J Stat Phys (2009) 136: 902–916
DOI 10.1007/s10955-009-9822-1

Generalization Learning in a Perceptron with Binary Synapses

Carlo Baldassi

Received: 15 December 2008 / Accepted: 8 September 2009 / Published online: 19 September 2009

Abstract We consider the generalization problem for a perceptron with binary synapses, implementing the Stochastic Belief-Propagation-Inspired (SBPI) learning algorithm which we proposed earlier, and perform a mean-field calculation to obtain a differential equation which describes the behaviour of the device in the limit of a large number of synapses N. We show that the solving time of SBPI is of order $N\sqrt{\log N}$, while the similar, well-known clipped perceptron (CP) algorithm does not converge to a solution at all in the time frame we considered. The analysis gives some insight into the ongoing process and shows that, in this context, the SBPI algorithm is equivalent to a new, simpler algorithm, which only differs from the CP algorithm by the addition of a stochastic, unsupervised meta-plastic reinforcement process, whose rate of application must be less than $\sqrt{2/(\pi N)}$ for the learning to be achieved effectively. The analytical results are confirmed by simulations.

Keywords Perceptron · Binary synapses · Learning · Online · Generalization · SBPI

1 Introduction

The perceptron was first proposed by Rosenblatt [17] as an extremely simplified model of a neuron, consisting in a number N of input lines, each one endowed with a weight coefficient (representing the individual synaptic conductances), all of which converge in a central unit (representing the soma) with a single output line (the axon). Typically, the output is computed from a threshold function of the weighted sum of the inputs, and the time in the model is discretized, so that, at each time step, the output does only depend on the input at that time step. The unit can adapt its behaviour over time by modifying the synaptic weights (and possibly the output threshold), and thus it can undergo a learning (or

C. Baldassi (✉)
Dipartimento di Fisica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
e-mail: carlo.baldassi@polito.it

C. Baldassi
Institute for Scientific Interchange Foundation, Viale Settimio Severo 65, 10133 Torino, Italy

memorizing) process. In this paper, we only consider the case in which the learning process is "supervised", i.e. in which there is some feedback from the outside, typically in the form of an "error signal", telling the unit that the output is wrong, as opposed to "unsupervised" learning, in which no feedback is provided to the unit.

Despite its simplicity, the perceptron model is very powerful, being able to process its inputs in parallel, to retain an extensive amount of information, and to plastically adapt its output over time in an on-line fashion. Furthermore, it displays a highly non-trivial behaviour, so that much research has been devoted to the study of its analytical properties and of the optimal learning strategies in different contexts (see e.g. [1, 6, 7, 9, 12, 13, 19] and references therein). In the supervised case, there are typically two scenarios: the first, which we will call "classification" problem in the rest of this paper, is defined by a given set of input-output associations that the unit must learn to reproduce without errors, while the second, which we will call "generalization" problem, is defined by a given input-output rule that the unit must learn to implement as closely as possible. Here, we will mainly focus our attention on this last problem.

Furthermore, we will restrict to the case in which the synaptic weights are assumed to be binary variables. Binary models are inherently simpler to implement and more robust over time against noise with respect to models in which the synaptic weights are allowed to vary over a continuous set of values, while having a comparable information storage capacity [6, 9, 13]; furthermore, some recent experimental results [15, 16], as well as some arguments from theoretical studies and computer simulations [2, 3, 10, 14], suggest that binary-synapses models could also be more relevant than continuous ones as neuronal models exhibiting long term plasticity. However, from an algorithmic point of view, learning is much harder in models with binary synapses than in models with continuous synapses: in the worst-case scenario, the classification learning problem is known to be NP-complete for binary weights [4], while it is easy to solve it effectively with continuous weights [6]. Even in the case of random, uncorrelated inputs-output associations, the solution space of the classification problem in the binary case is in a broken-symmetry phase, while in the continuous case it is not [13], implying that the learning strategies which successfully solve the learning problem in the latter case are normally not effective in the former.

Despite these difficulties, an efficient, easily implementable, on-line learning algorithm can be devised which solves efficiently the binary classification problem in the case of random, uncorrelated input stimuli [1]. Such an algorithm was originally derived from the standard Belief Propagation algorithm [5, 11, 20], and hence named 'Stochastic Belief Propagation-Inspired' (SBPI). The SBPI algorithm makes an additional requirement on the model, namely, that each synapse in the device, besides the weight, has an additional hidden, discretized internal state; transitions between internal states may be purely meta-plastic, meaning that the synaptic strength does not necessarily change in the process, but rather that the plasticity of the synapse does. The SBPI learning rules and hidden states requirements are the same as those of the well known clipped perceptron algorithm (CP, see e.g. [18]), the only difference being an additional, purely meta-plastic rule, which is only applied if the answer given by the device is correct, but such that a single variable flip would result in a classification error.

The SBPI algorithm was derived and tested in the context of the classification problem: in such scheme, all the input patterns are extracted from a given pattern set, randomly generated before the learning session takes place, and presented to the student device repeatedly, the outcome being compared to the desired one, until no classification errors are made any more. Since the analytical treatment of the learning process is awkward in such case, due to the temporal correlations emerging in the input patterns as a consequence of the

repeated presentations, we could only test the SBPI algorithm performance by simulations, and compare it to that of other similar algorithms, such as the CP algorithm and the cascade model [7]. It turned out that the additional learning rule which distiguishes the SBPI algorithm from the CP algorithm is essential to SBPI's good performance, and that there exists an optimal number of parameters for both the number of internal hidden states per synapse and for the rate of application of the novel learning rule.

