Univeristà degli Studi di Torino

PhD course in Complexity in post-genomic Biology

XX cycle

A novel algorithm for supervised learning in neuronal models with binary synapses

Carlo Baldassi

Advisors:
prof. Nicolas Brunel
prof. Riccardo Zecchina

Contents

Introduction 5

Chapter 1. The binary perceptron learning problem 7
1. Synaptic plasticity and long-term learning 7
1.1. Neuronal communication 7
1.2. Plasticity and learning 8
1.3. Learning experiments 9
1.4. Difficulties arising in the development of a theory of learning 9
1.5. Continuous vs discrete synapses 10
2. Perceptron models 11
2.1. Simplified neuronal models 11
2.2. The LIF model 11
2.3. The perceptron model 12
2.4. Different kinds of learning 13
2.5. Different kinds of network structures 13
2.6. Perceptrons storage capacity 14
2.7. Continuous vs binary synapses in supervised perceptron models 17
2.8. Continuous vs binary synapses in unsupervised models 19

Chapter 2. The SBPI algorithm 21
1. Cavity algorithms 21
1.1. Cavity methods for statistical physics 21
1.2. The Belief Propagation algorithm 22
1.3. BP on 0/1 perceptron 25
1.4. Reverting BP to an on-line algorithm 26
1.5. Discretization 26
1.6. Algorithms definitions 27
1.7. Performance comparison 29
1.8. Bounded hidden variables 31
1.9. Distribution of hidden variables 31
1.10. Optimal value of the number of hidden states K 31
1.11. A more biologically plausible algorithm 32
1.12. Heterogeneous synapses and sparse coding 34
1.13. Binary vs K state synapses 35
1.14. Robustness against noise 37
2. Generalization protocol 37
2.1. Histogram dynamics for the CP algorithm 39
2.2. Histogram dynamics for SBPI 40

2.3. Continuous limit 41
2.4. Density evolution for BPI 44
3. Discussion 44

Bibliography 47

Introduction

The ability to dynamically adapt to the external stimuli and to retain the memory of past events are among the brain's most striking and crucial features, and are one of the most active fields of past and current research, both on the theoretical and on the experimantal side. Both processes of learning from experience and of memory formation are widely believed to occur through mechanisms of synaptic plasticity, i.e. of modulations of the signal transmission between neurons. However, due to the huge degree of complexity of the processes involved, describing their properties is a major challenge for both theroeticians and experimentalists.

In fact, an established framework about the synapses' dynamics is still lacking, despite the huge amount of experimental data collected, and many aspects of brain computations are yet unclear, including the signal encoding and whether the synaptic efficacies have a discrete or continuous nature.

On the other hand, at least some of the modifications induced through synaptic plasticity have to be permanent, while the biological environment in which these processes happen is subject to a very high level of noise; thus, the existence of a discrete set of stable states in a synapse would significantly improve its robustness. Multistability could be induced by positive feedback loops in protein interaction networks of the post-synaptic density, the small and highly specialized structure which is found in the dendritic spines [**18, 35, 6**]. This is in agreement with some recent experiments, which have suggested single synapses could be similar to noisy binary switches [**26, 23**], meaning that each synapse would have only two states, one with high conductance and one with low conductance.

From the theoretical point of view, however, there is an important difference between models which use continuous synaptic efficacies and those which use binary synapses, since it is in general much easier to develop efficient and plausible learning protocols in the continuous case, both in the unsupervised learning scenario (in which synaptic modifications are only induced by the pre and post-synaptic activities) and in the supervised scenario (in which an external 'teaching' or 'error' signal is present).

In fact, it has been shown [**32, 4, 5, 11**] that the performance of binary synapses systems (in terms of information stored per synapse) in the unsupervised scenario is very poor, unless two conditions are met: (1) activity in the network is sparse (very low fraction of neurons active at a given time); and (2) transitions are stochastic, with in average a balance between up and down transitions. This poor performance has motivated further studies [**12**] in which hidden states are added to the synapse in order to provide it with a multiplicity of time scales, allowing for both fast learning and slow forgetting. The hidden synaptic states are not directly involved in the unit's electrical properties, but rather they influence its plasticity properties; modifications of the hidden states are thus called "meta-plastic". As for the visible synaptic states, the hidden states could be represented by stable points of a protein interaction network.

In the supervised learning scenario, for the prototypical network in which this type of learning has been studied, the one-layer perceptron which has to perform a set of input-output associations, no efficient algorithms are known to exist when synapses have a finite number of states, in the case the number of input-output associations to be learned scales with the number of synapses. In fact, while learning in systems with analog synapse can always be achieved with simple algorithms if a solution exists, learning in systems with binary synapses is known to be a NP-complete task [**2, 3**], meaning that in the general case it belongs to the hardest computational class of combinatorial optimization problems. Moreover, even the easier case in which the patterns which have to be classified are supposed to be generated at random, and

we consider the typical performance over an instantiation of the problem rather than the worst possible case, no learning algorithms are known to achieve the goal in a time which goes as a polynomial of the number of synapses, including models which make use of meta-plasticity.

Recently, 'message passing' algorithms have been devised that solve efficiently non-trivial random instances of NP-complete optimization problems, like e.g. K-satisfiability or graph coloring [**19, 20, 1, 7**]. One such algorithm, Belief Propagation (BP), has been applied to the binary perceptron problem and has been shown to be able to find efficiently synaptic weight vectors that solve the classification problem for a number of patterns close to the maximal capacity (above 0.7 bits per synapse)[**8**]. However, this algorithm has a number of biologically unrealistic features (e.g. memory stored in several analog variables).

Here, we present a novel algorithm, called SBPI, that is inspired from the BP algorithm but is modified in order to make it simpler and biologically realistic, while keeping very good performances, both in terms of learning time and of information storage capacity. This algorithm requires meta-plasiticity, and introduces also a new learning rule, directly inherited from the BP algorithm, which also proves to be able to boost the performance of other algorithms. We provide evidence about the qualitatively superior performance of SBPI with respect to other well known algorithms both using extensive computer simulations, in the case of learning a set of pattern classifications, and by an analytical mean field study, in the case of learning a rule from a teacher device.

CHAPTER 1

The binary perceptron learning problem

1. Synaptic plasticity and long-term learning

1.1. Neuronal communication. The communication between neurons in a neuronal tissue can happen through many channels, depending on the neuronal species involved and the specific area under consideration. Each communication vector will in general have peculiar spatial and time scales. As a general picture, it is widely believed that information is transmitted in the brain through the propagation of electrical signals, modulated by the presence of particular chemical species, generally referred to as "neurotransmitters"; there is also the possibility that the individual electric behavior is modified by the overall magnetic fields produced by the surrounding areas (e.g. in the cortex, where many electrical signals flow along parallel directions and global field oscillations are observed), but the role of such modulation, if any, is not yet clear.

There are essentially two ways of electrical signal propagation between neurons, a direct one and a chemical one. The direct one happens through structures known as "gap junctions", which are structures which directly connect the neurons' membranes and let the electrical field propagate from one neuron's membrane to another, just as a resistor would do in an electrical circuit. These structures are very simple: as for their internal state, they can only be open or closed, but there is the possibility of further global modulation by neuromodulators; furthermore, the signal transmitted through the gap junctions is essentially analog in nature, since the electric potential flows passively from one neuron's membrane to its neighbor's. One well known example of a brain area in which the gap junctions are of great importance is the eye's retina[**31**], which is a very specialized structure; their role in other brain areas is generally more obscure.

The second and most prominent kind of signal propagation happens instead through the synapses, which are specialized structures capable of transforming an electrical incoming signal into a chemical one and back to an electrical outgoing one, allowing much space for modulation in the intermediate steps. In each neuron, synaptic outputs occurs typically at the end of axons, where the actively transported electrical signals, the "action potentials" or "spikes", elicited at the level of the soma, need to get transmitted to another neuron; these signals are intrinsically binary in nature, at least as long as the axons are longer than about 1mm: at a first-level description, they are elicited when the neuron's membrane potential reaches a given threshold, which triggers an active mechanism (a short positive-feedback period followed by a refractory period) resulting in a sharp peak of the membrane potential (the "spike"), which is then propagated along the axon by the action of the Na-K channels which are present in the axon's membrane, while the sub-threshold oscillations of the potential are too weak to propagate with this mechanism, and die out rapidly.

When an action potential reaches a synapse, it triggers the opening, in the inter-synaptic medium, of some neurotransmitter vesicles (as this is a stochastic process, the exact number varies at each repetition); the neurotransmitter reaches some post-synaptic receptors on the

post-synaptic side, which in turn determine the opening of some ionic channels on the post-synaptic membrane which produce an electrical signal (either depolarizing or hyper-polarizing), which then propagates (mostly passively) through the post-synaptic neuron.

Once again, the effect of this process can be represented as that of a (directional) resistor in an electrical circuit, so that we can characterize each synapse by the value of its conductance, which is also called "synaptic weight" or "synaptic efficacy". Experimentally, the value of the synaptic conductance can be determined by the peak value of the post-synaptic potential (PSP) elicited by the arrival of a single spike (the incoming spikes being considered all equal).

In contrast to gap junctions, which act passively, synaptic connections can either exert an excitatory or an inhibitory action, depending on the neurotransmitter (which in turn depends on the pre-synaptic neuronal species) and the post-synaptic receptors. Mathematically, inhibitory synapses are often modeled as negative-weighted synapses. The sign of synapse cannot change over time, except in some cases during brain development.

1.2. Plasticity and learning. The whole synaptic signal transmission process, though being slower than the direct transmission by the gap junctions (on the milliseconds time scale), subject to random fluctuations and (nearly) unable to transmit the information contained in the sub-threshold electrical fluctuations of the soma, has the enormous advantage of being highly modulable, both by chemical action (e.g. of neuromodulators) and by alteration of the internal synaptic state, or even by synapse creation-removal processes.

This is the main reason for which synapses are widely believed to be the places where learning takes place, as experience can shape their properties and thus alter the signal transmission in a sensible way, a property which goes under the name of "synaptic plasticity". The signal transduction modulation can, and indeed does, happen both on the pre-synaptic side (e.g. by changing the average number of vesicles released upon action potential arrival, or the amount of neurotransmitter contained in each of them[**27, 30**]) and on the post-synaptic side (e.g. by changing the number of neurotransmitter receptors inserted in the membrane, or their state); each neuronal species can in principle be subject to different combinations of all these forms of plasticity, all of which can occur on different time scales and have peculiar results on the signal transmission and on the plasticity process itself. Nevertheless, the post-synaptic side looks more promising for the kind of learning we are going to address here, mainly because of the existence of the "dendritic spines", which are very specialized structures present on many post-synaptic neuron's dendritic trees; also, the post-synaptic part of the synapse can easily access the relevant information about both pre- and post-synaptic neurons potentials, while the pre-synaptic part requires the mediation of retrograde messengers.

The main distinction among different types of plasticity is the timescale by which the synaptic modifications last; plasticity is thereby usually divided into short-term and long-term, the former being typically associated with transitory synaptic modulation, and the latter with long-lasting modification of the synapse internal state. Of course, there is no precise boundary between these two forms, and their meaning can change depending on the context; in the present work, nonetheless, we will only address long-term plasticity, by which we mean that form of plasticity which induces a permanent modification in the synaptic conductance, on the time scale of years, and such that only another long-term plasticity event can alter it again.

If the plastic modification is in the direction of enhancing the synaptic conductance, the corresponding event goes under the name of "long-term potentiation", or LTP, while the opposite event is called "long-term depression" or LTD.

1.3. Learning experiments. The study of the plasticity properties of the synapses is believed to be of central importance in the study of the brain and neural tissues in general, and the number of theoretical and experimental studies which tried to address this issue is virtually countless. On the experimental side, nevertheless, due to the extremely difficult conditions required in order to have full control of the plasticity events, the synaptic machinery is still rather obscure. Also, the difficulty of recording the transmission of the signal through a single synapse is such that most experiments deal with statistical properties of some synaptic ensemble.

Plasticity and learning experiments are typically very different depending whether they are performed in vitro or in vivo: the former ones normally consist in the study of the effect of the application of some electrical stimulation on brain slices (several stimulation protocols are known which can elicit LTP/LTD in such situations), the latter ones involve instead behaving animals, learning to perform some task (or even just being subject to some stimulus) across a period of some days or months. As is normally the case, in vitro experiments have the obvious advantage of a better control over the conditions of the experiments, but the learning protocols used could in principle have nothing to do with the in vivo situation; furthermore, the interpretation of the experiments relies on a theoretical framework, and the groundings are not yet stable on this side as well.

1.4. Difficulties arising in the development of a theory of learning. On the theoretical side, the difficulty in understanding the learning process arises from the overwhelming complexity of the networks and of their constituents, which makes it impractical to perform detailed simulations and impossible to obtain an accurate analytical description; each theoretical model has to deal with some simplifications. On the simulations side, for example, it is possible to simulate the electric properties of single neurons with a very high level of accuracy, but it is still impossible to include into the model all the aspects which contribute to the synaptic dynamics (e.g. gene expression); furthermore, experimental data is not complete, and details often vary between one neuronal species and another. Even worse, simulating or analyzing a network with tens of thousands of neurons becomes impossible without further, crude simplifications, and again different neuronal species or neurons belonging to different brain areas may be subject to different working regimes, completely changing their properties concerning learning. Even the same neurons in the same brain area can work under different regimes under different external or internal conditions (e.g. in different times of the day), switching between two or more different behaviors.

In fact, neural coding, i.e. the way information is stored in the membrane's electric potentials travelling along the axons, is still a matter of intense debate in the field, as it is not yet clear the degree up to which the exact timing of the spikes is relevant: one popular assumption is that the information transmitted is rate-coded, i.e. that its nature is stochastic, and that only the average spike rate is meaningful for the sake of neural computing or neural decoding; the opposite view is that each single spike is relevant, and that the exact timing encodes valuable information up to the millisecond or even sub-millisecond scale. Another kind of neural code which has been proposed and for which evidence has been collected in many different situations consists in expressing some information by the timing of the spikes relative to the phase of some undergoing rhythm (i.e. global oscillation) in the area: for example, the incoming spikes received near the top of the field oscillation could carry a different amount or kind of information with respect to the ones received near the trough [**22**].

Indeed, the actual neural coding has a big influence over the learning model that one decides to consider, and the fact that none of such models can be regarded as paradigmatic increases the uncertainty about the evaluation of the single learning models.

1.5. Continuous vs discrete synapses. Assuming that the brain learns by altering the connections between its units, and that each synapse can be characterized by a single efficacy value, it becomes very important to determine whether those values are continuous or discrete variables. Indeed, as mentioned before, many neuronal species exist, and it may well be that the answer to this question depend on the brain area under consideration, but there's a general argument suggesting that the discrete model is more appropriate for long-term memory: if the information stored in a synapse has to last for a time of the order of tens of years, the problem of reliability of the storing devices becomes of great importance, and continuous-valued quantities are more prone to this problem than discrete-valued ones. In fact, most synaptic connections at the dendritic level are located onto structures called "dendritic spines", which are so small that, for each chemical species present, there's a number of the order of 10 or 100 molecules[**24**]; this implies that for any chemical reaction there is a much higher degree of stochasticity than in bigger structures as the soma, and that storing a value in the form of an average concentration value, for example, becomes impossible over long time scales. On the other hand, it has been shown [**35, 6, 24**] that it is possible to devise simple chemical networks which exhibit a small number of stable states and which turn out to be stable for a time of the order of 100 years even in presence of the extremely high chemical noise present in the synaptic boutons. The simplest and most stable situation is that in which there are only two different states, which would induce to look at the synapses as (noisy) binary switches.

