ScuDo

Scuola di Dottorato ∽ Doctoral School

WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Physics (30^{th}cycle)

Out of Equilibrium Statistical Physics of Learning

By

Luca Saglietti

******

Supervisor(s):
Prof. Riccardo Zecchina, Supervisor

Doctoral Examination Committee:
Prof. H.J. Kappen, Referee, Radboud University
Prof. F. Ricci-Tersenghi, Referee, "La Sapienza" University
Prof. M. Caselle, University of Turin
Prof. A. Montorsi, Polytechnic University of Turin
Prof. V. Penna, Polytechnic University of Turin

Politecnico di Torino
2018

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Luca Saglietti
2018

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Alla mia famiglia

Acknowledgements

I wish to express my most sincere gratitude and appreciation to Riccardo Zecchina, for inspiring me with his passion and constant scientific optimism, for always sharing his great research vision, for teaching me the value of exchanging ideas, and for offering me unique opportunities. Of course, I also wish to thank Carlo Baldassi, always willing to patiently listen to and answer even the most trivial doubts and questions I produce, always open for interesting scientific discussions, whose pure programming talent is uplifting (and somewhat unfair), and whose guidance has been crucial for my work since day one.

I want to thank Carlo Lucibello, for being the messy physicist he is, for bringing so many fresh ideas to the table, for being always available for cross-checking silly research proposals and for teaching me some good tricks of the trade. I also thank Alessandro Ingrosso, whom I shared my first years of everyday work with, and whose distinctive sense of humor both exhilarated me and drove me nuts (at times).

I want to thank Federica Gerace, my PhD companion, without whom I would have certainly missed some vital bureaucratic detail, thus failing to complete my PhD, and who's been always able to condone my foolish mockeries with a laugh. Many thanks to Enzo, for his crazy humour and his enthusiasm, and also to Chiara, Carla, Marco and Thomas, for tolerating our loud discussions and for being always ready for a coffee together. Thanks also to Andrea P, Andrea D, Anna, Marco, Alfredo, Luca and the rest of the group.

My deepest gratitude goes to my parents, to my brother and to my sister, whose confidence in my prospects and constant support have taken me a long way.

Finally, I want to thank my love, for being there for me everyday, even when this strange job keeps my head in the clouds.

Abstract

In the study of hard optimization problems, it is often unfeasible to achieve a full analytic control on the dynamics of the algorithmic processes that find solutions efficiently. In many cases, a static approach is able to provide considerable insight into the dynamical properties of these algorithms: in fact, the geometrical structures found in the energetic landscape can strongly affect the stationary states and the optimal configurations reached by the solvers. In this context, a classical Statistical Mechanics approach, relying on the assumption of the asymptotic realization of a Boltzmann Gibbs equilibrium, can yield misleading predictions when the studied algorithms comprise some stochastic components that effectively drive these processes out of equilibrium. Thus, it becomes necessary to develop some intuition on the relevant features of the studied phenomena and to build an *ad hoc* Large Deviation analysis, providing a more targeted and richer description of the geometrical properties of the landscape. The present thesis focuses on the study of learning processes in Artificial Neural Networks, with the aim of introducing an out of equilibrium statistical physics framework, based on the introduction of a *local entropy* potential, for supporting and inspiring algorithmic improvements in the field of Deep Learning, and for developing models of neural computation that can carry both biological and engineering interest.

Contents

Chapter 1

Theoretical framework

1.1 General introduction

The ability to learn and store information in our brain is the fundamental seed of what we call intelligence. It is generally believed that these mechanisms take place in real neural systems via plastic changes to the connections between neurons, the synapses, in response to external stimuli, either by creating or destructing the connections or by modifying their efficacy [9]. This idea is the basic inspiration of the Deep Learning research field, which in recent years has proved able to produce algorithms that achieve top performance in a host of complex applications, such as image or speech recognition, and is slowly closing the gap that still separates human and *in silico* computation [10, 11], [12]. However, these practical successes have been guided by intuition and numerical experiments, while obtaining a complete theoretical understanding of why these techniques work seems currently out of reach, due to the inherent complexity of the problem.

While the mainstream approach is based on the application of heuristic variants of the stochastic gradient descent algorithm, in deep continuous neural networks, it is rather unlikely that biological brains employ the same gradient-based learning strategy: real synapses are generally very noisy, and the estimated precision with which they can store information, although very difficult to assess conclusively, seems to range between 1 and 5 bits per synapse [13, 14]. Real synaptic efficacies might be better described by discrete quantities rather then continuous ones, and even machine learning applications (especially hardware implementations) could benefit from the implementation of simpler synaptic models and update protocols.

In the past decades, various methods borrowed from Statistical Physics of Disordered Systems, have been quite successful in studying the basic properties of neural-like systems [15]. These analyses predict a qualitative difference between neural networks when the synaptic weights are either continuous or discrete variables. Even in the simplest discrete neural network — the Perceptron with N binary synapses — the learning problem is known to be intractable in the worst-case [16], and its equilibrium description is dominated in the typical case by an exponential number of local minima [17–20], which easily trap standard search strategies based on free energy minimization, e.g. Monte Carlo [21, 22] (a familiar situation in spin glass phases, [23–25]); moreover, the optimal synaptic configurations are typically geometrically isolated (i.e. they have extensive mutual Hamming distances), and thus even harder to find for local search strategies [22].

However, the additional computational difficulties associated with discrete synapses are not insurmountable: a number of heuristic algorithms are known to achieve very good performances even in the extreme case of binary synapses [26–29]; some of those are even sufficiently simple and robust to be conceivably implementable in real neurons [27, 28]. In general, these algorithms are by no means bound to sample solutions uniformly at random and the underlying stochastic dynamics is not guaranteed to reach states described by an equilibrium probability measure, as would occur for ergodic statistical physics systems. An out of equilibrium description is indeed needed in order to capture structures which are relevant for these learning processes.

In most random combinatorial optimization problems the dominant *ground states* of the equilibrium Gibbs measure at zero temperature, are not relevant in the analysis of practical optimization algorithms and their dynamics. The algorithmically accessible states, in fact, are typically sub-dominant states characterized by a high internal entropy [30] (see also next paragraph). The structure of such sub-dominant states can be investigated by means of the Replica or the Cavity Methods [24], at least in the average case. As we will show throughout this thesis, this scenario also holds in the case of learning algorithms for Artificial Neural Networks, where much work is still needed in order to get a better understanding of what determines the success of the various heuristics.

1.2 Algorithms and Out-of-Equilibrium

In the past few decades, a very prolific knowledge exchange greatly boosted the development of a research field at the interface between Statistical Physics, Discrete Mathematics and Computer Science: Combinatorial Optimization [23]. The scope of this subject is impressive, since the problem of identifying the best choice over a set of possible alternatives is ubiquitous in a plethora of scientific fields, ranging from Biology to Economics. A generic optimization problem is defined by a given parametric space $\mathcal{S}$ and a loss-function $f : \mathcal{S} \to \mathbb{R}$, which is the objective of the optimization process: the goal of an effective algorithm is that of efficiently solving the problem, i.e. finding a configuration, $s^\star \in \mathcal{S}$, that minimizes the loss $s^\star = \mathrm{Argmin}_s f(s)$ in the shortest possible computational time.

In a combinatorial optimization problem, the feasible solution space $\mathcal{S}$ is finite: this allows a link with a Statistical Physics formalism, easily established by associating a partition function to the problem. In this context, the loss-function, properly rescaled by the number of variables of the problem N, plays the role of an energy: by tuning the inverse temperature, β, one can focus the measure on the solutions of the problem, eventually recovered in the limit $\beta \to \infty$. From a physicist perspective, it is interesting to study the thermodynamic limit, $N \to \infty$, in search of interesting *phase transitions* that could be relevant for understanding the behavior of the designed optimization algorithms, even in finite size instances.

An ideal framework for studying the onset of these complex collective phenomena is that of random Constraint Satisfaction Problems (CSPs). An instance of a CSP is defined through an extensive number, $M \propto N$, of constraints that need to be satisfied by a set of N variables: the hardness of the optimization task can be increased by adding more constraints to the problem. As the system becomes more and more frustrated, the zero-energy configurations in $\mathcal{S}$, satisfying all the constraints, get progressively decimated.

A first question that can be answered with the tools of Statistical Physics, is whether any solutions to the problem can be found at a given constraint density, $\alpha = M/N$, in the thermodynamic limit. Interestingly, in random CSPs, the system undergoes a sharp transition at a critical density α_C, going from a phase where exponentially many zero-energy configurations are present with very high probability, the so-called SAT phase, to a phase where the problem is no longer satisfiable and at best a small fraction of the constraints will be necessarily violated, the UNSAT phase.

However, setting the problem in the SAT phase is not a sufficient condition for guaranteeing the possibility of finding solutions algorithmically, as the most interesting CSPs are NP-complete problems: this means that the number of elementary computational operations needed to find a solution is expected to grow exponentially with N in the worst case, since one would be required to perform an extensive check over all the possible assignments for the variables. The efficient solvers are instead those that are able to provide solutions in less than exponential time. Very often, the intuition behind the design of these optimal algorithms exploits some knowledge on the geometrical organization of the solutions in $\mathcal{S}$. There is, in fact, a clear connection between the dynamical properties of the employed algorithms and the static properties of the energy landscape: the computational hardness is often associated with the coexistence of low-energy configurations and sub-optimal *metastable states*, that break ergodicity and "hide" the solutions of the problem, often grouped into various clusters of nearby configurations [31, 32].

The main analytical tools that allow a theoretical analysis of random CSPs, the Replica and the Cavity methods, are directly inherited from Disordered Systems, a quite modern but well established branch of Statistical Physics [24]. These methods were initially developed in the study of the thermodynamic properties of the so-called Spin glasses, i.e. spin models defined by an Hamiltonian that is dependent on some kind of randomness, usually enclosed in the couplings, fields or topology of the model [33, 34]: in order to average over the possible realizations of the Hamiltonian, one can exploit the so-called Replica trick (see chapter 2), a mathematical identity for evaluating the *quenched average* of the free energy. This average can in fact be extrapolated from the behavior of a replicated model, where the copies of the system are virtually interacting through the common realization of the disorder. The order parameters of these models, signaling the onset of the *spin glass* phase, explicitly describe a Gibbs measure composed of exponentially many disjoint groups of solutions, generating an ultrametric structure [35]. The transfer of knowledge to the context of Combinatorial Optimization was pioneered by Parisi and Mézard in 1985 [36], giving way to a series of important theoretical findings as well as substantial algorithmic developments [31].

In the Replica (or in the Cavity) method, in order to characterize the typical properties of large instances of a given CSP, it is necessary to make some assumptions on the relevant symmetries that characterize the phase space of problem. These geometrical properties are strongly related to the degree of correlation between the variables of the model, induced by the constraints of the problem: high correlations can lead to the *frustration* phenomenon, as any local change in the assignment of a single variable can require extensive rearrangements in the neighboring variables, in order to maintain the energy at the minimum. As the density α is increased, the system can thus undergo a series of *structural transitions* [31, 32], where the clusters of solutions progressively break apart into smaller clusters, characterized by higher internal correlations between the variables (see a sketch in figure 1.1). In the Replica Symmetric phase, where the Gibbs measure can be seen as a unique pure state, even relatively basic algorithms based on energy relaxation, as Monte Carlo Simulated Annealing [37], can usually find solutions of the CSP. Instead, after the occurrence of the Replica Symmetry Breaking phenomenon above a certain critical density α_D, the Gibbs measure scatters into a convex linear combination of pure states, and only special classes of algorithms are able to avoid being trapped in the metastable states that jam the solution space.

In the Replica Symmetry Breaking scheme proposed by Parisi [35], the number of disjoint pure states specified by a given internal entropy of solutions S is assumed to be exponential, $\mathcal{N}(S) \sim \exp(N\Sigma(S))$. The function Σ plays the role of an entropy of clusters, usually called *complexity*. In the thermodynamic limit one can evaluate the typical entropy $S^\star$ by means of a saddle point approximation, and obtain some information on the states that dominate the Gibbs measure. If the efficient algorithms sampled uniformly at random from the solutions of the problem, the probability of finding a solution belonging to the states characterized by the lowest free energy would tend to 1 for large instances. In this case, the equilibrium properties of the problem

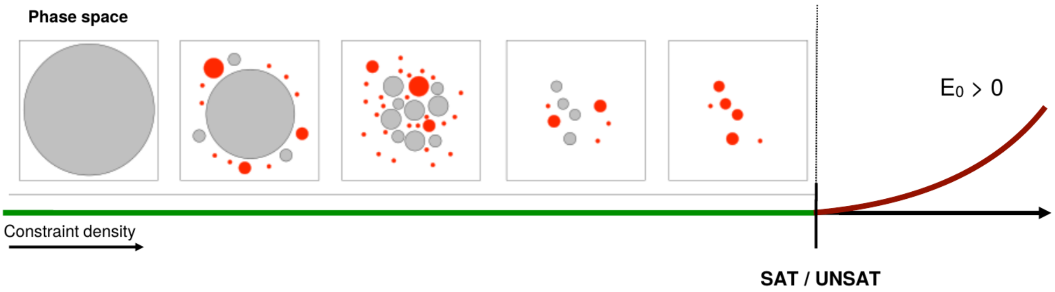


Fig. 1.1 **Sketch of the structural phase transitions** in the solution space of a CSP. The ensemble of solutions disconnects progressively as the constraint density is increased, going from a single large cluster to an exponential number of smaller clusters. The red colored clusters are those where a fraction of the variables is frozen. After the SAT/UNSAT transition, zero energy configurations can no longer be found. Adapted from [32].

would provide sufficient information on all the dynamical properties of the algorithms. However, the algorithms are often attracted towards sub-dominant states, characterized by specific geometrical properties [30].

After the dynamical transition at α_D, in fact, some of the dominant clusters can become partially *frozen* (the so-called *rigidity* transition [38, 30]), i.e. contain only solutions where a fraction of the variables forcedly take a precise value. This backbone structure can be very hard to "guess" algorithmically, as the search would require collective rearrangements of the variable assignments and the frozen variables would need to be correctly assigned altogether: this high degree of correlation could thus induce an exponential slowing down of the algorithms. Therefore, in this regime, the efficient solvers become attracted to the still existing *unfrozen* states, even though these are not numerically favored. After the final *freezing* transition, all the clusters become frozen and the algorithms stop working, even though solutions still exist until the SAT/UNSAT transition at α_C [38, 30]. It is important to underline that the dynamical behavior of the algorithms can thus be reflected only by a static analysis where also the large deviations of the problem are considered, in order to describe the dominant as well as the sub-dominant structures in the solution space.

The main objective of this PhD thesis is that of trying to transpose the same approach, initially employed for studying various CSPs, in the framework of discrete Artificial Neural Networks, where there is still a clear gap between the theoretical analysis, limited to the equilibrium properties of simplified models, and the abundance of algorithmic results, often relying on clever heuristic modifications that push the learning process out of equilibrium in order to achieve a better performance.

1.3 Outline of the thesis

The first chapter of this thesis, *Theoretical framework*, contains a brief introduction to the scientific background and to the aims and scope of our research work; furthermore, it provides an overview of the organization of the results, with a short summary of each chapter of the thesis. The rest of the material presented in this thesis will be organized mainly in two main parts, each one subdivided in chapters.

The first part, *Neural Networks: Equilibrium vs Algorithms*, will feature the following content:

- The second chapter, *Equilibrium Analysis*, will provide a brief general introduction to Artificial Neural Networks (ANN), specifically in the setting of discrete synaptic models. The chapter is mainly devoted to the theoretical analysis of the simplest feed-forward ANN model, the *Perceptron*, and to the description of two common learning scenarios, the *Random classification* and the *Teacher-student* problems. In particular, all the results will be obtained in the *Binary Perceptron* model first, to be then extended to the *Generalized Discrete Perceptron* case. The main analytic tool employed in the equilibrium analysis, the *Replica Method*, will be introduced operationally: we will first retrace the classical results of the Gardner analysis, both in the Replica Symmetric and in the Replica-Symmetry-broken frameworks. The analysis will then be extended to the Generalized Discrete Perceptron case, where some original results will be reported. After the standard Replica analysis, we will also report on some modern theoretical results on the fine geometrical structure of the solution space of the Binary Perceptron, obtained through the study of the so-called *Franz-Parisi potential*, and we will extend them to the general case of discrete synapses. Finally, we will here provide also a generalization of the Franz-Parisi potential computation to the Teacher-student scenario, in the binary case. The results in paragraph 2.6.2 are original and unpublished.

- The third chapter, *Learning with Discrete Synapses*, will be dedicated to the algorithmic side of the problem of learning in ANNs. After an introduction to *Belief Propagation*, a physics-inspired message-passing technique, we will report on some recent algorithmic advancements, regarding the proposal of efficient solvers for the Binary Perceptron learning problem. We will briefly describe some attempts at extending these heuristic learning algorithms to the more relevant (from a machine learning perspective) multi-layer case. In order to validate the geometrical landscape predictions obtained in the second chapter, we will also describe the results of some numerical tests, probing the neighborhood of the solutions found by the solvers. The numerical results will be compared with the theoretical predictions, highlighting the existence of a gap between the equilibrium analysis and the numerical findings, and suggesting the development of an *ad hoc* theoretical framework for better understanding these effective learning processes. The results described in sections 3.4-3.7 are novel and unpublished.

The second part, *Large Deviation Analysis*, will be organized as follows:

- The fourth chapter, *Novel Measure*, will describe the main theoretical results of this thesis. We will introduce a Large Deviation Analysis able to discover the presence of a complex sub-dominant structure in the solution space of the Binary (and Generalized Discrete) Perceptron. The main conceptual step will be that of defining a *local entropy potential*, counting the number of solutions in close proximity of a given configuration, and enhancing the statistical weight of dense clusters of solutions. In the theoretical analysis, again based on the Replica Trick, we will consider different scenarios and *Ansatz*s, uncovering a possible structural phase transition above which the dense structure disappears, in exact correspondence with the numerically recorded algorithmic thresholds.
- In the fifth chapter, *Entropy driven Monte Carlo*, we will devise a simple and theoretically under control algorithm, able to explicitly target the

dense regions in the space of solutions of Discrete Perceptrons. The framework will be that of a Markov Chain Monte Carlo strategy, where, instead of the usual energy (or loss function), we elect the local entropy as the objective to be maximized. The estimation of the local entropy will rely on the Belief Propagation algorithm, and the resulting learning process will be defined as a two level optimization procedure.

- The sixth chapter, *Robust Ensembles*, will provide a description of a general optimization scheme able to target the local entropy measure introduced in chapter four: the inverse temperature, associated to the entropy reweighting, can be set to positive integer values and treated as a replica index. This suggests the definition of a new statistical ensemble, that naturally favors dense regions of solution, and allows to bypass the employment of a second level algorithm for the local entropy estimation (as in EdMC, in chapter five). The suggested scheme is then applied in the context of the most common optimization algorithms: Simulated Annealing, Gradient descent and Belief Propagation.

- In the seventh chapter, *Stochastic Synapses*, we will introduce a stochastic formulation of ANNs that naturally gives prominence to dense regions of solutions in the loss landscape. We show that, in this context, binary solutions can be obtained through a simple gradient descent procedure on a set of real values, that parametrize a probability distributions over the binary synapses. All these properties will be confirmed through a theoretical analysis in the Binary Perceptron model and supported by numerical results. We will also show some preliminary result on the extension of this framework to deeper neural networks models.

Finally, the last part, *Conclusions*, will be devoted to the discussion of the results presented throughout the thesis and to the proposal of some directions for future investigation.

Part I

Neural Networks: Equilibrium *vs* Algorithms

Chapter 2

Equilibrium Analysis

In the past few decades Artificial Neural Networks (ANN) have become one of the most flexible and successful tools for machine learning applications in a variety of complex recognition tasks: from computer vision [12] and speech recognition to medical diagnostics and biological and physical data analysis. Some outstanding achievements, especially in the context of video and board-game playing AI, seem to question the very definitions of human-like intelligence, intuition and creativity [39, 40].

The computational power of these devices comes from the huge number of parameters that, without any task specific or rule-based programming, can be progressively fine-tuned to improve performance, in a example-based learning procedure. Of course, large-scale ANNs (the so-called Deep Neural Networks, DNNs in the following) have two main requirements, that initially slowed down the escalation to their present success: big and rich training datasets and adequate hardware support. Remarkably, despite the fact that the non-linear optimization needed for training DNNs takes place in a potentially complex, extremely roughed landscape, learning often occurs without getting trapped in local minima with poor prediction performance and over-fitting behaviors are surprisingly sporadic [41].

Unfortunately, the knowledge extracted by the enormous amount of research in Deep Learning is often empirical, resulting from a trial and error process mainly guided by intuition and numerical evidence, while the theoretical understanding of many of the employed heuristic techniques is incomplete. In recent years, there have been many parallels between the studies of algorithmic stochastic processes and out-of-equilibrium processes in complex systems [42–46], as a Statistical Physics approach has proven successful also in computer science applications, for understanding the behavior of local search algorithms for optimization and inference. The theoretical interest of these processes origins from the fact that the underlying dynamics are not guaranteed to reach states described by an equilibrium probability measure, as would happen in an ergodic system in classical Statistical Mechanics. Indeed, sets of relevant configurations that are typically inaccessible for an equilibrium process can become extremely attractive for the analyzed algorithms.

The statistical physics approach to the study of neural-like systems is complementary to the one of neurophysiology [15], being based on the interpretation of intelligence and learning as collective emerging properties of large ensembles of neurons, rather than being dependent on the overwhelming variety of microscopical differences, between each single neuron, that can be observed in biological brains. This assumption justifies the choice of extremely simplified models for the constituents of these large-scale artificial systems: the McCulloch–Pitts neuron [47], more than half a century old, is still the main building block in modern DNNs. In this basic model the neuron is represented as a bi-stable linear threshold unit, either in a excited or a quiescent state: it simply sums up the incoming stimuli, weighted by the corresponding synaptic coupling strengths, makes a comparison with a threshold value and outputs a binary signal representing its state (modeling, respectively, the input from the *dendrites*, the accumulation in the *soma* and the firing in the *axon* in a biological neuron).

A feed-forward neural network can then be obtained by connecting a set of McCulloch–Pitts neurons in a layered architecture, from the inputs to a series of intermediate "hidden" neurons, to the final output layer [15]. This device can be easily trained in a supervised learning procedure: a series of input vectors are sequentially presented in the first layer of the network, and the error signal is obtained from the comparison of the resulting output with the correct label associated to the pattern in input. This supervised signal can then be distributed in the neural network, inducing local modifications in the synaptic couplings, such that the correct output is eventually memorized in the network's parameters. In the statistical physics formulation one considers two typical supervised scenarios: the “classification” problem, where the task is that of perfectly performing a given set of input-output associations, and the more realistic “generalization” scenario, where the goal is that of inferring a correct association rule, which could be represented either explicitly, by a “teacher” device (in the so-called “teacher-student” scenario), or implicitly, through an additional set of examples which are only presented in the test phase [15]. Both these scenarios will be considered in the following.

The theoretical approach that is adopted throughout this thesis can be seen as an atomistic approach to Deep Learning: we will start by considering the simplest neural network models, where we are able to gather some analytical results, and then we will try to extrapolate to the more complex phenomena emerging from learning in large-scale neural networks, in the assumption that the qualitative behaviors of the building blocks should somehow be inherited by the more complex DNN architectures.

2.1 The Perceptron

Introduced by Rosenblatt in 1962 as a simplified model of a neuron [48], the Perceptron is the simplest example of a feed-forward neural network. The network has no hidden layers and is parametrized by a single vector of synaptic couplings $\{W_i\}_{i=1}^N$: given an input pattern ξ, the output is simply obtained as:

$$\tau\left(\xi; W\right) = \mathrm{sign}\left(W \cdot \xi - \theta\right) \tag{2.1}$$

(here we will set the firing threshold to $\theta = 0$, for simplicity); $\tau = 1$ represents the excited state of the neuron and $\tau = -1$ the quiescent one. Despite its simplicity, the Perceptron exhibits a variety of desirable computational features: the inputs are processed in parallel, the device is able to memorize an extensive amount of information and the learning procedure can be devised as a simple on-line process; however, it also suffers from strong computational limitations, being able to solve only linearly separable classification tasks.

From the statistical physics perspective, the Perceptron has become the “hydrogen atom” of the field. By using some techniques borrowed from the physics of disordered systems, one can study the constraint satisfaction problem

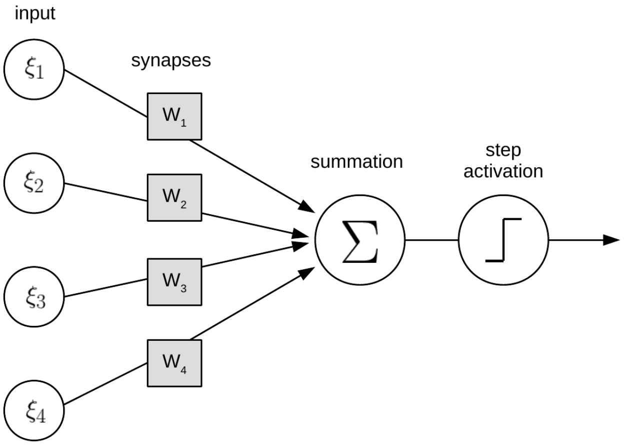


Fig. 2.1 **The perceptron**: a schematic representation.

of correctly classifying an extensive set of random patterns $\{\{\xi_i^\mu\}_{i=1}^N, \sigma^\mu\}_{\mu=1}^M$, with $M = \alpha N$ and α being the so-called storage load [49]. Each constraint can be simply represented by a Θ-function, nullifying the statistical weight of synaptic configurations that entail classification errors. Statistical mechanics mainly aims at producing exact results for the *typical* learning behaviour, which can be extracted by considering the thermodynamic limit of the model, $N \to \infty$: in this limit both the number of degrees of freedom of the neural network and the number of constraints in the training problem diverge, and one can study different learning regimes by tuning the $\mathcal{O}(1)$ parameter α.

In order to carry out the analytical computations one has to make very strong uncorrelation assumptions on the distribution of the patterns: usually one considers each component ξ_i^μ, to be i.i.d. uniformly drawn from $\{-1, 1\}$, while the binary outputs can either be uncorrelated themselves, as in the *random* or *classification* scenario, or can be determined by a teacher device W^T, with $\sigma^\mu = \text{sign}\left(W^T \cdot \xi^\mu\right)$, as in the *teacher-student* or *generalization* scenario. Once the training set is determined, one can study the probability distribution induced on the synaptic couplings $P_W(W)$ and the way it is affected by the addition of new constraints to the learning problem. The idea of considering a phase space of interactions, overturning the usual paradigm of Disordered Systems where the randomness is enclosed in the couplings of the model, was initially established by Elisabeth Gardner [50].

In the thermodynamic limit one is interested in measuring *self-averaging* quantities, whose probability distribution gets focused on a typical value as the variance vanishes with $N \to \infty$. In this case one can simply compute the average over the disorder, induced by the training patterns, and find the most probable value for the chosen quantity. Unfortunately not all physically meaningful quantities are automatically self-averaging: in the Perceptron, one would be interested in counting the number of solutions for a given instance of the learning problem, and possibly determine the critical value for α after which, with high probability, one cannot find any more solutions (or, in the case of the teacher–student problem, only the teacher is left as a viable solution). However, the *annealed* entropy density, obtained by taking the normalized logarithm of the average volume:

$$\left\langle \Omega\left(\{\xi^\mu,\sigma^\mu\}_{\mu=1}^{\alpha N}\right)\right\rangle_{\xi,\sigma} = \left\langle \int \mathrm{d}\mu\left(W\right) \prod_{\mu=1}^{\alpha N} \Theta\left(\sigma^\mu \frac{W\cdot\xi^\mu}{\sqrt{N}}\right)\right\rangle_{\xi,\sigma} \tag{2.2}$$

returns an average value which is very different from the sought most probable value of the entropy: this common problem is due to the long tails of the probability distribution, which causes the average to drift away from the mode, attracted by the *rare events* of the model. Because of the concavity the logarithm, the annealed approximation can however provide an upper bound to the correct value we are seeking to compute: intuitively it describes a best-case scenario in which both the degrees of freedom and the constraints are adapted to minimize the free energy of the system [15].

A correct theory of learning must instead consider the *quenched* entropy density:

$$\mathrm{S} = \left\langle \log\left(\Omega\left(\{\xi^\mu,\sigma^\mu\}_{\mu=1}^{\alpha N}\right)\right)\right\rangle_{\xi,\sigma} \tag{2.3}$$

which is technically much less straightforward to obtain. The main problem is that one would like to exploit the “translational invariance” of the problem and treat each component W_i homogeneously, factorizing over the index $i = 1, ..., N$, but the quenched average seems to require an integration over the synaptic weight measure $\mathrm{d}\mu\left(W\right)$ at each fixed realization of the disorder. Fortunately, there is a way of interchanging the order between the average and the logarithm, well-known since the 1970s, by exploiting the simple identity:

$$\log X = \lim_{n \to 0^+} \frac{X^n - 1}{n} \tag{2.4}$$

This gives rise to the *replica trick*: one can first consider an enlarged system with $n \in \mathbb{N}$ independent and identical replicas of the system, then take the disorder average, which introduces a coupling between the n copies, and in the end look for an analytical continuation of the result to continuous $n \in \mathbb{R}$ (in the thermodynamic limit), for retrieving the initial expression in the limit $n \to 0^+$. This formalism is very problematic from the mathematical point of view: for example, the extrapolation at $n = 0$ is not in correspondence of an accumulation point, and in the calculations the two limits $n \to 0^+$ and $N \to \infty$ are interchanged. However, the application of this method for studying artificial neural networks, first proposed by Elizabeth Gardner [49], has proven very successful.

During the Replica computation, since the minimization of the action in the full parametric space is unfeasible, the saddle point evaluation is always performed in a restricted subspace, defined through some kind of Ansatz on the geometrical structure of the described phase space, based on possible symmetries of the replicated model: the simplest possible choice is the Replica Symmetric (RS) Ansatz, where one assumes that the Gibbs space cannot be decomposed into a mixture of pure states, and is instead well described as a unique state, where the various replicas are completely symmetric (under permutation). In the continuous Perceptron, where the couplings are usually restricted to lay on the N-sphere, $\mathrm{d}\mu\left(W\right) = \left(\prod_i \mathrm{d}W_i\right) \delta\left(\sum_i W_i^2 - N\right)$, the Replica symmetric assumption yields correct results; this is closely related to the fact that the phase space of the problem happens to be convex and connected (which is physically similar to saying that the system dynamics is ergodic). The order parameter of the model is the overlap between two replicas, $q^{ab} = \sum_i \frac{W_i^a W_i^b}{N}$, with $a, b \in \{1, ..., n\}$: as more constraints are added the overlap $q \to 1$, and the volume of solutions shrinks, up to a single point at the critical storage capacity $\alpha_c = 2$ (the same result was also independently obtained by Cover [51]).

Also from the algorithmic point of view, the continuous Perceptron is a simple problem, as we will see in chapter 3: many learning rules are able to saturate the critical capacity of the model, from the classical Hebb and the Perceptron Rule, to the Pseudo-Inverse and Adaline Rules [15].

2.1.1 The Ising Perceptron

Throughout this thesis, we will mostly focus on the problem of learning in discrete artificial neural networks, i.e. networks in which the synapses are chosen from a discrete set of possible values. The simplest case, which can be regarded as the Ising model of ANN, is that of the binary Perceptron, where we restrict the synaptic couplings to the corners of the N-hyper-cube , $W \in \{-1, 1\}^N$. We will also consider the extension to synapses with more states, for example $W_i \in \{0, 1, ..., L\}$, in an analogous of the Potts model of Statistical Mechanics. The apparent simplification of the model, due to the restriction to "fewer" possible states, surprisingly turns out to produce a variety of complications, both on the analytical and on the algorithmic sides of the problem.

What motivates the choice of these discrete models is a series of biological, engineering and theoretical motivations. From the biological experiments, we know that the elementary computational step for learning is the modulation of synaptic efficacy: in the last years, some neuroscientific results have hinted at the fact that synaptic potentiation or depression might be induced through switch-like unitary events, allowing the shift between a restricted number of discrete stable states [13]; it was also suggested, from the analysis of some neurophysiological data, that synaptic efficacies might be able to store a few bits of information each (between 1 and 5) [14]. What is certain is the fact that the biological brain is an extremely noisy yet very robust computational environment, marginally affected by the high failure rate of its single components or the extreme variety and blurriness of the received external stimuli: these properties are hardly compatible with a system that relies on "floating-point" continuous precision, as various theoretical results show that binary-synapses models could be more adequate than continuous ones as neuronal models exhibiting long term plasticity [28], showing an enhanced robustness against noise.

From the engineering point of view, switch-like synapses might be the building block for neuromorphic applications: in order to obtain a hardware implementation of neurons, the current attempt is to rely on the Memristor technology [52, 53], where an electronic component can be switched into one of two possible states, representing a binary (on/off) synapse. The hardware implementation of artificial neural networks, able to autonomously modify their local structure, could potentially produce a revolution in the world of information technology. On the other hand, even in the more common software implementation of deep neural networks, it was often observed [54, 55] that big reductions in the precision of the numerical representation of the synapses is not accompanied by any clear deterioration in the computational performance, thus suggesting that a very convenient memory compression is possible. However, it is not clear how to devise training protocols that take place directly in a discrete space and that are able to maximize their performance in this setting.

Finally, the theoretical motivation: it is easy to see that there is a qualitative jump in difficulty between learning in a continuous setting and in discrete one. While the reduction in the storage capacity, when going to discrete models, is surprisingly small, the geometric structure of the space of solutions in the SAT region (i.e., when the constraints are satisfiable and zero energy solutions can be found), at $\alpha < \alpha_c$, becomes very complex and intriguing. As we will see in the following sections, the space of solutions is no longer convex nor connected (if we allow a transposition to the discrete setting of these concepts): typical solutions are far apart in Hamming distance, surrounded by an exponential number of *glassy* sub-optimal configurations that entail an extensive number of errors. At the same time, the Gibbs measure does not undergo the usual transitions, where the structures formed by the ensemble of solutions progressively break apart into smaller and more numerous pure states (clusters of solution). As a consequence, differently from other well-studied constraint satisfaction problem, the discrete Perceptron learning problem appears to be always in a hard phase (where only exponential algorithms should be able to find solutions), and local search algorithms can easily get trapped in the exponential local minima. However, the recent proposal of feasible and simple learning strategies [29, 28, 26] stimulates a deeper theoretical analysis of the geometry of the phase space of these models, requiring to move from the “classical” equilibrium description to a novel "out-of-equilibrium" analysis, able to grasp the relevant structures allowing the effectiveness of these algorithms.

2.2 The Replica Symmetric Analysis

We start our theoretical analysis of the binary Perceptron model by revisiting the Gardner analysis [56] in the special case of a synaptic measure of the form: $\mathrm{d}\mu\left(W\right) = \prod_i \left(\delta\left(W_i - 1\right) + \delta\left(W_i + 1\right)\right)$. We first consider the random classification problem [17–20], where both the components ξ_i^μ and the outputs σ^μ are drawn uniformly at random from $\{-1, 1\}$. It is easy to see that this model possesses a $\mathbb{Z}_2$ symmetry, since the cut in the phase space induced by any random association ξ, σ is the same one induced by $-\xi, -\sigma$: this allows one to set all the outputs to one (trivializing the corresponding average), without loss of generality.

As introduced in section **2.1**, the correct self-averaging quantity we want to evaluate is the quenched entropy density, i.e. the normalized logarithm of the typical number of solutions left in the space of synaptic couplings at a given storage load α. In order to compute the average over the disorder, represented by the possible choices for the training set, we need to apply the replica trick (see section **2.1**). Therefore, we first compute the replicated volume for positive and discrete values of n:

$$\left\langle \Omega^n \left(\{\xi^\mu\}_{\mu=1}^{\alpha N} \right) \right\rangle_\xi = \left\langle \int \prod_{a=1}^{n} \mathrm{d}\mu\left(W^a\right) \prod_{a=1}^{n} \prod_{\mu=1}^{\alpha N} \Theta\left(\frac{\sum_i W_i^a \xi_i^\mu}{\sqrt{N}} \right) \right\rangle_\xi \tag{2.5}$$

The usual way for proceeding in the calculations is to introduce an auxiliary variable for the arguments of the Θ-functions and to fix their value through δ-functions:

$$\Theta\left(\frac{\sum_i W_i^a \xi_i^\mu}{\sqrt{N}} \right) = \int \frac{\mathrm{d}\lambda^{\mu,a} \mathrm{d}\hat{\lambda}^{\mu,a}}{2\pi} \Theta\left(\lambda^{\mu,a}\right) \exp\left(i\hat{\lambda}^{\mu,a} \left(\lambda^{\mu,a} - \frac{\sum_i W_i^a \xi_i^\mu}{\sqrt{N}} \right) \right) \tag{2.6}$$

so that the pattern dependence is isolated in an exponential term. We are now able to perform the disorder average, obtaining in the large N limit:

$$\left\langle \prod_{\mu,a} \exp\left(-i\hat{\lambda}^{\mu,a}\frac{\sum_i W_i^a \xi_i^\mu}{\sqrt{N}}\right)\right\rangle_\xi = \prod_\mu \exp\left(-\frac{1}{2}\sum_{ab}\hat{\lambda}^{\mu,a}\hat{\lambda}^{\mu,b}\left(\sum_i \frac{W_i^a W_i^b}{N}\right)\right) + \mathcal{O}\left(N^{-2}\right) \tag{2.7}$$

In this expression we can see the appearance of the order parameter of the model, $q^{ab} = \sum_i \frac{W_i^a W_i^b}{N}$, representing the overlap between two different replicas a and b. We can thus introduce an integration over its possible values via a δ-function (where in this case the conjugate parameter $\hat{q}^{ab}$ needs to be imaginary, so we can directly substitute $\hat{q}^{ab} \to i\hat{q}^{ab}$), and factorize over the synaptic indices $i = 1, ..., N$ and the pattern indices $\mu = 1, ..., \alpha N$, to obtain the following expression for the replicated volume:

$$\begin{aligned}&\left\langle \Omega^n\left(\{\xi^\mu\}_{\mu=1}^{\alpha N}\right)\right\rangle_\xi = \\ &\int \prod_{a<b}\frac{\mathrm{d}q^{ab}\,\mathrm{d}\hat{q}^{ab}}{(2\pi/N)}\exp\left(-N\sum_{a<b}\hat{q}^{ab}q^{ab}\right)(G_S)^N\,(G_E)^{\alpha N}\end{aligned} \tag{2.8}$$

In a physics analogy, we introduced the entropic (i.e., linked to the volume) and the energetic (i.e., linked to the constraints) single-body partition functions, respectively:

$$G_S = \int \prod_{a=1}^n \mathrm{d}\mu\left(W^a\right)\exp\left(\sum_{a<b}\hat{q}^{ab}W^aW^b\right) \tag{2.9}$$

$$G_E = \int \prod_{a=1}^n \frac{\mathrm{d}\lambda^a \mathrm{d}\hat{\lambda}^a}{2\pi}\exp\left(-\frac{1}{2}\sum_{ab}q^{ab}\hat{\lambda}^a\hat{\lambda}^b\right) \tag{2.10}$$

In order to continue with the replica calculation, we need to chose an Ansatz on the structure of the overlap matrix: we start from the most basic one, the RS Ansatz described in section **2.1**, posing:

- $q^{ab} = 1$, if $a = b$; $q^{ab} = q$ otherwise.
- $\hat{q}^{ab} = \hat{q}$ for all $a < b$.

We can therefore proceed with the computation of the entropic term, where we can perform a Hubbard-Stratonovich transformation, introducing the Gaussian measure $\int \mathcal{D}z = \int_{-\infty}^{+\infty} \frac{dz}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$, to obtain:

$$G_S = \exp\left(-n\hat{q}\right) \int \mathcal{D}z \left(2\cosh\left(\sqrt{\hat{q}}z\right)\right)^n \tag{2.11}$$

therefore, taking the logarithm $\mathcal{G}_S = \log G_S$, in order to recover the saddle point expression for the entropy density, in the limit $n \to 0$, we get:

$$\mathcal{G}_S = -n\frac{\hat{q}}{2} + n \int \mathcal{D}z \log\left(2\cosh\left(\sqrt{\hat{q}}z\right)\right) \tag{2.12}$$

Similarly, we can compute the logarithm of the energetic term in the $n \to 0$ limit, obtaining:

$$\mathcal{G}_E = n \int \mathcal{D}z \log\left(H\left(-\frac{\sqrt{q}z}{\sqrt{1-q}}\right)\right) \tag{2.13}$$

Therefore, by neglecting the $\mathcal{O}\left(n\right)$ terms, the expression for the entropy in this simple case is given by:

$$\begin{aligned} \mathrm{S}_{RS} = &- \hat{q}\frac{(1-q)}{2} + \int \mathcal{D}z \log\left(2\cosh\left(\sqrt{\hat{q}}z\right)\right) \\ &+ \alpha \int \mathcal{D}z \log\left(H\left(-\frac{\sqrt{q}z}{\sqrt{1-q}}\right)\right) \end{aligned} \tag{2.14}$$

where, following the standard notation [15], we introduced the error-function:

$$H\left(x\right) = \frac{1}{2}\mathrm{erfc}\left(\frac{\mathrm{x}}{\sqrt{2}}\right) \tag{2.15}$$

that will be used throughout this thesis. To compute this entropy we first have to numerically determine the typical values for the RS overlap q and its conjugate parameter $\hat{q}$, by imposing the saddle point conditions, $\partial\mathrm{S}_{RS}/\partial q = 0$ and $\partial\mathrm{S}_{RS}/\partial\hat{q} = 0$, and iterating the obtained equations.

Unfortunately, if we study the behavior of the RS entropy we get some puzzling results:

- the entropy becomes negative after a threshold, found numerically at $\alpha_{S=0} = 0.833$: in a discrete model, where the entropy is associated to a counting operation of a numerable set of points, negative values of the entropy do not make sense, so the calculation must be incorrect above this threshold.

- the overlap q reaches the value 1 only much later, at a storage load $\alpha_{E>0} = 1.27$; after this threshold value, the zero temperature energy of the model is found to become strictly positive, so the RS calculation predicts this value as the critical capacity of the model.

Both these thresholds need to be compared with a third threshold, $\alpha = 1$, this time obtained from a sanity check: since we are studying the case where both the patterns and the synapses are binary, it is easy to see that storing more than N bits of information (the correct outputs) into N bits (the synapses) would produce an information paradox. It is therefore clear that the $q \to 1$ criterion is no longer sufficient (as it was in the continuous Perceptron, see 2.1) for determining the critical threshold α_C of the model, and that the symmetry that was assumed in the RS Ansatz might be spontaneously broken before, thus causing some wrong predictions for the energy.

One way of checking whether a specific Ansatz might be suitable for a replica calculation, is to evaluate the local stability of the saddle point [49], by studying the Hessian of the entropy (or of the free energy, in general) in correspondence of the typical values for the parameters. In the binary Perceptron model the instability is numerically found above $\alpha_{AT} = 1.015$, so the threshold prediction at $\alpha_{E<0}$ must be discarded. However, even below this threshold we need to check for the global stability of the RS solution, since the model (undergoing some sort of a first order transition) might have many stable stationary points, and in the limit $N \to \infty$ only the maximum will prevail. The global stability analysis can be done by considering one step of replica symmetry breaking, as in Parisi's 1RSB Ansatz, in the calculation of the quenched average.

2.3 1-step Replica Symmetry Breaking

The 1RSB Ansatz follows a hierarchical symmetry breaking scheme, originally proposed by Parisi [35], describing a disconnected space of solutions, populated by clustered structures: the solutions are now organized in an exponential number of groups, geometrically separated between each other. Because of the glassy landscape, ergodicty is assumed to be completely broken and the Gibbs measure is decomposed into a convex linear combination of pure states α, related to each one of the different clusters and weighed by a probability P_α. Thus, the expectation value of any given observable O will be given by:

$$\langle O\rangle = \sum_a P_a \langle O\rangle_a \tag{2.16}$$

In this more complex Ansatz we need to differentiate between the intra-cluster overlap q_1 and the inter-cluster overlap q_0 (with $q_1 > q_0$), defined as:

$$q_1 = \frac{1}{N}\sum_i \overline{\langle W_i\rangle_a^2} \tag{2.17}$$

$$q_0 = \frac{1}{N}\sum_i \overline{\langle W_i\rangle_a \langle W_i\rangle_b} \tag{2.18}$$

Moreover we also introduce the Parisi parameter $m = 1 - \sum_a \overline{P}_a^2$, such that $P(q) = m\delta(q-q_0) + (1-m)\,\delta(q-q_1)$. The Parisi parameter can be used for focusing the measure on different clusters, spanning the space of solutions from the dominant to the sub-dominant structures, and allowing us to find the correct (globally stable) extremum of the free energy. Specifically, the n replicas will be grouped in $\frac{n}{m}$ blocks, containing m replicas each. It is useful to split the replica indices $a = 1, ..., n$ in two new indices expliciting the hierarchical organization: $a = (\alpha, \beta)$, where $\alpha \in \{1, ..., n/m\}$ labels the blocks, and $\beta \in \{1, ..., m\}$ labels the replicas inside the blocks.

So we can go back to expression 2.8, and try the substitutions:

- $q^{\alpha\beta,\alpha'\beta'} = 1$, if $\alpha = \alpha', \beta = \beta'$; $q^{\alpha\beta,\alpha'\beta'} = q_1$ if $\alpha = \alpha'$ and $\beta \neq \beta'$; $q^{\alpha\beta,\alpha'\beta'} = q_0$ otherwise.
- $\hat{q}^{\alpha\beta,\alpha'\beta'} = \hat{q}_1$ if $\alpha = \alpha'$ and $\beta \neq \beta'$; $\hat{q}^{\alpha\beta,\alpha'\beta'} = \hat{q}_0$ otherwise.

We can continue the computation, by performing two Hubbard-Stratonovich transformation in the entropic and energetic terms, and obtain the 1RSB expression ([17]) for the entropy density:

$$\mathrm{S}_{1RSB} = -\frac{\hat{q}_1(1-q_1)}{2} - \frac{m}{2}(\hat{q}_1 q_1 - \hat{q}_0 q_0) - \frac{\hat{q}_0 q_0}{2} + \mathcal{G}_S + \alpha\mathcal{G}_E \tag{2.19}$$

with the definitions:

$$\mathcal{G}_S = -\frac{n}{2}\hat{q}_1 + \frac{n}{m}\int \mathcal{D}z_0 \log \int \mathcal{D}z_1 \left(2\cosh\left(\sqrt{\hat{q}_1 - \hat{q}_0}z_1 + \sqrt{\hat{q}_0}z_0\right)\right)^m \tag{2.20}$$

$$\mathcal{G}_E = \frac{n}{m}\int \mathcal{D}z_0 \log \int \mathcal{D}z_1 H\left(-\frac{\sqrt{q_1 - q_0}z_1 + \sqrt{q_0}z_0}{\sqrt{1-q_1}}\right)^m \tag{2.21}$$

At the critical threshold, we now expect the various clusters to shrink and eventually disappear, while being still well separated in the solution space: therefore we can study the limit $q_1 \to 1$ with $q_0 < 1$. One can see that near the threshold the conjugate parameter $\hat{q}_1$ explodes as $\sim 1/\sqrt{1-q_1}$, and obtain the new saddle point equation for m in this limit: by choosing a proper scaling, $\hat{q}_0 = \hat{q}/m^2$ and $q_0 = q$, the resulting equation can be recognized as equivalent to the requirement $\mathrm{S}_{RS}(\alpha_C) = 0$. We already know that this happens at the threshold found in the previous paragraph, $\alpha_C = \alpha_{S=0} = 0.833$, therefore the critical capacity can be correctly estimated through a so-called *zero entropy criterion* [17]. It is important to stress the impressive robustness of neural networks, as the storage capacity is only cut to roughly 40%, when going from continuous to binary synapses. In correspondence of α_C the inter-cluster overlap is still notably small, $q \sim 0.5$, and this is one of the reasons the $q \to 1$ criterion gave incorrect results in the RS Ansatz. The behavior of the entropy density is exactly the same one observed in the RS Ansatz until $\alpha < \alpha_C$, while above the critical threshold it is corrected, as the entropy remains fixed to 0 (avoiding unphysical negative values). In figure 2.2, we can see the entropy density curve as a function of the storage load α.

In general, in a 1RSB analysis, one assumes to find at each m an exponential number, $\mathcal{N}(m) \propto \exp(N\Sigma(m))$, of pure states characterized by different internal entropies: the dominant contribution to the measure is given by the Legendre transformation:

$$\mathrm{S} \sim \mathrm{extr}_m \left[m\mathcal{S}(m) + \Sigma(m)\right] \tag{2.22}$$

However, a part from the dominant solutions, described at the saddle point value for m, we can also look for different ones, by dropping this stationarity requirement and spanning the entire interval $0 \leq m \leq 1$. We can thus further characterize the sub-dominant clusters and separate their contributions to

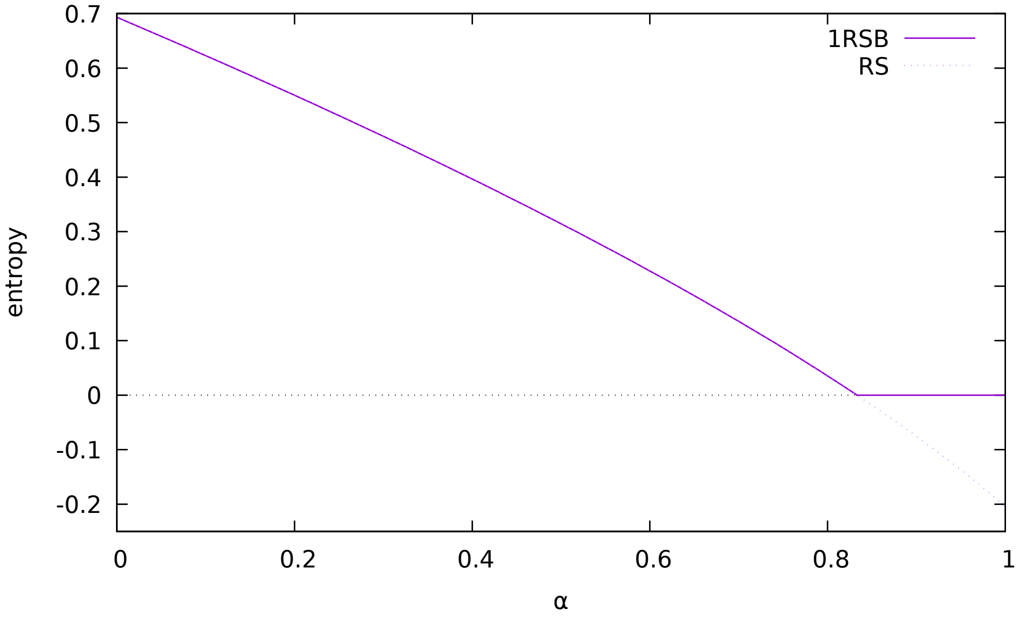


Fig. 2.2 **Entropy density curve** as a function of the storage load. The RS prediction is exact until the critical threshold $\alpha_c = 0.833$, after which it returns negative values for the entropy. However, this problem is fixed in the 1RSB Ansatz.

$\mathrm{S}\,(m)$, into an *internal entropy* $\mathcal{S}$ and a *complexity* (or external entropy) Σ, defined as:

$$\mathcal{S}\,(m) = \frac{\partial}{\partial m}\left[m\mathrm{S}\,(m)\right] \tag{2.23}$$

$$\Sigma\,(m) = -m^2\frac{\partial}{\partial m}\left[\mathrm{S}\,(m)\right] \tag{2.24}$$

These potentials represent, respectively, the number of solutions inside each cluster (of the size determined by m) and the number of clusters of this size.

Interestingly, in the binary Perceptron, no other solutions can be found below $\alpha < \alpha_C$ except for the one with $q_1 = 1$ and $q_0 = q_{RS}$, and the trivial one (equivalent to the RS Ansatz), with $m = 0$ and $q_1 = q_0 = q_{RS}$. From a finite temperature study, which can be found in detail in [17], we report that a total freezing phenomenon can be observed for values $\alpha > \alpha_C$: if $T > T_c$, there still is only the RS solution; if $T < T_c$ a 1RSB solution can be found with $m = T/T_c$, $q_0 = q$, $\hat{q}_0 = \hat{q}/m^2$, $q_1 = 1$ and $\hat{q}_1 = \infty$. The free energy of the systems is found to be independent of T and equal to the replica symmetric free energy at T_c. This transition is similar to a first order one, since the order parameter function (a sum of two δ-functions) is discontinuous, while the free energy is still continuous around T_c.

Because of the non-rigorous nature of the replica calculations, we cannot state that these results are correct, but it was proven (again in [17]) that they are robust with respect to additional steps of replica symmetry breaking. In fact, from the 2RSB calculation one can find no other solutions, implying that there are no further symmetry breaking effects to be taken into account. Therefore the geometrical picture in the binary Perceptron model seems to be the following:

- The equilibrium analysis is not able to find any sub-dominant structures in the solution space, with $0 \leq m \leq 1$ and $q_1 \neq 1$.
- The dominant solutions do not merge into clusters of various sizes, but seem to be organized into an exponential number of point-like pure states, for any $\alpha < \alpha_C$.
- Below the critical threshold, the model is RS with a unique pure state described by the overlap order parameter q_{RS}. However, this state is peculiar in that, in a 1RSB analysis, it can also be seen as an ensemble of point-like clusters, with $q_1 = 1$ and $q_0 = q_{RS}$: these states have zero internal entropy, but the complexity (i.e., the number of states) is equivalent to the RS internal entropy (i.e. the number of solutions in the RS cluster).

2.4 Learning from a teacher

The teacher-student scenario can be studied by means of the replica trick in a very similar way, resulting in some slightly more involved calculations [15]. In this case, of course, the statistics of the outputs σ^μ is not trivial, since they are determined by the teacher device W^T. However, since we want to describe the typical learning behavior, which would require an average over all the possible choices for the teacher, in full generality we can instead exploit the symmetry of the problem and fix the gauge $W_i^T = 1$, for all $i = 1, ..., N$.

Therefore, every constraint will now require an additional substitution for the label $\sigma = \text{sign}\left(\sum_i \xi\right)$, but we can exploit the scaling invariance of the Θ-function and write the constraints as:

$$\begin{aligned}\Theta\left(\frac{\sum_i \xi_i}{\sqrt{N}}\frac{\sum_i W_i^a \xi_i^\mu}{\sqrt{N}}\right) = \int \frac{\mathrm{d}u^\mu \mathrm{d}\hat{u}^\mu}{2\pi} \int \frac{\mathrm{d}\lambda^{\mu,a}\mathrm{d}\hat{\lambda}^{\mu,a}}{2\pi}\Theta\left(u^\mu \lambda^{\mu,a}\right) \\ \exp\left(i\hat{\lambda}^{\mu,a}\left(\lambda^{\mu,a} - \frac{\sum_i W_i^a \xi_i^\mu}{\sqrt{N}}\right) + i\hat{u}^\mu\left(u^\mu - \frac{\sum_i \xi_i^\mu}{\sqrt{N}}\right)\right)\end{aligned} \tag{2.25}$$

Now, in the disorder average, we have two terms depending on the patterns, so we need to introduce a new order parameter:

$$R^a = \frac{W^T \cdot W^a}{N} = \sum_{i=1}^{N} \frac{W_i^a}{N} \tag{2.26}$$

representing the alignment between the typical student, i.e. one of the dominant solutions of the training problem, and the teacher. The rest of the computation is very similar to the previous one, giving the following result:

$$\begin{aligned} S = &-\hat{q}\frac{(1-q)}{2} - \hat{R}R \\ &+ \int \mathcal{D}z \log\left(2\cosh\left(\sqrt{\hat{q}}z + \hat{R}\right)\right) \\ &+ 2\alpha \int \mathcal{D}z\, H\left(-\frac{R\, z}{\sqrt{q - R^2}}\right) \log\left(H\left(-\frac{\sqrt{q}z}{\sqrt{1-q}}\right)\right) \end{aligned} \tag{2.27}$$

If we analyze the behavior of the entropy density, similarly to the random classification case we can see that the RS Ansatz predicts a threshold, at $\alpha_D = 1.245$, after which the symmetry assumption becomes incorrect, since the entropy reaches unphysical negative values. Again, the computation can be revised in the 1RSB approach, where one can see that, after α_D, the RS saddle point no longer gives the global maximum of the entropy, which is instead found at the extremal value $R = 1$ (where of course the entropy density is $\mathrm{S} = 0$, as the phase space has shrunk to the single teacher configuration).

