UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

CORSO DI LAUREA MAGISTRALE IN FISICA

OUT-OF-EQUILIBRIUM ANALYSIS OF SIMPLE NEURAL NETWORKS

Relatore interno:
Prof. Sergio Caracciolo
Correlatore:
Prof. Riccardo Zecchina
Correlatore:
Dott. Carlo Lucibello

Ivan Amelio
863954

Codice PACS: 05.90.+m

Anno Accademico 2015-2016

July 5, 2016

Abstract

We consider a novel approach to learning in neural networks with discrete synapses [1, 2, 3] and discuss its possible extensions to simple continuous neural networks. The problem of learning is explained, in a general setting and from the statistical mechanics standpoint. The recent achivements in the training of discrete neural networks are reviewed: the statistical properties of solutions found by efficient algorithms are described by a non-equilibrium measure; conversely, this measure suggest new ways to design efficient algorithms. In the original part of the work we consider the simplest non-trivial model of continuous neural network: the perceptron with negative stability. We extend the Franz-Parisi equilibrium analysis and investigate some off-equilibrium features, both analytically and with simulations. The results show that the model is not a complex system and its dynamical behaviour differ drastically from both the discrete case and deeper architectures. Future perspectives are discussed.

Contents

Introduction

In the second half of last century physicists started to apply methods developed in fields like quantum field theory and condensed matter to the study of interdisciplinary topics, such as computer science, stochastic processes, biophysics, inference, economics and social sciences. The link between so different areas of science is provided by *probabilistic graphical models*: the interactions of spins in a magnetic material, the constraints of a boolean problem, the percolation of fluids through porous materials, the contacts of amminoacids in a DNA sequence and the network of friends on Facebook can be modelled in a simple way assigning to each degree of freedom a vertex and to each interaction a hyperedge in a graph. The Ising model in its thousands of variations is the purest example. One of the most important ideas underlying this approach is *universality*: when many degrees of freedom interact, many observable properties of the system are independent of the details of the interaction law, but, near second order phase transitions, only general features as symmetries and range of the interaction matter. In the sixties and in the seventies quantum field theories began to be set on lattices as a regularization scheme and the renormalization group (Wilson 1974) provided a unified formal description of the Higgs mechanism and critical phenomena. After these big achivements of equilibrium thermodynamics, the interest of the statistical physics community turned towards *non-equilibrium*, that is a main feature of the great majority of real life systems.

In 1975 Edwards and Anderson [4] proposed to describe the interactions between spins in a quenched magnetic material with an Ising model with random (*disordered*) couplings: this was the beginning of *spin glass theory[5]*. Optimization algorithms based on local minimization of a cost function share with glasses to get trapped in metastable states, so since 1985 methods from spin-glasses were widely applied to predict the statistical properties of many optimization-like problems: matching[6], learning in neural networks [7, 8]and K-satisfiability [9] are the emblem of the success of this approach. Another versatile tool developed by the physics community is the cavity method and its many implementations [10, 11, 12, 13, 9]. Such an extended application of statistical physics to optimization and computer science is conveniently recast within the formulation and notation of *information theory* [10].

At the same time, the devolopement of computers drove the progess in Artificial Intelligence and Machine Learning; in particular, neural networks are now a standard statistical analysis tool in many branches of engineering and science and in recent years big achivements have been obtained with the use of many-layer architectures (*deep learning*) [14, 15]. However, while the spin glass approach has provided theoretical results concerning the thermodynamics of few-layer neural networks [7, 16, 8, 17, 18, 19, 20], the dynamics of learning is a highly nonlinear and complex one and a big gap between theory and practice has to be filled.

In this work we consider a novel approach [1, 2, 3] that describes non-equilibrium features of efficient learning algorithms for neural networks with discrete synapses. The dynamics and performances of training algorithms are shown to be strictly related to the existence of dense clusters of solutions, similarly to the scenario in K-SAT [21]. Moreover, a simple and very flexible strategy to design new algorithms and the many perspectives it opens are presented. The original part of the work consists in discussing possible extensions to continuous neural networks.

1 Basic concepts and tools

In this Chapter we introduce some definitions and methods which will occur many times in what follows.

The protagonist of this work is the perceptron: this is the building block of every neural network and will be introduced in Section 1.1, in its discrete and continuous versions. We will define what learning means in the context of artificial neural networks and will mention some examples of training algorithms.

Then, we will address the question of the amount of information which can be stored in a network. This question can be answered within the Gardner analysis, which makes use of methods from spin-glass theory, namely the replica trick. To this end, we provide the sketch of solution for the continuous perceptron storage problem in the replica symmetric ansatz in Sec. 1.2.

Finally, in Sec. 1.3 we introduce probabilistic graphical models and discuss a useful tool for estimating marginals and thermodynamical properties in particular classes of graphs: Belief Propagation, a class of approximated message-passing algorithms.

In this Chapter we mainly refer to the textbooks [8, 5, 10].

1.1 Simple models of neural networks

Artificial neural networks were originally introduced as mathematical models of intelligence. There is a main difference between the way brain processes information and the way the CPU of a computer does: the CPU works in a serial manner, receiving one input at a time, performing the task, and so on. A neural network (NN) instead receives many inputs and treats them in a distributed way. A NN is a very complex dynamical system, which is usually studied with methods from statistics. Artificial NNs are used in a variety of applications, ranging from function approximation and data analysis to neuroscience modelling and robotics.

1.1.1 The perceptron

In 1957 Frank Rosenblatt proposed a simple model of artificial intelligence, the *perceptron*, consisting of several input "neurons" that can be active or inactive and connected to an output unit by means of edges called "synapses". We represent the state of the neurons with the vector x, $x_i = \pm 1$, while for the synaptic weights we reserve the notation W. For a given input, the output is chosen as

$$output(x) = sign(x \cdot W)$$

We have not specified the nature of the synapses yet. Two possibilities are considered:

- *continuous perceptron* with spherical constraint $W_i \in R,\ W^2 = N$. Up to now it has been studied more than the discrete one due to the importance

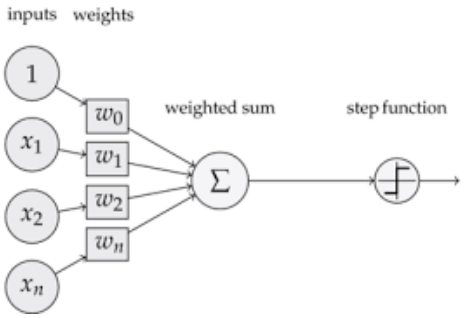

Figure 1: The perceptron, or single layer neural network. Once the synapses are trained, an output is given as the step function of the input weighted sum.

of continuous neural networks in computer science, as training continuous neural networks is easier thanks to derivative based srategies, see below.

- *discrete perceptron*, if $W_i = \pm 1$. A possible generalization of such "Ising" perceptron is the "Potts" perceptron $W_i = 0, 1, .., q$, as in [22]. The brain synapses are thought [23] to switch between a finite number of states, so discrete neural networks are of biological interest. Recently there is a renewed interest in binary computer implementation of multilayer networks [24], as we will mention later.

1.1.2 Statistical learning

How to choose the synaptic weights? The basic idea behind *learning* is that the network is stimulated with M inputs that we denote with ξ^μ $\mu = 1, .., M$ and we want the corresponding outputs to be σ^μ. The array $\{\xi^\mu, \sigma^\mu\}$ is called *training set.*

Several ways of choosing the training set can be conceived. For example in applications the input data (e.g. $L \times L$ pixels images from some dataset) are always correlated (and the very ultimate goal of machine learning, i.e. unsupervised learning, consists exactly in discovering these correlations or *features*, see Fig. 2). In the simplest scenario one chooses the patterns randomly or only with a magnetization [7]. In what follows we will consider only the unbiased case of ξ_i extracted ± 1 with uniform probability, and consider two schemes for choosing the requested outputs σ^μ:

- the *classification problem* consists in taking $\sigma^\mu = \pm 1$ randomly. Actually, it is convenient to require simply $\sigma^\mu = +1$ and reabsorbe the minus sign in the pattern ξ^μ. More in general, the idea behind classification is that the network can be trained to label data, where the data and their labels are somehow recorded in the synaptic weights (there will be a maximum *storage capacity* and in principle the issue of generalization, see below, is not needed)

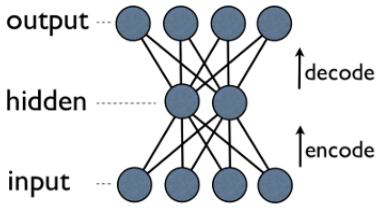

Figure 2: An autoencoder, that is a network whose goal is learning in an unsupervised way the identity function (output = input). The non triviality comes from the fact that the middle layer (*hidden layer*) has less bits than the input: to overcome this bottleneck the autoencoder has to learn a compressed representation of the input. This is possible if the input data are correlated and the dimensionality reduction is equivalent to discovering and exploit efficiently these correlations.

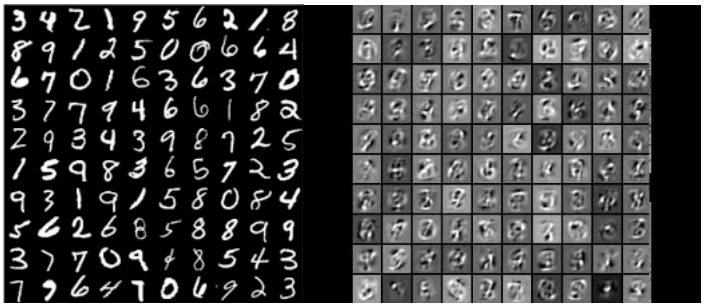

Figure 3: Right panel: a sample from the MNIST database, a training set of 60000 handwritten digits plus a test set of 10000. Left panel: an example of features learnt by the hidden units of an autoencoder (here used for pre-training purposes). The feature corresponding to a hidden unit is defined as the input pattern which maximizes the "activation" of the hidden unit.

- the *generalization or teacher-student problem,* instead, addresses the ultimate goal of inferring an unknown function from the values taken by the function itself over a sample set. In the perceptron this amounts to fixing a synaptic vector W^T to be inferred (this is called the *teacher*) and considering the training set $\{\xi^\mu, \sigma^\mu = sign(W^T \cdot \xi^\mu)\}$. The term generalization means that the *student* network should be able to give the same output as the teacher even on those inputs that are not part of the training set.

Moreover, for a given training set, there exist several ways of choosing which of the solutions of the training set retain. For example, one can pick randomly a synaptic configuration that satisfies the training set, i.e. we are weighting the solution space with the uniform measure: this is called *Gibbs learning*. But in the teacher-student scenario Gibbs learning is not the best choice for generalization. It is much more convenient minimizing the conditional likelihood of the training

set with respect to the psuedo-teacher which one wants to infer. This approach is referred to as *Bayesian learning*. In the continuous case this amounts to picking the center of mass of the solutions of the training.

Finally, from a more practical standpoint, one can conceive different training algorithms to find a solution. One important feature of such an algorithm is if it is *on-line* or *off-line*: the human brain is expected to be on-line, meaning that each time we are stimulated with a new idea or task we can learn this without "forgetting" all other abilities. Given a new training pattern, one wants a rule for updating the synaptic weights without the need to reconsider all the training set.

Here we report some examples of training algorithms:

- the celebrated *Hebb rule* (1949) is based on the biologically derived idea: "Cells [ndr, neurons] that fire together, wire together." With the above arrangement that $\sigma^\mu = +1,\ \forall \mu$ the original Hebb update rule is:

$$W_i^{t+1} = W_i^t + \xi_i^\mu \theta(\xi_i^\mu)$$

  where θ is the Heavside function.

- a quite general approach consists in defining a *loss* function or *energy* E that estimates the amount of error made in classifing the patterns of the training set, and trying to minimize this error with some derivative-based method. The first order method referred to as gradient descent (GD) consists in following at each step the direction of the (opposite) gradient of the loss function:

$$W^{t+1} = W^t - \eta^t \nabla E(W^t) \tag{1}$$

  where t counts the number of iterations and η is called *learning rate* and should be adjusted in a proper way. In the continuous perceptron case a possible choice is:

$$E(W) = \sum_\mu (-\sigma^\mu \frac{W \cdot \xi^\mu}{\sqrt{N}})^r \theta(-\sigma^\mu \frac{W \cdot \xi^\mu}{\sqrt{N}}) \tag{2}$$

  where r is an exponent (usually 1 or 2) and a useful normalization has been inserted. So this energy counts the mistaken patterns with a weight. For multilayer feedforward networks the gradient at node l can be computed as[1] $\frac{\partial E}{\partial \vec{W}_l} = \frac{\partial E}{\partial \sigma_l} \frac{\partial \sigma_l}{\partial \vec{W}_l}$, $\frac{\partial E}{\partial \sigma_l} = \sum_k \frac{\partial E}{\partial \sigma_k} \frac{\partial \sigma_k}{\partial \sigma_l}$, where σ_l is the ouput[2] of node l, $\frac{\partial \sigma_l}{\partial \vec{W}_l}$ is the variation of the ouput at node l relative to variations of its feeding synapses, and the sum is performed only over the nodes of the network that are in the immediately higher layer and that receive σ_l in

[1] we have dropped the μ index, but it is meant that this is the contribution of each pattern; the overall gradient is obtained summing over all the training set.

[2] for this purpose one usually defines the output at each node by means of a continuous and differentiable sigmoid function

input. Due to this observation the right way to compute the gradient is starting from higher (i.e. nearer to the output) layers and step back to the input: in this context GD is often referred to as *backpropagation*.

- *stochastic GD* (SGD): at each iteration one computes the gradient with respect to only a randomly exctracted subset of the training set. The size of this subset is called *batch-size*. Online learning corresponds to batch-size equal to 1. The main motivation behind SGD is that considering the whole batch at each iteration is too computationally expensive, while SGD in practice works well. There are other important and open issues about the use of SGD instead of GD: it seems that the noise in SGD enables the algorithm to avoid poor local minima and select minima that have both minor training and generalization error [25].

- Setting $r = 1$ in eq (2) and considering one pattern per iteration, the update (1) becomes:

$$W^{t+1} = W^t, \quad if\, \sigma^\mu = sign(\frac{W \cdot \xi^\mu}{\sqrt{N}})$$

$$W^{t+1} = W^t + \sigma^\mu \frac{\xi^\mu}{\sqrt{N}}, \quad otherwise$$

  This is referred to as the *perceptron rule*.

For a training algorithm it often useful to consider the (average) training and generalization error as a function of α and, for fixed α, of training time (1 *epoch* = 1 span of the training set).

1.1.3 Negative stability and geometrical interpretation

When training it can be convenient to require a condition stronger than merely producing the correct output. For example consider the pattern ξ^μ and, to fix the ideas, suppose you want the corresponding output to be $+1$. It would be nice if the overall scheme would be "stable", i.e. when receiving the input $\xi^\mu + \epsilon$, with ϵ a "small" correction which may also be due to a noisy data capture, the output would remain -1.

It is possible to address this problem by requiring that on the training set:

$$\sigma^\mu \frac{W \cdot \xi^\mu}{\sqrt{N}} > \kappa$$

The constant κ is called *stability* or *threshold*. Here it is clear the choice of normalization: if a finite fraction of input spins are randomly flipped by noise, the law of large numbers states that their weighted sum is $O(\sqrt{N})$.