On the experimental side, determining the discrete or continuous nature of a single synapse can be an awkward task, due to the great difficulty of performing simultaneous measurements in two cells undergoing plasticity, and it is currently possible only for in vitro experiments. These give indeed precious information, but have some big disadvantages: first, the network environment might be different from that of the living brain (for example, in brain slices many long-range connections are cut, and the activity state of the network could be different from that of the intact structure, and this could in turn be relevant for the learning mechanism). Furthermore, plasticity is induced during in vitro experiments by applying standard stimulation protocols, which are known to elicit changes in the synaptic strengths, but which are probably different form the ones that occur in a living animal, and it is not clear if the molecular underlying mechanisms are actually the same.

Recently, in a remarkable in vitro experiment [**23**], O'Connor and colleagues managed to detect single synaptic plasticity events in hippocampal pyramidal cells, and their evidence supports the idea that those synapses, though being very noisy, exhibit all-or-none potentiation, i.e. that they are binary, and that the state change occurs on very fast timescales, of the order of less than 10ms. During this experiment, it was shown that each synapse could only be potentiated (or depressed) at most once in a row, and that the net effect of depressing after potentiation, or the inverse, was non-detectable, supporting the idea that no intermediate states are present.

The discrete nature of the synaptic efficacies would thus solve the issue of long-term memory reliability, but at the same time it would impinge the unit's learning capability if standard learning algorithms were used. The purpose of the present work is to propose a novel learning scheme which would allow to overcome such difficulty, and we will come back on this problem after the introduction of the perceptron neuronal models.

2. Perceptron models

2.1. Simplified neuronal models. As mentioned above, neuronal models exist which can achieve a great degree of accuracy in simulating the cell membrane's depolarization (not concerning synaptic plasticity nor neuronal development). In such models the cell body is represented as a collection of compartments, the geometry of which can be taken from a real neuron by three dimensional scanning, over which both the passive and active electrical properties of the membrane are modeled, the active ones being due to the ionic channels inserted in the membrane itself. Virtually all of these models make use of the Hodgkin-Huxley equations, with excellent results; other variants can use Markov-chain models of the ionic channels to achieve an even better accuracy.

Unfortunately, such accurate models suffer from the drawbacks exposed in section 1.4 about plasticity and large networks; in order to overcome some of these difficulties and get more theoretical insight in the learning process (e.g. in view of electronic implementations), theorists normally deal with very simplified models, such that some analytical treatment is possible and that large-scale simulations become feasible. The most popular among such models are the "leaky integrate-and-fire" (LIF) and its generalizations, but historically these were preceded by the (even simpler) "perceptrons".

All of these models represent the neuron as a "point-like" unit, meaning that the precise geometry of the cell is not represented; the membrane depolarization is simply obtained by summing up the different contributions from all the neuronal inputs.

2.2. The LIF model. In the basic LIF model, the set of partial differential equations of the Hodgkin-Huxley model is drastically reduced to a single differential equation, describing the sub-threshold passive working regime, with an external instantaneous spiking mechanism added on top of it, in order to mimic the action potential process.

More specifically, the membrane voltage $V(t)$ obeys the following equation:

$$CV'(t) = -g_L V(t) + I(t)$$

where C is the capacity constant of the membrane, g_L its leak conductance, and $I(t)$ is the incoming current on the neuron. The model is completed by the prescription that when the voltage $V(t)$ reaches the spiking threshold θ_s a spike is emitted and the potential drops instantaneously to the reset potential θ_r. Since the rise and fall of the voltage in actual neurons during the action potential emission is typically much faster than the sub-threshold regime timescale, the spiking in the LIF model is represented mathematically as a Dirac delta in the voltage trace.

Additionally, a refractory period τ_r during which the voltage is kept fixed at θ_r can be added after the spike emission process, to account for the fact that the output firing rate of real neurons is bounded by the time constants of the molecular mechanisms and the need of regeneration after spike emission.

The LIF model is ideal for the study of large and complex networks, in which the incoming current on each unit can be divided into an "external" part, coming from outside the network, and an "internal" part, due to other units of the network itself. Thanks to the linearity of the device, this second contribution can be obtained as the sum, over all of the inputs, of the individual synaptic responses elicited by each incoming spike, the so-called "post synaptic potential" (PSP). Since each individual synapse i has a corresponding synaptic efficacy w_i associated with it, the

overall depolarization due to the internal input current can be written as:

$$V_I(t) = \sum_{i=1}^{N} \sum_j w_i K\left(t - t_i^{(j)}\right)$$

where N is the number of synapses, $t_i^{(j)}$ is the arrival time of the j-th spike on the i-th synapse and $K(t)$ is the PSP kernel, which is an alpha function (with the additional constraint that it is causal, i.e. that $K(t) = 0$ if $t \leq 0$). Of course, this equation has to be completed with the spiking mechanism described above.

In the above equation, the synaptic efficacies w_i are not explicitly dependent on time; it is very simple, however, to modify the model in order to account for the possibility that these quantities change their value, and this allows for simulation of any kind of learning process.

Indeed, apart from the issues arising from the cell geometry (which affect the linearity of the input summation), the dynamical range of the LIF model is much reduced with respect to that of an actual neuron, and the behavior is not as rich; some examples may include the fact that in real cells the spiking threshold is not really fixed, but varies dynamically, and the spiking itself is not instantaneous; that the spiking rate, even in presence of a constant input, is not stable during time but decreases as a result of the so-called "adaptation mechanism"; that real neurons can work in different regimes (e.g. spiking or bursting) while LIF neurons cannot. Some of these issues and others can be addressed by applying specific modifications to the base LIF model, depending on the situation under study and the actual need to reproduce the full dynamical range of the neurons.

2.3. The perceptron model. The simplest neuronal model, explicitly invented with the purpose of gaining insight into the amazing learning properties of the neural tissue, is the "perceptron", first proposed by Rosenblatt [**28**]; many different variants have been proposed since then, but they all share some properties, namely that the time in such models is discretized, and that the instantaneous depolarization is computed as the scalar product of the vector of the inputs with that of the synaptic weights; the output of the unit is then computed upon the depolarization by means of some simple function. In symbols:

$$\sigma^t = \chi\left(\sum_{i=1}^{N} w_i^t \xi_i^t\right) \tag{1}$$

where N is the number of synapses, ξ_i^t is the input incoming on the i-th synapse at time t, w_i^t is the i-th synaptic efficacy at time t, χ is the output function and σ^t is the unit output at time t.

The different variants of the model can be divided into two main categories, one in which both inputs and outputs are continuous variables, and another one in which they are binary. Making the parallel with the real neurons, the quantities that the input and output variables represent would be the firing rates in the continuous case, and either the spiking condition or an up/down state in the binary case. In the present work we will only deal with the latter scenario.

The most natural choice, from the biological point of view, is to choose the inputs and outputs to take the values 0 or 1. For example, after time discretization, it is possible to assign the value $\xi_i^t = 1$ to those synapses i for which at time t there has been at least one incoming spike, and $\xi_i^t = 0$ to those for which there has been none; in the same way, if the output is $\sigma^t = 1$, the unit fires at time t, and if $\sigma^t = 0$ it doesn't. The explicit form of the output function

is in this case:

$$\sigma^t = \Theta\left(\sum_{i=1}^{N} w_i^t \xi_i^t - \theta_s\right) \tag{2}$$

where θ_s represents a spiking threshold and Θ is the Heaviside step function, $\Theta(x) = 1$ if $x \geq 0$ and 0 otherwise.

In order to further simplify the device, mainly in view of analytical calculations, and for historical reasons as well, the most popular binary perceptron models use instead the values $+1$ and -1 for both the inputs and the outputs, and the following input-output relationship:

$$\sigma^t = \text{sign}\left(\sum_{i=1}^{N} w_i^t \xi_i^t\right) \tag{3}$$

In order to avoid the possibility of having a zero overall depolarization, which would require an inessential complication in the model definition, we also assume the number of synapses N to be odd when considering this ± 1 model.

The ± 1 model is not strictly equivalent to the 0/1 model, but its general properties are very similar. In this work, we will mostly deal with the ± 1 model, but resort to the 0/1 model in some special cases.

The main difference between the perceptron model and the LIF model described above consists in the way in which the incoming information is temporally integrated; although the latter is more accurate as a neuronal model, the perceptron is, in its simplicity, still complex enough to achieve remarkable results and to raise nontrivial challenges, offering a framework for developing simple and efficient learning protocols, and to allow for analytical calculations about their intrinsic properties; another issue, considered in the conclusions, is the fact that perceptron models are more convenient than integrate and fire (or more complicated) models for realizing electronic implementations.

2.4. Different kinds of learning. Given the model definition, it is possible to give a more precise meaning to what is meant by "learning"; still, different options are possible, depending on the network structure and on its purpose.

A first, fundamental distinction can be made between "supervised" and "unsupervised" learning protocols: the former ones are characterized by the presence of an external error signal, which is instead absent in the latter ones. Supervised learning models are intended at simulating the kind of learning which is achieved by trial and error, while unsupervised ones try to retain or exploit the information they read without any feedback from the exterior, as could be the case for transient memories for example, or for a pre-processing step in the elaboration of sensory information.

In turn, supervised learning scenarios can use global error signals, external to the network and delivered to all or many of its units, in which case is more appropriate to speak of "reinforcement learning", or they can use local signals, which act on the single units individually. In the present work, we deal with this last scenario.

2.5. Different kinds of network structures. The simplest network structure is of course given by a single unit, which could be used to extract some information from the inputs. Letting M different units operate in parallel on the same inputs would then allow to extract any number of features from an input stream, and the whole network would act as a mapping from the space of N bit numbers to the space of M bits numbers.

In this case, a supervised protocol would be used in order for the network to learn how to do a specific mapping, while an unsupervised protocol could serve the purpose of elaborating the input stream, for example by trying to compress it without loosing information while adapting to the inputs distribution. In both cases, the degree up to which the task can be accomplished depends on the units' structure, the learning algorithm and the task itself.

Another widely studied arrangement is the "fully-connected" network, in which $N+1$ units, each having N synapses, receive their inputs from all the others, and in turn deliver their output to all the others. Each unit is also supposed to be subject do an external drive which may force it in a specific state, and to be readable from outside the network. Such a network has the property that, after initialization, it can undergo its own internal dynamics and reach, after a transient phase, a stable state, called an "attractor" of the network, which may be then read out. The possibility to shape the attractors of the network could then be viewed as a learning process: each attractor is a memory which can be recalled if the network's state gets sufficiently close to it. The region of the network's phase space which has a given attractor as the endpoint of the dynamics is called the "basin of attraction" of that memory; the bigger this region is, the better will perform the network in retrieving partial or corrupted information and recognizing a previously stored memory, but there is a trade-off between the number and the size of the attractors in any given network.

In order to distinguish between memorization and retrieval, two distinct operation modes can be used: during the learning session, the memories are presented to the network through the external drive and the recurrent connections are weakened and plastic, while during the retrieval session the external input may be only partial and the recurrent connections are strong and non plastic.

In this case the difference between supervised and unsupervised learning scenarios is that in the former the attractors to be stored are known from the beginning, and repeatedly presented to the network, while in the latter there may be a continuous stream of memories, the goal being to keep a fading trace, storing more strongly the most recent ones and gradually forgetting the old ones.

A very simple variant of this arrangement allows the memorization of dynamical attractors, in the form of ordered sequences of patterns: it is sufficient to introduce a delay between the outputs emission and their receival as other units' inputs: in this way, the network can be instructed about the next step to take in response to a given input, and if this process is iterated a whole succession of network states can be learned and subsequently triggered by initialization. Of course, there's a trade-off between the number of sequences which is possible to store in this way and their length.

2.6. Perceptrons storage capacity. The paradigmatic supervised learning scenario consists of a single perceptron, for which the inputs are all taken from a subset of the possible inputs, and whose goal is to achieve a correct classification of the inputs into two predetermined categories. Due to the extremely simple structure of the perceptrons, achieving a perfect classification might be impossible, and either more units or a more complex structures might be needed. More precisely, the requirement for perfect learning to be possible is that the inputs belonging to the two categories can be separated by a hyperplane in the N-dimensional space of the inputs, a property known as "linear separability". The vector of synaptic weights which solves the learning problem would then be orthogonal to the separating hyperplane, but if the synaptic weights are not allowed to take arbitrary continuous values, it could be still impossible to achieve perfect learning even in presence of linear separability. Furthermore, it is important

to point out that the theoretical possibility to reach a solution is a different problem from finding a way to find such a solution. These issues will be considered in the next section.

If the inputs are taken from a known distribution, it is possible to define a "storage capacity" of the model as the average number of patterns per synapse which the device can learn to classify without errors. Unfortunately, no means of calculating such a capacity in the general case is known; still, it is possible to derive an asymptotic value in the limit in which the number of synapses N grows very large (known as "thermodynamic limit") for the simplest case, namely that of evenly random and independent inputs, by using techniques derived from the statistical physics of disordered system and information theory.

The most straightforward strategy adopted to calculate the maximum capacity of a device would be the following: first, suppose that the number of patterns to be learned is αN, then calculate the average number of solutions $\langle \mathcal{N}_{sol} \rangle_{\{\xi\}}$ to the learning problem (where $\langle \cdot \rangle_{\{\xi\}}$ denotes the average over the inputs) in the limit of large N, and finally find the value of α at which $\mathcal{N}_{sol}$ goes to zero. The number of solutions for a given α can be expressed as:

$$\mathcal{N}_{sol}(\alpha) = \int \prod_{i=1}^{N} d\mu(w_i) \left(\prod_{\mu=1}^{\alpha N} \Theta \left(\sigma_E^{\mu} \text{sign} \left(\sum_{i=1}^{N} w_i \xi_i^{\mu} \right) \right) \right) \tag{4}$$

where we used the binary ± 1model of eq. 3, letting σ_E^{μ} denote the expected output for pattern μ, and where $d\mu(w_i)$ denotes a measure for the synaptic weights, which can be used to specify if the synapses are continuous, in which case we would substitute:

$$d\mu(w) = dw$$

(in this case the number of solutions $\mathcal{N}_{sol}$ is actually the volume of the solution space) or binary, in which case we would use:

$$d\mu(w) = (\delta(w+1) + \delta(w-1))\, dw$$

where $\delta(x)$is the Dirac delta distribution. The rest of the computation would follow in a straightforward way by the assumption that the inputs are independent.

However, using the average $\langle \mathcal{N}_{sol} \rangle_{\{\xi\}}$ does not yield the desired results. This happens because the distribution of $\mathcal{N}_{sol}$ presents a very sharp peak but also a very long tail, so that its average is different from its mode. As a result, the direct calculation overestimates the capacity with respect to the observations, since every instance of the problem will fall with overwhelming probability in the region of the peak. This is reflected by the fact that the width of the distribution of $\mathcal{N}_{sol} / \langle \mathcal{N}_{sol} \rangle_{\{\xi\}}$ does not tend to zero in the thermodynamic limit, thus making the average a non informative quantity (it is said to be "non self-averaging").

The first successful approach towards this problem [**13, 14**] is based on a technique known as "replica method" [**21**]. The method assumes that the correct self-averaging quantity is not the number of solutions, but its logarithm, which is normally called "entropy", making the parallel with the statistical physics framework in which the replica theory was developed. This means that, in the thermodynamic limit, the average of the entropy per synapse $\varepsilon = \log(\mathcal{N}_{sol})/N$ tends to a finite asymptotic value, and that the variance of its distribution tends to zero. Thus, for sufficiently large N, ε is almost the same for every instantiation of the problem, and is obviously equal to its average:

$$\langle \varepsilon(\alpha) \rangle_{\{\xi\}} = \frac{1}{N} \langle \log(\mathcal{N}_{sol}(\alpha)) \rangle_{\{\xi\}} \tag{5}$$

.