It is natural to question the performance of the typical student on so far unseen input-output associations, by studying the *generalization error*:

$$\epsilon\left(\alpha\right)=\left\langle \Theta\left(-W^{T}\cdot\xi^{\star}\left(\sum_{i=1}^{N}\frac{W_{i}\cdot\xi_{i}^{\star}}{N}\right)\right)\right\rangle _{\left\{ \xi^{\mu},\sigma^{\mu}\right\} _{\mu=1}^{\alpha N}}\tag{2.28}$$

which is simply computed as the probability of obtaining a classification error on a new random pattern, after the learning procedure. This quantity is the main probe of the learning performance of an ANN, even when one deals with real world data. Of course, we expect from our geometrical intuition that the generalization properties of the solutions W are determined by their alignment with the teacher, measured by the R order parameter. In fact the simple result is:

$$\epsilon\left(\alpha\right)=\frac{1}{\pi}\arccos\left(R\left(\alpha\right)\right)\tag{2.29}$$

Because of the first order transition at $\alpha_D = 1.245$, in correspondence of this threshold we have a *discontinuous* jump to perfect learning: all the other solutions to the learning problem are decimated and the sole teacher is left [18, 15]. However, from the algorithmic point of view, there is a large window around α_D, extending until $\alpha \simeq 1.5$, where the problem of numerically finding the teacher remains hard, even when the provided information is theoretically sufficient for determining it. This is probably due to the presence of a metastable regime [15, 29]. In figure 2.3 we can see the entropy density and generalization curves as a function of the storage load α.

2.5 The generalized Perceptron

As stated in the introduction, we are also interested in considering the general discrete case, where the synaptic weights can take values in a finite discrete set. Let's consider, for simplicity, the set $W_i \in \{0, 1, \dots, L-1, L\}$, where L is the value of the highest available synaptic state. We can also consider a more general scenario for the input-output patterns, where we allow a bias in the statistics of the inputs and outputs $P\left(x\right) = f\delta\left(x-1\right) + \left(1-f\right)\delta\left(x\right)$, and we

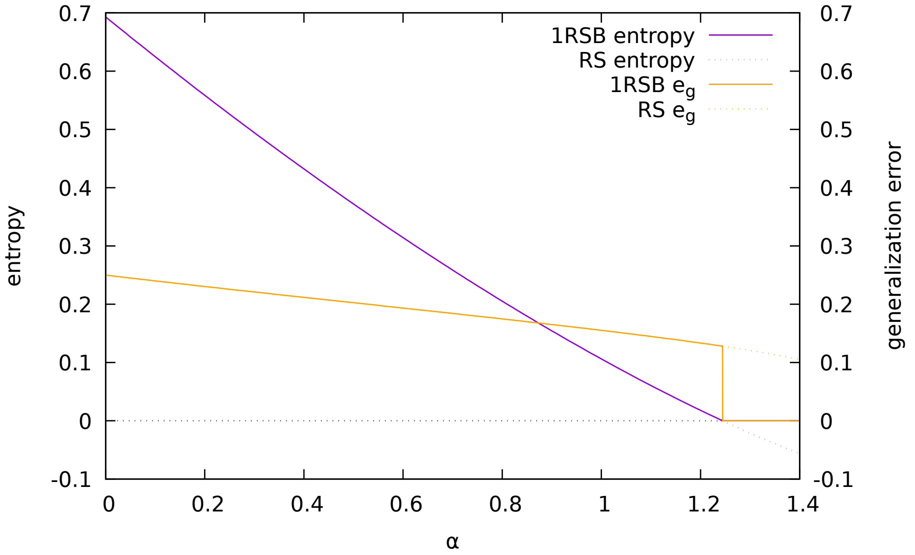


Fig. 2.3 **Entropy density and generalization error** as a function of the storage load. At low α, where the entropy density is strictly positive, the RS Ansatz produces correct results, in agreement with the 1RSB analysis. After the critical value $\alpha_D = 1.2445$, the extremal point $R = 1$ becomes the global minimum and the generalization error drops to zero discontinuously.

assume them to be sparse, $\xi_i^\mu, \sigma^\mu \in \{0, 1\}$: in this context the bias f is in fact called *sparsity* or *coding level* of the model.

Because of the different choices for the synapses and the pattern distribution, the average of the scalar product is no longer centered around 0, and we must introduce a firing threshold θ; therefore, the state of the generalized discrete neuron is determined as:

$$\tau\left(W, \xi\right) = \Theta\left(\sum_{i=1}^{N} W_i \xi_i - \theta N\right) \tag{2.30}$$

In order to repeat our Statistical Physics analysis in the random classification scenario, we thus have to modify the definition of the constraints to the form:

$$\Theta\Big(\left(2\sigma^\mu - 1\right)\left(\sum_{i=1}^{N} \frac{W_i \xi_i}{\sqrt{N}} - \theta\sqrt{N}\right)\Big) \tag{2.31}$$

Following step by step the calculations of section 2.2, we need to perform the average over the pattern distribution, giving:

$$\prod_{\mu,i} \left\langle \exp\left(-\frac{i}{\sqrt{N}}\left(\sum_a \hat{\lambda}_\mu^a W_i^a\right)\xi_i^\mu\right)\right\rangle_\xi = \tag{2.32}$$
$$\prod_\mu \exp\left(-i\bar{\xi}\sqrt{N}\left(\sum_a \hat{\lambda}_\mu^a \sum_i \frac{W_i^a}{N}\right) - \frac{\sigma_\xi^2}{2}\left(\sum_{ab}\hat{\lambda}_\mu^a \hat{\lambda}_\mu^b \sum_i \frac{W_i^a W_i^b}{N}\right)\right)$$

As we can see the result explicitly depends on the average $\overline{\xi}$ and the varianceσ_ξ^2 of the pattern distribution, and we need to introduce an additional order parameter for the L_1-norm of the synaptic weights. In order for the Perceptron to be able to balance its outputs it is natural to require that, on average:

$$\overline{W} = \frac{\theta}{f} \tag{2.33}$$

However, since the distribution of the outputs is also biased, we need to introduce an $O\left(\frac{1}{\sqrt{N}}\right)$ correction, controlled by a new order parameter, M:

$$\sum_i \frac{W_i}{N} = \overline{W} + \frac{M}{\sqrt{N}} \tag{2.34}$$

The only remaining difference with respect to the binary case comes from the fact that the L_2-norm of W is no longer trivially fixed to N, so we also add the associated order parameter $Q^a = \sum_i \left(W_i^a\right)^2 /N$. The computation can proceed in the same fashion of the one presented in section 2.2, obtaining the final result:

$$\mathrm{S} = -\hat{q}\frac{q}{2} - \hat{Q}Q - \hat{M}\overline{W} + \mathcal{G}_S + \alpha\mathcal{G}_E \tag{2.35}$$

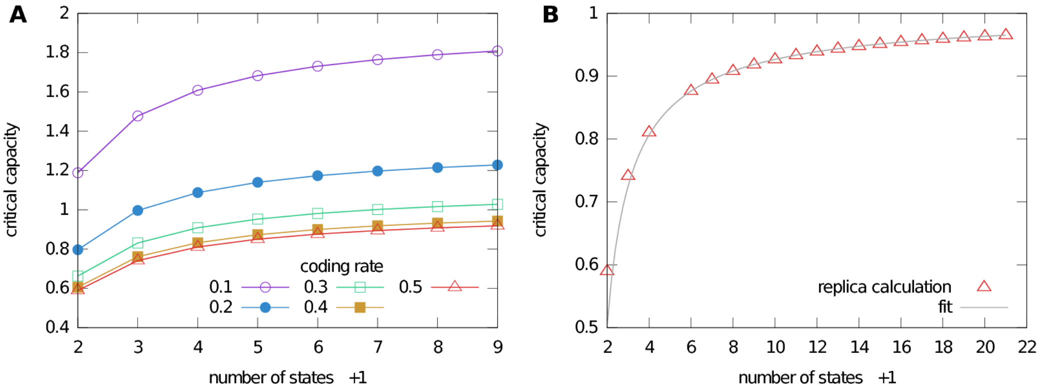


Fig. 2.4 **A. Critical capacity** α_c as a function of of the number of states per synapse $L+1$, for different values of the coding rate f. **B.** Same as in panel A, but only for the dense (unbiased) case $f = 0.5$, with a wider range of L, and showing a fit of the form $\alpha^\infty - \frac{a}{L^b}$ over the last part of the curve ($L \geq 5$). The fit parameters are $\alpha^\infty \simeq 1.0$, $a \simeq 0.5$, $b = 0.85$.

with the generalized entropic and energetic contributions:

$$\mathcal{G}_S = \int \mathcal{D}z \log\left(\sum_{l=0}^{L} \exp\left(\left(\hat{Q} - \frac{1}{2}\hat{q}\right) l^2 + \left(z\sqrt{\hat{q}} + \hat{M}\right) l\right)\right) \tag{2.36}$$

$$\begin{aligned}\mathcal{G}_E = &(1-f)\int \mathcal{D}z\ \log\left(H\left(\frac{\bar{\xi}M - z\sqrt{\sigma_\xi^2 q}}{\sqrt{\sigma_\xi^2\left(Q-q\right)}}\right)\right) \\ &+ f\int \mathcal{D}z\left\langle \log\left(H\left(-\frac{\bar{\xi}M + z\sqrt{\sigma_\xi^2 q}}{\sqrt{\sigma_\xi^2\left(Q-q\right)}}\right)\right)\right\rangle_s\end{aligned} \tag{2.37}$$

As it is clear from a simple comparison between the expressions obtained in the binary and the general discrete case, the qualitative scenario is not modified by they introduction of many synaptic states or by the sparse statistics of the patterns. The model is still Replica Symmetric, as the RS entropy density curve is exact until $\alpha_c = \alpha_{S=0}$, after which there is a freezing phenomenon. Of course, we expect the critical capacity to increase with the addition of more degrees of freedom to the model, as $L \to \infty$, and with a reduction in the information content of the single patterns, as $f \to 0$. We can thus use the RS expression for the entropy density, and the zero-entropy criterion, for computing the exact dependence of theoretical critical capacity of the system on the number of states per synapse $L+1$ and of the coding rate f.

We know from the literature that:

- The Perceptron with positive continuous weights (corresponding to the limit $L \to \infty$ of our discrete model) has a critical capacity of$\alpha_c = 1$ when the inputs are extracted uniformly from $\{0, 1\}$ (i.e., $f = 0.5$) [57];
- The binary Perceptron, with $W_i \in \{0, 1\}$ ($L = 1$) and unbiased inputs $f = 0.5$, reaches a capacity of $\alpha_c = 0.59$ [58]; the optimal neuronal threshold θ, in this case, is $\theta \simeq 0.16$.

Figure 2.4B shows the saturation of the continuous limit $\alpha_c = 1.$ as L is increased in the $f = 0.5$ case. From this study, one can extract an interesting general observation: the gain in capacity, when an ulterior synaptic state is added, decreases very rapidly after the first few values. Therefore, it seems that indefinitely increasing the synaptic precision could prove not to be a sensible engineering (or biological) strategy: notwithstanding a linear increase in the implementation cost one only obtains a small computational or representational advantage. This new theoretical result is consistent with the hypothesis that biological synapses would only need few bits of precision [3].

2.6 Local exploration of the solution landscape

In the last few sections we have described the geometrical organization of typical solutions of discrete Perceptron models: below the critical threshold α_C, these constraint satisfaction problems are always found to be in a peculiar RS phase, where the Gibbs measure is a pure state formed by an exponential number of point-like clusters, with an internal overlap $q_1 = 1$, and with an inter-cluster overlap $q_0 = q_{RS}$ equal to the value of the typical overlap obtained in the RS analysis.

We can continue our analysis by asking a different question: what does the immediate neighborhood of a typical solution look like? Is it possible to find other solutions in close proximity? The suitable theoretical framework for getting some insight about the local properties of the solution space was originally proposed by Parisi and Franz in [59]. In that context the goal was to give a characterization of the meta-stable state structures in mean-field spin glasses: they introduced a free energy potential, the so-called Franz-Parisi potential, measuring the cost of maintaining a system at a given temperature, while under the constraint of being at a fixed distance from an equilibrium configuration selected at a different temperature.

However, as proposed in [22], it is possible to consider a zero-temperature limit of this procedure and use the same formalism for describing the entropic landscape around a typical solution of a constraint satisfaction problem. Therefore, in the following, we investigate these local properties in the binary case, both in the classification and in the teacher-student scenarios [1]. The binary case is here considered for simplicity, while the generalization to the discrete Perceptron and sparse patterns can be found in [3]: we report that the qualitative results are unaltered also in that scenario.

2.6.1 Random Classification scenario

The computation of the Franz-Parisi potential is conceptually divided in two stages: first, one selects a reference configuration $\tilde{W}$ from the equilibrium Boltzmann-Gibbs measure at a certain inverse temperature β'; then, the idea is to evaluate the free energy of a coupled model where the configurations $\{W\}$, at inverse temperature β, are constrained to be at a distance D from the reference point:

$$\mathcal{S}_{FP}\left(\beta', \beta, D\right) = \frac{1}{N} \left\langle \frac{1}{Z\left(\beta'\right)} \sum_{\{\tilde{W}\}} e^{-\beta' E\left(\tilde{W}\right)} \log\left(\sum_{\{W\}} e^{-\beta E(W)} \delta\left(d\left(W, \tilde{W}\right) - D\right)\right)\right\rangle_{\{\xi,\sigma\}} \tag{2.38}$$

In the following we will consider the Perceptron as a constraint satisfaction problem, in the limit where both temperatures are set to zero ($\beta, \beta' \to \infty$). An important remark has to be made: the sampling of $\tilde{W}$ is completely unaffected by the coupling to the $\{W\}$ system, since this configuration is extracted at random from a flat distribution over all possible solutions (i.e., the Gibbs measure at zero temperature) and represents the *typical* solution (i.e. numerically dominant and thus most frequent). Again, as in section 2.2, we can set all the outputs $\sigma^{\mu} = 1$, $\mu = 1, ..., \alpha N$.

As noted above, the Franz-Parisi potential can be interpreted as a typical *local entropy* density of solutions:

$$\mathcal{S}_{FP}\left(D\right)=\frac{1}{N}\left\langle\left\langle\log\sum_{\{W\}}\prod_{\mu}\Theta\left(\sum_{i}\frac{W_i\xi_i^{\mu}}{\sqrt{N}}\right)\delta\left(d\left(W,\tilde{W}\right)-D\right)\right\rangle_{\tilde{W}}\right\rangle_{\{\xi\}} \tag{2.39}$$

where the averaging $\langle\cdot\rangle_{\tilde{W}}$ is performed over the flat measure on all solutions to the problem, and depends on the quenched noise $\{\xi^{\mu}\}$. For a generic function f, the two averages can be written as:

$$\left\langle\left\langle f\left(\{\xi^{\mu}\},\tilde{W}\right)\right\rangle_{\tilde{W}}\right\rangle_{\{\xi^{\mu}\}}=\left\langle\frac{\int\prod_i d\mu\left(\tilde{W}_i\right)\prod_{\mu}\Theta\left(\sum_i\frac{\tilde{W}_i\xi_i^{\mu}}{\sqrt{N}}\right)f\left(\{\xi^{\mu}\},\tilde{W}\right)}{\int\prod_i d\mu\left(\tilde{W}_i\right)\prod_{\mu}\Theta\left(\sum_i\frac{\tilde{W}_i\xi_i^{\mu}}{\sqrt{N}}\right)}\right\rangle_{\{\xi^{\mu}\}} \tag{2.40}$$

This kind of ensemble average can be rewritten using the replica trick: we write the denominator as the product of $\tilde{n}-1$ replicas and take the limit $\tilde{n}\to 0$, assigning the replica index 1 to the expression in the numerator and the indices 2 to $\tilde{n}$ to the others. In the following, we will always use the indices c and d for the replicas of the reference typical configuration:

$$\left\langle f\left(\{\xi^{\mu}\},\tilde{W}\right)\right\rangle_{\tilde{W}}=\lim_{\tilde{n}\to 0}\int\prod_{ic}d\mu\left(\tilde{W}_i^c\right)\prod_{c\mu}\Theta\left(\sum_i\frac{\tilde{W}_i^c\xi_i^{\mu}}{\sqrt{N}}\right)f\left(\{\xi^{\mu}\},\tilde{W}^1\right) \tag{2.41}$$

Moreover, we will use the indices $a,b\in\{1,\dots,n\}$ for the replicated W and $c,d\in\{1,\dots,\tilde{n}\}$ for the replicated $\tilde{W}$.

$$\begin{aligned}\mathcal{S}_{FP}\left(D\right)=&\frac{1}{N}\lim_{n,\tilde{n}\to 0}\frac{\partial}{\partial n}\left\langle\int\prod_{i,c}d\mu\left(\tilde{W}_i^c\right)\int\prod_{i,a}d\mu\left(W_i^a\right)\prod_{c\mu}\Theta\left(\sum_i\frac{\tilde{W}_i^c\xi_i^{\mu}}{\sqrt{N}}\right)\right.\\&\left.\times\prod_{a\mu}\Theta\left(\sum_i\frac{W_i^a\xi_i^{\mu}}{\sqrt{N}}\right)\prod_a\delta\left(\frac{1}{2}\sum_i\left(W_i^a-\tilde{W}_i^1\right)^2-2DN\right)\right\rangle_{\xi,\sigma}\\=&\frac{1}{N}\lim_{n\to 0}\frac{\partial}{\partial n}\Omega_{FP}^n\left(D\right)\end{aligned} \tag{2.42}$$

As usual, we first need to introduce some auxiliary variables for extracting the disorder dependence in the constraints:

$$\begin{aligned}&\prod_c \Theta\left(\sum_i \frac{\tilde{W}_i^c \xi_i^\mu}{\sqrt{N}}\right)\prod_a \Theta\left(\sum_i \frac{W_i^a \xi_i^\mu}{\sqrt{N}}\right) = \\ &\int \prod_{\mu,a} \frac{d\lambda_\mu^a d\hat{\lambda}_\mu^a}{2\pi} \int \prod_{\mu,c} \frac{d\tilde{\lambda}_\mu^c d\hat{\tilde{\lambda}}_\mu^c}{2\pi} \prod_{\mu,a} \Theta\left(\lambda_\mu^a\right) \prod_{\mu,c} \Theta\left(\tilde{\lambda}_\mu^c\right) \prod_{\mu,a} e^{i\lambda_\mu^a \hat{\lambda}_\mu^a} \\ &\times \prod_{\mu,c} e^{i\tilde{\lambda}_\mu^c \hat{\tilde{\lambda}}_\mu^c} \prod_{\mu,i} \left\langle \exp\left(-\frac{i}{\sqrt{N}}\left(\sum_a \hat{\lambda}_\mu^a W_i^a + \sum_c \hat{\tilde{\lambda}}_\mu^c \tilde{W}_i^c\right)\xi_i^\mu\right)\right\rangle_\xi\end{aligned} \tag{2.43}$$

Now we can perform the average over the pattern distribution:

$$\begin{aligned}&\prod_{\mu,i} \left\langle \exp\left(-\frac{i}{\sqrt{N}}\left(\sum_a \hat{\lambda}_\mu^a W_i^a + \sum_c \hat{\tilde{\lambda}}_\mu^c \tilde{W}_i^c\right)\xi_i^\mu\right)\right\rangle_\xi = \\ &\prod_\mu \exp\left(-i\sqrt{N}\left(\sum_a \hat{\lambda}_\mu^a \sum_i \frac{W_i^a}{N} + \sum_c \hat{\tilde{\lambda}}_\mu^c \sum_i \frac{W_i^c}{N}\right) + \right. \\ &\left. -\frac{1}{2}\left(\sum_{ab} \hat{\lambda}_\mu^a \hat{\lambda}_\mu^b \sum_i \frac{W_i^a W_i^b}{N} + \sum_{cd} \hat{\lambda}_\mu^c \hat{\lambda}_\mu^d \sum_i \frac{\tilde{W}_i^c \tilde{W}_i^d}{N} + 2\sum_{ac} \hat{\lambda}_\mu^a \hat{\tilde{\lambda}}_\mu^c \sum_i \frac{W_i^a \tilde{W}_i^c}{N}\right)\right)\end{aligned} \tag{2.44}$$

We can see that the coupled model can be described by the following order parameters:

- $\tilde{q}^{cd} = \sum_i \frac{\tilde{W}_i^c \tilde{W}_i^d}{N}$, representing the overlap between two typical solutions of the binary Perceptron.
- $q^{ab} = \sum_i \frac{W_i^a W_i^b}{N}$, representing the overlap between the coupled replicas W^a.
- $S^{ca} = \sum_i \frac{W_i^a \tilde{W}_i^c}{N}$, representing the overlap between one of the replicas W^a (coupled to $\tilde{W}^1$), and the reference solutions $\tilde{W}^c$.

We substitute these definitions in the expression of the replicated volume $\Omega^n(D)$, by using the integral representation of the Dirac delta distribution, and rearrange the integrals in order to factorize over the μ and i indices:

$$\begin{aligned}\Omega_{FP}^{n}\left(D\right) = &\lim_{\tilde{n}\to 0}\int\prod_{c>d}\frac{d\tilde{q}^{cd}d\hat{\tilde{q}}^{cd}}{\left(2\pi/N\right)}\int\prod_{a>b}\frac{dq^{ab}d\hat{q}^{ab}}{\left(2\pi/N\right)}\int\prod_{ca}\frac{dS^{ca}d\hat{S}^{ca}}{\left(2\pi/N\right)}\\ &\times\int\prod_{a}\frac{d\hat{D}^{a}}{2\pi}G_1\,\left(G_S\right)^{N}\,\left(G_E\right)^{\alpha N}\end{aligned} \tag{2.45}$$

where we have singled out a first term G_1 and the so-called entropic and energetic contributions G_S, G_E:

$$\begin{aligned}G_1 = &\exp\left(-N\left(\sum_{c>d}\hat{\tilde{q}}^{cd}\tilde{q}^{cd}+\sum_{a>b}\hat{q}^{ab}q^{ab}+\sum_{ca}\hat{S}^{ca}S^{ca}\right.\right.\\ &\left.\left.+\sum_{a}\hat{D}^{a}\left(1-2D-S^{1a}\right)\right)\right)\end{aligned} \tag{2.46}$$

$$\begin{aligned}G_S = &\int\prod_{c}d\mu\left(\tilde{W}^{c}\right)\int\prod_{a}d\mu\left(W^{a}\right)\exp\left(\sum_{c>d}\hat{\tilde{q}}^{cd}\tilde{W}^{c}\tilde{W}^{d}+\sum_{a>b}\hat{q}^{ab}W^{a}W^{b}+\right.\\ &\left.+\sum_{ca}\hat{S}^{ca}W^{a}\tilde{W}^{c}\right)\end{aligned} \tag{2.47}$$

$$\begin{aligned}G_E = &\int\prod_{a}\frac{d\lambda^{a}d\hat{\lambda}^{a}}{2\pi}\int\prod_{c}\frac{d\tilde{\lambda}^{c}d\hat{\tilde{\lambda}}^{c}}{2\pi}\prod_{a}\Theta\left(\lambda^{a}\right)\prod_{c}\Theta\left(\tilde{\lambda}^{c}\right)\\ &\times\exp\left(i\left(\sum_{a}\lambda^{a}\hat{\lambda}^{a}+\sum_{c}\tilde{\lambda}^{c}\hat{\tilde{\lambda}}^{c}\right)-\frac{1}{2}\sum_{a}\left(\hat{\lambda}^{a}\right)^{2}-\frac{1}{2}\sum_{c}\left(\hat{\tilde{\lambda}}^{c}\right)^{2}\right.\\ &\left.-\frac{1}{2}\sum_{(a,b)}\hat{\lambda}^{a}\hat{\lambda}^{b}q^{ab}-\frac{1}{2}\sum_{(c,d)}\hat{\tilde{\lambda}}^{c}\hat{\tilde{\lambda}}^{d}\tilde{q}^{cd}-\sum_{ac}\hat{\lambda}^{a}\hat{\tilde{\lambda}}^{c}S^{ac}\right)\end{aligned} \tag{2.48}$$

As in the previous computations, we continue by making an RS assumption in the structure of the replicated order parameters. However, we have to make a distinction between the overlaps S, involving the replica $\tilde{W}^{c=1}$ appearing in the distance constraint, and $\tilde{S}$ involving the other replicas of the reference configuration:

- $S^{ca} = S$ for $c = 1$, $S^{ca} = \tilde{S}$ for $c \neq 1$.
- $q^{ab} = q$, $\tilde{q}^{cd} = \tilde{q}$, $\hat{D}^{ca} = \hat{D}$.

We start by working on the first term G_1, where the $\tilde{n} \to 0$ limit can be taken directly, obtaining up to order $\mathcal{O}\left(n\right)$:

$$G_1 = \exp\left(-Nn\left(-\frac{1}{2}\hat{q}q + \hat{S}S - \hat{\tilde{S}}\tilde{S} + \hat{D}\left(1 - 2D - S\right)\right)\right) \tag{2.49}$$

In the entropic term, in order to be able to completely factorize over the indices $c = 1, ..., \tilde{n}$ and $a = 1, ..., n$, we have to reorganize the term:

$$\hat{\tilde{S}}\sum_a W^a \sum_c \tilde{W}^c = \frac{1}{2}\hat{\tilde{S}}\left(\sum_a W^a + \sum_c \tilde{W}^c\right)^2 - \frac{1}{2}\hat{\tilde{S}}\left(\sum_a W^a\right)^2 - \frac{1}{2}\hat{\tilde{S}}\left(\sum_c \tilde{W}^c\right)^2 \tag{2.50}$$

and then perform three Hubbard-Stratonovich transformations, for eliminating the squared sums, with the Gaussian variables x, z and $\tilde{z}$.

Then, we go through the following analytical steps:

1. Factorize over the replica index of the reference configurations.
2. Take the limit $\tilde{n} \to 0$, restoring the presence of the denominator.
3. Take the logarithm of the expression in the $n \to 0$ limit ($\mathcal{G}_S = \log G_S/n$).
4. Perform two rotations, between the integration variables ($\tilde{z}$, x) and (z, x).
5. Compute analytically the $\int \mathcal{D}x$ integral.

The final expression for the entropic term thus reads:

$$\begin{aligned}\mathcal{G}_S = &-\frac{\tilde{n}}{2n}\hat{\tilde{q}} - \frac{1}{2}\hat{q} + \\ &+ \int \mathcal{D}z \int \mathcal{D}\tilde{z} \frac{\sum_{\tilde{W}} \exp\left(\tilde{z}\sqrt{\hat{q}}\tilde{W}\right) \log\left(2\cosh\left(z\sqrt{\frac{\hat{q}\hat{\tilde{q}} - \hat{\tilde{S}}^2}{\hat{\tilde{q}}}} + \tilde{z}\frac{\hat{\tilde{S}}}{\sqrt{\hat{\tilde{q}}}} + \Delta\hat{S}\ \tilde{W}\right)\right)}{2\cosh\left(\tilde{z}\sqrt{\hat{q}}\tilde{W}\right)}\end{aligned} \tag{2.51}$$

where we defined $\Delta\hat{S} = \hat{S} - \hat{\tilde{S}}$.

We proceed similarly with the computation of the energetic term:

1. Reorganize the term coupled to the parameters S and $\tilde{S}$.

2. Take the $\tilde{n} \to 0$ limit.

3. Take the logarithm of the expression in the $n \to 0$ limit.

4. Set $\Delta S = S - \tilde{S}$.

5. Evaluate the $\hat{\tilde{\lambda}}$ and λ integrals, using the error function $H\left(x\right)$ defined in equation 2.15.

6. Perform the change of variables $z' = z - i\hat{\tilde{\lambda}}\frac{\Delta S\sqrt{\sigma_\xi^2}}{\sqrt{q-\tilde{S}}}$.

7. Perform two rotations between $(\tilde{z},\, x)$ and $(z,\, x)$

The point of the last steps where to be able to perform analytically the $\int \mathcal{D}x$ integral:

$$\int \mathcal{D}x\, H\left(-\frac{\tilde{z}\sqrt{\tilde{q}} + z\left(\frac{\Delta S\sqrt{\tilde{q}}}{\sqrt{q\tilde{q}-\tilde{S}^2}}\right) + x\left(\frac{\Delta S\sqrt{\left(\tilde{q}-\tilde{S}\right)\tilde{S}}}{\sqrt{\left(q\tilde{q}-\tilde{S}^2\right)\left(q-\tilde{S}\right)}}\right)}{\sqrt{1-\tilde{q}-\frac{(\Delta S)^2}{q-\tilde{S}}}}\right) =$$

$$H\left(-\frac{\tilde{z}\sqrt{\tilde{q}} + z\left(\frac{\Delta S\sqrt{\tilde{q}}}{\sqrt{q\tilde{q}-\tilde{S}^2}}\right)}{\sqrt{1-\tilde{q}-\frac{(\Delta S)^2}{q-\tilde{S}}+\frac{\Delta S^2\tilde{S}\left(\tilde{q}-\tilde{S}\right)}{\left(q\tilde{q}-\tilde{S}^2\right)\left(q-\tilde{S}\right)}}}\right) \tag{2.52}$$

Finally, we integrate over $\hat{\tilde{\lambda}}$ and $\tilde{\lambda}$ to obtain, in the $n \to 0$ limit:

$$\mathcal{G}_E = \log G_E/n = \tag{2.53}$$

$$\int \mathcal{D}z \int \mathcal{D}\tilde{z} \frac{H\left(-\frac{\tilde{z}\sqrt{\tilde{q}}+z\left(\frac{\Delta S\sqrt{\tilde{q}}}{\sqrt{q\tilde{q}-\tilde{S}^2}}\right)}{\sqrt{1-\tilde{q}-\frac{(\Delta S)^2}{q-\tilde{S}}+\frac{\Delta S^2\tilde{S}(\tilde{q}-\tilde{S})}{(q\tilde{q}-\tilde{S}^2)(q-\tilde{S})}}}\right)\log\left(H\left(-\frac{z\sqrt{\left(\frac{q\tilde{q}-\tilde{S}^2}{\tilde{q}}\right)}+\tilde{z}\sqrt{\frac{\tilde{S}^2}{\tilde{q}}}}{\sqrt{1-q}}\right)\right)}{H\left(-\frac{x\sqrt{\tilde{S}}+\tilde{z}\sqrt{\tilde{q}-\tilde{S}}}{\sqrt{1-\tilde{q}}}\right)}$$

Plugging all the terms into the expression of the volume, and recovering the initial expression in the $n \to 0$ limit, we can now write the saddle point approximation for the Franz-Parisi potential:

$$\mathcal{S}_{FP}\left(D\right) \approx -\frac{1}{2}\hat{q}\left(1-q\right)-\hat{S}S+\hat{\tilde{S}}\tilde{S}-\hat{D}\left(1-2D-S\right)+\mathcal{G}_{S}+\alpha\mathcal{G}_{E} \tag{2.54}$$

All the order parameters must satisfy the saddle point equations, obtained through the stationarity requirement $\delta\mathcal{S}_{FP}=0$. A useful observation that can be made, is that the reference solution is sampled independently from the flat Boltzmann distribution, so the typical value for the order parameter$\tilde{q}$ is the RS value, q_{RS}, found in the calculation presented in section 4.2.

Numerically, the resulting sub-system of 7 coupled equations can be easily solved by iteration, using Newton's method for the homogeneous equations. In order to minimize the number of homogeneous equations and strongly help convergence, one can recast the equations and use $\hat{Q}$ as a control parameter instead of D, since at the saddle point $\hat{Q}\left(D\right)$ is a bijective function of the distance.

As can be seen in figure 2.5, the qualitative result obtained from this calculation is very interesting: for all values of α, the typical solutions of the problem are extensively isolated, since a zero entropy gap is found in the close neighborhood ($D\to 0$) of the references $\tilde{W}$. Moreover, the minimal Hamming distance separating two solutions grows with the constraint density (see a sketch in figure 2.6). As suggested in [22], this geometrical landscape might explain the origin of the computational hardness in the binary Perceptron problem: when the number of synapses becomes large, learning strategies based on local exploration of the weight space (e.g., MCMC algorithms) will exhibit an exponential scaling in computational time for reaching finite storage capacities.

2.6.2 Teacher student scenario

We need to check whether the isolation property is only due to the random uncorrelated nature of the classification scenario. We therefore study the same Franz-Parisi potential also in the teacher-student-scenario [1], where the correct labels are correlated by the presence of a teacher: as in section **2.4** we can fix

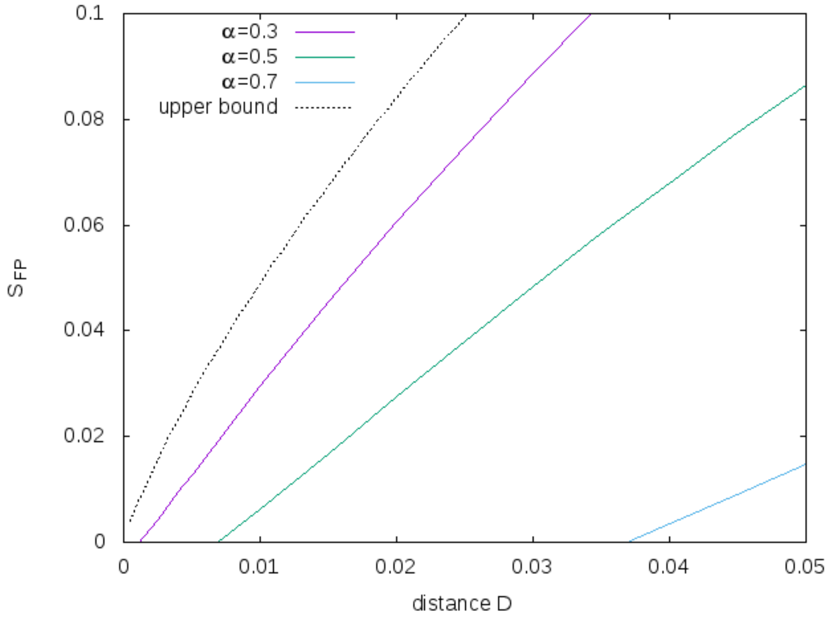


Fig. 2.5 **Franz Parisi potential** for the binary Perceptron in the random classification case. The upper bound is obtained by counting all the configurations at the given distance D (equivalent to the $\alpha = 0$ case). At all values of α, the Franz-Parisi potential becomes negative for positive values of the extensive distance D, signaling an entropy gap below that radius. Moreover, the gap widens as more constraints are added to the problem.

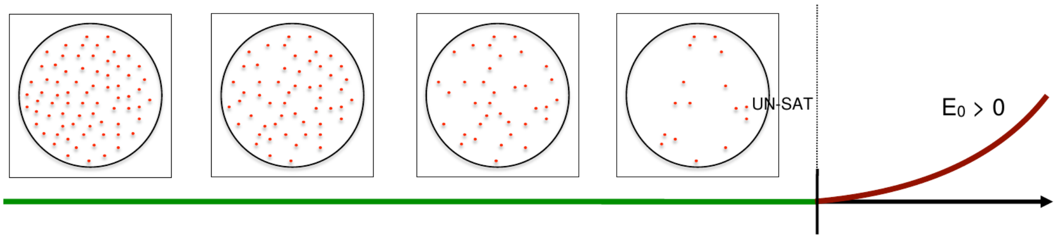


Fig. 2.6 **Sketch of the phase space** of the random binary Perceptron, as described by the equilibrium analysis. The typical solutions of the problem are extensively isolated at any pattern load.

arbitrarily the teacher: we choose again $W_i^T = 1$ for all $i = 1, ..., N$, so that for a given pattern ξ its correct output function is $\mathrm{sign}\left(\sum_i \xi_i\right)$.

The Franz-Parisi potential still consists of two steps: we generate αN random patterns, and we pick at random a typical student Perceptron with binary weights $\tilde{W} \in \{-1, 1\}$, which correctly classifies those patterns, and which we call “pseudo-teacher” in the following; then, we study the solution space of the training problem with an additional interaction term between the new student Perceptrons and the pseudo-teacher. In this way, we can describe the space of the solutions in the neighborhood of a given solution.

Both the patterns and the pseudo-teacher are considered as part of the quenched disorder and will be averaged out. As we have seen before, the addition of the correct outputs introduces an interaction between the students and the teacher: in this case we have two new order parameters, the overlap between the pseudo-teachers and the teacher $\tilde{R}^c = \sum_i \frac{\tilde{W}_i^c}{N}$, and the overlap between a student and the teacher $R^a = \sum_i \frac{W^a}{N}$. The computation follows straightforwardly the one presented in the previous paragraph: we only have to note that in the RS Ansatz the dependence on the replica indices in R and $\tilde{R}$ is completely dropped.

After some manipulation one obtains the entropic contribution to the Franz-Parisi potential:

$$\begin{aligned}\mathcal{G}_S = &-\frac{\hat{q}}{2} + \log 2 + \int Dz \log\cosh\left(z\sqrt{\hat{q}} + \hat{R} + \Delta\hat{S}\right) + \\ &-\int Dz \log\frac{\cosh\left(z\sqrt{\hat{q}} + \hat{R} + \Delta\hat{S}\right)}{\cosh\left(z\sqrt{\hat{q}} + \hat{R} - \Delta\hat{S}\right)} \\ &\times \int D\tilde{z} \frac{1}{1 + \exp\left(2\left(\left(z\hat{T} + \tilde{z}\sqrt{1 - \hat{T}^2}\right)\sqrt{\hat{\tilde{q}}} + \hat{\tilde{R}}\right)\right)}\end{aligned} \tag{2.55}$$

where we defined:

$$\Delta\hat{S} = \hat{S} - \hat{\tilde{S}} \tag{2.56}$$

$$\hat{T} = \frac{\hat{\tilde{S}}}{\sqrt{\hat{q}\hat{\tilde{q}}}} \tag{2.57}$$

and the energetic term, which instead reads:

$$\begin{aligned}\mathcal{G}_E =& 2\int Dz\int D\tilde{z}\frac{H\left(\frac{\tilde{R}\sqrt{p}\left(zT-\tilde{z}\sqrt{1-T^2}\right)-zR\sqrt{\tilde{q}}}{\sqrt{\left(\tilde{q}-\tilde{R}^2\right)(p-R^2)-\left(\tilde{S}-R\tilde{R}\right)^2}}\right)}{H\left(-\sqrt{\frac{\tilde{q}}{1-\tilde{q}}}\tilde{z}\right)} \\ &\times\int_{-\tilde{z}\sqrt{\frac{\tilde{q}}{1-\tilde{q}}}}^{\infty} D\tilde{\lambda}\log H\left(-\frac{\sqrt{p}\left(z\sqrt{1-T^2}+\tilde{z}T\right)+\frac{\Delta S}{\sqrt{1-\tilde{q}}}\tilde{\lambda}}{\sqrt{1-q}}\right)\end{aligned} \tag{2.58}$$

with the definitions:

$$\Delta S = S - \tilde{S} \tag{2.59}$$

$$p = q - \frac{\Delta S^2}{1-\tilde{q}} \tag{2.60}$$

$$T = \frac{\tilde{S}}{\sqrt{\tilde{q}p}} \tag{2.61}$$

Therefore the final expression for the local entropy density is thus given by:

$$\mathcal{S}_{FP}(D) = \ -\frac{1}{2}\hat{q}(1-q) - \hat{S}S + \hat{\tilde{S}}\tilde{S} - \hat{D}(1-2D-S) - R\tilde{R} + \log 2 + \mathcal{G}_S + \alpha\mathcal{G}_E \tag{2.62}$$

The order parameters $\tilde{q}$, $\tilde{R}$ and their conjugates can again be fixed to their RS value, found in the typical teacher-student setting (see section **2.4**). Therefore, we can add the restrictions that

$$\tilde{q} = \tilde{R} = q^\star \tag{2.63}$$

$$\hat{\tilde{q}} = \hat{\tilde{R}} = \hat{q}^\star \tag{2.64}$$

where $q^\star$ and $\hat{q}^\star$ satisfy:

$$q^\star = \int Dz\tanh\left(z\sqrt{\hat{q}^\star} + \hat{q}^\star\right) \tag{2.65}$$

$$\hat{q}^\star = \alpha\sqrt{\frac{2}{\pi}}\frac{1}{\sqrt{1-q^\star}}\int Dz\, GH\left(z\sqrt{q^\star}\right) \tag{2.66}$$

and where we defined the function GH as the ratio of the gaussian and the error-function:

$$GH\left(x\right)=\frac{G\left(x\right)}{H\left(x\right)} \tag{2.67}$$

When these constraints are added, the saddle point equation can be simplified, since the expressions for $\tilde{S}$ and $\hat{\tilde{S}}$ become equivalent to those for R and $\hat{R}$, respectively. More precisely: assuming $\tilde{S}=R$ it can be proved that $\hat{\tilde{S}}=\hat{R}$, and vice-versa. Numerical simulations confirm that this is the solution which is found when solving the full saddle point equations by an iterative procedure. Therefore, 6 order parameters remain to be determined. The reduction also allows a drastic simplification of the equations, allowing to perform a further analytical integration in each expression, after a change of variables.

First, we write the expressions of $\mathcal{G}_S$ and $\mathcal{G}_E$, which are needed for the computation of the entropy (2.62):

$$\mathcal{G}_S=\int Dz\,\log\cosh\left(z\sqrt{\hat{q}}+\hat{S}\right) \tag{2.68}$$

$$\mathcal{G}_E=2\int Dz\,\log H\left(z\sqrt{\frac{q}{1-q}}\right)H\left(z\frac{S}{q-S^2}\right) \tag{2.69}$$

Then, we write the simplified expressions for the saddle point equations: the overlaps read:

$$q=\int Dz\,\tanh\left(z\sqrt{\hat{q}}+\hat{S}\right)^2 \tag{2.70}$$

$$R=\int Dz\int D\tilde{z}\,\tanh\left(z\sqrt{\hat{q}}+\hat{S}\right)\tanh\left(z\frac{\hat{R}}{\sqrt{\hat{q}}}+\tilde{z}\sqrt{\hat{q}^\star-\frac{\hat{R}^2}{\hat{q}}}+\hat{q}^\star\right) \tag{2.71}$$

$$S=\int Dz\,\tanh\left(z\sqrt{\hat{q}}+\hat{S}\right) \tag{2.72}$$

and the conjugate parameters are given by:

$$\hat{q} = \frac{1}{\pi} \frac{\alpha}{\sqrt{1-q^2}} \int Dz \, \frac{H\left(z \frac{S}{q-S^2} \sqrt{\frac{1-q}{1+q}}\right)}{H\left(z\sqrt{\frac{q}{1+q}}\right)^2} \tag{2.73}$$

$$\begin{aligned} \hat{R} = {} & \frac{2\alpha}{\sqrt{1-q^\star}\sqrt{1-q}} \int Dw \int Dz \, \mathcal{G}\left(z\sqrt{\frac{q^\star}{1-q^\star}}\right) \\ & \times H\left(\sqrt{\frac{q^\star}{1-q^\star}} \frac{z\sqrt{q-\frac{R^2}{q^\star}} + w\frac{S-R}{\sqrt{q^\star}}}{\sqrt{q-\frac{R^2}{q^\star}-\frac{(S-R)^2}{1-q^\star}}}\right) \log H\left(\frac{w\sqrt{q-\frac{R^2}{q^\star}} + z\frac{R}{\sqrt{q^\star}}}{\sqrt{1-q}}\right) \end{aligned} \tag{2.74}$$

$$\hat{S} = \gamma + \sqrt{\frac{2}{\pi}} \frac{\alpha}{\sqrt{1-q}} \int Dz \, \mathcal{G}\left(z\sqrt{\frac{q-S^2}{1-q}}\right) \tag{2.75}$$

Note that q, S, $\hat{q}$ and $\hat{S}$ form a closed system of equations which can be solved independently. These equations are the same which would be obtained by studying the teacher student problem and coupling the students with the teacher via a distance constraint, except that the overlap S would play the role of the overlap between the student and the teacher in that case, rather than the pseudo-teacher. This fact, together with the facts that $\tilde{S} = R$ and that the entropy does not depend on R or $\hat{R}$, leads to the conclusion that the typical solution to the teacher student problems, which we chose as our pseudo-teachers, are indistinguishable from the teacher in everything except the generalization properties. It is also interesting to note that by setting R and $\hat{R}$ to 0, we recover the expressions for the classification problem, which also happens for the standard teacher-student problem.

Numerical evaluations of the Franz-Parisi potential show that, for all $\alpha > 0$, the local entropy is always negative in a neighborhood of $S = 1$, meaning that all typical solutions (including the teacher itself) are extensively isolated (see figure 2.7). The saddle point equations have two instability transitions: a global transition is observed when the value of the entropy, as computed at the saddle point becomes lower than that of the extremal point at $S = 1$. The second is characterized by a local transition, when the entropy curve changes concavity with respect to the overlap S. After this second point, the extremal point at $S = 1$ is the only solution to the saddle point equations.

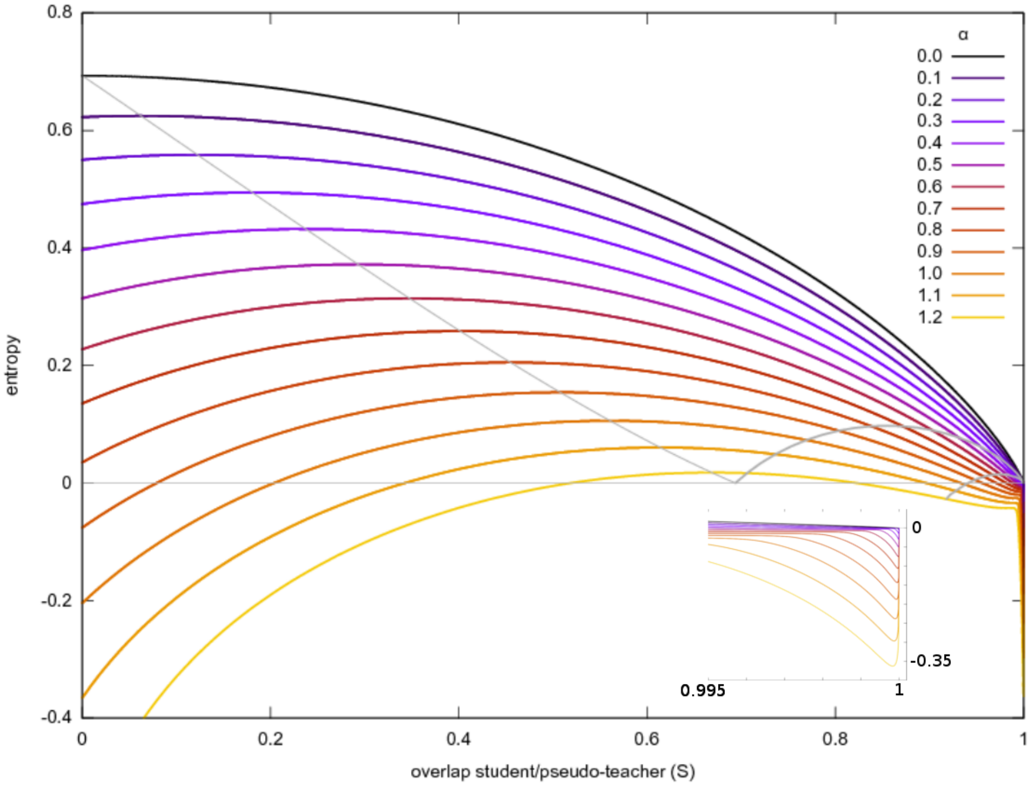


Fig. 2.7 **Franz-Parisi potential** for the binary Perceptron in the teacher-student scenario. As in the random classification case, the RS entropy becomes negative for a value $S < 1$ ($D > 0$) for all values of $\alpha < \alpha_D$. The black curve represents the case of unconstrained synapses, which provides an upper bound for local entropy evaluations. The four gray curves represent respectively: the zero entropy level, the typical entropy (obtained in correspondence of $\hat{D} = 0$, where the distance constraint is removed), the global instability transition (where the RS saddle point entropy estimate is lower than the extremal condition at $S = 1$, after $\alpha > \alpha_D$), and the local transition where the curve changes concavity with respect to S.

It might be interesting to compute also the average probability that the ensemble of students generalizes correctly with respect to the teacher, i.e. the probability that the average of the outputs of the students on a random new pattern has the same sign as that given by the teacher:

$$p_c = \left\langle \left\langle \left\langle \Theta \left(\left(\frac{1}{\sqrt{N}} \sum_i \xi_i^\star \right) \left\langle \text{sign} \left(\frac{1}{\sqrt{N}} \sum_i W_i \xi_i^\star \right) \right\rangle_W \right) \right\rangle_{\tilde{W}} \right\rangle_{\{\xi^\mu\}} \right\rangle_{\xi^\star} \tag{2.76}$$

where $\xi^\star$ is a test pattern, $\langle\cdot\rangle_{\xi^\star}$ is an average over i.i.d. test patterns, $\langle\langle\cdot\rangle_{\tilde{W}}\rangle_{\{\xi^\mu\}}$ is the average over training patterns and corresponding pseudo-teacher ensemble (see equation (2.40)), and $\langle\cdot\rangle_W$ is the average over the students which have learned the patterns $\{\xi^\mu\}$. We can write the latter as:

$$\left\langle \text{sign} \left(\frac{1}{\sqrt{N}} \sum_i W_i \xi_i^\star \right) \right\rangle_W = \tag{2.77}$$

$$\frac{\int \prod_i d\mu\left(W_i\right) \prod_\mu \Theta\left(\frac{1}{\sqrt{N}} \sum_i W_i \xi_i^\mu \frac{1}{\sqrt{N}} \sum_i \xi_i^\mu \right) e^{-\frac{\gamma}{2}\sum_i \left(W_i - \tilde{W}_i\right)^2} \text{sign}\left(\frac{1}{\sqrt{N}} \sum_i W_i \xi_i^\star \right)}{\int \prod_i d\mu\left(W_i\right) \prod_\mu \Theta\left(\frac{1}{\sqrt{N}} \sum_i W_i \xi_i^\mu \frac{1}{\sqrt{N}} \sum_i \xi_i^\mu \right) e^{-\frac{\gamma}{2}\sum_i \left(W_i - \tilde{W}_i\right)^2}}$$

where, as in the previous section, $\mu\left(W\right)$ is the measure over the W's, $d\mu\left(W\right) = \left(\delta\left(W+1\right) + \delta\left(W-1\right)\right) dW$, and $\tilde{W}$ is a pseudo-teacher. Note that the condition that $\prod_\mu \Theta\left(\frac{1}{\sqrt{N}} \sum_i \tilde{W}_i \xi_i^\mu \frac{1}{\sqrt{N}} \sum_i \xi_i^\mu \right) = 1$ is enforced in the $\langle\cdot\rangle_{\tilde{W}}$ operator, so it doesn't appear here.

We rewrite the average (equation 2.77) using the replica trick: we use the replica index 1 for the W's in the numerator, and we write the denominator as $n-1$ replicas, indexed from 2 to n, in the limit of $n \to 0$:

$$\begin{aligned} &\left\langle \text{sign} \left(\frac{1}{\sqrt{N}} \sum_i W_i \xi_i^\star \right) \right\rangle_W = \\ &\int \frac{dk d\hat{k}}{2\pi} e^{ik\hat{k}} \text{sign}\left(k\right) \lim_{n\to 0} \int \prod_{ia} d\mu\left(W_i^a\right) \prod_{a\mu} \Theta \left(\frac{1}{\sqrt{N}} \sum_i W_i^a \xi_i^\mu \frac{1}{\sqrt{N}} \sum_i \xi_i^\mu \right) \\ &\times \prod_a e^{-\frac{\gamma}{2}\sum_i \left(W_i^a - \tilde{W}_i\right)^2} \exp\left(-\frac{i\hat{k}}{\sqrt{N}} \sum_i W_i^1 \xi_i^\star \right) \end{aligned} \tag{2.78}$$

In order to perform the average over the training patterns $\{\xi^\mu\}$, we proceed as follows: we first expand equation 2.76 by substituting the arguments of the Θ function using Dirac deltas and expanding those with their integral representation:

$$
\begin{aligned}
p_c = & \int dx \frac{dy d\hat{y}}{2\pi} e^{iy\hat{y}} \Theta\left(xy\right) \qquad (2.79)\\
& \times \left\langle \delta\left(x - \frac{1}{\sqrt{N}}\sum_i \xi_i^\star\right) \left\langle \left\langle \exp\left(-i\hat{y}\left\langle \mathrm{sign}\left(\frac{1}{\sqrt{N}}\sum_i W_i \xi_i^\star\right)\right\rangle_W\right)\right\rangle_{\tilde{W}}\right\rangle_{\{\xi^\mu\}}\right\rangle_{\xi^\star}
\end{aligned}
$$

Then, we use the series representations of the exponential, $\exp\left(z\right) = \sum_{s=0}^{\infty} \frac{z^s}{s!}$, and perform the average over each individual term. For each value of s, we introduce a new replica index $l = 1, \ldots, s$, and we therefore need to compute (see equation 2.78):

$$
\begin{aligned}
& \left\langle \left\langle \prod_{l=1}^{s} \left\langle \mathrm{sign}\left(\frac{1}{\sqrt{N}}\sum_i W_i^l \xi_i^\star\right)\right\rangle_{W^l}\right\rangle_{\tilde{W}}\right\rangle_{\{\xi^\mu\}} = \qquad (2.80)\\
& \lim_{n,\tilde{n}\to 0} e^{-nsN\gamma} \int \prod_l \left(\frac{dk^l d\hat{k}^l}{2\pi} e^{ik^l\hat{k}^l} \mathrm{sign}\left(k^l\right)\right) \int \prod_{ic} d\mu\left(\tilde{W}_i^c\right)\\
& \times \int \prod_{ila} d\mu\left(W_i^{la}\right) \prod_{la} e^{-\gamma \sum_i W_i^{la}\tilde{W}_i^1} \prod_l \exp\left(-\frac{i\hat{k}^l}{\sqrt{N}}\sum_i W_i^{l1}\xi_i^\star\right)\\
& \times \left\langle \prod_{c\mu} \Theta\left(\frac{1}{\sqrt{N}}\sum_i \tilde{W}_i^c \xi_i^\mu \frac{1}{\sqrt{N}}\sum_i \xi_i^\mu\right) \prod_{la\mu} \Theta\left(\frac{1}{\sqrt{N}}\sum_i W_i^{la}\xi_i^\mu \frac{1}{\sqrt{N}}\sum_i \xi_i^\mu\right)\right\rangle_{\{\xi^\mu\}}
\end{aligned}
$$

The computation proceeds along the lines of the volume calculations presented above. The difference is that we now have sn replicas instead of n, and that there is the extra term for the replicas with $a = 1$. This extra term only affects the computation of the entropic part, which (with the RS Ansatz) now becomes:

$$
\begin{aligned}
G_S' = & \prod_i \int \prod_c d\mu\left(\tilde{W}_i^c\right) \int \prod_{la} d\mu\left(W_i^{la}\right) \exp\left(\hat{q}\left(\sum_l \sum_{a>b} W_i^{la}W_i^{lb} + \sum_{l>m}\sum_{ab} W_i^{la}W_i^{mb}\right) + \right.\\
& + \hat{\hat{q}} \sum_{c>d} \tilde{W}_i^c \tilde{W}_i^d + \hat{R}\sum_{la} W_i^{la} + \hat{\hat{R}}\sum_c \tilde{W}_i^c + \Delta\hat{S}\sum_{la} W_i^{la}\tilde{W}_i^1\\
& \left. + \hat{\hat{S}}\sum_{la} W_i^{la} \sum_c \tilde{W}_i^c - \frac{i\xi_i^\star}{\sqrt{N}}\sum_l \hat{k}^l W_i^{l1}\right) \qquad (2.81)
\end{aligned}
$$

We then take the limits $n \to 0$ and $\tilde{n} \to 0$, compute the integrals over $d\mu\left(W\right)$ explicitly and then expand for $N \to \infty$ up to the first order in N. The resulting expansion produces terms which happen to be related to $\frac{\partial}{\partial \hat{q}}\mathcal{G}_S$ and $\frac{\partial}{\partial \hat{R}}\mathcal{G}_S$. Therefore, we can use the saddle point equations and substitute the complicated integral expression with just R and q. The result is:

$$\begin{aligned}
G'_S =& \prod_i \left(1 - \frac{i}{\sqrt{N}}\xi^\star_i R \sum_l \hat{k}^l - \frac{1}{N} q \sum_{l>m} \hat{k}^l \hat{k}^m - \frac{1}{2N}\sum_l \left(\hat{k}^l\right)^2\right) \\
\simeq& \exp\left(-i\left(\frac{1}{\sqrt{N}}\sum_i \xi^\star_i\right) R \sum_l \hat{k}^l - \frac{q-R^2}{2}\left(\sum_l \hat{k}^l\right)^2 - \frac{1-q}{2}\sum_l \left(\hat{k}^l\right)^2\right) \\
=& \int Dz \prod_l \exp\left(-i\left(\frac{1}{\sqrt{N}}\sum_i \xi^\star_i\right) R\hat{k}^l - iz\sqrt{q-R^2}\hat{k}^l - \frac{1-q}{2}\left(\hat{k}^l\right)^2\right)
\end{aligned} \tag{2.82}$$

We can now go back to equation (2.80), having factorized everything with respect to the indices l:

$$\left\langle \mathrm{sign}\left(\frac{1}{\sqrt{N}}\sum_i W_i \xi^\star_i\right)^s\right\rangle_W = \tag{2.83}$$

$$\begin{aligned}
&\int Dz \left(\int \frac{dk d\hat{k}}{2\pi} e^{ik\hat{k}} \mathrm{sign}\left(k\right) \exp\left(-i\left(\frac{1}{\sqrt{N}}\sum_i \xi^\star_i\right) R\hat{k} - iz\sqrt{q-R^2}\hat{k} - \frac{1-q}{2}\hat{k}^2\right)\right)^s \\
&= \int Dz \left(\mathrm{erf}\left(\frac{R\left(\frac{1}{\sqrt{N}}\sum_i \xi^\star_i\right) + z\sqrt{q-R^2}}{\sqrt{2}\sqrt{1-q}}\right)\right)^s
\end{aligned}$$

Finally, we use this in the expression for p_c, obtaining:

$$\begin{aligned}
p_c =& \int dx \frac{dy d\hat{y}}{2\pi} e^{iy\hat{y}} \Theta\left(xy\right) \\
&\times \left\langle \delta\left(x - \frac{1}{\sqrt{N}}\sum_i \xi^\star_i\right) \int Dz \exp\left(-i\hat{y}\,\mathrm{erf}\left(\frac{R\left(\frac{1}{\sqrt{N}}\sum_i \xi^\star_i\right) + z\sqrt{q-R^2}}{\sqrt{2}\sqrt{1-q}}\right)\right)\right\rangle_{\xi^\star} \\
=& \int Dz \left\langle \Theta\left(\left(\frac{1}{\sqrt{N}}\sum_i \xi^\star_i\right)\left(R\left(\frac{1}{\sqrt{N}}\sum_i \xi^\star_i\right) + z\sqrt{q-R^2}\right)\right)\right\rangle_{\xi^\star} \\
=& 1 - \frac{1}{\pi}\arccos\left(\frac{R}{\sqrt{q}}\right)
\end{aligned} \tag{2.84}$$

where we got rid of the Dirac deltas, we used the fact that erf is an odd function, and we approximated the terms $\frac{1}{\sqrt{N}}\sum_i \xi_i^\star$ with a random variable distributed as a Gaussian centered in 0 with variance 1.

