1 - m_i^2) x_i^2}} \right), \tag{6}$$

where $H(x) \coloneqq \int_x^\infty dz e^{-z^2/2}/\sqrt{2\pi}$. Minimizing $-\mathcal{L}(m)$ instead of finding the solutions of problem (2) allows us to use the simplest method for approximately solving

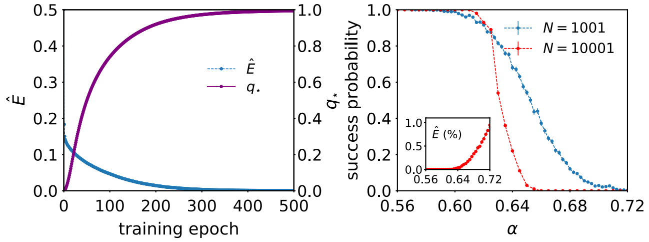

FIG. 1. (Left) The training error and the squared norm against the number of training epochs for $\alpha = 0.55$ and $N = 10001$ averaged over 100 samples. (Right) Success probability in the classification task as a function of the load α for networks of size $N = 1001, 10001$ averaging 1000 and 100 samples, respectively. In the inset, we show the average training error at the end of GD as a function of α.

continuous optimization problems, the gradient descent (GD) algorithm:

$$m_i^{t+1} \leftarrow \mathrm{clip}(m_i^t + \eta \partial_{m_i} \mathcal{L}(m^t)). \tag{7}$$

We could have adopted the more efficient SGD approach; however, in our case, simple GD is already effective. In the last equation, η is a suitable learning rate and $\mathrm{clip}(x) \coloneqq \max(-1, \min(1, x))$ is applied elementwise. The parameters are randomly initialized to small values, $m_i^0 \sim \mathcal{N}(0, N^{-1})$. At any epoch t in the GD dynamics, a binarized configuration $\hat{W}_i^t = \mathrm{sgn}(m_i^t)$ can be used to compute the training error $\hat{E}^t = (1/M)\mathcal{H}(\hat{W}^t)$. We consider a training set $\mathcal{D}$ where each input component x_i^μ is sampled uniformly and independently in $\{-1, 1\}$ (with this choice, we can set $y^\mu = 1 \quad \forall\ \mu$ without loss of generality). The evolution of the network during GD is shown in Fig. 1. The training error goes progressively to zero while the mean squared norm of the control variables $q_\star^t = (1/N)\sum_i (m_i^t)^2$ approaches 1. Therefore, the distribution Q_m concentrates around a single configuration as the training is progressing. This natural flow is similar to the annealing of the coupling parameter manually performed in local entropy inspired algorithms [25,26]. We also show in Fig. 1 the probability over the realizations of $\mathcal{D}$ of finding a solution of the binary problem as a function of the load $\alpha = M/N$. The algorithmic capacity of GD is approximately $\alpha_{\mathrm{GD}} \approx 0.63$. This value has to be compared to the theoretical capacity $\alpha_c \approx 0.83$, above which there are almost surely no solutions [9] and state-of-the-art algorithms based on message passing heuristics for which we have a range of capacities $\alpha_{\mathrm{MP}} \in [0.6, 0.74]$ [12,22,29]. Therefore, GD reaches loads only slightly worse than those reached by much more fine-tuned algorithms, a surprising result for such a simple procedure. Also, for α slightly above α_{GD}, the training error remains comparably low, as shown in Fig. 1. In our experiments, most variants of the GD procedure of Eq. (7) performed just as well: e.g., SGD or GD computed on the fields $h_i^t = \tanh^{-1}(m_i^t)$ rather than the magnetizations [30]. Other update rules for the control parameters can be derived as multiple passes of online Bayesian learning [31,32]. Variations of rule (7) towards biological plausibility are discussed in the Supplemental Material [33].

Deep networks.—We applied our framework to deep neural networks with binary stochastic weights and sgn activation functions. Using an uncorrelated neuron approximation, as in Ref. [6], we trained the network using the standard SGD algorithm with backpropagation. We give the details in the Supplemental Material [33]. On the MNIST benchmark problem [45], using a network with three hidden layers, we achieved ~1.7% test error, a very good result for a network with binary weights and activations and with no convolutional layers [46]. No other existing approach to the binary perceptron problem has been extended yet to deeper settings.