From the mathematical point of view, this approach introduces the difficulty that it is impossible to commute the operators $\langle\cdot\rangle_{\{\xi\}}$and $\log(\cdot)$ directly. The replica method gives a prescription which can be used to overcome this problem, by exploiting the identity

$$\log(x) = \lim_{n\to 0}\left(\frac{x^n - 1}{n}\right)$$

Having the logarithm expressed in this form, it is possible to commute the average and the limit operator:

$$\varepsilon(\alpha) = \lim_{n\to 0}\frac{1}{Nn}\left(\langle(\mathcal{N}_{sol}(\alpha))^n\rangle_{\{\xi\}} - 1\right) \tag{6}$$

The next step is to compute the average over the inputs in the case in which n is a positive integer value: if one succeeds in finding an analytical expression for this case, the limit can be subsequently taken by performing an analytic continuation. Although this step is not (yet) provably safe from the mathematical point of view, the results obtained are in perfect agreement with the observations and with other methods (see e.g. [**25**] and references therein).

The limitation that n be an integer number, allows the n-th power in eq. 6 to be substituted by the product of n identical non interacting replicas of the system, and finally, after some algebraic manipulation, to express the integrand as a function of a number of order parameters. One such parameter is for example the "overlap" between two arbitrary replicas, defined as

$$q_{ab} = \frac{1}{N}\sum_{i=1}^{N} w_i^a w_j^b$$

where a and b are (different) replica indices, and w_i^a, w_i^b are the replicated synaptic weights vectors. At this point, a crucial assumption has to be made about the structure of these order parameters in the replicated phase space; the simplest and most natural is to assume that, since all the replicas are equivalent, such space is perfectly symmetric, and reduce all the overlaps to a single parameter: $\forall a, b:\ q_{ab} = q$. This choice goes under the name of "replica symmetric" (RS) Ansatz; with it, one can reduce the integrand under study to an expression in which n is not required to be a discrete variable any more, and the limit for $n \to 0$ can be taken. The result is an expression of the form

$$\varepsilon(\alpha) = \frac{1}{N}\log\int dq d\hat{q}\exp\left(N\mathcal{F}(\alpha, q, \hat{q})\right) \tag{7}$$

where $\hat{q}$ is the conjugated order parameter with respect to the overlap, and for simplicity we omitted any other order parameter. As N diverges, we can approximate the integral by means of the saddle point method, which amounts at finding a solution to the system of equations

$$\begin{aligned}\frac{\partial}{\partial q}\mathcal{F} &= 0\\ \frac{\partial}{\partial \hat{q}}\mathcal{F} &= 0\end{aligned} \tag{8}$$

which is usually done numerically. Having found the solution q_0 and $\hat{q}_0$, expression 7 finally becomes:

$$\varepsilon(\alpha) = \mathcal{F}(\alpha, q_0, \hat{q}_0)$$

In some cases the RS assumption is sufficient to obtain the exact analytical result. Determining whether the results obtained in this way represent the correct solution or they are an

approximation is possible by inspecting the stability of the saddle point towards the breaking of the replica symmetry, but the calculation is more involved than the one sketched above. The theory of how to treat situations in which the replica symmetry is broken and how to perform such a calculation, along with the physical interpretation of the order parameters and the replica symmetry breaking itself, was first developed by Parisi, Mézard and Virasoro in [**21**] and successfully applied to many different problems [**25**].

For what concerns perceptron models, the RS assumption is sufficient to obtain the exact result both for continuous and binary synapses, but there's a caveat about the latter which will be explained in the next section. However, in both cases, the maximum capacity is given by the value of α for which $\varepsilon(\alpha) = 0$: for continuous, unbounded synapses this is $\alpha = 2$ [**10**]; for binary ± 1 synapses it is $\alpha = 0.83$ [**17, ?**]and for $0/1$ valued binary synapses it is $\alpha = 0.59$, provided that the threshold is set to its optimal value (which is also found by the saddle point method)[**15**].

2.7. Continuous vs binary synapses in supervised perceptron models. Even if the theoretical maximal capacity of the binary synapses model is not much reduced with respect to that of the continuous synapses model, there's still a big difference between the two models with respect to the difficulty of actually finding a solution to the learning problem. For the continuous-valued model, many simple and efficient algorithms are known to perform well on this task, the simplest being the "standard perceptron algorithm" (SP). It is a simple update protocol, which prescribes the modification that the synapses have to undergo every time a pattern is presented to the device:

(1) If the classification of the pattern is correct, nothing is changed
(2) If an error is made, update all the synapses as: $w_i^{t+1} = w_i^t + \eta \sigma_E^t$

where η is a fixed, small learning constant, and σ^t is the expected output for the pattern presented at time t. The patterns can be presented sequentially or randomly.

It can be easily demonstrated that, whenever a solution to the problem exists, there exists a small enough value of the parameter η such that the SP algorithm converges in a finite number of steps, and that the maximum learning time is a polynomial in N. This applies to any input pattern set, regardless of the underlying distribution[**28**]. More complicated algorithms, like e.g. the adatron algorithm or the back-propagation algorithm, can have better performances under appropriate circumstances.

The situation is completely different for binary models: in this case, the learning problem has been proved to be a non-polynomial complete (NP-C) class problem from the algorithmic point of view [**2, 3**]. The NP-C class of problems includes many well known optimization problems like the travelling salesman problem, the satisfiability problem or the coloring problem, and has the property that being able to solve all of the instances of one of the problems of this class in polynomial time (with respect to the size of the input) would allow to solve all of them in polynomial time. It is widely believed that a general solution for any of the NP-C problems which could always succeed in polynomial time does not exist; thus, NP-C problems are considered to be hard from the algorithmic point of view.

On the other hand, the requirement to find a solution for any possible instance of a problem (worst-case scenario) is not always of interest, especially if a very large number of variables is involved. In many situations, the typical learning time would be more important for practical purposes. Defining this quantity requires that a distribution is specified on the parameters of the problem. When the parameters are taken randomly and independently, many complex problems become tractable by cleverly exploiting their statistical properties. Statistical physics methods

have been remarkably useful in this field: one well known algorithm inspired by the analogies between optimization problems and the statistical physics is simulated annealing [**29, 9**]. This method makes a parallel between the quantity to optimize and the interaction energy of a material, and simulates a cooling process towards zero temperature, assuming that, eventually, the material will be in its ground state, i.e. the problem will have reached an optimal solution. The effectiveness of this approach is impressive in some cases, e.g. in the travelling salesman problem with randomly and uniformly distributed nodes, but is of little help in other cases, amongst which is the binary perceptron problem. Keeping the parallel with statistical physics, in some cases the cooling would require to be infinitely slow in order to reach the solution; if the cooling down is too fast, the system can have a transition from a liquid to a glassy, disordered phase, and get stuck. Another way to see the same phenomenon is to say that, for too low temperatures, the energy landscape in the phase space has an overwhelming number of very deep local minima, such that, once one is reached, it becomes impossible to get out of it and reach the global minimum.

In order to be more precise, the exact terms of the problem can be defined by using the concepts of replica symmetry breaking theory and their physical interpretation. The RS analysis of the previous section is exact when the synapses are allowed to assume continuous values: the solutions to the learning problem form a connected component in the phase space, which is a big local minimum in the energy landscape, if we define the energy of the problem as the number of errors the device makes. However, when the synapses are binary, replica symmetry is actually broken, and a "one-step-symmetry-breaking" analysis (1RSB) is needed: the space of all the replicas is assumed to have a clustered structure, so that, given any two replicas a and b, they will either belong to the same cluster and have overlap q_1, or belong to different clusters and have overlap $q_o < q_1$. The same applies to any other order parameters, even though the inequality relation may be reversed. A new parameter also enters into this 1RSB description, which can be interpreted as expressing a clusterization degree, and which mathematically has the role of a temperature; its thermodynamic conjugate is called the "complexity" (an analog to the entropy for the ordinary temperature), and expresses the logarithm of the number of clusters of the solution. Geometrically, the clusters are connected components in the phase space, so that two solution of the problem belonging to the same cluster can be transformed one into the other by small steps without getting out of the cluster itself. Solutions belonging to different clusters will on the contrary be far apart in the phase space, so that macroscopic modifications would be needed to transform one solution in the other. This same structure does not only hold for the solutions, but also for states of higher energy; what's worse, the number of clusters at a given energy is exponentially greater than the number of clusters at a lower energy, which explains why local search algorithms as SP or even simulated annealing get trapped in those exponentially numerous local minima.

The reason for which the RS value $\alpha = 0.83$ for the maximum storage capacity is nevertheless exact is that the 1RSB solution for this specific problem has $q_1 \to 1$, which means that the clusters of solutions tend to become point-like; thus, the overall structure of the solutions has in practice the same symmetry properties it would have if it would be RS, and the saddle point solutions are in one to one correspondence to those of the RS calculation. Nevertheless, it still is a hard problem algorithmically. Algorithms cable of solving problems in the 1RSB phase have been developed only recently [**19, 20, 1, 7**], showing that finding a solution by local search is not impossible even though the energy landscape is as described above. The logic behind such algorithms is to explore the space of the clusters, rather than that of the single states, and to gradually restrict the dynamics to one such cluster until a solution is found.

Using the standard perceptron algorithm SP and its variants on the binary perceptron problem gives indeed very poor results: the learning time grows at least exponentially with the number of synapses N, and the problem becomes intractable very rapidly. This will be shown in the next chapter, where we present a novel simple algorithm which outperforms all other known algorithms on this task, being capable of almost saturating the storage capacity bound, with a convergence time which we estimated to grow as $\mathcal{O}\left(N \log(N)^{1.5}\right)$, i.e. almost linearly.

2.8. Continuous vs binary synapses in unsupervised models. The problem of learning with discrete synapses is much harder than that of learning with continuous synapses also in the unsupervised scenario. Analogously to what happens in the supervised learning scenario, continuous recurrent networks can in principle store a number of attractors of order N, but learning with a limited number of synaptic states only allows to reach an order $\log(N)$ memories, if the fraction of active units at each time does not scale with N. The reason for this bad performance is that in the discrete case each "jump" of the synaptic state overwrites the previously stored information. This has been shown in [**4, 5**] using perceptron models, and further generalized by means of theoretical information techniques in [**11**], where it is pointed out that some degree of meta-plasticity is required in order to prevent the information trace left in each synapse from being erased at an exponential rate, i.e. in order for memories to last more than a logarithmic time.

A binary perceptron model which is able to considerably improve performance in the unsupervised scenario, the "cascade model", has been proposed in [**12**]. In this model, each synapse has only 2 visible states, but can have multiple hidden states, each corresponding to a different degree of plasticity. The transition scheme of the model is shown in Fig. 2. Using the cascade model, a memory trace in a recurrent network undergoes a phase during which the signal decays as a power law, whose duration depends on the number of internal states of each variable, followed by a phase of exponential decay. The exponential tail may be undetectable, and thus negligible, if the transition happens when the trace is already fainter than the noise.

From the biological point of view, meta-plasticity is a quite reasonable hypothesis, since there's no reason for the synaptic strength to be the only variable that enters the plasticity update rules. On the contrary, the synaptic terminals are highly specialized structures with complex dynamics and multiple time scales; representing the internal protein network as a collection of discrete stable states with some transition rule among them is a first-order approximation towards a more biologically plausible description.

The algorithm which we present in the next chapter similarly takes advantage of meta-plastic transitions in order to increase the amount of information retained about its history, but this feature alone is not yet sufficient to achieve satisfactory results in the supervised scenario.

CHAPTER 2

The SBPI algorithm

1. Cavity algorithms

1.1. Cavity methods for statistical physics. The calculation of section 2.6 is based on the replica theory approach to the problem of studying the thermodynamics of disordered systems. Even though its mathematical foundations are not yet assessed, this method has proved to yield the correct results in all the situations in which it has been tested.

Disordered systems are characterized by depending on a large number of parameters, which are considered to be fixed on a single instance of the problem, but which are supposed to be extracted form a probability distribution when looking at the problem in its generality (of course, this approach is only useful thanks to the existence of the self-averaging quantities, i.e. the fact that all systems behave in the same way in the thermodynamic limit). For this reason, these parameters are also called "quenched variables": in the case of the perceptron, it is the set of patterns which has to be learned (both the inputs and their associated outputs). The quenched variables are the source of the disorder.

In the replica method, the average over the quenched variables is performed at the very beginning of the calculation, prior to the Ansatz about the symmetry structure in the replica space. For this reason, the results which are obtained by this method are only able to describe the general properties of the problem under study, but give no information about the specific instances of the problem: in order to obtain those, a different approach is necessary, which is provided by the the "cavity method". The calculations in this case are performed on single instances of the problems, and the average over the quenched variables can (if needed) be performed afterwards, by collecting the results obtained on many different instances. The final results are equivalent with those of the replica method, but the single instance results contain information about the phase space which is specific to that realization of the quenched disorder, and can be used to find the global minima (the solutions) of the problem. The two methods can thus be seen as complementary, one being more suitable for investigation of the global properties of the problem (including the study of the symmetry breaking of the replicated phase space), the other one allowing to inspect single instances.

Virtually all optimization problems can be represented as bipartite graphs, in which there are two types of nodes: the ones representing the variables and the ones representing the constraints, or interactions, among variables, with edges connecting nodes of one type to nodes of the other type (see Fig. 1). Graphs provide a general framework, but their usefulness depends on the specific problem under consideration. All cavity method algorithms are based on message passing along the lines of such graphs: the messages can represent a binary information, a probability, a probability distribution, a distribution of a distribution and so on, depending on the replica symmetry breaking Ansatz. In the case of the Belief Propagation algorithm (BP), which will be discussed in the next section, they represent marginal probabilities over the space of the solutions.

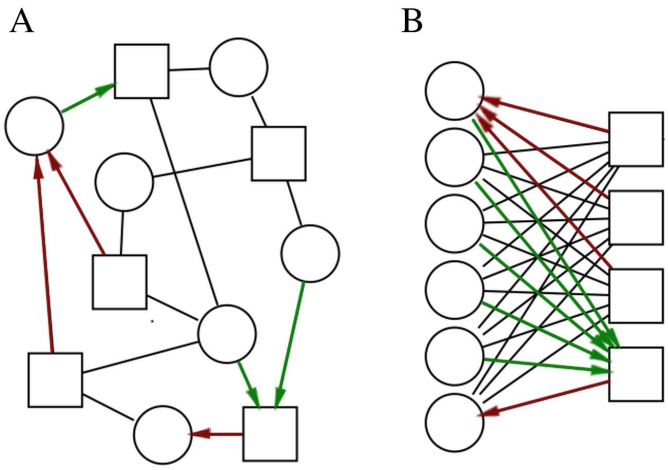

FIGURE 1. Different kinds of bipartite graphs. Rounds represent variables, squares represent constraints or interactions over variables, the edges show which variables the constraints act upon. The red and green arrows represent the two kinds of cavity messages flow; green arrows are messages originated from incoming red arrows, and vice-versa. **A.** Sparse graph. Each interaction involves exactly 3 variables. This could represent for example a 3-SAT problem. **B.** Fully connected graph. Each interaction involves all of the variables. This could represent the supervised learning problem, in which each variable is a synapse and each interaction a pattern to be classified.

The term "cavity" in the name of the method refers to the fact that they exploit the information which can be obtained about the structure of the energy spectrum by taking away one node from the graph at a time and carefully tracking the reshuffling of the energy levels induced by this modification. Of course, everything is simpler in the RS case than in the replica symmetry broken phases, since the clusterization structure is trivial, but still the calculation relies on the fact that the messages flowing through the links connected to the removed node are uncorrelated. This would be strictly true only in a tree-like graph, in which there are no loops, but the absence of loops is never the case for optimization problems or disordered systems. However, if the length of the loops in the problem diverges in the thermodynamic limit, the assumption can be taken as applying asymptotically, and to be a good approximation for large N. In fact, the cavity message passing algorithms prove successful in such situations (Fig. 1A).

On the other hand, looking at the representation of a perceptron problem as a graph (Fig. 1B), it is apparent that the situation is exactly opposite to the tree-like one: the graph is "fully connected", and the loops couldn't be more numerous neither shorter. Despite this fact, the cavity method still works, thanks to the fact that the correlations, due to the very large number of the contributions, tends to cancel out [**34**].