In the limit of $\hat{D} \to \infty$, we have $q = 1$ and $R = q^\star$, so we recover the usual expression for the generalization error of a single student:

$$p_c = 1 - \frac{1}{\pi} \arccos\left(q^\star\right) \tag{2.85}$$

In the limit of $\hat{D} \to 0$, instead, we have $q = R = q^\star$, so we recover the result for the Bayesian inference of standard teacher-student (which turns out to be the same as for the continuous case):

$$p_c = 1 - \frac{1}{\pi} \arccos\left(\sqrt{q^\star}\right) \tag{2.86}$$

Chapter 3

Learning with Discrete Synapses

One of the most striking properties of training artificial neural networks in a supervised scenario is how simple it can be to determine which is the best update for each synaptic connection W, in order to obtain a better classification performance. The first thing to do is to consider an appropriate loss function $\mathcal{L}$ (or energy function, in a Statistical Physics context), which compares the present output with the desired one. There are many possible alternatives, like the mean squared error or the cross-entropy loss functions [60], and many modifications of the loss can be introduced in order to enforce some desired properties in the network, such as a high sparsity degree (with an $L1$ norm) or small couplings (with an $L2$ norm). The recipe is simple: explicitate the functional dependency of the obtained output with respect to the synapse to be updated, and then evaluate the gradient of the chosen loss function. Given a small parameter ϵ, the correct update will simply given by:

$$W \leftarrow W - \epsilon \frac{\partial \mathcal{L}}{\partial W} \tag{3.1}$$

This procedure can be done fully in parallel with respect to all the synapses, exploiting the chain rule for evaluating the nested derivatives, as in the celebrated back-propagation algorithm [12, 61]. It is important to notice that, most often, the minimization of the loss function is not itself the real objective of the learning procedure and it should be regarded more as a tool, since the true goal is that of obtaining better classification performances. Moreover, this simple procedure in not guaranteed to reach the global minimum of the loss function.

In the last few decades, a big part of the machine learning community has been devoted to the optimization of these learning procedures, always looking for better heuristics for initializing optimally the learning algorithms and speeding them up, for enhancing the computational efficiency and (most of all) for obtaining better generalization performances when learning from real data. Notwithstanding all these developments, (stochastic) gradient descent has remained the main ingredient for learning (even though it might be "cooked" in different ways) [62].

In the prototypical case of the Perceptron with unbounded continuous synaptic weights, a specific choice for the loss function, $\mathcal{L} = \sum_{\mu} \Theta\left(-\sigma^{\mu} W \cdot \xi^{\mu}\right) W \cdot \xi^{\mu}$, produces the simple but effective Perceptron learning rule, able to find a perfect classifier (provided the problem is linearly separable) [63, 64] in a finite time. The algorithm works in a fully online regime, where the input patterns are presented sequentially: at each presentation of a pattern ξ one first computes the total input $\Delta = \sum_i W_i \xi_i$, and then simply increments the synaptic weights as in $W = W + \xi$ if $\Delta < 0$. Convergence in finite time can be proved in the case of unbounded synaptic weights [63] and for sign-constrained synaptic weights [65].

However, the algorithm doesn't achieve an extensive capacity when the synaptic weights can take only a finite number of states and in the extreme case, only two possible values (where the algorithm would become the so-called "clipped Perceptron" rule). Unfortunately, in the binary Perceptron any gradient descent based (or inspired) approach is apparently not applicable, and the problem of learning in this kind of architecture is known to be intractable in the worst-case [16]. As we have seen in the previous chapter 2, the classical Statistical physics description of the model shows that it is dominated in the large N limit by an exponential number of isolated local minima[17–20]. This situation is typical of a spin glass phase, which is common to many hard random optimization problems in which standard search strategies based on energy minimization are easily trapped [23–25, 66].

In the last decade, though, a series of algorithmic results [26–28] have shown that nearly optimal learning performance can be achieved also in the discrete setting. The striking performance of the simplified versions of these learning algorithms raises a few questions: in this chapter we try to characterize the features of the particular solutions these methods are able to find, and to what extent they correspond to the typical solutions described in the standard equilibrium Replica calculations.

3.1 BP in the Perceptron

The main building block for conceiving heuristic solvers in the discrete Perceptron problem is the Belief Propagation (BP) algorithm. Belief Propagation, or Sum-Product, is an iterative message-passing algorithm that can be used to describe a probability distribution, over a given instance of a CSP, within the Bethe-Peierls approximation [67, 68, 24]. BP stems directly from the Cavity Method, a statistical physics approach developed in the context of spin glasses and closely related to the Replica Method (see section **2.1**).

The algorithm is known to give exact results on tree graph; in the case of highly connected factor graphs (as the one associated to the Perceptron learning problem), instead, its approximation is based on a weak correlations assumption, closely linked to so-called Replica Symmetry (in replica analyses, see Chapter (2)): one assumes that, when an interaction is removed from the network, the nodes involved in that interaction become effectively independent.

At variance with the statistical mechanics results presented in the previous chapter, where an average over the quenched disorder is performed in the limit $N \to \infty$, here we are interested in single problem instances at finite N. Because of the so called self-averaging property, if N is large enough, we expect the quantities estimated by BP to match the typical case predictions, but even at finite N the results provide good approximations that can be exploited for algorithmic purposes.

The binary Perceptron learning problem has a natural formulation it terms of a CSP [26, 27][1], where the N synaptic variables $\{W_i\}_{i=1}^{N}$, with $W_i \in \{-1, +1\}$, have to satisfy $M = \alpha N$ constraints $\{\mathbb{X}_{\xi^\mu,\sigma^\mu}\}_{\mu=1}^{M}$, associated to a given set of input-output associations$\{\xi^\mu, \sigma^\mu\}_{\mu=1}^{M}$. We may define an energy function of the system simply as the number of violated constraints, namely:

$$H_0(x) = \sum_\mu E_\mu(W) = \sum_\mu \left(1 - \mathbb{X}_{\xi^\mu,\sigma^\mu}(W)\right) \tag{3.2}$$

The problem is fully-connected, in the sense that all the variables participate in all the constraints. Thus, if we associate a bipartite *Factor Graph* to the problem, we can connect with an edge each variable node $i = 1, ..., N$ (one for each synapse) to all the factor nodes $\mu = 1, ..., M$. This kind of graphical model representation can be very helpful in understanding the basic dynamics of message passing methods such as BP (see figure 3.1).

The BP equations are a set of coupled nonlinear equations for the *cavity messages* $\{u_{i\to\mu}, \hat{u}_{\mu\to i}\}$, which run along the edges of the Factor Graph:

$$u_{i\to\mu}(W_i) \propto \prod_{\nu\in\partial i\setminus\mu} \hat{u}_{\nu\to i}(W_i) \tag{3.3}$$

$$\hat{u}_{\mu\to i}(W_i) \propto \sum_{\{W_j\}_{j\neq i}} e^{-\beta E_\mu(W)} \prod_{j\setminus i} u_{j\to\mu}(W_j) \tag{3.4}$$

Each cavity message represents a marginal probability distribution in the absence of the constraint (or the variable) towards which the message is directed. The information we are looking for can be obtained from the fixed point $\left\{u^\star_{i\to\mu}, \hat{u}^\star_{\mu\to i}\right\}$ of these equations. Solving the equations by iteration produces an efficient, fully distributed technique (which gives origin to the name "message-passing" method) with a typical computational complexity scaling roughly as $\mathcal{O}\left(N^2 \log(N)\right)$ (almost linearly in the size of the input times an extensive number of patterns). Once the fixed point is reached, it is possible to compute local joint marginal probabilities and all the thermodynamic potentials, such as the average energy, the entropy and the free energy of the system, in terms of purely local contributions from variables, edges and factor nodes. The single variable marginals, for example, can be simply computed as $u_i(W_i) \propto \prod_\nu \hat{u}_{\nu\to i}(W_i)$. In the case of Ising-like systems (with binary ±1 variables) the messages associated to the possible values of W_i can be summarized in a single number, usually a cavity magnetization $m_{i\to\mu} = u_{i\to\mu}(W_i = +1) - u_{\mu\to i}(W_i = -1)$ (and analogous for the other set of messages). Moreover, it is always possible to set $\forall\mu : \sigma^\mu = 1$ without loss of generality, by means of the gauge transformation $\xi^\mu_i \to \sigma^\mu \xi^\mu_i$. With these simplifications, in the binary

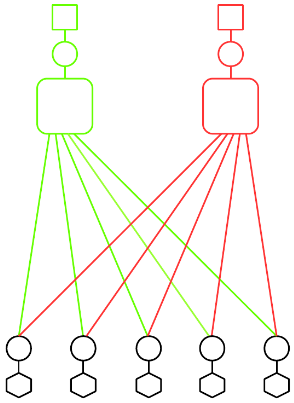


Fig. 3.1 **BP factor graph scheme** for a perceptron with $N = 5$ variables and trained on 2 patterns (green and red). The variable nodes are represented as circles, the interaction by other geometrical figures. The hexagons at the bottom represent external fields (priors) on the synaptic variables, the large squares with rounded corners represent Perceptron-like nodes, the small squares at the top represent external fields enforcing the desired output of the machine. The synaptic variables W_j are at the bottom (black circles), while the red and green variables are auxiliary and represent the output of the perceptron for the two patterns.

Perceptron the explicit BP equations for the cavity magnetizations read:

$$m_{i\to\mu} = \tanh\left(\sum_{\nu\neq\mu}\tanh^{-1}\left(\hat{m}_{\nu\to i}\right)\right) \tag{3.5}$$

$$\hat{m}_{\mu\to i} = \frac{\sum_{s=-\xi_i}^{N-1} D_{\mu\to i}\left(s\right) - \sum_{s=\xi_i}^{N-1} D_{\mu\to i}\left(s\right)}{\sum_{s=-\xi_i}^{N-1} D_{\mu\to i}\left(s\right) + \sum_{s=\xi_i}^{N-1} D_{\mu\to i}\left(s\right)}, \tag{3.6}$$

where:

$$D_{\mu\to i}\left(s\right) = \sum_{\{W_j\}_{j\neq i}} \delta\left(s, \sum_j W_j\xi_j\right)\prod_{j\neq i}\frac{\left(1+W_j m_{j\to\mu}\right)}{2} \tag{3.7}$$

is the convolution of the all cavity messages $m_{j\to\mu}$ impinging on the pattern node μ, except for $m_{i\to\mu}$. The computation of the convolutions can be done efficiently, limiting the complexity to an order $O\left(N^2\right)$.

In the case of large networks, $N \gg 1$, and of an extensive number of patterns, it is natural to apply the Central Limit Theorem and adopt a Gaussian approximation $\tilde{D}_{\mu\to i}\left(s\right) = \frac{1}{b_{\mu\to i}}G\left(\frac{s-a_{\mu\to i}}{b_{\mu\to i}}\right)$ for the distribution $D_{\mu\to i}\left(s\right)$ (where $G\left(s\right)$ denotes the normal distribution) [26]. Thus, it is sufficient to compute the mean $a_{\mu\to i}$ and variance $b^2_{\mu\to i}$ of the approximated distribution:

$$a_{\mu\to i} = \sum_{j\neq i} \xi_j^\mu m_{j\to\mu} \tag{3.8}$$

$$b^2_{\mu\to i} = \sum_{j\neq i} \left(1 - m^2_{j\to\mu}\right) \tag{3.9}$$

By doing so, equation (3.6) simply becomes:

$$\hat{m}_{\mu\to i} = \xi_i \, g\left(a_{\mu\to i}, b_{\mu\to i}\right) \tag{3.10}$$

where, using the error function $H\left(x\right) = \frac{1}{2}\text{erfc}\left(\frac{x}{\sqrt{2}}\right)$, we define:

$$g\left(a,b\right) = \frac{H\left(\frac{a-1}{b}\right) - H\left(\frac{a+1}{b}\right)}{H\left(\frac{a-1}{b}\right) + H\left(\frac{a+1}{b}\right)}. \tag{3.11}$$

At zero temperature, the free-entropy F of the system can be easily obtained, in the Gaussian approximation, as:

$$\begin{aligned} F_{\text{perc}} =& \sum_\mu \log\left(H\left(\frac{a_\mu}{b_\mu}\right)\right) - \sum_{i,\mu} \log\left(1 + m_{i\to\mu}\hat{m}_{\mu\to i}\right) \\ &+ \sum_i \log\left[\prod_\mu \left(1 + \hat{m}_{\mu\to i}\right) + \prod_\mu \left(1 - \hat{m}_{\mu\to i}\right)\right] \end{aligned} \tag{3.12}$$

while the normalized logarithm of the total number of solutions for a given instance is given by the entropy:

$$\begin{aligned} \mathcal{S}_{\text{perc}} =& \sum_\mu \log\left(H\left(\frac{a_\mu}{b_\mu}\right)\right) + \\ & - \sum_{i,\mu} \left[\frac{1+m_i}{2} \log\left(\frac{1+m_{i\to\mu}}{2}\right) + \frac{1-m_i}{2} \log\left(\frac{1-m_{i\to\mu}}{2}\right)\right] \\ & + \left(M-1\right)\left[\frac{1+m_i}{2} \log\left(\frac{1+m_i}{2}\right) + \frac{1-m_i}{2} \log\left(\frac{1-m_i}{2}\right)\right] \end{aligned} \tag{3.13}$$

Consistently with the theoretical predictions, BP equations always converge for $\alpha < \alpha_c$, and the entropy decreases monotonically with α, vanishing at the critical threshold $\alpha_c \sim 0.833$ (provided N is high enough).

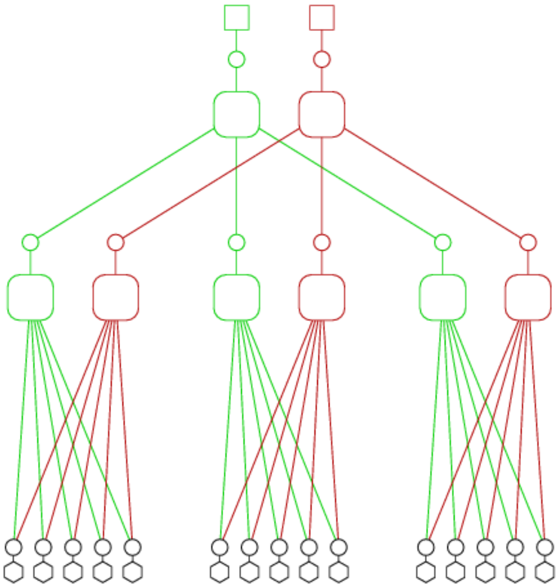


Fig. 3.2 **BP factor graph scheme** for a committee machine with $N = 15$ variables, $K = 3$ units in the second layer, trained on 2 patterns. The two patterns are distinguished by different colors. The graph can represent a fully-connected committee machine if the patterns are the same for all first-layer units, or a tree-like one if they are different. The variable nodes are represented as circles, the interaction by other geometrical figures. The hexagons at the bottom represent external fields on the synaptic variables, the large squares with rounded corners represent Perceptron-like nodes, the small squares at the top represent external fields enforcing the desired output of the machine. The synaptic variables W_j^k are at the bottom (black circles), while the rest of the variables are auxiliary and represent the output of each unit for a given pattern.

With the same factor nodes we can also represent the factor graph associated to the simplest two-layer binary neural network, the committee machine, as it can be seen in figure 3.2. In this case the weights in the second layer can be fixed all to 1, by exploiting a symmetry of the model, and therefore the variable nodes are associated only to the weights in the first layer.

3.2 Effective learning algorithms

As stated in the introduction of this chapter, until a few years ago only a handful of heuristic algorithms were believed — based on numerical evidence — to be able to solve (in sub-exponential time) the classification problem in the binary Perceptron, and achieve an extensive capacity in the large N limit: reinforced Belief Propagation (R-BP) [26], reinforced Max-Sum [29] (R-MS), SBPI [27] and CP+R [28]. In the classification case, they all achieve very high capacities, between $\alpha \simeq 0.69$ and $\alpha \simeq 0.75$. The same qualitative scenario holds in the generalization case, where all these algorithms perform well except in a finite ("hard") window $1 \lesssim \alpha \lesssim 1.5$ around the transition at $\alpha_{TS} = 1.24$ [28]. These algorithms also share the property of being local and distributed, and have typical solving times which scale almost linearly with the size of the input. SBPI and CP+R additionally have extremely simple requirements (only employing finite discrete quantities and simple, local and on-line update schemes), making them appealing for practical purposes and reasonably plausible candidates for biological implementations. For the sake of completeness, in the following we provide a brief description of these heuristics solvers for the binary Perceptron.

3.2.1 The R-BP algorithm

In order to turn BP into a solver, one needs to introduce some heuristic ingredient able to induce a collapse of the algorithm onto a single configuration [26], rather than the usual fixed points describing the probability distribution of ensembles of solutions of the problem. The idea is to add an extra term into Eqs. (3.6) and (3.5), enforcing $m_i = \pm 1$ at the fixed point, and thus extract the solution $W_i = \mathrm{sign}\,(m_i)$.

Inspired by the decimation technique [69], where the an increasing number of variables are progressively fixed (with an infinitely strong field), the R-BP algorithm introduces a "reinforcement" term, proportional to the field associated to the marginal magnetization of the variable, as computed in the previous iteration step:

$$m_i^{t+1} = \tanh\left(\sum_{\nu} \tanh^{-1}\left(\hat{m}^t_{\nu\to i}\right) + \rho \tanh^{-1}\left(m_i^t\right)\right). \tag{3.14}$$

This procedure induces a smooth decimation, where the less polarized variables receive an external field as well, but with a finite intensity, proportional to their polarization. In order to improve convergence, this term can be introduced stochastically (with increasing probability as BP progresses).

This method seems to achieve an algorithmic capacity of at least $\alpha \simeq 0.74$. Moreover, the scheme seems to be quite general, leading to very good results in a variety of different problems, even when standard BP would not converge or would provide a very poor approximation (see e.g. [70]).

3.2.2 The R-MS algorithm

This algorithm is the analogous of R-BP, the difference being that Max-Sum (MS) is chosen as the underlying algorithm rather then BP [29]. The MS equations can be derived as a particular zero-temperature limit of BP: in this process they normally become computationally simpler, since they require only sum and max operations, in contrast with the hyperbolic functions included in the BP equations. MS can also be seen as a heuristic extension (to loopy graphs) of the dynamic programming approach.

The heuristic reinforcement term introduced in R-MS acts very similarly to the previous case (R-BP), but here it is also needed in order to ensure convergence of the MS algorithm (which would otherwise be problematic, since the factor graph associated to the Perceptron is far from acyclic and the ground-state of the problem is degenerate). The addition of a reinforcement introduces a slight dependence on the initialization, which can be controlled by reducing ρ (at the cost of an increased computational time).

The resulting characteristics of R-MS are very similar to those of BP and extensive numerical tests give an equivalent capacity of about $\alpha \simeq 0.75$.

3.2.3 The SBPI algorithm

Derived as a crude simplification of the R-BP algorithm [27], in an attempt to remove all features which would be unrealistic from a biological point of view, SBPI is an on-line algorithm, like the Perceptron rule. All the information needed for the update is contained in the same $\Delta = \sum_i W_i \xi_i$ computed above, but in this case it is not the synapse itself to be updated, but an integer (and odd) internal state variable h_i, that takes values in a finite range ($|h_i| \leq K - 1$) and whose sign determines the synaptic value $W_i = \mathrm{sign}\,(h_i)$.

The update rule is familiar: $h_i \to h_i + 2\xi_i^\mu \sigma^\mu$, and it is applied depending on the value of Δ: if $\Delta > 1$, then nothing is done; if $\Delta = 1$, then only the synapses for which $W_i = \xi_i^\mu \sigma^\mu$, are updated with probability p_s; if $\Delta \leq 0$, then all synapses are updated. The parameter $p_s \in [0, 1]$ makes the algorithm stochastic, and its optimal value is usually found between $0.3 - 0.4$.

Because of the simple dependency of the synaptic values only through the sign of the hidden variables, the algorithm inherits the property of finding solutions which are more resistant against accidental changes in the internal states, in the presence of noise and degradation. Rather surprisingly, the measured critical capacity of this algorithm, $\alpha \simeq 0.69$, is only slightly reduced with respect to the original R-BP algorithm.

3.2.4 The CP+R algorithm

This last algorithm was derived as an utter simplification of SBPI [28]. The main observation that led to its conception is the following: in the generalization scenario, the effect of the update in the $\Delta = 1$ case (the near-threshold type of event which distinguishes the SBPI algorithm from the Clipped Perceptron algorithm) is on average equivalent to applying an unspecific reinforcement to all the internal states $h_i \to h_i + 2\text{sign}\left(h_i\right)$ with probability p_r (in contrast with the reinforcements described above). This increment needs to be applied at a calibrated rate, in order to keep most of the hidden states near the boundary (at 0) and allow the associated synapses to change sign during the learning procedure.

The equivalence with SBPI only holds in the context of the on-line generalization task, but CP+R can be adapted, with some minor modifications, also to the classification context. The simplifications introduced in CP+R greatly reduce the overhead associated with the complex computations required by the faster algorithms, while maintaining a scaling behavior of order $\mathcal{O}\left(N \log N\right)$. With some fine-tuning, the CP+R algorithm reaches the same capacity of SBPI: $\alpha \simeq 0.69$.

3.3 Extension to multi-layer networks

It is easy to imagine that, if the restriction to discrete synapses can turn the learning problem in the simplest neural network architecture into a hard task, finding an effective approach in multi-layer discrete networks can be a very hard problem. The main issue is due to the fact that the message-passing algorithms, which inspired the effective heuristics for the Perceptron, in this scenario suffer from inherent convergence problems[29, 26]. An easy to spot source for these problems is the permutation symmetry: when single Perceptrons are stacked and connected to obtain a more complex architecture, if the Perceptrons in the uppers layers are initialized in the same unbiased way, during the message-passing iterations they will exchange the same exact messages with the rest of the network and will not be able to differentiate. This kind of symmetry is disruptive for the classification performance, since the network becomes completely redundant.

A seemingly reasonable solution is to apply to these variables a small random external field, which could potentially play a symmetry-breaking role. This heuristic seems to help, but even in the case of two-layer binary networks (committee machines), the results obtained with BP are at least questionable: it seems that a growing number of distinct BP fixed points (not imputable only to the permutation symmetry) may be found, and using the information obtained through the message-passing procedure for finding single solutions, as in the R-BP algorithm, requires a very delicate fine-tuning of the reinforcement rate. In fact, the extension of BP to multi-layer networks unfortunately introduces all sorts of numerical stability problems, due to the “loopy” nature of the factor graph and to the presence of long-range correlations which are neglected in the BP approximation. For example, the gaussian approximation in equation (3.10) and, in some cases, even the finite machine precision can cause the message-passing procedure to go off the rails.

All these numerical issues motivated the search of a simplified heuristic [1], inspired by the efficient ones described above. It is indeed possible to heuristically extend the CP+R algorithm to the case of a multi-layer classifier, obtained by stacking two layers of fully-connected committee machines, with L possible output labels. Because of a symmetry associated to any simultaneous

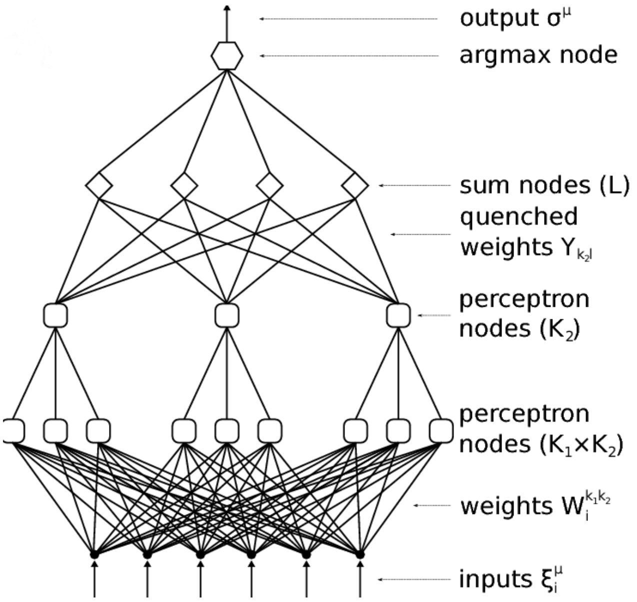


Fig. 3.3 **Multi-layer architecture** considered in the extension of the CP+R algorithm. The neural network can be seen as a “committee of committees”, with an argmax at end, in order to allow for a multi-label classification.

change in the sign of a synapse in the top layers and in all the synapses directly below it, it is sufficient to learn only the synapses in the first layer.

More specifically, the architecture (in figure 3.3) consists of an array of K_2 committee machines, each comprising K_1 hidden units. The K_2 outputs are sent to L summation nodes (each one specifically associated to a possible label), and the maximum one is chosen as the predicted output of the network. The non-linear function represented by the network can be written as:

$$\phi\left(\xi\right) = \mathrm{argmax}_{l\in\{1,\ldots,L\}} \left(\sum_{k_2=1}^{K_2} Y_{k_2 l}\,\mathrm{sign}\left(\sum_{k_1=1}^{K_1} \tau\left(W^{k_1 k_2}, \xi_i\right)\right)\right) \tag{3.15}$$

where $Y_{k_2 l} \in \{-1, 1\}$ are random quenched binary weights, defining mutually perpendicular directions associated *a priori* to the labels, and $W^{k_1 k_2} \in \{-1, 1\}^N$ are the synaptic weights learned by the algorithm.

The unsupervised reinforcement term, characteristic of CP+R, can be left unaltered from the single-layer version of the algorithm. Instead, it is necessary to design a scheme for back-propagating the observed errors ($\phi\left(\xi\right) \neq \sigma$) down to the synapses $W^{k_1 k_2} \in \{-1, 1\}^N$. The main idea is that allowing for too many changes of the synapses in the first layer can destabilize the learning procedure quite easily. A possible cure of this problem is the following: first, one needs to find all the committee machines which contributed to the error, i.e. all those for which sign $\left(\sum_{k_1=1}^{K_1}\tau\left(W^{k_1k_2},\xi_i\right)\right)\neq Y_{k_2\sigma}$. Then, for each of these, the signal is further propagated only in the branch corresponding to the hidden unit whose mistake is easiest to fix, i.e. for which $Y_{k_2\sigma}\sum_{i=1}^{N}W_i^{k_1k_2}\,\xi_i$ is less negative. In these branches, the update of the hidden states associated to the synapses simply follows the standard CP+R rule. The generalization performance can be highly improved if a "robustness" requirement is added, such that an error signal is emitted also when $\phi\left(\xi\right)=\sigma$, but the gap between the two maximum outputs of the L committees is smaller then some threshold r.

This extension allows us to test the algorithm on real world data, for example on the MNIST database benchmark [71], which consists of $7\cdot10^4$ gray-scale images of hand-written digits ($L=10$). The last 10^4 images are reserved for assessing the generalization performance of the learned network. We observed that it is very easy to learn perfectly the training set, and that very good generalization errors can be reached despite the binary nature of the synapses, without any specialization of the architecture for this particular dataset. Moreover, the algorithm seems to completely avoid over-fitting, even when the considered networks are very large. The smallest network which was found to be able to achieve zero training error had $K_1=11$ and $K_2=30$, with $r=0$, reaching a generalization error around 2.4%. Very large networks can achieve much better generalization error rates, e.g. 1.25% with $K_1=81$, $K_2=200$, $r=120$.

3.4 Bayesian approach

As we said before, it is nearly impossible to employ effectively a pure R-BP approach in a multi-layer context, mostly due to convergence problems during the message-passing iterations. But since we are now able to find a particular solution of the problem with the extended CP+R heuristic, an interesting question is whether this information could be used for bootstrapping the BP algorithm into an easier region, where it could be more effective. Geometrically, the idea would be similar to the one in the Franz-Parisi potential: once a configuration is “planted” (via the addition of external fields on the synaptic variables), one can look at its neighborhood and obtain an estimate of the single variable marginal magnetizations representing the ensemble of solution at a given distance from the central configuration (which can be seen as a “state”).

From a Bayesian point of view, in the generalization task this method could be used to estimate a “local” bayesian prediction, as an alternative to the maximum likelihood prediction given by the single solution found by the heuristic solver. The Bayesian output is obtained by first building a factor graph containing only the factors and the auxiliary variables associated to the pattern to be tested, with an unconstrained output. Then, on each synaptic variable node an external field is applied, with an intensity given by its marginal magnetization ($h_i^{\text{ext}} = \tanh^{-1}(m_i)$) measured in the “state” found by the bootstrapped BP on the training set. The BP convergence in the testing factor graph is trivial, as in the large N limit the network simply computes a Gaussian propagation, where the mean output of each Perceptron is obtained by first evaluating mean and variance of the distribution of its pre-activation (depending on the average inputs $\langle x_i \rangle$):

$$\mu = \sum_i m_i \langle x_i \rangle \tag{3.16}$$

$$\sigma^2 = \sum_i \left(1 - m_i^2 \langle x_i \rangle^2\right) \tag{3.17}$$

and then by applying the special transfer function $\text{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right) = 2H\left(-\frac{\mu}{\sigma}\right) - 1$. We note that, a part from the Gaussian approximation, also the correlations between the outputs/inputs are completely neglected in this formalism, because of the implicit assumption in the definition of the BP factor graph. If we also ignore the off-diagonal correlation terms in the argmax evaluation, we can simply pick the maximum of the distribution on the output labels as the maximum likelihood prediction.

We can therefore make a direct comparison, in the generalization performance, between three possible predictions:

1. The output of a single solution, which in this formalism is obtained in the limiting case of an infinite external field in the direction of the solution found by the solver.

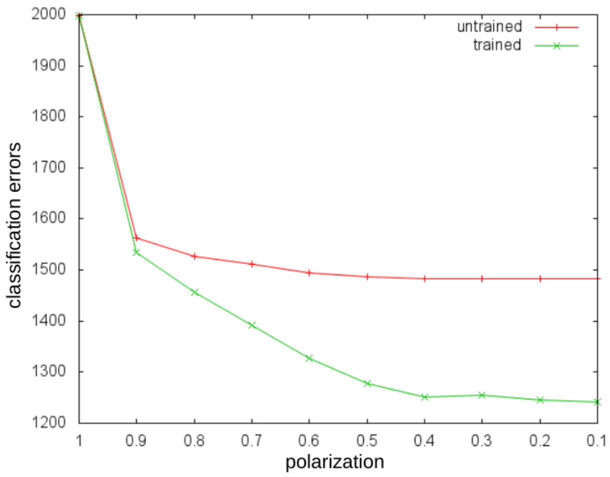


Fig. 3.4 **Generalization error** measured as a function of the polarization of the planted field in the BP bootstrapping phase, in the three above considered scenarios, on the MNIST test set (10^4 images) after learning only from 10^4 images of the training set. While the single solution performance can be seen from the score at a magnetization $m = 1$, the red and the green curves show the progressive improvement registered in the generalization task as the initial polarization was decreased.

2. The local Bayesian prediction (labeled as “trained” in figure 3.4), with the marginal magnetizations found by converging BP on the training set factor graph, after planting the solution with a finite field.

3. The output obtained as a flat average over all the configurations (not only the solutions) in a close neighborhood of a solution (labeled as “untrained”). In this case the marginal magnetizations, to be used as external fields in the testing phase, are simply obtained by rescaling uniformly the binary solution weights to a value $|m_i| < 1$, and thus represent a depolarized solution.

The efficacy of the Bayesian procedure can be particularly seen when the solution found by the solver is clearly in an over-fitting regime (for example by training a network on a small subset of the patterns usually present in the MNIST training set). In this case the local Bayesian prediction’s performance distances both the single solution and the flat average ones. However, even the trivial operation of depolarizing the solution marginals is quite effective in reducing the generalization error (see figure 3.4).

Unfortunately, the gap between these method closes as one chooses a better generalizing solution as the planted configuration. In this case, the improvement of the Bayesian approach with respect to the single solution prediction is marginal, and, surprisingly, seems to be almost equivalent (except for the very long computational times required for obtaining convergence of BP) to the one observed in the case of a simple depolarization of the solution. Moreover, an increase in the generalization error can be observed when the radius of the state explored with BP (or in the depolarization) becomes larger than a small threshold.

3.5 The algorithmic solutions

It would be most impressive if any algorithm was able to find solutions which are isolated points in an exponentially large phase space. This "golf course" scenario, usually characterizing the so-called *frozen phase*, is known to prevent local search algorithms and standard optimization techniques, e.g. Simulated Annealing, from working. The heuristic modifications of massage-passing algorithms, like BP or MS, and even Survey Propagation [32] are usually failing as well, in a situation like this [66]. For instance in the K-SAT problem, all the algorithms are able to find solutions when these are all part of a big cluster ($\alpha < \alpha_D$), and some are able to find them even when the cluster breaks apart into exponentially many smaller cluster ($\alpha_D < \alpha < \alpha_{cond}$). But as the remaining solutions "condense" into a sub-dominant number of clusters and, eventually, only the isolated ("frozen") solutions are left, all these algorithms stop working [38].

From the equilibrium analysis we presented in chapter 2, we know that the Percepton's typical solutions are characterized (in the 1RSB analysis) by an internal overlap $q_1 = 1$, and are always extensively isolated, for any value of α, like the frozen solutions of K-SAT. Some numerical result are in agreement with this picture: Simulated Annealing's solving time is in fact known to diverge, also for "easy" instances, when the number of synapses N is large. Nevertheless, the heuristics presented above (in section 3.2) are almost able to saturate the SAT/UNSAT threshold α_C, despite the adverse landscape.

We need to characterize the type of solutions found by effective algorithms, in order to see if they possess some special property that makes them “dynamically” attractive for learning processes. We can easily check one of the defining property of typical solutions, the geometrical isolation. In figure 3.5, we can see the fraction of isolated solution found by SBPI, as a function of α and of the size of the network N. The results are quite counter-intuitive:

- When the considered pattern load is not high enough, $\alpha \sim 0.5$, almost all the solutions found by this algorithm are not isolated, differently from what we expected from the equilibrium analysis.
- Near the critical algorithmic threshold $(0.65 < \alpha_U < 0.7)$ the fraction of isolated solutions increases, but it also vanishes as N is increased.

A simple justification for the detection of connected solutions would come from finite size effects, but the phenomenon becomes more accentuated in large instances. It seems that the effective algorithms systematically move towards dense regions of the phase space which might be invisible to the standard 1RSB analysis. The increment in the fraction of isolated solutions as α is increased also suggests that the “clusters” found by the algorithms might fragment before disappearing completely at an $\alpha < \alpha_C$. An important question is whether these clusters can somehow be seen as well defined states (in the Statistical physics sense).

We can also check the isolation property in the teacher-student scenario. Intuitively, one could think that in this case the teacher itself might be a special solution, and might belong to these connected regions of solutions. However, this is not the case: the teacher is predicted to be extensively isolated, just like typical solutions are, and in this case this fact can also be checked numerically (as we don’t have to find it first, which would not be possible since it is frozen). Yet, we can also see that the same type of connected structures are reached by the working heuristic solvers also in this scenario.

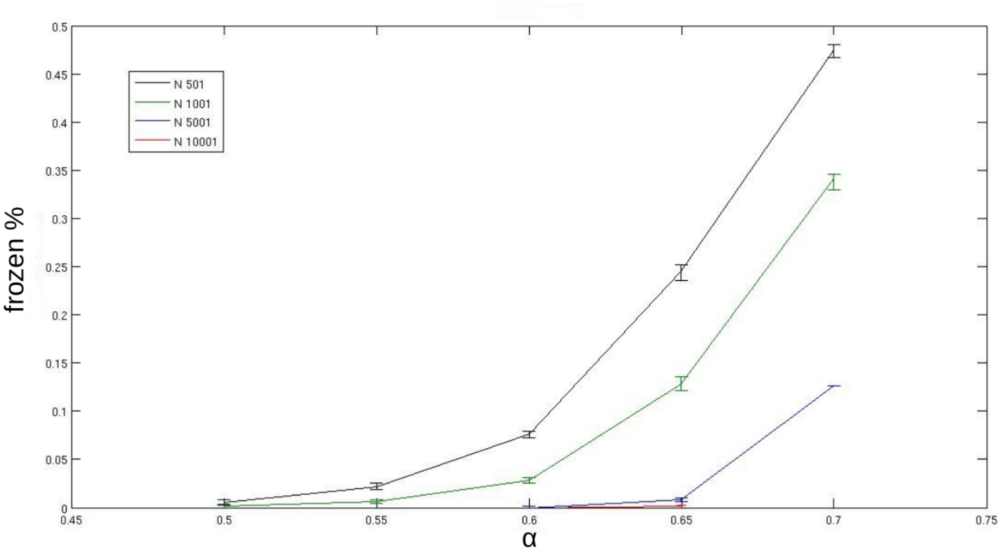


Fig. 3.5 **Fraction of isolated solutions** found by the heuristic algorithms (SBPI) in the random classification case of the Perceptron. As the size of the Perceptron N is increased, the fraction of isolated ("frozen") solutions shrinks, while it becomes larger as more constraints are added o the problem (and α is increased).

3.6 Whitening: Frozen vs Unfrozen

A possible approach for detecting the presence of well defined states in the space of solutions is the so-called whitening procedure, developed in the context of sparse random graphs.

The name for this method origins in a different CSP, the coloring problem, where one has to assign a color, chosen from a family of q colors, to each of the nodes of a given graph, with a constraint that prohibits equally colored adjacent nodes. The problem gets interesting when the graph is colorable in many diffent ways and a rich landscape of geometrical structures can be found in the space of solutions of the problem [72].

The whitening procedure is defined as a "reversed" coloring process: once a legal coloring is produced, one looks for the nodes whose color can be changed without violating any constraint, assigning them with the white color (which denotes a "free" state: these nodes can now be variably assigned a color which is most convenient for freeing more variables). The procedure continues as a cascade of white assignments, until one possibly finds the "core" node assignments that cannot be whitened: in this way one can highlight those nodes that are essential in the legal coloring initially produced.

A slightly more complicated version of this procedure, closely related to message-passing algorithms (as BP and TAP), is defined as the directional coloring process: in this case the white state can be assigned to the cavity messages (instead of the variable states, making it harder for the process to spread into the entire factor graph). When the directional whitening can no longer proceed, an extremal directional whitening configuration is found. The variables which cannot be assigned a white color are defined to be "frozen", while the others are said to be "unfrozen" (see also section 1.2).

At this point we can have different possible scenarios:

- If the model is below the dynamical phase transition, at $\alpha < \alpha_D$, then only one trivial whitening is possible: all the graph becomes white.
- After the transition, at $\alpha_D < \alpha < \alpha_C$, there is an exponentially large number of extremal directional whitenings. Their number $\exp(N\Sigma(\alpha))$ is in one to one correspondence with the number of states (well separated clusters of solutions, detectable in a 1RSB analysis) in the phase space of the problem. When different legal coloring assignments end up in the same extremal directional whitening, it means that the two solutions belong to the same cluster.
- above α_C, typically no legal coloring can be produced, thus the whitening process is ill-defined.

It can be proven that the local equilibrium condition for the extremality of a directional whitening is equivalent to the BP or the TAP equations in this context.

Thus, we can try to apply a similar procedure also in the Perceptron model: in this case a white state will be assigned to a synapse whose value can be chosen arbitrarily without making any classification errors. The directional whitening, moreover, can be studied in the following way:

1. A solution is found with one of the effective heuristics described in section 3.2.

2. The BP algorithm is initialized in correspondence of the solution (by running it to convergence in the presence of strong external fields, on the variable nodes, in the direction of the solution).

3. The external fields are removed.

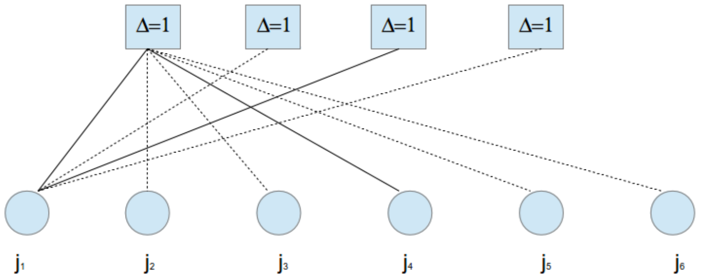
4. The BP algorithm is run again and convergence is reached.

Now, there are two possible scenarios for point 4: the BP marginals can either remain fixed (as set by the external fields), or they can be driven away, reaching the closest BP fixed point. In the first case the solution is completely frozen and represents a point-like state (with $q_1 = 1$). In the second case one ends up in the BP fixed point relative to the state the solution belongs to (with $q_1 < 1$).

Unfortunately, due to the fully-connected nature of the problem, this approach is not able to give a better characterization of the space of solutions in the Perceptron. In fact, we couldn't find any states except the point-like ones (when the solver ends up in an isolated solution, with $q_1 = 1$ as predicted in [17]), and the so-called Replica Symmetric BP fixed point, which is reached from all the unblocked solutions (solutions which are found not to be isolated at $\mathcal{O}(1)$ distances) or any other random initialization of the BP messages. It is important to stress that the point-like states appear to have a vanishing basin of attraction for BP (or TAP), since every small perturbation of the messages in the initialization (point 2. above) allows the message-passing algorithms to flow away towards the RS point.

In order to explain this trivial behavior we can consider the factor graph associated to a given Perceptron instance: since we are trying to study the directional whitening process, we can simplify the graph, excluding those factor nodes that correspond to patterns with a stability $\Delta^\mu = W \cdot \xi^\mu$ strictly higher than 1 (when N is odd the stabilities can only take odd integer values $\{..., -3, -1, 1, 3, ...\}$), i.e. patterns that are robust to any one flip in the synapses. In fact, if we imagine the BP algorithm, at convergence these factors are already sending white (flat) messages to all the variables.

Therefore, consider the simplified factor graph where only the unstable patterns are present: we can connect all the variables with the factor nodes with two kind of lines, dashed if a flip in the value of the variable will increase the stability (in this case the pattern is sending back a white message), and



.

Fig. 3.6 **Factors constraining the value of the synapses** in the simplified factor graph associated to the Perceptron problem, where only the patterns with a stability at the threshold ($\Delta^{\mu} = 1$) are considered. The dashed lines link the synapses with the patterns that would gain in stability from a flip in the value of the synapses. The solid lines, instead, link the variables with the patterns whose stability would decrease, violating the constraint.

solid if a flip decreases the stability, implicating an error in the classification of that pattern (see figure 3.6). These solid lines are thus representing the constraints actually "felt" by each synapse.

If we are in a situation where the solution is blocked (i.e. isolated), it means that all the synapses receive at least one solid line from the unstable patterns. If we consider a directional whitening process starting from one of these solid lines, we can see that the messages sent to the other synapses by the associated pattern wouldn't become dashed and be "freed", since the stability Δ^{μ} would be utterly decreased by a wrong assignment for the variable we started the whitening from. Therefore the directional whitening process would stop immediately: this kind of solution is completely frozen.

On the other hand, we have the case where one of the synapses only receives dashed lines from the unstable patterns, implying that it is unblocked, since a flip would still produce a configuration with stable, correctly classified patterns. If the messages sent by this variable to the unstable patterns are set to white in a directional whitening process, automatically all the stabilities are increased to 3, and the rest of the variables are released from their constraints, i.e. all the solid lines become dashed. Therefore the whitening spreads to the entire graph.

An exemplary case is represented by the teacher, in a teacher-student learning scenario: as observed above this solution is typically isolated. The whitening procedure described above, using the BP algorithm, finds a point-like state in correspondence of this configuration. However, if the training set is selected in a way that all the patterns which have a stability lower than 3 (for the teacher) are discarded, i.e. if we consider a situation in which the teacher is unblocked and receives only dashed lines, the whitening procedure ends up in the RS fixed point.

3.7 Random walk on a connected cluster of solutions

Now that we know that the effective learning algorithms land on solutions which are unblocked, i.e. connected to other solutions at one spin-flip, and completely unfrozen, i.e. they do not seem to be part of well defined states in the phase space, we can ask some more questions about the geometrical structures they belong to:

- How far, in terms of hamming distance, do these structures extend to? Is this distance extensive?
- What kind of distribution for the overlaps can one find among solutions belonging to the same connected group?
- Where are these structures located in the phase space, with respect to the typical frozen solutions?
- Do they contain an extensive or sub-extensive number of solutions?
- How many distinct connected components can be found by the algorithms?
- What happens at the threshold value where the heuristic solvers stop working?

In order to find an answer to these questions we have to explore the neighborhood of these unblocked solutions: a natural way of doing this is to implement a

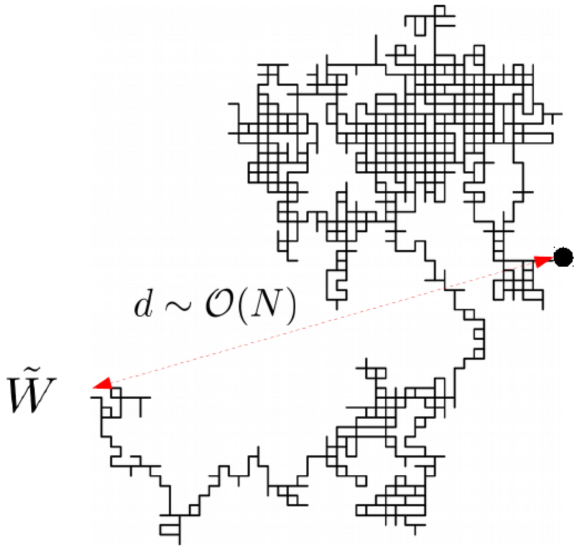


Fig. 3.7 **Random walk on the solutions** belonging to the same connected component. These kind of structures, despite being invisible to the replica analysis, seem to be ubiquitous in the space of solutions of the Perceptron.

random walk restricted to the solutions of the problem, i.e. a zero temperature Monte Carlo, during which we can measure the quantities of interest [1].

In order to estimate the radius of the connected components we can simply allow the random walk to evolve for a long time, and see what is the maximum reached distance. However, because of the curse of dimensionality, it is more efficient to bias the exploration in the outwards direction: this can be easily done, for example, by sorting the indices, in the move proposal of the MC, in a way that privileges the choice of synapses still taking the same value of the starting configuration $\tilde{W}$. These simulations show that the connected structures extend throughout the phase space, reaching $\mathcal{O}(N)$ distances.

The overlap distribution can be estimated either by memorizing many unique solutions touched by one (or more) random walks, and sampling the overlaps by choosing uniformly pairs of solutions, or by starting two parallel MC chains and seeing if the overlap evolves towards an asymptotic value: in both cases we observe that the mode of the distribution is peaked in a value which is compatible with the typical overlap between isolated solutions predicted in the replica calculations (with q in the RS Ansatz, or the external overlap q_0 in the 1RSB ansatz). This means that this kind of structure is almost ubiquitous in the phase space.

In order to count numerically both the number of solutions grouped together and the number of connected components which are present in the solution space, we need to explore extensively these structures, which poses serious numerical feasibility issues. Even if we kept only an hash table corresponding to the already reached solutions (trying to avoid problems with a fast saturation of the memory for large instances), the exploration of a structure pervading an exponential space would be possible only if such structure was fractal, containing a sub-exponential number of solutions. In this type of scenario the number of "open" directions, in a percolation procedure that keeps branching every time a new direction of the connected component is found, remains bounded.

However, the observed scenario is different: the number of contained solutions appears to be exponential, putting out of question any extensive exploration (at least when the size of the network, N, gets any close to the values where a comparison with the replica calculations would make sense). We can therefore proceed in two alternative ways:

1. Since these connected structures seem to disappear at a threshold below α_c, we can find an unblocked solution $\tilde{W}$ and keep adding constraints (patterns to be classified correctly) until the associated connected component is decimated to the point that an extensive exploration becomes possible in reasonable times.

2. We can devise a method for extrapolating the number of contained solutions from some quantities that can be measured during a random walk. The method must be conceived in a way that a sufficiently large sampling guarantees a good guess of the size of the structure.

The first approach can help us answering two of the proposed questions: suppose we move towards the algorithmic threshold and reach the point where the entropy of solutions inside the connected clusters is sufficiently limited, and suppose we then find many different solutions with one of the heuristic algorithms. We can now check how many distinct connected components were reached, by exploring all the neighborhoods of the registered solutions. The answer given by the simulations is that, a part from the detection of a few small clusters (which might be located at the "boundaries" of the main component), there is a bigger cluster which is almost always targeted by the heuristics. This result is even more impressive if one compares the number of solutions contained in these structures (near the threshold) and the estimate of the total number of solutions still present in the phase space (obtained with BP or with the replica calculations): the solvers reach always the same kind of solutions even though they are outnumbered by various orders of magnitude by the typical solutions.

Moreover, we can now memorize completely a giant connected component, and start adding more constraints to the problem. Because of these additions, some of the solutions in the cluster might get decimated, and this can lead to a fracture of the connected component into many sub-components. We can register the hierarchical relationship between these connected clusters and observe how fast this kind of structures disappear. In figure 3.8, we can see a depiction of this process, as measured in an instance with size $N = 201$. Between $\alpha = 0.756$ and $\alpha = 0.78$ all the solutions which had been part of the same connected component got decimated. By comparing this disappearance phenomenon at different sizes, we conjecture that it becomes a sharp transition in the limit $N \to \infty$. After this cluster has disappeared no algorithm seems to be able to find solutions, even though exponentially many (isolated) solutions are known to be left until $\alpha_c = 0.833$.

In order to give a final answer to question regarding the extensive/sub-extensive entropy of solutions contained in the connected components, we need to try the second approach proposed above. A possible strategy for obtaining an approximation of this quantity within a given radius is the following: we can start many random walks from the "central" solution initially found by the algorithm $\tilde{W}$. During each MC trajectory, we record and update the average Outgoing Branching Factor (OBF) and the average Ingoing Branching Factor (IBF) at each distance D, taking into account the number of unique solutions observed at that radius. We also need to consider a special case, where the random walk exploration took us to terminal which was going inwards: we thus introduce a Probability of Birth (PB), again function of the distance.

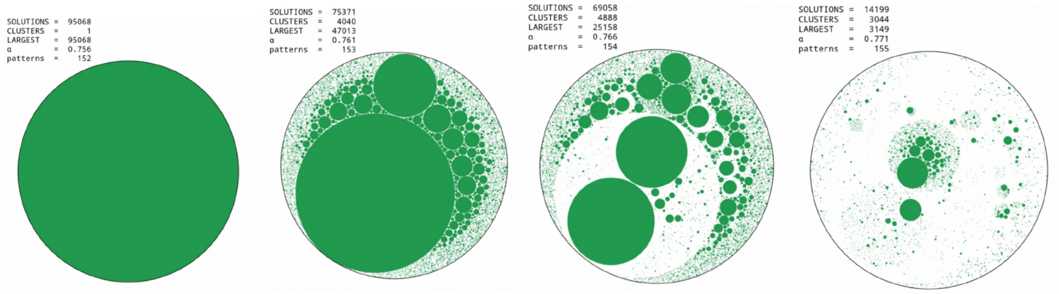


Fig. 3.8 **Disappearance of the cluster** in a binary Perceptron of size $N = 201$. The system was prepared very close to the algorithmic threshold, where an extensive exploration of the connected components found algorithmically is feasible. In this depiction we show the hierarchical relationships between the surviving components of the cluster, after the addition of new random patterns. The "children" components in each picture are contained inside the same area occupied by the "father" component in the picture to its left. Notice that this depiction is not meant to correctly represent distance relationships between the solutions. After the addition of a few more constraints all the solution disappear (around $\alpha \sim 0.78$).

By using all these quantities we can estimate the rate of growth of the cluster as one moves away from the starting solution, and therefore the entropy of solutions within a given radius. Of course the sampling required for a correct estimation at large distances diverges very rapidly, but at small distances the procedure seems to be quite robust. The number $\mathcal{N}$ of solutions at distance D can be found recursively by using the relation:

$$\mathcal{N}(D+1) = \frac{\mathrm{OBF}(D)}{\mathrm{IBF}(D)} \left(1 + \mathrm{PB}(D)\right) \mathcal{N}(D). \tag{3.18}$$

This procedure was checked to converge to the correct values in the scenario (described above) where an extensive exploration is also possible. By applying this method to different regimes, we can see that the number of solutions (at a distance of order N from the starting one) grows exponentially with N (see figure **3.9**).