In order to understand the reason for the SBPI new rule's effectiveness, it is necessary to give an analytical description of the learning process under the CP and SBPI learning rules; to this end, we consider here the problem of generalization from examples: in such scheme, the input patterns are generated afresh at each time step, and the goal is to learn a linearly separable classification rule, provided by a teacher device. Perfect learning is achieved if the student's synapses match the teacher's ones. Using a teacher unit identical to the student to define the goal input-output function ensures that a solution to the problem exists and is unique, but we also briefly consider the case of a non-perfect-learnable rule, provided by a continuous-weights teacher perceptron. The learning process in this case is much easier to treat analytically than in the classification case, since the input patterns are not temporally correlated, and the dynamical equations for this system can be derived by a mean field calculation for the case of a learnable rule. This problem is in fact easier to address than the classification problem, and optimal algorithms can be found which solve it in the binary synapses case as well (see e.g. [8, 18, 19]); however, such algorithms are not suitable to be considered as candidates for biological models for online learning, being either too complex or requiring to perform all intermediate operations with an auxiliary device with continuous synaptic weights.

The resulting differential equation set that we obtained gives some insight on the learning dynamics and about the reason for SBPI's effectiveness, and allows for a further simplification of the SBPI algorithm, yielding an even more attractive model of neuronal unit, both from the point of view of biological feasibility and of hardware manufacturing design simplicity. With a special choice for the parameters, the solution to the equation set is simple enough to be studied analytically and demonstrate that the algorithm converges in a number of time steps which goes as $N\sqrt{\log N}$. All the results are confirmed by simulations.

The outline of the rest of this paper is as follows: in Sects. 2 and 3 we define in detail the learning algorithm and the generalization problem, respectively. In Sects. 4 and 5 we derive the mean-field dynamics for the CP and SBPI algorithms, and in Sect. 6 we derive the set of continuous differential equations which describes the process in the $N \to \infty$ limit and exhibit a solution. In Sect. 7 we consider a special case in which the equation set can be simplified and derive some analytical results on convergence time in such case. In Sect. 8 we consider the case of bounded hidden states. In Sect. 9 we briefly consider the case of a non learnable rule. In Sect. 10 we discuss the simplified algorithm derived in Sect. 5. We summarize our results in the last section.

2 The SBPI Learning Algorithm

The device we consider is a binary perceptron with N synapses, each of which can take the values $w_i = \pm 1$, receiving inputs $\xi_i^\mu = \pm 1$, with output $\sigma^\mu = \pm 1$, and threshold $\theta = 0$. Thus, the device output is given as a function of the inputs and of the internal state as

$$\sigma^\mu = \operatorname{sign}\left(\sum_{i=1}^{N} w_i \xi_i^\mu\right)$$

Furthermore, each synapse is endowed with a discretized internal variable h_i, which only plays an active role during the learning process; for simplicity, we will consider it to be an odd-valued integer. At any given time, the sign of this quantity gives the value of the corresponding synaptic weight, $w_i = \text{sign}(h_i)$. We will start by considering the case of unbounded hidden states, and then turn to the bounded case.

SBPI is an on-line supervised learning algorithm; upon presentation of a pattern $\{\xi_i^\mu, \sigma_D^\mu\}$, where σ_D^μ is the desired output, the stability is computed as

$$\Delta^\mu = \sigma_D^\mu \left(\sum_{i=1}^{N} w_i \xi_i^\mu \right)$$

The way synaptic weights are updated depends on the value of Δ^μ:

1. If $\Delta^\mu > \theta_m$, then nothing is done.
2. If $0 < \Delta^\mu \leq \theta_m$, then only the synapses for which $w_i = \xi_i^\mu \sigma_D^\mu$, are updated, and only with probability p_s.
3. If $\Delta^\mu \leq 0$, then all synapses are updated.

Here, θ_m is a secondary threshold, expressed as an even integer, and $p_s \in [0, 1]$. The update rule applies to the hidden synaptic variables:

$$h_i \to h_i + 2\xi_i^\mu \sigma_D^\mu$$

The factor 2 is required in order to keep the value of the hidden variables odd, which in turn is useful for avoiding the ambiguous, but otherwise immaterial, $h_i = 0$ case. Note that the only actual plasticity events occur when the hidden variables change sign; also, the update in rule 2 is always in the direction of increasing the hidden variables' modulus, thus reinforcing the synaptic value by making it less likely to switch.

When the probability p_s or, equivalently, when the secondary threshold θ_m are set to 0, rule 2 is never applied and the algorithm is reduced to the CP algorithm.

In the special case, $p_s = 1$ and $\theta_m = 2$, we refer to the algorithm as to BPI.

3 Definition of the Generalization Learning Problem

The protocol which was originally used to obtain the SBPI update rules was that of classification of random patterns extracted from a given set; learning of the correct classification was achieved by repeated presentations of the patterns from the set and application of the update rules. The maximum number of input-output associations that the system could memorize in this way was shown by simulations to be proportional to the number of synapses N, the coefficient of proportionality being fairly close to the maximal theoretical value, with an order $O(\log(N)^{1.5})$ presentations per pattern required on average.

Here instead we will consider the problem of learning a rule from a teacher perceptron, identical to the student (the case of a different teacher device being considered in Sect. 9); the patterns are generated at random at each time step, each input ξ_i being extracted independently with probability $P(\xi_i = +1) = P(\xi_i = -1) = 1/2$, and the desired output is given by the teacher. Thus, the goal is to reach a perfect overlap with the teacher, an event which can be thought of as the student having learned an association rule. An optimal learning algorithm for this problem, which reaches the solution in about $1.245N$ steps in the limit of large N, can be derived by the Bayesian approach [19] (which is equivalent to the Belief

Propagation approach [5] in this case); however, this optimal algorithm does not work in an on-line fashion, as it requires to keep the memory of each pattern which was presented thus far to the device. An on-line approximation of the optimal algorithm, proposed in [19] and later re-derived from a different approach in [1] as an intermediate step towards SBPI, overcomes this problem at the expense of a lower performance, but it still requires the internal storage of continuous quantities, and complex computations to be performed at each time step.