1.2. The Belief Propagation algorithm. The BP algorithm (also known as Bethe-Pierls approximation in statistical physics), has been developed independently in the context of statistical physics of disordered systems and of information theory, in the context of error correction codes for signal transmission (see [**16**]). Its results are equivalent with an RS description of

the problem under study, and it is suitable to describe the perceptron problem for the reasons exposed in sections 2.7 and 1.1. In this scheme, the learning problem is converted in the problem of computing some probability marginals: let us consider the set $\mathcal{W}$ of all the possible weight vectors, and the subset $\mathcal{W}^*$ of all the (unknown) synaptic weights vectors which properly implement the input/output mapping of the patterns. A uniform sampling of this set defines a probability space space over the set $\mathcal{W}^*$ of all the solutions to the problem. Over this space we are interested in single marginals, that is in the probabilities

$$p_i^{\pm} = P_{\mathcal{W}}(w_i = \pm 1) = |\{\mathbf{w} \in \mathcal{W}^* : w_i = \pm 1\}| / |\mathcal{W}^*| \tag{9}$$

that the single synapses take a certain binary value in a randomly chosen solution (here $|\cdot|$ denotes number of element of a finite set).

The computation of these marginals constitutes the first step in the process of finding the optimal synaptic weights, after which one typically proceeds iteratively by fixing the synaptic weights accordingly.

Under the weak correlations assumption, it is possible to write a closed set of equations for the marginals which can be solved efficiently by iteration. In turn, the iteration scheme can be implemented as a distributed computation, a fact which opens the possibility of implementing a dynamical scheme governed by local rules which actually solves the equation and hence provides the marginals we are interested in. This is the feature which allows to revert the BP into a simple enough scheme, to be considered of potential biological interest.

The BP approach consists first in finding the marginal probabilities for synaptic weights w_i on the solutions of restricted (cavity) problems. Thanks to the symmetry of the ± 1 perceptron, we can simplify the notation and assume without loss of generality that, for all patterns μ, $\sigma_{exp}^{\mu} = +1$. Let us first remove pattern μ from the interactions of synapse i. We can define a space of the solutions to the restricted problem as

$$\mathcal{W}_{i\to\mu}^* = \left\{ \mathbf{w} \in \mathcal{W} : \sum_j w_j \xi_j^\nu > 0 \, \forall \nu \neq \mu, \sum_{j\neq i} w_j \xi_j^\mu > 0 \right\} \tag{10}$$

and write the probability that, if the synapse i has synaptic weight w_i, it belongs to such a solution, as

$$p_{i\to\mu}^{w_i} = P_{\mathcal{W}_{i\to\mu}^*}(w_i)$$

where $P_{\mathcal{X}}$ is the uniform measure over $\mathcal{X}$.

Then, on the original graph, let us remove variable i from all patterns but μ. The restricted space of the solutions is defined as

$$\mathcal{W}_{\mu\to i}^* = \left\{ \mathbf{w} \in \mathcal{W} : \sum_{j\neq i} w_j \xi_j^\nu > 0 \, \forall \nu \neq \mu, \sum_{j} w_j \xi_j^\mu > 0 \right\}$$

and the corresponding cavity probability is

$$\eta_{\mu\to i}^{w_i} = P_{\mathcal{W}_{\mu\to i}^*}(w_i)$$

where $P_{\mathcal{X}}$ is defined as above.

These variables can be thought of as messages sent along the graph edges. The two type of messages flow in opposite directions. The BP equations describe how the message form a node to another depends on all the other incoming messages on the sender node, except for the

message coming from the receiver node. Thus, the two signals flowing on each link should in principle carry different pieces of information. In symbols, BP equations read:

$$\eta_{\mu\to i}^{w_i} \propto \sum_{\{w_j : j\neq i\}} \prod_{j\neq i} p_{j\to\mu}^{w_j} \Theta\left[\sum_j w_j \xi_j^\mu\right] \tag{11}$$

$$p_{i\to\mu}^{w_i} \propto \prod_{\nu\neq\mu} \eta_{\nu\to i}^{w_i} \tag{12}$$

where $\Theta[x]$ denotes the Heaviside function ($\Theta[x]=1$ if $x\geq 0$, $\Theta[x]=0$ otherwise), i,j indices run over $1,\ldots,N$ and μ,ν are pattern indices. The $\propto$ symbol indicates normalization prefactors that ensure $\eta_{\mu\to i}^{+}+\eta_{\mu\to i}^{-}=1$ and $p_{i\to\mu}^{+}+p_{i\to\mu}^{-}=1$.

On a solution of Eqs. 11-12, BP estimation of marginals in Eq. 9 can be computed as:

$$p_i^{w_i} \propto \prod_{\mu} \eta_{\mu\to i}^{w_i} \tag{13}$$

The standard way to solve Eqs. 11-12 is by iteration. Calling $S=\left(\{\eta_{\mu\to i}\},\{p_{\mu\to i}\}\right)_{\mu,i}$ and considering the function $f: S\mapsto f(S)$ defined by right-hand sides of Eqs. 11-12, we can build the sequence S_t from the iteration $S_t=f^{(t)}(S_0)$, where S_0 represents some initial condition (either random or for instance uniform), until the distance of two consecutive terms $\|S_{t+1}-S_t\|$ is zero or small enough; then , we can consistently evaluate Eq. 13. From the single variables marginals we can derive other thermodynamic quantities, in particular we can evaluate the entropy of the solution space (which in this case has to be interpreted as a complexity). Computing the average over many different samples at large N, this method yields the same results as the computation performed by means of the replica method.

The information obtained from the marginals in the single instances also allows to find a solution to the learning problem. One approach could be a decimation scheme, in which the most polarized variable (the one for which $\left|p_i^+-p_i^-\right|$ is greatest) is fixed, the corresponding graph is reduced, and the iteration scheme is restarted on the reduced graph, until a solution is (eventually) found. A better approach for this problem is to introduce in the equation a reinforcement term, gradually polarizing all the variables at the same time.

In this forced scheme, the right-hand term in Equation 12 has to be replaced by $p_i^{w_i}\prod_{\nu\neq\mu}\eta_{\nu\to i}^{w_i}$so to drive the system to converge to a single configuration. With weak correlation assumptions, the reinforced equations in terms of $h=\tanh^{-1}(p^+-p^-)$ and $u=\eta^+-\eta^-$ become in a leading order approximation:

$$h_i^{t+1} = \sum_{t'\leq t}\sum_{\nu} u_{\nu\to i}^{t'} \tag{14}$$

$$m_i^{t+1} = \tanh\left(h_i^{t+1}\right) \tag{15}$$

$$u_{\mu\to i}^{t} = \frac{1}{\sqrt{N}} f\left(\frac{1}{\sqrt{N}}\sum_{j\neq i}\xi_j^\mu m_j^t, \frac{1}{N}\sum_{j\neq i}\left(1-(m_j^t)^2\right)\right) \tag{16}$$

where

$$f(a,b)=\frac{1}{\sqrt{b}}\frac{G\left(\frac{a}{\sqrt{b}}\right)}{H\left(-\frac{a}{\sqrt{b}}\right)} \tag{17}$$

and the auxiliary functions G and H are defined by:

$$G(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \tag{18}$$

$$H(x) = \int_x^\infty G(y)\, dy \tag{19}$$

The idea is that in the course of the learning process the 'hidden variables' h_i go progressively towards large positive or negative values, and hence variables m_i become closer and closer to $+1$ or -1. Hence, at the end of the learning process, the synaptic weights can be set to the sign of m_i.

The algorithm as described is able to find a solution in a reasonable time (i.e. not exponentially growing with N) up to $\alpha \sim 0.6$, but higher values can be reached by weakening the reinforcement term of eq. 14, thus producing a slower, but more accurate algorithm, which can store more than $\alpha N \sim 0.7$ patterns, very close to the maximum storage capacity 0.83. In any case, whatever the values of the parameters, the number of steps required for convergence is, up to $\alpha \sim 0.6$, sub-linear in N, thus extremely fast. Note that all other known algorithm's learning time grows exponentially with N for *any* value of α, thus their storage capacity, to the leading order, is 0.

The attractiveness of the BP scheme comes from its distributed nature, allowing to find a global optimal solution from local computations only. However, from a practical point of view, the above presented algorithm is off-line in nature: at each step, all patterns have to interact with the device at once, and each variable has to track an extensive number of analog quantities. This would exclude in practice the possibility of using such a scheme in a biologically plausible context, or more generally in a context in which the input patterns come in sequence, one after the other. However, as we shall see, the fully-reinforced equation set can be easily reverted to an on-line learning scheme.

1.3. BP on 0/1 perceptron. The derivation of the forced BP algorithm for the 0/1 perceptron from the BP scheme closely follows the track of the previous section. The main differences are that in this case the threshold θ is different from 0 and that we need to take explicitly into account what the expected output for the patterns is.

The 0/1 perceptron model was defined in section 2.3; here we add a parameter to the description, the "coding level" f, which represents the average fraction of active inputs ($\xi_i^\mu = 1$) and active outputs ($\sigma^\mu = 1$) per pattern. Both the maximal capacity and the optimal threshold depend on this parameter.

The optimal threshold θ can be computed by means of the replica method, as done in [**15**] (it amounts at adding the equation $\partial_\theta \mathcal{F} = 0$ in the saddle point equations of sec. 2.6), and is a function of α. In the dense coding case $f = 0.5$, the maximal theoretical capacity is $\alpha_{\max} \approx 0.59$, which can be obtained by optimally setting the threshold as $\theta \approx 0.16N$. For lower values of α, the optimal threshold (with respect to the number of solutions of the learning problem) is higher, and reaches $0.25N$ at $\alpha = 0$. However, the simulation results which we will show in the following do not take this into account, as we have found that they were not much affected by the value of θ, and that setting it to the value corresponding to $\alpha_{\max}$ was optimal even at lower values of α.

When varying f the picture is similar; furthermore, the ratio between the optimal value of θ (taken at $\alpha_{\max}$) with respect to the average number of active inputs in each pattern fN is almost constant, going from 0.32 for $f = 0.5$ to 0.30 for $f = 3 \cdot 10^{-3}$. Thus, with these settings, about 30% of the synapses will be active after learning in all cases.

1.4. Reverting BP to an on-line algorithm. The reinforced BP equations can describe an on-line learning protocol by switching to asynchronous update, choosing a time scale τ defined by $N\alpha\tau = t$, and picking the pattern randomly at time τ, giving:

$$m_i^\tau = \tanh\left(h_i^\tau\right) \tag{20}$$

$$h_i^{\tau+1} = h_i^\tau + \frac{\xi_i^\tau}{\sqrt{N}} f\left(\frac{1}{\sqrt{N}}\sum_{j\neq i}\xi_j^\tau m_j^\tau, \frac{1}{N}\sum_{j\neq i}\left(1-(m_j^\tau)^2\right)\right) \tag{21}$$

$$w_i^{\tau+1} = \operatorname{sign}\left(h_i^{t+1}\right) \tag{22}$$

This (on-line) algorithm is fast and solves the learning process up to large values of α, but it still has some unpractical features:

- It requires that each synapse keeps a memory of an analog variable (m_i or h_i);
- The two arguments of the function f have to be computed individually for each synapse.

The second issue can be partly fixed by considering that the function f can be computed at once, and the single synapse values can be obtained as corrections based on purely local information. Then, the algorithm can be further simplified in order to get rid of all the analog variables, while keeping a high capacity and fast convergence; however, the local correction to the global signal is a crucial step for the algorithm's performance.

The corresponding on-line BP-inspired equations for the 0/1 perceptron model are:

$$m_i^\tau = \tanh\left(h_i^\tau\right) \tag{23}$$

$$h_i^{\tau+1} = h_i^\tau + \xi_i^\tau f_{01}\left(\sigma_{exp}^\tau, \frac{1}{2}\sum_{j\neq i}\xi_j^\tau\left(1+m_j^\tau\right), \frac{1}{4}\sum_{j\neq i}\xi_j^\tau\left(1-(m_j^\tau)^2\right)\right) \tag{24}$$

$$w_i^{\tau+1} = \Theta\left[h_i^{\tau+1}\right] \tag{25}$$

where

$$f_{01}(\sigma,a,b) = \frac{1}{2\sqrt{b}}\left(\frac{G\left((a+\sigma-\theta)/\sqrt{b}\right)}{\sigma-(2\sigma-1)H\left((a+\sigma-\theta)/\sqrt{b}\right)}\right) \tag{26}$$

As before, we only need to keep the internal variables h_i, updating them at each time step upon presentation of a pattern $\left(\xi^\tau, \sigma_{exp}^\tau\right)$.

1.5. Discretization. The problem of storing and managing continuous variables can be overcome by crudely simplifying f as a Heaviside step function of the first argument, and replace the tanh with a sign function:

$$m_i^\tau = \operatorname{sign}\left(h_i^\tau\right) \tag{27}$$

$$h_i^{\tau+1} = h_i^\tau + \xi_i^\tau \Theta\left[-\sum_{j\neq i}\xi_j^\tau m_j^\tau\right] \tag{28}$$

where we removed two inessential factors $N^{-\frac{1}{2}}$. This algorithm is roughly equivalent to the continuous one in the last part of the learning process, when all the variables are almost fully polarized. As a last step, we identify the m_i fields with the synaptic weights w_i and avoid the

ambiguous $h_i = 0$ case by initializing all the h_i's to odd values and introducing a factor 2 in their update term, so that they can only assume odd values (this does not affect the results):

$$h_i^{\tau+1} = h_i^\tau + 2\xi_i^\tau \Theta\left[-\sum_{j\neq i}\xi_j^\tau w_j^\tau\right] \tag{29}$$

$$w_i^{\tau+1} = \text{sign}\left(h_i^{\tau+1}\right) \tag{30}$$

In this way, we are left with a single discrete variable for each synapse, and the updating signal is much simpler. The performance of this algorithm, which we refer to as "BP-inspired" algorithm (BPI), is still good, but it achieves a lower capacity then the continuous one (about $\alpha \sim 0.3$). Using a stochastic version of the Θ function turned out to be sufficient to recover the same capacity as the continuous reinforced BP. The procedure will be explained in detail in next section.

The same discretization process can be applied to eqs. 23-26 for the 0/1 perceptron, by substituting m_i by its sign and the function f_{01} by a step function, so that the internal hidden variables h_i can only take integer values; we further restrict them to take odd values, and the equations become:

$$h_i^{\tau+1} = h_i^\tau + 2\xi_i^\tau\left(2\sigma_{exp}^\tau - 1\right)\Theta\left[-\left(2\sigma_{exp}^\tau - 1\right)\left(\sum_{j\neq i}\xi_j^\tau w_j^\tau - \theta\right)\right] \tag{31}$$

$$w_i^{\tau+1} = \Theta\left[h_i^{\tau+1}\right] \tag{32}$$

1.6. Algorithms definitions. The standard perceptron algorithm presented in section 2.7 uses continuous weights and a learning step parameter η. However, if the synaptic weights are unbounded, this algorithm can be discretized by simply rescaling everything by a factor η^{-1}. Thus, we can redefine the SP algorithm by the single equation

$$w_i^{\tau+1} = w_i^\tau + 2\xi_i^\tau \Theta\left[-\sum_j \xi_j^\tau w_j^\tau\right] \tag{33}$$

which is very similar to Eq. 29. The main difference is of course given by the fact that in the SP algorithm the variables are not binary and no hidden values are present. The single equation 33 can be split in a two-rules prescription for clarity:

SP algorithm. Upon presentation of a pattern, the overall depolarization is computed as $\Delta = \sum_j \xi_j^\tau w_j^\tau$, then

(1) If $\Delta > 0$, then $w_i^{\tau+1} = w_i^\tau$ (do nothing)
(2) If $\Delta < 0$, then $w_i^{\tau+1} = w_i^\tau + 2\xi_j^\tau$ (update all the synapses)

Note that the depolarization can only assume odd values, due to the simplifying assumption that N is odd. The straightforward way to turn this algorithm to work on a binary device would be to simply bound the weights to assume only the values ± 1, but the results of this strategy, and of its variants, are extremely poor. Another possibility is to turn the synaptic weights which undergo the updating to hidden variables, while using their sign as the actual synaptic weights. This is known as "clipped perceptron" algorithm (CP):

CP algorithm. Upon presentation of a pattern, the overall depolarization is computed as $\Delta = -\sum_j \xi_j^\tau w_j^\tau$, then

(1) If $\Delta > 0$, then $h_i^{\tau+1} = h_i^\tau$ (do nothing)
(2) If $\Delta < 0$, then $h_i^{\tau+1} = h_i^\tau + 2\xi_j^\tau$ (update all the synapses)

Then update the synaptic weights as $w_i^\tau = \mathrm{sign}\,(h_i^\tau)$.