Statistical mechanics analysis.—We now proceed with the analytical investigation of the equilibrium properties of the stochastic perceptron, which partly motivates the good performance of the GD dynamics. The starting point of the analysis is the partition function

$$Z = \int_\Omega \prod_i dm_i \delta\left(\sum_i m_i^2 - q_\star N\right) e^{\beta \mathcal{L}(m)}, \tag{8}$$

where $\Omega = [-1, 1]^N$, β is an inverse temperature, and we constrained the squared norm to $q_\star N$ in order to mimic the natural flow of $q_\star^t$ in the training process. The dependence on the training set $\mathcal{D}$ is implicit in last equation. We shall denote with $\mathbb{E}_\mathcal{D}$ the average over a training set with input and output components independently and uniformly distributed in $\{-1, 1\}$. We investigate the average properties of the system for large N and fixed load $\alpha = M/N$ using the replica method in the replica symmetric (RS) ansatz [47]. Unfortunately the RS solution becomes locally unstable for very large β. Therefore, instead of taking the infinite β limit to maximize the likelihood, we will present the results obtained for β large but still in the RS region. The details of the free energy calculation and of the stability check can be found in the Supplemental Material [33].

Energy of the binarized configuration.—We now analyze some properties of the mode of the distribution $Q_m(W)$, namely, $\hat{W}_i = \mathrm{sgn}(m_i)$, that we call the binarized configuration (BC). The average training error per pattern is

$$E = \lim_{N \to \infty} \frac{1}{\alpha N} \mathbb{E}_\mathcal{D} \left[\sum_{(x,y) \in \mathcal{D}} \left\langle \Theta\left(-y \sum_i \mathrm{sgn}(m_i) x_i \right) \right\rangle \right], \tag{9}$$

where $\langle \bullet \rangle$ is the thermal average over m according to the partition function (8), which implicitly depends on $\mathcal{D}$, $q_\star$, and β. The last equation can be computed analytically within the replica framework (see the Supplemental Material [33]). In Fig. 2 (left), we show that for large β

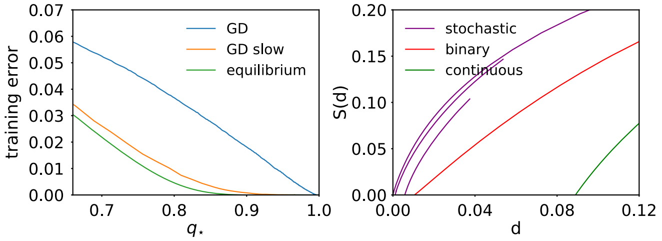

FIG. 2. (Left) Energy of the binarized configuration versus the control variable $q_\star$. We show the equilibrium prediction of Eq. (9) and numerical results from the GD algorithm and a GD algorithm variant where after each update we rescale the norm of m to $q_\star$ until convergence before moving to the next value of $q_\star$ according to a certain schedule. The results are averaged over 20 random realizations of the training set with $N = 10001$. (Right) Entropy of binary solutions at fixed distance d from the BCs of the spherical, binary, and stochastic perceptron ($q_\star = 0.7$, 0.8, and 0.9 from bottom to top) at thermodynamic equilibrium. In both figures, $\alpha = 0.55$, also $\beta = 20$ for the stochastic perceptron and $\beta = \infty$ for the spherical and binary ones.

the BC becomes a solution of the problem when $q_\star$ approaches 1. This is compared to the values of the training error obtained from GD dynamics at corresponding values of $q_\star$ and a modified GD dynamics where we let the system equilibrate at fixed $q_\star$. The latter case, although we are at finite N and we are considering a dynamical process that could suffer from the presence of local minima, is in reasonable agreement with the equilibrium result of Eq. (9).

Geometrical structure of the solution space.—Most solutions of the binary perceptron problem are isolated [13], while a subdominant but still exponentially large number belongs to a connected dense region [14]. Solutions in the dense region are the only ones that are algorithmically accessible. Here, we show that the BCs of the stochastic binary perceptron typically belong to the dense region, provided $q_\star$ is high enough. To prove this, we count the number of solutions at a fixed Hamming distance d from the typical BC (this corresponds to fixing an overlap $p = 1\text{–}2d$). Following the approach of Franz and Parisi [48], we introduce the constrained partition function

$$\mathcal{Z}(d, m) = \sum_{W} \prod_{(x,y)\in\mathcal{D}} \Theta\left(y \sum_i W_i x_i\right) \times \delta\left(N(1-2d) - \sum_i \mathrm{sgn}(m_i) W_i\right), \qquad (10)$$

where the sum is over the $\{-1, +1\}^N$ binary configurations. The Franz-Parisi entropy $\mathcal{S}(d)$ is then given by

$$\mathcal{S}(d) = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathcal{D}} \langle \log \mathcal{Z}(d, m) \rangle. \qquad (11)$$