This approach is practically useful only if $\kappa \geq 0$. Nonetheless nothing forbids to set $\kappa < 0$, with some major differences in the continuous case:

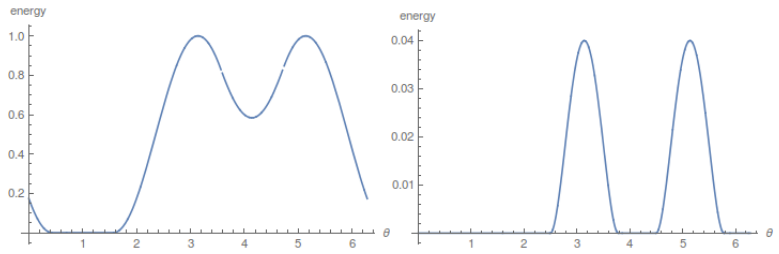

Figure 4: Training energy lanscape in the spherically constrained continuous perceptron, with $N = 2$ and $M = 2$. On the horizontal axis the angle θ parametrizes the circumference. The left panel shows the positive stability scenario, the optimization problem is not convex but the set of solution is. On the right a negative perceptron sample, notice that the space of solution is disconnected.

- there is a nice geometrical interpretation of what satisfying a pattern does mean (here we explicitly consider the spherical constraint a when speaking of "sphere" we actually refer to the surface of the sphere) :
  given a pattern ξ^μ and supposing, to fix the ideas, that $\sigma^\mu = +1$, this select in the $\kappa = 0$ case an emisphere, for $\kappa > 0$ a (convex[3]) spherical cap, while in the negative stability case a spherical cap is forbidden. So the space of solution is the intersection of convex sets and so it is connected and convex in the $\kappa \geq 0$ case; it is not so for negative stability, and at least in principle we expect to be fragmented in several different disconnected domains.

- if $\kappa \geq 0$ and neglecting the spherical constraint the problem of training is convex: one can choose a loss function (2) with $r \geq 1$ and notice that this is the sum of (non strictly) convex function so it is convex. The spherical constraint is not strictly neccessary for $\kappa > 0$, because $W = 0$ is not a solution and scaling a solutions of a factor >1 yields a new solution; on the other end the same scaling on a configuration that is on a "bad" direction increases the loss function, so that the norm of W is controlled[4]. For $\kappa < 0$ the error function is still convex without the spherical constraint, but the point is that in a neighbourhood of $W = 0$ there are trivial solutions of little norm and this region is an attractor: from each direction the loss function decreases with the norm. The result is that the (in)stability of the input-output reflects itself also in the (in)stability of training. However we remark that, independently of the stability, with the spherical constraint

[3] A subset of the surface of the sphere is convex if the sector of the sphere subtended by it is convex according to the usual definition.

[4] this does not mean that the two problems are equivalent, in fact the spherical one is not convex. We mean that in principle one could train the network with a positive stability and not controll the norm, and obtain solutions acceptable at least for the $\kappa = 0$ training (where only the direction matters). However it is clear that it is meaningful to speak of the magnitude of κ only in relation to the norm of W.

the problem is *not* convex for any choice of the training set, as Fig. 4 shows.

The "negative perceptron" then is not very interesting under a practical point of view, but it is a less trivial and richer model. Actually in [26] the authors show that there is an analogy between the negative perceptron and the problem of packing hard spheres. One can think of adding a hard sphere on a spherical surface already containing other spheres in this way: the existing spheres are points on the surface while the new sphere has a finite radius depending on κ, i.e. it requires a free spherical cap. Franz and Parisi have shown that this model at the SAT-UNSAT transition (see below) belongs to the same universality class of mean-field hard spheres models at the jamming transition.

1.2 Gardner analysis

In this Section we show the statistical mechanics approach to the problem of learning. In 1.2.1 we specify the concept of typical volume of the space of solutions, while in 1.2.2 we outline the computation of this volume by means of the replica trick (Elizabeth Gardner [7, 16]), for the continuous perceptron in the generalization scenario. We posticipate the details of the computation to Section 3, but we present already here the phase diagram of the classification problem and the storage capacity.

1.2.1 Self-averaging quantities

We consider the training problem within the generalization setup, for the continuous perceptron. How many patterns is the network able to learn? Given a training set, the volume of solutions is

$$\Omega(\{\xi^\mu, \sigma^\mu\}) = \int d\mu(W) \prod_\mu^M \theta(\sigma^\mu \frac{W \cdot \xi^\mu}{\sqrt{N}} - \kappa) \tag{3}$$

where recall that $M = \alpha N$ and the measure of integration is given in the great N limit by

$$d\mu(W) = (2\pi e)^{-N/2} \delta(W^2 - N) dW$$

We want the "average" volume of solutions, in some sense, where the average is over the training set extracted in a random unbiased way. The temptation is to consider the average of the volume (3) : the problem reduces to computing the product of the averages of the θ's, so it is technically easy. This approach is called *annealed approximation*.

Despite the charm of this easiness, this computation does not give the information we really want. What we want is: we generate randomly a sample and measure the volume of solutions and repeat many times. We expect that, if N is very large, in the striking majority of cases we find nearly the same volume. Mathematically, we expect that increasing N the distribution of volumes concentrates in probability around a typical value Ω_{typ}. This behaviour has been

called *self-averageness* with relation to magnetic systems, in which one expects that the extensive quantities of a (large) portion of the system generalize to the whole system or to other large portions by simply scaling N.

The volume computed within the annealed approximation does not capture this concept. Essentially, the problem is that Ω is not an extensive quantity, but a superextensive one. With an abuse of notation and neglecting the normalization constants, we can formally write:

$$\Omega \sim e^{N\omega}, \quad p(\omega) \sim e^{N\, s(\omega)}$$

where ω and s are intensive N-independent quantities. The typical volume is given by ω_{typ} where ω_{typ} maximizes s. The annealed average volume is given by saddle point:

$$\langle\Omega\rangle \sim \int d\omega\, e^{N\omega + N\, s(\omega)} \sim e^{N\, \max_\omega\{\omega + s(\omega)\}} \neq e^{N\, \arg\max_\omega\, s(\omega)} = \Omega_{typ}$$

so in the annealed approximations the relevant volume is determined from both the number and volume of the samples.

The right quantity to be averaged is the extensive and self-average:

$$S_{typ} = \langle log\Omega\rangle \tag{4}$$

Such an average is called *quenched*.

1.2.2 Replica computation

The entropy (4) can be computed with a well-known trick in the theory of spin-glasses [4] :

$$log\Omega = \lim_{n\to 0} \frac{\Omega^n - 1}{n}$$

hence:

$$\langle log\Omega\rangle = \lim_{n\to 0} \frac{\langle\Omega^n\rangle - 1}{n}$$

One can think of n integer so that this reduces to computing the average of n independent systems with the same disorder:

$$\langle\Omega^n\rangle = \Omega(\{\xi^\mu, \sigma^\mu\}) = \left\langle \int d\mu(W_1)...d\mu(W_n) \prod_{\mu,a} \theta(\sigma^\mu \frac{W_a\cdot\xi^\mu}{\sqrt{N}} - \kappa) \right\rangle_{\{\xi,\sigma\}}$$

where the a's label the different *replicas* of the system . Then we will extend the result found by analytical continuation for every $n \in R$, and suppose this is really equal to $\langle\Omega^n\rangle$.

Introducing the order parameters

$$Q_{ab} = \frac{W_a\cdot W_b}{N}$$

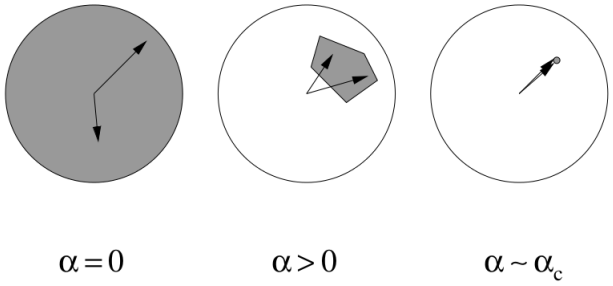

Figure 5: Sketch of the shrinking of the space of solutions when increasing the training set size.

the above expression can be decoupled both in the i and μ indices and the average performed explicitly, so to get something in the form:

$$\langle log\Omega \rangle = \lim_{n\to 0} \frac{\int dQ_{ab}\, e^{N\, s(Q_{ab})} - 1}{n} \tag{5}$$

At this point $\int dQ_{ab}\, e^{N\, s(Q_{ab})}$ is computed by the saddle point method. A subtetly here is that the limit $N \to \infty$ has been performed before $n \to 0$. Practically, one assumes for symmetry reasons that the saddle point occures for a matrix $Q_{ab} = \widetilde{q}$ and maximizes[5] with respect to $\widetilde{q}$: this is referred to as Replica Symmeric (RS) ansatz. The physical interpretation of $\tilde{q}$ is that of typical overlap between solutions: in the great majority of samples the great majority of solutions has got overlap $\tilde{q}$, according to this analysis.

Only at this point the limit $n \to 0$ is taken, yielding something like:

$$\frac{1}{N}\langle log\Omega \rangle = \frac{1}{N} \lim_{n\to 0} \frac{e^{nN\, extr_{\tilde{q}} s(\widetilde{q}) + O(n^2)} - 1}{n} = extr_{\widetilde{q}} s(\widetilde{q})$$

An accurate calculation yields:

$$s(\alpha) = extr_{\tilde{q}}\{\frac{1}{2}\log(1-\tilde{q}) + \frac{1}{2}\frac{\tilde{q}}{1-\tilde{q}} + \alpha \int Dz\ \log H(\frac{\kappa - \sqrt{\tilde{q}}z}{\sqrt{1-\tilde{q}}})\}$$

where here and in what follows:

$$H(x) = \int_x^{+\infty} Dz, \quad Dz = \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}}$$

We expect the overlap to be 0 at $\alpha = 0$ and then to grow until a critical value of α in which the space of solutions shrinks to a point and therefore $q \to 1$. Expanding the saddle point equation in this limit (we will show later the details

[5] actually as $n \to 0$ some subtlety is involved [5]

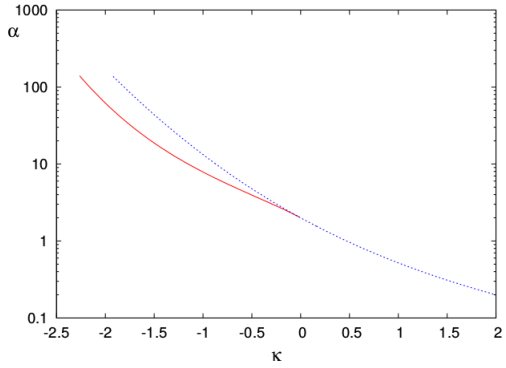

Figure 6: The phase diagram of the continuous perceptron generalization problem is plotted in the stability-training set size plane. The graphic is taken from [26]. The blue dotted line marks the SAT-UNSAT phase transion, as computed in the RS ansazt: increasing κ makes more diffucult to satisfy the training constraint, notice the logarithmic scale of the y-axis. In red the de Almeida-Thouless line (AT line) for negative stability. For positive stability the AT line coincides with the SAT-UNSAT one [27] . For $\kappa = 0$, $\alpha_c = 2$. The RS SAT-UNSAT line is correct for $\kappa \geq 0$ while it provides a lower bound in the negative stability region.

of the computation) the storage capacity α_c happens to satisfy:

$$\alpha_c(\kappa) = \frac{1}{\int_{-\infty}^{\kappa} Dz\ (\kappa - z)^2} \tag{6}$$

Using the language from the field of random Constraint Satisfaction Problems (CSP), the portion of the $\alpha - \kappa$ plane in which solutions exist (does not exist) is called SAT (UNSAT) region, and the boundary between these two regions is called SAT-UNSAT transition.

A very similar analysis can be applied to the study of the teacher-student scenario. In this case the training problem is always SAT because at least the teacher is a solution, while the quantity of interest is the overlap between the teacher and a typical solution, which is usually denoted as R and is directly related to the generalization error through (see Fig. 7):

$$\epsilon_g = \frac{1}{\pi} \arccos R$$

1.2.3 Replica symmetry breaking

The result provided in eq. (6) has been derived in the RS ansazt, i.e. under the assumption that the saddle point integral (5) get its maximum for RS Q_{ab} matrices.

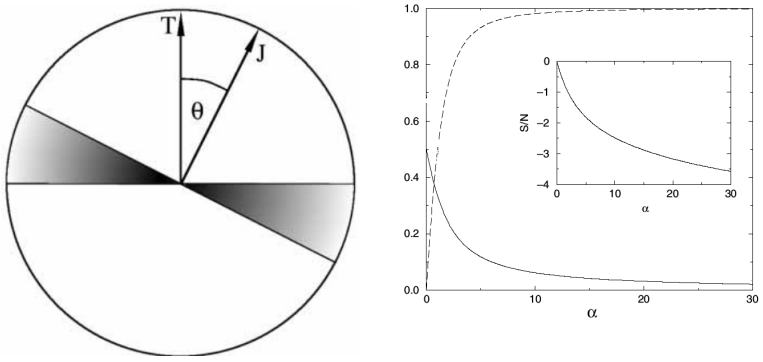

Figure 7: Left panel: the geometric interpretation of the teacher-student scenario. The teacher is denoted by T and the student by J. The figure shows the projection of the W space (the N-dimensional spherical surface) on the plane selected by the two vectors T, J. Patterns whose projection falls in the shaded region will be classified wrongly by the student. The expression for the generalization error follows noticing that $R = \cos\theta$.
Right panel: (solid line) generalization error as a function of the size of the training set in the teacher-student perceptron at $\kappa = 0$. Notice that it monotonically decreases with α, starting from the random guess value $1/2$. The dashed line is the overlap R, that starts from 0 and has 1 as asymptotic limit. Inset: the entropy, that is negative because the model is continuous and decreases monotonically with α. We add that in the disrete perceptron there is a first order transition through which the teacher becomes the only solution to the problem. The generalization error and the entopy exhibit a vertical drop to 0 at this critical point.

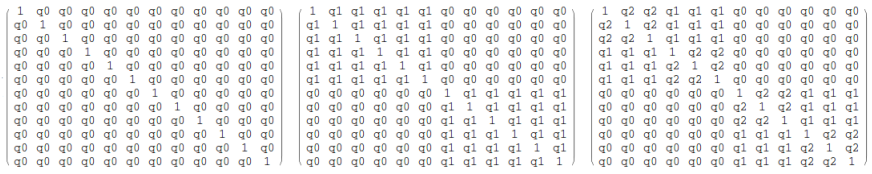

Figure 8: RS, 1RSB and 2RSB examples of *Parisi matrices* with $n = 12$, generated with a recursive code in Mathematica.

A brief historical note: the replica trick (and also the overlap of typical configurations in the same state of a spin-glass system) was first introduced by Edwards and Anderson in a nearest-neighbour disordered Ising model of structural spin glass [4]. Sherrington and Kirkpatrick [28] extended the method to the fully connected version of disordered Ising model (SK model). They noticed that some inconsistencies occured within the RS ansazt (for example negative entropies at low temperatures), while their results agreed well with the numerical experiments at high (i.e. near the spin-glass transition or higher) temperatures.

De Almeida and Thouless showed [29] that the RS solution is unstable in the space of Q_{ab} matrices and Giorgio Parisi [30] found the right scheme of *replica symmetry breaking* (RSB): the first step of RSB consists in dividing Q_{ab} in n/m blocks, within each block replicas are treated symmetrically and blocks themselves are treated simmetrically. So instead of only one order parameter q_1 that describes the typical overlap between configurations in the same *state* or *cluster*, two other parameters are needed: the overlap q_0 between configurations of different clusters and the probability m or $1 - m$ that two configurations be in different or the same states.