In order to simplify the notation in the rest of this paper, we will assume that the student is always trained only on patterns whose desired output is $+1$, which can be insured in this way: at each time τ a new pattern $\{\chi_i^\tau\}_i$ is generated randomly and presented to the teacher, whose output is σ_T^τ; then, the pattern $\{\xi_i^\tau\} = \{\sigma_T^\tau \chi_i^\tau\}$ is presented to the student, with desired output $\sigma_D^\tau = +1$. Also, we can assume, without loss of generality, that all the teacher's synapses are set to $w_i^T = +1$. This implies that the student will only be presented patterns in which there are more positive than negative inputs.

In the following, we shall show that it is possible to describe the average learning dynamics and estimate the time needed for the student to reach overlap 1 with the teacher, $q = \frac{1}{N}(w \cdot w^T) = 1$.

4 Histogram Dynamics for the CP Algorithm

We will do a mean-field-like approximation to the problem: at each time step, given the histogram of the hidden variables at a time τ, $P^\tau(\{h_i\})$, we compute the average distribution (over the input patterns) at time $\tau + 1$, $P^{\tau+1}(\{h_i\})$, and iterate. The approximation here resides in the fact that, at each step, what we obtain is an average quantity (a single histogram), which we use as input in the following step, while a more complete description would involve the evolution of the whole probability distribution over all the possible resulting histograms. Therefore, we are implicitly assuming that the spread of such probability distribution around its average is negligible; our results confirm this assumption.

We will start from the simpler case of the CP algorithm (no rule 2), and temporarily drop the index τ.

Let us first compute the probability of making a classification error. This only depends on the current teacher-student overlap q. We will denote by q_+ (q_-) the fraction of student synapses which are set to $+1$ (-1), so that the overlap is $q = q_+ - q_- = 2q_+ - 1$. In the following, we have to consider separately the $+1$ and -1 synapses: we denote by ν_+ the number of positive inputs over the positive synapses, and by ν_- the number of positive inputs over the negative synapses. Because of the constraint on the patterns, there have to be more positive inputs than negative ones, i.e. $\nu_+ + \nu_- > \frac{N}{2}$. The perceptron will classify the pattern correctly if $\nu_+ + (q_- N - \nu_-) > \frac{N}{2}$, thus the probability that the student makes an error is given by

$$p_e = 2 \int d\mu(\nu_+) d\mu(\nu_-) \Theta\left(\nu_+ + \nu_- - \frac{N}{2}\right) \Theta\left(-\left(\nu_+ + (q_- N - \nu_-) - \frac{N}{2}\right)\right)$$

where $\mu(\nu_\pm)$ is the measure over $\nu_\pm$ without the constraint on the pattern (which is explicitly obtained by cutting half of the cases and renormalizing). In the large N limit, this is a normal distribution, centered on $\frac{q_\pm N}{2}$ with variance $\frac{q_\pm N}{4}$, thus we can write the above probability as

$$p_e = 2 \int Dx_+ Dx_- \, \Theta\left(\sqrt{q_+} x_+ + \sqrt{q_-} x_-\right) \Theta\left(-\sqrt{q_+} x_+ + \sqrt{q_-} x_-\right)$$

$$= 1 - \frac{2}{\pi} \arctan\left(\sqrt{\frac{q_+}{q_-}}\right)$$
$$= \frac{1}{\pi} \arccos(q) \tag{1}$$

where we used the shorthand notation $Dx = dx \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ ((1) is the standard relation between the generalization error and the teacher-student overlap in perceptrons, see e.g. [6]).

We then focus on a synapse with negative value, and compute the probability that there is an error and that the synapse receives a positive input:

$$\begin{aligned} &P(\Delta < 0 \wedge \xi_i = 1 | w_i = -1) \\ &= 2 \int Dx_+ Dx_- \left(\frac{1}{2} + \frac{x_-}{2\sqrt{q_- N}}\right) \Theta\left(\sqrt{q_+} x_+ + \sqrt{q_-} x_-\right) \Theta\left(-\sqrt{q_+} x_+ + \sqrt{q_-} x_-\right) \\ &= \frac{p_e}{2} + \frac{1}{\sqrt{2\pi N}} + \mathcal{O}\left(\frac{1}{N}\right) \end{aligned}$$

The probability that a negative-valued synapse receives a negative input, and that an error is made, is very similar:

$$P(\Delta < 0 \wedge \xi_i = -1 | w_i = -1) = \frac{p_e}{2} - \frac{1}{\sqrt{2\pi N}} + \mathcal{O}\left(\frac{1}{N}\right)$$

The probabilities for positive-valued synapses are simpler:

$$P(\Delta < 0 \wedge \xi_i = \pm 1 | w_i = +1) = \frac{p_e}{2} + \mathcal{O}\left(\frac{1}{N}\right)$$

Therefore, a positive-valued synapse (which is thus correctly set with respect to the teacher) has an equal probability of switching up or down one level, while a negative-valued one (which is thus wrongly set) has a higher probability of switching up than down. The histogram dynamics can be written in a first-order approximation as:

$$\begin{aligned} P^{\tau+1}(h) = {} & P^\tau(h)[1 - p_e^\tau] \\ & + P^\tau(h+2)\left[\frac{p_e^\tau}{2} - \frac{\Theta(-(h+2))}{\sqrt{2\pi N}}\right] \\ & + P^\tau(h-2)\left[\frac{p_e^\tau}{2} + \frac{\Theta(-(h-2))}{\sqrt{2\pi N}}\right] \end{aligned} \tag{2}$$

where, as usual, the h's are assumed do be odd. It can be easily verified that normalization is preserved by this equation.