This algorithm performs much better than the "cropped" one in which there are only two states; nevertheless, convergence still takes an exponential time in N. The hidden variables h_i implement a form of meta-plasticity, since they participate in the learning process, but they don't alter the unit's output but by their sign.

Another algorithm which uses meta-plastic states to boost learning in a binary synapses context is the cascade model mentioned in section 2.8, but such model was introduced in an unsupervised learning scenario, and, though being better than the "cropped" model, it performs worse than CP in the supervised context.

In fact, the CP algorithm and the BPI algorithm described by eqs. 29-30 are very similar, the only difference occurring when $\Delta = 1$:

BPI algorithm. Upon presentation of a pattern, the overall depolarization is computed as $\Delta = -\sum_j \xi_j^\tau w_j^\tau$, then

(1) If $\Delta > 2$, then $h_i^{\tau+1} = h_i^\tau$ (do nothing)
(2) If $\Delta = 1$, then $h_i^{\tau+1} = h_i^\tau + 2\xi_j^\tau \Theta\left[h_i^\tau \xi_j^\tau\right]$ (update only the synapses for which $w_i^\tau = \xi_i^\tau$)
(3) If $\Delta < 0$, then $h_i^{\tau+1} = h_i^\tau + 2\xi_j^\tau$ (update all the synapses)

Then update the synaptic weights as $w_i^\tau = \mathrm{sign}\,(h_i^\tau)$.

Rule 2 in the BPI algorithm is only applied when the unit's output is barely correct, meaning that a single synaptic flip could potentially produce a classification error. In this case, those synapses which are crucial for the correct response are updated by pushing them away from 0, thus reducing the chance of a switch. Even though this rule is applied when a specific value of Δ is found, the overall effect is not negligible. Note that this rule is directly inherited from the cavity procedure, because it acts on those variables which, if removed, would change the outcome of the device.

In order to investigate the effect of rule 2 on performance, we also simulated a stochastic version of the BPI algorithm, in which such a rule is only applied with probability p_s for each presented pattern:

SBPI algorithm. The same as BPI, but rule 2 becomes:

2.: If $\Delta = 1$, then
- with probability p_s: $h_i^{\tau+1} = h_i^\tau + 2\xi_j^\tau \Theta\left[h_i^\tau \xi_j^\tau\right]$ (update only the synapses for which $w_i^\tau = \xi_i^\tau$)
- with probability $1 - p_s$: $h_i^{\tau+1} = h_i^\tau$ (do nothing)

The SBPI algorithm thus actually comprises both the BPI algorithm, when $p_s = 1$, and the CP algorithm, when $p_s = 0$.

The CP, BPI and cascade algorithms are sketched in Fig. 2.

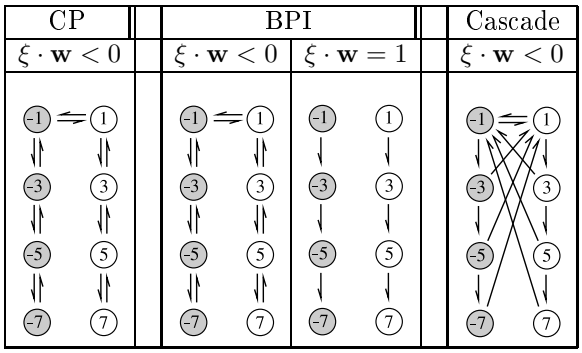

FIGURE 2. Schematic representation of transitions between synaptic states in the CP algorithm and the BPI algorithm. The cascade model introduced by Fusi et al [**12**] is shown for comparison. Circles represent the possible states of the internal synaptic variable h_i. Grey circles correspond to $w_i = -1$, white ones to $w_i = 1$. Clockwise transitions happen when $\xi_i = 1$, counter-clockwise when $\xi_i = -1$. Horizontal transitions are plastic (change value of synaptic efficacy w_i), vertical ones meta-plastic (change internal state only). Downwards transitions make the synapse less plastic, upward ones more plastic. When the output of the neuron is erroneous, $\xi \cdot w < 0$: transitions occur to the nearest neighbor internal state. In the CP algorithm, when the output is correct, $\xi \cdot w > 0$: no transitions occur. In the BPI algorithm, when the output is barely correct $\xi \cdot w = 1$ (a single synaptic flip could have caused an error): transitions are made towards less plastic states only. When the output is safely correct, $\xi \cdot w > 1$: no transitions occur. In the cascade model, 'down' transitions are towards nearest neighbors, while 'up' transitions are towards the highest state with opposite sign. Transition probabilities decrease with increasing $|h|$, see[**12**] for more details

1.7. Performance comparison. The performance of the SBPI algorithm was first investigated numerically with unbounded hidden variables, for different values of α, N and p_s. It turns out that it performs remarkably well, provided the probability p_s is chosen appropriately - with $p_s \approx 0.3$ the system can reach a capacity of order 0.65 with a convergence time that increases with N in a sub-linear fashion (see Fig. 3). On the other hand, the deterministic BPI ($p_s = 1$) has a significantly lower capacity ($\alpha \approx 0.3$), but for those lower values of α it performs significantly faster than the SBPI algorithm - for $\alpha = 0.3$ the time increases approximately as $(\log N)^{1.5}$, as shown in Fig. 3D. As an example, the algorithm perfectly classifies 38400 patterns with 128001 synapses with around 35 presentations of each pattern only. By eliminating completely rule 2 (i.e. CP) convergence time becomes exponential in N rather than logarithmic, for every tested value of α, as shown by the supra-linearity of the blue curves in Fig. 3. Hence, the specificity of rule 2 with respect to synapses (only synapses that actually went in the right direction for the current pattern should be modified) is a crucial feature which makes the BPI algorithm qualitatively superior. Moreover the convergence time increases only mildly with α, as shown in Fig. 3.

We also find that there is a tradeoff between convergence speed and capacity: for each value of α, there is an optimal value of p_s that minimizes average convergence time. This optimal

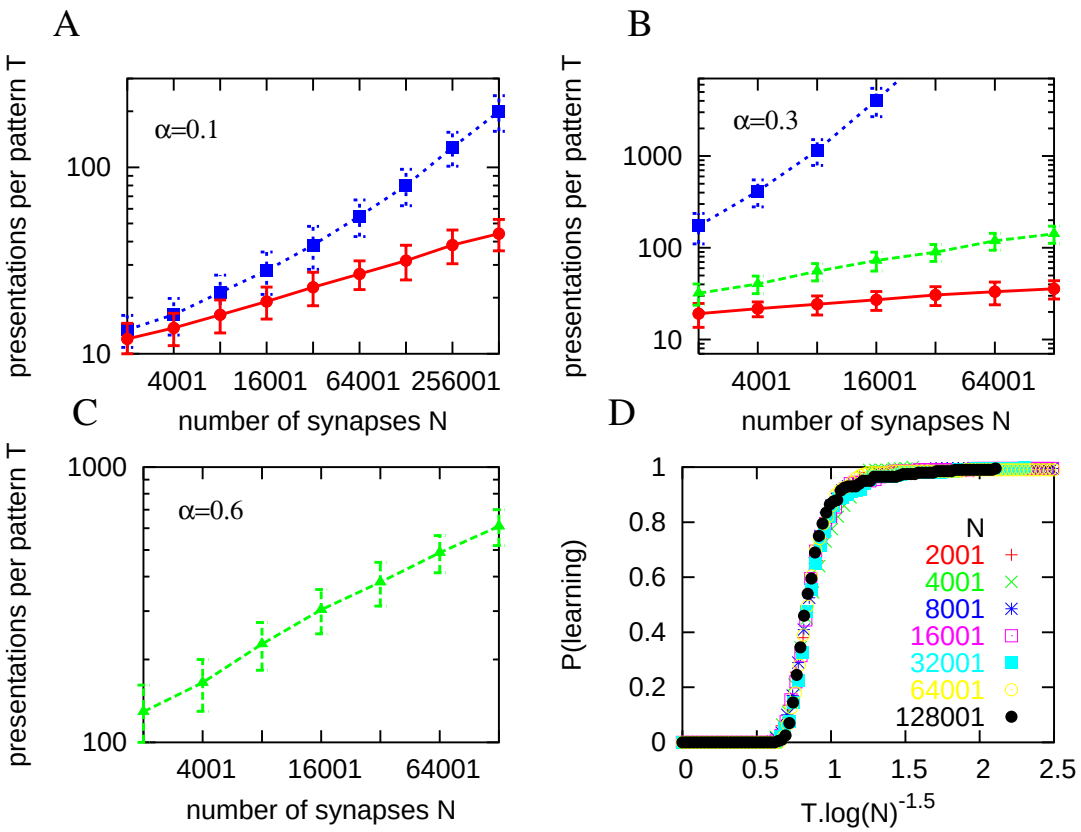

FIGURE 3. Performance of the BPI algorithm with unbounded hidden variables: **A-C** convergence time vs. N for different values and α (indicated on each graph). Points correspond to number of iterations per pattern until the algorithm converges averaged over 200 pattern sets, vertical bars are standard deviations. Blue lines: CP, Red lines: BPI, Green lines: SBPI with $p_s = 0.3$. The latter is the only one which can reach $\alpha = 0.6$, but performs worse than BPI for $\alpha \leq 0.3$ (it is absent from panel **A** for clarity). **D.** Probability that the BPI algorithm learns perfectly $0.3 \cdot N$ patterns in less than $T = x \cdot \log(N)^{1.5}$ iterations per pattern vs x for various values of N

value decreases with α; for $\alpha = 0.3$ it is close to 1, and decreases to 0.3 at $\alpha = 0.65$. Hence, decreasing p_s enhances the capacity, at the cost of a slower convergence; nevertheless Fig. 3C shows that for values of $\alpha \leq 0.60$ SBPI ($p_s = 0.3$) learns perfectly the set of input-output associations in a time that scales sub-linearly with N. Above $\alpha \geq 0.7$ the algorithm fails to solve instances in a time shorter than the chosen cutoff time of 10^4. Note that for $p_s = 0.3$ the convergence time depends in a more pronounced way on α than in the $p_s = 1$ case.

We have also investigated an algorithm in which p_s is itself a dynamical variable that depends on the fraction of errors averaged over a long time window - such an algorithm with an adaptive p_s is able to combine faster convergence at low values of α with high capacity associated with low values of p_s (not shown).

1.8. Bounded hidden variables. We now turn to the situation when there is only a limited number of states K of the hidden variables h_i, since it is unrealistic to assume that a single synapse can maintain an arbitrarily large number of hidden states. Thus, we investigated the performance of an algorithm with symmetrical hard bounds on the values of the hidden states, $|h_i| \leq K-1$ for all i.

Figure 4 shows what happens when the number of internal states is kept fixed while varying N. For the number of states we have considered, $(10 \leq K \leq 40)$, the optimal value of p_s is 1, since in general the stochastic version of the algorithm requires a larger number of states to be efficient. Here, we defined the capacity as the number of patterns for which there is 90% probability of perfect learning in 10^4 iterations, and plotted in Fig. 4 the corresponding critical α against N for different values of the states number K, comparing BPI, CP, and the cascade model (defined as in Fig. 2). We also compared these algorithms that have only 2 'visible' synaptic states but K hidden states, with the SP algorithm with K 'visible' states, $w_i = h_i$.

It turns out that BPI achieves a higher capacity than the SP algorithm with K visible states, when K is fixed and N is sufficiently large, even though the maximal capacity of the binary device is lower. Interestingly, adding an equivalent of rule 2 to the SP algorithm allows it to overcome BPI. This issue is further discussed in section 1.13.

It is also interesting to note that at very low values of N, performance is better using 20 states than with an infinite number of states. Intuitively, this may be due to the fact that in the unbounded case some synapses are pushed too far and get stuck at high values of h_i, i.e. they lose all their plasticity, while a solution to the learning problem would require them to come back to the opposite value of w_i.

The last panel in Fig. 4 compares how convergence time changes with α for the same four algorithms, with the same number of synapses and same number of states per synapse: while the cascade model has a clear exponential behavior, the BPI and SP algorithms maintain nearly constant performance almost up to their critical point. The CP algorithm is somehow in between, its performance degrading rapidly with increasing α (note the logarithmic scale).

1.9. Distribution of hidden variables. Fig. 5A shows the final distribution histogram of the hidden variables h_i for one sample with $N = 64001$ after learning with $\alpha = 0.3$, for the BPI algorithm. When the number of allowed states is infinite, the distribution has the shape of two bell-like curves. The width of the distribution is proportional to $\sqrt{N}$, as shown in Fig. 5B. The shape and the scaling will be discussed in the generalization context, where they find an analytical explanation (section 2).

Introducing an upper and lower bound on h leads to the appearance of two peaks in the distributions at these bounds. These bounds stop the synapses that would otherwise tend to go to very large positive or negative values. If the bounds are large enough, this has no adverse effect on learning because those synapses that reach such large values of h never change sign during the learning process. Reducing further the number of states starts to affect the shape of the whole distribution when the value of the bounds becomes smaller than the location of the peaks of the distribution in the unbounded case. At this point the whole distribution changes, and the convergence time starts to change compared to the unbounded case.

1.10. Optimal value of the number of hidden states K. In order to determine the optimal number of internal states K for a given number of synapses N, we performed some test with N ranging from 1001 to 32001 and looked for the value of K which maximized capacity. Fig. 6 shows that the optimal number of internal states K scales roughly like $\sqrt{N}$, both in the ± 1 and in the 0,1 scenarios.

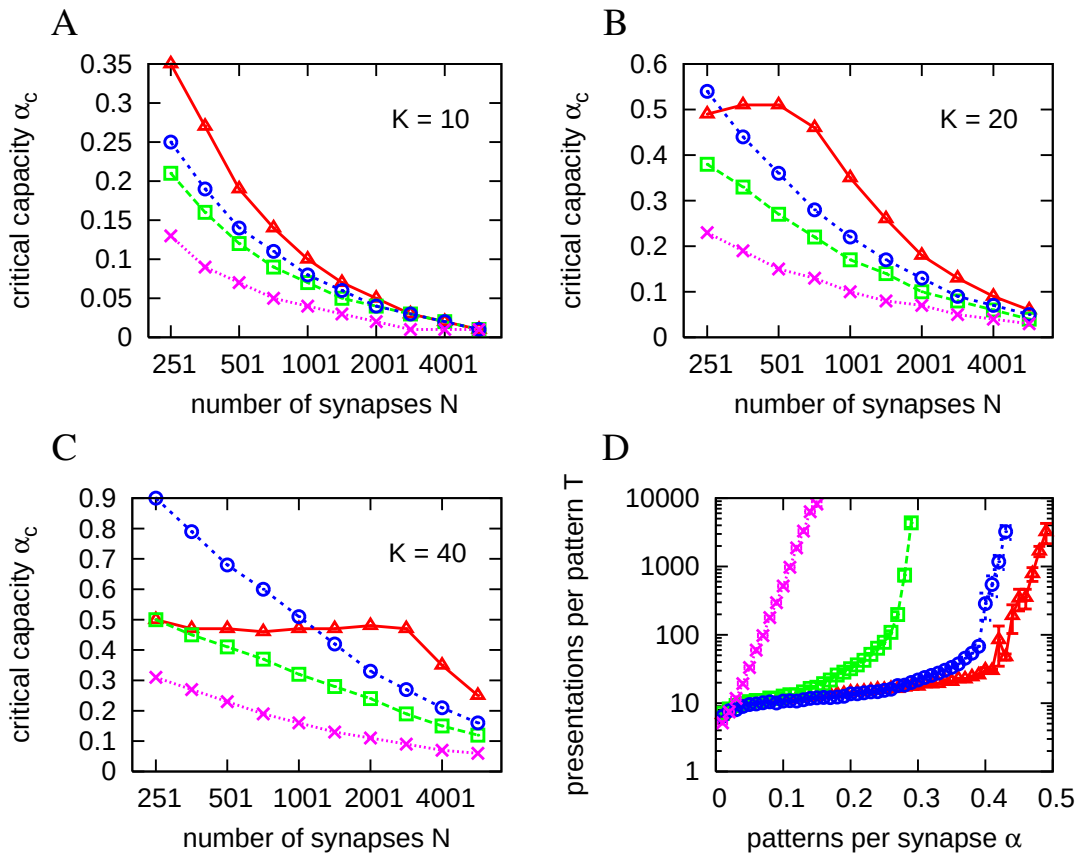

FIGURE 4. Performance of various algorithms with hard-bounded hidden variables. Red: BPI, green: CP, blue: SP, purple: cascade model. **A**, **B**, **C**. Critical capacity vs N, with fixed number of internal states K. **D**. Convergence time vs α at $N = 1415$, $K = 40$, averaged over 100 samples. Figures for different number of states and synapses are qualitatively similar

Following the observation that an appropriate number of internal states K can increase BPI capacity, we searched for the value of K that optimizes capacity, and found that it scales as $\sqrt{N}$, consistent with the distribution of hidden states. The fact that the capacity is optimal for a finite value of K makes the **BPI** algorithm qualitatively different from the other three, whose performance increases monotonically with K.