The 1RSB procedure can be iterated within each diagonal block to yield kRSB matrices. The continuous limit of this procedure takes the name of fullRSB solution; here the order parameters become a function and the saddle point equation a partial differential equation. The transition from RS to fullRSB is (usually) continuous, while the transition RS-1RSB is discontinuous. The SK spin-glass phase happens to be a fullRSB phase.

It can be shown [27] that in the SAT region at $\kappa \geq 0$ the RS solution is stable and the instability line (AT line) coincides with the SAT-UNSAT boundary. For negative κ the RS anszt becomes instable already in the SAT region. For any κ, the transition is a continuous fullRSB one [31].

1.3 Belief Propagation

A quite general problem in statistical mechanics and in probabilistic models with many variables is estimating local observables like magnetizations or quantities that involve only few variables, e.g. two point correlations functions and thus susceptibilities, while being given the joint probability distribution of all variables. For example, in an Ising model the probability of a configuration is given

by Boltzmann distribution and the energy is a sum over terms of one or two spins, so it could be computed from the knowledge of magnetizations and 2-spin correlations. If there isn't any clever trick or approximation, one has to sum the partition function or marginal partition functions, that is nearly always unfeasible.

In this subsection we are interested in discussing probabilistic models whose joint probability distribution factorizes in the product of many functions of few variables each. In 1.3.1 we set up the problem and introduce an useful representation for probabilistic models. In paragraph 1.3.2 we derive the cavity equations for tree-like graphical models and in 1.3.3 write down the algorithm known as Belief Propagation; in subsection 1.3.4 we state its exactness on tree factor graphs and discuss possible extensions.

1.3.1 Probabilistic graphical models

Consider a systems consisting of N variables $x_1, ..., x_N$ with $x_i \in \chi$, χ being a finite alphabet. The most general case is that every degree of freedom of the system interact with each other in a complicated way, i.e. not only through 2-body, 3-body,... interactions but with an N-body interaction, and with distribution probability $p(x_1, ..., x_N)$ that cannot be simplified in any way. The other extreme is that the N variables do not interact, i.e. they are independent variables and their joint probability distribution is simply the product of the marginals:

$$p(x_1, ..., x_N) = \prod_i p(x_i)$$

In the striking majority of cases, however, the system belongs to an intermediate class, the interactions being 2-body or p-body like, and with many variables that do not interact directly but only through other degrees of freedom. In this case:

$$p(x_1, ..., x_N) = \frac{1}{Z} \prod_{a=1}^{M} \psi_a(\overrightarrow{x}_{\partial a})$$

where Z is a normalization and the ψ_a's are M non-negative real *compatibility functions* that represent the interactions between groups of degrees of freedom, the variable of each group being denoted by $\overrightarrow{x}_{\partial a}$. The network of mutual interactions of such a system admits a nice graphical representation: to each variable there corresponds a *variable node*, directly interacting variable nodes being linked to a same *factor node* $a \in F$. The resulting bipartite graph is referred to as the *factor graph* of the probabilistic model, see examples in Fig. 9 and 10.

Two (sets of) variables which do not interact directly, but are in the same connected component of the factor graph, are not independent, but are correlated by "intermediate" variables. This idea is made more precise by the following *global Markov property*:

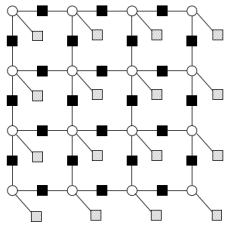

Figure 9: Factor graph associated with a 2D Ising or Edwards-Anderson model. The round white nodes are the variables nodes of the spins, the black square boxes represent the 2-body couplings between near spins, and the gray square boxes are the local external magnetic fields, which bias the spins in an independent way.

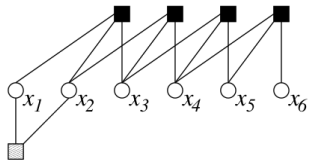

Figure 10: Factor graph relative to a Markov chain of memory 2, the gray box representing the initial conditions.

Let $\overrightarrow{x}_A, \overrightarrow{x}_B, \overrightarrow{x}_S$ be three disjoint sets of variables. If any path joining a node of A with a node of B contains an element of S, then S *separates* A and B. If this holds then $\overrightarrow{x}_A$ and $\overrightarrow{x}_B$ are conditionally independent with respect to $\overrightarrow{x}_S$, i.e.:

$$p(\overrightarrow{x}_A, \overrightarrow{x}_B|\overrightarrow{x}_S) = p(\overrightarrow{x}_A|\overrightarrow{x}_S)\, p(\overrightarrow{x}_B|\overrightarrow{x}_S)$$

1.3.2 The cavity approach

Given a probabilistic model, we would like to solve efficiently at least three tasks:

- compute the marginal distributions of single variables or of small groups of variables
- sample points from $p(x_1, ..., x_N)$
- sum the partition function or, equivalently, compute the free-entopy of the system.

In systems whose factor graph is a *tree* (i.e. the underlying graph is connected and acyclic) the three taks above can be achieved, by exploiting the following **cavity** approach:

each variable node i can be seen as the root of the tree and spans the branches a's, so that the marginal can be computed as

$$p(x_i) = \sum_{x_k,\, k\neq i} p(x_i, x_k) = \frac{1}{Z} \prod_a \sum_{\overrightarrow{x}_a} \Psi_{a\to i}(x_i, \overrightarrow{x}_a) \equiv \frac{1}{Z} \prod_a Z_{a\to i}(x_i)$$

where with $\Psi_{a\to i}$ we have indicated the product of all compatibility functions belonging to the branch a. Descending down the tree:

$$Z_{a\to i}(x_i) \equiv \sum_{\overrightarrow{x}_a} \Psi_a(x_i, \overrightarrow{x}_a) = \sum_{j\in\partial a\setminus i} \psi_a(x_i, \vec{x}_j) \prod_j \prod_{b\in\partial j\setminus a} \sum_{\overrightarrow{x}_{\partial j\setminus b}} \Psi_{b\to j}(x_j, \overrightarrow{x})$$

The mechanism is that at each step the branch contribution factorizes in the contributions of its child branches. In the end:

$$Z_{a\to i}(x_i) = \sum_{j\in\partial a\setminus i} \psi_a(x_i, \vec{x}_j) \prod_{j\in\partial a\setminus i} Z_{j\to a}(x_j)$$

$$Z_{i\to a}(x_i) = \prod_{b\in\partial i\setminus a} Z_{b\to i}(x_i)$$

1.3.3 Belief Propagation

The above equations are called *RS cavity equations* and in the end are simply relations between marginals and conditional probabilities of the system. The cavity equations can be turned into the algorithm known as Belief Propagation (BP) by trying to solve them by recurrence and hoping they converge to a fixed point. To this end it is useful to recast the cavity equations in term of messages, so to obtain the update rules known as *BP equations*:

$$\nu^t_{i\to a}(x_i) \propto \prod_{b\in\partial i\setminus a} \nu^{t-1}_{b\to i}(x_i)$$

$$\nu^t_{a\to i}(x_i) \propto \sum_{x_j,\, j\in\partial a\setminus i} \psi_a(x_i, \vec{x}_j) \prod_{j\in\partial a\setminus i} \nu^t_{j\to a}(x_j) \tag{7}$$

with the further conditions:

$$\nu^t_{i\to a}(x_i) \propto 1 \quad if\ \partial i\setminus a = \{\}$$

$$\nu^t_{a\to i}(x_i) \propto \psi_a(x_i) \quad if\ \partial a\setminus i = \{\}$$

where the ν^t are called *messages at time t* [6] and the symbol $\propto$ means that the messages are to be taken normalized, so that the fixed point messages and the partial partition functions are simply related by a normalization factor:

[6] we will usually denote with only ν the fixed point messages.

$$\nu_{i\to a}(x_i) = \frac{Z_{i\to a}(x_i)}{\sum_x Z_{i\to a}(x)}$$

$$\nu_{a\to i}(x_i) = \frac{Z_{a\to i}(x_i)}{\sum_x Z_{a\to i}(x)}$$

This normalization is crucial to the probabilistic interpretation of the fixed point messages:

- $p(x_i) = \prod_{a\in\partial i} \nu_{a\to i}(x_i)$, i.e. the marginal probability of a variable is given by the product of incoming messages
- $\nu_{i\to a}(x_i)$ is the marginal probability of variable i in a modified graphical model in which the factor a has been erased
- $\nu_{a\to i}(x_i)$ is the marginal probability of variable i in a modified graphical model in which the factors $b \in \partial i/a$ have been erased

The local nature of the update and the fact that there are input and output messages at each node at each time suggests the name of *message-passing algorithms*.

Now we briefly show how from the knowledge of fixed point messages it is possible to solve the three problems mentioned at the beginning of the previous paragraph.

We start considering the problem of computing the marginal distribution of a few variables. Let R be a connected[7] subgraph of factor nodes $F_R \subset F$ and variable nodes $\{\vec{x}_R\} = \cup_{a\in F_R}\partial a$, and define $\partial R = \cup_{i\in R}\partial i \backslash F_R$. Then the marginal of $\{\vec{x}_R\}$ is given by the product of messages incoming in R from ∂R weighted with the compatibility functions of the factor in F_R, everything evaluated in $\vec{x}_R$:

$$p(\vec{x}_R) \propto \left(\prod_{a\in F_R} \psi_a(\vec{x}_{\partial a})\right)\left(\prod_{a\in\partial R} \nu_{a\to i(a)}(x_{i(a)})\right) \tag{8}$$

where $i(a)$ is the only[8] vertex in $\partial a \cap R$, $a \in \partial R$. The condition that R contain few variable nodes is due to the fact that the normalization constant has to be computed. In particular if $F_R = \{a\}$:

$$p(x_{\partial a}) \propto \psi_a(x_{\partial a}) \prod_{i\in\partial a} \nu_{i\to a}(x_i)$$

The second problem was sampling $\vec{x} \in \chi^N$ according to $p(\vec{x})$. This problem can be reduced to sampling from one variable marginals: for $i = 1, ..., N$ we sample a value x_i' according to $p(x_i|\vec{x}_U)$. We fix x_i to the extracted value by

[7] the problem of computing the marginal or the correlation function of two nodes far away in the graph requires a method known as *susceptibility propagation*.

[8] this is true if the graph is a tree and if the subgraph R is connected, as we can always take (otherwise we repeat for each connected component).

introducing a new factor node and add i to U. We run BP on the modified graph and compute $p(x_{i+1}|\vec{x}_U)$. We repeat until $U = \{1, ..., N\}$.

The third problem is computing the free-entropy of the system $\phi[p] = \log Z[p] = H[p] - \beta U[p]$, where H is the usual entropy associated with a probability distribution and

$$\beta U[p] = -\sum_{\vec{x}} p(\vec{x}) \sum_{a} \log \psi_a(\overrightarrow{x}_{\partial a})$$

in analogy with the Boltzmann distribution. For tree graphs the free-entropy can be expressed [11] in terms of the messages as the sum of local contributions of three kind:

$$\phi(\vec{\nu}) = \sum_{a} F_a + \sum_{i} F_i - \sum_{(i,a)} F_{(i,a)} \tag{9}$$

where:

$$F_a = \log \left[\sum_{x_{\partial a}} \psi_a(x_{\partial a}) \prod_{i \in \partial a} \nu_{i \to a}(x_i) \right]$$

$$F_i = \log \left[\sum_{x_i} \prod_{a \in \partial i} \nu_{a \to i}(x_i) \right]$$

$$F_{(i,a)} = \log \left[\sum_{x_i} \nu_{i \to a}(x_i) \nu_{a \to i}(x_i) \right]$$

1.3.4 Theorems and extensions

In the previous paragraph we derived all results having in mind tree-like graphical models. Here we precise the validity of this approach with a theorem for tree graphs and discuss where we can extend BP to non tree-like models.

Theorem Given a tree graphical model of diameter T, the BP algorithm converges in at most T iterations, independently of the initialization of the messages. The fixed point messages yields the right marginals of the problem. Moreover, the Bethe free entropy (9) computed with the fixed point messages is equal to the free entropy of the system.

Now we want to apply the BP approach to general graph. We start noticing that both the BP equations and the Bethe free entropy (as a function of the messages) are meningful independently of the topology of the graph. In particular, if the messages converge we can take them as an approximation of the marginals and free-entropy.
We mention a variational result [11] that holds in general graphical models, and will be useful later:

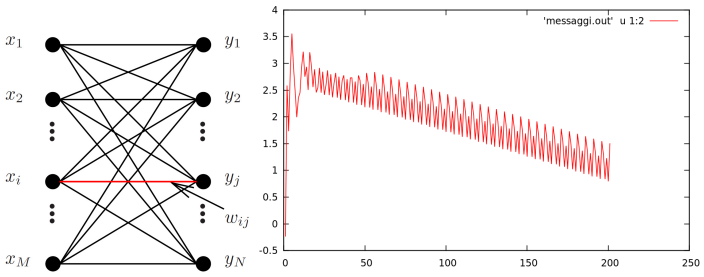

Figure 11: The *bipartite matching* or *assignement* problem can be solved by the so called Min-Sum meassage-passing algorithm. It can be rigourosly proven [12] that this approach always return an optimal assignement and a bound on the number of BP iterations can be provided. Nevertheless, the messages do not converge, but the exit condition is that they oscillates with some period but for a uniform shift. This behaviour is shown in the right panel, where we plot the value of a message versus time in a random instance of bipartite matching, with weights generated from an exponential distribution. Nice analytical results exist for the random matching problem [6, 32]. We have mentioned the matching problem in order to give an idea of the great flexibility of message-passing algorithms, useful in many other contexts, e.g. l*ow density parity check codes*.

Theorem The stationary points of the Bethe free entropy $F(\vec{\nu})$ are fixed point of BP. BP has at least one fixed point and finite fixed points are stationary points of $F(\vec{\nu})$.

Finally, we address the question of when we can expect the results of BP to be reasonable approximations. The key reason why BP works in tree graphs is that when we consider the modified cavity model, the different branches of each variable node are independent factor graphs. If there are loops, this is no longer true. We can thus expect that BP provide good approximations if variables adjacent to a same vertex are weakly correlated in the modified cavity model. This is expected to happen at least in two circumstances:

- for graphs with only "large" (at least $O(\log N)$) loops
- for fully connected graphs which can be treated in a replica symmetry assumption: indeed, in fully connected models two-point correlations are linked to the clustering property and are hence zero in pure states.

However, as fig. 11 shows, message-passing is a really flexible strategy for dealing with several problems.

2 Accessible states in simple discrete neural networks

In this chapter we want to explain the background and the motivation behind this work, following the chronological steps of the discovery of dense clusters of solutions for the training of neural networks with discrete synapses.

In section 2.1 the original puzzle about training simple discrete neural networks is described; this puzzle led to a large deviation analysis that predicts the existence of cluster of solutions, extremely attractive regions for proper algorithms.

In sec. 2.2 the idea is to exploit the knowledge of the structure of solutions in order to conceive new algorithms.

Finally, in the last paragraph we discuss the future developments and the importance of this line of research.