Note that, if p_e is very small, $\frac{p_e}{2} - \frac{1}{\sqrt{2\pi N}}$ may become negative, which is meaningless; in terms of the overlap, this happens when $q_- N < \frac{\pi}{2}$, i.e. when convergence is reached up to just one or two synapses (in fact, this does never happen with the CP algorithm, which does not appear to ever converge to the solution or to even get close to convergence). This is due to the fact that the gaussian approximation we used is not valid any longer when q_- is of order N^{-1}; note however that this is not really an issue for practical purposes, as simulations show that whenever the algorithm gets into this region, convergence is eventually reached in short time.

5 Histogram Dynamics for the SBPI Algorithm

We now turn to SBPI. We have to compute the probability that the new rule 2 is applied, which happens when $0 < \Delta \leq \theta_m$ with probability p_s; thus:

$$\begin{aligned} p_b &= 2p_s \int Dx_+ Dx_- \Theta\left(\sqrt{q_+}x_+ + \sqrt{q_-}x_-\right)\Theta\left(\sqrt{q_+}x_+ - \sqrt{q_-}x_-\right) \\ &\quad \times \Theta\left(-\sqrt{q_+}x_+ + \sqrt{q_-}x_- + \frac{\theta_m}{\sqrt{N}}\right) \\ &= \frac{p_s\theta_m}{\sqrt{2\pi N}} + \mathcal{O}\left(\frac{1}{N}\right) \end{aligned} \tag{3}$$

The leading term is of order $N^{-\frac{1}{2}}$, so there's no need to distinguish between positive and negative synapses here, because the difference between the two cases is of order N^{-1}. Thus, each synapse has a probability $p_b/2$ of moving away from 0 and a probability $p_b/2$ of standing still, since only half of the synapses are involved in rule 2 each time it is applied.

We note that the result does not depend on the internal state of the device: it is a constant, acting for both positive and negative synapses. Furthermore, we see that we can reduce the number of parameters by defining

$$k = p_s\theta_m \tag{4}$$

This means that, in the generalization context and in the limit of large N, rule 2 in the SBPI algorithm can be substituted by a stochastic, generalized and unsupervised reinforcement process. We shall come back to this issue in Sect. 10.

Using (3) we can add rule 2 to (2), getting the full SBPI dynamics:

$$P^{\tau+1}(h) = P^\tau(h)J_0^\tau + P^\tau(h+2)J_-^\tau(h+2) + P^\tau(h-2)J_+^\tau(h-2) \tag{5}$$

where

$$\begin{aligned} J_0^\tau &= 1 - p_e^\tau - \frac{k/2}{\sqrt{2\pi N}} \\ J_-^\tau(h) &= \frac{p_e^\tau}{2} - \Theta(-h)\frac{1}{\sqrt{2\pi N}} + \Theta(h)\frac{k/2}{\sqrt{2\pi N}} \\ J_+^\tau(h) &= \frac{p_e^\tau}{2} + \Theta(-h)\frac{1}{\sqrt{2\pi N}} + \Theta(h)\frac{k/2}{\sqrt{2\pi N}} \end{aligned} \tag{6}$$

The agreement between this formula and the simulations is almost perfect, except when the average number of wrong synapses is very low, i.e. when q_-N is of order 1, as can be seen in Fig. 2.

6 Continuous Limit

Equations (5) and (7) can be converted to a continuous equation in the large N limit, by rescaling the variables:

$$t = \frac{\tau}{N} \tag{7}$$

$$x = \frac{h}{\sqrt{N}} \tag{8}$$

and using a probability density

$$p(x,t) = \sqrt{N} P^{Nt}\left(\sqrt{N}x\right) \tag{9}$$

Note that the $\sqrt{N}$ scaling of the hidden variables is the same which we found empirically in the classification learning problem [1].

Using these and taking the limit $N \to \infty$ we get the partial differential equation set:

$$\begin{aligned}\frac{\partial p}{\partial t}(x,t) &= 2p_e(t)\frac{\partial^2 p}{\partial x^2}(x,t) \\ &\quad - \frac{1}{\sqrt{2\pi}}\frac{\partial p}{\partial x}(x,t)[(4-k)\Theta(-x) + k\Theta(x)] \\ &\quad + \delta(x)\Theta(-x)\gamma^-(t) + \delta(x)\Theta(x)\gamma^+(t)\end{aligned} \tag{10}$$

$$p_e(t) = \frac{1}{\pi}\arccos(q(t)) \tag{11}$$

$$q(t) = 2\int_0^\infty dx p(x,t) - 1 \tag{12}$$

where $\delta(x)$ represents the Dirac delta function.

The two quantities $\gamma^-(t)$ and $\gamma^+(t)$ don't need to be written explicitly, since they can be specified by imposing two conditions on the solution, normalization and continuity:

$$\int_{-\infty}^{+\infty} p(x,t) = 1 \tag{13}$$

$$p(0^-,t) = p(0^+,t) \tag{14}$$

The reason for the continuity requirement, (14), is the following: if there would be a discontinuity in $x = 0$, the net probability flux through that point would diverge, as can be seen by direct inspection of (5) and considering how the τ and h variables scale with N. Note that, in the BPI case $k = 2$, enforcing these two constraints simply amounts at setting $\gamma^{\pm}(t) = 0$, as discussed in the next section.

As a whole, (10) is non-local, since the evolution in each point depends on what happens at $x = 0$; on the other hand, it greatly simplifies away from that point: on either side of the x axis, it reduces to a Fokker-Planck equation, with a time-dependent coefficient of diffusion, and a constant drift term. In general, the drift term is different between the left and right side of the x axis, and depends on k; this difference gives rise to an accumulation of the probability distribution on both sides of the point $x = 0$ (expressed by the two Dirac delta functions in the equation).