Finally, one can also study the Franz-Parisi potential (estimate the total entropy of solutions at each distance), in a single problem instance, by exploiting BP (with the addition of external fields of variable intensity $\gamma(D)$). With this analysis we can observe that, as we leave the core of these dense regions of connected solutions, some isolated solutions pop out "around" the connected

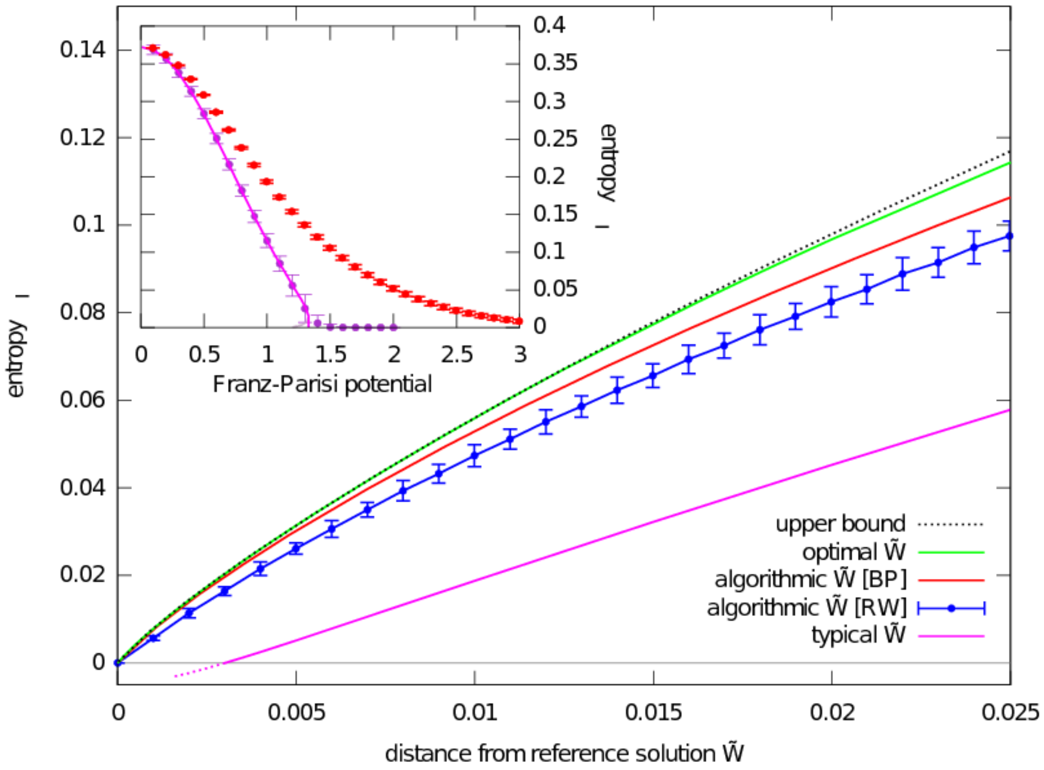


Fig. 3.9 **Numerical evidence of the existence of clusters of solutions**. Entropy at a given distance from a reference solution $\tilde{W}$, in the classification case at $\alpha = 0.4$. From bottom to top: (magenta) theoretical prediction for a typical $\tilde{W}$; (blue) numerical estimate based on a random walks on connected solutions starting from one provided by SBPI, with $N = 1001$; (red) estimate from Belief Propagation using a solution from SBPI, with $N = 10001$; (green) theoretical curve for the optimal $\tilde{W}$ as computed in the next chapter; (dotted black) upper bound ($\alpha = 0$ case, all configurations are solutions). The random-walk points underestimate the number of solutions since they only consider single-flip-connected clusters; the BP curve is lower than the optimal because in the latter $\tilde{W}$ is optimized as a function of the distance, while in the former it is fixed. *Inset*: comparison between a typical solution and one found with SBPI, in the teacher-student case at $\alpha = 0.5$ with $N = 1001$. Larger potentials correspond to smaller distances. Top points (red): SBPI reference solution, entropy computed by BP; bottom curve (magenta): theoretical prediction for a typical solution; bottom points (purple): BP results using the teacher as reference.

component. At small distances, though, the entropy of the connected cluster almost saturates both the total entropy of solutions (as estimated by BP) and the maximal entropy (the upper bound is given by total number of configurations at that distance), which means that these structures are extremely dense. All these results are collected in figure **3.9**, from [1].

We also performed a random-walk exploration in the space of solutions in the case of a multi-layer network, starting from a solution obtained with the extended version of the CP+R algorithm. The results are qualitatively similar: the simplified algorithm lands on a solution which is part of a large connected cluster, whose extremely high local density decreases as one moves away from the core found by the solver. This observation could also explain the small improvements in the generalization obtained with the Bayesian approach presented in section 3.4: if most of the configurations surrounding the planted solution are solutions themselves, there is no need to run BP again in order to obtain a good generalization performance.

Part II

Large Deviation Analysis

Chapter 4

Novel Measure

All the results gathered in the previous chapter leaves us facing a quite confusing conundrum: even in the case of the simplest discrete neural network model, the Perceptron, there seems to be a discrepancy between the theoretical picture, obtained analytically through the replica method, and a series of numerical and algorithmic findings.

On the one hand, the classical Statistical Mechanics description of the model depicts an energy landscape dominated by an enormous number (exponential in the number of synapses N) of local minima, where the typical optimal synaptic configurations are geometrically isolated (with mutual Hamming distances of $\mathcal{O}(N)$). This kind of landscape can easily trap standard optimization strategies based on energy minimization, e.g. Monte Carlo [21, 22], and any kind of local search algorithm should fail [22].

On the other hand, after the introduction of a few heuristic modifications, the Belief Propagation approach proves to be able to achieve nearly optimal performance and succeeds in finding a special class of the ephemeral solutions of the Perceptron model. Moreover, one of the defining features of these particular solutions – namely their high density near the core of a cluster pervading the solution space – is in open contrast with the theoretical predictions for the geometrical landscape of the problem, and one could even question whether the Parisi ansatz usually employed in standard equilibrium Replica calculations is suited for the description of such type of structures.

In this chapter we will give a general introduction to the theoretical concepts and methods that we developed in the last few years [1, 4, 2, 3] in order to find an answer to these open questions.

4.1 Sub-dominant structures

Since a landscape analysis based on an unbiased Gibbs measure leads to the description of solutions which cannot realistically be found by any algorithm, in order to filter out the typical optimal configurations (that dominate the picture) and highlight other classes of solutions, we need to carry out a large deviations analysis: instead of the standard Boltzmann-Gibbs weight (an indicator function $\mathbb{X}_\xi\left(W\right) = \prod_{\mu=1}^{\alpha N}\Theta\left(\sigma^\mu W\cdot\xi^\mu\right)$ over the solutions of the problem, in a CSP formulation), we can consider a reweighed measure, where the probability of any configuration $\tilde{W}$ is given by:

$$P\left(\tilde{W};y,D\right)=\frac{\mathbb{X}_{\xi,\sigma}\left(\tilde{W}\right)e^{yN\mathcal{S}\left(\tilde{W},D\right)}}{\sum_{\tilde{W}'}\mathbb{X}_{\xi,\sigma}\left(\tilde{W}'\right)e^{yN\mathcal{S}\left(\tilde{W},D\right)}}. \tag{4.1}$$

The reweigthing is done through a *local entropy* function $\mathcal{S}\left(\tilde{W},D\right)=\frac{1}{N}\log\mathcal{N}\left(\tilde{W},D\right)$, where:

$$\mathcal{N}\left(\tilde{W},D\right)=\sum_{\{W\}}\mathbb{X}_\xi\left(W\right)\delta\left(W\cdot\tilde{W},N\left(1-2D\right)\right) \tag{4.2}$$

counts the number of solutions W at normalized Hamming distance D from the reference $\tilde{W}$. When the value of the inverse temperature y is set to 0 we retrieve the typical case, while as we increase y the measure gets more focused on denser regions of solutions in the landscape. In the limit $y\to 0$, instead, $\mathcal{S}\left(D,y\right)$ reduces to the computation *à la* Franz-Parisi of [22].

It is important to stress the role of the distance parameter D: it serves for the definition of a neighborhood of the reference configuration inside which we are counting the solutions – consequently the name *local entropy.*

Instead of using the Kronecker δ-function, a possibly more natural definition for the counting function could be:

$$\mathcal{N}\left(\tilde{W},\gamma\right)=\sum_{\{W\}}\mathbb{X}_\xi\left(W\right)\exp\left[-\frac{\gamma}{2}\left(W-\tilde{W}\right)^2\right] \tag{4.3}$$

where γ is the Laplace conjugate of D, seemingly playing the role of an elastic interaction between $\tilde{W}$ and the surrounding solutions. While this definition is more easily implementable in an algorithmic setting (as we will see in chapters 5 and 6), in the thermodynamic limit it becomes completely equivalent to the definition 4.2, since in high dimensions the points counted in $\mathcal{N}\left(\tilde{W},\gamma\right)$ accumulate on the "circumference" of radius $D\left(\gamma\right)$. On the other hand, the first definition avoids some technical issues in the calculations, arising when the Legendre transform between D and γ loses its injection property.

We thus want to study the following free entropy density: We can study the typical behavior of these modified measures as usual within the replica theory, by computing their corresponding average free entropy density:

$$\Phi\left(D,y\right)=\frac{1}{N}\left\langle \log\sum_{\tilde{W}}\mathbb{X}_{\xi,\sigma}\left(\tilde{W}\right)e^{y\mathcal{S}_{\xi,\sigma}\left(\tilde{W},D\right)}\right\rangle \tag{4.4}$$

This free entropy describes a system in which each configuration $\tilde{W}$ is constrained to be a solution, and has an additional energy $\mathrm{E}\left(\tilde{W},D\right)=-N\mathcal{S}\left(\tilde{W},D\right)$ which favors configurations surrounded by an exponential number of other solutions. From this quantity we can also obtain the typical values of the local entropy density:

$$\mathcal{S}\left(D,y\right)=\frac{\partial}{\partial y}\Phi\left(D,y\right) \tag{4.5}$$

and of the external entropy density, or complexity:

$$\Sigma\left(D,y\right)=\Phi\left(D,y\right)-y\mathcal{S}\left(D,y\right) \tag{4.6}$$

While the first depends on the number of solutions inside each cluster (centered in $\tilde{W}$), the second quantity counts the entropy (i.e. the logarithm of the number) of clusters. There are mainly two possible scenarios for the distribution of the desired outputs σ^{μ}:

1. the *classification* (or *storage*) case, in which they are i.i.d. random variables and therefore the patterns are completely uncorrelated.

2. the *generalization* (or *teacher-student*) scenario, in which the correct results are provided by a "teacher" device, i.e. another Perceptron with

synaptic weights W^{T}. Without loss of generality, the teacher synaptic weights W^{T} can all be set to 1, thanks to a symmetry of the problem.

We here consider the first case, as the second is slightly more involved, but follows the same exact line of reasoning and yields similar qualitative results [1].

Replica trick In order to study the large-deviation free entropy density, we can first compute the replicated volume $\Omega^n(D,y)$, using the relation $\Phi(D,y) \equiv \frac{1}{N}\lim_{n\to 0}\frac{\partial}{\partial n}\Omega^n(D,y)$. The quenched average over the set of patterns $\{\xi^\mu,\sigma^\mu\}_{\mu=1,\ldots,\alpha N}$ can be evaluated by using the replica trick. In the following n will denote the number of replicas of the reference configuration $\tilde{W}$, and the letters c and d, with $c,d\in\{1,\ldots,n\}$, will indicate the associated replica indices. The same trick can also be exploited for expanding the $\mathcal{N}\left(\tilde{W},S\right)^y$ term, introducing ny "student" replicas, with the real replica indices denoted by $a,b\in\{1,...,y\}$.

Remembering the teacher-student problem we can portrait a geometrical situation of this kind: each of the n reference solutions, distributed in the solution space, will look like a "pseudo-teacher", surrounded by y "student" replicas at a fixed distance D. In this computation, though, only the $n\to 0$ limit must be taken while y will remain as a parameter of the problem.

We thus need to evaluate the replicated volume:

$$\begin{aligned}\Omega^n(D,y) = &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (4.7)\\ &\left\langle\int\prod_{i,c}d\mu\left(\tilde{W}_i^c\right)\int\prod_{i,ca}d\mu\left(W_i^{ca}\right)\prod_c\mathbb{X}_{\xi,\sigma}\left(\tilde{W}^c\right)\prod_{ca}\mathbb{X}_{\xi,\sigma}\left(W^{ca}\right)\right.\\ &\left.\times\prod_{ca}\delta\left(\frac{1}{2}\sum_i\left(W_i^{ca}-\tilde{W}_i^c\right)^2-2DN\right)\right\rangle_{\xi,\sigma}\end{aligned}$$

We can follow the steps of a standard replica calculation (cf. with chapter 2), substituting the arguments of the theta functions in the $\mathbb{X}_\xi$ terms via Dirac-delta functions, expanding these delta functions using their integral representation, factorizing the expression where the patterns are involved and taking the average over the disorder in the large N limit:

$$\prod_{\mu,i}\left\langle \exp\left(-\frac{i}{\sqrt{N}}\left(\sum_{c}\hat{\tilde{\lambda}}_{\mu}^{c}\tilde{W}_{i}^{c}+\sum_{ca}\hat{\lambda}_{\mu}^{ca}W_{i}^{ca}\right)\xi_{i}^{\mu}\right)\right\rangle_{\xi} = \tag{4.8}$$

$$\prod_{\mu}-\frac{1}{2}\left(\sum_{ca,db}\hat{\lambda}_{\mu}^{ca}\hat{\lambda}_{\mu}^{db}\sum_{i}\frac{W_{i}^{ca}W_{i}^{db}}{N}+\sum_{c,d}\hat{\tilde{\lambda}}_{\mu}^{c}\hat{\tilde{\lambda}}_{\mu}^{d}\sum_{i}\frac{\tilde{W}_{i}^{c}\tilde{W}_{i}^{d}}{N}+\right.$$
$$\left.+2\sum_{ca,d}\hat{\lambda}_{\mu}^{ca}\hat{\tilde{\lambda}}_{\mu}^{d}\sum_{i}\frac{W_{i}^{ca}\tilde{W}_{i}^{d}}{N}\right)$$

Now we can introduce the order parameters for the overlaps of the model, which can be fixed by introducing the related Dirac delta distributions:

- $\sum_i \frac{\tilde{W}_i^c \tilde{W}_i^d}{N} = \tilde{q}^{cd}$, the overlap between two reference solutions $\tilde{W}$.
- $\sum_i \frac{W_i^{ca} W_i^{db}}{N} = q^{ca,db}$, the overlap between two student solutions W.
- $\sum_i \frac{W_i^{ca} \tilde{W}_i^d}{N} = S^{ca,d}$, the overlap between a student and a reference.

After the substitutions in the expression of the volume, one can manage to factorize over the indices μ and i (thus removing all those indices), getting:

$$\Omega^{n}\left(D,y\right)=\int\prod_{\substack{c,a>b\\ c>d,ab}}\frac{dq^{ca,db}d\hat{q}^{ca,db}}{\left(2\pi/N\right)}\int\prod_{c>d}\frac{d\tilde{q}^{cd}d\hat{\tilde{q}}^{cd}}{\left(2\pi/N\right)}\int\prod_{ca,d}\frac{dS^{ca,d}d\hat{S}^{ca,d}}{\left(2\pi/N\right)} \tag{4.9}$$
$$\times\int\prod_{ca}\frac{d\hat{D}^{ca}}{2\pi}G_{1}\,G_{S}^{N}\,G_{E}^{\alpha N}$$

with the definitions for the G_1 and for the entropic and energetic terms G_S,G_E:

$$
\begin{aligned}
G_1 = &\exp\left(-N\left(\sum_c\sum_{a>b}\hat{q}^{ca,cb}q^{ca,cb}+\sum_{c>d}\sum_{ab}\hat{q}^{ca,db}q^{ca,db}+\sum_{c>d}\hat{\tilde{q}}^{cd}\tilde{q}^{cd}+\right.\right.\\
&\left.\left.+\sum_{ca,d}\hat{S}^{ca,d}S^{ca,d}+\sum_{ca}\hat{D}^{ca}\left(1-2D-S^{ca,c}\right)\right)\right)
\end{aligned}
\tag{4.10}
$$

$$
\begin{aligned}
G_S = &\int\prod_c d\mu\left(\tilde{W}^c\right)\int\prod_{ca}d\mu\left(W^{ca}\right)\exp\left(\sum_c\sum_{a>b}\hat{q}^{ca,cb}W^{ca}W^{cb}+\sum_{c>d}\sum_{ab}\hat{q}^{ca,db}W^{ca}W^{db}+\right.\\
&\left.+\sum_{c>d}\hat{\tilde{q}}^{cd}\tilde{W}_i^c\tilde{W}_i^d+\sum_{dca}\hat{S}^{d,ca}W^{ca}\tilde{W}^d\right)
\end{aligned}
\tag{4.11}
$$

$$
\begin{aligned}
G_E = &\int\prod_c\frac{d\tilde{\lambda}^c d\hat{\tilde{\lambda}}^c}{2\pi}\int\prod_{ca}\frac{d\lambda^{ca}d\hat{\lambda}^{ca}}{2\pi}\prod_c\Theta\left(\tilde{\lambda}^c\right)\prod_{\mu,ca}\Theta\left(\lambda^{ca}\right)\exp\left(i\left(\sum_c\tilde{\lambda}^c\hat{\tilde{\lambda}}^c+\sum_{ca}\lambda^{ca}\hat{\lambda}^{ca}\right)\right)\\
&\times\exp\left(\left(-\frac{1}{2}\sum_c\left(\hat{\tilde{\lambda}}^c\right)^2-\frac{1}{2}\sum_{ca}\left(\hat{\lambda}^{ca}\right)^2-\sum_{c>d}\hat{\tilde{\lambda}}^c\hat{\tilde{\lambda}}^d\tilde{q}^{cd}+\right.\right.\\
&\left.\left.-\sum_c\sum_{a>b}\hat{\lambda}^{ca}\hat{\lambda}^{cb}q^{ca,cb}-\sum_{c>d}\sum_{ab}\hat{\lambda}^{ca}\hat{\lambda}^{db}q^{ca,db}-\sum_{ca,d}\hat{\lambda}^{ca}\hat{\tilde{\lambda}}^d S^{d,ca}\right)\right)
\end{aligned}
\tag{4.12}
$$

4.2 Replica Symmetric Ansatz

In order to proceed with the computation, we now need to make a simplification and put forward an Ansatz on the structure of the parameters describing the replicated system. We start from the simplest one, a replica symmetric Ansatz; notice however that the reweighting term already introduced a natural grouping of the students W in n sets of y replicas, each surrounding a certain reference solution $\tilde{W}$, thus leading to a situation formally similar to a 1RSB description (see section **2.3**).

We thus have to make a distinction between the typical overlap q_1, between replicas found surrounding the same reference $\tilde{W}$ (say W^{ca},W^{cb}), and the overlap q_0, between replicas referred to different ones (say W^{ca},W^{db} with different $c \neq d$): we can assume the first one to be larger, since the distance constraint shared by students of the same $\tilde{W}$ increases their correlation. We also expect these students to have an overlap S with their own reference which is higher than the typical overlap $\tilde{S}$, with another $\tilde{W}$. We therefore set:

- $q^{ca,cb} = q_1$ for $(a \neq b)$, $q^{ca,db} = q_0$ for $(c \neq d)$
- $S^{ca,c} = S$, $S^{ca,d} = \tilde{S}$ for $(c \neq d)$
- $\tilde{q}^{cd} = \tilde{q}$, $\hat{D}^{ca} = \hat{D}$

With these assumptions, neglecting the $O\left(n^2\right)$ terms, we find for G_1:

$$\begin{aligned} G_1 = &\exp\left(-Nny\left(\frac{(y-1)}{2}q_1q_1 - \frac{y}{2}\hat{q}_0q_0 - \frac{1}{2}\frac{\hat{\tilde{q}}\tilde{q}}{y}\right.\right. \\ &\left.\left.+\hat{S}S - \hat{\tilde{S}}\tilde{S} + \hat{D}\left(1-2D-S\right)\right)\right) \end{aligned} \tag{4.13}$$

Moving to the computation of the entropic term we start by recasting

$$\hat{\tilde{S}}\sum_{ca}W^{ca}\sum_{c}\tilde{W}^c = \frac{1}{2}\hat{\tilde{S}}\left(\sum_{ca}W^{ca} + \sum_{c}\tilde{W}^c\right)^2 - \frac{1}{2}\hat{\tilde{S}}\left(\sum_{ca}W^{ca}\right)^2 - \frac{1}{2}\hat{\tilde{S}}\left(\sum_{c}\tilde{W}^c\right)^2 \tag{4.14}$$

and then we can proceed by:

1. introduce the variables x, z_0, $\tilde{z}$ to perform three Hubbard-Stratonovich transformations, in order to get rid of the squared sums involving the replica index c and to factorize over it;
2. perform the last Hubbard-Stratonovich transformation and factorize over the index a as well;
3. consider the limit $n \to 0$, bringing the logarithm inside the Gaussian integration;
4. perform two rotations of the integration variables in order to evaluate analytically the $\int \mathcal{D}x$ integral.

In the end we obtain:

$$\begin{aligned}\mathcal{G}_S =& \frac{1}{n}\log G_S = \qquad (4.15)\\ & -\frac{1}{2}\hat{\hat{q}} - \frac{y}{2}\hat{q}_1 + \int \mathcal{D}\tilde{z} \int \mathcal{D}z_0 \log \left\{ \sum_{\tilde{l}=\pm 1} \exp\left(\left(\tilde{z}\sqrt{\hat{\hat{q}} - \frac{\hat{\hat{S}}^2}{\hat{q}_0}} + z_0 \frac{\hat{\hat{S}}}{\sqrt{\hat{q}_0}} \right) \tilde{l} \right) \right.\\ & \left. \times \int \mathcal{D}z_1 \left[2\cosh\left(z_0\sqrt{\hat{q}_0} + z_1\sqrt{\hat{q}_1 - \hat{q}_0} + \left(\hat{S} - \hat{\hat{S}} \right) \tilde{l} \right) \right]^y \right\}\end{aligned}$$

In a similar way, we reorganize the summations in the energetic term, with the substitution:

$$\tilde{S}\sum_{ca}\hat{\lambda}^{ca}\sum_{c}\hat{\hat{\lambda}}^{c} = \frac{1}{2}\tilde{S}\left(\left(\sum_{ca}\hat{\lambda}^{ca} + \sum_{c}\hat{\hat{\lambda}}^{c}\right)^2 - \left(\sum_{c}\hat{\hat{\lambda}}^{c}\right)^2 - \left(\sum_{ca}\hat{\lambda}^{ca}\right)^2\right) \qquad (4.16)$$

Next, we can perform the following operations, in order to evaluate the $\mathcal{D}x$ integral:

1. perform three Hubbard-Stratonovich transformations, introducing x, z_0, and $\tilde{z}$, and factorizing over the index c;

2. evaluate the Gaussian integral in the variable $\hat{\hat{\lambda}}$;

3. a fourth Hubbard-Stratonovich transformation, introducing the variable z_1 and factorizing over the replica index a;

4. evaluate also the $\hat{\lambda}$ Gaussian integral;

5. perform a rotation between z_0 and x;

6. change the sign of x and perform another rotation between $\tilde{z}$ and x;

7. perform a shift in the variable $\tilde{\lambda}$.

Now, after the integral in x, using the fact that$\int Dz\, H\left(az+b\right) = H\left(\frac{b}{\sqrt{1+a^2}}\right)$, we take the logarithm of the energetic term in the $n \to 0$ limit, and after rotating z_1 and $\tilde{\lambda}$ we can introduce the error functions to finally find:

$$\begin{aligned}\mathcal{G}_E =& \frac{1}{n}\log G_E = \\ &\int \mathcal{D}z_0 \int \mathcal{D}\tilde{z} \log\left(\left(\int \mathcal{D}z_1 H\left(-\frac{z_0\sqrt{q_0}+z_1\sqrt{q_1-q_0}}{\sqrt{(1-q_1)}}\right)^y H\left(\tilde{C}\left(s,z_0,z_1,\tilde{z}\right)\right)\right)\end{aligned} \tag{4.17}$$

with the definition:

$$\tilde{C}\left(s,z_0,z_1,\tilde{z}\right) = -\frac{\tilde{z}\sqrt{\left(\tilde{q}-\frac{\tilde{S}^2}{q_0}\right)}+z_0\sqrt{\frac{\tilde{S}^2}{q_0}}+z_1\frac{\left(S-\tilde{S}\right)}{\sqrt{(q_1-q_0)}}}{\sqrt{(1-\tilde{q})-\frac{\left(S-\tilde{S}\right)^2}{(q_1-q_0)}}} \tag{4.18}$$

Putting the pieces together and using the saddle point method we finally obtain a leading order estimate of the free energy density function in the large N limit:

$$\begin{aligned}\Phi\left(D,y\right) \approx -\Bigg(\Bigg(&-\frac{y}{2}\left(1-q_1\right)\hat{q}_1-\frac{y^2}{2}\left(\hat{q}_1q_1-\hat{q}_0q_0\right)-\frac{1}{2}\left(1-\tilde{q}\right)\hat{\tilde{q}}-y\left(\hat{S}S-\hat{\tilde{S}}\tilde{S}\right)+ \\ &-y\hat{D}\left(1-2D-S\right)\Bigg)+\mathcal{G}_S+\alpha\mathcal{G}_E\Bigg)\end{aligned} \tag{4.19}$$

where $\mathcal{G}_S$ and $\mathcal{G}_E$ are defined according to equations 4.15and 4.17. The stationarity condition implies the following saddle point equations:

$$\begin{aligned}
&\tilde{q} = -2\frac{\partial}{\partial\hat{\tilde{q}}}\mathcal{G}_S; && q_0 = -\frac{2}{y^2}\frac{\partial}{\partial\hat{q}_0}\mathcal{G}_S; && q_1 = \frac{2}{y\left(y-1\right)}\frac{\partial}{\partial\hat{q}_1}\mathcal{G}_S; \\
&\tilde{S} = -\frac{1}{y}\frac{\partial}{\partial\hat{\tilde{S}}}\mathcal{G}_S; && S = 1-2D; && 0 = \frac{1}{y}\frac{\partial}{\partial\hat{S}}\mathcal{G}_S - S; \\
&\hat{\tilde{q}} = -2\alpha\frac{\partial}{\partial\tilde{q}}\mathcal{G}_E; && \hat{q}_0 = -\frac{2\alpha}{y^2}\frac{\partial}{\partial q_0}\mathcal{G}_E; && \hat{q}_1 = \frac{2\alpha}{y\left(y-1\right)}\frac{\partial}{\partial q_1}\mathcal{G}_E; \\
&\hat{D} = \hat{S} - \frac{\alpha}{y}\frac{\partial}{\partial S}\mathcal{G}_E; && \hat{\tilde{S}} = -\frac{\alpha}{y}\frac{\partial}{\partial\tilde{S}}\mathcal{G}_E;.
\end{aligned} \tag{4.20}$$

We are thus left with a system of 11 coupled equations that can be solved by recursion, and three control parameters (α, y and D).

It is important to stress the geometrical implications of the dependence on the distance D of the overlaps describing the structure of the dense cluster, in contrast with what usually happens in the 1RSB Parisi Ansatz. This means that a special type of symmetry, which usually holds inside the clusters found by the standard 1RSB, implying that the contained solutions are typically equidistant from one another (therefore one single overlap is sufficient for describing their geometrical structure) is broken in this sub-dominant structure, that becomes exponentially denser near its core.

Consistency check: $y = 1$ case In the case $y = 1$, the integrals over z_1 in $\mathcal{G}_S$ and $\mathcal{G}_E$ can be computed explicitly, using the formulas:

$$\int Dz\, H\left(az+b\right) = H\left(\frac{b}{\sqrt{1+a^2}}\right) \tag{4.21}$$

$$\int Dz\, \cosh\left(az+b\right) = e^{\frac{a^2}{2}}\cosh\left(b\right) \tag{4.22}$$

As expected, the dependency of $\Phi\left(D,1\right)$ on q_1 and $\hat{q}_1$ cancels out. Furthermore, the resulting equations seem to have only solutions with $\tilde{q} = q_0 = \tilde{S}$ and $\hat{\tilde{q}} = \hat{q}_0 = \hat{\tilde{S}}$. Therefore, we are left with only 3 order parameters (besides S which is set by the external parameter D): q, $\hat{q}$ and $\hat{S}$. The resulting simplified expression is:

$$\Phi\left(S,1\right) = \left(1-2q\right)\hat{q} + S\hat{S} - \hat{D}\left(1-2D-S\right) - \mathcal{G}_S^1 - \alpha\mathcal{G}_E^1 \tag{4.23}$$

$$\mathcal{G}_S^1 = \int Dz\, \log\left(\sum_{\tilde{l}=\pm1} e^{z\tilde{l}\sqrt{\hat{q}}}\left(2\cosh\left(z\sqrt{\hat{q}} + \tilde{l}\left(\hat{S}-\hat{q}\right)\right)\right)\right) \tag{4.24}$$

$$\mathcal{G}_E^1 = \int Dz\, \log\left(\int_{-z\sqrt{\frac{q}{1-q}}}^{\infty} D\lambda\, H\left(-\frac{z\sqrt{q\left(1-q\right)} + \left(S-q\right)\lambda}{\sqrt{\left(1-q\right)^2 - \left(S-q\right)^2}}\right)\right) \tag{4.25}$$

This expression has two limiting cases which can be verified analytically:

- when S is equal to the value for the overlap of the single Perceptron q_{RS}, we have $\Phi\left(\frac{1-q_{RS}}{2}, 1\right) = -2\mathcal{S}_{RS}$ where $\mathcal{S}_{RS}$ is the RS entropy of the single Perceptron, since in that case the contributions of the $\tilde{W}$ and W terms in Φ essentially factorize; this case is essentially equivalent to the computation of [19];
- when on the other hand $D = 0$ and $S = 1$, we obtain the degenerate case $\Phi\left(0,1\right) = -\mathcal{S}_{RS}$, since in that case the contribution of the W terms vanishes.

4.3 Large y limit

From a physicist perspective, it is natural to consider the limiting case $y \to \infty$, looking for the (possibly unique) ground state of the free energy 4.4. This means that we want to characterize the optimal reference solution $\tilde{W}^{\star}$ surrounded by most other solutions at the given distance D. If even in thermodynamic limit an exponentially large cluster of solutions exists in the solution landscape of our problem, we expect to find $\mathcal{S}^{\star}(D) > 0$ in any small neighborhood of $D = 0$.

In the large y limit the computation gets greatly simplified: it is easy to observe that the $\mathbb{X}_{\xi,\sigma}\left(\tilde{W}\right)$ constraint on the reference configuration effectively disappears and that the fixed point for the saddle point equations remains unaltered when it is completely removed. The reason is the following: in the expression for energetic term $\mathcal{G}_E$ of equation (4.19), the function $H\left(\tilde{C}\left(s, z_0, z_1, \tilde{z}\right)\right)$ is not elevated to the power of y, and thus becomes effectively irrelevant, implying that $\mathcal{G}_E$ is constant with respect to the order parameters $\tilde{q}$ and $\tilde{S}$. In turn, looking at the saddle point equations, this means that $\hat{\tilde{q}}$, and $\hat{\tilde{S}}$ are all 0.

Thus the large y case can formally be obtained from the final expression 4.19 by setting to zero the order parameters describing the planted configuration $\tilde{q}$, $\tilde{S}$, and their conjugates (even though the order parameters are not 0, and their value can be obtained by carefully performing the limit). Moreover the integration over $\tilde{z}$ in the $\mathcal{G}_S$ and $\mathcal{G}_E$ terms can be carried out analytically. The final expression for the free entropy in this limit is thus the same as for the unconstrained case, namely:

$$\begin{aligned}\Phi\left(D, y\right) = -\Bigg(\bigg(&-\frac{y}{2}\left(1-q_1\right)\hat{q}_1 - \frac{y^2}{2}\left(\hat{q}_1 q_1 - \hat{q}_0 q_0\right) - y\hat{S}S + \\ &-y\hat{D}\left(1-2D-S\right)\bigg) + \mathcal{G}_S + \alpha\mathcal{G}_E\Bigg)\end{aligned} \tag{4.26}$$

$$\mathcal{G}_S = \int \mathcal{D}z_0 \log\left\{\sum_{\tilde{l}=\pm 1}\int \mathcal{D}z_1\left[2\cosh\left(z_0\sqrt{\hat{q}_0} + z_1\sqrt{\hat{q}_1-\hat{q}_0} + \hat{S}\tilde{l}\right)\right]^y\right\} \tag{4.27}$$

$$\mathcal{G}_E = \int \mathcal{D}z_0 \log\left(\int \mathcal{D}z_1 H\left(-\frac{z_0\sqrt{q_0} + z_1\sqrt{\left(q_1-q_0\right)}}{\sqrt{\left(1-q_1\right)}}\right)^y\right) \tag{4.28}$$

It is easy to notice the resemblance of this expression with the standard 1RSB case (in section 2.3), with the inverse temperature y taking the role of the Parisi parameter m. However, 1-RSB solution of the standard equations shows no hint of the dense regions which we find in the present work, even if we relax the requirement $0 \leq m \leq 1$ of [17]. This shows that the constraint on the distance is crucial to explore these sub-dominant regions.

In order to take the $y \to \infty$ limit we need to make a self-consistent Ansatz for the scaling of some order parameters with y: there is a term $\frac{y^2}{2}\left(q_1\hat{q}_1 - q_0\hat{q}_0\right)$ which diverges unless $q_1\hat{q}_1 - q_0\hat{q}_0 = \mathcal{O}\left(y^{-1}\right)$. This suggests that $q_1 - q_0 = \mathcal{O}\left(y^{-1}\right)$ and $\hat{q}_1 - \hat{q}_0 = \mathcal{O}\left(y^{-1}\right)$, i.e. a situation in which groups of solutions relative to different $\tilde{W}$ would tend to become the same (and therefore $\tilde{q} \to 1$). So we can define $q_0 = q$ and consider the scaling $q_1 \to q + \frac{\delta q}{y}$. Similarly we can pose $\hat{q}_0 = \hat{q}$, $\hat{q}_1 \to \hat{q} + \frac{\delta\hat{q}}{y}$.

We now want to evaluate the $\int \mathcal{D}z_1$ integrals appearing in the energetic and the entropic terms through a first order saddle point approximation: for this purpose we need to rescale the integration variable $z_1' = \sqrt{y}z_1$in the entropic and energetic terms. The z_1 integrals and the summation over $\tilde{l}$ in the entropic term are thus replaced by maximum functions, obtaining:

$$\begin{aligned} &\lim_{y\to\infty} \Phi\left(D, y\right) \approx \qquad\qquad (4.29)\\ &\lim_{y\to\infty} y \left(-\frac{1}{2}\left(1-q\right)\hat{q} - \frac{1}{2}\delta q\hat{q} - \frac{1}{2}q\delta\hat{q} - \hat{S}S - \hat{D}\left(1-2D-S\right)\right.\\ &\quad + \int \mathcal{D}z_0 \max_{\tilde{l}=\pm 1}\left(\max_{z_1}\left(-\frac{z_1^2}{2} + \log\left(2\cosh\left(z_0\sqrt{\hat{q}} + z_1\sqrt{\delta\hat{q}} + \hat{S}\tilde{l}\right)\right)\right)\right)\\ &\quad \left. + \alpha\int \mathcal{D}z_0 \max_{z_1}\left(-\frac{z_1^2}{2} + \log\left(H\left(-\frac{z_0\sqrt{q} + z_1\sqrt{\delta q}}{\sqrt{1-q}}\right)\right)\right)\right) \end{aligned}$$

where the constant vanishing term $\frac{\log y}{y}$ was neglected. Since S doesn't appear in the energetic term, at the saddle the equation $\hat{S} = \hat{D}$ holds, and after the appropriate substitutions S comes out from the picture and can be ignored. The typical values for the parameters can then be found by iterating the saddle point equations, found by posing all the derivatives with respect to the order parameters to 0.

After a study of the function inside the maximum (*argGs*, in figure 4.1) in the final expression of the entropic term, it is easy to see that the optimal value

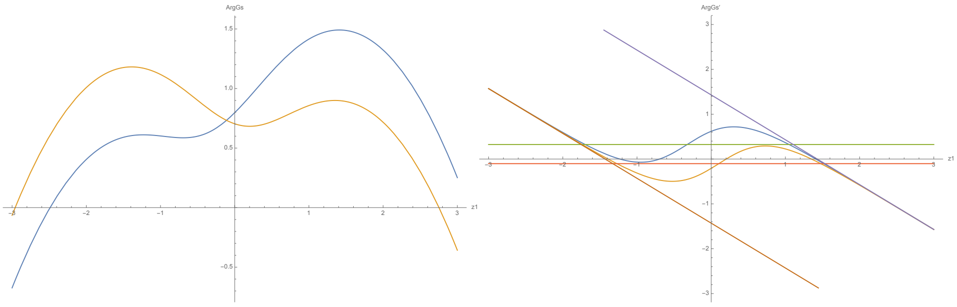


Fig. 4.1 **Study of the** $argGs$ **function**: the parameters where set (randomly) to the values $\hat{q} = 0.1$, $\delta\hat{q} = 2.0$, $\hat{D} = 0.3$, $z0 = 0.5$. In figure (a) the function $argGs$ is plotted, for the two values $\tilde{l} = 1$ (blue curve) and $\tilde{l} = -1$ (yellow curve). The global maximum is always found in the case $\tilde{l} = \text{sign}\,(z_0)$. From the study of the derivative, in figure (b), we can see that $argGs'$ (blue and yellow curve for the cases $\tilde{l} = \pm 1$) is always between the two axes $y = \pm\sqrt{\delta\hat{q}} - z_1$ (purple and orange lines). At fixed $\tilde{l}$, when the vertical symmetry axis (at the values $\left(z_0\sqrt{\hat{q}} \pm \hat{D}\right)/\sqrt{\delta\hat{q}}$, green and red lines) passes through the origin as z_0 is increased, the extremal value $z_1^\star$switches side (from a value around $\sqrt{\delta\hat{q}}$ to $-\sqrt{\delta\hat{q}}$, see also figure (a)).

of $\tilde{l}$ is always sign (z_0), and the Newton algorithm can be initialized around $\sqrt{\delta\hat{q}}\text{sign}\,(z_0)$.

Once the optimal value $z_1^\star$ is found, all the other derivatives of the free energy (in the saddle point equations) can be reduced to simple expressions involving this value, since for example for any $p \in \left\{\hat{q}, \delta\hat{q}, \hat{D}\right\}$:

$$\frac{\partial \mathcal{G}_S}{\partial p} = \int \mathcal{D}z_0 \frac{\partial}{\partial p}\left(\log\left(2\cosh\left(\text{arg}\right)\right)\right) = \int \mathcal{D}z_0 \frac{z_1^\star}{\sqrt{\delta q}} \frac{\partial}{\partial p}\left(\text{arg}\right) \tag{4.30}$$

Now we can finally obtain an entropy phase diagram, which can be seen in figure 4.2, showing that:

1. For all $\alpha < \alpha_c$, there is a neighborhood of $D = 0$ where $\mathcal{S}\,(D) > 0$, implying the existence of extensive clusters of solutions. Furthermore, for all α, the curves for $\mathcal{S}\,(D)$ are all approximately equal around $D = 0$; in particular, they all approximate the case for $\alpha = 0$ where all points are solutions. This implies that the clusters of solutions are extremely dense at their core.

2. For large distances, as expected, $\mathcal{S}(D)$ collapses with a second-order transition onto the equilibrium entropy, i.e. this regime is dominated by the typical solutions.

Finally we can say that there seems to exists sub-dominant dense regions of solutions, with a core whose local entropy almost saturates the total volume of configurations. These structures can be found up to a critical value of $\alpha_U = 0.755$, where the curves interrupt, signaling some symmetry breaking phenomenon (which reminds of the cluster disappearance transition observed in the numerical experiments); this phenomenon occurs before the SAT/UNSAT transition at $\alpha_C = 0.833$. This could explain why the efficient algorithms, attracted by this kind of structure, are unable to find solutions up to this threshold.

Problems with the large y limit Despite being simple and providing some insight into the phenomenon at study, the $y \to \infty$ limit proves to be quite problematic.

A thorough analysis shows that, in this limit, the replica symmetric Ansatz yields clearly unphysical results:

- all the interrupted curves (above $\alpha_U = 0.755$) actually break into two branches separated by an empty gap, reappearing at small distances $D \to 0$;
- The disappearance/appearance points correspond to stationary points for the derivative of the local entropy;
- in all these cases the local entropy measured at the value of D at which the right branch reappears is higher than the one measured at the end of the left branch (see figure **4.3**); this value is found to exceed also the value of the RS entropy at the same α, which is in contradiction with the sub-dominant nature of the dense cluster;
- the positive right branch is observed even when the value of α exceeds the known SAT/UNSAT threshold ($\alpha_c = 0.833$), even after the information paradoxical threshold $\alpha = 1$.

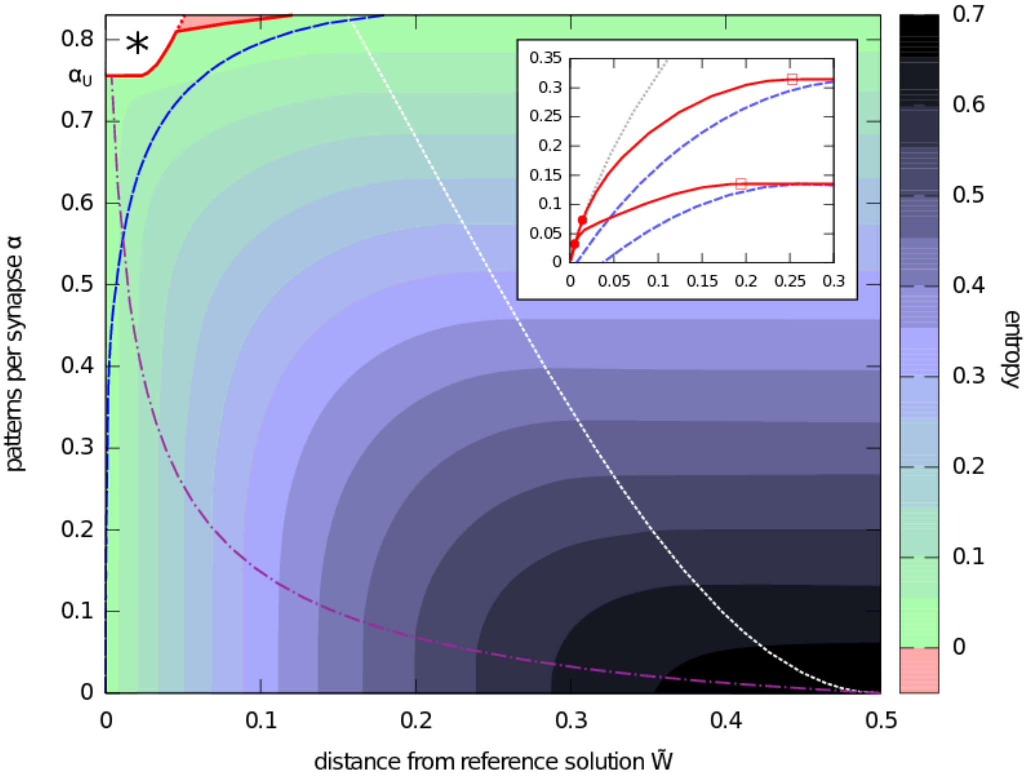


Fig. 4.2 **Entropy phase diagram**. Entropy levels are indicated by shaded areas (decreasing from bottom-right to top-left). The white area marked with an asterisk in the top-left corner denotes a region where there is no solution to the equations. Red solid curve: minimum distance, either due to a first-order transition (in α or d), or to the entropy becoming negative. White short-dashed curve: second-order transition to the RS solution (level curves to its right are horizontal). Purple dot-dashed curve: corresponds to a difference of 10^{-3} with the entropy of the $\alpha = 0$ case, so that the level curves to its left are quasi-vertical (high density of solutions). Blue long-dashed curve: minimal distance for typical $\tilde{W}$ (zero-entropy line). *Inset*: entropy vs distance, two examples: $\alpha = 0.5$ (bottom) and $\alpha = 0.7$ (top). Dotted gray curve: $\alpha = 0$ case. Red continuous curves: optimal $\tilde{W}$. Blue dashed curves: typical $\tilde{W}$. White squares: RS transition points (white line in main plot). Full dots: the curves tend to the $\alpha = 0$ case (purple curve in the main plot).

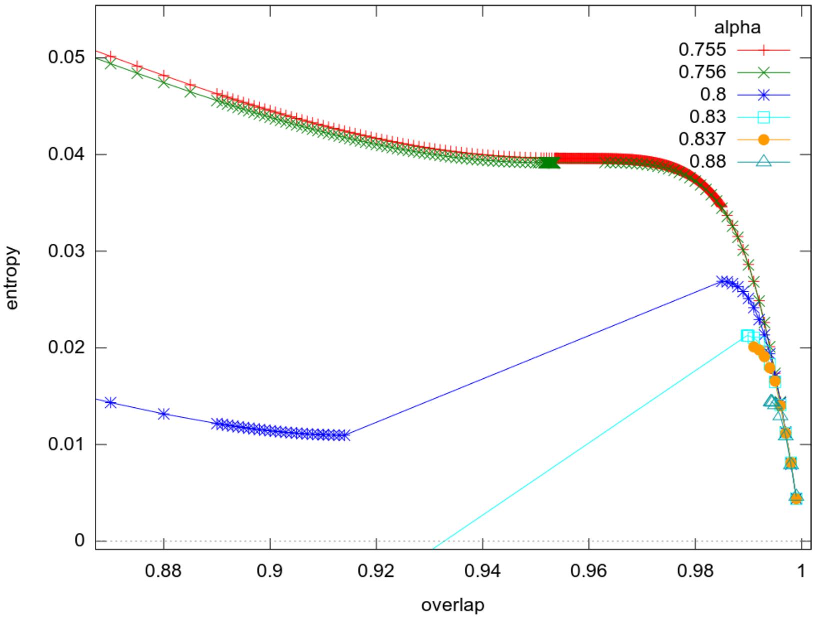


Fig. 4.3 **Unphysical branches** for the local entropy in the RS Ansatz and large y limit. In this case, even above the SAT/UNSAT threshold $\alpha_c = 0.833$ we find a right branch exhibiting an unrealistically high entropy of solutions at small distances. The gap between the branches is found to correspond to the region where the derivative $\partial \mathcal{S}(D,y)/\partial D$ becomes positive ($\partial \mathcal{S}(D,y)/\partial S$ negative).

All these observations point to the fact that the current Ansatz is incorrect, at least in this large y limit. One way of proving this could be to break the symmetry and see if the analytic results are stable to this modification in the Ansatz. Anyway, we can first check the value of the complexity, which should give a measure of the likelihood of observing this phenomenon in a real numerical experiment. If the complexity is very small $\Sigma \to 0^+$, it means that typically the number of clusters is of order $\mathcal{O}(1)$, but with negative complexities we are actually describing structures that appear with a probability that is depressed exponentially in the large N limit.

In order to extract the complexity, we need to consider the second order in the saddle point approximation for the free entropy density $\Phi(D,y)$. Consider an expression of the type:

$$I(y,\epsilon) = \frac{1}{y} \log \int Dz \, \exp\left(y\, f\left(\frac{z}{\sqrt{y}}, \epsilon \right) \right) \tag{4.31}$$

where we want to extract the second order saddle point approximation in the limit $y \to \infty$, when the parameters in f are perturbed by a quantity proportional to $\epsilon = \frac{1}{y} \to 0$. In order to obtain the $\mathcal{O}\left(\frac{1}{y}\right)$ correction to the leading term we need to consider two contributions: the first one comes from the second order in $I(y, 0)$, while the second correction comes from the first order in $I\left(y, \frac{1}{y}\right)$.

Consider for example:

$$f(z, \epsilon) = \log\cosh\left(\sqrt{\hat{q} + \hat{q}'\epsilon} z_0 + \sqrt{\delta\hat{q} + \delta\hat{q}'\epsilon} z + \tilde{l}\left(\hat{S} + \hat{S}'\epsilon\right)\right) \tag{4.32}$$

With $\epsilon = 0$, in the limit of large y, we have:

$$I(y, 0) = -\frac{(z^\star)^2}{2} + f(z^\star, 0) - \frac{1}{2y}\log\left(1 - \frac{\partial^2 f}{\partial z^2}(z^\star, 0)\right) \tag{4.33}$$

where $z^\star$ is the usual maximum, i.e.: $z^\star = \frac{\partial f}{\partial z}(z^\star, 0)$. The correction with $\epsilon = \frac{1}{y}$, instead, is simply given by:

$$I\left(y, \frac{1}{y}\right) = I(y, 0) + \frac{1}{y}\frac{\partial f}{\partial \epsilon}(z^\star, 0) \tag{4.34}$$

Overall the term at order y^{-1} has the form:

$$-\frac{1}{2}\log\left(1 - \frac{\partial^2 f}{\partial z^2}(z^\star, 0)\right) + \frac{\partial f}{\partial \epsilon}(z^\star, 0) \tag{4.35}$$

Therefore, by using the relationship $z^\star = \frac{1}{\sqrt{\delta\hat{q}}}\tanh\left(\sqrt{\hat{q}} z_0 + \sqrt{\delta\hat{q}} z^\star + \tilde{l}^\star \hat{S}\right)$, we get, for the entropic term:

$$G_S' = \int Dz_0 \left(-\frac{1}{2}\log\left(1 - \delta\hat{q} + (z^\star)^2\right) + \frac{1}{2\sqrt{\delta\hat{q}}} z^\star \left(\frac{\hat{q}'}{\sqrt{\hat{q}}} z_0 + \frac{\delta\hat{q}'}{\sqrt{\delta\hat{q}}} z^\star + 2\tilde{l}^\star \hat{S}'\right)\right), \tag{4.36}$$

while substituting $z^\star = \sqrt{\frac{\delta q}{1-q}}\mathcal{G}\left(\frac{-\sqrt{q}z_0 - \sqrt{\delta q}z^\star}{\sqrt{1-q}}\right)$ in the energetic contribution:

$$\begin{aligned} G'_E = \int Dz_0 \left(-\frac{1}{2}\log\left(1 + z^\star\left(z^\star\left(1 + \frac{\delta q}{1-q}\right) + \frac{\sqrt{q\delta q}}{1-q}z_0\right)\right) + \right. \\ \left. + \frac{1}{2}\left(\frac{z^\star\left(z_0\sqrt{\delta q\, q} + \delta q\, z^\star\right)}{1-q} + q'\frac{z^\star}{1-q}\left(\frac{z_0}{\sqrt{\delta q\, q}} + z^\star\right) + \delta q'\frac{\left(z^\star\right)^2}{\delta q}\right)\right) \end{aligned} \tag{4.37}$$

Thus, the overall complexity is given by:

$$\begin{aligned} \Sigma\left(D, y\right) = & -\frac{1}{2}\left(\delta\hat{q}\left(1-q\right) + \delta q\left(\delta\hat{q} - \hat{q}\right) + \delta\hat{q}'\, q + \delta\hat{q}\, q' + \hat{q}\left(\delta q' - q'\right) + \right. \\ & \left. + \hat{q}'\left(\delta q - q\right) + \hat{q}' + 2S\,\hat{S}'\right) + G'_S + \alpha G'_S \end{aligned} \tag{4.38}$$

Now we need to find the value of the perturbations of the order parameters, by setting to 0 all their derivatives, yielding:

$$\frac{\partial\Sigma}{\partial q'} = \frac{1}{2}\left(\hat{q} - \delta\hat{q}\right) + \frac{\alpha}{2\left(1-q\right)}\int Dz_0\, z^\star\left(z^\star + \frac{1}{\sqrt{q\,\delta q}}z_0\right) \tag{4.39}$$

$$\frac{\partial\Sigma}{\partial\delta q'} = -\frac{\hat{q}}{2} + \frac{\alpha}{2\delta q}\int Dz_0\,\left(z^\star\right)^2 \tag{4.40}$$

$$\frac{\partial\Sigma}{\partial\hat{q}'} = \frac{1}{2}\left(-\left(1-q\right) - \delta q\right) + \frac{1}{2\sqrt{\delta\hat{q}\,\hat{q}}}\int Dz_0\, z_0 z^\star \tag{4.41}$$

$$\frac{\partial\Sigma}{\partial\delta\hat{q}'} = -\frac{q}{2} + \frac{1}{2\delta\hat{q}}\int Dz_0\,\left(z^\star\right)^2 \tag{4.42}$$

$$\frac{\partial\Sigma}{\partial\hat{S}'} = -S + \frac{1}{\sqrt{\delta\hat{q}}}\int Dz_0\,\tilde{l}^\star z^\star \tag{4.43}$$

These are nothing but the old saddle point equations already obtained for the non-perturbed order parameters, so we find that all terms proportional to q', $\delta q'$, $\hat{q}'$, $\delta\hat{q}'$ and $\hat{S}'$ cancel out in the expression of Σ, as expected since the free entropy is variational. Thus, the complexity is given only by the terms which don't include the new order parameters:

$$\begin{aligned}\Sigma\left(D,y\right)=&-\frac{1}{2}\left(\delta\hat{q}\left(1-q\right)+\delta q\left(\delta\hat{q}-\hat{q}\right)\right)+\int Dz_{0}\left(-\frac{1}{2}\log\left(1-\delta\hat{q}+\left(z^{\star}\right)^{2}\right)\right)\\&+\alpha\int Dz_{0}\left(-\frac{1}{2}\log\left(1+z^{\star}\left(z^{\star}\left(1+\frac{\delta q}{1-q}\right)+\frac{\sqrt{q\,\delta q}}{1-q}z_{0}\right)\right)+\right.\\&\left.+\frac{1}{2}\frac{z^{\star}\left(z_{0}\sqrt{q\,\delta q}+\delta q\,z^{\star}\right)}{1-q}\right)\end{aligned}\tag{4.44}$$

In the large y limit the external entropy $\Sigma\left(D,y\right)$ is found to be negative for all values of the parameters, thus canceling out the unphysical results shown above. This signals a problem with the RS Ansatz, and implies that we should instead consider replica-symmetry-broken solutions. In geometrical terms, the interpretation is as follows: the RS solution at $y\to\infty$ implies that the typical overlap between two different reference solutions $\tilde{W}^{a}$ and $\tilde{W}^{b}$, as computed by $\tilde{q}=\frac{1}{N}\sum_{i}\tilde{W}_{i}^{a}\tilde{W}_{i}^{b}$, tends to 1, and therefore that there should be a single solution of maximal local entropy density. The fact that the RS assumption is wrong implies that the structure of the configurations of maximal density is more complex, and that, at least beyond a certain y, the geometry of the reference configurations $\tilde{W}$ breaks into several clusters.

Still, for a large range of values for α this limit is in good agreement with the more involved analyses and can provide some insight into the physical phenomena under study.

4.4 Finite y

Before giving up with the *RS* Ansatz, we can try to study the problem at finite y [1]: a motivation is given by the fact that the replica symmetric scenario is more directly comparable with the typical Franz-Parisi potential, providing the most straightforward way to demonstrate the radically different picture about the nature of the solutions painted by the large deviations analysis and the equilibrium case. Moreover, from the technical point of view, the 1RSB equations will produce a larger system of equations, involving multiple nested integrals that are very computationally expensive for arbitrary y.

In order to obtain a two-dimensional phase diagram, we need to choose a criterion for fixing the value of y while α and D are varied. We have seen that $\Sigma\left(D, y\right)$ needs to be positive in order to get physically meaningful results: the systems are discrete, so this quantity measures the logarithm of the number of reference configurations $\tilde{W}$ (at the given y and D) divided by N, and negative values would indicate rare events instead of typical instances [73].

It turns out that, for all values of α and D, there is a value of y beyond which $\Sigma\left(D, y\right) < 0$. Therefore, we can search for the value $y^{\star} = y^{\star}\left(\alpha, d\right)$ at which $\Sigma\left(D, y^{\star}\right) = 0$, i.e. the highest value of y for which the RS analytical results are consistent, representing structures that appear $\mathcal{O}\left(1\right)$ times in the space of solutions (as we expect the dense cluster to do). It is worth noting that following this criterion we still get $\tilde{q} < 1$, which implies that the number of reference solutions $\tilde{W}$ is larger than 1. For each couple of α and D, the sought value of the inverse temperature can be found by interpolating between different saddle point solutions at varying values of y.