We show how to compute $\mathcal{S}(d)$ in the Supplemental Material [33]. In Fig. 2 (right), we compare $\mathcal{S}(d)$ for the stochastic perceptron with the analogous entropies obtained substituting the expectation $\langle \bullet \rangle$ over m in Eq. (11) with a uniform sampling from the solution space of the spherical (the model of Ref. [8]) and the binary (as in Ref. [13]) perceptron. The distance gap between the BC and the nearest binary solutions [i.e., the value of the distance after which $\mathcal{S}(d)$ becomes positive] vanishes as $q_\star$ is increased: in this regime, the BC belongs to the dense cluster, and we have an exponential number of solutions at any distance $d > 0$. Typical binary solutions and binarized solutions of the continuous perceptron are isolated instead [finite gap corresponding to $S(d) = 0$ at small distances]. In the Supplemental Material [33], we provide additional numerical results on the properties of the energetic landscape in the neighborhood of different types of solutions, showing that solutions in flatter basins achieve better generalization than those in sharp ones.

Conclusions.—Our analysis shows that stochasticity in the synaptic connections may play a fundamental role in learning processes by effectively reweighting the error loss function, enhancing dense robust regions, suppressing narrow local minima, and improving generalization.

The simple perceptron model allows us to derive analytical results as well as to perform numerical tests. Moreover, as we show in the Supplemental Material [33], when considering discretized priors, there exists a connection with the dropout procedure which is popular in modern deep learning practice. However, the most promising immediate application is in the deep learning scenario, where this framework can be extended adapting the tools developed in Refs. [6,7] and where we already achieved state-of-the-art results in our preliminary investigations.

Hopefully, the general mechanism shown here can also help shed some light on biological learning processes, where the role of low precision and stochasticity is still an open question. Finally, we note that this procedure is not limited to neural network models; for instance, the application to constraint satisfaction problems is straightforward.

C. B., H. J. K., and R. Z. acknowledge ONR Grant No. N00014-17-1-2569.

[1] H. Sebastian Seung, Learning in spiking neural networks by reinforcement of stochastic synaptic transmission, Neuron **40**, 1063 (2003).

[2] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

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

[4] L. Bottou, F. E. Curtis, and J. Nocedal, Optimization methods for large-scale machine learning, arXiv:1606.04838.

[5] D. Soudry, I. Hubara, and R. Meir, Expectation backpropagation: Parameter-free training of multilayer neural networks with real and discrete weights, in *Proceedings of Neural Information Processing Systems 2014*, pp. 1–9.

[6] J. M. Hernández-Lobato and R. P. Adams, Probabilistic backpropagation for scalable learning of Bayesian neural networks, J. Mach. Learn. Res. **37**, 1 (2015).

[7] O. Shayar, D. Levi, and E. Fetaya, Learning discrete weights using the local reparameterization trick, arXiv:1710.07739.

[8] E. Gardner, The space of interactions in neural network models, J. Phys. A: Math. Gen. **21**, 257 (1988).

[9] W. Krauth and M. Mézard, Storage capacity of memory networks with binary couplings, J. Phys. France **50**, 3057 (1989).

[10] M. Opper and O. Winther, A Mean Field Approach to Bayes Learning in Feed-Forward Neural Networks, Phys. Rev. Lett. **76**, 1964 (1996).

[11] S. Solla and O. Winther, Optimal perceptron learning: An online Bayesian approach, in *On-Line Learning in Neural Networks*, edited by D. Saad (Cambridge University Press, Cambridge, England, 1998), pp. 1–20.

[12] A. Braunstein and R. Zecchina, Learning by Message Passing in Networks of Discrete Synapses, Phys. Rev. Lett. **96**, 030201 (2006).

[13] H. Huang and Y. Kabashima, Origin of the computational hardness for learning with binary synapses, Phys. Rev. E **90**, 052813 (2014).

[14] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, Subdominant Dense Clusters Allow for Simple Learning and High Computational Performance in Neural Networks with Discrete Synapses, Phys. Rev. Lett. **115**, 128101 (2015).

[15] S. Franz and G. Parisi, The simplest model of jamming, J. Phys. A **49**, 145001 (2016).

[16] S. Cocco, R. Monasson, and R. Zecchina, Analytical and numerical study of internal representations in multilayer neural networks with binary weights, Phys. Rev. E **54**, 717 (1996).

[17] M. Mézard, The space of interactions in neural networks: Gardner's computation with the cavity method, J. Phys. A **22**, 2181 (1989).

[18] G. Parisi, M. Mézard, and M. A. Virasoro, *Spin Glass Theory and Beyond* (World Scientific Singapore, 1987).

[19] A. Montanari and M. Mézard, *Information, Physics and Computation* (Oxford University Press, New York, 2009).

[20] D. Saad, *On-Line Learning in Neural Networks* (Cambridge University Press, Cambridge, England, 1998).