2.1 Dense clusters of solutions in learning with discrete synapses

The main concern of this section is on simple architectures of neural networks with discrete synapses. The interest for discrete synapses is of twofold nature:

- the hardware of computers is based on binary units [24]
- experiments on single brain synapses suggest that the neural activity is characterized by switch-like events between a finite number of synaptic states [23]

Under a mathematical point of view, the training of a neural network can be recast in the language of random constraint satisfaction problems (CSPs), with an associated factor graph like in Fig. 12.

In 1971 Stephen Cook [33] proved that the K-SAT problem, where N boolean variables are given and have to satisfy $M = \alpha N$ logical clauses involving K variables each, is NP complete. During the last two decades there has been a renewed interest in random CSPs, as it has been understood that even if the worst case complexity of many CSPs is NP, the average instances of a problem may be solved efficiently in polynomial time [9].

2.1.1 Franz-Parisi entropy and equilibrium analysis

The problem of training a discrete neural network is NP hard even if considering the simple perceptron, and also considering the average case up to 2005 the most efficient training algorithms could only learn a logarithmic number of patterns in a time polynomial in N [13]. The theoretical explaination of this numerical hardness has been provided by Huang and Kabashima [34] in 2014, computing the so-called Franz-Parisi potential [35] for the discrete perceptron. The original

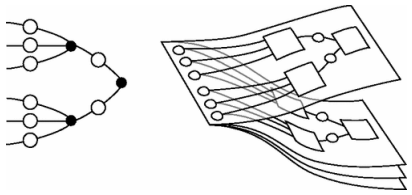

Figure 12: On the left a simple architecture of feedforward neural network, namely the *tree-like committee* machine. The committe is a two-layer feedforward neural network consisting of N input and K hidden units; each hidden unit is linked to a subset of the inputs (in the follow we will mention three-like and fully-connected architectures) and behaves as a single perceptron. The final ouput is given by the "majority vote" of the intermediate outputs provided by the hidden units. The committee shown has $K = 2$ hidden units (middle black dots). On the right the associated factor graph, where every "sheet" represents the constraint provided by each pattern in the training set (notice that the empty circles are the synaptic weights that here play the role of variables of the system).

idea of Franz and Parisi was studying the metastable states of spin glass models. They considered the *p-spin spherical model*

$$H(\{\sigma_i\}) = -\sum_{i_1,..,i_p} J_{i_1,..,i_p}\sigma_{i_1}...\sigma_{i_p}$$

where $\{\sigma_i\}$ are N continuous real variables constrained on the sphere $\sum \sigma_i^2 = N$ and the J's are disordered couplings distributed independently of each other according to a Gaussian distribution. The thermodynamical (equilibrium) analysis predicts [36] that at high temperatures this system lives in a paramagnetic replica symmetric phase (paramagnetic means that the typical overlap between Boltzmann configuration is 0), while at a certain temperature T_S the system undergoes a spin-glass transition to a 1RSB phase, characterized by overlaps $q_1 > q_0$, $q_0(T_S) = 0$. However, a dynamical analysis shows that below $T_D > T_S$ the dynamics of the system gets trapped in metastable states, which are the precursors of the discontinuous static transition.

The Franz-Parisi method couples the main system to a reference one, by requiring that its configurations be at fixed distance from a configuration of the auxiliary system sampled according to the equilibrium distribution. In general the reference and main systems can be at different (inverse) temperatures $\tilde{\beta}$ and $\beta = 1/T$. The Franz-Parisi potential is the free-energy of the main system averaged over the Boltzmann distribution of the reference one:

$$NV_{FP}(s) = \left\langle \frac{1}{Z(\tilde{\beta})} \int d\tilde{\sigma} e^{-\tilde{\beta}H(\tilde{\sigma})} \{-T \log Z(\tilde{\sigma}, s, \beta) - F(\beta)\} \right\rangle_J$$

$$Z(\tilde{\sigma}, s, \beta) = \int d\sigma \, e^{-\beta H(\sigma)} \delta(s - \frac{\sigma \cdot \tilde{\sigma}}{N})$$

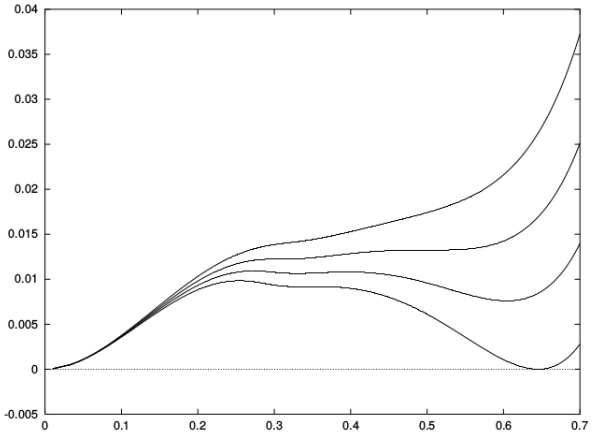

Figure 13: Franz-Parisi potential versus overlap s at $T = \tilde{T}$ for various temperatures (lower lines correspond to lower temperatures)

where $\tilde{\sigma}$ denote the configurations of the reference system, s is the fixed overlap (equivalent to fixing the distance) and F serves as a normalization, so that the potential is zero if $s = q_{typ}$, that is equivalent to uncoupled systems. The Franz-Parisi potential is precisely the potential associated with the thermodynamical force needed to constrain the main system.

The Franz-Parisi potential can be computed with the standard replica method (while the dynamical analysis requires different tools such as diagrammatic expansions or calculation of the *complexity* of TAP free energy [36]) to yield, for temperatures greater than T_S (we consider $T = \tilde{T}$), a plot like Fig. 13 . The first (left) minimum corresponds to $s = q_{typ} = 0$, as we are in the paramagnetic phase. At $T > T_D$ this is the only minimum. At $T = T_D$ a second minimum develops, corresponding to the appearance of metastable states. This minimum reaches 0 at T_S in $s = q_1$. For lower temperatures the analysis requires a RSB computation, but it is expected that both minima have 0 potential and shift following $s = q_1, q_0$.

We go back to the discrete perceptron now. It is known [20] that the zero temperature storage problem is SAT and replica symmetric up to $\alpha \simeq 0.833$, where a RSB and UNSAT transition occurs. Huang and Kabashima [34] considered the storage scenario and computed the Franz-Parisi *entropy* at zero temperature, i.e. the number of solutions at fixed overlap[9] s from a reference synaptic configuration $\tilde{W}$ drawn with uniform probability from the set of all

[9] actually the replica computation is performed introducing a Lagrange parameter conjugate to the overlap, γ, which acts as an external field of direction $\tilde{W}$:

$$\mathbb{H}_{FP}(\gamma) = \left\langle \sum_{\tilde{W}} \chi(\tilde{W}, \xi) \log \aleph(\tilde{W}, \gamma) \right\rangle_{\xi}$$

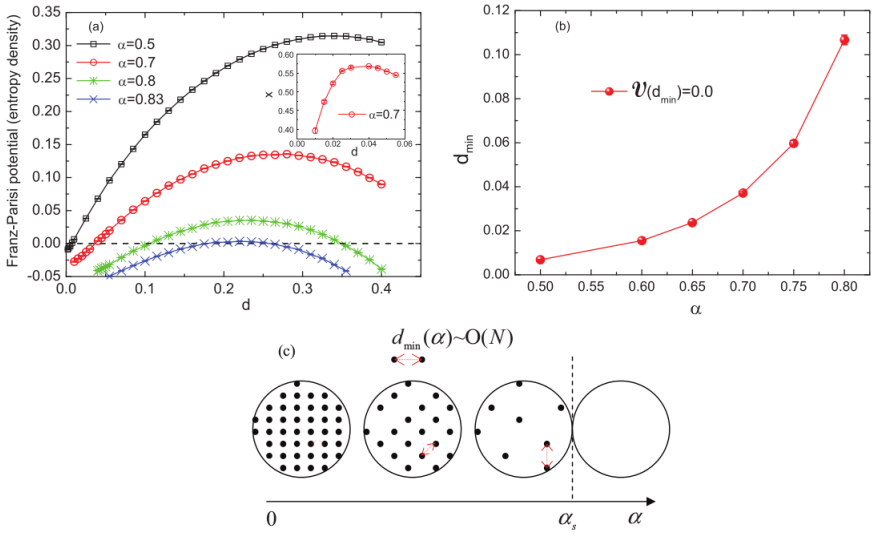

Figure 14: In the left upper panel the Franz Parisi entropy is plotted versus the Hamming distance, for various values of α (notice that for $\alpha \to \alpha_c$ the entropy sinks below zero). The inset shows the Lagange parameter conjugate to the distance (at the maximum the entropy changes concavity). Right upper panel: critical distance as a function of the training set size. Lower panel: qualitative sketch of the space of solutions at different α.

possible solutions

$$\mathbb{H}_{FP}(s) = \left\langle \sum_{\tilde{W}} \chi(\tilde{W},\xi) \log \aleph(\tilde{W},s) \right\rangle_{\xi}$$

$$\aleph(\tilde{W},s) = \sum_{W} \chi(W,\xi)\delta(s - \frac{W\cdot\tilde{W}}{N})$$

where $\chi(W,\xi) = \prod_\mu \theta(W\cdot\xi^\mu)$ (we are taking all outputs positive: $\sigma^\mu = +1$). Since $\log\aleph$ is only extensive while the number of solutions is exponential in N, the sum over $\tilde{W}$ is equivalent to a sum over typical solutions.

The Franz-Parisi entropy for the discrete perceptron is shown in Fig. 14. As expected, the entropy at fixed distance decreases increasing α. The other

$$\aleph(\tilde{W},s) = \sum_{W} \chi(W,\xi) e^{\gamma W\tilde{W}}$$

expected feature is that the maximum is taken for distance corresponding[10] to the typical overlap between typical solutions. What is remarkable is that there exists a distance $d(\alpha)$ such that, below this distance, around typical solutions there are at most few (in a not exponential number) solutions, corresponding to zero Franz-Parisi entropy.

The conclusion is that typical solutions are isolated, hence a glassy energy landscape and the computational hardness of the training problem. This last interpretation is not immediate and certainly not rigorous (the typical solutions may be distant but with large basins of attraction). There are mainly two ideas behind such interpretation, coming from the experience gathered in the study of other random CSP with binary variables:

- if two solution are several spin flips distant, then an algorithm may succeed in satysfing $M-1$ patterns, but the partial solution found has not roubustness with respect to the missing Mth pattern; on the contrary, if a solutions is surrounded by many other solutions it is easier to find a little correction to fit the addition of a new pattern. With analogy to KSAT, it is said that such isolated solutions are *frozen*.

- in the KSAT and in the Q-coloring problems it is known that algorithms succeed to find solutions only if these solutions are in (well separated) clusters of exponentially many close-by solutions, while there is a frozen phase in which every known algorithm fails because the landscape is dominated by metastable states [37, 9, 21]

In conclusion, even though a rigourous proof would require a finite temperature analysis of metastable states, which is technically more difficult because of RSB effects, also at the light of the findings discussed below in this Chapter we think that the logic of Huang and Kabashima (isolated solutions ergo computational hardness) is essentially correct for all free-energy minimization based algorithms.

2.1.2 Bumping into non-equilibrium solutions

It was a period of fervent interest in message-passing algorithms when Alfredo Braunstein and Riccardo Zecchina [13] (2005) tried to use BP for finding solutions of the storage problem of simple (perceptron and one hidden layer) discrete neural networks. Being fully connected RS models BP is expected to yield , for large enough N, the correct marginals and the correct entropy of the problem. Actually, the entropy estimated with BP, averaged on different samples, matches the replica theoretical computation, see Fig. 15. To turn BP into a solver for the storage problem (i.e. we want single solutions, not their entropy) the naive idea is to take the polarizations of the marginals $W_i \leftarrow \arg\max p(W_i)$. However,

[10] the distance between two configurations is defined as the *Hamming distance* i.e. the fraction of spin flips neccessary to take one configuration in the other:

$$d = \frac{1-q}{2}, \quad q = \frac{W \cdot W'}{N}$$

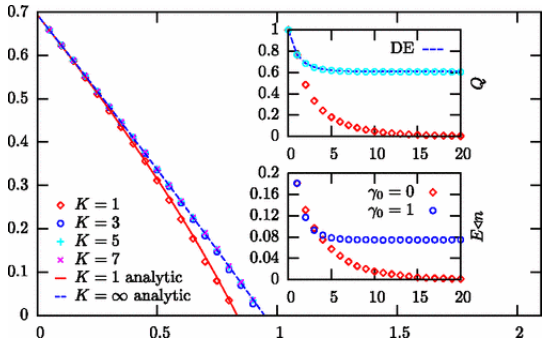

Figure 15: BP entropy vs α for $N = 3465$ and different numbers K of hidden units (the considered architecture is a committee machine, $K = 1$ is the perceptron). The BP estimate matches the replica analytical results. The upper inset shows Q as a function of the number of BP iterations, with and without reinforcement, where $\gamma^t = (\gamma_0)^t$. With reinforcement the variables are completely polarized after some time. The dashed line is the prediction for BP as given by the *density evolution* analysis (DE)[10]. The lower inset shows the normalized number of violated constraints E/N versus time. The data of both insets have been averaged over simulations with $N = 10^5 + 1,\ \alpha = 0.6$.

this approach neglects completely the correlations between synapses, which are important for finding single solutions. The inconsistency of this attempt is quantified by the quantity $Q = 1 - \frac{1}{MN}\sum_{i,\mu} m_{i\to\mu}^2$, where $m_{i\to\mu} = \sum_{W_i} W_i p_{i\to\mu}(W_i)$ is the *cavity magnetization* of synapsis i in the problem without pattern μ. Q is zero if the marginals are Kronecker deltas and 1 if are uniform. As Q doesn' t approach zero, see Fig. 15,we cannot use BP for finding single solutions.

A possible way out is to *decimate* the problem: we take the n most polarized (according to the non cavity magnetization) variables at each iteration and fix them to be ± 1, run BP on the new problem and so on. Braunstein and Zecchina proposed a very simple but more continuous and fully local version of this trick and called *it reinforcement*: at time $t+1$ each site is subject to an external field proportional to the site magnetization at time t. In this way at the fixed point the variables are completely polarized. Namely, the BP equations are modified in this way

$$h_i^{t+1} = h_{BP,i}^{t+1} + \{0\, w.p.\, \gamma^t,\ h_i^t\, w.p.\, 1-\gamma^t\}$$

where the h's are linked to magnetizations by $m = \tanh h$ and are the auxiliary non cavity quantities used to compute the messages from synapses to factors.

Anyway the details of the algorithm are not very important and actually many efficient variations exist [1]. What is important is that there exist heuristics which succeed in finding solutions of the training problem up to $\alpha_A \simeq 0.75$ and in running time of order $O(N^2 \log N)$. The results for the original reinforcement learning [13] are shown in Fig. 16

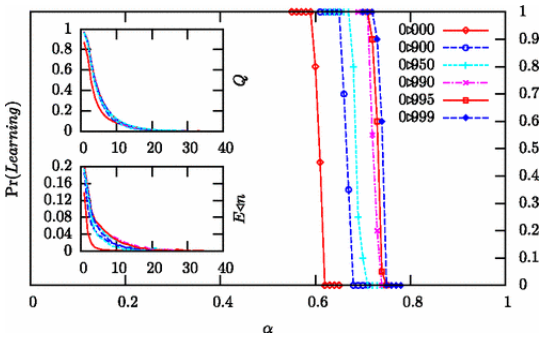

Figure 16: The probability of success of reinforcement as a function of α and different γ_0's, $\gamma^t = (\gamma_0)^t$. For each α 20 samples are considered with $N = 10^5 + 1$ and $K = 1$ (binary perceptron). With softer decimations ($\gamma_0 \to 1$) the algorithmic storage capacity increases up to $\alpha \simeq 0.74$, but with the drawback that more time is required, according to the scaling: $running\text{-}time \propto \frac{1}{1-\gamma_0}$. The insets show Q and E/N versus time for K in different ranges and suggest that the behaviour of the algorithm is robust in K.