For a system with a number of states that optimizes capacity, the optimal value for p_s is 0.4, rather than 0.3 as in the unbounded case. With these settings it is possible to reach a capacity α_c of almost 0.7 bits per synapse, very close to the theoretical limit $\alpha_{\max} \simeq 0.83$. Convergence time at high values of α scales roughly linearly with N, but with a very small prefactor ($\approx 2 \cdot 10^{-3}$).

1.11. A more biologically plausible algorithm. The ± 1 perceptron model with dense coding (equal probability of $+$ or -1 inputs) is studied mainly for its simplicity, but it has

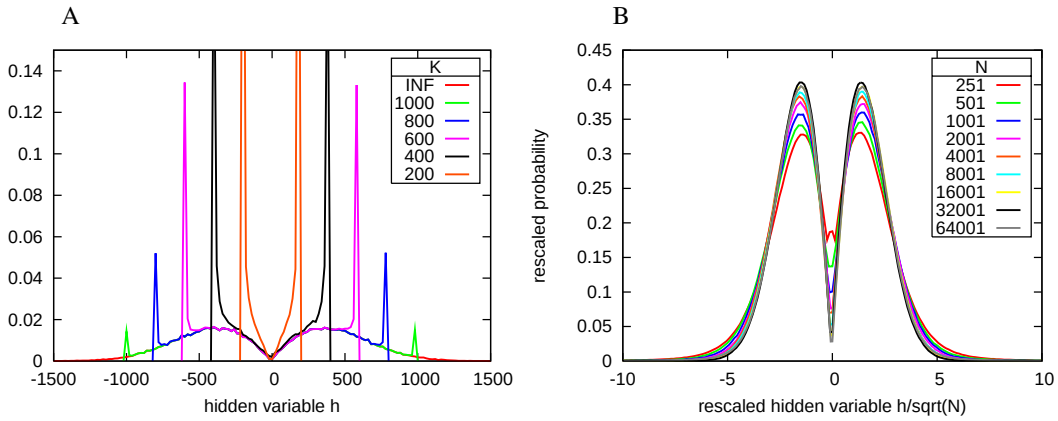

FIGURE 5. Distribution of hidden variables after convergence in the **BPI** algorithm for $\alpha = 0.3$. **A.** Effect of bounds on the hidden variable h on its distribution, shown for $N = 64001$; number of hidden states indicated in the figure. Histogram step size: 20. **B.** The width of the distribution is proportional to $\sqrt{N}$: Histograms obtained with different values of N are shown rescaled by $1/\sqrt{N}$.

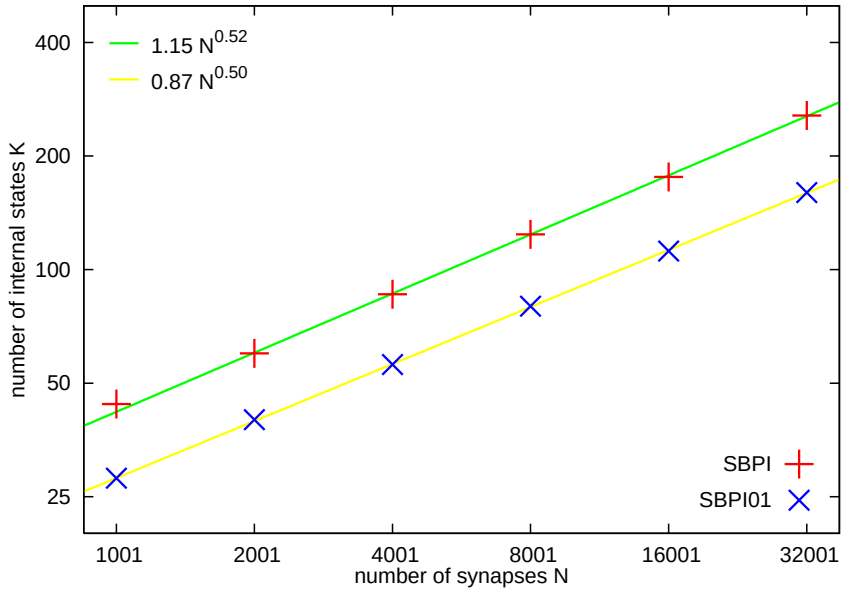

FIGURE 6. Optimal number of hidden states vs N for **SBPI** and **SBPI01**. Number of samples goes from 100 to 20. The fits show that the scaling is close to $\sqrt{N}$ in both cases. Precise values of the fit parameters are: **SBPI** coefficient: 1.15 ± 0.06, exponent: 0.521 ± 0.005; **SBPI01** coefficient: 0.87 ± 0.02, exponent: 0.502 ± 0.003

the biologically implausible feature of symmetry between the two states of activity. Thus, for biological modeling purposes, the 0/1 model is surely more appropriate.

Comparing Eqs. 29-30 and Eqs. 31-32 it is immediate to see the differences between the two models: **(i)** The quantity to be evaluated at each pattern presentation is no longer $\Delta = \sum_j \xi_j^\tau w_j^\tau$, but rather $\Delta = \left(2\sigma_{exp}^\tau - 1\right)\left(\sum_j \xi_j^\tau w_j^\tau - \theta\right)$, which is positive if the pattern is correctly classified and negative otherwise. **(ii)** Synaptic weights are now computed as $w_i^\tau = \Theta\left[h_i^\tau\right]$, making the synapse active (inactive) if the hidden variable is positive (negative), respectively. The performance of this algorithm is qualitatively very similar to the one for the ± 1 case, with a lower capacity - about 0.25, to be compared with a theoretical limit of 0.59 [**15**].

Like before, we have studied a stochastic version of the algorithm in which rule 2 is applied with probability p_s, but we also found out that for this algorithm it was optimal to use this rule only for those patterns which require $\sigma_{exp}^\mu = 0$. We have also introduced a new parameter, θ_m, the threshold for applying rule 2. The SBPI01 algorithm was then defined as:

SBPI01 Algorithm. Compute$\Delta = \left(2\sigma_{exp}^\tau - 1\right)\left(\sum_j \xi_j^\tau w_j^\tau - \theta\right)$, then

(1) If $\Delta \geq \theta_m = 1$, then $h_i^{\tau+1} = h_i^\tau$ (do nothing)
(2) If $0 \leq \Delta < \theta_m = 1$, then
   (a) If $\sigma^\tau = 0$, then, with probability p_s, do $h_i^{\tau+1} = h_i^\tau + 2\xi_j^\tau\left(1 - w_j^\tau\right)$ (update only synapses with $w_i^\tau = 0$, $\xi_i^\tau = 1$)
   (b) Else $h_i^{\tau+1} = h_i^\tau$ (do nothing)
(3) If $\Delta < 0$, then $h_i^{\tau+1} = h_i^\tau + 2\xi_i^\tau\left(2\sigma^\tau - 1\right)$ (update all the synapses)

Since rule 2 is only applied to patterns with $\sigma_{exp}^\mu = 0$, the meta-plastic changes affect only silent synapses (for which $w_j^\tau = 0$) involved in the pattern (those for which $\xi_i^\tau = 1$). Note that using rule 2 only for patterns for which $\sigma^a = 0$ not only optimizes performance, but also makes the algorithm simpler, since in this way there is only the need for one secondary threshold $(\theta - \theta_m)$ instead of two (which would have been required if rule 2 had to be applied in all cases). The opposite choice, i.e. using rule 2 only for patterns for which $\sigma_{exp}^\mu = 1$, can also be taken with similar results.

As in the preceding case, introducing boundaries for the hidden variables h_j can further improve performance, and the number of states K which maximizes capacity scales again roughly as $\sqrt{N}$ (see Fig. 6), while reducing K too much hinders the algorithm's behaviour. In the case of dense coding, $f = 0.5$, and using the optimal value $p_s = 0.4$, SBPI01 can reach a storage capacity α_c beyond 0.5 bits per synapse for sufficiently high N, very close to the maximum theoretical value $\alpha_{\max} \simeq 0.59$.

1.12. Heterogeneous synapses and sparse coding. One possible way to increase capacity with a very limited number of available states is to use 'sparse' coding, i.e. a low value for f. In an unsupervised learning scenario, it has been shown that purely binary synapses (e.g. only two hidden states) can perform well if f is chosen to scale as $\log N/N$ [**33, 5**]. In order to test the SBPI algorithm in a harder scenario, we chose an intermediate scaling $f = 1/\sqrt{N}$. In addition, we also introduced heterogeneity in synaptic efficacies. Possible synaptic weights were no longer 0 and 1, but 0 and a_i where a_i was drawn from a Gaussian distribution with mean 1, and standard deviation 0.1. Likewise, the threshold θ_m used for the implementation of rule R2 was drawn randomly at each pattern presentation from a Gaussian distribution centered in 1 with

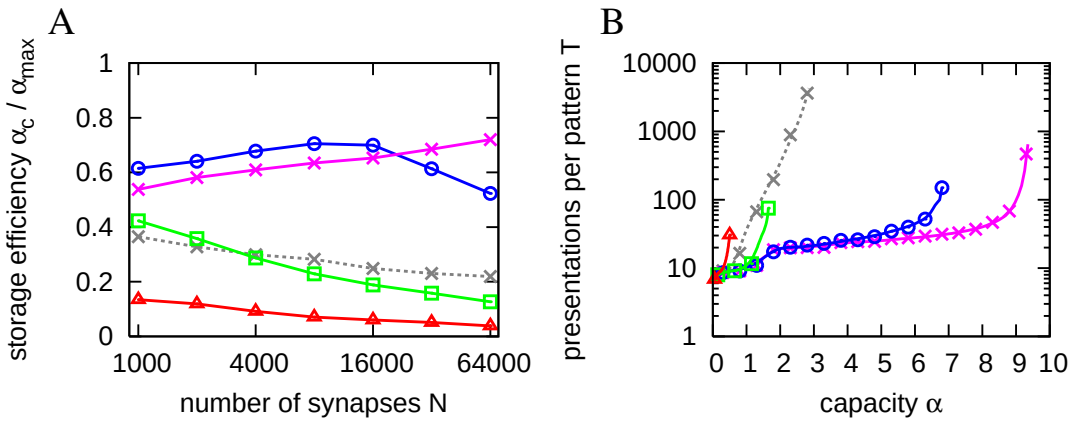

FIGURE 7. Performance of SBPI-Het for different number of states K, with coding level $f = N^{-\frac{1}{2}}$. Number of samples ranges from 100 for $N = 1000$ to 10 for $N = 64000$. Triangles: $K = 2$, Squares: $K = 4$, Circles: $K = 10$, Crosses: $K = 20$. Dashed line: cascade model with $K = 20$. **A.** Storage efficiency vs N **B.** Convergence time vs α for $N = 64000$

variance 0.1. The resulting algorithm SBPI-Het was shown to have very similar performance to SBPI01 in the $f = 0.5$ case.

In Fig. 7A we show the maximum capacity α_c (defined as for Fig. 4) reached in the sparse coding case divided by the maximum theoretical value $\alpha_{\max}$ (which depends on f), with $p_s = 1$, N ranging from 1000 to 64000 and low number of internal states. The figure shows that a synapse with only two states (i.e. with no meta-plasticity) has a capacity of only about 10% of the maximal capacity in the whole range of N investigated. Adding hidden states up to $K = 10$ improves significantly the performance, which reaches about 70% of the maximal capacity for sizes of N of order 10000. In fact, for such values of N the capacity decreases when one further increases the number of states. The optimal number of states increases with N as in the dense coding case, but with a milder dependence on N. In fact, simple arguments based on unsupervised application of rule 2 predicts in this case an optimal number of states scaling as $N^{1/4}/\sqrt{\log N}$, which seems to be roughly consistent with our numerical findings. Fig. 7B shows convergence time versus α for $N = 64000$. It demonstrates again the speed of convergence of the SBPI algorithm, while the cascade model is significantly slower.

1.13. Binary vs K state synapses. In order to make the problem of learning with binary synapses tractable, we ended up 'hiding' a multi-state variable inside each synapse. This raises naturally the question of the practical usefulness of such a device: from the architectural point of view, it may be questionable whether it is better to use a binary device with K hidden states than one with K visible states; in fact, the latter has a greater theoretical capacity. However, the BPI algorithms can be superior either when the learning phase and the recalling phase are totally distinct or in presence of noise or unreliable devices.

The hidden variables are only necessary during learning; thus, the overhead required for storing and managing the hidden variables may be limited to that period. Note that this was already possible using the original BP algorithm, but the BPI version is both faster and much easier to implement. In an on-line setting, in which learning has to occur in real time, noise

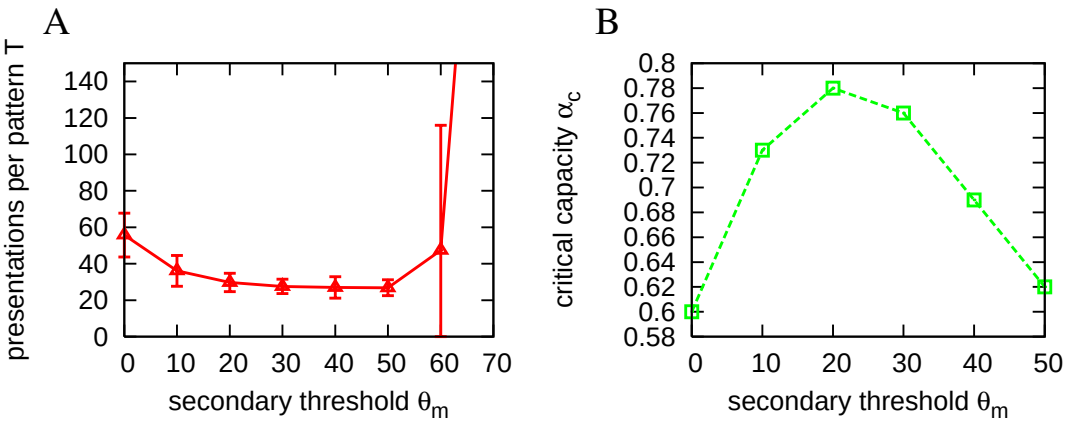

FIGURE 8. Performance of the MP algorithm with $N = 4001$ synapses and $K = 100$ visible states as the secondary threshold θ_m varies. **A.** Convergence time, averaged over 50 samples of $0.5N$ patterns each. **B.** Maximum achieved capacity (at least 90% successes on 25 samples, with cutoff time of 1000 presentations per pattern).

resistance is the primary reason for using binary synapses; this issue is discussed in the next section.