For negative x, (10) reads:

$$\frac{\partial p}{\partial t}(x,t) = 2p_e(t)\frac{\partial^2 p}{\partial x^2}(x,t) - \frac{(4-k)}{\sqrt{2\pi}}\frac{\partial p}{\partial x}(x,t) \tag{15}$$

If the initial distribution, at time t_0, is a gaussian centered in x_0 with variance v_0, then the solution to this equation is a gaussian whose center $\bar{x}(t)$ and variance $v(t)$ obey the

equations:

$$\bar{x}(t) = x_0 + \frac{4-k}{\sqrt{2\pi}}(t - t_0) \tag{16}$$

$$v(t) = v_0 + 4\int_{t_0}^{t} dt' p_e(t') \tag{17}$$

Let us call $g^-(x, t, t_0)$ such a solution, assuming $x_0 = 0$ and $v_0 = 0$ (i.e. assuming the initial state to be a Dirac-delta centered in 0). We can define in an analogous way a solution to the $x > 0$ branch of (10):

$$\frac{\partial p}{\partial t}(x,t) = 2p_e(t)\frac{\partial^2 p}{\partial x^2}(x,t) - \frac{k}{\sqrt{2\pi}}\frac{\partial p}{\partial x}(x,t) \tag{18}$$

As before, this equation transforms gaussians into gaussians: the corresponding solution $g^+(x, t, t_0)$ only differs from g^- in that the centre of the gaussian moves to the right with a velocity proportional to k, rather than $4 - k$.

Overall, this gives a qualitative understanding of what happens during learning: away form $x = 0$, on both sides there's a diffusion term (the same for both), which tends to 0 if the majority of the synapses gets to the right side of the x axis. The synapses are 'pushed' right by the drift with 'strength' k on the right side and $4 - k$ on the left side. Right at $x = 0$, there's a bi-directional flux between the two sides of the solution, such that the overall area is conserved and that the curve is continuous (even if the derivatives are not). Thus, it is evident that both $k \leq 0$ and $k \geq 4$ are very poor choices (and they include the CP algorithm, which corresponds to $k = 0$). If the majority of the synapses eventually reaches the right side, the diffusion stops and the drift dominates. The evolution of the histograms at different times for different values of k is shown in Fig. 1.

An analytical solution to (10) can be written in terms of the functions $g^\pm$ defined above: the flux through $x = 0$ gives rise, in the continuous limit, to the generation of Dirac deltas in the origin, which in turn behave like gaussians of 0 variance that start to spread and shift. Due to the homogeneity of the equation, this allows to write a solution as a weighted temporal convolution of evolving gaussians: first, we write the initial condition as $p(x, 0) = p_0(x)$; then, we define $p_0^-(x, t)$ as the time evolution of $p_0(x)$ under (15) and $p_0^+(x, t)$ as the time evolution of $p_0(x)$ under (18) (these can normally be computed easily, e.g. by means of Fourier transforms). This allows us to write the solution in the form:

$$p(x,t) = \Theta(-x)p^-(x,t) + \Theta(x)p^+(x,t) \tag{19}$$

where

$$p^\pm(x,t) = p_0^\pm(x,t) + \int_0^t dt' \gamma^\pm(t') g^\pm(x,t,t') \tag{20}$$

with the constraints given in (13) and (14). This solution can be verified by direct substitution in (10); it is not likely to be amenable to further analytical treatment, but it is sufficient for numerical integration, which indeed shows an almost perfect agreement with the data obtained through histogram evolution at large N, as shown in Fig. 2a.

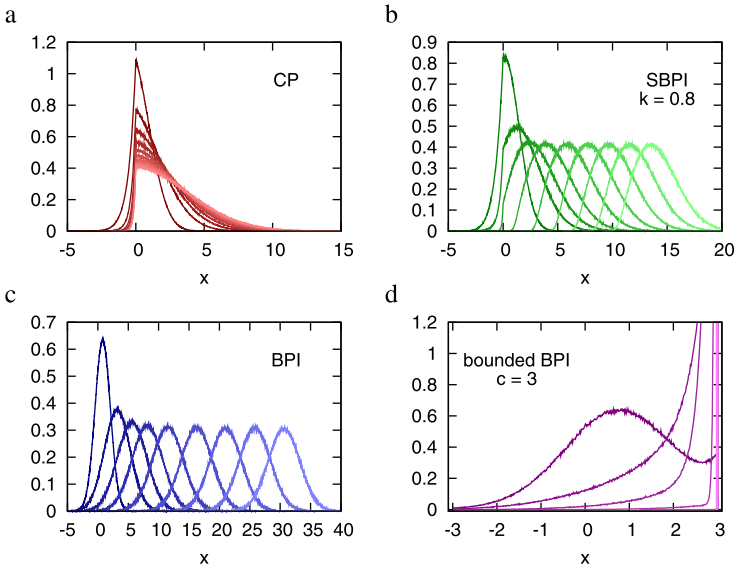

Fig. 1 Evolution of the histograms with time (*dark lines* to *light lines*, taken in time steps of $\Delta t = 3$, from $t = 1$ to $t = 25$), in simulations with four different algorithms (500 samples at $N = 32001$). In panels (**a**) and (**b**), the positive and negative sides of the curve obey different differential equations; in the CP algorithm there's no drift term on the right side, and thus the majority of the synapses stays near zero, causing a significant fraction of the synapses to be pushed back to the negative side. The distributions are gaussians for the unbounded BPI algorithm (**c**), while setting a boundary makes the histograms accumulate at the boundary (**d**). In all cases, the initial distribution was random, with all the synapses at $h = \pm 1$

7 Density Evolution for BPI

In the BPI case, i.e. when $k = 2$, the two sides of (10) are equal; thus, the terms $\gamma^{\pm}(t)$ both vanish, and (10) simplifies to:

$$\frac{\partial p}{\partial t}(x,t) = 2p_e(t)\frac{\partial^2 p}{\partial x^2}(x,t) - \sqrt{\frac{2}{\pi}}\frac{\partial p}{\partial x}(x,t) \tag{21}$$