After Huang and Kabashima 2014 paper a question arised naturally: solutions found by reinforcement based algorithms are typical or non-equilibrium solutions?

Numerical experiments [1] suggests that the solutions found by efficient algorithms are not isolated typical solutions but they belong to clusters of solutions of extensive size:

- taken an algorithmic solution $\tilde{W}$ and performing a random walk (random spin flips)[11] solutions are found up to a number of spin flips $O(N)$. In particular, the number of solutions at fixed Hamming distance grows exponentially in N, as can be estimated sampling with a random walk configurations at fixed distance and different N.

- the entropy of solutions is computed with BP on the modified system with external field proportional to an algorithmic solution $\tilde{W}$, the proportionality constant being the Lagrange parameter associated with the distance[2]. The results are shown in Fig. 15: the *local entropy* of solutions from an algorithmic solution does not go to zero for small distances and is higher than the Franz-Parisi entropy (local entropy from typical solutions) even when this is not zero.

The straightforward conclusion is that the equilibrium analysis does not capture the behaviour of reinforcement based heuristics. The numerical results suggest

[11] more precisely they consider only solutions in the same connected cluster, where two solutions W, W' are connected if there exist a sequence of solutions $W, W_1, ..., W'$ such that W_i, W_{i+1} differ for a single spin-flip

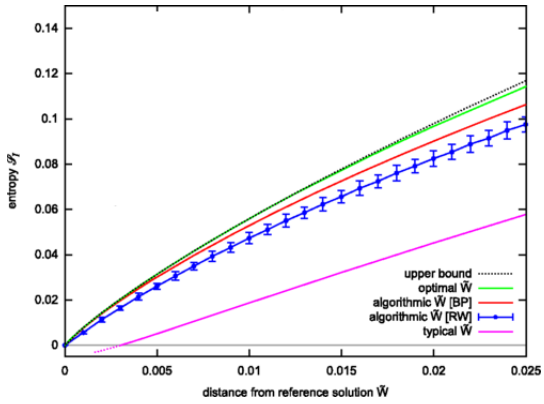

Figure 17: Entropy of solutions at fixed distance from a reference solution $\tilde{W}$ (local entropy), for the storage problem at $\alpha = 0.4$. $\tilde{W}$ is a typical solution in the magenta line (Franz-Parisi entropy), a solution found algorithmically in the blue (RW estimate of entropy) and red (BP estimate) lines, a large-deviation solution with $y = +\infty$ ("optimal") in the green line (analytic local entropy). The black dotted line is the greedy upper bound provided by elementary combinatorics ($\alpha = 0$). Notice that the RW line is below the BP one because the former counts only the solutions in the same connected cluster. Similarly, the BP line is below the optimal one because in the latter the $\tilde{W}$ is optimized at each distance, while in the former $\tilde{W}$ is independent of the distance. The fact that the numerical results fit so well the theoretical prediction at small d suggests that the densest regions are the most attractive to algorithms.

that the solutions found by reinforced algorithms may be better described by a local entropy dependent measure.

2.1.3 Large deviation analysis

The numerical findings of the previous paragraph suggest that the effective heuristics manage to find clusters consisting in an exponential number of solutions but still subdominant with respect to the more numerous isolated typical solutions. In order to study analytically these non-equilibrium solutions, their non isolated nature suggests to reweight each solution with its local entropy, namely to study the *large deviation* partition function [1]:

$$\mathcal{Z}(y,s) = \sum_{\tilde{W}} \chi(\tilde{W},\xi) e^{y \log \aleph(\tilde{W},s)} \tag{10}$$

$$\aleph(\tilde{W},s) = \sum_{W} \chi(W,\xi)\delta(s - \frac{W\cdot\tilde{W}}{N})$$

This is a system with formal Hamiltonian $\mathcal{H}_s(\tilde{W}) = -\log \aleph(\tilde{W},s)$ and formal inverse temperature y. The *local* or *internal entropy* $\mathcal{S}_\mathcal{I}$ is precisely defined as

the (opposite of the) thermodynamical internal energy of the reweighted system:

$$\mathcal{S}_{\mathcal{I}}(y,s) = -\frac{1}{N}\left\langle \left\langle \mathcal{H}_s(\tilde{W}) \right\rangle_{y,s} \right\rangle_{\xi} = \left\langle \left\langle \frac{1}{N}\log \aleph(\tilde{W},s) \right\rangle_{y,s} \right\rangle_{\xi}$$

where $\langle\cdot\rangle_{y,s} = \sum_{\tilde{W}} \cdot \chi(\tilde{W},\xi)e^{y\log\aleph(\tilde{W},s)}$ denotes reweigthed Bolzmann averages. The local entropy $\mathcal{S}_{\mathcal{I}}$ reduces to the Franz-Parisi entropy in the limit $y \to 0$. Another quantity of interest is the *external entropy* $\mathcal{S}_{\mathcal{E}}$, defined as the entropy of the reweighted system, i.e. as the number of configurations $\tilde{W}$ that dominate the large deviation measure:

$$\mathcal{S}_{\mathcal{E}}(y,s) = \mathcal{F}(y,s) - y\mathcal{S}_{\mathcal{I}}(y,s)$$

where $\mathcal{F}(y,s) = \frac{1}{N}\log\mathcal{Z}(y,s)$ is the free entropy of the reweighted system.

In the case of the perceptron, the thermodynamics of the reweighted system can be solved by the replica method, the steps are: take the analytical continuation of y integer, replicate the system so to have a pair of replica indices and finally perform the quenched average over the training set. (We don' t focus on the details of the computation as we will be dealing with very similar calculations in Chapter 3 and 4.)

Performing the computation with the RS assumpion yields an external entropy $\mathcal{S}_{\mathcal{E}}(y,s)$ which above a certain temperature $y > y^* = y^*(\alpha,s)$ is negative, this for every α, s. This unphysical result resembles the low temperature inconsistencies of spin glass computations at inadeguate level of RSB. A 1RSB computation being technically unfeasible, the fix is picking $y = y^*$ and assessing if the results are reasonable.

Fig. 18 shows $\mathcal{S}_{\mathcal{I}}(y*(\alpha,s),s)$ for various α's. Up to $\alpha \simeq 0.77$ the local entropy curves are monotonic; for $\alpha \simeq 0.78$ there is a region, at small distances, with negative derivative, but at smaller distances the curve is acceptable; for greater α the entropy becomes negative, then there is a gap [12] but a meaningful curve reappears near zero distance.

Moreover, for every training set size, for small enough distances and for large enough distances, i.e. comparable with the typical distances, $y^* \to \infty$ and the local entropy reaches in the former case the upper bound $\alpha = 0$ and in the latter the Franz-Parisi entropy. This suggests that the RS computation is reliable at least in the $\alpha < 0.77$ region and enables to draw the following conclusions:

- for all $\alpha < \alpha_c$ there exist dense clusters of solutions. These clusters are thought to be in sub-exponential number ($\mathcal{S}_{\mathcal{E}}(y*,s) = 0$ and the RSB unconstrained computation mentioned below, Sec. 2.2.1, suggests that with successive steps of RSB this is the case at any $y > 0$ and in particular at $y = \infty$), but they contain an exponential number of solutions. Furthermore, the cores of these clusters are very dense.

[12] the saddle point equations don' t have acceptable (continuous) solution in this region

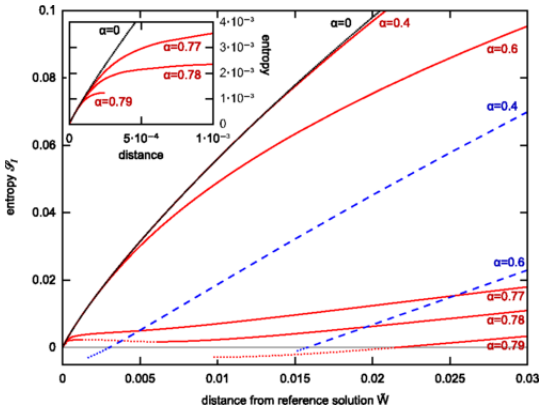

Figure 18: The local entropy $\mathcal{S}_{\mathcal{I}}(y*(\alpha,s),s)$ is shown as a function of distance $d=\frac{1-s}{2}$ for various values of α. The black line is the $\alpha=0$ geometrical bound, while the blue dashed lines are the Franz-Parisi entropy. The red lines are dotted if their derivative is negative and there is a gap in the $\alpha\simeq 0.79$ curve. The inset shows a small neighborouhood of $d=0$: it seems that here the approach is consistent also for great α and hints that the core of the clusters are very dense.

- the RS nature of the space of solutions at low α suggests that the clusters of solutions are immersed in the same connected[13] region ["il clusterone" for friends], extremely accessible to algorithms; increasing α the size of cluster cores shrinks until the clusters become disconnected (RSB transition) and what remains of the dense cores appears as isolated solutions, hard to find for algorithms.

2.1.4 Teacher-student case

The teacher-student scenario shows similar non-equilibrium features [1, 2]. In this context a very interesting issue concerns the non-equilibrium generalization properties.

The generalization problem for the discrete perceptron admits an exponential number of solutions up to $\alpha_{TS}=1.245$ where a first-order transition occurs such that the teacher remains the only solution,with a corresponding vertical drop of the generalization error [38]. Given a teacher W^T and a configuration W the probability of classify wrongly a random pattern (generalization error) is given by the simple geometrical relation (see Fig.) $p_e=\frac{1}{\pi}\arccos\frac{W\cdot W^T}{N}$.

In [1] the Franz-Parisi equilibrium analysis is extended to this scenario and it holds that:

- typical solutions are isolated for every α, even when adding a positive stability;

[13] we misuse the word "connected", meaning it holds an RS description for the $\tilde{W}$. Actually, numerical experiments suggest that algorithmic solutions are connected.

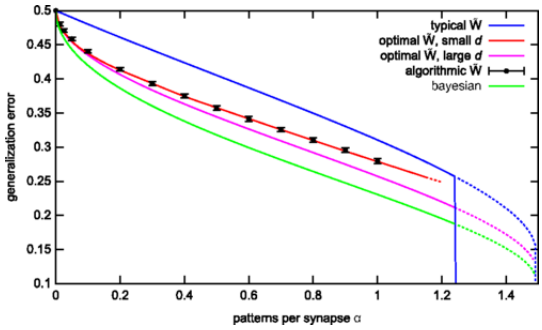

Figure 19: Generalization error versus α in the teacher-student scenario. In blue the usual replica result, in green the Bayesian case (center of mass of the solution space) which is a probabilistic lower bound. In red and magenta the generalizations of solutions typical with respect to the reweighted measure, obtained maximizing the local entropy at small and large distances respectively. The generalization error decreases increasing d up to a value d_{middle}, such that $\mathcal{S}_{\mathcal{I}}(d_{middle}) = \mathcal{S}_{\mathcal{I}}(d_{typ}) = \mathbb{H}_{FP}(d_{typ})$; in practice $d(\tilde{W}(d_{middle}), W_{typ}) = d_{middle}$ for any W_{typ} and there is a plateau of $\mathcal{S}_{\mathcal{I}}(d)$ for $d \in [d_{middle}, d_{typ}]$. Notice that the numerical results fit the small d line, showing that the densest regions are the most attractive.

- the teacher resembles a typical solution: its overlap with typical solutions has the same value as the overlap between typical solutions; suprisingly, its Franz-Parisi entropy shows that the teacher is isolated from other solutions.

Numerical experiments were thus performed, see Fig. 19:

- taking the teacher as reference configuration, the local entropy estimated with BP agrees with the replica computation (it is not exactly the Franz-Parisi entropy but even simpler, and with no reweighting)
- taking an algorithmic solution as reference configuration, the local entropy estimated with BP does not agree with the Franz-Parisi computation
- the generalization error of algorithmic solutions is lower than the theoretical prediction for typical solutions

The large deviation approach eq. (10) can be straightforwardly extended to this case, the only adjustment being contained in the χ functions and in the fact that the quenched average involves also the teacher. Once again, the analytical results describe well the numerical findings, see Fig. 19.

For both classification and generalization learning, a similar scenario is expected to remain valid for multilayer discrete networks (in [1] up to 3 hidden

layers are considered) and for Potts synapses, i.e. with $W_i = 0, 1, ..., q$ [22].

We have shown numerically that reinforced algorithms find solutions immersed in dense clusters of solutions and have been able to predict analytically some properties of these regions. The missing step is a dynamical explanation of the behaviour of algorithms. This will be discussed in the next Section.

2.2 Dynamics and algorithm development

The large deviation measure describes but does not explain the behaviour of reinforced algorithms. In this Section we try to understand the dynamical reasons that drive an algorithm towards dense regions of solutions. We start trying to design algorithms that exploits the structure of the solutions space, the reason of doing this is threefold:

- as a proof of concept that dense regions of solutions are able to attract algorithms for their very dense nature
- in paragraph 2.2.2 a BP-based algorithm is introduced inspired by theoretical considerations. With some simplifications the reinforcement-based heuristics are recovered; this should provide a (non-rigourous) proof of why the dynamics drive algorithms towards dense clusters of solutions
- for the very goal of designing new efficient algorithms, for learning problems and hopefully for other random CSPs

The conclusion seems that learning is possible in presence of dense regions of solutions that attract algorithms somehow sensible to the local entropy landscape, i.e. the dynamics is a local-entropy driven one. This closes the loop.

2.2.1 Entropy Driven Monte Carlo

Free-energy based algorithms fail in training neural networks with discrete synapses, for the reasons discussed in paragraph 2.1.1. The numerical results and the large deviation analysis suggest that effective algorithms are attracted towards dense regions of solutions.

The natural idea [2] is thus introducing a simple Monte Carlo replacing, as objective function, the energy of the non reweighted system with the energy of the reweighted system, i.e. the local entropy. In practice, the algorithm tries to maximize the local entropy with a Metropolis-like [39] step, the local entropy of each configuration being evaluated by the already mentioned BP with external field proportional to the considered configuration, and with proportionality constant that we call γ. This algorithm has been called *Entropy driven Monte Carlo* (EdMC).

Actually, the EdMC doesn't sample solutions from the *constrained* large deviation measure of eq. (10), whose space of configurations consists only in solutions, but from the *unconstrained* large deviation measure:

$$\mathcal{Z}(y,\gamma) = \sum_{\tilde{W}} e^{y \log \aleph(\tilde{W},\gamma)} \tag{11}$$

$$\aleph(\tilde{W},\gamma) = \sum_{W} \chi(W,\xi) e^{\gamma W \tilde{W}}$$

in which the $\tilde{W}$ are not required to be solutions and the constraint on the distance has been implemented in a soft way introducing the Legendre parameter γ. The local entropy $\mathcal{S}_{\mathcal{I}}(y,s)$ is thus related to the *local free entropy* $F(\tilde{W},\gamma) = \frac{1}{N}\log \aleph(\tilde{W},\gamma)$ by Legendre transform: the former is used for the analytical computation, while the latter is more suitable for the algorithm implementation.