From the results (shown in figure **4.4**), we observe that up to a certain α_U (where $\alpha_U \simeq 0.77$ in the classification case and $\alpha_U \simeq 1.1$ in the generalization case), the $\mathcal{S}\left(D\right)$ curves are monotonic in D. Beyond α_U, there is a transition in which there appear regions of D (dotted in figure 4.4) which are not correctly described by the RS Ansatz (since geometric bounds are violated, see the discussion in the SM for details), and must be described at a higher level of replica symmetry breaking (RSB). We speculate that this transition signals a change in the structure of the space of solutions: for $\alpha < \alpha_U$, the densest cores of solutions are immersed in huge connected structure; for $\alpha > \alpha_U$, this structure fractures and the dense cores become isolated and hard to find (see a sketch of this transition in figure 4.5).

Using the vanishing complexity criterion is sufficient to derive results which are geometrically valid across most values of the control parameters α and D. There are two exceptions to this observation, though, both occurring at high values of α and in specific regions of the parameter D. Let us indicate with $\left[D_L, D_R\right]$ these regions, with $0 < D_L < D_R < 1$:

- The most obvious kind of problem occurs occurs at $\alpha \gtrsim 0.79$, where $\mathcal{S}\left(D, y\right) < 0$ for $D \in \left[D_L, D_R\right]$. The standard treatment for this kind of problem is to break the replica symmetry, until the process reaches a level

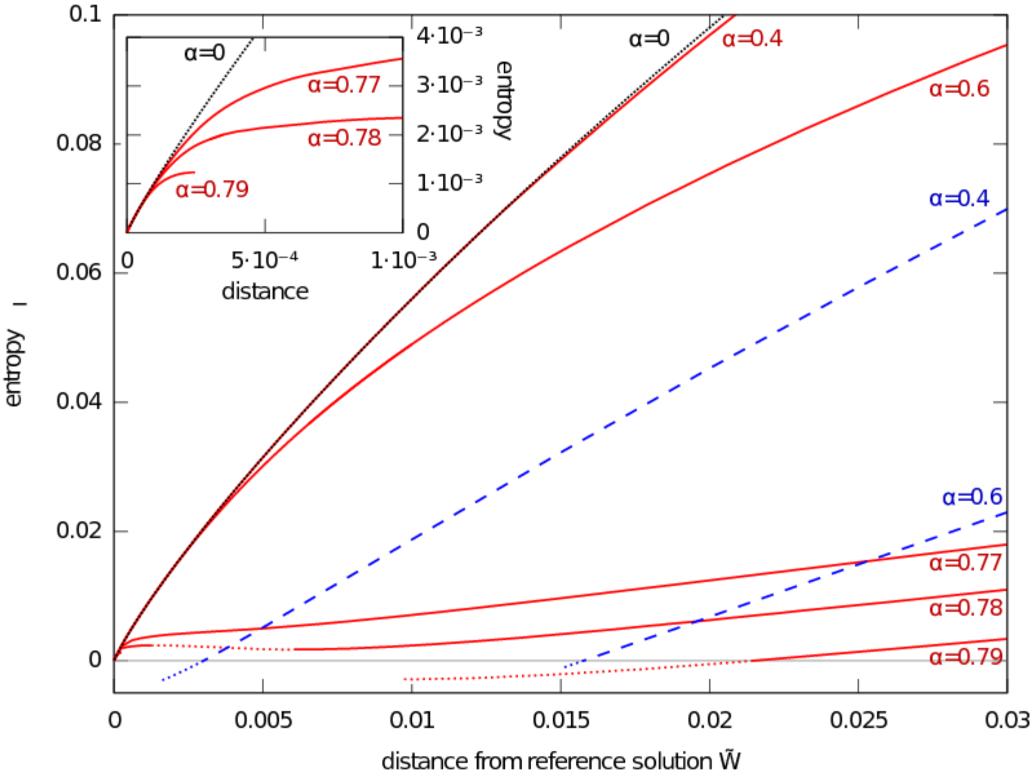


Fig. 4.4 **Local entropy** curves at varying distance d from the reference solution $\tilde{W}$ for various α (classification case). Black dotted curve: $\alpha = 0$ case (upper bound). Red solid curves: RS results. Up to $\alpha = 0.77$, the curves are monotonic. At $\alpha = 0.78$, a region incorrectly described within the RS Ansatz appears (dotted; geometric bounds are violated at the boundaries of the part of the curve with negative derivative). At $\alpha = 0.79$, the solution is discontinuous (a gap appears in the curve), and parts of the curve have negative entropy (dotted). Blue dashed curves: equilibrium analysis (typical $\tilde{W}$) [22] (dotted parts are unphysical): the curves are never positive in a neighborhood of $d = 0$. *Inset*: zoom of the region around $d = 0$ (notice the solution for $\alpha = 0.79$, followed by a gap).

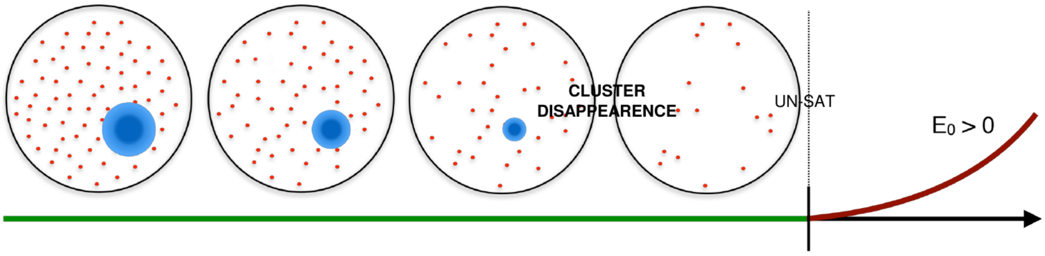


Fig. 4.5 **Sketch of the phase space** of the random binary Perceptron, as described by the large deviation analysis. The dense region of solutions disappears before the SAT/UNSAT transition.

where the local entropy remains positive or null for any α and D (in the equilibrium calculations a single step of symmetry breaking is sufficient, but this is not guaranteed to hold for this large deviation analysis).

- Another type of transition occurs between $\alpha \simeq 0.77$ and $\alpha \simeq 0.79$, where the $\frac{\partial}{\partial D}\mathcal{S}(D,y) \leq 0$ in $[D_L, D_R]$. A closer inspection of the order parameters reveals that, in this interval, $q_1 \geq S$. At the transition points $q_1 = S$, which is manifestly unphysical: paradoxically any of the solutions W (which are exponential in number, since $\mathcal{S} > 0$) could play the role of the reference solution $\tilde{W}$, yet the number of $\tilde{W}$ should be sub-exponential at $\Sigma = 0$. Clearly, those regions are inadequately described within the RS Ansatz.

As for the parts of the curves which are outside these problematic regions, the results obtained under the RS assumption are reasonable, and in very good agreement with the numerical evidence. In order to assess whether the RS equations are stable, further steps of RSB would be needed; unfortunately, this would multiply the number of order parameters and complicate the system of equations.

Since the extremal case is found only in the limit of $y = \infty$ (where the RS solution is inadequate) the values found for $\mathcal{S}(\Sigma = 0)$ might be seen as lower bound. However, when $D \to 0$ the sought value $y^{\star} \to \infty$. The same happens when $D \to (1 - q_{RS})/2$, where the distance constraint starts including the typical equilibrium solutions of the standard analysis and $\mathcal{S}$ becomes equivalent to the standard entropy of the equilibrium ground states.

4.5 Unconstrained case, 1RSB in the large y limit

We are also interested in considering the case in which the reference configuration $\tilde{W}$ is not constrained to be a solution. This scenario is vital for designing search algorithms where the local entropy optimization is substituted to the usual energy optimization (with the additional constraint on the reference we get a hybrid approach which is not easily implementable). It is possible to show that even when this explicit constraint is removed, if the local entropy is estimated at every step and the configuration for which it is maximum is obtained, in all likelihood the algorithm will end up in a solution (see [2]).

Unfortunately, in this case a finite y study is not sufficient for obtaining reasonable results when assuming replica symmetry, so a 1RSB Ansatz needs to be considered. Specifically, we presume that the symmetry breaking phenomenon occurs at the level of the $\tilde{W}$ variables (external replica symmetry breaking), while an RS description for the student variables W is kept. This appears as a geometrically consistent assumption, in that the clusters we are analyzing are dense and we do not expect any internal fragmentation of their geometrical structure (also in agreement with experiments of section **3.7**).

In order to simplify the computation of the saddle point equations, we also consider the limit $y \to \infty$. In this case the problem of the negative external entropy is not cured, as the complexity is found to be negative for all values of α and S. However, its magnitude is greatly reduced with respect to the analogous RS solution at $y \to \infty$; furthermore, its value tends to zero when $S \to 1$ (the region which is most crucial for proving the properties of the dense cluster), and all the other unphysical results of the RS solution appear to be fixed at this step of RSB.

Starting from expression 4.9 for the replicated volume in the constrained reweighted measure, we drop the constraint $\mathbb{X}_{\xi,\sigma}\left(\tilde{W}\right)$ on the reference configurations (thus getting rid of the parameters $\tilde{q}$, $\tilde{S}$ and their conjugates) and we opt for an external 1RSB Ansatz. With this scheme we describe a geometrical situation where the n replicas are organized in $\frac{n}{m}$ blocks of m replicas each. The Parisi 1RSB parameter m will be optimized alongside the other order parameters. We introduce the multi-index $c = (\alpha, \beta)$, where $\alpha \in \{1, ..., n/m\}$

labels a block of m replicas, and $\beta \in \{1,...,m\}$ indexes the replicas inside each block. The structure for the overlap matrix $q^{ca,db}$ becomes:

$$q^{\alpha\beta,a;\alpha'\beta',b} = \begin{cases} 1 & \text{if}\, \alpha = \alpha', \beta = \beta', a = b \\ q_2 & \text{if}\, \alpha = \alpha', \beta = \beta', a \neq b \\ q_1 & \text{if}\, \alpha = \alpha', \beta \neq \beta' \\ q_0 & \text{if}\, \alpha \neq \alpha' \end{cases} \tag{4.45}$$

and similarly for the conjugated parameter matrix $\hat{q}^{ca,db}$, while the Ansatz for the rest of the order parameters can remain unchanged from the RS case. Because of the presence of the reference configurations and their students, we will obtain an expression which is formally similar to a standard 2RSB description.

After some calculations one can obtain the following expression for the free entropy density $\Phi_{1RSB}\left(D,y\right)$:

$$\begin{aligned} \Phi_{1RSB}\left(D,y\right) \approx -\Big(& y^2\frac{m}{2}\hat{q}_0 q_0 - y^2\frac{m-1}{2}\hat{q}_1 q_1 - y\frac{y-1}{2}\hat{q}_2 q_2 - \frac{y}{2}\hat{q}_2 \\ & - y\hat{D}\left(1-2D\right) + \mathcal{G}_S + \alpha\mathcal{G}_E\Big) \end{aligned} \tag{4.46}$$

$$\mathcal{G}_S = \frac{1}{m}\int \mathcal{D}z_0 \log \int \mathcal{D}z_1 Z\left(z_0,z_1\right)^m \tag{4.47}$$

$$Z\left(z_0,z_1\right) = \int \mathcal{D}z_2 \sum_{\tilde{l}=\pm 1}\left[2\cosh\left(z_0\sqrt{\hat{q}_0} + z_1\sqrt{\hat{q}_1-\hat{q}_0} + z_2\sqrt{\hat{q}_2-\hat{q}_1} + \hat{D}\,\tilde{l}\right)\right]^y \tag{4.48}$$

$$\mathcal{G}_E = \frac{1}{m}\int \mathcal{D}z_0 \left\langle \log \int \mathcal{D}z_1 \left[\int \mathcal{D}z_2\; H\left(-\frac{z_0\sqrt{q_0} + z_1\sqrt{q_1-q_0} + z_2\sqrt{q_2-q_1}}{\sqrt{1-q_2}}\right)^y\right]^m \right\rangle_s \tag{4.49}$$

where the trivial saddle point equation $\hat{S} = \hat{D}$ was already substituted.

In the $y \to \infty$ limit we have to choose again a scaling for the overlap parameters: it is natural to set in this case $q_2 \to q_1 + \frac{\delta q}{y}$ and $\hat{q}_2 \to \hat{q}_1 + \frac{\delta \hat{q}}{y}$, where in the limit the replica symmetry is restored. Moreover, in order to maintain the correct scaling with y, we can $m \to \frac{x}{y}$. The $\int \mathcal{D}z_2$ integral can now be computed by a saddle point approximation, after rescaling the integration variable (similarly to the case of z_1 in the RS Ansatz), and to the leading order in y we find:

$$\lim_{y\to\infty} \Phi_{1RSB}\left(D,y\right) \approx \lim_{y\to\infty} -y\left(\frac{x}{2}(\hat{q}_0 q_0 - \hat{q}_1 q_1) - \frac{1}{2}(\delta\hat{q}\, q_1 + \hat{q}_1\, \delta q) - \frac{1}{2}\hat{q}_1\left(1-q_1\right) + \right.$$
$$\left. -\hat{D}\left(1-2D\right) + \mathcal{G}_S^\infty + \alpha\mathcal{G}_E^\infty\right) \tag{4.50}$$

$$\mathcal{G}_S^\infty = \frac{1}{x}\int \mathcal{D}z_0\ \log\int \mathcal{D}z_1\ e^{xA_S(z_0,z_1)} \tag{4.51}$$
$$A_S\left(z_0,z_1\right) = \max_{\tilde{l}=\pm1,z_2}\left\{-\frac{z_2^2}{2} + \log\left(2\cosh\left(\left(z_0\sqrt{\hat{q}_0} + z_1\sqrt{\hat{q}_1-\hat{q}_0} + z_2\sqrt{\delta\hat{q}} + \hat{D}\,\tilde{l}\right)\right)\right)\right\} \tag{4.52}$$

$$\mathcal{G}_E^\infty = \frac{1}{x}\int \mathcal{D}z_0\left\langle\log\int \mathcal{D}z_1\ e^{xA_E(s,z_0,z_1)}\right\rangle_s \tag{4.53}$$
$$A_E\left(s,z_0,z_1\right) = \max_{z_2}\left\{-\frac{z_2^2}{2} + \log H\left(-\frac{z_0\sqrt{q_0} + z_1\sqrt{q_1-q_0} + z_2\sqrt{\delta q)}}{\sqrt{1-q_1}}\right)\right\} \tag{4.54}$$

Differently from the previous case, in addition to α and D in this system of equations we have the control parameter x, which we can optimize on by requiring $\frac{\partial\Phi_{1RSB}}{\partial x} = 0$. When this saddle point condition is required the external complexity is naturally set to zero, as we required in the finite y study. In fact, we have:

$$m\Phi_{1RSB}\left(D,y\right) = \Sigma_0\left(D,y\right) + m\left(\Sigma_1\left(D,y\right) + y\mathcal{S}\left(D,y\right)\right), \tag{4.55}$$

where Σ_0 denotes the external complexity, Σ_1 the internal complexity and $\mathcal{S}$ is the (internal) local entropy. It is clear that when the condition $\partial\Phi_{1RSB}/\partial m = 0$ is required we directly get $\Sigma_0 = 0$. In the large y limit we are instead using $x = my$, so by posing $\Phi' = \Phi_{1RSB}/y$ we have (dropping the D and α dependence):

$$x\Phi' = \Sigma_0 + \frac{x}{y}\Sigma_1 + x\mathcal{S} = x\left(\Phi_0 + \Phi_1/y\right), \tag{4.56}$$

therefore $\partial\Phi/\partial x = 0$ implies $\Sigma_0 = 0$ (in this situation, though Σ_1, the complexity inside the cluster of $\tilde{W}$ is always negative). The system of saddle point equations is very sensitive to any change in x, so a good way of setting the

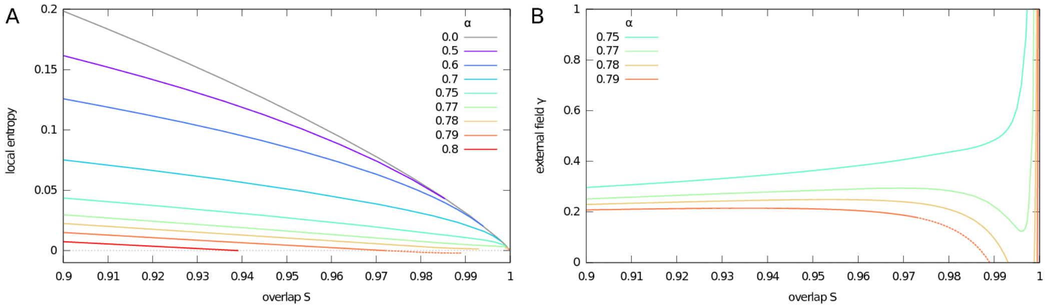


Fig. 4.6 **A. Local entropy vs overlap** S, at various values of α. All curves tend to the $\alpha = 0$ case for sufficiently high S. For $\alpha \gtrsim 0.77$, a gap appears, i.e. a region of S where no solution to the saddle point equations exists. For $\alpha \gtrsim 0.79$, some parts of the curve have negative entropy (dashed). **B. Relationship** between the overlap S and its conjugate parameter, the external field γ. Up to $\alpha \lesssim 0.75$, the relationship is monotonic and the convexity does not change for all values of S; up to $\alpha \lesssim 0.77$, a solution exists for all S but the relationship is no longer monotonic, implying that there are regions of S that can not be reached by using γ as an external control parameter. The gap in the solutions that appears after $\alpha \gtrsim 0.77$ is clearly signaled by the fact that γ reaches 0; below $\alpha_c = 0.83$, a second branch of the solution always reappears at sufficiently high S, starting from $\gamma = 0$.

associated derivative to 0 is to reach convergence of the other order parameters (by iterating the other equations) at fixed values of x, and then interpolate. The internal complexity Σ_1 can be found from a first-order expansion in y, giving:

$$\Sigma_1 = \frac{1}{2}\left(-\delta\hat{q} - \delta q\delta\hat{q} + \delta\hat{q}q_1 + \delta q\hat{q}_1\right) + \mathcal{C}_S^{\infty} + \alpha\mathcal{C}_E^{\infty} \tag{4.57}$$

$$\mathcal{C}_S^{\infty} = -\frac{1}{2}\int Dz_0 \frac{\int Dz_1\, e^{x\,\tilde{B}(z_0,z_1)} \log\left(1 - \delta\hat{q} + z_S^{\star}\left(z_0,z_1\right)^2\right)}{\int Dz_1\, e^{x\,\tilde{B}(z_0,z_1)}} \tag{4.58}$$

$$\mathcal{C}_E^{\infty} = -\frac{1}{2}\int Dz_0 \frac{\int Dz_1\, e^{x\,B(z_0,z_1)} \left(\log\left(1 + z_E^{\star}\left(z_0,z_1\right)^2 + b\left(z_0,z_1\right)\right) - b\left(z_0,z_1\right)\right)}{\int Dz_1\, e^{x\,B(z_0,z_1)}} \tag{4.59}$$

The phase diagram can be seen in figure 4.6. We also show the relationship between the overlap S and its conjugate parameter γ, which is easier to use in an algorithmic setting in which the local entropy is defined according to equation 4.3. Qualitatively, from the results of the 1RSB analysis we can observe that:

1. For all α below the critical value $\alpha_c = 0.83$, the local entropy in the region of $S \to 1$ tends to the curve corresponding to $\alpha = 0$, implying that for small enough distances the region around the ground states $\tilde{W}^\star$ is extremely dense (almost all points are solutions).

2. The internal complexity is always negative, but tends to 0 as $D \to 0$.

3. The unphysical branches of the local entropy (found for the RS Ansatz in the $y \to \infty$ limit) no longer exist above $\alpha_c = 0.833$.

4. $q_1 < S$ for all the saddle points. When $S \to 1$ (i.e. $D \to 0$, at small distances) we observe that $S \sim \sqrt{q_1}$, which means that $\tilde{W}$ is actually barycentric to its set of students W. Moreover $q_0 \neq q_1$ holds everywhere, so the number of optimal $\tilde{W}$ is larger than 1 (but sub-exponential).

5. There is a transition at $\alpha_U \simeq 0.77$ after which the local entropy curves are no longer monotonic; in fact, we observe the appearance of a gap in S where the system of equations has no solution. We speculatively interpret this fact as signaling a transition between two regimes: one for low α in which the ultra-dense regions are immersed in a huge connected structure, and one at high α in which the structure of the sub-dominant solutions fragments into separate regions.

6. As before, in the limit $y \to \infty$ the local entropy takes exactly the same value as for the constrained case in which the $\tilde{W}$ are required to be solutions, and the same is true for the parameters that are common to both cases. The external entropy, however, is different.

7. For the unconstrained case, we can compute the probability that the reference configuration $\tilde{W}$ makes an error on any one of the patterns (see figure 5.2). It turns out that this probability is a decreasing function when $D \to 0$ (going exponentially to 0) and an increasing function of α. For low values of α, this probability is extremely low, such that at finite values of N the probability that $\tilde{W}$ is itself a solution to the full pattern set is $\simeq 1$.

The simulation results, where available, seem to be in remarkable agreement with the predictions of this Ansatz (see 3 for the algorithmic thresholds and the numerical results on the disappearence of the cluster). Furthermore, the results of the 1RSB analysis at $y \to \infty$ and of the RS analysis of the constrained case at $y = y^{\star}(D)$ are both qualitatively and quantitatively very similar. We can speculate that this solution provides a reasonable approximation to the real behavior at $y \to \infty$.

The threshold α_U signals a transition between an "easy" and a "hard" computational phases, related to the accessibility of the dense regions of solutions that allow the existence of efficient algorithms able to solve the training task. The problem with the non monotonicity of the entropy curves will become clearer in the next chapter, where we will introduce a new solver based on the concept of the local entropy: in that setting we will employ a conjugate field in order to fix the distance D, and this is obviously no longer possible after α_U.

It is natural to make a comparison with other constraint satisfaction problems like random K-satisfiability (K-SAT), where in particular there is a "frozen" phase where solutions become isolated and no algorithm is known to work [32]. Contrary to the K-SAT example, however, in artificial neural networks this transition cannot be seen in the equilibrium analysis – which would predict that the problem is intractable at all values of α. This latter observation is presumably linked with the complex geometrical properties of the dense regions found by our large deviation analysis: they are not "states" in the usual sense (in the context of Statistical Physics of complex systems), since they are not clearly separated clusters of configurations.

Our analysis (theoretical and numerical) is not sufficient to completely characterize this peculiar geometrical structure: we know that is must be extensive, that the density seems to vary in a rather smoothly (i.e. such that it is algorithmically easy to find a path towards a solution in the local entropy landscape), and that there are several (but less than exponentially many) regions of highest density.

4.6 Generalized Perceptron case

Very similar results can be obtained also in the case of multiple synaptic states, $\{0,1\}$ patterns and various sparsity levels [3].

In figure 4.7, for example we can see the local entropy plotted against the distance in one representative case, for a Perceptron with synaptic states $l \in \{0,1,2,3,4\}$ and a coding level $f = 0.1$ for the sparse patterns, and compared with the Franz-Parisi potential at different values of α. The numerical solution of the saddle point equations becomes numerically extremely challenging around the transition point α_U, therefore some curves couldn't be completed. The most notable features that emerge from this figure are qualitatively equivalent to the binary case presented above:

- Typical solutions are isolated: the Franz-Parisi potential curve (typical $\tilde{W}$ in the figure) becomes negative in a neighborhood of $D = 0$, determining an extensive gap between neighboring solutions.
- Up to a certain $\alpha_U < \alpha_c$ (between 1.55 and 1.62 for the specific case in the plot), non-typical dense regions of solutions exist: at small distances the local entropy curves tend to collapse onto the $\alpha = 0$ curve, which corresponds to the upper bound where each configuration is a solution.
- Between α_U and α_c, there are regions of D where either there is no solution to the equations or the solution leads to a negative local entropy; both these phenomena indicate a change in the structure of the dense clusters, either disappearing or breaking into small disconnected and isolated components.

It would be interesting to determine numerically the transition point α_U, where the dense regions seem to disappear (or are at least no longer easily accessible), as a function of the number of states available to the synapses, measuring the effective gain in capacity of the device as they are added. This problem is extremely challenging from the computational point of view, due the time-consuming task of solving the system of equations that result from the replica analysis and to purely numerical issues related to the finite machine precision available and the trade-offs involved between computational time and

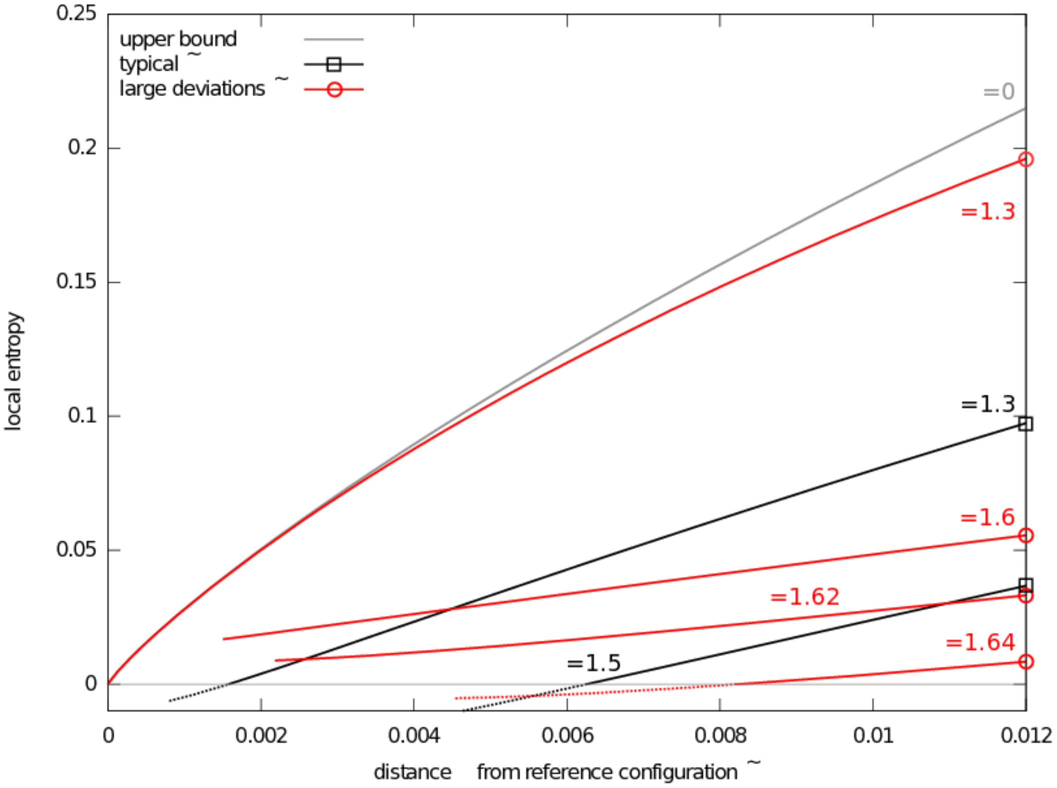


Fig. 4.7 **Local entropy density** as a function of the distance D from the reference configuration $\tilde{W}$, comparing the typical case from the Franz-Parisi analysis (black lines, marked with squares) with the large deviations case (red lines, marked with circles), at various values of the number of patterns per variable α. The upper bound (gray dashed curve) corresponds to the $\alpha = 0$ case where every configuration is a solution. The unphysical portions of the curves, where the local entropy becomes negative, are dotted. For the typical case, all curves eventually go below zero at some $D_{min} > 0$, for all values of α, i.e. typical solutions are isolated. For the large deviations case, the curves for the $\Phi\left(D, y^{\star}\left(D\right)\right)$ case (RS analysis) and the $\Phi_{1RSB}\left(D, \infty\right)$ case (1RSB analysis) yield results which are too close to be distinguished in the plot at this resolution. The "large deviations $\tilde{W}$" curve at $\alpha = 1.6$ is interrupted due to numerical problems in solving the equations, but it could continue up to $D = 0$, approaching the upper bound for small α. The curves for $\alpha = 1.62$ and $\alpha = 1.64$ are interrupted because the equations stop having solutions at some value of $D > 0$ (α_U transition, see text). The large deviations curve at $\alpha = 1.3$ is also essentially indistinguishable from the RS computation performed at $y = \infty$.

increased precision. All these problems are exacerbated near the transition point.

For this reason, the pathological RS analysis (in the limit $y \to \infty$) can be employed to provide an estimate of α_U, which can be obtained reasonably efficiently. As it can be seen in the binary case, this estimate is not too far away from the better one obtained in the much more expensive 1RSB Ansatz ($\alpha_U = 0.755$ against $\alpha_U \simeq 0.76$); in the case of the multi valued model of figure 4.7, the RS analysis at $y \to \infty$ gives $\alpha_U \simeq 1.6$ while the 1RSB analysis gives α_U between 1.55 and 1.62. Therefore, we can use the RS analysis at $y \to \infty$ to explore the behavior of α_U when varying the number of states and the coding level of the patterns. The transition is most easily detected by studying the derivative of the local entropy as a function of the distance $\partial_D \mathcal{S}\left(D, \infty\right)$ (α_U is found when it becomes tangent to the x axis, reaching the value 0).

It is a bit trickier to find the optimal value $z_1^\star$, in the entropic term, when the synapses can take more than two values, since the degeneracy of the stationary points is increased linearly. We can nevertheless simplify the search for the maximum by studying the argument of the function, see figure **4.8**:

$$\mathcal{G}_S = \int \mathcal{D}z_0 \max_{\tilde{l}} \left(\frac{\hat{\hat{Q}}}{y} \tilde{l}^2 + \max_{z_1} \left(-\frac{z_1^2}{2} + \log \left(f \left(\hat{Q}, \hat{q}, \delta\hat{q}, \hat{M}, S\tilde{l} \right) \right) \right) \right) \tag{4.60}$$

$$f \left(\hat{Q}, \hat{q}, \delta\hat{q}, \hat{M}, S\tilde{l} \right) = \sum_l \exp \left(\left(\hat{Q} - \frac{1}{2}\hat{q} \right) l^2 + \left(z_0 \sqrt{\hat{q}} + z_1 \sqrt{\delta\hat{q}} + \hat{M} + S\tilde{l} \right) l \right) \tag{4.61}$$

In figure 4.9A the behavior of α_U as a function of the number of states, for various values of the coding level f, is plotted. It is clearly similar to that of the critical capacity α_c, cf. figure 2.4. Since in the limit $L \to \infty$ the device should behave as a model with continuous synapses, we expect the ratio of these two thresholds to converge to 1, as it is found in figure 4.9B.

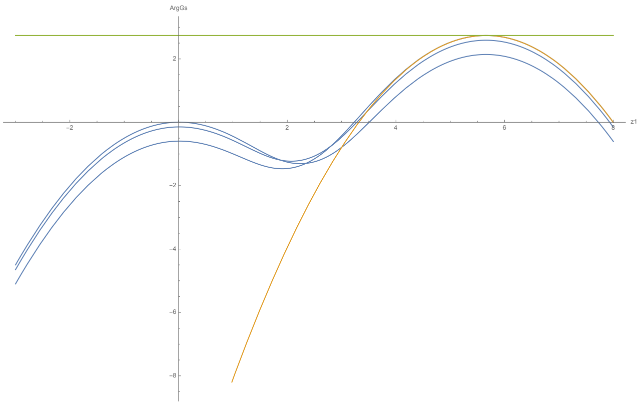


Fig. 4.8 **Study of the** $argGs$ **function** for multiple synaptic states: the parameters where set (randomly) to the values $\hat{Q} = 0.2$, $\hat{q} = 3.$, $\delta\hat{q} = 8.$, $\hat{D} = 0.30$, $z_0 = -2.5$, $l \in \{0, 1, 2\}$. The function $argGs$ can now have more than two local maxima for each value of $\tilde{l}$ (blue curves). The easiest way to find the maximum is to search explicitly for the couple $\{\tilde{l}, l\}$ that maximizes the quantity: $-\frac{\tilde{l}}{2}\hat{D} + (\hat{Q} - \frac{1}{2}\hat{q} + \frac{1}{2}\delta\hat{q})l^2 + (z_0\sqrt{\hat{q}} + \hat{D}\tilde{l})l$ (corresponding to the green tangent of the yellow parabola, that represents the maximal mode of $argGs$), and then to initialize Newton's algorithm at $z_1 = \sqrt{\delta\hat{q}}l$.

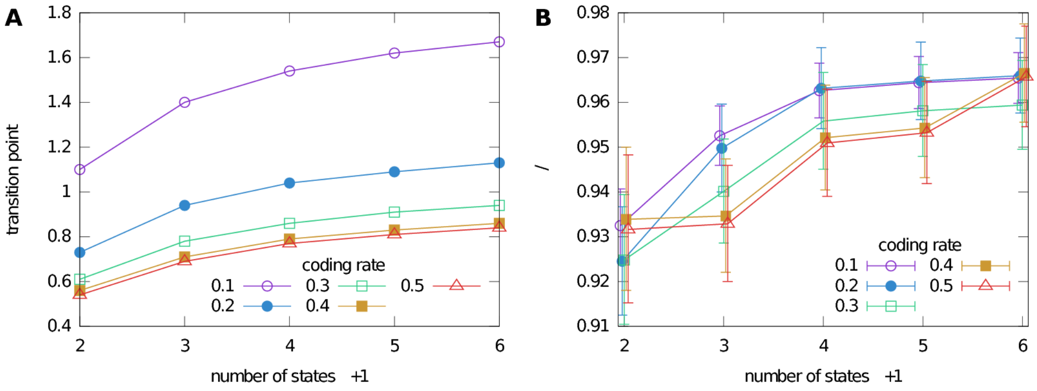


Fig. 4.9 **A. Transition point** α_U as a function of of the number of states per synapse $L + 1$, for different values of the coding rate f, as computed in the approximation of RS at $y \to \infty$. **B.** Same as panel A, but α_U is divided by the critical capacity α_c. Error bars indicate the errors induced by the precision with which the values were determined. Points for different values of f are slightly shifted relative to each other for improved legibility. Despite the limited number of values, a general tendency of this value to increase with L is observed (the ratio is expected to tend to 1 for $L \to \infty$), while the dependency on f is less clear.

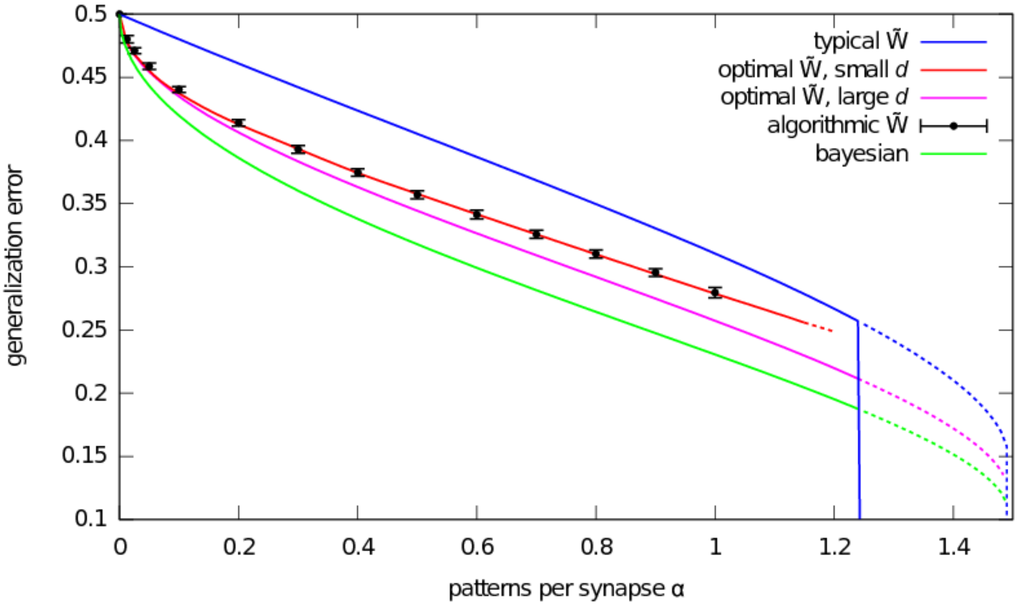


Fig. 4.10 **Generalization error (teacher-student scenario)**. From top to bottom: (blue) typical solution; (red) optimal $\tilde{W}$ at $d = 0.025$ (this solution disappears after $\alpha \simeq 1.2$); (black points) solutions from SBPI at $N = 10001$, 100 samples per point; (magenta) optimal $\tilde{W}$ at the value of d for which S_I is maximum (i.e. it equals the equilibrium entropy); (green) Bayesian case: error from the average over all solutions. At $\alpha_{TS} = 1.245$ is the first-order transition to perfect learning; between α_{TS} and $\alpha = 1.5$ there is a meta-stable regime; the dashed parts of the curves correspond to unphysical solutions of the RS equations with negative entropy.

4.7 Teacher-student scenario

The solution to the system of equations stemming from the RS saddle point produces qualitatively very similar results for both the classification (with $\alpha < \alpha_c$) and the generalization (with $\alpha < \alpha_{TS}$) case [1].

In the teacher-student scenario, the relevant quantity is the generalization error, i.e. the ability of giving a correct prediction on a newly presented pattern extracted from the same distribution of those in the training set. The analytical prediction for the generalization error rate is found to be simply dependent on the alignment of the student with the teacher: $p_e = \frac{1}{\pi} \arccos\left(\frac{1}{N} W \cdot W^{\mathrm{T}}\right)$.

As can be seen in figure 4.10, the generalization properties of the optimal reference solutions $\tilde{W}$ are generally much better than those of typical solutions.

Moreover, the curve for small D is found to be in striking agreement with the numerical results, produced using solutions obtained from the SBPI algorithm. The generalization error decreases monotonically when D is increased, and saturates to a plateau when $\mathcal{S}\left(D\right)$ equals equilibrium entropy (of the typical solutions).

This good generalization property might be justified with a Bayesian argument: given a new pattern-output association, the optimal Bayesian prediction is obtained by averaging the outputs of all the solutions of the training problem, as in:

$$P\left(\sigma|\xi_{new};\left\{\xi^{\mu},\sigma^{\mu}\right\}_{\mu=1}^{\alpha N}\right)=\int dW\,P\left(\sigma|W,\xi_{new}\right)P\left(W|\left\{\xi^{\mu},\sigma^{\mu}\right\}_{\mu=1}^{\alpha N}\right)\tag{4.62}$$

Since a solution in the sub-dominant cluster is immersed in a dense region of solutions, its output can be seen as a local Bayesian estimator of the output of its neighboring solutions. This means that the weight of this output in the full Bayesian prediction is likely larger than the output of a typical isolated solution, therefore the high (exponential) density guarantees good generalization properties.

Also in the case of multi-layer networks, the same qualitative scenario seems to hold: if one considers a random-walk constrained to the solutions of the training problem, the generalization properties of the starting solution (obtained with the extension of the CP+R algorithm) are found to be better than those of the neighboring solutions, found in later stages of the random walk, as it can be seen in figure 4.8.

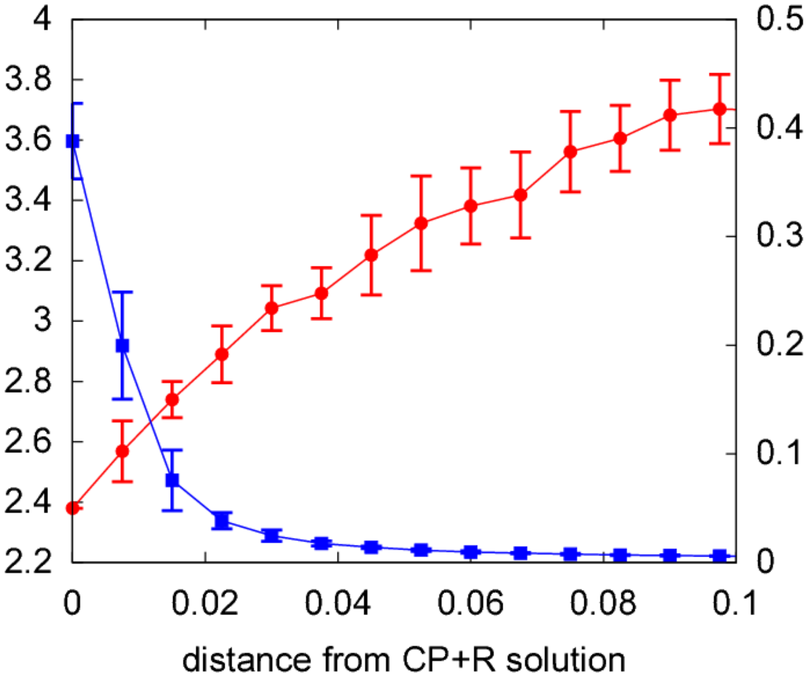


Fig. 4.11 **Generalization vs density** in a multi-layer network (with $K_1 = 11$, $K_2 = 30$, $r = 0$). Performing a random walk over solutions to the training set, one can observe that, moving away from this solution, the generalization error (red, circles) increases, and the solution density (blue, squares) decreases. The same qualitative behavior is observed with all network sizes.

Chapter 5

Entropy driven Monte Carlo

The results of the Large Deviation Analysis presented in the previous section finally put forward an explanation for the success of the heuristic solvers of the binary Perceptron. In a landscape dominated by frozen solutions, which cannot be found in sub-exponential time by local search strategies [66], the algorithms exploit the presence of a region in which the solutions accumulate and form a complex connected structure – branching with a decreasing density to the whole space of solutions – and sample solutions near the core of this cluster. The key ingredient for highlighting analytically these special structures was the introduction of a local entropy potential, that was used for enhancing the statistical weight of dense regions of solutions.

Now that we know what kind of solutions attract the efficient algorithms, we can devise more theoretically under-control solvers that explicitly target the dense cluster of solutions [2, 3]. Markov Chain Monte Carlo (MCMC) algorithms are often used in the context of combinatorial optimization for approximating the stationary distribution π of the studied problem. This distribution is monotonically decreasing with respect to the objective function to be minimized, and can be made more focused on the optima by tuning a properly introduced temperature, a simple procedure exploited in the Simulated Annealing algorithm [37]. Depending on the smoothness of the stationary distribution the sampling process can rapidly converge to low energy minima or it can get trapped in sub-optimal local minima of the loss-function. Typically, there is a trade-off between the optimality of the sampled configurations and the form of π: at high temperature, sampling from smooth and close to uniform distributions is usually easy, but the obtained configurations are most often far from optimal. On the other hand, as the statistical weight of the minima is increased by lowering the temperature, in hard optimization problems one has to deal with the emergence of a glassy landscape, where the number of meta-stable minima that can trap the MCMC, breaking the ergodicity of the sampling process, is exponential.

In section (**4.5** large dev: unconstrained case), we have seen that the dense cluster can be found even by lifting the requirement of selecting a solution as the reference configuration, since it is sufficient to look for configurations immersed in a zero energy configuration neighborhood. This fact suggests that it is possible to treat the local entropy as a pure objective function and to define a novel MCMC scheme, the *Entropy-driven Monte Carlo* (EdMC), where this new "energy" is maximized in a simple Metropolis procedure. The computation of the local entropy is obviously more involved than that of the energy, but EdMC is able to explore a smoother landscape (see figure **5.1**) than the one seen by a standard Simulated Annealing (SA), that is usually hindered by the proliferation of local minima. Moreover, EdMC offers a numerical method for validating the Replica calculations on single problem instances, and can help in understanding the properties of the heuristic BP-inspired algorithms that are able to find a solution in the binary Perceptron.

5.1 Energy of the reference configuration

An important question is what is the optimal radius – defining the neighborhood considered in the local entropy estimation – to choose in order to be confident that an algorithm like EdMC would eventually land on a solution. In order to address this question, we need to take a look at the behavior of the typical energy density of the unconstrained reference configuration $\tilde{W}$, as a function of the selected distance D or, equivalently, of the typical overlap $S = 1 - 2D$ (between the surrounding solutions and the reference).

The energy density can be easily related to the probability of classifying incorrectly a pattern $\xi^{\star}$, drawn uniformly at random from the training set.

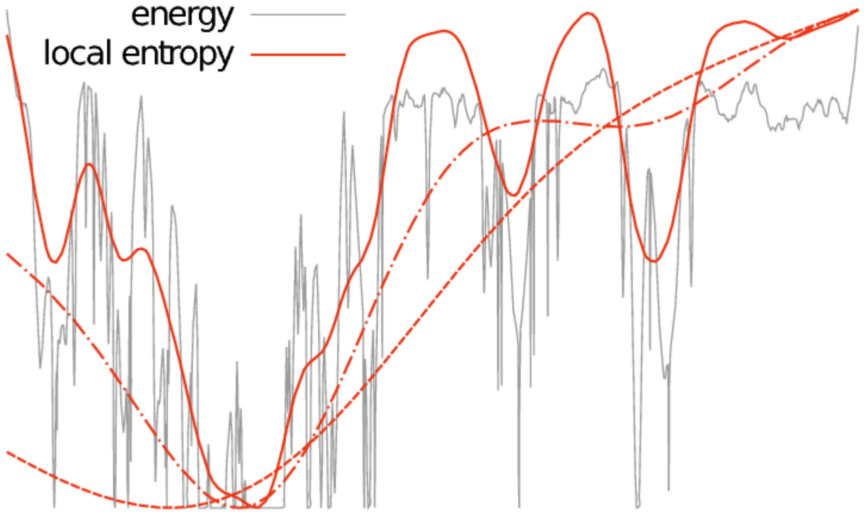


Fig. 5.1 **Energy landscape compared to local entropy landscape** in an illustrative toy example. The energy landscape (gray curve) can be very rugged, with a large number of narrow local minima. Some isolated global minima can also be observed on the right. On the left, there is a region of denser minima which coalesce into a wide global optimum. The red curves show the local entropy landscape (equation 5.7 with the opposite sign) computed at increasing values of the interaction parameter γ, i.e., at progressively finer scales. At low values of γ (dashed curve), the landscape is extremely smooth and the dense region is identifiable on a coarse-grained scale. At intermediate values of γ (dot-dashed curve) the global minimum is narrower and located in a denser region, but it does not correspond to a global energy minimum yet. At large values of γ (solid curve) finer-grain features appear as several local minima, but the global minimum is now located inside a wide global optimum of the energy. Note that in a high-dimensional space the isolated global minima can be exponentially more numerous and thus dominate the equilibrium measure, but they are “filtered out” in the local entropy description.

This quantity can be obtained by calculating:

$$P\left(\sigma^{\star}\neq 1\right)=\left\langle \Theta\left(-\frac{1}{\sqrt{N}}\sum_{i}\tilde{W}_{i}\xi_{i}^{\star}\right)\right\rangle_{\tilde{W}} \tag{5.1}$$

where the average is defined over the re-weighted unconstrained measure $d\mu_{W}\left(\tilde{W}\right)=d\mu\left(\tilde{W}\right)\mathcal{N}_{\xi}\left(\tilde{W},S\right)^{y}$. This calculation can be carried out straightforwardly by exploiting the replica trick, rewriting the ensemble average as:

$$\begin{aligned}\lim_{n\to 0}\int d\mu_{W}\left(\tilde{W}\right)\Theta\left(-\frac{1}{\sqrt{N}}\sum_{i}\tilde{W}_{i}\xi_{i}^{\star}\right)\left(\int d\mu_{W}\left(\tilde{W}\right)\right)^{n-1}=\\ \lim_{n\to 0}\int\prod_{c}d\mu_{W}\left(\tilde{W}^{c}\right)\Theta\left(-\frac{1}{\sqrt{N}}\sum_{i}\tilde{W}_{i}^{1}\xi_{i}^{\star}\right)\end{aligned} \tag{5.2}$$

We have thus introduced $n-1$ unconstrained replicas of the reference solution, leaving out the replica index 1 for the probing $\tilde{W}$-replica, coupled to the pattern $\xi^{\star}$ by the constraint. In this way one can first average out the quenched disorder, and then recover the initial expression in the $n\to 0$ limit.

When one extracts the overlaps referred to the reference configurations by introducing vanishing constraints (i.e. when $\gamma\to 0$), the conjugate parameters related to these overlaps tend to vanish as well. Therefore, if one organizes the calculation in the same way of the previously presented ones, the entropic terms cancel out and the only non-zero contribution to the average comes from the energetic part. The final expression, in the 1RSB Ansatz, is the following:

$$P\left(\sigma^{\star}\neq 1\right)=\int Dz_{0}\frac{\int Dz_{1}\left(\int Dz_{2}H\left(A\right)^{y}\right)^{m-1}\int Dz_{2}H\left(A\right)^{y}H\left(-C\right)}{\int Dz_{1}\left(\int Dz_{2}H\left(A\right)^{y}\right)^{m}} \tag{5.3}$$

with the definitions:

$$A\left(z_{0},z_{1},z_{2}\right)=\frac{z_{0}\sqrt{q_{0}}+z_{1}\sqrt{q_{1}-q_{0}}+z_{2}\sqrt{q_{2}-q_{1}}}{\sqrt{1-q_{2}}} \tag{5.4}$$

$$C\left(z_{0},z_{1},z_{2}\right)=\frac{z_{0}\frac{\tilde{S}_{0}}{\sqrt{q_{0}}}+z_{1}\frac{\tilde{S}_{1}-\tilde{S}_{0}}{\sqrt{q_{1}-q_{0}}}+z_{2}\frac{S-\tilde{S}_{1}}{\sqrt{q_{2}-q_{1}}}}{\sqrt{1-\frac{\tilde{S}_{0}^{2}}{q_{0}}-\frac{\left(\tilde{S}_{1}-\tilde{S}_{0}\right)^{2}}{q_{1}-q_{0}}-\frac{\left(S-\tilde{S}_{1}\right)^{2}}{q_{2}-q_{1}}}} \tag{5.5}$$

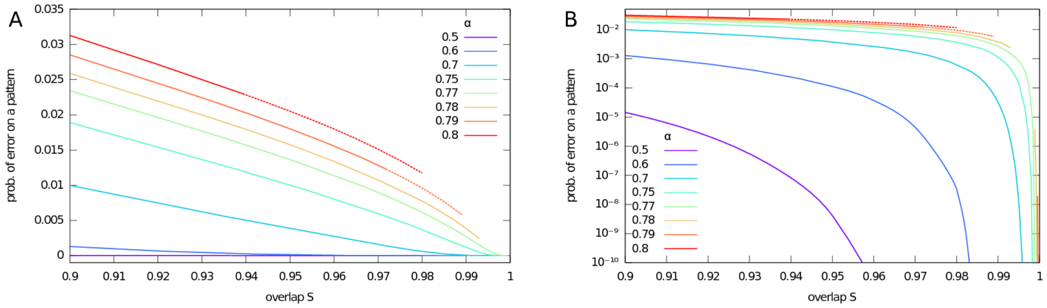


Fig. 5.2 **A. Probability of a classification error** by the optimal reference configuration $\tilde{W}$, for various values of α, as a function of S. The dashed parts of the curves correspond to the parts with negative local entropy (cf. figure 4.6); the curves have a gap above $\alpha \gtrsim 0.77$. **B.** Same as panel A, but in logarithmic scale on the y axis, which shows that all curves tend to zero errors for $S \to 1$.

In the limit $y \to \infty$, posing the same scaling behaviors as in section **4.3**, we get:

$$P\left(\sigma^{\star} \neq 1\right) = \int Dz_0 \frac{\int Dz_1 e^{x\,B(z_0,z_1)} H\left(-\frac{z_0 \frac{\tilde{S}_0}{\sqrt{q_0}} + z_1 \frac{\tilde{S}_1 - \tilde{S}_0}{\sqrt{q_1 - q_0}} + z_2 \frac{\delta S}{\sqrt{\delta q}}}{\sqrt{1 - \frac{\tilde{S}_0^2}{q_0} - \frac{\left(\tilde{S}_1 - \tilde{S}_0\right)^2}{q_1 - q_0}}}\right)}{\int Dz_1 e^{x\,B(z_0,z_1)}} \tag{5.6}$$

The analytic curves are plotted in figure 5.2, where we can see that, as long as the cluster exists, the probability of a classification error drops exponentially as the overlap S is increased to 1 (i.e. for small distances $D \to 0$ and a strong couplings $\gamma \to \infty$).

Unfortunately, if one starts the learning procedure from a random configuration directly at a high value for γ, the information provided by the local entropy is not sufficient for reaching the dense cluster. Therefore, in the same spirit of the annealing procedure employed in SA, we can initialize γ to a small value and devise a learning scheme in which the MC is slowly led to lower energy solutions by increasing the coupling γ, focusing the local entropy evaluation to smaller and smaller neighborhoods of the reference configuration and biasing the measure towards denser and denser regions of solutions. In the following, we call this special annealing the “scoping” procedure.

5.2 Implementation of the algorithm

A solution to the learning problem can thus be found by maximizing, at high enough γ, the local free entropy $\Phi\left(\tilde{W},\gamma\right)$:

$$\Phi\left(\tilde{W},\gamma\right)=\frac{1}{N}\log\mathcal{N}\left(\tilde{W},\gamma\right) \tag{5.7}$$

In this case, for algorithmic purposes, we choose to use γ, in the expression of $\mathcal{N}\left(\tilde{W},\gamma\right)=\sum_{\{W\}}\mathbb{X}_{\xi}\left(W\right)\mathrm{e}^{\gamma W\cdot\tilde{W}}$, to define a soft constraint for delimiting the region where the solutions are counted (as noted before, in the $N\to\infty$ limit it can be seen as the Legendre conjugate of the typical distance D between $\tilde{W}$ and the solutions $\{W\}$).

The main practical difficulty in implementing the EdMC algorithm is thus estimating the local free entropy $\Phi\left(\tilde{x},\gamma\right)$: a natural choice is to obtain it in the Bethe approximation given by the Belief Propagation (BP) algorithm. In order to determine $\Phi\left(\tilde{x},\gamma\right)$, one needs to study a slightly different system than the one defined by $H_0\left(W\right)$ (cf. with section **4.1**), in which the variables W are coupled to the external fields $\gamma\tilde{W}$, with $\tilde{W}_i\in\{-1,+1\}$ and $\gamma\in\mathbb{R}$:

$$H\left(W;\tilde{W}\right)=H_0\left(W\right)-\gamma\sum_{i=1}^{N}W_i\tilde{W}_i \tag{5.8}$$

Thus, in this expression, the directions of the external fields $\tilde{W}_i$ are treated as external control variables, and the parameter γ sets the magnitude of the external fields. The cavity magnetizations are thus modified, as in:

$$m_{i\to\mu}=\tanh\left(\sum_{\nu\neq\mu}\tanh^{-1}\left(m_{\nu\to i}\right)+\gamma\tilde{W}_i\right) \tag{5.9}$$

and similarly for the total magnetization: $m_i=\tanh\left(\sum_{\mu}\tanh^{-1}\left(\hat{m}_{\mu\to i}\right)+\gamma\tilde{W}_i\right)$.

The local free entropy $\Phi\left(\tilde{x},\gamma\right)$ is then obtained as the zero-temperature limit of the free energy of the system described by $H\left(x;\tilde{x}\right)$, and its definition is the same of equation 3.12, except for using the modified magnetizations. Similarly, with the new $\{m_i\}_{i=1}^{N}$ one can obtain an estimate of the average overlap S and the local entropy $\mathcal{S}$:

$$S\left(\tilde{W},\gamma\right)=\frac{1}{N}\sum_i \tilde{W}_i m_i \tag{5.10}$$

$$\mathcal{S}\left(\tilde{W},\gamma\right)=F\left(\tilde{W},\gamma\right)-\gamma S\left(\tilde{W},\gamma\right) \tag{5.11}$$

Therefore, the EdMC learning scheme can be defined as a very straightforward two-level optimization process: $\tilde{W}$ is initialized at random; at each step $\Phi\left(\tilde{W},\gamma\right)$ is computed by the BP algorithm; randomly chosen local moves (spin flips of the reference configuration $\tilde{W}$) are accepted or rejected using a standard Metropolis rule, at a temperature y^{-1}. After a number of accepted moves, we apply a "scoping" increment to γ (thus reducing D), until we eventually find a solution.