[21] L. Zdeborová and M. Mézard, Constraint satisfaction problems with isolated solutions are hard, J. Stat. Mech. (2008) P12004.

[22] C. Baldassi, A. Braunstein, N. Brunel, and R. Zecchina, Efficient supervised learning in networks with binary synapses, Proc. Natl. Acad. Sci. U.S.A. **104**, 11079 (2007).

[23] C. Baldassi, F. Gerace, C. Lucibello, L. Saglietti, and R. Zecchina, Learning may need only a few bits of synaptic precision, Phys. Rev. E **93**, 052313 (2016).

[24] C. Baldassi, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, Local entropy as a measure for sampling solutions in constraint satisfaction problems, J. Stat. Mech. (2016) 023301.

[25] C. Baldassi, C. Borgs, J. T. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes, Proc. Natl. Acad. Sci. U.S.A. **113**, E7655 (2016).

[26] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. T. Chayes, L. Sagun, and R. Zecchina, Entropy-SGD: Biasing gradient descent into wide valleys, arXiv:1611.01838.

[27] A. Braunstein, L. Dall'Asta, G. Semerjian, and L. Zdeborová, The large deviations of the whitening process in random constraint satisfaction problems, J. Stat. Mech. (2016) 053401.

[28] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, Binarized neural networks: Training deep neural networks with weights and activations constrained to $+1$ or -1, arXiv:1602.02830.

[29] C. Baldassi and A. Braunstein, A max-sum algorithm for training discrete neural networks, J. Stat. Mech. (2015) P08008.

[30] This has the advantage that it does not require clipping.

[31] M. Opper and O. Winther, A Bayesian approach to on-line learning, in *On-Line Learning in Neural Networks*, edited by D. Saad (Cambridge University Press, Cambridge, England, 1998), pp. 363–378.

[32] T. P. Minka, Expectation propagation for approximate Bayesian inference, in *Proceedings of Uncertainty in Artificial Intelligence (UAI), 2001*, pp. 362–369.

[33] See the Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.120.268103, which contains further discussions on biologically plausible algorithms, details of the replica analysis, and algorithmic extensions to deep networks. It also provides the additional Refs. [34–44].

[34] C. Baldassi, Generalization learning in a perceptron with binary synapses, J. Stat. Phys. **136**, 902 (2009).

[35] C. Baldassi and R. Zecchina, Efficiency of quantum vs. classical annealing in nonconvex learning problems, Proc. Natl. Acad. Sci. U.S.A. **115**, 1457 (2018).

[36] T. M. Bartol, C. Bromer, J. P. Kinney, M. A. Chirillo, J. N. Bourne, K. M. Harris, and T. J. Sejnowski, Hippocampal spine head sizes are highly precise, Cold Spring Harbor Laboratory, 2015, DOI: 10.1101/016329.

[37] A. Engel, *Statistical Mechanics of Learning* (Cambridge University Press, Cambridge, England, 2001).

[38] E. Gardner and B. Derrida, Optimal storage properties of neural network models, J. Phys. A: Math. Gen. **21**, 271 (1988).

[39] H. Horner, Dynamics of learning for the binary perceptron problem, Z. Phys. B **86**, 291 (1992).

[40] Y. Loewenstein and H. Sebastian Seung, Operant matching is a generic outcome of synaptic plasticity based on the covariance between reward and neural activity, Proc. Natl. Acad. Sci. U.S.A. **103**, 15224 (2006).

[41] D. H. O'Connor, G. M. Wittenberg, and S. S.-H. Wang, Graded bidirectional synaptic plasticity is composed of switch-like unitary events, Proc. Natl. Acad. Sci. U.S.A. **102**, 9679 (2005).

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. **15**, 1929 (2014).

[43] L. Wan, M. Zeiler, S. Zhang, Y. Lecun, and R. Fergus, Regularization of neural networks using dropconnect, in

Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013.

[44] D. Yuret, Knet: Beginning deep learning with 100 lines of Julia, in *Proceedings of Machine Learning Systems Workshop at NIPS 2016*.

[45] Y. LeCun, C. Cortes, and C. J. C. Burges, MNIST handwritten digit database, AT&T Labs, http://yann.lecun.com/exdb/mnist, 2010.

[46] C. Baldassi, C. Borgs, J. T. Chayes, C. Lucibello, L. Saglietti, E. Tartaglione, and R. Zecchina (to be published).

[47] M. Mézard and A. Montanari, *Information, Physics, and Computation* (Oxford University Press, New York, 2009).

[48] S. Franz and G. Parisi, Recipes for metastable states in spin glasses, J. Phys. **5**, 1401 (1995).