For the unconstrained large deviation measure a 1RSB computation is required[14] and is possible: indeed, in the constrained case there were solutions that formed the center of clusters and this were up to α_{RSB} in the same connected region described by the RS solution; here the center of a cluster consists in a bunch of configurations that in general are not solutions (described by an on-diagonal Parisi parameter) and the overlap between clusters is now off-diagonal in the replica overlap matrix Q_{ab}. Even with the 1RSB assumption there is some problem: at finite y the external entropy is negative, but in the limit $y \to \infty$ one recovers meaningful results. In particular the $\tilde{W}$'s in the same cluster collapse to an unique configuration that is also a solution[15] (a core of the constrained large deviation analysis) and the local entropy curves seem exactly[16] those of the constrained reweighted analysis, see. Fig. 20 . In particular, the results seem especially consistent at $s \to 1$, as here the external entropy tends to zero. For the generalization problem similar considerations hold.

The EdMC maximizes the local free entropy $F(\tilde{W},\gamma)$ with a Metropolis-like move (we don't want to enter into the details here). A *scoping* procedure is required in order to make the algorithm effective: γ is gradually increased to enforce the local entropy maximization on always shorter scales, see Fig. 21. However, the really amazing fact is that the algorithm works well even at zero temperature : this suggests that while the energy landscape is glassy, the local entropy landscape has not metastable states and the optimization problem becomes convex-like, see Fig. 21. Correspondingly, the EdMC succeeds in a small number of iterations compared to energy based algorithms such as simulated annealing (SA), and while the number of iterations required by SA scales exponentially in N at fixed α, the EdMC scaling is polynomial in N, with an α-dependent exponent, see Fig. 22 . The main bottleneck of EdMC actually is the estimate of the local free entropy $F(\tilde{W},\gamma)$.

Similar performances were achived also in the 4-SAT problem, where dense clusters of solutions are known to exist up to a certain clause density. The

[14] a posteriori this is quite reasonable, see below; in practice it was understood from the fact that the RS external entropy was positive even above α_c.

[15] in [2] this follows from the large y scalings $m \to x/y$, $q_2 \to q_1 + \delta q/y$, see the paper for the notation

[16] the cores of extremely dense regions of solutions are expected to be themselves solutions, so one recover the constrained results

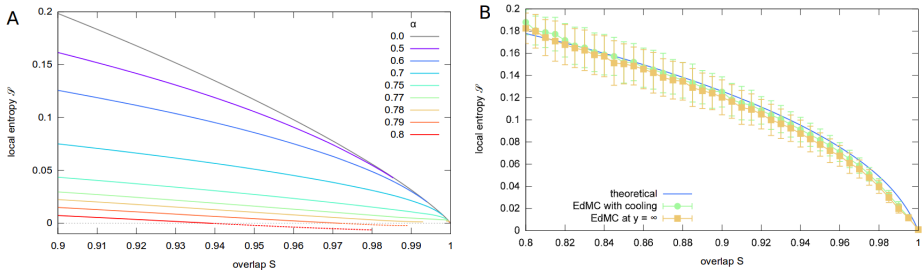

Figure 20: (A) Local entropy (theoretical computation) versus overlap for various training set sizes: apart from regions of the curves where 2RSB effects appear (gaps, negative entropies), the local entropy reaches the upper bound $\alpha = 0$ at sufficiently small distances. (B) Local entropy versus overlap at $\alpha = 0.6$, $N = 201$. The experimental points (averaged over 700 samples) represent BP estimates of the local entropy found using the EdMC procedure at fixed γ to maximize $F(\tilde{W}, \gamma)$ (i.e. without stopping when a solution was found); s and the local entropy $\mathcal{S}_{\mathcal{I}}(s)$ are estimated with BP at the final step. The remarkable fact is that both finite and zero temperature EdMC follow the same curve: this is consistent with the idea that there are not barriers in the local entropy landscape.

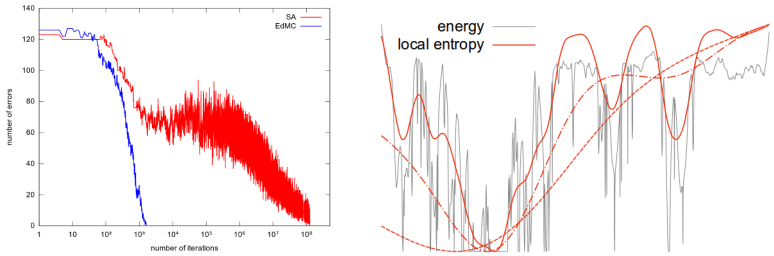

Figure 21: Left panel: a typical trajectory of standard (energetic) simulated annealing (red) and of EdMC (blue) in the plane (iterations, violated patterns) for the training of a discrete perceptron with $N = 801, \alpha = 0.3$. EdMC is run at zero temperature, so it is remarkable that the energy decreases nearly monotonically in time. Notice the logarithmic scale on the x-axis: EdMC requires few iterations with respect to SA. See [2] for the details of the algorithms. Right panel: sketch of the energy versus local entropy landscape and of the scoping procedure. The red dashed line corresponds to small γ and entropic exploration of large distances; increasing γ (dashed-dotted and solid lines) the algorithm searches shorter scales of distances. The plot represents the true local entropy of a toy 1D energetic landscape generated by a stochastic process.

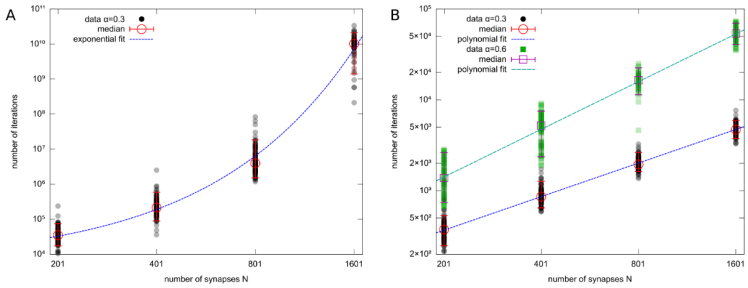

Figure 22: Number of iterations required to find a solution as function of the system size N for SA and EdMC, notice the log-log scale. 100 samples have been tested for each size. (A) Simulated annealing at $\alpha = 0.3$; the curve is fitted by $e^{a+bN}, b \simeq 0.0088$. (B) EdMC at $\alpha = 0.3$, fitted by aN^b, $b \simeq 1.2$ and $\alpha = 0.6$, fitted by aN^b, $b \simeq 1.7$.

main problem at $\alpha > \alpha_{RSB}$ is that the fragmentation of the connected region of solutions would require a RSB cavity method to estimate the local entropy.

2.2.2 Replicated algorithms

The effectiveness of EdMC is based on the non glassy nature of the local entropy landscape. In practice, the performances of EdMC are quite limited by the need of computing the local entropy at each iteration. On the other hand, the reinforcement algorithm is very simple and efficient. This imbalance has prompted the search for more efficient theoretically grounded algorithms.

Setting y integer, the large deviation partition function (11) can be seen as a system of y real replicas W_a [3] interacting with a "pivot" $\tilde{W}$:

$$\mathcal{Z}(y,\gamma) = \sum_{\tilde{W}} \aleph(\tilde{W},\gamma)^y = \sum_{W_a,\tilde{W}} \prod_{a=1}^{y} \chi(W_a,\xi) e^{\gamma \tilde{W} \sum_a W_a} \tag{12}$$

This approach is valid for general finite temperature systems, replacingχ with $e^{-\beta H(W_a)}$. Actually, the "pivot" may be traced out:

$$\mathcal{R}(y,\beta,\gamma) = \sum_{W_a} e^{-\beta \sum_a H(W_a) + \log \sum_{\tilde{W}} e^{\gamma \tilde{W} \sum_a W_a}}$$

The above partition function defines the *Robust Ensemble*. Configurations W_a (for an a) sampled from the Robust Ensemble are expected (see below) to follow the same distribution as the large deviation degree of freedom $\tilde{W}$. The term "robust" comes from the robustness of solutions belonging to dense clusters.

In practice, one can simply apply the traditional algorithms to the system (12), i.e. considering y systems and coupling them together. A scoping proce-

dure makes the replicas converge[17] to the pivot $\tilde{W}$, hence the equivalence with sampling from the large deviation measure. This simple strategy of replicating the system is not completely new [40], being beforehand motivated by a naive "entropic exploration" argument.

In [3] the three more obvious *replicated algorithms* are considered and applied to simple discrete neural networks:

- replicated Simulated Annealing
- replicated Gradient Descent (how to map the discrete synapsis problem to a differentiable one is non trivial)
- replicated Belief Propagation and its variations

All three show very good performances and scalings similar to EdMC and reinforcement, see [3] for details. In particular we spend a few words on replicated BP. The naive way of replicate BP leads to a replica symmetry form of the marginals at each site and doesn 't work well. The right approach is, once the problem has been replicated, to assume that messages do not depend on the replica index. In this way one has to deal with only one copy of the system and the BP equations for the pivot contain addends of the kind $y \times message$ where y can be taken real. The details will be explained in Chapter 4, where the approach will be applied to the continuous perceptron. Anyway, in [3] it is shown experimentally that this approach, called *focusing BP*, spontaneously breaks the replica symmetry. Indeed, the messages can be used to estimate the overlap between relevant[18] configurations (formally this is the overlap between real replicas before the focusing BP semplification and can be analytically extended in the passage to focusing BP). Such overlap is shown to be driven by the scoping procedure from the inter-cluster Parisi order parameter to the intra-cluster one, the Parisi overlaps being those predicted by the unconstrained large deviation theory. The problem of naive replicated BP is the same of BP, but for neglecting typical solutions and considering only configurations in dense regions, i.e. it estimates correctly the total unconstrained large deviation entropy but doesn' t focus on the single clusters. It is not completely clear from a theoretical standpoint why focusing BP manage to break this symmetry instead. Moreover, with a semplification focusing BP is identical to the reinforcement algorithm (having y replicas has somehow the effect of reinforcing the polarization of sites with greater magnetization).

In the end, we have shown that successful algorithms are attracted towards dense regions of solutions: indeed, not only the algorithmic solutions fit this description, but we have also gathered evidence that the local entropy maximization problem is convex-like at least until the clusters of solutions are in the

[17] actually, in simulations a solution is found just before the collapse

[18] typical equilibrium solutions is if they didn't exist, only configurations surrounded by many solutions are relevant.

same connected region. Finally, we have introduced a simple strategy to implement efficient algorithms and have proven that the previously known heuristics are nothing but clever ways of breaking the symmetry of replicated BP.

2.3 Discussion and perspectives

In the end we have shown that it is possible to solve efficiently the training of discrete neural networks problem. Efficient algorithms are driven towards dense regions of solutions, described by the large deviation measure (10). These regions shows also good generalization properties. The correspondence between effectiveness and dense clusters of solutions seems to be complete: the EdMC search proves that the local entropy lanscape is not glassy, replicating algorithms, i.e. sampling from the non-equilibium measure, provides effective algorithms, and finally all efficient existing algorithms are generated as variations of theoretically-grounded tecniques. On the other hand, when the dense region of solutions breaks down, all known algorithms cease to succeed.

This is not a totally new discovery, as this correspondence has been thought to exist in other random CSPs with discrete variables, such as K-SAT. Here it is known [9, 21] that the different levels of hardness of the problem occurs at clause densities corresponding to transitions in the space of solutions. However the above program has not been completed.

What remains to be done? Different lines of research stems naturally from such a clear and simple correspondence between local entropy and algorithmic accessibility :

- Extend the analysis to finite temperature, i.e. understand if even though perfect training may be not possible, there exist accessible regions of low energy.
- Try to apply this finite temperature approach to other optimization problems with discrete variables, such as energy minimization in SK and other spinglass models.
- Inquire if the large deviation analysis remains true when dealing with different learning problems, in particular when the input patterns are correlated and for unsupervised learning. This is a priority.
- Design very efficient algorithms for deep discrete neural networks
- Seen the robustness and accessibility of large deviation solutions, we can' t help thinking that the discussed mechanisms may play a role in the amazing and still mysterious way human brain processes information.
- In game theory it is known that in some model there exist "crystalline" optimal strategies at high risk versus optimal robust bunch of "mixed" strategies. In particular, our focus is on a problem from quantum control [41], in which it has been shown that dense regions of optimal strategies

exist. Interestingly, this family of strategies correspond to those found by humans asked to play this game (*gamefication* of optimization problems).

- The other priority is to investigate extentions to *continuous* deep neural networks: deep learning works amazingly, but a theoretical description is lacking.

The outlined goals are very ambitious and will require a lot of time, but we think they are worth it.

2.3.1 Goal of this work

The aim of this work is inquiring if the large deviation results extend to learning in simple *continuous* neural networks. In particular, our main concern is on the continuous perceptron with negative stabilities, the choice being driven by the following observations:

- already the discrete perceptron shows the clustering of solutions, so it seems natural to start the analysis with the simplest possible model
- nevertheless the space of solutions of the perceptron with positive or zero stability is convex; the negative perceptron, instead, has disconnected regions of solutions
- the negative perceptron has some interest in the packing of hard spheres context.

The original programme of the work was:

- equilibrium analysis à la Franz-Parisi, at positive and negaive κ in the classification scenario, with the hope to observe a qualitative change between the two cases
- reweighted analysis, at positive and negaive κ in the classification scenario, both using replicas or, if technically too hard, trying the focusing BP estimate of local entropy on single large samples
- show with gradient-descent based simulations that training a negative perceptron is hard and replicated GD is required
- understand if SGD works as the standard or replicated GD
- extend both the theoretical analysis and the simulations to the generalization scenario

All the analytical results have to be derived below the AT line.

The last three points are motivated by the observation that in the training of deep neural networks a simple GD either fails or yields solutions with bad generalization properties, while SGD works better, so we guessed the reason

was that it finds more robust solutions, and may be possibly explained by our off-equilibrium theory.

We briefly anticipate the results of the analysis:

- Franz-Parisi entropy: no qualitative difference between the positive and negative stability cases.
- reweighted analysis: we performed the replica computation and derived the saddle point equations. Its numerical solution is highly non-trivial, so we posticipated it (and finally gave it up, see below).
- focusing BP: it yields meaningful results for a certain range of distances, elsewhere it lacks of convergence (we don' t know if this is due to numerical problems or to a problem in the approach). We are not able to state that the local entropy it yields in the region of convergence is higher than the Franz-Parisi entropy.
- the really discouraging result is, though, that simple Gradient Descent succeds to finding solutions up to great α, well above the AT line. So there is no computational hardness. On the contrary, the question is whether there are local minima or all minima are solutions.
- nonetheless we tried replicated GD in order to understand if GD and replicated GD select the same solutions or different ones. We could not detect significant differences in the statistics of outcomes.

The conclusion is that the negative perceptron is not a good model to investigate non-equilibrium effects and computational hardness. We think that this may be due to the geometrical nature of the space of synapses, a sphere embedded in an euclidean space.

Our interest now is in deeper architectures: here the space of synaptic weights is a cartesian product of spheres, and even though different "domains" (corresponding to discrete states) may be equally acceptable considering only one sphere and tracing out the others, they may have long-range inter-sphere correlations, possibly with some domain combinations more robust than others.

Hence, in the last Chapter we will discuss in more detail our perspectives and hopes in the field of deep neural networks. In particular, the strategy of replicating algorithms is very flexible and similar heuristics have already been implemented in the literature with promising results [40].