One could also consider an annealing procedure for y, but in practice, we found that the performance is good enough when it is kept fixed to a high value: in many regimes it is even possible to adopt a greedy strategy, setting $y=\infty$ and thus considering a zero temperature Monte Carlo. It seems to be more relevant, instead, to start from low values of γ and to increase it gradually.

5.3 Numerical results

Now that we have designed the EdMC optimization scheme, we can begin the numerical analysis by comparing the EdMC results, observed on finite size single-problem instances, with the theoretical predictions obtained via replica calculations. We therefore ran EdMC on a series of samples at size $N=201$ and $\alpha=0.6$. The scoping increment was determined by using the relationship between the coupling parameter γ and the resulting polarization in the magnetizations, $\gamma=\tanh^{-1}(p)$, and by implementing a linear increment in p ($p\in[0.4,0.9]$, in steps of 0.1).

In order to study the behavior of the free entropy $F(\tilde{x},\gamma)$ as a function of γ, we carried on with each EdMC simulation even when a solution was already found. The inverse temperature was both set to ∞, in a greedy version of the algorithm, and to a finite value, slowly incremented in an annealing procedure with an exponential rate of 1.01 (applied every 10 accepted moves).

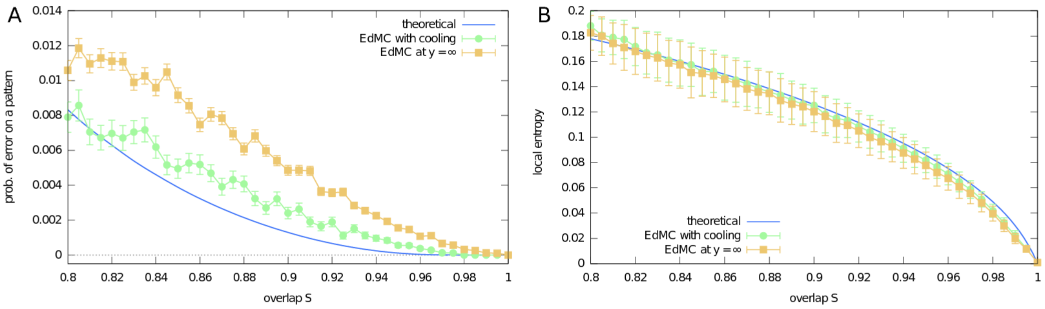


Fig. 5.3 **A. Probability of error** on a pattern (cf. figure 5.2) **B. Local entropy** (cf. figure 4.6A). See text for details on the procedure. For the version with cooling, 700 pattern sets were tested for each value of γ. For the $y = \infty$ version, 2000 samples were used. Error bars represent standard deviation estimates of the mean values.

The chosen stopping criterion was the consecutive rejection of $5N$ consecutive move proposals.

As we can see in figure 5.3, the recorded values of the local entropy $\mathcal{S}$ and of the error probability per pattern, as a function of the overlap S, are in good agreement with the theoretical analysis. The observed qualitative behavior is the same: the error rate goes to zero at $S \to 1$ and the entropy is positive until $S = 1$, as it should be in a dense cluster. We note that, in the more accurate version with an annealing in the inverse temperature y, the gap between the theoretical values and the measured ones is partially closed. The remaining discrepancy could be due to:

- finite size effects, since $N = 201$ is rather small;
- inaccuracy of the Monte Carlo sampling, which can be handled by lowering the cooling rate for y;
- inaccuracy of the theoretical curves due to additional RSB effects.

With the chosen settings, we note that the average number of errors per pattern set is almost always less than 2 for all points plotted in figure 5.3A, and that a zero energy configuration was always reached by EdMC. Also note that, in the plots, the noise recorded in the averages is only ascribable to the tails of the error distribution, while the modes and the medians are always found at 0.

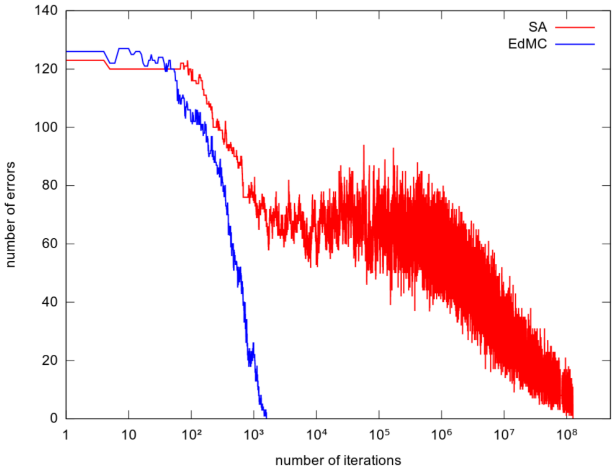


Fig. 5.4 **Typical trajectories** of standard Simulated Annealing (red curve, right) and Entropy-driven Monte Carlo (blue curve, left), for $N = 801$, $\alpha = 0.3$. Notice the logarithmic scale in the x axis. EdMC is run at 0 temperature with fixed $\gamma = \tanh^{-1}(0.6)$, SA is started at $y_0 = 1$ and run with a cooling rate of $f_y = 1.001$ for each 10^3 accepted moves, to ensure convergence to a solution.

We can now test the efficacy of our method as a solver, by comparing its performance, at various problem sizes N and different values of α, with a standard energetic MCMC. EdMC, in fact, shows a remarkable ability in retrieving solutions, even in a greedy zero temperature setting, with a relatively small ($\mathcal{O}(N)$) number of required MC steps. In the same setting the standard MCMC would get immediately trapped in a local minimum, even at small N. Instead, in order to find a solution also with the energetic MCMC, we employed a Simulated Annealing (SA) with initial inverse temperature $y_0 = 1$, increased by a factor f_y, every 10^3 accepted moves (the cooling rate f_y was optimized for each problem instance).

In figure 5.4 we show a comparison between a two typical trajectories, exemplifying the difference between standard SA and EdMC (at $y = \infty$ with fixed $\gamma = \tanh^{-1}(0.6)$) on the very same instance: the number of required accepted moves, in order to reach a solution with EdMC, is of 4 or 5 orders of magnitude smaller than the ones in the energetic SA. This highlights the qualitatively different smoothness of the landscape explored by EdMC.

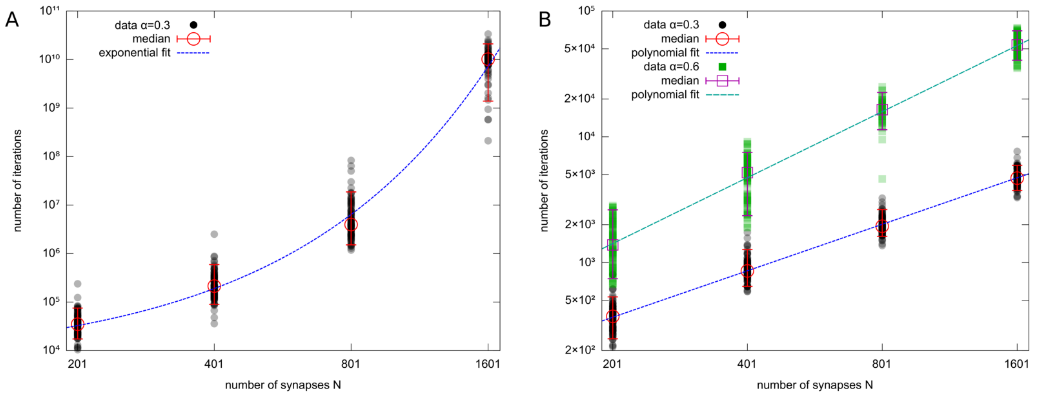


Fig. 5.5 **Number of iterations** required to reach 0 energy in log-log scale, as a function of the problem size N. **A: Simulated Annealing** at $\alpha = 0.3$, **B: EdMC** at $\alpha = 0.3$ (bottom) and $\alpha = 0.6$ (top). See text for the details of the procedure. Notice the difference in the y axes scales. For both methods, 100 samples were tested for each value of N. Color shades reflect data density. Empty circles and squares represent medians, error bars span the 5-th to 95-th percentile interval. The dashed lines are fitted curves: the SA points are fitted by an exponential curve $\exp(a + bN)$ with $a = 8.63 \pm 0.06$, $b = (8.79 \pm 0.08) \cdot 10^{-3}$; the EdMC points are fitted by two polynomial curves aN^b with $a = 0.54 \pm 0.04$, $b = 1.23 \pm 0.01$ for $\alpha = 0.3$, and with $a = 0.14 \pm 0.02$, $b = 1.74 \pm 0.02$ for $\alpha = 0.6$.

More importantly, a scaling analysis with the size of the network N (varied between between 201 and 1601), shows two radically different behaviors between the two MCMC strategies (as pictured in figure 5.4):

- the behavior of the standard SA is clearly exponential at $\alpha = 0.3$, and no solution can be found at $\alpha = 0.6$ already at $N = 201$.
- in the case of EdMC (panel B in the figure), instead, the gathered data can be fit by polynomial curves, with a scaling of $\sim N^{1.23}$ for $\alpha = 0.3$, and $\sim N^{1.74}$ for $\alpha = 0.6$.
- Even in the simple $\alpha = 0.3$ instances, a difference of several orders of magnitude was recorded in the required number of iterations in the two cases.

A detailed description of these numerical experiments can be found in the original paper ([2]).

5.4 Generalization to the Potts model

The existence of sub-dominant dense clusters of solutions was also proven in a discrete Perceptron with more than two synaptic states. Similarly to the binary case, we can therefore design an algorithm that exploits directly the local entropy information, obtaining radically different results from a normal SA on the energy function [3].

The only differences in the implementation of the EdMC algorithm, with respect to the binary case, are the following:

- In the BP equations, the reduction to a single magnetization associated to each synapse is no longer possible. Instead, one has to associate a pair of cavity messages to each available synaptic state.
- Since the $L2$ norm of the discrete synaptic configurations (parametrized by the order parameter Q in the replica calculations presented in section 2.5) is no longer trivially fixed to N, we cannot simplify the distance constraint $-\gamma/2\left(W-\tilde{W}\right)^2 \to \gamma W\cdot\tilde{W}$, as in the local entropy definition of equation 5.7. Therefore, when the external fields are applied, in order to plant the reference configuration $\tilde{W}$, the field intensity on each synaptic state will be given by $\gamma\left(W\cdot\tilde{W}-\left(W^2+\tilde{W}^2\right)/2\right)$.
- In order to be coherent with choice of the euclidean distance (instead of a Hamming distance), in the definition of the neighborhood determined by γ, the MC move proposal cannot be given by a uniform choice between the available synaptic states. We instead opt for proposing configurations $\tilde{W}'$, that are obtained by picking uniformly at random a synaptic index i and then randomly increasing or decreasing $\tilde{W}_i$ to the closest available states.
- Since the patterns considered in this case are sparse, $\xi_i^\mu, \sigma^\mu \in \{0,1\}$, we had to introduce a firing threshold θ, for balancing the output distribution. In both EdMC and the SA, θ was set *a priori* to its optimal value, determined analytically via replica calculations.

The tests performed in this case show that, while standard energetic SA (simply using the number of misclassified patterns as objective function) is immediately trapped by the exponentially large number of local minima, the entropic version can again reach a solution even in the greedy case ($y \to \infty$).

Figure 5.6 shows the results of a test on one sample for $N = 501$, $\alpha = 1.2$, $L = 4$, $f = 0.1$. Although the search space is considerably larger, the behavior of the algorithm is very similar to what was observed in [2] for the binary, balanced and unbiased case: while EdMC reaches 0 errors in a few iterations, standard SA plateaus and only eventually finds a solution, in several orders of magnitudes more iterations. Some heuristic enhancements, presented below, can further accelerate EdMC performance.

The considered scoping procedure was qualitatively similar to that of the binary case, even though the values of the couplings associated to small distances,$\gamma\left(D\right)$, are usually larger with respect to the former. In order to guarantee BP convergence in the early stages, we thus started with external fields of low intensity $\gamma = 0.5$ and progressively increased it, by $\Delta\gamma = 1.0$ after each greedy optimization procedure (a cycle through all the synapses $i = 1, ..., N$).

On the other hand in the SA we observed that, at high enough α, the standard Monte Carlo would get trapped even when implementing a very slow annealing in the temperature [21, 22]. We therefore decided to advantage the SA optimization, resorting to a more informative definition for the energy function, measuring the sum of the negative stabilities Δ (cf. with the Perceptron learning rule and section 3.2):

$$E_{\Delta}\left(\tilde{W}\right) = \sum_{\mu}\left(-\left(2\sigma^{\mu}-1\right)\left(\sum_{i}\tilde{W}_{i}\xi_{i}^{\mu}-\theta N\right)\right)_{+} \tag{5.12}$$

where $\left(x\right)_{+} = x$ if $x > 0$, 0 otherwise. In the annealing scheme we adopted a cooling rate of $r_y = 1.005$, applied every time 100 accepted moves are observed.

5.5 Accelerating the algorithm

Despite the fact that the number of MC steps is drastically reduced (with respect to standard SA) and scales well with the size N, the computational time required by the EdMC scheme can still be very high, due to the fact that

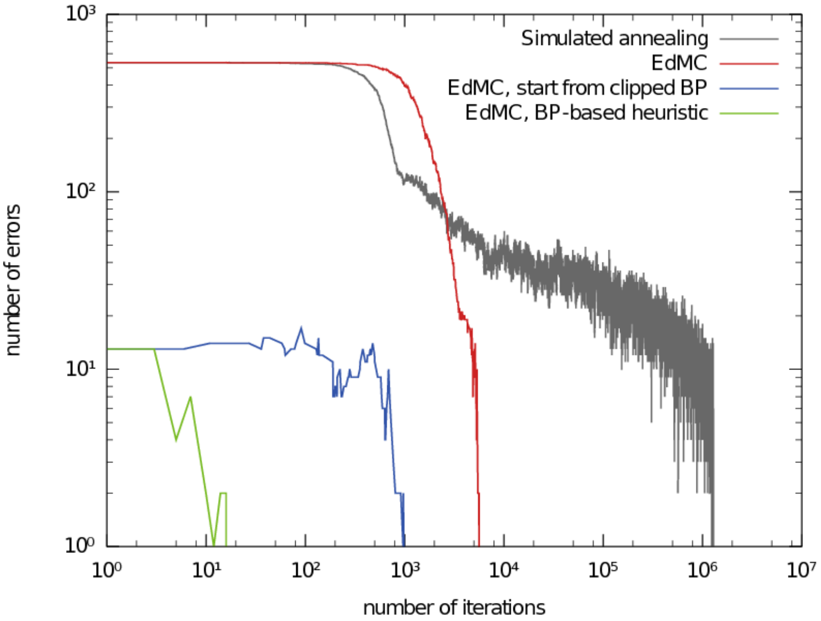


Fig. 5.6 **Comparison between different Monte Carlo-based solver algorithms** for one sample with $N = 501$, $\alpha = 1.2$, $L = 4$ and $f = 0.1$. The curves show in log-log scale the number of errors of the system as a function of the number of iterations (note that while the number of errors is used as the energy throughout the rest of the paper, none of the algorithms shown here uses it as its objective function). The curves shown are labeled from worst to best: simulated annealing on E_Δ (gray curve, see equation (5.12), more than 10^6 iterations required to find a solution); EdMC starting from random initial condition with zero-temperature dynamics (red curve, less than 10^4 iterations), EdMC using BP marginals as initial condition with zero-temperature dynamics (blue curve, less than 10^3 iterations); EdMC using BP marginals both as initial condition and to propose the Monte Carlo moves (green curve, less than 10^2 iterations). The local-entropy landscape is clearly much smoother than the energy landscape (even when using the energy E_Δ).

convergence of the BP algorithm has to be awaited at each move proposal. We can therefore introduce some heuristic modifications, which are able to greatly boost the speed of the algorithm.

Instead of starting from a random configuration, which could raise some numerical issues related to BP convergence (especially at high α), a good starting point can instead be found by using the information in the BP Replica Symmetric fixed point (which can be easily reached, in absence of external fields, for all $\alpha < \alpha_c$): the initial configuration can thus be chosen as $\tilde{W} = \text{sign}\left(m_{RS}\right)$ in the binary Perceptron. In the Potts Perceptron case, instead, one can assign to $\tilde{W}_i$ the argmax between the RS state marginal probabilities $u_i\left(W\right)$.

Moreover, it is also possible to use the BP fixed point messages for the proposal of efficient MC moves, rather than performing them at random. The idea is that an extensive number of synaptic flips (in the direction of maximum free energy) can be done all at once in a single step. Each time BP reaches convergence, one can identify the set of synapses whose cavity marginals m_i, *in absence* of the external field that plants the reference configuration, are not in agreement with the direction $\tilde{W}_i$ of the field itself. Once this set of synaptic indices is ranked, so that the ones associated to the largest difference between the external and the cavity fields come first, we can propose a collective move changing all the identified synapses and compute the new value of F. As in a normal Metropolis algorithm, the move is always accepted if there is an increase in the free energy, or with probability $e^{y\Delta F}$ otherwise. If the collective flip is accepted, the heuristic procedure can move on, computing a new set of synaptic variables to be changed with the help of BP marginals. If, on the contrary, the collective move is rejected, the new proposal can be on the reduced set where the last ranked variable is removed; this procedure can be repeated until the set is empty, and in the end one goes back to the standard EdMC scheme. We observed that most of these collective moves are immediately accepted. The physical interpretation is that, in this way, one tries to maximize, at each step, the local contributions to the Bethe free energy associated to each variable W_i, in presence of an external field $\gamma\tilde{W}_i$.

Finally, in case BP was unable to reach convergence, one can still obtain some information from $\overline{F}$, obtained from the time average of the cavity marginals

```
**Input:** problem sample; parameters $t_{\max}$, $t_{\text{step}}$, $y$, $\gamma$, $f_y$ and $f_\gamma$
;
Randomly initialize $\tilde{x}_i^0$. Alternalively, run BP with $\gamma = 0$ and set
  $\tilde{x}_i^0 = \text{sign}(h_i)$;
Run BP with external fields $\gamma \tilde{x}_i^0$;
Compute free energy $F^0$ from BP fixed point ($\bar{F}^0$ if BP does not
  converge);
$t \leftarrow 0$;
while $t \leq t_{\max}$ do
    Retrieve fields $h_i^t$ ($\bar{h}_i^t$ if BP did not converge);
    for $i = 1$ to $N$ do $\Delta_i \leftarrow \tilde{x}_i^t \left(\gamma \tilde{x}_i^t - h_i^t\right)$;
    Collect $V = \{i \mid \Delta_i > 0\}$ and sort it in descending order of $\Delta_i$;
    accepted $\leftarrow$ FALSE;
    while NOT accepted do
        Propose a flip of the $\tilde{x}_i^t$ for all $i \in V$, producing $\tilde{x}^{t+1}$;
        Run BP with new proposed external fields $\gamma \tilde{x}_i^{t+1}$;
        Compute free energy $F^{t+1}$ from BP fixed point ($\bar{F}^{t+1}$ if BP does
          not converge);
        with probability $e^{y\left(F^{t+1} - F^t\right)}$ do accepted $\leftarrow$ TRUE;
        if NOT accepted then
            Remove the last element from $V$;
            if $|V| = 0$ then exit and run EdMC with $\tilde{x}^t$ as initial
              configuration;
        end
    end
    $t \leftarrow t + 1$;
    Compute energy $E$ of configuration $\tilde{x}^t$;
    if $E = 0$ then retrieve solution $\tilde{x}^* = \tilde{x}^t$ and exit;
    if $t \equiv 0 \pmod{t_{\text{step}}}$ then
        Annealing: $y \leftarrow y \times f_y$;
        Scoping: $\gamma \leftarrow \gamma \times f_\gamma$ (run BP and update $F^t$);
    end
end
```

Algorithm 1: Heuristic EdMC with *Annealing* and *Scoping*.

(over a few BP iterations), which can be useful for bootstrapping the EdMC to a region where convergence can be obtained more easily.

With the implementation of the heuristic modifications, the EdMC algorithm turns out to be extremely fast and capable of solving hard instances, achieving a higher algorithmic threshold α_U. The resulting algorithm is detailed in Algorithm 1.

5.6 Generalization to multi-layer continuous networks: Entropy SGD

The idea of searching for regions with a high local entropy of zero energy configurations cannot be directly exported to the continuous case: first of all, the usual losses $\mathcal{L}$ that drive the learning processes in the continuous case are not bounded below and do not allow for a clear analogous of the zero energy requirement; secondly, the concept of entropy itself is not well defined as in the case of discrete sets of configurations.

However, it is still possible to find a connection between local geometric properties of the objective function and the generalization performance of the solutions, found by the learning algorithms employed in deep learning [74, 75]. It seems that a quite general qualitative picture holds in most cases, independent of the chosen neural network architecture and heuristic solver: there is a correlation between the well generalizing synaptic configurations and the local minima that reside in “wide valleys” of the energy landscape, rather than in sharp isolated wells. A possible measure of this geometric property can be obtained through the study of the Hessian of $\mathcal{L}$, and in [74] it was observed that in correspondence of “good” solutions the landscape looks flat in most of the directions.

Similar to what we sketched in section 4.7, also in [74] the proposed intuitive explanation for this phenomenon comes from a Bayesian argument: in a full Bayesian approach, where the prior is concentrated on the configurations with minumum expected loss, the weight of wide valleys is much larger than that of narrow, sharp valleys at a similar level in the training loss function. The analogy with the discrete case also extends to the algorithmic side: while most heuristic solvers are biased “by hand” towards well generalizing solutions, through the introduction of noise-robustness requirements and by perturbing the networks during the training phase, as a byproduct they also end up in wide valleys without any explicit indication.

Entropy Stochastic Gradient Descent (ESDG) is a learning algorithm introduced in [74], as the analogous of EdMC in the context of multi-layer continuous networks: instead of simply minimizing the original loss $\mathcal{L}(\tilde{W})$, the proposed algorithm maximizes a local entropy function:

$$F\left(\tilde{W},\gamma\right) = \log \int_{W\in\mathbb{R}^{\mathbb{N}}} \exp\left(-\mathcal{L}\left(W\right) - \frac{\gamma}{2}\left(W-\tilde{W}\right)^2\right) dW \tag{5.13}$$

Similarly to what happens in the EdMC, the practical problem is that of evaluating this quantity (or its gradient): as in the former case, the answer comes from a MCMC method, the Stochastic Gradient Langevin Dynamics (SGLD), which is the suitable choice in this continuous setting. The resulting algorithm is again a two-level optimization process that, through a scoping procedure in γ, eventually lands on solutions with low generalization errors. The numerical results show that state-of-the-art generalization performance can be achieved in various benchmarks and with different neural network architectures, together with a boost (about x2) in the speed of the learning procedure with respect to a standard SGD.

Chapter 6

Robust Ensembles

In the last section, we have seen that the introduction of the local entropy density, crucial for discovering dense regions of solutions in the phase space of the discrete Perceptron model, is able to turn an apparently hard learning problem into an easy one: the landscape explored by simple optimization algorithms like Simulated Annealing is drastically modified, and the process is pulled towards accessible low-energy configurations. The main practical difficulty, associated with this straightforward strategy, comes from the high computational cost of evaluating the local density of solutions $\mathcal{S}$, a quantity whose precise measure would require exponential times. The simplest solution to this problem was to estimate it in the Bethe approximation, through the Belief Propagation algorithm, which brought us to the definition of the Entropy driven Monte Carlo scheme [2], a two-level optimization procedure which proved to be successful in finding solutions that belong to the sub-dominant connected cluster, overtaking the challenges posed by the glassy landscape of the Perceptron. A very similar strategy was also implemented in the context of continuous multi-layer neural networks [74]: in this case the average over the local entropy measure was approximated via a Stochastic Gradient Langevin Dynamics.

All the solutions found by these “entropic” algorithms share some interesting properties: they are *rare* in the space of solutions (i.e., not emerging in a standard equilibrium description) yet *algorithmically accessible* (i.e., attractive for the efficient heuristics), and *robust* (i.e., immersed in dense regions of “good” configurations). Moreover, there is a relation between the robustness of solutions and their good generalization ability: this can be intuitively understood in a Bayesian framework, where the robust solutions can be seen as representatives of extensive sets of configurations surrounding them.

The main question we now want to address is whether it is possible to avoid this two-level strategy formulation, while still targeting the same local entropy measure which proved to be a crucial ingredient for achieving good learning performance [4]. Moreover, we also aim at providing a general learning paradigm which could be applied to any given learning algorithm, e.g. Simulated Annealing (SA), Stochastic Gradient Descent (SGD) and Belief Propagation (BP), effectively turning energy-based local search strategies into local-entropy-based ones.

6.1 Real replicas

In statistical physics, the canonical ensemble is usually introduced to describe the equilibrium (i.e., long-time limit) properties of a stochastic process, in terms of a probability distribution over the configurations W of the system:

$$P\left(W;\beta\right)=Z\left(\beta\right)^{-1}\exp\left(-\beta E\left(W\right)\right), \tag{6.1}$$

where $E\left(W\right)$ is the energy of the configuration, β is an inverse temperature, and the normalization factor $Z\left(\beta\right)$ is the partition function of the model. In particular, in the context of optimization problems, $E\left(W\right)$ plays the role of a cost function to be minimized, and one is interested in the limit $\beta\to\infty$, where a uniform weight is assigned to the sought global minima of the energy function.

Unfortunately, in some special cases, this standard equilibrium description might be insufficient for capturing the relevant structures which, despite being “hidden” in the vast landscape, are specifically targeted by effective optimization strategies [1]. This motivated the introduction of a different measure, which ignores isolated solutions and enhances the statistical weight of large, accessible regions of solutions:

$$P\left(\tilde{W};\beta,y,\gamma\right)=Z^{-1}\left(\beta,y,\gamma\right)e^{y\,\Phi\left(\tilde{W},\beta,\gamma\right)}. \tag{6.2}$$

Here y is a parameter that has the formal role of an inverse temperature and $\Phi\left(\tilde{W},\gamma,\beta\right)$ is a "local free entropy":

$$\Phi\left(\tilde{W},\beta,\gamma\right)=\log\sum_{\{W\}}e^{-\beta E(W)-\frac{\gamma}{2}\,d\left(W,\tilde{W}\right)} \tag{6.3}$$

where $d\left(\cdot,\cdot\right)$ denotes a distance between configurations, which needs a proper definition according to the model under consideration. Note that, in this new statistical measure:

- the limit $\beta\to\infty$ corresponds (up to an additive constant) to measuring a "local entropy", i.e. counting the number of minima of the energy around the reference configuration $\tilde{W}$, and weighting them (via the parameter γ) by the distance $d(W,\tilde{W})$.
- At large values of y, only the configurations $\tilde{W}$ that are surrounded by an exponential number of local minima will have a non-negligible weight.
- By increasing the value of γ, it is possible to focus on narrower neighborhoods around $\tilde{W}$: at large values of γ also the reference $\tilde{W}$ is expected (with high probability) to share the same properties of the surrounding minima [2].

Suppose now that we take y to be a (non-negative) integer: in this case we can rewrite the partition function of the large deviation distribution equation (6.2) as:

$$\begin{aligned}Z\left(\beta,y,\gamma\right)=&\sum_{\tilde{W}}e^{y\,\Phi\left(\tilde{W},\beta,\gamma\right)}= \\ &\sum_{\tilde{W}}\sum_{\{W^a\}_{a=1}^{y}}e^{-\beta\sum_{a=1}^{y}E(W^a)-\frac{\gamma}{2}\sum_{a=1}^{y}d\left(W^a,\tilde{W}\right)}\end{aligned} \tag{6.4}$$

This partition function describes a system of y identical "real" replicas of the system, subject to their usual energies $E\left(W^a\right)$ but also interacting with the reference system $\tilde{W}$. The adjective *real* is here employed to stress the fact that

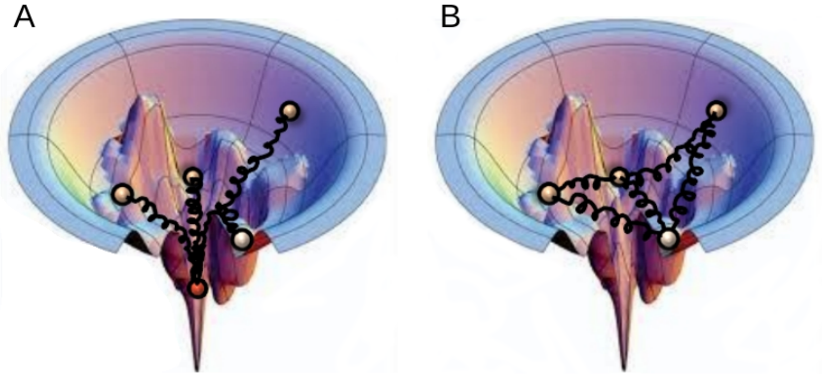


Fig. 6.1 **Sketch of the Robust Ensemble** optimization process. In plot A, we can see the real replicas exploring the loss landscape while being attracted towards the reference configuration through an elastic interaction (as in equation 6.4). In plot B, the reference configuration is integrated out, producing an effective interaction between the real replicas (as in equation 6.5).

the y replicas are not to be confused with the *virtual* replicas usually introduced in the "replica trick": we are not interested in taking the limit $y \to 0$, since the actual model comprises y identical interacting objects. Once an expression for all integer y is found, the general case of $y \in \mathbb{R}$ can be recovered by analytic continuation. The equilibrium statistics of this enlarged system are thus exactly equivalent to the original large deviation analysis, provided the replicas W^a are eventually traced out.

This new way of rewriting the expression for the local entropy measure suggests a simplified recipe for explicitly targeting the dense states: introduce y replicas of the model, add an interaction term depending on the distance with respect to a reference configuration, and run the algorithm over the resulting extended system. In fact, in most cases, this scheme can be further improved upon by directly tracing out the reference $\tilde{W}$, obtaining a system of y identical interacting replicas with a slightly more complicated mutual coupling (a sketch can be seen in figure 6.1), describing the *robust ensemble* (RE):

$$Z(\beta, y, \gamma) = \sum_{\{W^a\}} e^{-\beta\left(\sum_{a=1}^{y} E(W^a) + A(\{W^a\},\beta,\gamma)\right)} \tag{6.5}$$

$$A(\{W^a\}, \beta, \gamma) = \frac{1}{\beta} \log \sum_{\tilde{W}} e^{\frac{\gamma}{2} \sum_{a=1}^{y} d\left(W^a, \tilde{W}\right)} \tag{6.6}$$

The main advantage of considering an optimization procedure in the replicated system is that it avoids the need to use BP for the local entropy estimation, which makes this algorithmic procedure much simpler and more general [4].

6.2 Connection with other RSB methods

Let us go back to expression 6.3, for the local free entropy, and consider the large-deviation partition function associated to a central central configuration $\tilde{W}$, who is also subjected to the standard energy function $E\left(\tilde{W}\right)$:

$$Z\left(\beta,\beta',y,\gamma\right)=\sum_{\tilde{W}}e^{-\beta E\left(\tilde{W}\right)+y\phi\left(\tilde{W},\gamma,\beta'\right)} \tag{6.7}$$

Note that, in general, the two temperatures β and β' can be different. We can thus start by expanding this expression, setting $y\in\mathbb{N}$ (as above) to obtain the replicated system, and then trace out the reference configuration $\tilde{W}$: the result, in the special case $\beta=0$ (i.e. with an unconstrained reference), reads:

$$Z\left(0,\beta',y,\gamma\right)=\sum_{\{W^a\}}e^{-\beta'\sum_{a=1}^{y}E(W^a)+\log\sum_{\tilde{W}}\exp\left(-\frac{\gamma}{2}\sum_{a=1}^{y}d\left(W^a,\tilde{W}\right)\right)} \tag{6.8}$$

This expression can be directly compared with the derivation, proposed in Ref. [76], of an ergodicity-breaking scheme that reproduces the 1RSB Ansatz. In that work, an auxiliary pinning field (playing the same role of the “planted” reference $\tilde{W}$) was introduced in order to break the Replica Symmetry: the system thus obtained was characterized by a free energy in which, instead of the usual Hamiltonian, the energy had the form of a “local free entropy”. The system was then replicated y times and $\tilde{W}$ was traced out, as in the RE, obtaining a new extended system of y real replicas with pairwise interaction. Eventually, the limit of vanishing interaction $\gamma\to0^+$ was considered, recovering an equilibrium description identical to the 1RSB Ansatz (in the original system), where y was mapped onto the Parisi parameter m.

If we make a comparison with the RE, the main difference is given by the fact that, in our case, the interaction is not removed and we thus have an explicit dependence on a distance parameter $D\left(\gamma\right)$: this allows us to explore more thoroughly the phase space of the studied problem and discover subdominant structures which are invisible in the equilibrium picture [6]. In our framework, we also have no reason to restrict the analysis to the "physical" range $y \in [0,1]$ (where the standard 1RSB parameters acquire a straightforward interpretation), being interested in the limit of large y, instead. However, if we complete the ergodicity-breaking procedure, taking the same limits $\gamma \to 0^+$ and $\beta = 0$, indeed our large deviations equations (in section 4.4) reduce to the standard 1RSB case (cf. with section 2.3).

6.3 Application to various algorithms

In the following, we show how the RE algorithmic formula can be applied straightforwardly in the context of learning in discrete neural networks, greatly enhancing the performance of standard optimization strategies, namely SA, SGD and BP [4]. Instead of restricting the numerical analysis to the Perceptron, we will consider the more general case of two-layer neural networks, or fully-connected committee machines. Notice that, as pointed out before, the synaptic weights in the second layer can all be set to 1, without loss of generality. Thus, the ensemble of y replicas can be parametrized by a multi-dimensional binary array $W_i^{ka} \in \{-1,1\}$, where $k \in \{1,\dots,K\}$ indexes the unit, $i \in \left\{1,\dots,\frac{N}{K}\right\}$ is the synaptic index and $a \in \{1,\dots,y\}$ is the replica index. The output of each unit is obtained by calculating $\tau(\xi;W) = \text{sign}\left(\sum_{i=1}^{N/K} W_i \xi_i\right)$, then the output of the network is given by the majority vote $\zeta\left(\left\{\tau^k\right\}_k; \left\{W^k\right\}_k\right) = \text{sign}\left(\sum_{k=1}^{K} \tau\left(\tau^k; W^k\right)\right)$, where τ^k represents the input to the k-th hidden unit.

6.3.1 Replicated Simulated Annealing

As the simplest example of an optimization strategy based on the RE (equation 6.5), we can start by adopting a Monte Carlo method [25] for sampling the new measure. In order to find a solution, the temperature $1/\beta$ will be slowly lowered, as in standard Simulated Annealing, until either a zero of the energy is reached or a stopping criterion is met. We can obtain a fair comparison with a standard SA on the energy by simply considering the case in which the interaction between the replicas is absent (i.e. $\gamma = 0$, which is equivalent to running y

parallel independent optimization processes). In the coupled version, together with the annealing procedure in β, we also implement a scoping procedure (see section **5.2**), gradually increasing the interaction γ (and reducing the average distance D between the replicas). This intuitively corresponds to exploring the energy landscape on progressively finer scales.

Implementation

The aim of the MC algorithm, in the binary case, is to sample from the probability distribution:

$$\begin{aligned} P\left(\{W^a\}\right) &\propto \sum_W \exp\left(-\beta \sum_{a=1}^{y} E\left(W^a\right) + \gamma \sum_{a=1}^{y} \sum_{j=1}^{N} W_j^a W_j\right) \\ &\propto \exp\left(-\beta \sum_{a=1}^{y} E\left(W^a\right) + \sum_j \log\left(2\cosh\left(\gamma \sum_{a=1}^{y} W_j^a\right)\right)\right) \end{aligned} \tag{6.9}$$

At fixed values of β and γ, the sampling can be performed straightforwardly by using the Metropolis rule: the proposed move is to flip (i.e. change sign to) a random synaptic weight from a random replica. The energy variation associated to a candidate move, though, includes the interaction term, introducing a bias that favors movements in the direction of the center of mass of the system of y replicas:

$$k_j = \frac{1}{2}\left(\log\left(\frac{\cosh\left(\gamma + \gamma \sum_{b\neq a} W_j^b\right)}{\cosh\left(-\gamma + \gamma \sum_{b\neq a} W_j^b\right)}\right)\right) \tag{6.10}$$

This bias, which can take only a finite, $\mathcal{O}(1)$, set of possible values, can be entirely accounted for by adding a prior on the choice of the moves, while still maintaining the detailed balance condition (of course, this reduces to the standard Metropolis rule for $\gamma = 0$). In general, given a transition probability $P\left(W \to W'\right)$, from a state W to W', the detailed balance equation reads:

$$P\left(W\right) P\left(W \to W'\right) = P\left(W'\right) P\left(W' \to W\right) \tag{6.11}$$

The probability is usually split in two parts: $P\left(W \to W'\right) = C\left(W \to W'\right) A\left(W \to W'\right)$, where C is the probability of proposing the given move, and A is the probability of accepting it. In the standard Metropolis the choice of the proposed index j

is uniform in the interval$\{1,\dots,N\}$, and the move is accepted with probability $\min\left(1, e^{-\beta\Delta E_{W\to W'}-2k_jW_j}\right)$.

Instead, the effect of the field can be almost completely incorporated in the proposal of the move. One can first organize the possible choices of indices in y classes $\{K_c\}_{c=-y}^{y}$, based on the possible values of $W_jk_j = c$, and then assume that the probability of choosing one of the two classes $K_c \bigcup K_{-c}$ is a simple function of their cardinality $q_c = n_c + n_{-c} = |K_c| + |K_{-c}|$, while the indices in each set can be assigned with a uniform probability. With these assumptions, one would like to find a form for the conditional probability $\hat{P}_c(n_c, q_c) = \frac{N}{q_c}P(c; n_c, q_c)$, of choosing class K_c (between K_c and K_{-c}), such that the following condition holds:

$$e^{-2k_jW_j}\frac{C(W'\to W)}{C(W\to W')} = 1 \tag{6.12}$$

Unfortunately this condition cannot be always satisfied, and one is left with a residual rejection rate $a_c(n_c, q_c)$ in the special case $n_c = q_c$: this scenario is connected to the "condensation" phenomenon which can be observed in the limit of very large γ (and very large c), where an aligned configuration of replicas is found to be highly favored.

The conditional probability can thus be written in terms of the hypergeometric function:

$$\hat{P}_c(n_c, q_c) = \phi\left(n_c, q_c, e^{-2c}\right)(1-\delta_{n_c,q_c}) + \delta_{n_c,q_c} \tag{6.13}$$

$$\phi(n, q, \lambda) = \lambda\frac{n}{q-n+1}\,{}_2F_1(1, 1-n; q-n+2; \lambda) \tag{6.14}$$

$$A(W\to W') = \min\left(1, e^{-\beta\Delta E_{W\to W'}}\right)a_c(n_c, q_c) \tag{6.15}$$

where the residual rejection rate is given by:

$$a_c(n_c, q_c) = \begin{cases} 1-\delta_{n_c,q_c}\left(1-e^{-2c}\right)^{q_c} & \text{if } c > 0 \\ 1 & \text{if } c \le 0 \end{cases} \tag{6.16}$$

The biased sampling procedure is thus the following: choose a class pair $K_c \bigcup K_{-c}$ at random with probability $\frac{q_c}{N}$, then choose either K_c or K_{-c} according to $\hat{P}_c$, finally pick another index uniformly at random within the class. Of

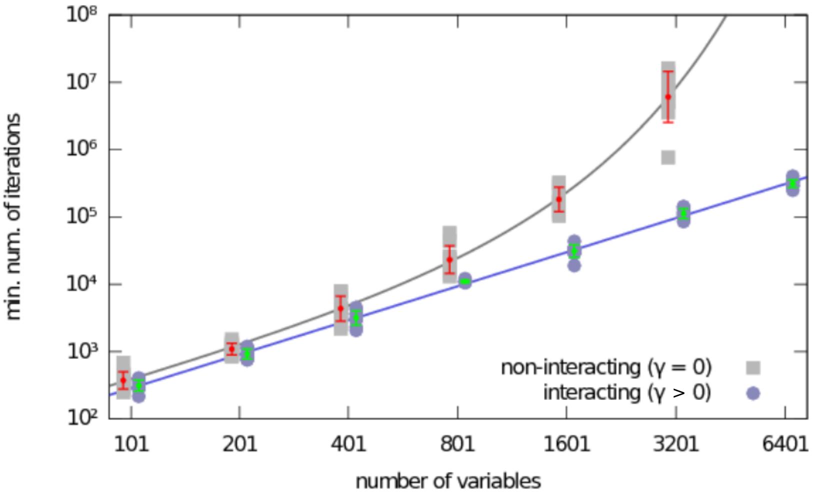


Fig. 6.2 **Replicated Simulated Annealing** on the Perceptron, comparison between the interacting version (i.e. which seeks regions of high solution density) and the non-interacting version (i.e. standard SA), at $\alpha = 0.3$ using $y = 3$ replicas. With optimized annealing/scoping parameters, the minimum number of iterations required to find a solution scales exponentially with N for the standard case, and polynomially for the interacting case. 10 samples were tested for each value of N (the same samples in both cases). The bars represent averages and standard deviations (taken in logarithmic scale) while the lines represent fits. The interacting case was fitted by a function aN^b with $a \simeq 0.13$, $b \simeq 1.7$, while the non-interacting case was fitted by a function $aN^b e^{cN^d}$ with $a \simeq 0.2$, $b \simeq 1.5$, $c \simeq 6.6 \cdot 10^{-4}$, $d \simeq 1.1$. Data is not available for the non-interacting case at $N = 6401$ since we couldn't solve any of the problems in a reasonable time (the extrapolated value according to the fit is $\sim 3 \cdot 10^9$). The two data sets are slightly shifted relative to each other for presentation purposes.

course, the advantage of this method consists in reducing the rejection rate, but at the same time the move proposal becomes more computationally expensive, so this approach is not well suited for systems in which computing the energy cost is very easy. This efficient sampling procedure was utterly refined in [77].

Numerical results

In figures 6.2 and 6.3, we show the numerical results of RSA applied to the learning problems in the binary Perceptron and in the committee machine. A scaling analysis demonstrates that the interaction is crucial for finding a solution in polynomial time. We also note that the gap in performance between the interacting and non-interacting cases widens with increasing storage loads.

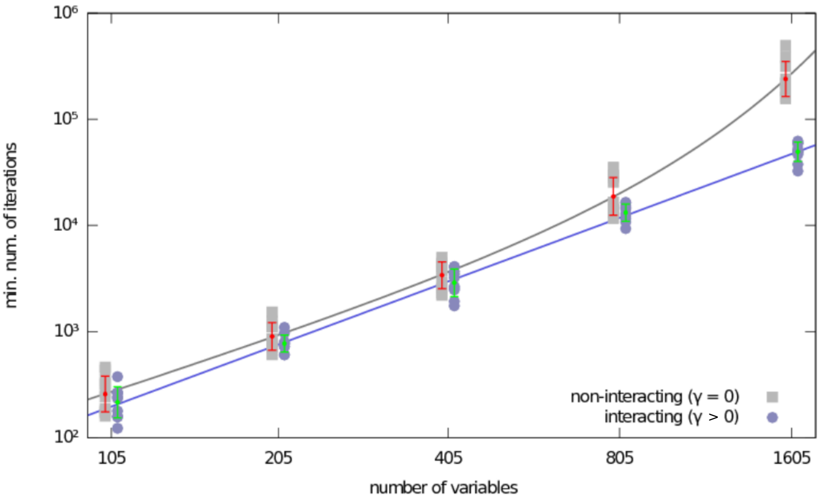


Fig. 6.3 **Replicated Simulated Annealing** on the fully-connected committee machine, with $K = 5$ hidden units, comparison between the interacting version (i.e. which seeks regions of high solution density) and the non-interacting version (i.e. standard SA), at $\alpha = 0.2$ using $y = 3$ replicas. This is the analogous of figure 2 of the main text for a committee machine, showing similar results. 10 samples were tested for each value of N (the same samples were used for the two curves). The bars represent averages and standard deviations (taken in logarithmic scale) while the lines represent fits. The interacting case was fitted by a function aN^b with $a \simeq 0.02$, $b \simeq 2.0$, while the non-interacting case was fitted by a function $aN^b e^{cN^d}$ with $a \simeq 0.08$, $b \simeq 1.7$, $c \simeq 4.2 \cdot 10^{-5}$, $d \simeq 1.5$. The two data sets are slightly shifted relative to each other for presentation purposes.

Finally, we remark that, in the replicated MC we substitute the two-level strategy of EdMC, where BP was employed to get an instantaneous estimate of the local free entropy, with a single level strategy defined in a replicated system. In the new formulation, though, the replicas need to equilibrate at each value of γ and β, and with a finite sampling there is no guarantee that the landscapes explored by the RSA and EdMC will be exactly the same: it is possible that, when BP can actually provide the needed information, the latter method is affected by even fewer local minima.

6.3.2 Replicated Gradient Descent

When the size of the network is very large, Monte Carlo methods can become very computationally expensive. One simple alternative general method for finding minima of the energy is using Gradient Descent (GD) or one of its many variants. All these algorithms are generically called back-propagation algorithms in the neural networks (NN) context [61]. In particular, Stochastic GD (SGD) is the basis of most of the recently developed "deep learning" techniques employed in Machine Learning. In the following, we demonstrate that performing the gradient descent over the RE energy defined in equation (6.6), leads to a noticeable improvement in the performance of the algorithm; moreover, the solutions found by the algorithm are indeed part of a dense regions, as expected.

Implementation

Gradient Descent is defined only for differentiable systems, and thus it needs some adaptations in order to be applied to the case of systems with discrete variables.

One possible work-around is a generalization to a mini-batch learning scenario of the "Clipped Perceptron" (CP) algorithm [78]: we can associate an auxiliary continuous variable $\mathcal{W}$ to each binary synaptic variable W, binding them through the relationship $W = \mathrm{sign}\left(\mathcal{W}\right)$. The gradients will now be evaluated in correspondence of the real synapses W, but stored in the auxiliary variables:

$$\left(\mathcal{W}_i^k\right)^{t+1} = \left(\mathcal{W}_i^k\right)^t - \eta \frac{1}{|m\left(t\right)|} \sum_{\mu \in m(t)} \frac{\partial}{\partial W_i^k} E^\mu \left(W^t\right) \tag{6.17}$$

$$\left(W_i^k\right)^{t+1} = \operatorname{sign}\left(\left(\mathcal{W}_i^k\right)^{t+1}\right) \tag{6.18}$$

where η is the learning rate and $m\left(t\right)$ is a set of pattern indices (the so-called minibatch). The CP algorithm is recovered in the case of a single layer network without replication ($K = 1$, $y = 1$), of a fixed learning rate, and in the fully-online regime ($|m\left(t\right)| = 1$). In that case, since $E^\mu\left(W\right) = R\left(-\sum_i W_i \xi_i^\mu\right)$, with $R\left(x\right) = \frac{1}{2}\left(x+1\right)\Theta\left(x\right)$, the gradient becomes $\partial_{W_i} E^\mu\left(W\right) = -\frac{1}{2}\xi_i^\mu \Theta\left(-\sum_i W_i \xi_i^\mu\right)$. The relation (6.18) is scale-invariant, so we can just set $\eta = 4$ and obtain

$$\mathcal{W}_i^{t+1} = \mathcal{W}_i^t - 2\xi_i^\mu \Theta\left(-\sum_i W_i^t \xi_i^\mu\right) \tag{6.19}$$

where the auxiliary quantities $\mathcal{W}$ can be restricted to discrete values as well, if they are initialized as integers. We note that the CP rule by itself does not achieve an extensive capacity in the large N limit; it is however possible to make it efficient, as in the CP+R heuristic algorithm (see section 3.2) or by adding the interaction term as in the RE.

In the two-layer case ($K > 1$) the energy associated to a wrong classification can be defined as the minimum number of spin flips needed to correct the output. The computation of the gradient becomes more involved, but of course gives a non-zero contribution only in case of error, and only for those units k which contribute to the energy computation. Also in this case, since by setting $\eta = 4$ the gradient is restricted to 3 possible integer values, we could use discretized variables for the $\mathcal{W}$. It is interesting to point out that a slight variation of this update rule in which only the most easily-fixable unit is affected gives the extended CP+R rule, decribed in section 3.2, giving good results on a real-world learning task when the uniform reinforcement term was added. Note that, the difference between the two rules becomes irrelevant in the later stages of learning, when the overall energy is low.

Once we have the gradient of $E\left(W\right)$ separately for each system, we can add the interaction of the RE (with the traced-out center), and obtain the full

SGD update:

$$\begin{aligned}\left(\mathcal{W}_i^a\right)^{t+1} = \left(\mathcal{W}_i^a\right)^t - \eta \frac{1}{\left|m\left(t\right)\right|} \sum_{\mu \in m(t)} \frac{\partial E^\mu}{\partial W_i}\left(W\right)\Bigg|_{W=\left(W^a\right)^t} \\ + \eta' \left(\tanh\left(\gamma \sum_{b=1}^{y}\left(W_i^b\right)^t\right) - \left(W_i^a\right)^t\right)\end{aligned} \tag{6.20}$$

where we used $\eta' = \frac{\gamma}{\beta\eta}$ as a control parameter, such that it remains finite in the limit $\beta, \gamma \to \infty$; in this limit the tanh reduces to a sign.

The update equation (6.20) can be implemented in the following way: at each time step, we pick uniformly at random a replica a and compute the gradient with respect to a mini-batch of $m\left(t\right)$ patterns, we partially update $\mathcal{W}^a$ and W^a, we compute the gradient with respect to the interaction term with the stored value of $\sum_{a=1}^{y} W_i^a$, update it, and then complete the updates of $\mathcal{W}^a$ and W^a. This scheme can be easily parallelized, since it alternates the standard learning periods in which each replica acts independently with brief interaction periods, similarly to what was done in [79]. In our tests, we kept fixed the learning rates η and η' during the training process, and we implemented the usual scoping procedure.

Numerical results

In figure 6.4 we can see the results obtained in the case of the fully-connected committee machine: the introduction of the interaction term greatly improves the capacity of the network (from 0.3 to almost 0.6), and generally reduced the number if required presentations of the dataset (epochs); moreover, when the algorithm fails to solve the instance the reached configurations have a lower error rate than the non-interacting version. We also observed the same qualitative results in the Perceptron, where a capacity exceeding 0.7 can be reached, suggesting the fact that Replicated SGD is able to achieve near-optimal learning performance.

Relationship with EASGD

It is interesting to note that a very similar learning strategy—a replicated system in which each replica is attracted towards a reference configuration,

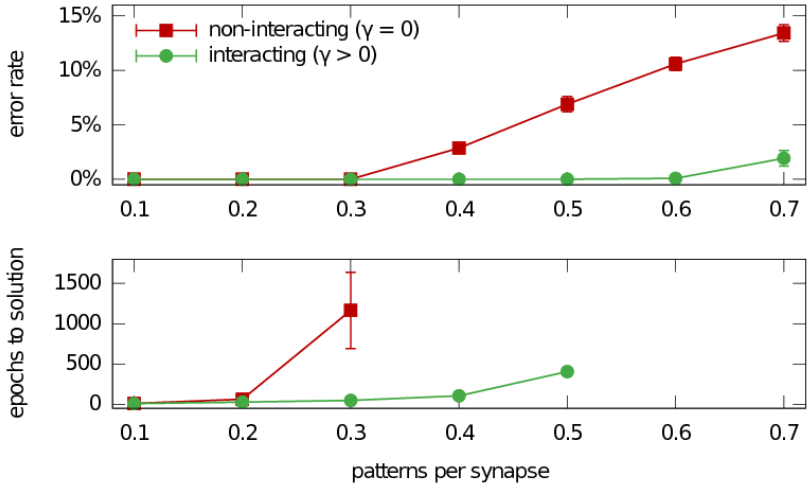


Fig. 6.4 **Replicated Stochastic Gradient descent** on a fully-connected committee machine with $N = 1605$ synapses and $K = 5$ units in the second layer, comparison between the non-interacting (i.e. standard SGD) and interacting versions, using $y = 7$ replicas and a minibatch size of 80 patterns. Each point shows averages and standard deviations on 10 samples with optimal choice of the parameters, as a function of the training set size. Top: minimum training error rate achieved after 10^4 epochs. Bottom: number of epochs required to find a solution. Only the cases with 100% success rate are shown (note that the interacting case at $\alpha = 0.6$ has 50% success rate but an error rate of just 0.07%).

called Elastic Averaged SGD (EASGD)—was proposed in [79] (see also [80]).The context was that of deep convolutional networks with continuous variables, and EASGD was heuristically introduced to exploit parallel computing environments under communication constraints. In this work, the strategy of replicating the system and introducing the elastic interaction was concurrent with the employment of the usual deep learning techniques (e.g. momentum), so it is difficult to fully disentangle the effect of the various heuristics. However, their results clearly demonstrate a benefit from introducing the replicas in terms of training error, test error and convergence time.

It might be plausible that the general underlying reason for the effectiveness of the method is similar, related to the possibility of accessing robust low-energy states in the space of configurations, despite a conclusive assessment is difficult due to the great jump in complexity in the choice of the network architecture.

6.3.3 Replicated Belief Propagation

As we have seen in section 3.2, BP can be turned into a solver with the addition of a “reinforcement” term [26]: for each variable, a time-dependent local field, proportional to its most recent marginal probability estimation, is introduced and is gradually increased to induce a polarization of the system towards a single configuration, in a soft decimation fashion. Despite the clear effectiveness of reinforced BP, as a solver in a variety of problems, even when BP would suffer from convergence problems, a real understanding of the reasons for its performance is not at hand. Intuitively, R-BP progressively focuses on smaller and smaller regions in the phase space of the problem, by looking in the “most promising” direction, as given by the current estimation of the probability distribution. This process has thus some qualitative similarities with the scoping procedure employed to search for dense regions of solutions (as in the previous sections).

This analogy can be explored more thoroughly by writing the BP equations for the replicated system described by equation (6.4). There are two possible approaches [6], which lead to the same results:

1. we can use the local entropy as the energy function, writing a second-level BP to estimate the local entropy itself. The obtained equations are very

similar to the so called 1-step replica-symmetry-breaking (1RSB) cavity equations (see [24] for a general introduction).

2. The second approach is to explicitly replicate the system, but, in order to take into account the correlations between the replicated copies of the variables, one has to consider N vector variables $\left\{W_j^a\right\}_{a=1}^y$ of length y. We can simplify the equations by assuming Replica Symmetry, i.e. that all marginals are invariant under permutation of the replica indices: $P_j\left(\left\{W_j^a\right\}_{a=1}^y\right) = P_j\left(\sum_{a=1}^y W_j^a\right)$. The resulting message passing algorithm reproduces quite accurately the analytical results at the RS level.

As explained in section 4.3, this type of RS Ansatz becomes wrong at high values of α, γ and y, due to the onset of correlations between the variables (i.e. to a replica-symmetry-breaking effect). From the geometrical point of view, in the RS approximation the solution assumes that there is a single dense region, while the occurrence of RSB effects might imply that there are several maximally dense regions in the RE. As a consequence these two algorithms are not very good candidates for the definition of an effective solver. A more correct description would in fact require a third level of BP equations, or, equivalently, an assumption of symmetry-breaking in the structure of the marginals $P_j\left(\left\{W_j^a\right\}_{a=1}^y\right)$.

Implementation

Fortunately, there is a simplified way of turning BP in the replicated system into an efficient solver, still well described by the theoretical results but also very similar to the reinforced BP algorithm. Instead of considering the joint distribution over each replicated vector variable (which is required for a more correct treatment of the correlations), at a certain site j, one can naively replicate the original factor graph y times; then, every replicated site j will be connected to an extra variable $W_j^\star$, thus introducing the y interactions between all the W_j^a and $W_j^\star$.

If we now make a symmetry assumption between the replicas, implying that each replica of the system would behave in exactly the same way and that same messages would be exchanged along the edges of the graph, regardless of the replica index, we can avoid the redundancy and work with a single system.