Interestingly, the rule set we propose for BPI becomes useful even when using a device with a limited number of visible states K: in this case, the learning problem rapidly becomes hard from the algorithmic point of view as N gets large. The binary case is the extreme example of this situation; as we have shown in Fig. 4, the SP algorithm may perform worse than the SBPI algorithm with the same number of states in such a situation. Since the capacity of the visible-state device has to be greater than that of the binary device, the reduced efficiency is due to the SP algorithm. We found that some efficiency could be recovered by using a modified version of this algorithm, in which an analog of the rule R2 for BPI was added. The modified perceptron algorithm MP was defined as:

MP algorithm. Upon presentation of a pattern, the overall depolarization is computed as $\Delta = \sum_j \xi_j^\tau w_j^\tau$, then

(1) If $\Delta > \theta_m$, then $w_i^{\tau+1} = w_i^\tau$ (do nothing)
(2) If $0 < \Delta \leq \theta_m$, then $w_i^{\tau+1} = w_i^\tau + 2\xi_i^\tau \Theta\left[w_i^\tau \xi_i^\tau\right]$ (only update synapses for which w_i is on the correct side)
(3) If $\Delta < 0$, then $w_i^{\tau+1} = w_i^\tau + 2\xi_j^\tau$ (update all the synapses)

This is very similar to the BPI algorithm, in which the h_i's are replaced by the w_i's and the secondary threshold is $\theta_m \neq 1$. The SP algorithm can be recovered by setting $\theta_m = 0$.

Fig. 8 shows that both convergence speed and storage capacity are higher with $\theta_m \neq 0$; the optimal value is different for different tasks (cfr. Fig. 8A and B), and has a strong dependence on the number of states K (not shown). With the proper settings, this algorithm reaches slightly higher capacities than BPI even with very few states K compared to the number of synapses N, though being still more sensitive to noise, as discussed in the next session.

1.14. Robustness against noise. Binary devices have the advantage of simplicity and robustness against noise. Here, we address the issue of resistance against noise which might affect the multi-stable hidden states. Intuitively, the fact that the synaptic weights in the BPI algorithm only depend on the sign of the corresponding hidden variables, suggests that a device implementing such learning scheme would be more resistant against accidental changes in the internal states with respect to a device in which the multi-stable state is directly involved in the input summation.

We tested this in two different situations: one in which noise is added during the learning process and afterwards, and another one in which it is only applied after learning has occurred. The first setting mimics the situation in which the multi-stable elements representing the internal states are not reliable on the learning time scale; the latter represents a situation in which learning sessions occur on much faster time-scales compared to the time during which the stored memories have to be available for recalling.

We compared a binary device with hidden states (implementing SBPI) with a perceptron with visible states implementing a standard perceptron algorithm SP and the modified version described in section 1.13, MP. For proper comparison, all of these devices had the same number of synapses $N = 4001$ and the same overall number of stable states (K hidden states of BPI were compared to K visible states of the standard perceptron). The optimal value (the one maximizing robustness) of the secondary threshold for the MP algorithm was found to be $\theta_m \approx 30$ for the bounded case $K = 100$ and $\theta_m \approx 180$ for the unbounded case.

Protocol 1. We added gaussian noise to the multi-stable states during the learning process, once after each presentation of the whole pattern set. The process was carried on even after perfect learning was eventually achieved. We generated random numbers according to a normal distribution with standard deviation z, truncated them towards 0, doubled them and added them to the states value (truncation is needed in order to keep the state values integer, doubling to keep them odd). Thus, using $z = 1$ for example, each synapse had a 68% probability of staying unchanged, a 28% probability of making one step upwards or downwards, etc. Each run consisted in 10000 presentations per pattern; as a measure of robustness, we averaged the number of errors made by each device in the last 1000 presentations, a time at which it has reached its asymptotic value. The results are shown in Fig. 9A-B. The binary device shows a higher resistance to noise: even at the lowest noise level, $z = 1$, the K-visible state device was unable to keep the error rate to 0.

Protocol 2. Each simulation was divided into a short learning period (200 presentations per pattern) and a longer recalling period during which noise was applied and memories were tested without any further learning. The protocol for noise application was the following: at each iteration, each synapse had a fixed probability $p_Z = 0.1$ to switch one state up or down with equal probability. After each iteration, the whole pattern set was probed and the corresponding number of errors recorded. Note that the time scale of the recalling period is arbitrary with respect to that of the learning period. Results are shown in Fig. 9C-D. We found that the binary device with K hidden states was remarkably more robust than the K-visible state device, especially at short times. Of course, in the limit of very long times all three rules perform equally badly, since all memory of the stored patterns is erased, but at any finite time the system with binary synapses is significantly better.

2. Generalization protocol

With the learning protocol that we have used so far, making analytical predictions about the synapes' dynamics under the SBPI algorithm is a very hard task. The reason for this is that

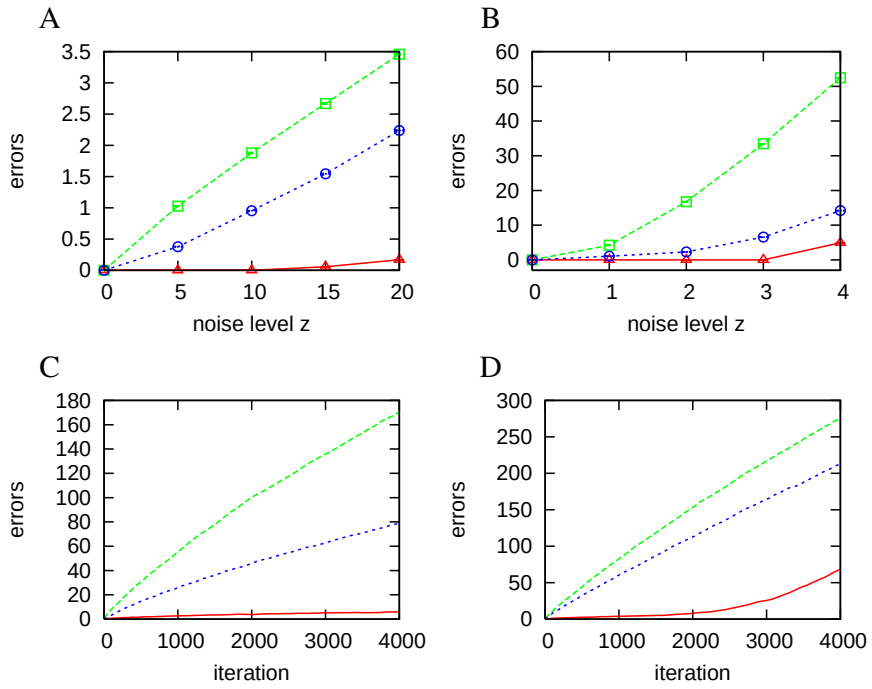

FIGURE 9. Robustness to noise of various algorithms. In all tests we used $N = 4001$ synapses trained on $0.55N$ patterns; Red lines: SBPI with parameter $p_s = 0.4$, Green lines: SP, Blue lines: MP with optimal value for θ_m. Results for both bounded ($K = 100$) and unbounded cases are shown. Points were obtained by averaging over 25 samples for protocol 1, 100 samples for protocol 2. **A.** Protocol 1, unbounded case. **B.** Protocol 1, bounded case. **C.** Protocol 2, unbounded case. **D.** Protocol 2, bounded case.

patterns have to be presented repeatedly, which means that the temporal history of the inputs has very strong correlations.

Here instead we will consider a generalization protocol, in which a general classification rule has to be learned from a continuous stream of random patterns, never repeating. In order to ensure that a solution exists, we generate the expected output of the patterns from a teacher perceptron, and train a student perceptron on that value. Thus, the goal is equivalent to reaching a perfect overlap with the teacher, which can be thought of as the student having learned an association rule. As in the previous case, we can simplify the notation, we can always train the student only on patterns whose expected output is $+1$, 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 expected output $\sigma_{exp}^\tau = +1$. Also, we can assume, without loss of generality, that all the teacher's synapses are set to $w_i^T = +1$. That's because, even being so, they don't acquire any special property, and they are hidden from the student, which is initialized at random. This implies that the student will only be presented patterns in which there are more positive than negative inputs, and that "positive

synapse" can be read, in what follows, as being synonymous to "correctly set synapse", while "negative synapse" can be read as "wrongly set synapse".

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, $q = \frac{1}{N}\left(w \cdot w^T\right) = 1$.

2.1. Histogram dynamics for the CP algorithm. We will do a mean-field-like approximation to the problem: at each time step, we suppose that we know the histogram distribution of the hidden variables at a time τ, $\mathrm{HIST}^\tau\left(\{h_i\}\right)$; then we compute the average distribution (over the input patterns) at time $\tau+1$, $P^{\tau+1}\left(\{h_i\}\right)$, and finally we identify this with the new histogram, $\mathrm{HIST}^{\tau+1} = P^{\tau+1}$.

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, in symbols $\nu_+ + \nu_- > \frac{N}{2}$. 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

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

where $\mu\left(\nu_\pm\right)$ 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

$$\begin{aligned} p_e &= 2\int Dx_+ Dx_- \,\Theta\left(\frac{q_+ N}{2} + \frac{\sqrt{q_+ N}}{2} x_+ + \frac{q_- N}{2} + \frac{\sqrt{q_- N}}{2} x_- - \frac{N}{2}\right) \cdot \\ &\qquad \cdot \Theta\left(-\frac{q_+ N}{2} - \frac{\sqrt{q_+ N}}{2} x_+ - \frac{q_- N}{2} + \frac{\sqrt{q_- N}}{2} x_- + \frac{N}{2}\right) \\ &= 1 - \frac{2}{\pi}\arctan\left(\sqrt{\frac{q_+}{q_-}}\right) \\ &= \frac{1}{\pi}\arccos\left(q\right) \end{aligned} \tag{34}$$

where we used the shorthand notation $Dx = dx \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$.

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:

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

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

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

The probabilities for positive-valued synapses instead are simpler:

$$P\left(\Delta<0\wedge\xi_i=\pm1|w_i=+1\right)=\frac{p_e}{2}$$

Thus, a positive-valued synapse has an equal probability of switching up or down one level, while a negative-valued one has a higher probability of switching up than down. The histogram dynamics can be written as:

$$\begin{aligned} P^{\tau+1}\left(h\right) &= P^\tau\left(h\right)\left[1-p_e^\tau\right]+P^\tau\left(h+2\right)\left[\frac{p_e^\tau}{2}-\frac{\Theta\left(-\left(h+2\right)\right)}{\sqrt{2\pi N}}\right]+ \\ &+ P^\tau\left(h-2\right)\left[\frac{p_e^\tau}{2}+\frac{\Theta\left(-\left(h-2\right)\right)}{\sqrt{2\pi N}}\right] \end{aligned} \tag{35}$$

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 one or two synapses (in fact, this does not happen with the CP algorithm, which does not appear to ever converge). This is justified by 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 in all cases convergence is eventually reached, which is intuitive.

2.2. Histogram dynamics for SBPI. In order to move from CP to SBPI, we have to compute probabilities for the new rule R2 to be applied, which happens when $0<\Delta\leq\theta_m$ with

probability p_s; thus:

$$\begin{aligned} p_b &= 2p_s \int d\mu(\nu_+)\, d\mu(\nu_-) \left(\frac{\nu_-}{q_- N}\right) \Theta\left(\nu_+ + \nu_- - \frac{N}{2}\right) \Theta\left(\nu_+ + (q_- N - \nu_-) - \frac{N}{2}\right) \cdot \\ &\qquad \cdot \Theta\left(-2(\nu_+ - \nu_-) + (q_+ - q_-)N + \theta_m\right) \\ &= 2p_s \int Dx_+ Dx_- \, \Theta\left(\sqrt{q_+}x_+ + \sqrt{q_-}x_-\right) \Theta\left(\sqrt{q_+}x_+ - \sqrt{q_-}x_-\right) \\ &\qquad \cdot \Theta\left(-\sqrt{q_+}x_+ + \sqrt{q_-}x_- + \frac{\theta_m}{\sqrt{N}}\right) \\ &= \frac{p_s \theta_m}{\sqrt{2\pi N}} \end{aligned} \tag{36}$$

Since this term is already of order $N^{-\frac{1}{2}}$, there's no need to distinguish between positive and negative synapses. 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 R2 each time it is applied.

We may 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

$$k = p_s \theta_m \tag{37}$$

Using eq. 36 we can add rule R2 to eq. 35, getting the full SBPI dynamics:

$$\begin{aligned} P^{\tau+1}(h) &= P^\tau(h)\left[1 - p_e^\tau - \frac{k/2}{\sqrt{2\pi N}}\right] + \\ &+ P^\tau(h+2)\left[\frac{p_e^\tau}{2} - \Theta(-(h+2))\frac{1}{\sqrt{2\pi N}} + \Theta(-(h+2))\frac{k/2}{\sqrt{2\pi N}}\right] + \\ &+ P^\tau(h-2)\left[\frac{p_e}{2} + \Theta(-(h-2))\frac{1}{\sqrt{2\pi N}} + \Theta(h-2)\frac{k/2}{\sqrt{2\pi N}}\right] \end{aligned} \tag{38}$$

The agreement between this formula and the simulations is almost perfect, up to when the average number of wrong synapses is very low, i.e. $q_- N$ is of order 1.

2.3. Continuous limit. Equation 38 can be converted to a continuous equation in the large N limit, by rescaling the variables:

$$t = \frac{\tau}{N} \tag{39}$$

$$x = \frac{h}{\sqrt{N}} \tag{40}$$

and using a probability density

$$p(x,t) = \sqrt{N} P^{Nt}\left(\sqrt{N}x\right) \tag{41}$$

Note that the $\sqrt{N}$ scaling of the hidden variables is the same that we found in the classification learning problem (Sec. 1.9).

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

$$\frac{\partial p}{\partial t}(x,t) = 2p_e(t)\frac{\partial^2 p}{\partial x^2}(x,t) - \frac{1}{\sqrt{2\pi}}\frac{\partial p}{\partial x}(x,t)\left[(4-k)\Theta(-x) + k\Theta(x)\right] + \tag{42}$$

$$+\delta(x)\Theta(-x)\gamma^-(t) + \delta(x)\Theta(x)\gamma^+(t) \tag{43}$$

$$p_e(t) = \frac{1}{\pi}\arccos(q(t)) \tag{44}$$

$$q(t) = 2\int_0^\infty dx\, p(x,t) - 1 \tag{45}$$

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

$$\int_{-\infty}^{+\infty} p(x,t) = 1 \tag{46}$$

$$p(0^-,t) = p(0^+,t) \tag{47}$$

The reason for the continuity requirement is that, if this would not be the case, the net probability flux through $x=0$ would diverge, as can be seen by direct inspection of eq. 38 and considering the scalings. Note that, in the 'standard' BPI case $k=2$, these two constraints simply amount at setting $\gamma^\pm(t) = 0$, as discussed in the next section.

As a whole, equation 42 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 the coefficient of diffusion depending on time. The constant drift is different between the left and right side of the x axis and depends on k, and this difference gives rise to an accumulation of probabilities on both sides of the point $x=0$ (expressed by the two Dirac deltas in the equation).

For negative x, equation 42 reads:

$$\frac{\partial p}{\partial t}(x,t) = 2p_e(t)\frac{\partial^2 p}{\partial x^2}(x,t) - \frac{4-k}{\sqrt{2\pi}}\frac{\partial p}{\partial x}(x,t) \tag{48}$$

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

$$\bar{x}(t) = x_0 + \frac{4-k}{\sqrt{2\pi}}(t-t_0) \tag{49}$$

$$v(t) = v_0 + 4\int_{t_0}^{t} dt' p_e(t') \tag{50}$$

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 analogue way a solution to the $x > 0$ branch of equation 42:

$$\frac{\partial p}{\partial t}(x,t) = 2p_e(t)\frac{\partial^2 p}{\partial x^2}(x,t) - \frac{k}{\sqrt{2\pi}}\frac{\partial p}{\partial x}(x,t) \tag{51}$$

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 velocity proportional to k rather than $4-k$.