3 Continuous perceptron: equilibrium analysis

In this Chapter we extend the equilibrium analysis to the continuous perceptron. Firstly, we delay a little more on the details of the Gardner analysis near the SAT-UNSAT line, then we turn to the analytical computation of the Franz-Parisi entropy by means of the replica method, solve the saddle point equations and discuss the results. The scenario is qualitatively different from the discrete case

3.1 Gardner analysis (reprise): critical region

In this Section we reconsider the Gardner analysis: in 1.2 we hinted at the way Gardner [7, 8] faced the continuous perceptron storage problem using the replica method, wrote down the RS saddle point equation and reported the phase diagram. Here we focus on the analytical derivation of the SAT-UNSAT line and other asymptotics behaviours from the saddle point equation; moreover we consider the problem of solving numerically the saddle point equation. The reason why repeat this work is to get some useful advice in the analytical and numerical treatment of the asymptotic limit of Gaussian integrals, so to be well trained when facing more difficult saddle point equations, as in Section 3.2.

3.1.1 Analysis of asymptotic behaviour of Gardner saddle point

The entropy of the problem is given by maximizing:

$$\phi(\alpha) = extr_{\tilde{q}}\{\frac{1}{2}\log(1-\tilde{q}) + \frac{1}{2}\frac{\tilde{q}}{1-\tilde{q}} + \alpha \int Dz \ \log H(\frac{\kappa - \sqrt{\tilde{q}}z}{\sqrt{1-\tilde{q}}})\} \tag{13}$$

As at the SAT-UNSAT transition we expect all the solutions to have shrunk to a small region, for α near $\alpha_c$we can expand for $\delta q = 1 - \tilde{q} \to 0$:

$$\phi(\alpha) = extr_{\delta q}\{\frac{1}{2}\log\delta q + \frac{1}{2}\frac{1}{\delta q} + \alpha \int_{-\infty}^{\kappa} Dz \ [-\frac{(\kappa - z)^2}{2\delta q} + \frac{1}{2}\log\delta q]\} + O(1) \tag{14}$$

Here we have exploited the series expansion:

$$H(x) \simeq \frac{1}{\sqrt{2\pi}x} e^{-\frac{x^2}{2}} (1 - \frac{1}{x^2} + \frac{3}{x^4} + ...) \ , \quad x \to \infty \tag{15}$$

which can be obtained from some integration by part. From $H(x) = 1 - H(-x)$ one gets the asymptotic beheaviour at $-\infty$.

Deriving (14) in δq one obtains the saddle point equation in this limit:

$$0 = \frac{1}{\delta q} - \frac{1}{\delta q^2} + \frac{\alpha}{\delta q^2}\int_{-\infty}^{\kappa} Dz \ (\kappa - z)^2 + \frac{\alpha}{\delta q}\int_{-\infty}^{\kappa} Dz + O(1)$$

For fixed $\alpha = \alpha_c - \delta\alpha$ the solution of this equation is, at the first order in $\delta\alpha$, something like $\delta q = C \cdot \delta\alpha + O(\delta\alpha^2)$[19]:

$$0 = \frac{1}{C^2\delta\alpha^2}\{\alpha_c \int_{-\infty}^{\kappa} Dz\ (\kappa - z)^2 - 1\} +$$

$$+\frac{1}{C \cdot \delta\alpha}\{-\frac{\int_{-\infty}^{\kappa} Dz\ (\kappa - z)^2}{C} + \alpha_c \int_{-\infty}^{\kappa} Dz + 1\} + O(1)$$

The elimination of the coefficient of $\delta\alpha^{-2}$ yields the critical storage

$$\alpha_c = \frac{1}{\int_{-\infty}^{\kappa} Dz\ (\kappa - z)^2}$$

while the term $\delta\alpha^{-1}$ provides a condition for C:

$$C = \frac{\alpha_c^{-1}}{1 + \alpha_c \int_{-\infty}^{\kappa} Dz}$$

For $\kappa = 0$: $\alpha_c = 2$, $C = 1/4$, so $1 - \tilde{q} = \frac{2-\alpha}{4}$. The terms of order $O(1)$ in the saddle point equation are eliminated by terms $O(\delta\alpha^2)$ in δq.

Plugging the asymptotic behaviours in (13), the entropy shows a logarithmic divergence as a function of the number of inputs at the SAT-UNSAT transition:

$$\phi(\alpha) = \frac{1}{2}\log\delta\alpha + \frac{\alpha_c \int_{-\infty}^{\kappa} Dz}{2}\log\delta\alpha + O(1)$$

In particular, $\phi(\alpha) = \log\delta\alpha$ for $\kappa = 0$. The scaling of both the oder parameter and the entropy with $\delta\alpha$ are consistent with those found numerically.

3.1.2 Numerics of asymptotic Gardner saddle point

Performing integrals in a naive way, numerical problems at $\tilde{q} \to 1$ emerge fiercely. We plotted the underlying entropy

$$\phi(\tilde{q}) = \frac{1}{2}\log(1 - \tilde{q}) + \frac{1}{2}\frac{\tilde{q}}{1 - \tilde{q}} + \alpha \int Dz\ \log H(\frac{\kappa - \sqrt{\tilde{q}}z}{\sqrt{1 - \tilde{q}}})$$

for $\kappa = 0., \alpha = 2.$ as a function of the order parameter and noticed that for $\tilde{q} \gtrsim 0.98$ there are clear numerical issues, see Fig. 23.

The problem probably comes from the numeric evaluation of the logarithm of a very small number (indeed the denominator of the argument of $\log H$ is $\sqrt{1-q}$. We try to fix this by means of a splitting of the energetic term:

[19] The term of order 2 actually enters the following equation, but with coefficient $\{\alpha_c \int_{-\infty}^{\kappa} Dz\ (\kappa - z)^2 - 1\}$, that will be set to 0.

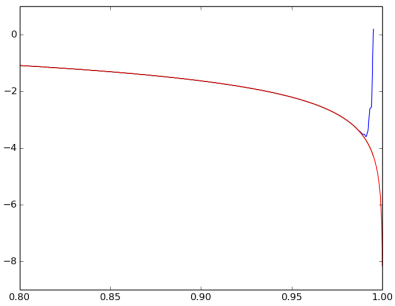

Figure 23: Underlying entropy at $\kappa = 0., \alpha = 2.$ as a function of the order parameter, with (red line) and without (blue) cutoff. The fact of taking exactly $\alpha = 2$ is not important: for $\alpha = 1.99$ the blue line looks the same, while the red one has a minimum very near to $\tilde{q} = 1$.

$$\begin{aligned}\tilde{G}_E \quad \simeq \quad & \int Dz \, \{\log H(\frac{\kappa - \sqrt{\tilde{q}}z}{\sqrt{1-\tilde{q}}}) \cdot \theta(\delta - \frac{\kappa - \sqrt{\tilde{q}}z}{\sqrt{1-\tilde{q}}}) + \\ & + [-\frac{1}{2}\left(\frac{\kappa - \sqrt{\tilde{q}}z}{\sqrt{1-\tilde{q}}}\right)^2 - \log\left(\sqrt{2\pi}\frac{\kappa - \sqrt{\tilde{q}}z}{\sqrt{1-\tilde{q}}}\right)] \cdot \theta(\frac{\kappa - \sqrt{\tilde{q}}z}{\sqrt{1-\tilde{q}}} - \delta)\}\end{aligned}$$

where in theory $\delta \gg 1$, in practice we take $\delta = 10$ (indeed $H(10) \approx 8 \cdot 10^{-24}$). This approximations seems to be insensitive of the cutoff for $\delta \geq 3$ and, as shown in Fig. 23 leads to a major improvement in the computation of Gaussian integrals in the critical region.

Solving the saddle point equation

Extremizing (13) yields the integral of $e^{-\frac{x^2}{2}}/H$, which can be treated with the expansion (15).

We tried to solve the saddle point equations with Newton method and properly tuning the algorithm parameters we managed to find reasonable solutions up to $\alpha \simeq 1.998$ with $\kappa = 0$.

Insead, using the bisection method (actually the function fzero from the Julia package Roots) we arrived beyond $\alpha = 1.99999$.

3.2 Franz-Parisi entropy

Now we will extend the equilibrium analysis to the continuous perceptron by computing the Franz-Parisi entropy. In paragraphs 3.2.1 and 3.2.2 we perform the replica computation in the RS ansatz. The small and large distances limit of the saddle point equations are considered in 3.2.3, where it is shown that a

large distance vertical asymptote exist for the Franz-Parisi entropy. Finally, the entropy is plotted and discussed in Section 3.2.4.

3.2.1 Replica method

The Franz-Parisi entropy is given by the log number of solutions at a fixed distance from typical solutions:

$$N\mathbb{H}_{FP}(d,\alpha) \equiv \langle \frac{1}{Z} \int d\mu(\tilde{W}_a)\ \chi_\xi(\tilde{W})\ log\aleph(\tilde{W},d)\rangle_\xi \tag{16}$$

where $d\mu(W) = \delta(W^2 - \bar{q}N)dW$, $Z = \int d\mu(W)\chi_\xi(W)$ and $\aleph(\tilde{W},d) = \int d\mu(W)\chi_\xi(W)\ \delta(W\cdot\tilde{W} - dN)$. The replica trick takes the form:

$$N\mathbb{H}_{FP}(d,\alpha) = \lim_{n\to 0, R\to 1} \partial_R \frac{1}{n} \langle \{ \int d\mu(\tilde{W}_a)\ \chi_\xi(\tilde{W})\ \aleph^{R-1}(\tilde{W}_a,d)\}^n\rangle_\xi$$

where the integration runs over the n $\tilde{W}_a$ variables and over the $n\times(R-1)$ W_{ar}'s.

As a first step we write the solution constraint $\chi_\xi$as $\chi_\xi(W) = \prod_\mu g(\frac{W\cdot\xi^\mu}{\sqrt{N}})$, $\mu = 1,...,\alpha N$ and we intoduce variables x_a^μ, x_{ar}^μthough Dirac deltas, which we expand introducing $\hat{x}_a^\mu, \hat{x}_{ar}^\mu$(we drop the tilde in the following):

$N\mathbb{H}_{FP} = \lim_{n\to 0,R\to 1} \partial_R \frac{1}{n} \big\langle \int d\mu(W_a) d\mu(W_{ar}) dx_a^\mu dx_{ar}^\mu d\hat{x}_a^\mu d\hat{x}_{ar}^\mu \times$
$\prod_{\mu a} g(x_a^\mu)\ \prod_{\mu ar} g(x_{ar}^\mu)\ \prod_{ar}\delta(W_a\cdot W_{ar} - dN)\times$
$\times\cdot exp\{i\sum_{\mu a}\hat{x}_a^\mu(\frac{W_a\cdot\xi^\mu}{\sqrt{N}} - x_a^\mu) + i\sum_{\mu ar}\hat{x}_{ar}^\mu(\frac{W_{ar}\cdot\xi^\mu}{\sqrt{N}} - x_{ar}^\mu)\big\rangle_\xi$

Here and in what follows we neglect multiplicative constants such as $(2\pi)^{-1}$ because in the limit $n\to 0$ they become additive constants. In the limit $N\to\infty$ it is possible to perform the quenched average$\Big\langle exp\{i\frac{W_i\xi_i}{\sqrt{N}}\}\Big\rangle_\xi = 1 - \frac{1}{2N}W_i^2$:

$N\mathbb{H}_{FP} = \lim_{n\to 0,R\to 1}\partial_R\frac{1}{n}\Big[\int d\mu(W_a)d\mu(W_{ar})dx_a^\mu dx_{ar}^\mu d\hat{x}_a^\mu d\hat{x}_{ar}^\mu\times$
$\times\prod_{\mu a} g(x_a^\mu)\ \prod_{\mu ar} g(x_{ar}^\mu)\ \prod_{ar}\delta(W_a\cdot W_{ar} - dN)\cdot exp\{-i\sum_{\mu a}\hat{x}_a^\mu x_a^\mu -$
$-i\sum_{\mu ar}\hat{x}_{ar}^\mu x_{ar}^\mu - \frac{1}{2N}\sum_{\mu i}(\sum_a \hat{x}_a^\mu W_a^i + \sum_{ar}\hat{x}_{ar}^\mu W_{ar}^i)^2\}\Big] =$
$= \lim_{n\to 0,R\to 1}\partial_R\frac{1}{n}\Big[\int d\mu(W_a)d\mu(W_{ar})dx_a^\mu dx_{ar}^\mu d\hat{x}_a^\mu d\hat{x}_{ar}^\mu\ \prod_{\mu a} g(x_a^\mu)\times$
$\times\prod_{\mu ar} g(x_{ar}^\mu)\ \prod_{ar}\delta(W_a\cdot W_{ar} - dN)\, exp\{-i\sum_{\mu a}\hat{x}_a^\mu x_a^\mu - i\sum_{\mu ar}\hat{x}_{ar}^\mu x_{ar}^\mu\}$
$\times exp\{-\frac{1}{2N}\sum_\mu(\sum_{ab}\hat{x}_a^\mu\hat{x}_b^\mu W_a\cdot W_b + \sum_{arbs}\hat{x}_{ar}^\mu\hat{x}_{bs}^\mu W_{ar}\cdot W_{bs} +$
$+2\sum_{abr}\hat{x}_{ar}^\mu\hat{x}_b^\mu W_{ar}\cdot W_b)\}]$

Notice that up to this point the computation holds equally well for both the discrete and the continuous perceptron.

We introduce the order parameters (some are trivially fixed by the deltas but we deal with all $(nR)^2$ of them):

$$Q_{ab} = \frac{W_a\cdot W_b}{N}$$

$$Q_{ar,b} = \frac{W_{ar}\cdot W_b}{N} \text{ and } Q_{a,br} = \frac{W_a\cdot W_{br}}{N}$$

$$Q_{ar,bs} = \frac{W_{ar} \cdot W_{bs}}{N}$$

through Delta functions $\delta(Q - W \cdot W) = \int d\hat{Q} e^{N\hat{Q}(Q - W \cdot W)}$that we resolve with specular order parameters $\hat{Q}$. In this way we decouple the sums/products over $i = 1, ..., N$ and $\mu = 1, ...\alpha N$ and we get:

$$N\mathbb{H}_{FP} = \lim_{n \to 0, R \to 1} \partial_R \frac{1}{n} \int dQ_{ab} d\hat{Q}_{ab} dQ_{ar,b} d\hat{Q}_{ar,b} dQ_{a,bs} d\hat{Q}_{a,bs} dQ_{ar,bs} d\hat{Q}_{ar,bs} \times$$
$$\times e^{N\left[\sum_{ab} Q_{ab}\hat{Q}_{ab} + \sum_{abr} Q_{ar,b}\hat{Q}_{ar,b} + \sum_{abr} Q_{a,br}\hat{Q}_{a,br} + \sum_{abrs} Q_{ar,bs}\hat{Q}_{ar,bs}\right]} \times G_E^{\alpha N} G_S^N \tag{17}$$

$$G_E = \int dx_a d\hat{x}_a dx_{ar} d\hat{x}_{ar} \prod_a g(x_a) \prod_{ar} g(x_{ar})\, exp{-i} \left\{ \sum_a x_a \hat{x}_a + \sum_{ar} x_{ar} \hat{x}_{ar} \right\} \times$$
$$exp - \frac{1}{2} \left[\sum_{ab} \hat{x}_a \hat{x}_b Q_{ab} + 2 \sum_{abr} \hat{x}_{ar} \hat{x}_b Q_{ar,b} + \sum_{abrs} \hat{x}_{ar} \hat{x}_{bs} Q_{ar,bs} \right] \tag{18}$$

$$G_S = \int dW_a dW_{ar} exp\{ - \sum_{ab} \hat{Q}_{ab} W_a W_b - \sum_{abr} \hat{Q}_{ar,b} W_{ar} W_b -$$

$$- \sum_{abr} \hat{Q}_{a,br} W_a W_{br} - \sum_{abrs} \hat{Q}_{ar,bs} W_{ar} W_{bs} \}$$

In the first of the above expressions there are three implied Delta that fix $Q_{aa} = Q_{ar,ar} = \bar{q}$ and $Q_{ar,b} = d$. In the last one the integration dW is one-dimensional, and not N-dimensional as in all other previous equations.