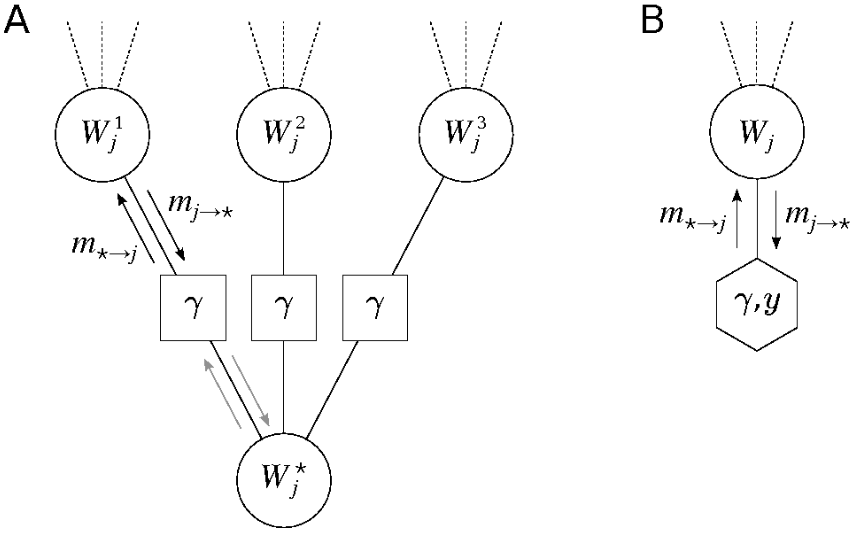


Fig. 6.5 **A**. **Portion of a BP factor graph** for a replicated variable W_j with $y = 3$ replicas and a reference configuration $W_j^\star$. The dashed lines represent edges with the rest of the factor graph. The squares represent the interactions $\gamma W_j^\star W_j^a$. All BP messages (arrows) are assumed to be the same in corresponding edges. **B**. **Transformed graph** which represents the same graph as in A but exploits the symmetry to reduce the number of nodes, keeping only one representative per replica. The hexagon represents a pseudo-self-interaction, i.e. it expresses the fact that $m_{\star\to j}$ depends on $m_{j\to\star}$ and is parametrized by γ and y.

This representation is effectively identical to the original one, except for the fact that the elastic interaction between the replicas has now been mapped in a self-interaction at each variable site W_j, exchanging messages with $y-1$ identical copies of itself through an auxiliary variable (that can be traced out). The factor node structure is shown graphically in figure 6.5. Thus, at each iteration step t, each variable receives an extra message of the form:

$$m_{\star\to j}^{t+1} = \tanh\left((y-1)\tanh^{-1}\left(m_{j\to\star}^{t}\tanh\gamma\right)\right)\tanh\gamma \tag{6.21}$$

where $m_{j\to\star}^{t}$ is the cavity magnetization resulting from the standard factor graph (without the self interaction) at time t. After this approximated transformation, the inverse temperature y reappears as a continuous parameter, and is no longer constrained to integer values. This version of the algorithm will be referred to as “focusing Belief Propagation” (fBP).

Numerical results

The proposed algorithm, fBP, has a straightforward application as a solver: the best results are obtained when one employs a scoping procedure on γ, and at the same an annealing in the inverse temperature y, until a solution is found. However, it is also interesting to compare the numerical results, at fixed values of y and γ, with the analytical predictions obtained in the binary Perceptron case.

The local entropy density can be computed from the entropy of the whole replicated system (from the BP messages at their fixed point), by subtracting the entropy of the reference variables. The result is then normalized by the number of variables N and of replicas y. Finally, we need to take a Legendre transform by subtracting the interaction term γS, where the overlap S between each replica’s weights is computed as:

$$S = \frac{1}{N}\sum_j \frac{m_{j\to\star}m_{\star\to j} + \tanh(\gamma)}{1 + m_{j\to\star}m_{\star\to j}\tanh(\gamma)} \tag{6.22}$$

In particular, the resulting estimate of the local entropy at high y is in very good agreement with the 1RSB predictions up to at least $\alpha = 0.6$, as can be seen figure 6.7, where we set $y = 21$ and demonstrate that the fBP curve deviates

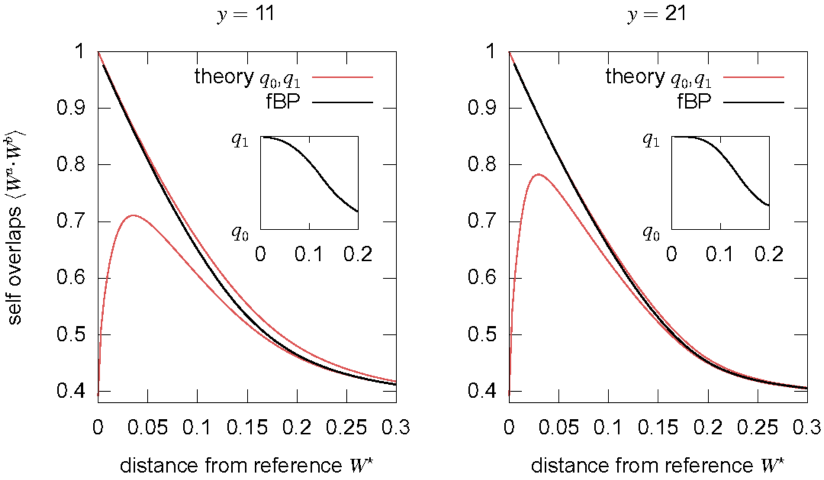


Fig. 6.6 **Focusing BP** (fBP) spontaneously breaks replica symmetry: the overlap order parameter q (black thick curves) gradually transitions from the inter-cluster overlap q_0 and the intra-cluster overlap q_1 of the replica theory (red thin curves, $q_0 < q_1$) as the distance to the reference $W^\star$ goes to 0 (i.e. as $\gamma \to \infty$). The insets provide an alternative visualization of this phenomenon, plotting $(q - q_0) / (q_1 - q_0)$ against the distance. These results were obtained on a Perceptron with $N = 1001$ at $\alpha = 0.6$, averaging over 50 samples. The two panels shows that the transition occurs at larger distances (i.e. at smaller γ) at larger y.

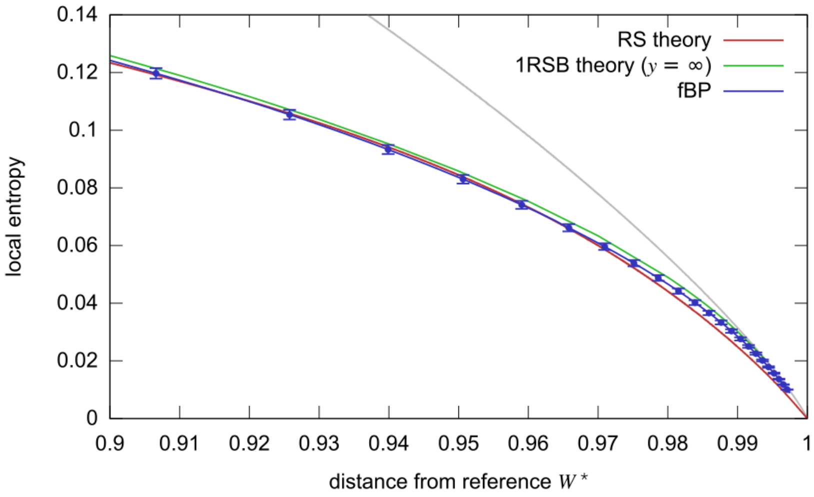


Fig. 6.7 **Comparison of local entropy curves** between the fBP results and the analytical predictions, for the case of the Perceptron with $\alpha = 0.6$. The algorithmic results (blue curve) were obtained with $N = 1001$ at $y = 21$, averaging over 50 samples. Error bars indicate the estimated standard deviation of the mean. The RS results (red curve) were also obtained with $y = 21$. The 1RSB results, however, are for the $y = \infty$ case, and it is therefore to be expected that the corresponding curve is slightly higher.

from the RS prediction and is very close to the 1RSB case. This suggests that the algorithm has spontaneously chosen one of the possible states of high local entropy in the RE, achieving an effect akin to the spontaneous symmetry breaking of the 1RSB description. Within the state, replica symmetry holds, so that the algorithm is able to eventually find a solution to the problem.

This replica-symmetry breaking behavior can be better highlighted, as in figure **6.6,** by studying the average overlap between replicas (defined as $q = \frac{1}{N} \sum_j W_j^a W_j^b$): it's value is close to q_0 (the average overlap between replicas belonging to different states) for low γ, but it becomes close to q_1 (the average overlap between replicas in the same state) at high γ. This analysis confirms a scenario in which the fBP algorithm spontaneously chooses a high density state, breaking the symmetry in a way which seems to approximate well the 1RSB description.

Furthermore, fBP can also be used to obtain an estimate of the value of α at which the accessible dense states cease to exist, even in cases, like multi-layer networks, where analytical calculations are unfeasable. Figure 6.8 shows the result of experiments performed on a committee machine. The implementation closely follows [26] with the addition of the self-interaction equation (6.21), except that great care is required to get a correct numerical estimation of the local entropy at large γ, due to numerical issues. The figure shows that dense states are found by fBP nearly up to $\alpha = 0.6$, in agreement with the results obtained with the replicated GD, and $\alpha = 0.6$ is also the value of the algorithmic threshold, after which fBP is no longer able to find solutions.

In figure 6.8, we can also see how fBP is able to break the permutation symmetry affecting the committee machine (cf. section 3.3) only once a high enough coupling γ is employed.

Focusing BP vs Reinforced BP

If we make a comparison between equation 3.14 (in section 3.2) and equation (6.21), we can see that the reinforced introduces a self interaction term of the form:

$$m_{\star\to j}^{t+1} = \tanh\left(\rho \tanh^{-1}\left(m_j^t\right)\right) \tag{6.23}$$

In order to find a solution with R-BP, usually the reinforcement parameter ρ is changed dynamically, increasing from an initial value of 0 up to 1 during the BP message-passing iterations. The only possible fixed points are thus corresponding to completely polarized configuration, i.e. one where $m_j \in \{-1, +1\}$ for all j. Similarly, in the fBP scheme the two external parameters γ and y need to diverge in order to ensure that the marginals m_j become completely polarized as well.

The main difference between these two schemes is the fact that the self-interaction $m_{\star\to j}$ is a function of a cavity marginal $m_{j\to\star}$ in the case of fBP, and of a non-cavity marginal m_j in case of R-BP. However, a relationship between the two formulations can be found by considering the BP fixed points reached at constant values for the parameters γ, y and ρ. The self-consistency condition between the quantities $m_{\star\to j}$, $m_{j\to\star}$ and m_j read:

$$m_j = \tanh\left(\tanh^{-1}\left(m_{\star\to j}\right) + \tanh^{-1}\left(m_{j\to\star}\right)\right) \tag{6.24}$$

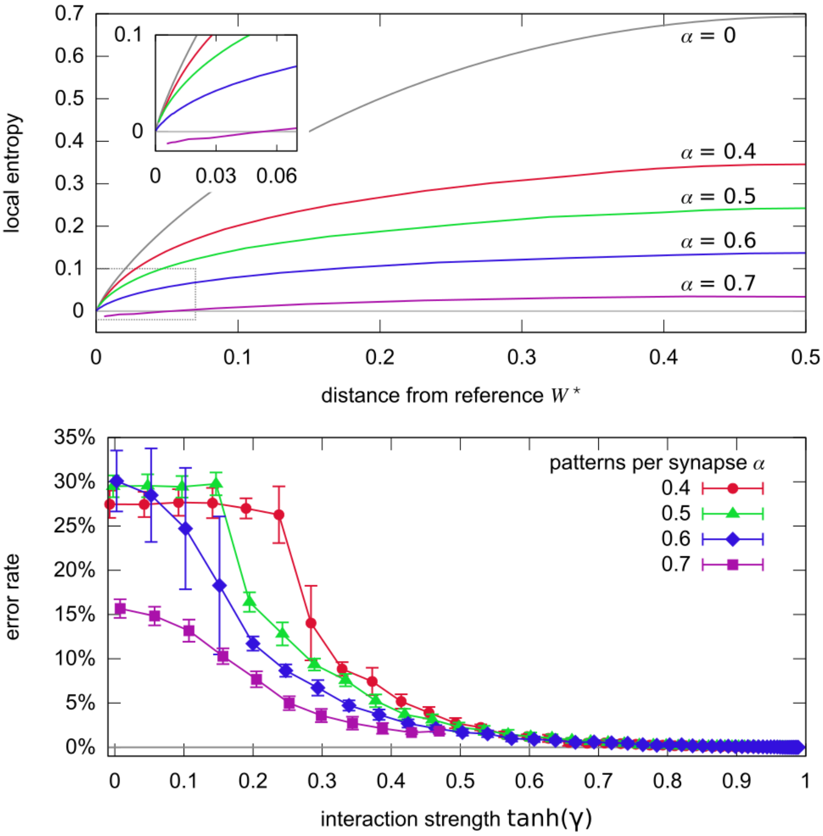


Fig. 6.8 **Results of fBP** on a committee machine with $N = 1605$, $K = 5$, $y = 7$, increasing γ from 0 to 2.5, averages on 10 samples. Top: local entropy versus distance to the reference $W^\star$ for various α (error bars not shown for clarity). The topmost gray curve ($\alpha = 0$) is an upper bound, representing the case where all configurations within some distance are solutions. Inset: enlargement of the region near the origin indicated by the rectangle in the main plot. This shows that dense states exist up to almost $\alpha = 0.6$: at this value of α, dense states are only found for a subset of the samples (in which case a solution is also found). Negative local entropies (curve at $\alpha = 0.7$) are unphysical, and fBP fails shortly after finding such values. Bottom: error rates as a function of $\tanh(\gamma)$. For $\alpha \leq 0.6$, all curves eventually get to 0. However, only 7 out of 10 samples reached a sufficiently high γ at $\alpha = 0.6$, while in 3 cases the fBP equations failed. The curve for $\alpha = 0.7$ is interrupted because fBP failed for all samples, in each case shortly after reaching a negative local entropy. The plateaus at $\alpha = 0.4$ and $\alpha = 0.5$ are regions where the solution to the equations are symmetric with respect to the permutation of the hidden units: fBP spontaneously breaks that symmetry as well.

Therefore in this case of R-BP we get:

$$m_j = \tanh\left(\frac{1}{1-\rho}\tanh^{-1}\left(m_{j\to\star}\right)\right) \tag{6.25}$$

while the analogous expression in the fBP case is:

$$m_j = \tanh\left(\tanh^{-1}\left(m_{j\to\star}\right) + \tanh^{-1}\left(\tanh\left(\left(y-1\right)\tanh^{-1}\left(m_{j\to\star}\tanh\gamma\right)\right)\tanh\gamma\right)\right) \tag{6.26}$$

Even though the second expression appears to be much more complicated, if we let $\gamma \to \infty$ and $y = \frac{1}{1-\rho}$,it simplifies exactly to the former, obtaining an exact mapping between fBP and R-BP. However, since we are looking for a physical motivation for the efficacy of R-BP, this limiting case doesn't allow for a straightforward connection with the reweighted entropic measure, since the requirement $\gamma \to \infty$ seem to rule out the "non-local" effect of the interaction.

On the other hand, we can devise a possible annealing protocol for fBP, in which both γ and y start from low values and are progressively increased, related by the parameter ρ:

$$\gamma = \tanh^{-1}\left(\rho^x\right) \tag{6.27}$$

$$y = 1 + \frac{\rho^{1-2x}}{(1-\rho)} \tag{6.28}$$

With this choice of y it is possible to match the derivative between the curves of eqs. (6.25) and (6.26) in the point $m_{j\to\star} = 0$. Note also that both $\gamma \to \infty$ and $y \to \infty$ in the limit $\rho \to 1$, thus ensuring that, in that limit, the only fixed points of the iterative message passing procedure are completely polarized, consistently with the notion that we are looking regions of maximal density ($y \to \infty$) at small distances ($\gamma \to \infty$). If we now set $x = 0$, we obtain again the exact map onto the standard reinforcement relations. However, even at different values of x one can observe the same qualitative and quantitative behaviors, as can be seen in figure (6.9) (representing the case $x = 0.5$). Moreover, the performances of the resulting algorithm are hardly distinguishable from R-BP. In this sense, we can say that the effectiveness of reinforced BP is due to the fact that it targets the same accessible dense states described in the RE.

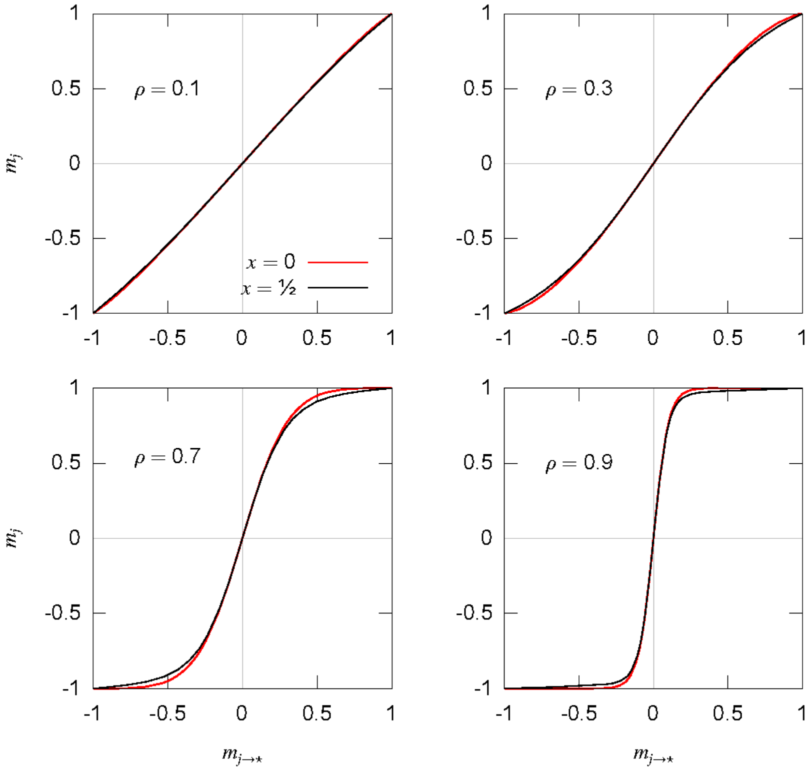


Fig. 6.9 **Plots of eq.** (6.26), comparison of protocols defined by eqs. (6.28) and (6.27) with two different values of the parameter x. The $x = 0$ case (thick red lines) corresponds to standard reinforcement. The curves are in fact very similar across the whole range of $\rho \in [0, 1]$ and $x \in [0, 1]$, and consequently display similar performance properties in practice.

Chapter 7

Stochastic Synapses

In the previous chapter, we have seen the proposal of a novel optimization strategy able to reroute simple learning algorithms towards dense regions of solutions, avoiding the many poor local minima that characterize the loss landscape in feed-forward ANNs. We have also proposed some qualitative arguments supporting the correlation of a high local entropy density with a very desirable property in the generalization scenario (see section 5.6), namely the enhanced robustness to noise of these special solutions. This robustness translates into excellent generalization performance for the optimal configurations of the Robust Ensemble, indicating that they are able to act as local Bayesian representatives of the entire neighborhoods of low energy configurations surrounding them [4].

We can thus try to adopt a reversed approach (with respect to [4]), where the system is purposely required to learn in a stochastic (i.e., noisy) setting and see if the high local entropy can result as an emerging property. In nature, synaptic weights are known to be plastic, low precision and unreliable: it is thus important to understand if this synaptic stochasticity can help or hinder the learning process [81].

In order to learn with stochastic synapses, we need to move to a Bayesian learning framework, where instead of looking for a single assignment for the synaptic weighs one is interested in characterizing a probability distribution over the possible assignments, the so-called posterior distribution [5]. The posterior is to be inferred in the assumption of maximizing the likelihood of some input-output associations, prescribed by a given dataset $\mathcal{D}$. One can devise a learning procedure in which the posterior distribution obtained at the previous step is used as a prior (similar to the Empirical Bayes approach), from which the stochastic synapses are sampled independently for each pattern. Thus, one obtains a stochastic machine that learns a "noise-robust" parametrization of the probability distribution, such that any sampled network configuration is likely to achieve a correct classification of the training patterns.

In this case, the learning process affects the parameters of the prior distribution, instead of the synapses themselves, and this allows the employment of gradient descent algorithms that would otherwise be inappropriate in a binary setting. At late stages of the learning procedure, the prior is expected to peak around a single binary configuration, which can then be proposed as the solution found by the algorithm: even though the final prior apparently neglects the neighborhood of this configuration, being very focused on a small region, the fact that it was obtained through a dynamical procedure, where the scope of the posterior distribution is slowly narrowed, implies that the local entropy is playing a role in its choice.

In the following, we will first introduce a more rigorous description of this framework, and then we will describe a theoretical analysis in the Binary Perceptron model, once again chosen as the prototypical model of an ANN, linking the Bayesian approach to our Large Deviation analysis (see chapter 4).

7.1 Maximum likelihood and the Bayesian approach

Let us consider a typical Deep Learning classification task: we have a training set $\mathcal{D} = \{(x^\mu, y^\mu)\}_{\mu=1}^{M}$ of M input-output associations, where the correct labels are represented by indicator vectors in a K-dimensional space. For any input $x \in \mathcal{X}$ and for any assignment of the synaptic weights W, an ANN defines a probability density function $P(y \,|\, x, W)$ over the K possible categories: while the propagation of the activity through the network is completely deterministic, involving scalar products and the application of a non-linear activation function, the employment of a softmax function in the last layer can turn the outcome into a probabilistic prediction. The goal of the learning procedure is to adjust the parameters W according to the supervised error-signal resulting from wrong classifications. From a mathematical point of view, the learning problem can be framed as a *log-likelihood* $\tilde{\mathcal{L}}(W)$ maximization over the synaptic weights W:

$$\max_W \tilde{\mathcal{L}}(W) := \sum_{(x,y)\in\mathcal{D}} \log P(y \,|\, x, W) \tag{7.1}$$

This optimization problem is usually approximately solved by taking $-\tilde{\mathcal{L}}(W)$ as a loss-function and by applying some heuristically improved version of the Gradient Descent (GD) procedure [62].

The aim of a Bayesian approach, instead, is that of obtaining an approximation of the *posterior* distribution $P(W \,|\, \mathcal{D}) \propto P(\mathcal{D} \,|\, W)P(W)$, with a proper choice of a *prior*, $P(W)$, for the synaptic weight distribution. Ideally, with the knowledge of the posterior distribution one could obtain the best possible prediction in a generalization scenario, given by the weighted average output $\hat{y}(x;\mathcal{D}) = \mathrm{argmax}_y \int dW\, P(y \,|\, x, W)P(W \,|\, \mathcal{D})$. Unfortunately, in real applications, an exact computation of $P(W \,|\, \mathcal{D})$ and of the Bayesian integral is almost always unfeasible. For this reason, there have been various proposals of methods for approximating the posterior distribution, based on variational approximations, strong factorization assumptions, or obtained through Monte Carlo estimations [82–85].

Inspired by the analytical analogy between the computation of the local entropy reweighed measure and the Stochastic Binary Perceptron model (see following paragraphs), it is possible to devise a novel optimization method able to find binary solutions of the problem (7.1), that shares some features with the Bayesian approach. We first introduce a factorized family of probability distributions over the synaptic weights (the analog of the Bayesian prior), $Q_\theta(W)$, parametrized by a set of variables θ. Since the synapses can take one of two possible values, a single parameter θ_i per synapse is sufficient. As stated in the introduction, we formulate the learning problem as follows:

$$\max_\theta \quad \mathcal{L}(\theta) := \sum_{(x,y)\in\mathcal{D}} \log \mathbb{E}_{W\sim Q_\theta}\, P(y \,|\, x, W) \tag{7.2}$$

Here, $\mathcal{L}(\theta)$ is the log-likelihood of a model where the synaptic weights are independently sampled, according to the prior $Q_\theta(W)$, in correspondence of each classification pattern $(x, y) \in \mathcal{D}$. This approach bears great resemblance with some variational methods for approximating the Bayesian posterior distribution (as in [84], for example), except that in our case the usual order between the sum and the expectation in equation (7.2) is inverted. This subtle difference is quite relevant, in that it introduces the notion of stochastic synapses into the model. Within this scheme we can obtain an approximation to the Bayesian integral, $\hat{y}(x) = \text{argmax}_y \int dW\, P(y \,|\, x, W) Q_\theta(W)$, averaging over all the solutions included in the prior, as well as the simpler predictor $\hat{y}(x) = \text{argmax}_y P(y \,|\, x, \hat{W})$, choosing the mode of the prior $\hat{W} = \text{argmax}_W Q_\theta(W)$ as a deterministic assignment for the synaptic weights.

In general, in deep networks, the Bayesian problem of equation (7.2) is more involved than the maximum likelihood (7.1), because of the computational difficulty of dealing with the distributions $P(W \,|\, \mathcal{D})$ instead of single instance parameters W. Also notice that, for any zero-temperature solution W^* of (7.1), provided that it belongs to the same parametric family of $Q_\theta\,(W)$, we have that $\delta(W - W^*)$ is a solution also of the second problem (7.2).

As we will see in the following, there are at least two reasons for choosing the complications involved with $\mathcal{L}(\theta)$, instead of $\tilde{\mathcal{L}}(W)$:

- In the discrete setting, this formulation allows the application of GD algorithms for training the network, since the training takes place at the level of the continuous parameters θ of the distribution.
- The optimization over $\mathcal{L}(\theta)$ exhibits some useful dynamical properties: the GD in the θ-space naturally incorporates the regularization property of the Bayesian approach, but also induces a high robustness to the sought solutions. In fact, if one initializes in a configuration of the parameters with a small L_2 norm, representing a wide probability distribution including a very large ensemble of synaptic configurations, the gradient naturally evolves towards a corner $\delta(W - W^*)$: the norm of θ gradually increases, mimicking the *scoping procedure* (cf. with the previous chapters), as the Bayesian posterior gets more and more focused on high local entropy regions.

7.2 Stochastic Perceptron

We now proceed in the theoretical analysis by specializing to the case of the Perceptron, chosen as a prototypical example of discrete feed-forward NN. We will also consider the case of random unbiased i.i.d. binary input patterns: $x_i \sim \frac{1}{2}\delta\left(x_i - 1\right) + \frac{1}{2}\delta\left(x_i + 1\right)$, as usual. The symmetry of the problem allows us to set all the outputs to 1 without loss of generality: $\forall \mu \, y^{\mu} = 1$.

Consider a stochastic version of the same network, in which the values of the weights W are extracted from some probability distribution $Q\left(W;\theta\right)$ parametrized by the vector θ_i, $i = 1, ..., N$. We will denote with $\left\langle \cdot \right\rangle_{\theta}$ the average over W for a given parametrization θ: of course, for some function g, the average is obtained as $\left\langle g\left(W\right)\right\rangle_{\theta} = \sum_W Q\left(W;\theta\right) g\left(W\right)$. In order to make a clear connection with the analyses presented in the previous chapters, we can choose to extract each weight independently, as:

$$Q\left(W;\theta\right) = \prod_{i=1}^{N} \sigma\left(\gamma W_i \theta_i\right) \tag{7.3}$$

where the stochastic machine parameters are normalized to $\sum_{i=1}^{N} \theta_i^2 = N$, a global parameter γ is introduced as an explicit scale, and σ is the logistic function:

$$\sigma\left(x\right) = \frac{1}{1 + e^{-2x}} = \frac{e^{x}}{e^{x} + e^{-x}}. \tag{7.4}$$

With this definition, we see that the average value of each weight is simply: $\left\langle W_i \right\rangle_{\theta_i,\gamma} = \tanh\left(\gamma\theta_i\right)$. From the statistical physics perspective, the control parameters θ play the role of external fields, and the average represents a magnetization $m_i = \left\langle W_i \right\rangle_{\theta_i,\gamma}$ of the state described by the prior. The connection with the entropic reweighting term of equation 4.3, in section 4.1, is clear: if we suppose to take also the parameters θ to be binary, the resulting prior is equivalent to:

$$\sigma\left(\gamma W_i \theta_i\right) \sim \exp\left(\gamma\left(W_i \theta_i\right)\right) \sim \exp\left(-\frac{\gamma}{2}\left(W_i - \theta_i\right)^2\right). \tag{7.5}$$

therefore, we are simply generalizing the elastic coupling with the reference system to the case where $\tilde{W} = \theta$ can take continuous values.

In the stochastic Perceptron, the probability distribution $P(y\,|\,x,W)$ over the possible output classes, $y\in\{-1,+1\}$, can be simply obtained as:

$$P(y\,|\,x,W)=\Theta\left(y\sum_{i=1}^{N}W_i\,x_i\right). \tag{7.6}$$

Then, as mentioned above, one can measure the average performance of the stochastic network on the training set, for a given value of the parameters θ, by using a log-likelihood function $\mathcal{L}\left(\theta;\{x^\mu,y_d^\mu\}\right)$. There are two slightly different possible definitions for the log-likelihood, corresponding to two scenarios:

1. The *full-batch scenario*, usually considered in variational Bayes methods, in which we extract an assignment of weights W and test it on the whole pattern set. $\mathcal{L}$ is therefore the log-likelihood of extracting a value of the weights that correctly classifies *all* the patterns simultaneously:

$$\max_\theta \mathcal{L}(\theta;\{x^\mu,y_d^\mu\}):=\mathbb{E}_{W\sim Q_\theta}\sum_{(x,y)\in\mathcal{D}}\log\,P(y\,|\,x,W) \tag{7.7}$$

2. A *fully-stochastic scenario*, which we consider here, where the weights W are extracted independently for each pattern. $\mathcal{L}$ is therefore the log-likelihood of achieving a perfect overall classification when extracting a new set of weights for each pattern:

$$\max_\theta\quad\mathcal{L}(\theta;\{x^\mu,y_d^\mu\}):=\sum_{(x,y)\in\mathcal{D}}\log\mathbb{E}_{W\sim Q_\theta}\,P(y\,|\,x,W) \tag{7.8}$$

It could indeed be possible to interpolate between the two scenarios, working with mini-batches (as usually done in Deep Learning), but this is left for future work. After the introduction of the likelihood function, we can then define an optimization task over the parameters θ, for a given training set, that consists in trying to maximize $L\left(h;\{x^\mu,y_d^\mu\}\right)$, and study theoretically the behavior of the associated free entropy.

The average free entropy potential in the full-batch case is simply given by:

$$\phi\left(\gamma,\beta\right)=\mathbb{E}_x\,\log\int d\mu\left(\theta\right)\left(\sum_W\prod_i\sigma\left(W_i\theta_i\gamma\right)\prod_\mu\Theta\left(\frac{1}{\sqrt{N}}\sum_{i=1}^N x_i^\mu W_i\right)\right)^\beta \tag{7.9}$$

The result of the replica calculations, in the binary control case, is in fact identical to the large deviation analysis of chapter 4 (with the trivial mapping $\beta \to y$). Also in the case of continuous θ, we don't expect qualitatively different results. The fully stochastic case, instead, cannot be mapped onto the local entropy calculation as straightforwardly.

7.3 Equilibrium analysis and stability

The fully-stochastic settings allows us to obtain an explicit expression of the log-likelihood in the limit of large N, by simply applying the Central Limit Theorem (CLT). For later convenience, instead of using directly the parameters θ, we write the prior distribution $Q\left(W;\theta\right)$ through the induced magnetizations, according to:

$$Q_m\left(W\right) = \prod_{i=1}^{N}\left[\frac{1}{2}(1+m_i)\delta_{W_i,+1} + \frac{1}{2}(1-m_i)\delta_{W_i,-1}\right]. \tag{7.10}$$

where, clearly, we require $m_i \in [-1,1]$ $\forall i$.

In order to apply the CLT, to evaluate analytically the distribution over the outputs, we just need to compute the mean and variance of the random variable $\sum_{i=1}^{N} x_i^\mu W_i$, given by:

$$\sum_{i=1}^{N} x_i^\mu \left\langle W_i\right\rangle_{\theta_i,\gamma} = \sum_{i=1}^{N} x_i^\mu m_i \tag{7.11}$$

$$\sum_{i=1}^{N}\left(\left\langle \left(x_i^\mu W_i\right)^2\right\rangle_{\theta_i,\gamma} - \left\langle x_i^\mu W_i\right\rangle_{\theta_i,\gamma}^2\right) = \sum_{i=1}^{N}\left(1-m_i^2\right)x_i^2 \tag{7.12}$$

and transform the sum over W into a Gaussian integral. We can then write the log-likelihood function of equation (7.2) as:

$$\mathcal{L}(m) = \sum_{(x,y)\in\mathcal{D}} \log H\left(-\frac{y\sum_i m_i\, x_i}{\sqrt{\sum_i (1-m_i^2)x_i^2}}\right), \tag{7.13}$$

where, as usual, $H\left(y\right) = \int_y^\infty dy\, \frac{e^{-\frac{y^2}{2}}}{\sqrt{2\pi}} = \frac{1}{2}\mathrm{erfc}\left(\frac{y}{\sqrt{2}}\right)$.

We can therefore study the properties of the partition function:

$$Z = \int_{\Omega} \prod_i \mathrm{d}m_i \; \delta\left(\sum_i m_i^2 - q_* N\right) e^{\beta \mathcal{L}(m)} \tag{7.14}$$

where $\Omega = [-1,1]^N$, and β is the usual inverse temperature. The squared norm of m, $q_\star$, is constrained to a value ≤ 1, in order to be able to mimic the gradual increment of q_* in the training process (analogous of the scoping procedure in the calculations of the previous chapters). Otherwise, the minima of the energy are exactly the same of the standard binary Perceptron, since the log-likelihood $\mathcal{L}$ is maximized in all the corners associated to a solution of the binary problem, therefore the case $q_\star = 1$ is trivial. The partition function of equation 7.14 has an implicit dependence on the quenched disorder, represented by the training set $\mathcal{D} = \{x^\mu, y^\mu\}_{\mu=1}^{\alpha N}$.

We want to investigate the typical properties of this system in the thermodynamic limit and at fixed storage load α: as usual, we will employ the replica method, limiting our analysis to the RS ansatz for simplicity. In the following, we will denote with $\mathbb{E}_{\mathcal{D}}$ the expectation over the possible choices of training set (with i.i.d. input and outputs with zero mean and variance one). We can thus study the average asymptotic free entropy:

$$\phi = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathcal{D}} \log Z_N \tag{7.15}$$

In order to perform the disorder average, we replicate n times the system and consider the replicated partition function:

$$\mathbb{E}_D\, Z_N^n = \mathbb{E}_D \int_{\Omega} \prod_{a=1}^{n} \prod_{i=1}^{N} \mathrm{d}m_i^a \; \prod_{a=1}^{n} \delta\left(\sum_{i=1}^{N} (m_i^a)^2 - q_* N\right) \prod_{\mu=1}^{M} \prod_{a=1}^{n} H^{\beta}\left(\frac{y^\mu \sum_{i=1}^N x_i^\mu m_i^a}{\sqrt{N}}\right). \tag{7.16}$$

After averaging over the patterns and factorizing over the synaptic and the pattern indices, $i = 1, ..., N$ and $\mu = 1, ..., \alpha N$ respectively, we can write the leading order $\mathcal{O}(N)$ expression for the free entropy:

$$\mathbb{E}_{\mathcal{D}}\, Z_N^n \sim \int \prod_a \frac{\mathrm{d}\hat{q}_{aa}}{2\pi} \prod_{a<b} \frac{\mathrm{d}\hat{q}_{ab}\mathrm{d}q_{ab}}{2\pi} e^{N\phi[\hat{q},q]} \tag{7.17}$$

with the following definition for the replicated action:

$$\phi[\hat{q}, q] = -\frac{1}{2}\sum_{a,b} \hat{q}_{ab}\, q_{ab} + G_S + \alpha G_E \tag{7.18}$$

$$G_S[\hat{q}] = \log \int_\Omega \prod_a \mathrm{d}m_a \; e^{\frac{1}{2}\sum_{ab} \hat{q}_{ab} m_a m_b}, \tag{7.19}$$

$$G_E[q] = \log \int \prod_a \frac{\mathrm{d}\hat{\lambda}_a \mathrm{d}\lambda_a}{2\pi} \; e^{-\frac{1}{2}\sum_{ab} q_{ab}\hat{\lambda}_a\hat{\lambda}_b + i\hat{\lambda}_a\lambda_a} \prod_a H^\beta(\lambda_a). \tag{7.20}$$

Notice that the overlap $q_{aa} \equiv q_*$ is fixed by the L_2 norm constraint. If we denote with $\ll \bullet \gg_S$ and $\ll \bullet \gg_E$ the expectations taken according to the single-body partition function in the logarithms of equation (7.19) and equation (7.20), the saddle point evaluation of the replicated partition function yields:

$$\hat{q}_{ab} = -\alpha \ll \hat{\lambda}_a\hat{\lambda}_b \gg_E \qquad a > b, \tag{7.21}$$

$$q_{ab} = \ll m_a m_b \gg_S \qquad a > b, \tag{7.22}$$

$$q_{aa} \equiv q_* = \ll m_a^2 \gg_S \,. \tag{7.23}$$

where the last equation can be used as an implicit equation for determining the conjugated order parameter $\hat{q}_{aa}$.

In order to continue the computation and obtain the original typical free entropy as an analytic continuation to $n \to 0^+$, we need to make an Ansatz on the structure of the order parameters. We therefore consider the Replica Symmetric (RS) Ansatz:

- $q_{ab} = q_0$, $\hat{q}_{ab} = \hat{q}_0$ for $a \neq b$.
- $\hat{q}_{aa} = \hat{q}_1$ $\forall a$. Of course, $q_1 = q_\star$ because of the constraint.

The computation follows the same lines of the ones presented in the previous chapters, and the final RS prediction for the average free entropy is found to be:

$$\phi_{RS} = \underset{q_0, \hat{q}_0, \hat{q}_*}{\mathrm{extr}} \quad \frac{1}{2}(q_0\hat{q}_0 - q_*\hat{q}_1) + G_S(\hat{q}_0, \hat{q}_1) + \alpha G_E(q_0, q_*) \tag{7.24}$$

where:

$$\mathcal{G}_S(\hat{q}_0,\hat{q}_1) = \int \mathcal{D}z_0 \log \int_{-1}^{1} \mathrm{d}m \; e^{\frac{1}{2}(\hat{q}_1-\hat{q}_0)m^2+\sqrt{\hat{q}_0}z_0 m}, \tag{7.25}$$

$$= -\frac{1}{2}\log a + \int \mathcal{D}z_0 \;\log\left[\sqrt{\frac{\pi}{2}}e^{-\frac{b^2}{2a}}\left(\mathrm{erfi}\left(\frac{a-b}{\sqrt{2}\sqrt{a}}\right)+\mathrm{erfi}\left(\frac{a+b}{\sqrt{2}\sqrt{a}}\right)\right)\right] \tag{7.26}$$

$$\mathcal{G}_E(q_0,q_*) = \int \mathcal{D}z_0 \log \int \mathcal{D}z_1 \; H^\beta\left(-\frac{\sqrt{q_0}z_0+\sqrt{q_*-q_0}z_1}{\sqrt{1-q_*}}\right). \tag{7.27}$$

As usual, we would be interested in studying the limiting case $\beta \to \infty$, but unfortunately the RS solution becomes locally unstable for large β. We conjecture that ϕ_{RS} gives good predictions at low β, while for exploring lower temperatures, $\beta \gg 1$ (required in the context of the maximization of the log-likelihood), a symmetry-broken replica analysis is needed.

In order to keep things simple, we will present the results obtained for large values of β still in the RS stable region. The local stability criterion of the free energy functional of equation (7.18) at the RS stationary point, involving the eigenvalues of the Hessian (see [49] for the original calculation), can be recast as the condition:

$$\alpha\gamma_E\gamma_S < 1. \tag{7.28}$$

where γ_E and γ_S are the relevant eigenvalues of the Hessians, of the $G_E[q]$ and $G_S[\hat{q}]$ functionals, at small values of n. They can be computed as:

$$\gamma_E = \int \mathcal{D}z_0 \left[\overline{\hat{u}^2}(z_0) - \left(\overline{\hat{u}}(z_0)\right)^2\right]^2, \tag{7.29}$$

$$\gamma_S = \int \mathcal{D}z_0 \left[\overline{m^2}(z_0) - \left(\overline{m}(z_0)\right)^2\right]^2. \tag{7.30}$$

where the averages in the last equations are defined as:

$$\overline{\hat{u}^k}(z_0) \equiv \frac{\int \frac{\mathrm{d}\hat{u}\mathrm{d}u}{2\pi}\; \hat{u}^k \, e^{-\frac{1}{2}(q_*-q_0)\hat{u}^2+i\hat{u}u+i\hat{u}\sqrt{q_0}z_0}H^\beta(u)}{\int \frac{\mathrm{d}\hat{u}\mathrm{d}u}{2\pi}\; e^{-\frac{1}{2}(q_*-q_0)\hat{u}^2+i\hat{u}u+i\hat{u}\sqrt{q_0}z_0}H^\beta(u)} \tag{7.31}$$

$$\overline{m^k}(z_0) \equiv \frac{\int_{-1}^{1}\mathrm{d}m\; m^k\, e^{\frac{1}{2}(\hat{q}_1-\hat{q}_0)m^2+\sqrt{\hat{q}_0}z_0 m}}{\int_{-1}^{1}\mathrm{d}m\; e^{\frac{1}{2}(\hat{q}_1-\hat{q}_0)m^2+\sqrt{\hat{q}_0}z_0 m}} \tag{7.32}$$

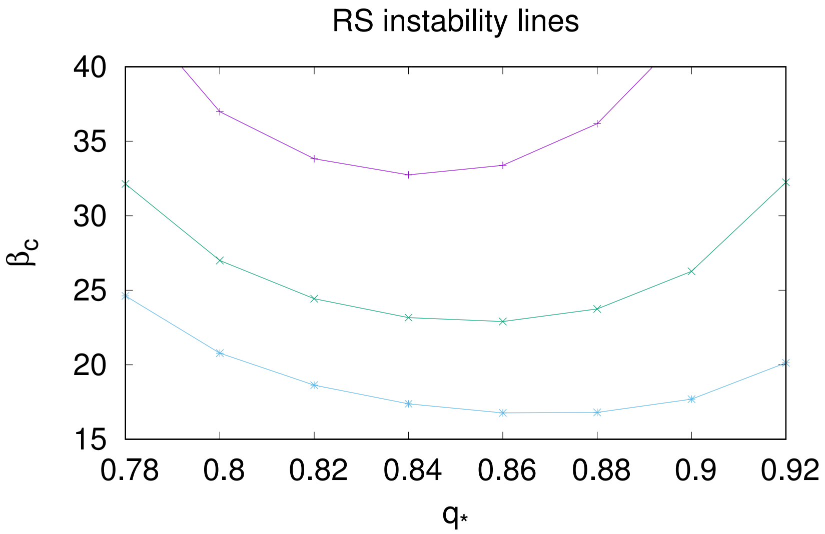


Fig. 7.1 **Critical value** β_c **for the stability** of the RS solution for different storage loads α, as a function of q_*. Above β_c the RS solution is locally unstable.

In figure 7.1, we can see the behavior of the stability line $\beta_c(q_*)$, at various values of α: the critical temperature is always found at finite (but large) values.

7.4 Gradient Descent in the Stochastic Perceptron

We have reduced the stochastic learning problem over a binary machine to an optimization problem over the parameters θ of the probability distribution, which can be assumed to be continuous and without any residual stochasticity. We can thus try to perform the optimization with the usual methods of continuous optimization, in particular by using the gradient descent method and its variants.

The gradient of the loss-function $-\mathcal{L}(m)$ produces the simple update rules:

$$m_i^{t+1} \leftarrow \mathrm{clamp}\bigg(m_i^t + \eta\, \partial_{m_i^t} \mathcal{L}\left(m^t\right)\bigg). \tag{7.33}$$

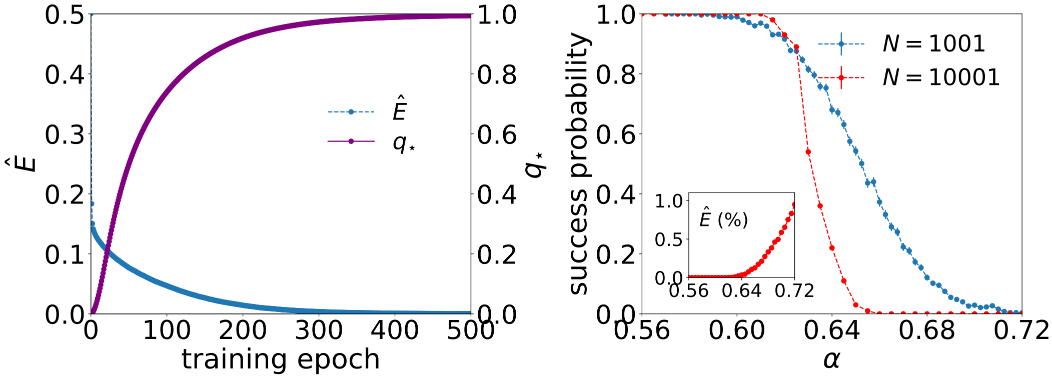


Fig. 7.2 (*Left*) **The training error** $\hat{E}$ and the squared norm against the number of training epochs, for $\alpha = 0.55$ and $N = 10001$, averaged over 100 samples. (*Right*) **Success probability** in the classification task as a function of the load α for networks of size $N = 1001$ and $N = 10001$ averaging 1000 and 100 respectively. We set the inverse temperature to $\beta = 20$, where the RS results are stable and supposedly correct, but also quantitatively close to the $\beta = +\infty$ limit.

where η is a suitable learning rate and the clamping function $\mathrm{clamp}(x) := \max(-1, \min(1, x))$ is required to ensure that the magnetizations do not leave the allowed interval $[-1, 1]$. Their initial value, instead, can be set to be small, distributed as $m_i^0 \sim \mathcal{N}(0, 1/\sqrt{N})$. At each epoch t in the GD dynamics, the training error $\hat{E}(t) = \frac{1}{M}\mathcal{H}(\hat{W}^t)$ can be computed with respect to the mode of the distribution, the clipped configuration $\hat{W}_i^t = \mathrm{sign}(m_i^t)$.

In figure 7.2, we can see the evolution of the network during the above defined learning procedure, in the case of full-batch GD. The network approaches zero training error while the L_2 norm of the magnetizations (corresponding to the order parameter q_* in the replica calculations) reaches one. Therefore, Q_m focuses around a single corner of the distribution, a binary synaptic configuration, as the training procedure progresses. This natural dynamical flow is similar to the scoping procedure on the coupling parameter γ, which had to be manually performed in the local entropy inspired algorithms presented in the previous sections.

We also show the performance of this algorithm as a solver, by measuring over many realizations of $\mathcal{D}$ the probability of finding a solution of the binary problem: in figure 7.2, we see its behavior as a function of the storage load $\alpha = M/N$. In the case of a pure full-batch GD (at zero temperature), the measured algorithmic capacity is approximately $\alpha_{GD} \approx 0.63$. This value has to be compared with the capacities achieved by the algorithms described in the previous chapters $\alpha_{MP} \in [0.6, 0.74]$.

In our numerical experiments, also many variants of the GD procedure of equation (7.33) have achieved qualitatively similar performance:

- the *natural gradient* $(1 - m_i^2)\partial_{m_i}$[86], where the term multiplying the derivative of the log-likelihood helps preventing the magnetizations from saturating immediately ±1, even with high learning rates;
- the explicit gradient on the fields $\theta_i = \operatorname{arctanh}(m_i^t)$, which is slightly more expensive than the previous ones (because of the repeated applications of tanh and arctanh functions), but allows one to avoid the clamping of the magnetizations.
- stochastic gradient descent (SGD), where the gradient is evaluated only over a random mini-batches of the training set, injecting noise into the learning procedure: this method is very effective for avoiding local minima, and reaches a slightly higher algorithmic capacity than standard GD;
- more sophisticated updates rules involving some message-passing iterations, which could be derived through an on-line Bayesian learning approach [87, 88].

7.5 Energy of a clipped configuration

Building on the theoretical results presented above, we can now make a comparison between analytic predictions and numerical results, with respect to some properties of the mode of the distribution $Q(W; m)$, namely the clipped configuration $\hat{W}_i = \operatorname{sign}(m_i)$.

The probability of a classification error on a pattern of the training set, p_e, associated to a clipped configuration obtained from a typical magnetization $m\,(q_\star)$ (sampled according to the Gibbs measure (7.14)), is defined as:

$$p_e = \lim_{N\to\infty} \frac{1}{\alpha N} \mathbb{E}_{\mathcal{D}} \left[\sum_{(x,y)\in\mathcal{D}} \left\langle \Theta \left(-y \sum_{i=1}^{N} \operatorname{sign}(m_i) x_i \right) \right\rangle_m \right] \tag{7.34}$$

where $\langle \bullet \rangle$ is the thermal average, with an implicit dependence on $\mathcal{D}$, q_* and β. We can relate this quantity with the typical value of the energy of a clipped configuration, which can be computed analytically within the replica framework:

$$\begin{aligned} E = \quad & 1 - \lim_{N\to\infty} \lim_{n\to 0} \mathbb{E}_{\mathcal{D}} \left[\int_\Omega \prod_{a,i} dm_i^a \prod_a \delta \left(\sum_i (m_i^a)^2 - q_{aa} N \right) \times \right. \\ & \left. \times \Theta \left(\frac{\sum_i \operatorname{sign}(m_i^1) x_i^1}{\sqrt{N}} \right) e^{\beta \sum_a \mathcal{L}(m^a)} \right] \end{aligned} \tag{7.35}$$

At this point, we need to introduce two distinct order parameters, $q_{ab} = \frac{1}{N}\sum_i m_i^a m_i^b$ and $p_a = \frac{1}{N}\sum_i \operatorname{sign}(m_i^1) m_i^a$, where q_{ab} represents the typical overlap between the magnetizations of different replicas, while p_a describes the overlap between a magnetization and the clipped solution. We therefore get the following expression for E:

$$\begin{aligned} E = & 1 - \lim_{N\to\infty} \lim_{n\to 0} \int \prod_{a<b} dq_{ab} \prod_{a\leq b} \frac{d\hat{q}_{ab}}{2\pi} \int \prod_a \frac{dp_a d\hat{p}_a}{2\pi} \\ & \times e^{N\phi(q_{ab},\hat{q}_{ab},p_a,\hat{p}_a)} \mathcal{G}'_E (q_{ab}, p_a) \end{aligned} \tag{7.36}$$

with the following definitions:

$$\phi (q_{ab}, \hat{q}_{ab}, p_a, \hat{p}_a) = -\frac{1}{2} \sum_{a,b} \hat{q}_{ab} q_{ab} - \sum_a \hat{p}_a p_a + G_S (\hat{q}_{ab}, \hat{p}_a) + \alpha G_E (q_{ab}) \tag{7.37}$$

$$\begin{aligned} \mathcal{G}'_E (q_{ab}, p_a) = & \int \prod_a \frac{du_a d\hat{u}_a}{2\pi} \int \frac{d\tilde{u} d\hat{\tilde{u}}}{2\pi} \prod_a H^\beta \left(-\frac{u_a}{\sqrt{1-q_*}} \right) \Theta(\tilde{u}) \times \\ & \times \exp \left(i \sum_a u_a \hat{u}_a - \frac{1}{2} \hat{\tilde{u}}^2 + i \tilde{u} \hat{\tilde{u}} - \frac{1}{2} \sum_{a,b} q_{ab} \hat{u}_a \hat{u}_b - \hat{\tilde{u}} \sum_a p_a \hat{u}_a \right) \end{aligned} \tag{7.38}$$

$$\mathcal{G}_S[\hat{q}, \hat{p}] = \log \int_{-1}^{1} \prod_a dm^a \exp \left(\frac{1}{2} \sum_{a,b} \hat{q}_{ab} m^a m^b + \operatorname{sign}(m^1) \sum_a \hat{p}_a m^a \right) \tag{7.39}$$

$$\mathcal{G}_E[q] = \log \int \prod_a \frac{du_a d\hat{u}_a}{2\pi} \prod_a H^\beta \left(-\frac{u_a}{\sqrt{1-q_*}} \right) \exp \left(i \sum_a u_a \hat{u}_a - \frac{1}{2} \sum_{a,b} q_{ab} \hat{u}_a \hat{u}_b \right) \tag{7.40}$$

After a proper Ansatz on the structure of the order parameters, the expression can be evaluated at the saddle point, in the thermodynamic limit $N \to \infty$. In the Replica Symmetric Ansatz we pose:

- As before: $q^{aa} = q_\star$, $q^{ab} = q$ for $a \neq b$.
- $p^1 = p$, $p^a = \tilde{p}$ for $a \neq 1$.

and we obtain the following expressions for the action ϕ and $\mathcal{G}'_E$:

$$\phi_{RS} = -\frac{1}{2} n q_* \hat{q}_* - \frac{1}{2} n(n-1) q\hat{q} - p\hat{p} - (n-1)\tilde{p}\hat{\tilde{p}} + \ln \mathcal{G}_S\left(\hat{q}_*, \hat{q}, \hat{p}, \hat{\tilde{p}}\right) + \ln \mathcal{G}_E\left(q_*, q\right) \tag{7.41}$$

$$\begin{aligned}\mathcal{G}'_E\left(q_*, q, p, \tilde{p}\right) &= \int \mathcal{D}z \int \mathcal{D}\hat{\tilde{u}} \int \frac{d\tilde{u}}{\sqrt{2\pi}} \theta(\tilde{u}) \exp\left(i\tilde{u}\hat{\tilde{u}}\right) \\ &\times \left[\int \frac{du d\hat{u}}{2\pi}\, H^\beta\left(-\frac{u}{\sqrt{1-q_*}}\right) \exp\left(-\frac{1}{2}\left(q_* - q\right)\hat{u}^2 + i\sqrt{q}z\hat{u} + iu\hat{u} - \tilde{p}\hat{u}\hat{\tilde{u}}\right)\right]^{n-1} \\ &\times \int \frac{du d\hat{u}}{2\pi}\, H^\beta\left(-\frac{u}{\sqrt{1-q_*}}\right) \exp\left(-\frac{1}{2}\left(q_* - q\right)\hat{u}^2 + i\sqrt{q}z\hat{u} + iu\hat{u} - p\hat{u}\hat{\tilde{u}}\right)\end{aligned} \tag{7.42}$$

We also defined:

$$\begin{aligned}\mathcal{G}_S\left(\hat{q}_*, \hat{q}, \hat{p}, \hat{\tilde{p}}\right) = \log \int \mathcal{D}z \int_{-1}^{1} dm^1 &\left[\int_{-1}^{1} dm \exp\left(\sqrt{\hat{q}}zm + \frac{1}{2}\left(\hat{q}_* - \hat{q}\right)m^2 + \hat{\tilde{p}}\,\mathrm{sign}(m^1)m\right)\right]^{n-1} \\ &\times \exp\left(\sqrt{\hat{q}}zm^1 + \frac{1}{2}\left(\hat{q}_* - \hat{q}\right)\left(m^1\right)^2 + \hat{p}\,\mathrm{sign}(m^1)m^1\right)\end{aligned} \tag{7.43}$$

$$\mathcal{G}_E\left(q_*, q\right) = \log \int \mathcal{D}z \left[\int \frac{du d\hat{u}}{2\pi}\, H^\beta\left(-\frac{u}{\sqrt{1-q_*}}\right) \exp\left(-\frac{1}{2}\left(q_* - q\right)\hat{u}^2 + i\left(\sqrt{q}z + u\right)\hat{u}\right)\right]^n \tag{7.44}$$

Now we can substitute $n = 0$ and recover the expression of the ensemble average:

$$\phi\left(q_*, q, \hat{q}_*, \hat{q}, p, \tilde{p}, \hat{p}, \hat{\tilde{p}}\right) = -p\hat{p} + \tilde{p}\hat{\tilde{p}} + \ln \mathcal{G}_S\left(\hat{q}_*, \hat{q}, \hat{p}, \hat{\tilde{p}}\right) \tag{7.45}$$

$$\mathcal{G}'_E\left(q_*, q, p, \tilde{p}\right) = \int \mathcal{D}z \frac{\int \mathcal{D}u\, H\left(-\frac{\left(\frac{(p-\tilde{p})}{\sqrt{q_*-q}}u - \frac{\tilde{p}}{\sqrt{q}}z\right)}{\sqrt{1-\frac{\tilde{p}^2}{q}-\frac{(p-\tilde{p})^2}{q_*-q}}}\right)\; H^\beta\left(-\frac{\sqrt{q_*-q}u-\sqrt{q}z}{\sqrt{1-q_*}}\right)}{\int \mathcal{D}u\; H^\beta\left(-\frac{u\sqrt{q_\star-q}-\sqrt{q}z}{\sqrt{1-q_*}}\right)} \tag{7.46}$$

where we substituted $\mathcal{G}_E\left(q_*, q\right) = 1$ and:

$$\mathcal{G}_S\left(\hat{q}_*, \hat{q}, \hat{p}, \hat{\tilde{p}}\right) = \int \mathcal{D}z_0 \int_{-1}^{1} dm^1 \frac{\exp\left(\sqrt{\hat{q}_0}zm^1 + \frac{1}{2}\left(\hat{q}_* - \hat{q}\right)\left(m^1\right)^2 + \hat{p}\,\mathrm{sign}(m^1)m^1\right)}{\int_{-1}^{1} dm \exp\left(\sqrt{\hat{q}}zm + \frac{1}{2}\left(\hat{q}_* - \hat{q}\right)m^2 + \hat{\tilde{p}}\,\mathrm{sign}(m^1)m\right)} \tag{7.47}$$

We do not need to compute again the saddle point value for the parameters obtained in the calculation presented above (in section 7.3), yet only the typical overlaps between the clipped configuration and the other magnetizations:

$$p = \int \mathcal{D}z_0 \frac{\int_{-1}^{1} dm\, \mathrm{sign}(m)m\; e^{\frac{1}{2}(\hat{q}_*-\hat{q}_0)m^2+\sqrt{\hat{q}_0}z_0 m}}{\int_{-1}^{1} dm\, e^{\frac{1}{2}(\hat{q}_*-\hat{q}_0)m^2+\sqrt{\hat{q}_0}z_0 m}} \tag{7.48}$$

$$\tilde{p} = \int \mathcal{D}z_0 \frac{\left(\int_{-1}^{1} dm\; e^{\frac{1}{2}(\hat{q}_*-\hat{q}_0)m^2+\sqrt{\hat{q}_0}z_0 m}\mathrm{sign}(m)\right)\left(\int_{-1}^{1} dm\; m\, e^{\frac{1}{2}(\hat{q}_*-\hat{q}_0)m^2+\sqrt{\hat{q}_0}z_0 m}\right)}{\left[\int_{-1}^{1} dm\, e^{\frac{1}{2}(\hat{q}_*-\hat{q}_0)m^2+\sqrt{\hat{q}_0}z_0 m}\right]^2} \tag{7.49}$$

In figure 7.3, we compare the analytical prediction of the average energy with the values obtained in the numerical experiments. For large β (i.e., at low temperatures) the clipped configuration becomes a zero-energy solution of the problem as q_* approaches one. While the numerical results in the greedy full-batch GD procedure are sub-optimal compared to the theoretical predictions, the values of the training error as a function of q_* approaches the analytic curve if the GD dynamics is controlled in the following way: a given value of q_* is fixed, the system is equilibrated for 10^3 iterations, and only then $q_\star$ is allowed to increase.