If the initial distribution is a gaussian centered in x_0 and variance v_0, $p(x,0) = \frac{1}{\sqrt{v_0}}G(\frac{x-x_0}{\sqrt{v_0}})$, then the evolution of the distribution is described by the following system of equations:

$$p(x,t) = \frac{1}{\sqrt{v(t)}}G\left(\frac{x-\bar{x}(t)}{\sqrt{v(t)}}\right) \tag{22}$$

$$\bar{x}(t) = x_0 + \sqrt{\frac{2}{\pi}}t \tag{23}$$

$$v(t) = v_0 + 4\int_0^t dt' p_e(t') \tag{24}$$

$$p_e(t) = \frac{1}{\pi}\arccos(q(t)) \tag{25}$$

$$q(t) = \mathrm{erf}\left(\frac{\bar{x}(t)}{\sqrt{v(t)}}\right) \tag{26}$$

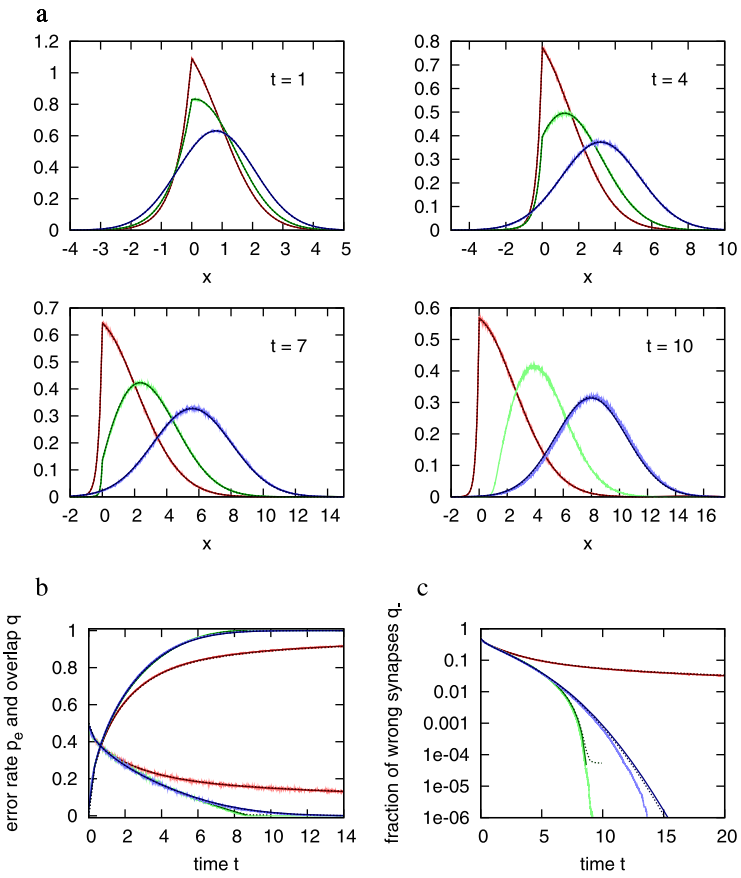

Fig. 2 (Color online) Comparison between simulations (*light solid lines*), histogram evolution (*solid lines*) and continuous probability density evolution (*dark dotted lines*), for three different algorithms (*red*: CP, *green*: SBPI with $k = 0.8$, *blue*: BPI), at different times. The curves were taken at $N = 32001$, and initialized as for Fig. 1. The agreement between the simulations and the two analytical predictions is almost perfect, except when q_- is very small. (**a**) Histograms at different times (*leftmost curves*: CP, *middle*: SBPI, *rightmost*: BPI). The analytical curves are not available for SBPI at $t = 10$ since at that point the algorithm has already converged and the approximations used are no longer valid. (**b**) Average overlap q (*top* curves, starting from 0) and error rate p_e (*bottom* curves, starting from 0.5) vs time (from worst to best performance: CP, BPI, SBPI). (**c**) Fraction of wrong synapses q_- vs time, in logarithmic scale (from topmost to bottommost: CP, BPI, SBPI). This can be used as an estimate of the convergence time with N; the BPI curve is fit asymptotically by a curve which goes like $t \propto \sqrt{\log(q_-^{-1})}$ (not shown)

Thus, the gaussian shape of the distribution is preserved, but its center and its variance evolve in time: the center moves to the right at constant speed, while the variance derivative is proportional to the error rate. Convergence is thus guaranteed, since the variance can grow at most linearly, which means that the width of the distribution can grow at most as $\sqrt{t}$, while the center's speed is constant. Thus, for sufficiently large times, the negative tail of the distribution, which determines the error rate ($p_e \sim \sqrt{1-q}$ when $q \to 1$), will be so small that the variance will almost be constant, and this in turn implies that the error rate decreases exponentially with time. If we define the convergence time T_c as the time by

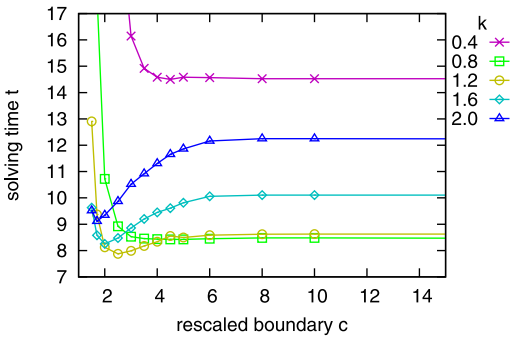

Fig. 3 Solving time with different values of the rescaled boundary c. *Each point* represents the average over 80 samples at $N = 32001$ (standard deviations are smaller than the point size). The overall optimum is found at $c = 2.5$ with $k = 1.2$. These results are consistent with those found with $N = 64001$ (not shown)

which the number of wrong synapses becomes less than 1, i.e. when $Nq \sim 1$, we find that asymptotically $T_c \sim \sqrt{\log N}$, which means that the non-rescaled convergence time is almost linear with the number of synapses.