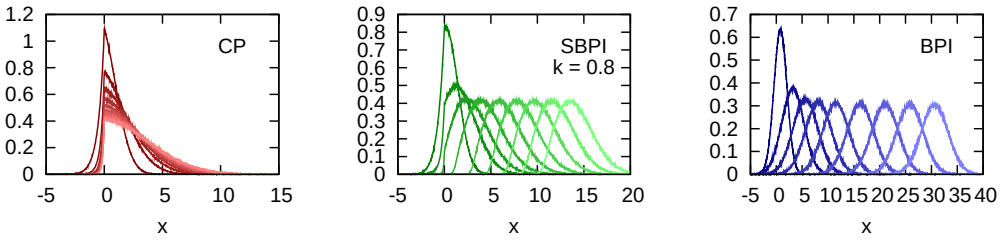

FIGURE 10. Evolution of the histograms with time (dark lines to light lines, taken in steps of $3N$, from $t = 1$ to $t = 25$), in simulations with three different algorithms (500 samples at $N - 32001$). In the first two panes, the positive and negative sides of the curve obey different differential 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 BPI algorithm. In all cases, the initial distribution was random, with all the synapses at $h = \pm 1$.

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). If the majority of the synapses eventually reaches the right side, the diffusion stops and the drift dominates. Furthermore, even if the learning process is slightly different, it is clear that a similar process is responsible for the shape of Fig. 5B. The evolution of the histograms at different times for different values of k is shown in Fig. 10.

Analytically, a solution to equation 42 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 eq. 48 and $p_0^+(x,t)$ as the time evolution of $p_0(x)$ under eq. 51 (these can normally be computed easily, e.g. by means of Fourier transforms). This allows us to write the solution in the form:

$$\begin{aligned} p(x,t) &= \Theta(-x)\left[p_0^-(x,t) + \int_0^t dt'\,\gamma^-(t')\,g^-(x,t,t')\right] + \\ &+ \Theta(x)\left[p_0^+(x,t) + \int_0^t dt'\,\gamma^+(t')\,g^+(x,t,t')\right] \end{aligned} \tag{52}$$

with the constraints given in eqs. 46 and 47. This solution can be verified by direct substitution in eq. 42; 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. 11A.

2.4. Density evolution for BPI. In the case $k = 2$, the two sides of equation 42 are equal; thus, the terms $\gamma^{\pm}(t)$ are both nought, and eq. 42 simplifies to:

$$\frac{\partial p}{\partial t}(x,t) = 2p_e(t)\frac{\partial^2 p}{\partial x^2}(x,t) - \sqrt{\frac{2}{\pi}}\frac{\partial p}{\partial x}(x,t) \tag{53}$$

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

$$p(x,t) = \frac{1}{\sqrt{v(t)}}G\left(\frac{x-\bar{x}(t)}{\sqrt{v(t)}}\right) \tag{54}$$

$$\bar{x}(t) = x_0 + \sqrt{\frac{2}{\pi}}t \tag{55}$$

$$v(t) = v_0 + 4\int_0^t dt' p_e(t') \tag{56}$$

$$p_e(t) = \frac{1}{\pi}\arccos(q(t)) \tag{57}$$

$$q(t) = \mathrm{erf}\left(\frac{\bar{x}(t)}{\sqrt{v(t)}}\right) \tag{58}$$

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 equal 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 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.

Fig. 11B 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. 11C.

3. Discussion

We have presented a simple on-line supervised algorithm, which leads to very fast learning of random input-output associations, up to close to the theoretical capacity, in a system with binary synapses and a finite number of hidden states. The performance of the algorithm depends crucially on a rule which leads to synaptic modifications only if the currently shown pattern is 'barely learned'. In this situation, the rule requires the synapse to have meta-plastic changes only. Only synapses that contributed to the correct output need to change their hidden variable, in the direction of stabilizing the synapse in its current state. This rule originates directly from the Belief Propagation algorithm. We have shown that this addition allows the BPI algorithm to learn a fraction of bits of information per synapse with at least roughly an order of magnitude less presentations per pattern than any other known learning protocol already at moderate system sizes and moderate values of α. We have also found that the same learning rule boosts

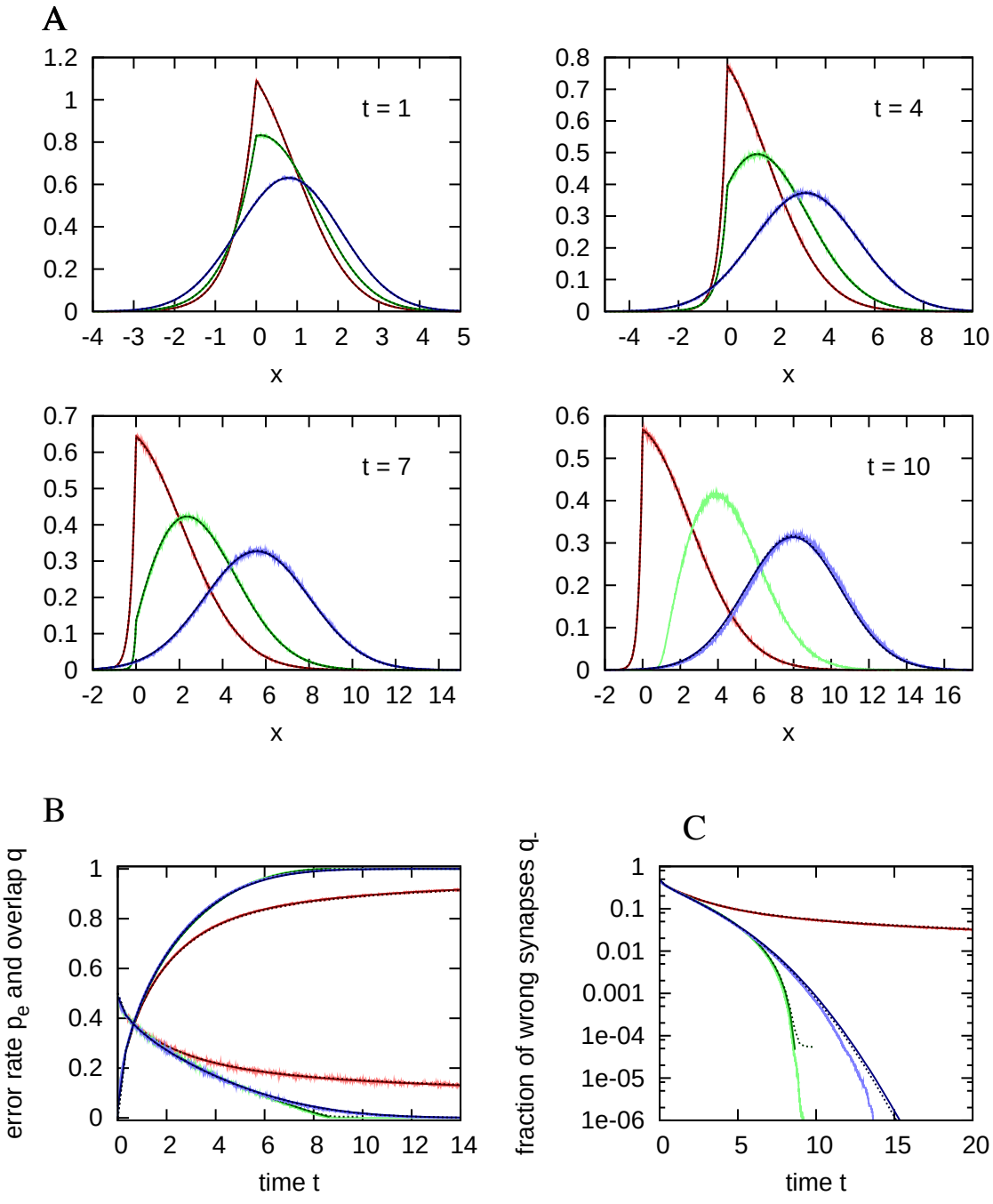

FIGURE 11. Comparison between simulations (light solid lines), historgam evolution (solid lines) and continuous probability density evolution (dark dotted lines), for three different algorithms (red: CP, green: SBPI with $k = 0.8$, red: BPI), at different times. The curves were taken at $N = 32001$, and initialized as for Fig. 10. The agreement between the simulations and the two analytical predictions is almost perfect, except when q_- is very small. **A.** Histograms at different times. 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 and error rate p_e vs time. **C.** Fraction of wrong synapses q_- vs time, in logarithmic scale. This can be used as an estimate of the convergence time with N; the last part of the BPI curve is well fitted by $\sqrt{\log(N)}$.

also the performance of the SP algorithm in multiple-visible-state devices, but that nevertheless a model with only two visible synaptic states and K hidden states is much more robust to noise than a model with K visible states. Finally, we have shown that in the slightly different scenario of generalization learning, where analytical predictions are possible, convergence of BPI can be proved in a time which is almost linear with the number of synapses, while CP does not seem to solve the problem at all.

Since the additional simple rule 2 has such a spectacular effect on performance, it is possible that neurological systems that learn in presence of supervision, though being much more complex devices than perceptrons, have found a way to implement such a rule; however, testing this experimentally is likely to be an awkward task with the current techniques, as it would require the ability to track the synapses plasticity and to detect meta-plastic modifications, and test whether they occur even in absence of an error signal, and if in such case the modification is in the direction of reducing the plasticity.

From the point of view of large-scale electronic implementations, using binary switches instead of continuous values is a big advantage in terms of simplicity, and would allow the use of currently available CMOS technology. The hidden variables are only needed during the learning period, and thus they could be stored separately if the learning and retrieval operational modes are distinct, but in any case they need not to be as reliable as they would if they were directly used in the output computation, and thus the overhead associated with their storage and management could be greatly reduced.

Finally, from a more general perspective, the research presented here demonstrates the possibility to successfully extend the application of message-passing algorithms to problems whose representative factor graph connectivity is very high, which is a rather common situation in computational biology. Moreover, these kind of algorithms are distributed in nature and are able to explore efficiently the global phase space by means of local computations only, and their study could be of great importance in order to understand the nature of the computations performed in biological networks.

Bibliography

[1] M. Mezard A. Braunstein, R. Zecchina. Survey propagation: an algorithm for satisfiability. *Random Structures and Algorithms*, 27:201–226, 2005.

[2] R.L. Rivest A.L. Blum. Training a 3-node network is np-complete. *Neural Networks*, 5:117–127, 1992.

[3] A. Amaldi. On the complexity of training perceptrons. O. Simula T. Kohonen, K. Makisara, J. Kangas, redaktorzy, *Artificial Neural Netwroks*, wolumen 1, strony 55–60. Elsevier Science Publisher, North-Holland, Amsterdam, 1991.

[4] D. J. Amit, S. Fusi. Constraints on learning in dynamics synapses. *Network*, 3:443–464, 1992.

[5] D. J. Amit, S. Fusi. Dynamic learning in neural networks with material sysnapses. *Neural Computation*, 6:957–982, 1994.

[6] U. S. Bhalla, R. Iyengar. Emergent properties of networks of biological signaling pathways. *Science*, 283:381–387, 1999.

[7] A. Braunstein, R. Mulet, A. Pagnani, M. Weigt, R. Zecchina. Polynomial iterative algorithms for coloring and analyzing random graphs. *Phys. Rev. E*, 68:036702, 2003.

[8] A. Braunstein, R. Zecchina. Learning by message-passing in networks of discrete synapses. *Phys. Rev. Lett.*, 96:030201, 2006.

[9] V. Cerny. A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–51, 1985.

[10] A. Engel, C. van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, 2001.

[11] S. Fusi, L. F. Abbott. Limits on the memory storage capacity of unbounded synapses. *Nature Neurosci.*, 10:485–493, 2007.

[12] S. Fusi, P. J. Drew, L. F. Abbott. Cascade models of synaptically stored memories. *Neuron*, 45(4):599–611, Feb 2005.

[13] E. J. Gardner. The phase space of interactions in neural network models. *J. Phys. A: Math. Gen.*, 21:257–270, 1988.

[14] E. J. Gardner, B. Derrida. *J. Phys. A: Math. Gen.*, 21:271, 1988.

[15] H. Gutfreund, Y. Stein. Capacity of neural networks with discrete synaptic couplings. *J. Phys. A: Math. Gen.*, 23:2613–2630, 1990.

[16] Y. Weiss J. S. Yedidia, W. T. Freeman. *Understanding Belief Propagation and its generalizations*, rozdzia/l 8, strony 236–239. Morgan Kaufman, 2003.

[17] W. Krauth, M. Mézard. Storage capacity of memory networks with binary couplings. *J. Phys. France*, 50:3057, 1989.

[18] J. E. Lisman. A mechanism for memory storage insensitive to molecular turnover: a bistable autophosphorylating kinase. *P. N. A. S. USA*, 82:3055–3057, 1985.

[19] G. Parisi M. Mezard, R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297:812, 2002.

[20] M. Mezard, , R. Zecchina. Random k-satisfiability: from an analytic solution to a new efficient algorithm. *Phys. Rev. E*, 66:056126, 2002.

[21] M. Mézard, G. Parisi, M. A. Virasoro. *Spin Glass Theory and beyond*. World Scientific: Singapore, 1987.

[22] M.A. Wilson M.R. Mehta, A.K. Lee. Role of experience and oscillations in transforming a rate code into a temporal code. *Nature*, 417:741–746, 2005.

[23] D. H. O'Connor, G. M. Wittenberg, S. S-H. Wang. Graded bidirectional synaptic plasticity is composed of switch-like unitary events. *Proc Natl Acad Sci U S A*, 102:9679–9684, 2005.

[24] J. E. Lisman P. Miller, A. M. Zhabotinsky, X-J. Wang. The stability of a stochastic camkii switch: dependence on the number of enzyme molecules and protein turnover. *PLoS Biol*, 3(4):e107, Mar 2005.

[25] G. Parisi. Spin glasses and fragile glasses: statics, dynamics, and complexity. *Proc. Natl. Acad. Sci. USA*, 103:7945–7947, 2006.

[26] C. C. Petersen, R. C. Malenka, R. A. Nicoll, J. J. Hopfield. All-or-none potentiation at CA3-CA1 synapses. *Proc.Natl.Acad.Sci.USA*, 95:4732–4737, 1998.

[27] C. M. Powell. Gene targeting of presynaptic proteins insynaptic plasticity and memory. *Neurobiol. learn. mem.*, 85(1):2–15, 2006.

[28] F. Rosenblatt. *Principles of neurodynamics*. Spartan Books, New York, 1962.

[29] C. D. Gelatt S. Kirkpatrick, M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[30] M. Youssoufian S. Oleskevich, B. Walmsley. Presynaptic plasticity at two giant auditory synapses in normal and deaf mice. *Journal of Physiology*, 560:709–719, 2004.

[31] P. Sterling, J.B. Demb. *Retina*, rozdzia/l 6, strony 217–269. Oxford University Press, 2004.

[32] M. Tsodyks. Associative memory in neural networks with binary synapses. *Mod. Phys. Lett. B*, 4:713–, 1990.

[33] D. Willshaw, O. P. Buneman, H. Longuet-Higgins. Non-holographic associative memory. *Nature*, 222:960–962, 1969.

[34] S. Uda Y. Kabashima. *A BP-based algorithm for performing bayesian inference in large perceptron-type networks*, wolumen 3244, strony 479–493. Springer Berlin / Heidelberg, 2004.

[35] A. M. Zhabotinsky. Bistability in the Ca^{2+}/calmodulin-dependent protein kinase-phosphatase system. *Biophys. J.*, 79:2211–2221, 2000.