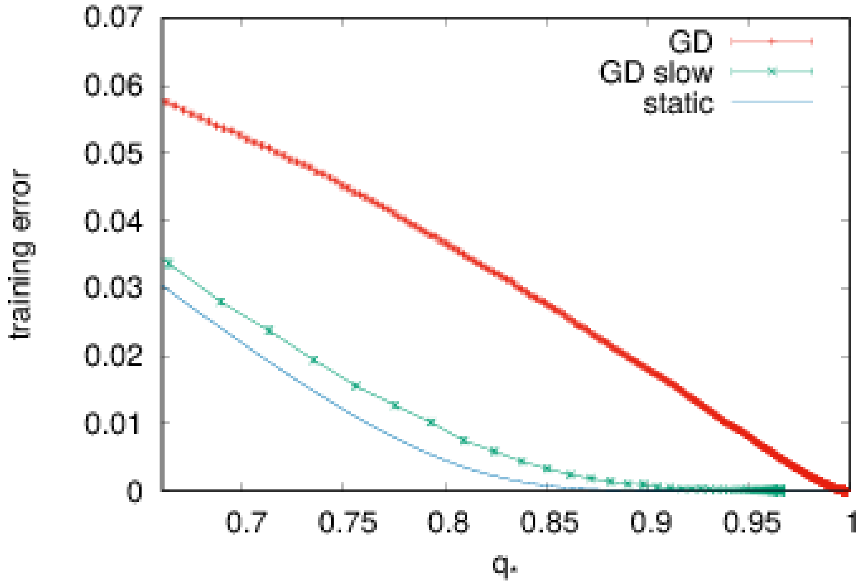


Fig. 7.3 **Energy of the clipped center** versus the norm of the control variables q_*. We show the static prediction of equation (7.34), and numerical results from the GD algorithm and a GD algorithm variant where after each update we rescale the norm of m to q_* until convergence before moving to the next value of q_*. For GD we average over 20 random realization of the training set with $N = 10001$.

7.6 Franz-Parisi potential

We know from the previous chapters that while most solutions of the binary Perceptron are isolated, a sub-dominant but still exponentially large number of solutions belong to a dense connected cluster: these are the only type of solutions which can be accessed by sub-exponential algorithms. Since the stochastic synapses formulation allowed us to define an alternative way of finding binary solutions, we also expect this algorithm to end up in the same sub-dominant structures.

It is, in fact, possible to show that the clipped configurations of the stochastic binary Perceptron typically belong to dense regions of solutions, when q_* is high enough. We can study the Franz-Parisi potential (cf. with section 2.6), by fixing an extensive radius $D\,N$ and counting the number of binary solutions at this Hamming distance from a clipped configuration. It is interesting to make a comparison between the possible clipped configurations, corresponding to three different choices for the measures from which m is extracted. We can treat all these cases in parallel, by keeping an implicit dependence on the domain of integration Ω^N of m, and on the energy function f:

1. The typical solutions of the binary Perceptron, with $\Omega = \{-1, 1\}$ (implying $q_\star = 1$) and $f = \Theta(\cdot)$: this corresponds exactly to the case presented in sub-section 2.6.1.

2. The typical solutions of the continuous Perceptron, with $\Omega = \{-\infty, \infty\}$ and $f = \Theta(\cdot)$: in this case, the clipped configurations are qualitatively similar to those that would be found algorithmically by employing the Clipped Perceptron algorithm (cf. with chapter 3).

3. The maximum-likelihood solutions of the binary Perceptron, with $\Omega = [-1, 1]$ and $f = H\left(-\frac{\cdot}{\sqrt{1-q_\star}}\right) = \mathcal{L}(m)$.

We thus introduce the coupled partition function:

$$\begin{aligned}\mathcal{Z}(d, m) = \sum_{\{W_i\}} \prod_{(x,y)\in\mathcal{D}} \Theta\left(y \sum_i W_i x_i\right) \\ \times\, \delta\left(N(1-2D) - \sum_i \operatorname{sign}(m_i) W_i\right)\end{aligned} \tag{7.50}$$

In order to characterize the typical behavior of the coupled system, we need to take the expectation $\langle\bullet\rangle_m$ over m, specializing to the three cases listed above.

The distance constraint that couples the binary Perceptron system with the reference system of the clipped configuration, is equivalent to the requirement of a fixed overlap between $\frac{1}{N}\sum_i W_i \operatorname{sign}(\tilde{W}_i) \equiv p$. The Franz-Parisi free entropy is thus defined as:

$$S(p) \equiv \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathcal{D}} < \log \sum_{\{W_i=\pm1\}} \delta\left(\sum_i \operatorname{sign}(\tilde{W}_i) W_i - pN\right) \prod_\mu \theta\left(\sum_i x_i^\mu W_i\right) >_{\tilde{W};\mathcal{D}} \tag{7.51}$$

with:

$$< \bullet >_{\tilde{W};\mathcal{D}} \equiv \frac{\int_\Omega \prod_i \mathrm{d}\tilde{W}_i \;\bullet\; \delta\left(\sum_i \tilde{W}_i^2 - q_* N\right) \prod_\mu f\left(\frac{\sum_i x_i^\mu \tilde{W}_i}{\sqrt{N}}\right)}{\int_\Omega \prod_i \mathrm{d}\tilde{W}_i \; \delta\left(\sum_i \tilde{W}_i^2 - q_* N\right) \prod_\mu f\left(\frac{\sum_i x_i^\mu \tilde{W}_i}{\sqrt{N}}\right)} \tag{7.52}$$

The calculation follows the same steps of the ones presented in the previous chapters, giving in the RS Ansatz the following expression:

$$\phi_{FP}(S) = -\frac{1}{2}\hat{Q}\left(1-Q\right) + \hat{s}_0 s_0 - \hat{s}_1 s_1 - \hat{p}p + G_S + \alpha G_E \tag{7.53}$$

$$G_S = \int \mathcal{D}z_0 \, \frac{\int_\Omega \mathrm{d}\mu\left(\tilde{W}\right) \int \mathcal{D}\eta \; e^{\frac{1}{2}(\hat{q}_1-\hat{q}_0)\tilde{W}^2+\sqrt{\hat{q}_0}z_0\tilde{W}} A_S(\tilde{W},\eta,z_0)}{\int_\Omega \mathrm{d}\mu\left(\tilde{W}\right) \tilde{W} \; e^{\frac{1}{2}(\hat{q}_1-\hat{q}_0)\tilde{W}^2+\sqrt{\hat{q}_0}z_0\tilde{W}}} \tag{7.54}$$

$$G_E = \int \mathcal{D}z_0 \frac{\int \mathcal{D}\eta\mathcal{D}z_1 \; f\left(\sqrt{q_0}z_0 + \sqrt{a}z_1 + \frac{s_1-s_0}{\sqrt{b}}\eta\right) \log H\left(-\frac{\sqrt{b}\eta+\frac{s_0}{\sqrt{q_0}}z_0}{\sqrt{1-Q}}\right)}{\int \mathcal{D}z_1 \; f\left(\sqrt{q_0}z_0 + \sqrt{q_* - q_0}z_1\right)} \tag{7.55}$$

where we defined:

$$A_S(\tilde{W},\eta,z_0) = \log 2\cosh\left((\hat{s}_1-\hat{s}_0)\tilde{W} + \hat{p}\,\mathrm{sign}(\tilde{W}) + \sqrt{\frac{\hat{Q}\hat{q}_0-\hat{s}_0^2}{\hat{q}_0}}\eta + \frac{\hat{s}_0}{\sqrt{\hat{q}_0}}z_0\right) \tag{7.56}$$

$$a = q_* - q_0 - \frac{\left(s_1-s_0\right)^2}{\left(Q-s_0\right)}\left(1 - \frac{s_0\left(q_0-s_0\right)}{\left(Qq_0-s_0^2\right)}\right) \tag{7.57}$$

$$b = \frac{Qq_0 - s_0^2}{q_0} \tag{7.58}$$

Now we can differentiate between the three possible scenarios:

1. Binary Perceptron:

$$G_S = \int \mathcal{D}z_0 \, \frac{\sum_{\tilde{W}=\pm1} \int \mathcal{D}\eta \; e^{\sqrt{\hat{q}_0}z_0\tilde{W}} A_S(\tilde{W},\eta,z_0)}{2\cosh\left(\sqrt{\hat{q}_0}z_0\tilde{W}\right)} \tag{7.59}$$

$$A_S(\tilde{W},\eta,z_0) = \log 2\cosh\left((\hat{s}_1-\hat{s}_0)\tilde{W} + \sqrt{\frac{\hat{Q}\hat{q}_0-\hat{s}_0^2}{\hat{q}_0}}\eta + \frac{\hat{s}_0}{\sqrt{\hat{q}_0}}z_0\right) \tag{7.60}$$

2. Continuous Perceptron, in the assumption $\hat{q}_0 > \hat{q}_1$:

$$G_S = \int \mathcal{D}\tilde{W}\mathcal{D}z_0 \ \log 2\cosh\left[\left(\frac{\hat{s}_1 - \hat{s}_0}{\hat{q}_0 - \hat{q}_1}\sqrt{2\hat{q}_0 - \hat{q}_1} + \frac{\hat{s}_0}{\sqrt{2\hat{q}_0 - \hat{q}_1}}\right)\tilde{W} + \right.$$
$$\left. + \gamma \mathrm{sign}(\tilde{W}) + \sqrt{\hat{Q} - \frac{\hat{s}_0^2}{2\hat{q}_0 - \hat{q}_1}}\, z_0\right] \tag{7.61}$$

3. Stochastic binary Perceptron:

$$G_S = \int \mathcal{D}z_0 \frac{\tilde{I}_+ + \tilde{I}_-}{\tilde{I}_0} \tag{7.62}$$

where we defined:

$$\tilde{I}_+ = a\int \mathcal{D}\eta \ f_+(a\eta + b)\times$$
$$\times \log 2\cosh\left(\sqrt{\frac{\hat{Q}\hat{q}_0 - \hat{s}_0^2}{\hat{q}_0} + (\hat{s}_1 - \hat{s}_0)^2(a\eta + b) + \hat{p} + \frac{\hat{s}_0}{\sqrt{\hat{q}_0}}z_0}\right) \tag{7.63}$$

$$\tilde{I}_- = a\int \mathcal{D}\eta \ f_-(a\eta + b)\times$$
$$\times \log 2\cosh\left(\sqrt{\frac{\hat{Q}\hat{q}_0 - \hat{s}_0^2}{\hat{q}_0} + (\hat{s}_1 - \hat{s}_0)^2(a\eta + b) - \hat{p} + \frac{\hat{s}_0}{\sqrt{\hat{q}_0}}z_0}\right) \tag{7.64}$$

$$\tilde{I}_0 = \sqrt{\frac{\pi}{2|\hat{q}_1 - \hat{q}_0|}} \times \left(\mathrm{erf}_\epsilon\left(\frac{|\hat{q}_1 - \hat{q}_0| + \sqrt{\hat{q}_0}z_0}{\sqrt{2|\hat{q}_1 - \hat{q}_0|}}\right) + \mathrm{erf}_\epsilon\left(\frac{|\hat{q}_1 - \hat{q}_0| - \sqrt{\hat{q}_0}z_0}{\sqrt{2|\hat{q}_1 - \hat{q}_0|}}\right)\right) \tag{7.65}$$

$$a = \sqrt{\frac{\hat{Q}\hat{q}_0\left(\hat{q}_0 - \hat{q}_1\right) + \hat{q}_0\left(\hat{s}_1 - 2\hat{s}_0\right) + \hat{q}_1\hat{s}_0^2}{\left(\hat{q}_0 - \hat{q}_1\right)\left(\hat{q}_0\left(\hat{Q} + \left(\hat{s}_0 - \hat{s}_1\right)^2\right) - \hat{s}_0^2\right)}} \tag{7.66}$$

$$b = \frac{\hat{q}_0 z_0}{(\hat{q}_0 - \hat{q}_1)}\frac{\hat{s}_1 - \hat{s}_0}{\sqrt{Q\hat{q}_0 - \hat{s}_0^2 + \hat{q}_0(\hat{s}_1 - \hat{s}_0)^2}} \tag{7.67}$$

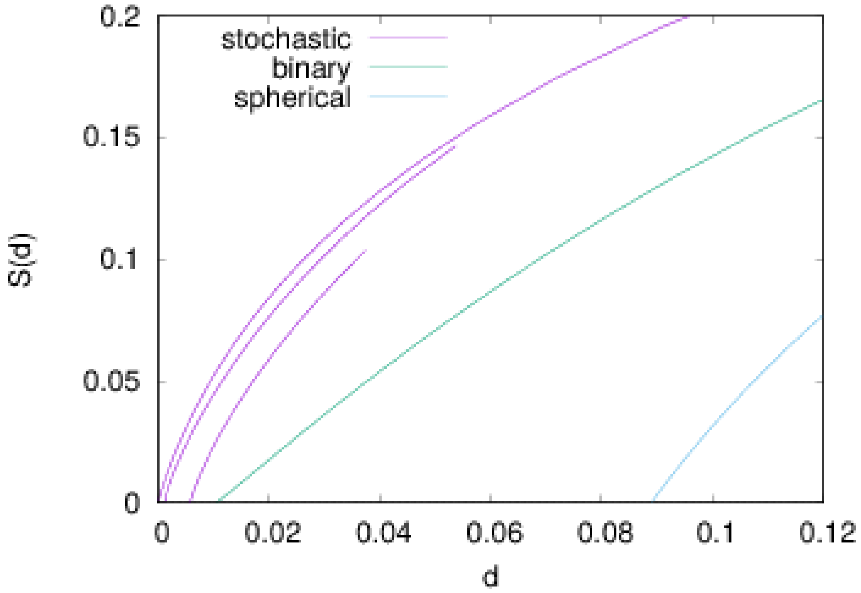


Fig. 7.4 **Local entropy of binary solutions** at fixed distance d from clamped configurations (CCs) of the spherical, binary and stochastic Perceptron ($q_* = 0.7, 0.8$ and 0.9 from bottom to top) at thermodynamic equilibrium. In both figures $\alpha = 0.55$, also $\beta = 20$ for the stochastic Perceptron and $\beta = \infty$ for the spherical and binary ones.

and the function erf_ϵ is defined as:

$$\mathrm{erf}_\epsilon\left(\cdot\right) = \begin{cases} \mathrm{erf}\left(\cdot\right) & \hat{q}_1 - \hat{q}_0 > 0 \\ \mathrm{erfi}\left(\cdot\right) & \hat{q}_1 - \hat{q}_0 < 0 \end{cases} \tag{7.68}$$

In figure 7.4, we see the comparison between the entropy $\mathcal{S}(D)$ obtained in the stochastic Perceptron model, and the analogous entropies in the two other cases. It is clear that, as q_* is increased, the zero-entropy gap between the clipped configuration and the nearest binary solutions closes: this clearly indicates that the clipped maximum likelihood configuration belongs to the dense cluster of solutions.

We also performed some numerical experiments for a validation of the theoretical predictions. The reference configuration $W = \mathrm{sign}(m)$ is selected by running the GD algorithm until a given configuration, at a fixed norm q_*, is reached. Then the single instance Franz-Parisi entropy is computed through the Belief Propagation (BP) algorithm (see section 3.1). In figure 7.3, we can

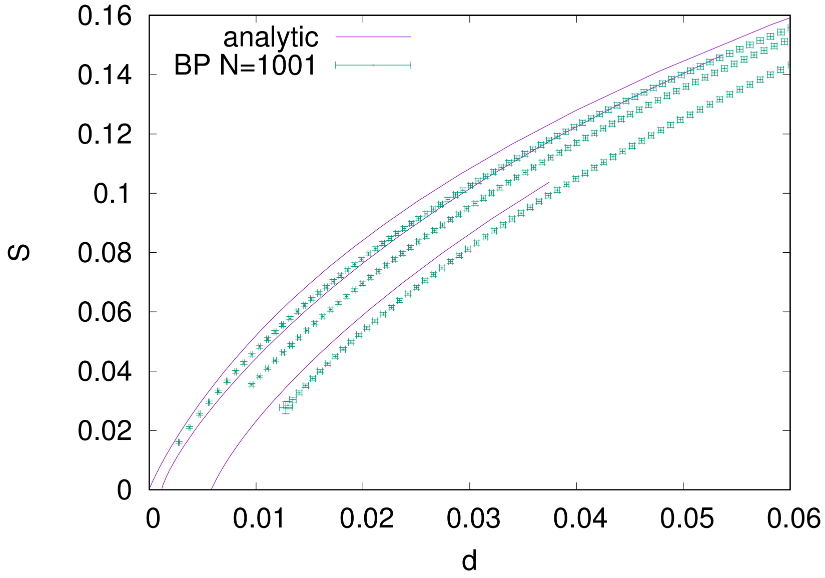


Fig. 7.5 **Franz-Parisi potential**, numerical estimation and theoretical predictions. Dashed green and violet lines, theoretical Franz-Parisi entropy for different values of q_*, $0.8, 0.9$ respectively. Solid blue and yellow lines, numerical estimate of Franz-Parisi entropy for different values of q_*, $0.8, 0.9$ respectively averaged over 100 samples. $\alpha = 0.6$.

see a direct comparison with the analytical curves for two different values of q_*. Again, the numerical results seem to be sub-optimal: this is not only due to finite size effects but also to the greedy nature of the full-batch GD algorithm we employed for finding the reference configuration.

7.7 Binary control variables and Dropout

It is also interesting to consider the case in which the control variables m are constrained to the corners of the hyper-cube of side $\sqrt{q_\star}$ (therefore $m_i = \sqrt{q_*}\tilde{W}_i$, with $\tilde{W}_i \in \{-1, 1\}$, and $\sum_i m_i^2 = q_\star N$). In this case the computation becomes simpler, since the likelihood can be written as:

$$\begin{aligned} \mathcal{L}(m) &= \sum_{(x,y)\in\mathcal{D}} H\left(-\frac{\sum_{i=1}^{N}\sqrt{q_\star}\tilde{W}_i x_i^\mu}{\sqrt{\sum_i\left(1-q_\star\tilde{W}_i^2\right)}}\right) \\ &= \sum_{(x,y)\in\mathcal{D}} H\left(-\frac{\rho}{\sqrt{N}}\sum_{i=1}^{N}\tilde{W}_i x_i^\mu\right) \end{aligned} \tag{7.69}$$

with the definition $\rho = \sqrt{q_\star}/\sqrt{1-q_\star}$. It is clear that, in the limit $q_* \to 1$, the new control parameter $\rho \to \infty$ and the error function H becomes an Heaviside Θ-function, recovering exactly the binary Perceptron model.

This formulation allows a nice connection with the Dropout/Dropconnect schemes, commonly employed in the Deep Learning context. The Dropout technique was introduced in [89], as a strategy for uncorrelating the hidden units, helping against the vanishing gradient [90] and avoiding over-fitting during the training of large ANNs. In recent years this method has become one of the most successful heuristics for improving the generalization properties of huge feed-forward models. From the practical point of view, in Dropout the synapses are deterministic, but the inputs of the various layers can be set to 0 with some probability η, usually $\eta \leq 0.5$; Dropconnect [91], instead, is a slight variation over the same idea, where some synaptic couplings can be randomly dropped. Notice that both these techniques introduce a source of noise in the training process. It is also easy to see that in a single-layer model the two schemes are indistinguishable.

Let us call $u = (u_i)_{i=1}^N \in \{0,1\}^N$ the dropout mask, and η the probability of dropout, i.e. $u_i \sim \eta\delta(u_i) + (1-\eta)\,\delta(u_i - 1)$. Also suppose, in connection with the fully-stochastic case discussed above, that we generate a new dropout mask at each presentation of a pattern. By applying the CLT we obtain:

$$\begin{aligned}\mathcal{L}\left(\tilde{W},\eta\right) &= \sum_{(x,y)\in\mathcal{D}} \left\langle \Theta\left(\frac{1}{\sqrt{N}}\sum_{i=1}^N x_i^\mu \tilde{W}_i u_i\right)\right\rangle_{u,\eta} \\ &= \sum_{(x,y)\in\mathcal{D}} H\left(-\sqrt{\frac{1-\eta}{\eta}}\frac{1}{\sqrt{N}}\sum_{i=1}^N x_i^\mu \tilde{W}_i\right) \end{aligned} \tag{7.70}$$

Thus, by setting $\eta = 1 - q_\star$, we have a direct mapping between these two kinds of stochastic machines (in expectation and in the limit of large inputs). Again, $\gamma \to \infty$ implies $\eta \to 0$ (i.e., no Dropout) and the problem is reduced to a standard binary Perceptron.

We can now study the free entropy potential in the special case of binary variables and the control parameter ρ of equation 7.69:

$$\phi = \mathbb{E}_\mathcal{D}\,\log\sum_{\tilde{W}} H\left(-\rho\frac{\sum_i x_i^\mu \tilde{W}_i}{\sqrt{N}}\right)^\beta \tag{7.71}$$

The computation is very similar to (but simpler than) the continuous one, except that now the order parameters represent overlaps between the $\tilde{W}$ rather than the magnetizations m, giving after the factorization over the spatial and the pattern indices:

$$e^{n\phi} = \int\prod_{a>b}\frac{dq^{ab}d\hat{q}^{ab}N}{2\pi}\exp\left(-N\sum_{a>b}q^{ab}\hat{q}^{ab}\right)(G_S)^N\,(G_E)^{\alpha N} \tag{7.72}$$

$$\begin{aligned} G_S[\hat{q}] &= \log\sum_{\{\tilde{W}^a\}} e^{\frac{1}{2}\sum_{ab}\hat{q}_{ab}\tilde{W}^a\tilde{W}^b}, \\ G_E[q] &= \log\int\prod_a\frac{\mathrm{d}\hat{\lambda}_a\mathrm{d}\lambda_a}{2\pi}\; e^{-\frac{1}{2}\sum_{ab}q_{ab}\hat{\lambda}_a\hat{\lambda}_b + i\hat{\lambda}_a\lambda_a}\prod_a H^\beta(-\rho\lambda_a) \end{aligned}$$

and in the RS Ansatz one obtains the following expressions:

$$\phi = N \max_{q,\hat{q}} \left\{ \frac{1}{2} q\hat{q} + \log \mathcal{G}_S + \alpha \mathcal{G}_E \right\} \tag{7.73}$$

with the definitions:

$$\mathcal{G}_E = \lim_{n \to 0} \frac{1}{n} \log G_E = \int Dz_0 \log \int Dz_1 H \left(\rho \left(z_1 \sqrt{1-q} + z_0 \sqrt{q} \right) \right)^{\beta} \tag{7.74}$$

$$\mathcal{G}_S = \lim_{n \to 0} \frac{1}{n} \log G_S = -\frac{\hat{q}}{2} + \int Dz \, \log 2 \cosh \left(z \sqrt{\hat{q}} \right) \tag{7.75}$$

We can also compute the average energy of the model, i.e. the log-likelihood, which can simply be obtained by taking the derivative:

$$\mathcal{L} = \frac{\partial \phi}{\partial \beta} = \alpha \int Dz_0 \left\langle \log H \left(\rho \left(z_1 \sqrt{1-q} + z_0 \sqrt{q} \right) \right) \right\rangle_{z_1} \tag{7.76}$$

where we defined an averaging operator $\langle \cdot \rangle_{z_1}$ with a measure proportional to the weight $G(z_1) H \left(\rho \left(z_1 \sqrt{1-q} + z_0 \sqrt{q} \right) \right)^{\beta}$. This allows us to find the entropy of the $\tilde{W}$, via the Legendre transformation: $\Sigma = \phi - \beta \mathcal{L}$. Since the problem is discrete, in this case we can use the usual criterion $\Sigma \geq 0$ to detect when the RS solution is acceptable (cf. with section 4.2).

As in section 7.5, we can also compute (by the usual replica trick) the typical number of errors per pattern made both by the stochastic machine,ϵ_{stoch}, and by the reference configuration,$\epsilon_{\tilde{\text{W}}}$, given respectively by:

$$\epsilon_{\text{stoch}} = \int Dz_0 \left\langle H \left(\rho \left(z_1 \sqrt{1-q} + z_0 \sqrt{q} \right) \right) \right\rangle_{z_1} \tag{7.77}$$

$$\epsilon_{\tilde{\text{W}}} = \int Dz_0 \left\langle \Theta \left(- \left(z_1 \sqrt{1-q} + z_0 \sqrt{q} \right) \right) \right\rangle_{z_1} \tag{7.78}$$

Notice that these formulas imply $\epsilon_{\tilde{W}} \leq \epsilon_{\text{stoch}}$, as expected, and that their difference tends to vanish at $\rho \to \infty$.

Finally, we can also check the the local instability criterion for the RS solution, given by $\alpha\gamma_E\gamma_S > 1$, where:

$$\gamma_E = \frac{1}{(1-q)^2}\int Dz_0 \left(\left\langle z_1^2\right\rangle_{z_1} + \left\langle z_1\right\rangle_{z_1}^2\right)^2 \tag{7.79}$$

$$\gamma_S = \int Dz \left(1 - \tanh^2\left(z\sqrt{\hat{q}}\right)\right)^2 \tag{7.80}$$

Let us define $\beta^\star = \beta^\star(\alpha,\rho)$ such that $\Sigma(\beta^\star) = 0$. This should give an upper bound for the region where the RS solution makes sense. In this region, one finds the following general behavior:

- $\beta^\star$ exists for all p in the SAT region $\alpha \in [0, 0.83]$; beyond that, there is no solution near $p \to 1$.
- $\beta^\star \to \infty$ for both $p \to 0$ (i.e. $\rho \to 0$) and $p \to 1$ (i.e. $\rho \to \infty$), as in the case of continuous magnetizations.
- The solution at $\beta = \beta^\star$ is always locally unstable below a certain p, and stable above it, for all $\alpha \in [0, 0.83]$. However, we are mostly interested in the region near $p \to 1$.
- The errors ϵ_{stoch} and $\epsilon_{\tilde{W}}$ tend to 0 for $p \to 1$, for all $\alpha \in [0, 0.83]$. The derivative at $p \to 1$ seems to always be 0 as well, implying the existence of wide (i.e. extensive in N) good regions. These regions become very small above $\alpha \gtrsim 0.77$. It is unclear if these regions are physical, since we are above the threshold α_U measured in chapter 4.
- In the region $\alpha \in [0, 0.75]$, where the solution to the local-entropy computation exists for all p, the local entropy version consistently has a slightly lower error than the fully-stochastic log-likelihood version. This was expected since the high local entropy is only a byproduct of the stochastic synapses formulation, while it was the main requirement of the large deviation analysis of chapter 4.

In figure 7.4, we show a comparison between numerical measurements and analytic predictions for the average energy. In this case, since the control variables are binary, we resort to a MC algorithm for equilibrating over the log-likelihood at fixed values of the parameter $q_\star$.

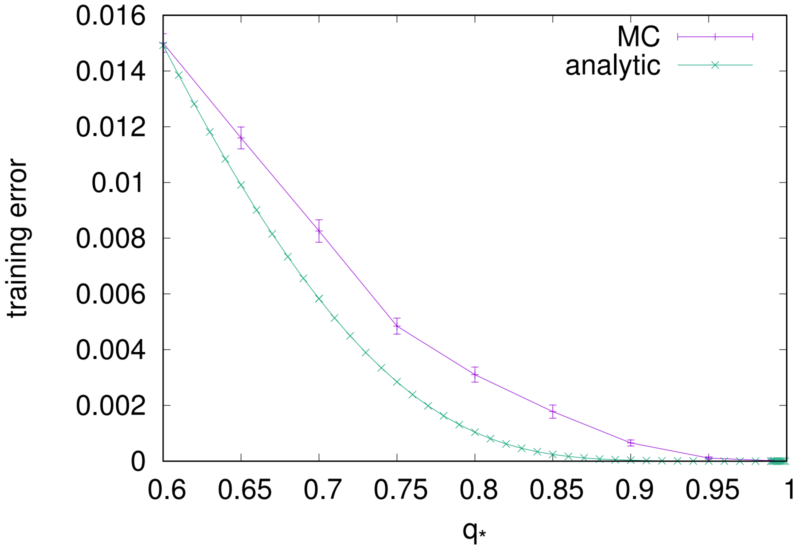


Fig. 7.6 **Binary model**: Energy of the clipped center versus the norm of the control variables q_{*}. Red curve, MC simulation at $N = 1001$, averaged over 100 samples. Green curve, analytical result determined through the replica approach. $\alpha = 0.55$.

7.8 Deep Networks

The learning strategy proposed throughout this chapter, namely going from a learning problem in a binary setting to a stochastic optimization problem over a parametrized probability distribution, can be very effective in training feed-forward ANNs on real data. However, when dealing with K-label classification tasks, in order to define the log-likelihood it is necessary to give a proper definition to the underlying stochastic process, that determines the output of the network and its probability $P\left(y|\, x, W\right)$. Let us first consider the case in which the network outputs a vector $\tau \in \left\{-1, 1\right\}^{K}$, producing an independent binary classification in correspondence of all the possible labels. Consider a single pattern, whose correct label $k^{\star} \in \left\{1, ..., K\right\}$ is specified by the output vector $y^{\mu} = \delta_{k,k^{\star}}$:

1. One could simply ask the network to give an output $\tau = 1$ in correspondence of the correct class, determining a log-likelihood with the shape of

a cross-entropy loss:

$$\mathcal{L}^{\mu}\left(m\right)=\sum_{k=1}^{K}y_{k}^{\mu}\log P\left(\tau_{k}=1\right|x^{\mu},W)=y_{k^{\star}}^{\mu}\log P\left(\tau_{k^{\star}}=1\right|x^{\mu},m) \tag{7.81}$$

2. Alternatively, one could also try to obtain $\tau=-1$ in all the wrong classes, with the log-likelihood:

$$\mathcal{L}^{\mu}\left(m\right)=\sum_{k=1}^{K}y_{k}^{\mu}\log P\left(\tau_{k}=\left(2y_{k}-1\right)\right|x^{\mu},m) \tag{7.82}$$

It is possible, instead, to define the stochastic process differently, so that the loss-function becomes more similar to the softmax function usually employed in Deep Learning. One can consider a stochastic model where the output of the network is accepted only if it is is an indicator on one of the K classes, otherwise the trajectory is rejected and the extraction of the synapses is repeated. The unnormalized probability of obtaining the desired output is still:

$$P\left(k^{\star}|x^{\mu},m\right)=P\left(\tau_{k^{\star}}=1|x^{\mu},m\right)\prod_{k\neq k^{\star}}P\left(\tau_{k}=-1|x^{\mu},m\right) \tag{7.83}$$

However, if we also take into account the normalization we obtain an expression for the likelihood of a single datapoint, wich can be simplified into the form:

$$\mathcal{L}^{\mu}\left(m\right)=\frac{\rho_{k^{\star}}R\left(k^{\star}|x^{\mu},m\right)}{\sum_{k=1}^{K}\rho_{k}R\left(k|x^{\mu},m\right)} \tag{7.84}$$

where $R\left(k|x^{\mu},m\right)=P\left(\tau_{k}=1|x^{\mu},m\right)/P\left(\tau_{k}=-1|x^{\mu},m\right)$, and where we introduced the weights ρ_{k}, with $\rho_{k}=1\ \forall k\neq k^{\star}$, playing the role of a robustness parameter, to be fixed at a value $0<\rho_{k^{\star}}\leq1$. This can encourage a higher probability in correspondence of the correct label.

The drawback, with these definition for the stochastic process, is the fact that the space of possible outputs is exponential, with 2^{K}possible labels. Therefore, the probability of actually obtaining an acceptable output $\tau=2\delta_{k,k^{\star}}-1$ is exponentially small, and for example a MC sampling of $P(y|\,x,W)$, starting from a random configuration of the parameters θ, would have a very high rejection rate $\to 1$.

The other problem concerns the way the probability $P\left(\tau_{k^\star}=1|x,m\right)$ is computed in practice: taking care of the potential correlations between the inputs poses serious technical problems and, on the other hand, even a MC sampling would become unfeasible with growing numbers of hidden layers. Similar to what we did in section 3.4, we can completely neglect the correlations and simply work in a factorized Gaussian approximation (see also [82]), where the standard back-propagation algorithm can be applied. In the simulations, we chose to employ the *natural gradient* (with $(1-m_i^2)\partial_{m_i}$ instead of ∂_{m_i}), with a learning rate equal to 1; the loss-function was set to be that of equation 7.84, with $\rho_{k_*}=0.5$. Moreover, we found to be very important, in practice, to apply the Dropout heuristic, with $\eta=0.25$ in the input layer and $\eta=0.3$ in the intermediate layers. This effectiveness is probably due to two reasons: first, one of the properties of Dropout is that of uncorrelating the hidden units, in accordance with the naive assumption we made in the factorized approximation; secondly, the error function can suffer from the vanishing gradient problem close to saturation [90], in the late stages of the training procedures, and the Dropout can help enhancing the error signal received by the small magnetizations.

On the MNIST digit recognition benchmark [71], we obtained the following generalization performance:

- $\sim 1.3\%$ with a fully connected architecture, with two hidden layers with 801 units each.
- $\sim 1.2\%$ with three hidden layers of size 801.

These results are very promising, given that we are training a network with binary weights and no convolutional layers [7].

Part III

Conclusions

In this PhD thesis, we approached the problem of learning in Artificial Neural Networks with discrete synapses, both from a theoretical and an algorithmical point of view. The relevance of this subject is rapidly escalating in the Deep Learning community, as the impressive success of DNNs, in a variety of complex recognition tasks, is accompanied by growing memory and computational costs, calling for methods of obtaining more compact and robust representations of ANNs. This apparent simplification, going from continuous to discrete synaptic weights, might also be crucial for developing more realistic models of neural computation as well as hardware implementations of ANNs, but encompasses a series of technical and theoretical complications.

The initial theoretical objective of our work was that of tracing back the effectiveness of a few heuristic solvers to the static properties of the loss landscape, and to resolve the clear discrepancy between the equilibrium analytical predictions and the dynamical properties of these learning processes, in the Binary Perceptron. The main mathematical tool we employed is the Replica Trick, borrowed from the Physics of Disordered Systems: the goal was that of obtaining a Large Deviation analysis able to enhance the statistical weight of configurations immersed in dense regions of solutions, since the solutions found by the algorithms exhibited this peculiar feature. The key idea was to introduce a local entropy potential, measuring the number of neighboring solutions, and using it as a modifier of the standard energy-based Boltzmann-Gibbs measure: the dominating effect of the isolated solution was thus canceled out, and a sub-dominant dense ("unfrozen") cluster of solutions was discovered in the loss landscape. This novel structure was also found to break apart and disappear at a certain constraint density, very close to the measured algorithmic threshold [1].

Conceptually, the local-entropy-reweighting formalism can be seen as a generalization of the 1RSB formalism: compared to the ergodicity breaking scheme described by the Parisi Ansatz, in our scheme we keep an additional dependency on a distance parameter, that explicitly introduces a notion of locality in our model, and potentially allows the description of structures where the usual ultra-metric symmetry of the Gibbs states is broken [6].

We extended our analysis also to the case of the Generalized Perceptron, where a discrete set of possible values for the synaptic weights is allowed, and the training set input and output statistics can be biased. The same qualitative picture holds also in this case, and we where able to show that the overall benefit of adding more synaptic states rapidly vanishes, highlighting the relevance of the problem of learning with discrete synapses [3].

Building on the theoretical understanding obtained through these Large Deviation analyses, we developed a series of algorithms that can target the sub-dominant dense regions of solutions explicitly:

- EdMC, a MCMC optimization scheme, was first introduced as a proof of concept: in this simple solver the objective function is the local entropy itself, estimated in the Bethe approximation through Belief Propagation. A simple Simulated Annealing procedure, both in the usual temperature β and in γ, a parameter controlling the radius inside which the local entropy is estimated, can focus the measure on smaller and denser regions, easily providing solutions of the Perceptron. The landscape explored by this solver is much smoother than the roughed energy landscape, and the process is able to avoid the exponentially numerous meta-stable states even in the greedy zero-temperature limit $\beta \to \infty$. To prove the versatility of this strategy, we also applied it to the 4-SAT problem, obtaining good performance also in the hard region [2]. The main bottleneck in further generalizations remains the problem of computing the local entropy efficiently, as the validity of the cavity approximation has to be assessed for each problem at hand and BP may not be applicable.
- In order to avoid the two-level formulation, based on the employment of BP for the local entropy estimate, we proposed an alternative and more general strategy for obtaining solutions immersed in dense clusters: we defined the Robust Ensemble, where the original partition function is replicated and the replicas interact elastically with a central reference. Any optimization strategy (e.g., Simulated Annealing, Stochastic Gradient Descent, Belief Propagation) applied on this system, instead of the original one, is naturally attracted towards regions with high local entropy of low energy configurations. Our simple recipe can be easily adapted to any learning algorithm: one only needs to run a set of processes in parallel and couple them, to drive them towards high local entropy regions [4].

Similar to EdMC, this type of strategy was proven to be quite general, as it was shown to be very effective also in the K-SAT problem.

- Finally, we also showed that the introduction of a source of stochasticity at the level of the synapses can be exploited as a tool for directing the learning process into the dense cluster. The robustness required for learning in this noisy setting can in fact force the network to learn representations that are unaffected by small local perturbations, similarly to what would happen in a dense region of solutions. This stochastic framework naturally induces a Bayesian treatment of the neural network model, where the aim is that of learning a continuous parametrization of a probability distribution over the synaptic states: this allows one to employ a simple gradient descent procedure on these parameters, which would not be directly applicable in the discrete context. We were able to prove analytically that, in the Perceptron architecture, this learning procedure ends up in the same dense sub-dominant states found in the original Large Deviation analysis [5]. Moreover, this procedure can be easily generalized to deeper architectures [7] and different constraint satisfaction problems.

The idea of searching for high local entropy regins seems to be crucial in the generalization context: in the teacher-student scenario the solutions inside the cluster show remarkably smaller generalization errors with respect to the typical isolated solutions. Moreover, also in the numerical tests performed on real-world data (e.g., the handwritten-digit image-recognition benchmark MNIST), we observed that the well-performing learning algorithms invariably end up in dense regions of solutions, and walking away from their core harshly hampers the generalization performance [1]. The intuitive explanation of this property could be the following: the center of these wide, very robust regions can be interpreted as a Bayesian estimator for the whole extensive neighborhood. This is even more naturally understood when the stochastic synapses are considered, where the mode of the probability distribution, a configuration at the core of the dense cluster, is in fact the solution that carries the largest weight in the Bayesian integral.

It is becoming clear that a phenomenon quite similar to the one we first observed in simple discrete ANNs, is also manifesting in the context of complex deep neural network models, currently employed in machine learning applications. In [79], for example, an algorithm mainly developed for obtaining efficient parallelization of the training process, EASGD, also exhibited a nice generalization performance boost and its definition is actually equivalent to a simple SGD procedure in the Robust Ensemble. Moreover, it seems plausible that many effective heuristics, shaped and tuned in order to find solutions that generalize well, actually search for wide flat regions in the loss landscape, which are the transposition of high local entropy regions in the continuous setting.

Some progress towards designing more explicit and interpretable learning heuristics was presented in [74], where, building on our theoretical analysis and on the observation of a correlation between good generalization scores and the presence of wide valleys, the authors designed an algorithm akin to EdMC for deep continuous networks: Entropy-SGD achieves state-of-the-art performance by exploiting the geometric properties of the energy landscape, targeting regions with a high entropy, in this case estimated through a Langevin Dynamics. These findings are in countertrend with respect to the widespread belief that deep networks present multiple equivalent local minima with the same loss. Moreover, Parle, a hybrid algorithm inpired by the RE and EASGD, shows the potential of a parallel approach, where an explicit redirection towards the well-generalizing flat minima is accompanied also by a generous wall-clock time speedup, with infrequent communication requirements between the processes [92]. It might even be possible to exploit this parallel formulation for splitting the dataset instead of sharing it. All in all, these results seem to motivate a fundamental reconsideration of distributed machine learning in non-convex problems, as DNNs.

Another research direction is that of finding a role for the local entropy also in the unsupervised learning scenario, both in attractor neural networks and in generative models like the Restricted Boltzmann Machine. The enhanced robustness to noise might in fact be relevant for modeling and memorizing real data, that is often fuzzy and ambiguous. In this direction, in [8] we propose a new learning rule, Delayed Correlation Matching, that proves that the learning process can be built on highly noisy measurements and very small signals. However, the link with the reweighted measure is not yet formed, and it probably requires a more general rethinking of the true objective of inference processes.

In the attempt of exporting our novel algorithmic strategies to different CSPs and other problems, we still have to investigate a proper way of generalizing the definitions of locality and neighborhoods, since Hamming or Euclidean distances might not be suited for capturing the relevant structures in some cases. Another intriguing problem is the development of a theoretical framework for those special out-of-equilibrium processes that are attracted to accessible states: it might be possible to characterize their stationary state by a large local entropy, even when the system is unable to reach thermodynamic equilibrium due to the underlying stochastic forces.

Finally, it is interesting to note that in [93] the authors proved a connection between Quantum Annealing and the RE in the Perceptron: the proposed intuitive explanation, is that the quantum fluctuations naturally drive the QA optimization process towards wide flat regions, since the system is able to lower its kinetic energy by delocalizing. Indeed, this is one of the few known models where the quantum limit corresponds to the optimal algorithmic setting.

Publications

[1] Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Subdominant Dense Clusters Allow for Simple Learning and High Computational Performance in Neural Networks with Discrete Synapses. *Physical Review Letters*, 115(12):128101, September 2015.

[2] Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Local entropy as a measure for sampling solutions in constraint satisfaction problems. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(2):P023301, February 2016.

[3] Carlo Baldassi, Federica Gerace, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Learning may need only a few bits of synaptic precision. *Phys. Rev. E*, 93:052313, May 2016.

[4] Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes. *Proceedings of the National Academy of Sciences*, 113(48):E7655–E7662, 2016.

[5] C. Baldassi, F. Gerace, H. J. Kappen, C. Lucibello, L. Saglietti, E. Tartaglione, and R. Zecchina. On the role of synaptic stochasticity in training low-precision neural networks. *ArXiv e-prints*, October 2017.

[6] Carlo Baldassi, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. *In preparation.*

[7] Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Carlo Lucibello, Luca Saglietti, Enzo Tartaglione, and Riccardo Zecchina. *In preparation.*

[8] Carlo Baldassi, Federica Gerace, Alessandro Ingrosso, Luca Saglietti, and Riccardo Zecchina. From pseudo-likelihood to a differential learning rule for stochastic neural networks. *In preparation.*

References

[9] Donald Olding Hebb. *The organization of behavior: A neuropsychological approach.* John Wiley & Sons, 1949.

[10] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[13] Daniel H. O'Connor, Gayle M. Wittenberg, and Samuel S.-H. Wang. Graded bidirectional synaptic plasticity is composed of switch-like unitary events. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9679–9684, 2005.

[14] Thomas M Bartol, Cailey Bromer, Justin P Kinney, Michael A Chirillo, Jennifer N Bourne, Kristen M Harris, and Terrence J Sejnowski. Hippocampal spine head sizes are highly precise. *bioRxiv*, 2015.

[15] Andreas Engel and Christian Van den Broeck. *Statistical mechanics of learning.* Cambridge University Press, 2001.

[16] Edoardo Amaldi. On the complexity of training perceptrons. *Kohonen et al*, pages 55–60, 1991.

[17] Werner Krauth and Marc Mézard. Storage capacity of memory networks with binary couplings. *J. Phys. France*, 50:3057–3066, 1989.

[18] Haim Sompolinsky, Naftali Tishby, and H. Sebastian Seung. Learning from examples in large neural networks. *Physical Review Letters*, 65(13):1683, 1990.

[19] Haiping Huang, K. Y. Michael Wong, and Yoshiyuki Kabashima. Entropy landscape of solutions in the binary perceptron problem. *Journal of Physics A: Mathematical and Theoretical*, 46(37):375002, 2013.

[20] Tomoyuki Obuchi and Yoshiyuki Kabashima. Weight space structure and analysis using a finite replica number in the ising perceptron. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(12):P12014, 2009.

[21] Heinz Horner. Dynamics of learning for the binary perceptron problem. *Zeitschrift für Physik B Condensed Matter*, 86(2):291–308, 1992.

[22] Haiping Huang and Yoshiyuki Kabashima. Origin of the computational hardness for learning with binary synapses. *Physical Review E*, 90(5):052813, 2014.

[23] Olivier C Martin, Rémi Monasson, and Riccardo Zecchina. Statistical mechanics methods and phase transitions in optimization problems. *Theoretical computer science*, 265(1):3–67, 2001.

[24] Marc Mézard and Andrea Montanari. *Information, Physics, and Computation.* Oxford University Press, January 2009.

[25] Cristopher Moore and Stephan Mertens. *The nature of computation.* Oxford University Press, 2011.

[26] Alfredo Braunstein and Riccardo Zecchina. Learning by message-passing in neural networks with material synapses. *Phys. Rev. Lett.*, 96:030201, 2006.

[27] Carlo Baldassi, Alfredo Braunstein, Nicolas Brunel, and Riccardo Zecchina. Efficient supervised learning in networks with binary synapses. *Proceedings of the National Academy of Sciences*, 104(26):11079–11084, jun 2007.

[28] Carlo Baldassi. Generalization Learning in a Perceptron with Binary Synapses. *Journal of Statistical Physics*, 136(5):902–916, sep 2009.

[29] Carlo Baldassi and Alfredo Braunstein. A max-sum algorithm for training discrete neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(8):P08008, 2015.

[30] Luca Dall'Asta, Abolfazl Ramezanpour, and Riccardo Zecchina. Entropy landscape and non-gibbs solutions in constraint satisfaction problems. *Physical Review E*, 77(3):031118, 2008.

[31] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

[32] Florent Krzakala, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborova. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proceedings of the National Academy of Sciences*, 104(25):10318–10323, 2007.

[33] S F Edwards and P W Anderson. Theory of spin glasses. ii. *Journal of Physics F: Metal Physics*, 6(10):1927, 1976.

[34] David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. 35:1792+, 12 1975.

[35] Giorgio Parisi, Marc Mézard, and Miguel Angel Virasoro. *Spin glass theory and beyond*. World Scientific Singapore, 1987.

[36] Marc Mézard and Giorgio Parisi. Replicas and optimization. *Journal de Physique Lettres*, 46(17):771–778, 1985.

[37] Scott Kirkpatrick, Mario P Vecchi, et al. Optimization by simmulated annealing. *science*, 220(4598):671–680, 1983.

[38] Raffaele Marino, Giorgio Parisi, and Federico Ricci-Tersenghi. The backtracking survey propagation algorithm for solving random k-sat problems. 7, 08 2015.

[39] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[40] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[41] Jiquan Ngiam, Adam Coates, Ahbik Lahiri, Bobby Prochnow, Quoc V Le, and Andrew Y Ng. On optimization methods for deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 265–272, 2011.

[42] Patrick Charbonneau, Jorge Kurchan, Giorgio Parisi, Pierfrancesco Urbani, and Francesco Zamponi. Fractal free energy landscapes in structural glasses. *Nature communications*, 5, 2014.

[43] Federico Ricci-Tersenghi and Guilhem Semerjian. On the cavity method for decimated random constraint satisfaction problems and the analysis of belief propagation guided decimation algorithms. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(09):P09001, 2009.

[44] Paul C Bressloff. *Stochastic processes in cell biology*, volume 41. Springer, 2014.

[45] David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[46] Anthony Holtmaat and Karel Svoboda. Experience-dependent structural synaptic plasticity in the mammalian brain. *Nature Reviews Neuroscience*, 10(9):647–658, 2009.

[47] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec 1943.

[48] C. Van Der Malsburg. *Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, pages 245–248. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.

[49] Elizabeth Gardner and Bernard Derrida. Optimal storage properties of neural network models, 1988.

[50] Elizabeth Gardner. The space of interactions in neural network models, 1988.

[51] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 1965.

[52] Dmitri B Strukov, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. The missing memristor found. *nature*, 453(7191):80–83, 2008.

[53] Andy Thomas. Memristor-based neural networks. *Journal of Physics D: Applied Physics*, 46(9):093001, 2013.

[54] Christos Louizos, Karen Ullrich, and Max Welling. Bayesian compression for deep learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3289–3299. Curran Associates, Inc., 2017.

[55] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[56] Elizabeth Gardner and Bernard Derrida. Three unfinished works on the optimal storage capacity of networks. *J. Phys. A: Math. Gen.*, 22:1983–1996, 1989.

[57] Nicolas Brunel, Vincent Hakim, Philippe Isope, Jean-Pierre Nadal, and Boris Barbour. Optimal Information Storage and the Distribution of Synaptic Weights. *Neuron*, 43(5):745–757, sep 2004.

[58] Hanoch Gutfreund and Yaakov Stein. Capacity of neural networks with discrete synaptic couplings. *Journal of Physics A: Mathematical and General*, 23(12):2613, 1990.

[59] Silvio Franz and Giorgio Parisi. Recipes for metastable states in spin glasses. *Journal de Physique I*, 5(11):1401–1415, 1995.

[60] Robert Hecht-Nielsen et al. Theory of the backpropagation neural network. *Neural Networks*, 1(Supplement-1):445–448, 1988.

[61] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[62] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

[63] F. Rosenblatt. *Principles of Neurodynamics.* Spartan Book, 1962.

[64] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry.* MIT Press, Cambridge, MA, USA, 1969.

[65] D J Amit, C Campbell, and K Y M Wong. The interaction space of neural networks with sign-constrained synapses. *Journal of Physics A: Mathematical and General*, 22(21):4687, 1989.

[66] Lenka Zdeborová and Marc Mézard. Constraint satisfaction problems with isolated solutions are hard. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(12):P12004, oct 2008.

[67] David JC MacKay. *Information theory, inference and learning algorithms.* Cambridge university press, 2003.

[68] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312, 2005.

[69] Andrea Montanari, Federico Ricci-Tersenghi, and Guilhem Semerjian. Solving constraint satisfaction problems through belief propagation-guided decimation. *arXiv preprint arXiv:0709.1667*, 2007.

[70] Marc Bailly-Bechet, Christian Borgs, Alfredo Braunstein, J Chayes, A Dagkessamanskaia, J-M François, and Riccardo Zecchina. Finding undetected protein associations in cell signaling by belief propagation. *Proceedings of the National Academy of Sciences*, 108(2):882–887, 2011.

[71] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[72] Giorgio Parisi. On local equilibrium equations for clustering states. *arXiv preprint cs/0212047*, 2002.

[73] Olivier Rivoire. Properties of Atypical Graphs from Negative Complexities. *Journal of Statistical Physics*, 117(3-4):453–476, November 2004.

[74] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *CoRR*, abs/1611.01838, 2016.

[75] Levent Sagun, Léon Bottou, and Yann LeCun. Singularity of the Hessian in Deep Learning. *ArXiv e-prints*, pages 1–6, nov 2016.

[76] Rémi Monasson. Structural Glass Transition and the Entropy of the Metastable States. *Physical Review Letters*, 75(15):2847–2850, October 1995.

[77] Carlo Baldassi. A method to reduce the rejection rate in monte carlo markov chains. *Journal of Statistical Mechanics: Theory and Experiment*, 2017(3):033301, 2017.

[78] Matthieu Courbariaux and Yoshua Bengio. Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv preprint arXiv:1602.02830*, 2016.

[79] Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with elastic averaging sgd. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 685–693. Curran Associates, Inc., 2015.

[80] Sixin Zhang. Distributed stochastic optimization for deep learning (thesis). *arXiv preprint arXiv:1605.02216*, 2016.

[81] H Sebastian Seung. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, 40(6):1063–1073, 2003.

[82] José Miguel Hernández-Lobato and Ryan P Adams. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. *Journal of Machine Learning Research*, 37:1–6, feb 2015.

[83] Daniel Soudry, Itay Hubara, and R Meir. Expectation Backpropagation: parameter-free training of multilayer neural networks with real and discrete weights. *Neural Information Processing Systems 2014*, 2(1):1–9, 2014.

[84] Alex Graves. Practical Variational Inference for Neural Networks. *Nips*, pages 1–9, 2011.

[85] Oran Shayar, Dan Levi, and Ethan Fetaya. Learning discrete weights using the local reparameterization trick. *arXiv preprint arXiv:1710.07739*, 2017.

[86] Shun-ichi Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276, feb 1998.

[87] Manfred Opper and Ole Winther. A Bayesian approach to on-line learning. In David Saad, editor, *On-line learning in neural networks*, pages 363–378. 1998.

[88] Thomas P Minka. Expectation Propagation for Approximate Bayesian Inference F d. *Uncertainty in Artificial Intelligence (UAI)*, 17(2):362–369, 2001.

[89] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[90] Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Master's thesis, Institut fur Informatik, Technische Universitat, Munchen*, 1991.

[91] Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1058–1066, 2013.

[92] Pratik Chaudhari, Carlo Baldassi, Riccardo Zecchina, Stefano Soatto, and Ameet Talwalkar. Parle: parallelizing stochastic gradient descent. *arXiv preprint arXiv:1707.00424*, 2017.

[93] Carlo Baldassi and Riccardo Zecchina. Efficiency of quantum versus classical annealing in non-convex learning problems. *arXiv preprint arXiv:1706.08470*, 2017.