Figure 2b shows the overlap and error rate as a function of time; the agreement of the analytical solution with the simulation data is almost perfect, except when q_- is very small, as shown in Fig. 2c.

8 Bounded Hidden Variables

We can easily introduce a limit over the number of available hidden states, by setting a maximum value $c\sqrt{N}$ for the modulus of h. Obviously, if c is too small the algorithm's performance is impaired, while if c is large enough it has no effect; in between, the behavior depends on the value of k. It turns out that setting a boundary over h can effectively improve performance for BPI ($k = 2$), but it has almost no effect for the optimal SBPI algorithm, with $k \sim 0.8$, similarly to what happens in the classification problem scenario studied in [1]: in fact, the optimum in this case was found to occur with $k = 1.2$, at $c = 2.5$. The results are summarized in Fig. 3. An example of bounded histogram evolution is in the last panel of Fig. 1.

9 Teacher with Continuous Synaptic Weights

The above results were derived in the scenario of the generalization of a learnable rule, the desired output being provided by a binary perceptron. In this section, we consider instead the case of a non learnable rule, provided by a perceptron with continuous synaptic weights extracted at random from a uniform distribution in the range $[-1, 1]$: the minimum generalization error is no longer 0 in this case, and our previous mean-field approach is not able to provide a simple analytic solution; however, (1) still holds true (in the limit of large N), and hence the best possible assignment of the student's weights is obtained by taking the sign of the teacher's weights, in which case the generalization error is equal to $1/6$.

Our simulations show (Fig. 4a) that even in this case the SBPI algorithm outperforms the CP algorithm when the parameter p_s is chosen in the appropriate range, and that there exists an optimal value for p_s such that the generalization error rapidly gets very close to the optimal value, even though the optimum is reached in exponential time (arguably due to the fact that the region around the solution is very flat in this case, because some of the teacher's weights are so small that their inference is both very difficult and not very relevant).

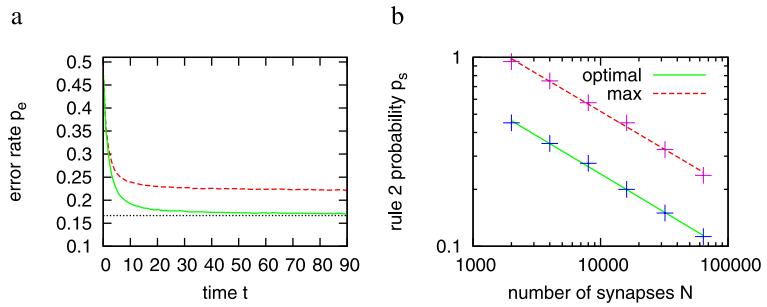

Fig. 4 (Color online) Simulation results using a teacher with continuous weights. (**a**) Average error rate p_s vs time for $N = 501$, 1000 samples (*dotted black line*: minimum possible error; *red dashed curve*: CP; *green solid curve*: SBPI with optimal $p_s = 0.8$). The learning curves scale with N following the same power law shown in the next panel. (**b**) Scaling of the p_s parameter with N, in logarithmic scale, and best fit. The fitting curves have the form aN^{-b}, the fitting parameters are $a = 9.9 \pm 0.8$, $b = 0.403 \pm 0.008$ (optimal) and $a = 20.5 \pm 2.4$, $b = 0.400 \pm 0.013$ (maximum)

One important difference between this case and the previous one is that both the optimal and the maximum value of the parameter p_s (the maximum value is the one above which the performance becomes equal or worse than that of CP) are not fixed with varying N: rather, they both scale following the same power law (Fig. 4b).

10 A Simplified Algorithm: CP + R

We have shown in Sect. 5 that, in the limit of a large number of synapses N and in the context of the generalization learning of a learnable rule, the effect of the additional rule which distinguishes the SBPI algorithm from the CP algorithm, and which is responsible for the superior performance of the former with respect to the latter, is on average equivalent to applying an unspecific, constant and low-rate meta-plastic reinforcement to all the synapses (see (3)). This reinforcement process is only effective if it is not too strong, because otherwise it would overcome the effect of the learning by keeping all of the synapses away from the plastic transition boundary (i.e. away from $x = 0$, in the notation of Sect. 6).

This suggests that the SBPI algorithm can be further simplified, leading to a "clipped perceptron plus reinforcement" algorithm (CP + R), i.e. the CP algorithm with the additional prescription that, at each time step τ, each synaptic weight undergoes a meta-plastic transition $h_i^\tau \to h_i^\tau + 2\text{sign}(h_i^\tau)$ with probability p_r, where $0 < p_r < \sqrt{\frac{2}{\pi N}}$ (the time step index τ does not increment in the reinforcement process, because it is superimposed to the standard learning rules and acts in parallel with them). Any value of p_r greater then 0 makes a qualitative difference with respect to CP.

The CP + R algorithm is only equivalent to the SBPI algorithm in the generalization of a learnable rule scenario. Indeed, in the case of a the non-learnable rule of Sect. 9, the relationship of (3) does not hold any more; however, the CP + R algorithm still proves as effective as SBPI when the parameter p_r is properly set (not shown).

In the classification problem, on the other hand, the performance of CP + R is worse in terms of capacity by a factor of the order of 2 with respect to SBPI. However, our preliminary results show that the difference in such scenario between the two algorithms shows up only in the latest phases of the learning (when the temporal correlations in the inputs make a

difference), and that simply reducing the rate of application of the reinforcement process p_r during the learning along with the error rate is sufficient to recover the SBPI performance even in that case. This will be the subject of a future work.

From the architectural point of view, such CP + R algorithm is even simpler than the SBPI algorithm (which was already derived as a crude simplification of the Belief Propagation algorithm); thus, it may be an even better candidate for modelling supervised learning in biological networks, which have very strict requirements about robustness, simplicity and effectiveness. Its only serious drawback with respect to SBPI is that the random reinforcement must be applied sparingly, since the probability is of order $O(1/\sqrt{N})$, which would require some fine-tuning mechanism of the cells behaviour; SBPI, on the other hand, requires detection of near-threshold events in order to trigger the reinforcement rule, which may also be problematic. Furthermore, even if the learning rate under CP + R is sub-optimal with respect to the generalization protocol problem, its extreme simplicity and robustness might be attractive for hardware implementations of binary perceptron units with very large number of synapses as well, because it is adaptable to both the classification and the generalization scenarios, and, even in the latter (algorithmically easier) case, it greatly reduces the overhead associated with the complex computations required by the faster algorithms, while still having a very good scaling behaviour with N, as the steps required grow at most as $\mathcal{O}(N\sqrt{\log N})$.

11 Summary

In this paper, we have studied analytically and through numerical simulations the SBPI algorithm dynamics in the supervised generalization learning scenario and in the limit of a large number of synapses N.

The original goal, which was that of claryfing the role of the novel learning rule introduced by this algorithm, was approached by studying the average dynamics of the internal synaptic state and separating the contributions due to the different learning rules, which allowed us to derive a partial differential equation describing the learning process in terms of a diffusion process. The solution of such equation in a (non-optimal) special case provided us with an estimate for the learning time, which turned out to scale as $N\sqrt{\log N}$. The analytical predictions were found to be in excellent agreement with the numerical simulations.

We have also obtained some results from simulations under circumstances in which the previous analytical approach failed, and found that the SBPI algorithm can be further optimized by setting properly a hard boundary to the number of internal synaptic states (scaling as $\sqrt{N}$), which confirms our previous results in the context of classification learning, and that its enhanced effectiveness with respect to CP is not limited to learnable rules.

The analytical results, together with their interpretation in terms of the synaptic states' dynamics, have also suggested the introduction of a novel, simplified algorithm, called CP + R, which proved in our preliminary results to be as much effective as SBPI under all the circumstances in which we have tested it (with some minor adjustments), making it a good candidate for biological and electronic implementations, and which will be the subject of a future work.

Acknowledgements I thank A. Braunstein, N. Brunel and R. Zecchina for helpful discussions and comments on the manuscript.

References

1. Baldassi, C., Braunstein, A., Brunel, N., Zecchina, R.: Efficient supervised learning in networks with binary synapses. Proc. Natl. Acad. Sci. USA **104**, 2079–2084 (2007)
2. Bhalla, U.S., Iyengar, R.: Emergent properties of networks of biological signaling pathways. Science **283**, 381–387 (1999)
3. Bialek, W.: Stability and noise in biochemical switches. Adv. Neural Inf. Process. Syst. **13**, 103–109 (2000)
4. Blum, A.L., Rivest, R.L.: Training a 3-node network is np-complete. Neural Netw. **5**, 117–127 (1992)
5. Braunstein, A., Zecchina, R.: Learning by message-passing in networks of discrete synapses. Phys. Rev. Lett. **96**, 030,201 (2006)
6. Engel, A., van den Broeck, C.: Statistical Mechanics of Learning. Cambridge University Press, Cambridge (2001)
7. Fusi, S., Drew, P.J., Abbott, L.F.: Cascade models of synaptically stored memories. Neuron **45**(4), 599–611 (2005)
8. Golea, M., Marchand, M.: On learning perceptrons with binary weights. Neural Comput. **78**, 333–342 (1993)
9. Gutfreund, H., Stein, Y.: Capacity of neural networks with discrete synaptic couplings. J. Phys. A Math. Gen. **23**, 2613–2630 (1990)
10. Hayer, A., Bhalla, U.S.: Molecular switches at the synapse emerge from receptor and kinase traffic. PLoS Comput. Biol. **1**(2), e20 (2005)
11. Kabashima, Y., Uda, S.: A BP-based algorithm for performing Bayesian inference in large perceptron-type networks. In: Algorithmic Learning Theory. Lecture Notes in Computer Science, vol. 3244, pp. 479–493. Springer, Berlin (2004)
12. Kinouchi, O., Caticha, N.: Optimal generalization in perceptrons. J. Phys. A **25**, 6243 (1992)
13. Krauth, W., Mézard, M.: Storage capacity of memory networks with binary couplings. J. Phys. **50**, 3057 (1989)
14. Miller, P., Zhabotinsky, A.M., Lisman, J.E., Wang, X.J.: The stability of a stochastic camkii switch: dependence on the number of enzyme molecules and protein turnover. PLoS Biology **3**(4), e107 (2005)
15. O'Connor, D.H., Wittenberg, G.M., Wang, S.S.H.: Graded bidirectional synaptic plasticity is composed of switch-like unitary events. Proc. Natl. Acad. Sci. USA **102**, 9679–9684 (2005)
16. Petersen, C.C., Malenka, R.C., Nicoll, R.A., Hopfield, J.J.: All-or-none potentiation at CA3-CA1 synapses. Proc. Natl. Acad. Sci. USA **95**, 4732–4737 (1998)
17. Rosenblatt, F.: Principles of Neurodynamics. Spartan Books, New York (1962)
18. Rosen-Zvi, M.: On-line learning in the ising perceptron. J. Phys. A **33**, 7277–7287 (2000)
19. Solla, S.A., Winther, O.: Optimal perceptron learning: an online Bayesian approach. In: On-Line Learning in Neural Networks. Cambridge University Press, Cambridge (1998)
20. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations. In: Exploring Artificial Intelligence in the New Millennium, pp. 236–239. Morgan Kaufman, San Mateo (2003). Chap. 8