Now we proceed to the limit $N \to 0$ so that we can use the saddle point method.

A first ansatz is that $Q_{ar,b} = Q_{b,ar}$ and the same for the $\hat{Q}$

Calling:

$$Q = \begin{pmatrix} Q_{ab} & Q_{ar,b}^T \\ Q_{ar,b} & Q_{ar,bs} \end{pmatrix} \tag{19}$$

we notice that $G_S$is the gaussian integral of matrix $2\hat{Q}$, i.e. (leaving out constants):

$$G_S^N = exp\{ - \frac{N}{2} \log\det 2\hat{Q} \}$$

The $\hat{Q}$'s are determined with the usual trick for the derivation of $\log\det$ [36], $\frac{\partial}{\partial M_{ab}} \log\det M = (M^{-1})_{ab}$, to yield:

$$\hat{Q} = \frac{1}{2} Q^{-1}$$

As a result, the exponential with the $Q\hat{Q}$-like sums yields a multiplicative constant which can be neglected as becomes an additive constant when $n \to 0$, and so:

$$N\mathbb{H}_{FP} = \lim_{n\to 0, R\to 1} \partial_R \frac{1}{n} e^{N\{extr_Q \frac{1}{2} logDetQ + \alpha \log G_E\}}$$

Using the fact that $\frac{1}{2} logDetQ = (n),\ G_E = 1 + nG'_E + O(n^2)$:

$$\mathbb{H}_{FP} = \lim_{R\to 1} \partial_R extr_Q \{\lim_{n\to 0} \frac{1}{n} \frac{1}{2} logDetQ\} + \alpha G'_E$$

(notice that here the term 1 that would yield $1/n$ is eaten not by subtracting 1 but by deriving R, see below)

3.2.2 RS ansatz

We start considering an RS ansatz for the Parisi matrix Q_{ab}. In general one begins with the RS case to see where are the problems; moreover some information can be obtained already at the RS level. The RS saddle point equations gives also an idea of the complexity of solving the equations for higher level of RSB. Here stands the difficulty of further RSB steps; typically at each step a nested integral is added. So we take an RS ansatz for the reference (equilibrium) configurations; this yields a formally 1RSB form for the non reference block of the Parisi matrix:

$$Q_{aa} = \bar{q}, \quad Q_{ab} = q_0$$

$$Q_{ar,a} = d, \quad Q_{ar,b} = d_0$$

$$Q_{ar,ar} = \bar{q}, \quad Q_{ar,as} = p_1, \quad Q_{ar,bs} = p_0$$

with $p_0 < p_1 < \bar{q}$.
It will be useful in the following that:

$$Q^{-1}_{aa} = \frac{\bar{q} + (n-2)q_0}{(\bar{q} - q_0)(\bar{q} + (n-1)q_0)}$$

$$Q^{-1}_{ab} = \frac{-q}{(\bar{q} - q_0)(\bar{q} + (n-1)q_0)}$$

In the expressions above $\bar{q}$ and d are given and q_0 is the same one previously found as saddle point of the free energy. The variational parameters are d_0, p_0, p_1.

$$\mathbb{H}_{FP} = extr_{d_0,p_0,p_1} \lim_{R\to 1} \partial_R \{\lim_{n\to 0} \frac{1}{n} \frac{1}{2} logDetQ + \alpha G'_E\}$$

Entropic term

Exploiting $detQ = detQ_{ab} \times \det(Q_{ar,bs} - Q_{ar,b}Q_{ab}^{-1}Q_{ar,b}^{T})$ [20], this reduces to computing the determinant of then$n(R-1) \times n(R-1)$ 1RSB-like Parisi matrix:

$Q'_{ar,ar} = \bar{q}' = \bar{q} - \gamma$, $Q_{ar,as} = q'_1 = p_1 - \gamma$, $Q_{ar,bs} = q'_0 = p_0 - \gamma_0 \rightarrow \frac{1}{2} logDetQ =$

$= \frac{1}{2}\{\log[\bar{q}' + (R-2)q'_1 + (n-1)(R-1)q'_0] + n(R-2)\log[\bar{q}' - q'_1] + (n-1)\log[\bar{q}' + (R-2)q'_1 - (R-1)q'_0]\}$

$$\lim_{n\to 0} \frac{1}{2n} logDetQ = \frac{1}{2}\left[\frac{(R-1)q'_0}{\bar{q}' - q'_1 + (R-1)(q'_1 - q'_0)} + (R-1)\log[\bar{q}' - q'_1] + \right.$$

$$\left. + \log[1 + \frac{(R-1)(q'_1 - q'_0)}{\bar{q}' - q'_1}]\right] =$$

$$= \frac{1}{2}\left[(R-1)\frac{p_0 - \gamma_0}{\bar{q} - p_1 + (R-1)(p_1 - p_0 - (\gamma - \gamma_0))} + \right.$$

$$\left.(R-1)\log[\bar{q} - p_1] + \log[1 + \frac{(R-1)(p_1 - p_0 - (\gamma - \gamma_0))}{\bar{q} - p_1}]\right] =$$

$$\lim_{n\to 0} \frac{1}{2n} logDetQ = \frac{1}{2}\{log(\bar{q} - q_0) + \frac{q_0}{\bar{q} - q_0} + (R-2)\log[\bar{q} - p_1] +$$

$$+ \log[\bar{q} - p_1 + (R-1)(p_1 - p_0 - (\gamma - \gamma_0))] +$$

$$+ (R-1)\frac{p_0 - \gamma_0}{\bar{q} - p_1 + (R-1)(p_1 - p_0 - (\gamma - \gamma_0))}\}$$

where:

$\gamma - \gamma_0 = \frac{d - d_0}{(\bar{q} - q_0)^2}(\bar{q}d + 3\bar{q}d_0 - q_0 d_0 - 3q_0 d)$

$\gamma_0 = -\frac{d - d_0}{(\bar{q} - q_0)^2}(q_0(d + d_0) - 2q_0 d_0)$

Deriving and doing the limit with respect to R:

$$\lim_{R\to 1} \partial_R \lim_{n\to 0} \frac{1}{n}\frac{1}{2} logDetQ = \frac{1}{2}\log(\bar{q} - p_1) + \frac{1}{2}\frac{p_1 - \gamma}{\bar{q} - p_1}$$

where $\gamma = \frac{d - d_0}{(\bar{q} - q_0)^2}(\bar{q}(d + d_0) - 2q_0 d)$

---

[20] proof: Consider the block matrix $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$. If A is invertible then $\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} A & 0 \\ C & 1 \end{pmatrix}\begin{pmatrix} 1 & A^{-1}B \\ 0 & D - CA^{-1}B \end{pmatrix}$. The result follows from Binet theorem and the observation $\det \begin{pmatrix} A & B \\ 0 & C \end{pmatrix} = \det A \det C$

Energetic term

Plugging the RS ansatz in G_E:

$$G_E = \int dx_a d\hat{x}_a dx_{ar} d\hat{x}_{ar} \prod_a g(x_a) \prod_{ar} g(x_{ar}) \times$$

$$exp - i \left\{ \sum_a x_a \hat{x}_a + \sum_{ar} x_{ar} \hat{x}_{ar} \right\} exp - \frac{1}{2} \Bigg[(\bar{q} - q_0) \sum_a \hat{x}_a^2 + q_0 (\sum_a \hat{x}_a)^2 +$$

$$+2(d - d_0) \sum_{ar} \hat{x}_{ar} \hat{x}_a + 2d_0 (\sum_b \hat{x}_b)(\sum_{ar} \hat{x}_{ar}) + (\bar{q} - p_1) \sum_{ar} \hat{x}_{ar}^2 +$$

$$+ (p_1 - p_0) \sum_a (\sum_r \hat{x}_{ar})^2 + p_0 (\sum_{ar} \hat{x}_{ar})^2 \Bigg] \tag{20}$$

We want to decouple the n a-index replicas. The terms that couple different replicas a, b are $q_0 (\sum_a x_a)^2 + 2d_0 (\sum_a x_a)(\sum_{ab} x_{ab}) + p_0 (\sum_{ab} x_{ab})^2 = q_0 u^2 + 2d_0 uv + p_0 v^2 = \lambda_1 \tilde{u}^2 + \lambda_2 \tilde{v}^2$

where $u = \sum_a x_a, v = \sum_{ab} x_{ab}and\tilde{u}, \tilde{v} = \alpha_\pm u + \beta_\pm v$. We suppose $\lambda_1, \lambda_2 > 0$ cioè $q_0 p_0 - d_0^2 > 0$.

Now we can H-S transform with respect to $\tilde{u}^2, \tilde{v}^2$producing the integrals in z_1, z_2:

$$G_E = \int Dz_1 Dz_2 dx_a d\hat{x}_a dx_{ar} d\hat{x}_{ar} \prod_a g(x_a) \prod_{ar} g(x_{ar}) \times$$

$$exp - i \left\{ \sum_a x_a \hat{x}_a + \sum_{ar} x_{ar} \hat{x}_{ar} \right\} exp - \frac{1}{2} \Bigg[2i\sqrt{\lambda_1} (\alpha_+ \sum_a \hat{x}_a + \beta_+ \sum_{ar} \hat{x}_{ar}) z_1 +$$

$$+2i\sqrt{\lambda_2} (\alpha_- \sum_a \hat{x}_a + \beta_- \sum_{ar} \hat{x}_{ar}) z_2 + (\bar{q} - q_0) \sum_a \hat{x}_a^2 + 2(d - d_0) \sum_{ar} \hat{x}_{ar} \hat{x}_a +$$

$$+(\bar{q} - p_1) \sum_{ar} \hat{x}_{ar}^2 + (p_1 - p_0) \sum_a (\sum_r \hat{x}_{ar})^2 \Bigg] \tag{21}$$

Now the $\sum_a$ in $\int Dz_1 Dz_2$ factorizes:

$$G_E = \int Dz_1 Dz_2 \Bigg[\int dx d\hat{x} dx_r d\hat{x}_r g(x) \prod_r g(x_r) exp - i\{x\hat{x} + \sum_r x_r \hat{x}_r\}$$

$$exp - \frac{1}{2} \{ (\bar{q} - q_0) \hat{x}^2 + 2(d - d_0) \sum_r \hat{x}_r \hat{x} + (\bar{q} - p_1) \sum_r \hat{x}_r^2 + (p_1 - p_0) (\sum_r \hat{x}_r)^2 \}$$

$$exp - \{ i(\sqrt{\lambda_1} \alpha_+ z_1 + \sqrt{\lambda_2} \alpha_- z_2) \hat{x} + i(\sqrt{\lambda_1} \beta_+ z_1 + \sqrt{\lambda_2} \beta_- z_2) \sum_r \hat{x}_r \} \Bigg]^n \tag{22}$$

We rewrite this with the notation:
$A(z_1, z_2) = \sqrt{\lambda_1}\alpha_+ z_1 + \sqrt{\lambda_2}\alpha_- z_2$
$B(z_1, z_2) = \sqrt{\lambda_1}\beta_+ z_1 + \sqrt{\lambda_2}\beta_- z_2$

$$G_E = \int Dz_1 Dz_2 \left[\int dx d\hat{x} dx_r d\hat{x}_r g(x) \prod_r g(x_r) exp - i\{x\hat{x} + \sum_r x_r \hat{x}_r\} \right.$$
$$exp - \frac{1}{2}\{(\bar{q} - q_0)\hat{x}^2 + 2i(-i)(d - d_0)\sum_r \hat{x}_r \hat{x} + (\bar{q} - p_1)\sum_r \hat{x}_r^2 + (p_1 - p_0)(\sum_r \hat{x}_r)^2\}$$
$$\left. exp - \{iA\hat{x} + iB\sum_r \hat{x}_r\} \right]^n \quad (23)$$

We now integrate in $d\hat{x}$, that is kind of Fourier-transforming a Gaussian:

$$G_E = \int Dz_1 Dz_2 \left[\int \frac{dx}{\sqrt{\bar{q} - q_0}} dx_r d\hat{x}_r g(x) \prod_r g(x_r) exp - i\{\sum_r x_r \hat{x}_r\} \right.$$
$$exp - \frac{1}{2}\{(\frac{(x + A - i(d - d_0)\sum_r \hat{x}_r)^2}{\bar{q} - q_0} + (\bar{q} - p_1)\sum_r \hat{x}_r^2\} \times$$
$$\left. \times exp - \frac{1}{2}\{(p_1 - p_0)(\sum_r \hat{x}_r)^2 + iB\sum_r \hat{x}_r\} \right]^n = \quad (24)$$

$$= \int Dz_1 Dz_2 \left[\int \frac{dx}{\sqrt{\bar{q} - q_0}} dx_r d\hat{x}_r g(x) \prod_r g(x_r) exp - i\{\sum_r x_r \hat{x}_r\} \right.$$
$$exp - \frac{(x + A)^2 - 2i(d - d_0)(x + A)\sum_r \hat{x}_r - (d - d_0)^2(\sum_r \hat{x}_r)^2}{2(\bar{q} - q_0)}\} \times$$
$$\left. \times exp - \frac{1}{2}\{(p_1 - p_0)(\sum_r \hat{x}_r)^2 + 2iB\sum_r \hat{x}_r + (\bar{q} - p_1)\sum_r \hat{x}_r^2\} \right]^n \quad (25)$$

We now Hubbard-Stratonovich transform the term with $(\sum_r \hat{x}_r)^2$:

$$G_E = \int Dz_1 Dz_2 \left[\int Dw \frac{dx}{\sqrt{\bar{q} - q_0}} dx_r d\hat{x}_r g(x) \prod_r g(x_r) exp - i\{\sum_r x_r \hat{x}_r\} \right.$$
$$exp - \frac{(x + A)^2 - 2i(d - d_0)(x + A)\sum_r \hat{x}_r}{2(\bar{q} - q_0)} + (\bar{q} - p_1)\sum_r \hat{x}_r^2\} \times$$
$$\left. \times exp\{i\sqrt{p_1 - p_0 - \frac{(d - d_0)^2}{(\bar{q} - q_0)}}(\sum_r \hat{x}_r)w + -iB\sum_r \hat{x}_r\} \right]^n \quad (26)$$

so to factorize the r index:

$$G_E = \int Dz_1 Dz_2 \left[\int Dw \frac{dx}{\sqrt{\bar{q}-q_0}} g(x) exp - \frac{(x+A)^2}{2(\bar{q}-q_0)} \right.$$
$$\left\{ \int dy d\hat{y} \cdot g(y) \cdot exp\{ -\frac{(\bar{q}-p_1)}{2}\hat{y}^2 - iy\hat{y} + \frac{i(d-d_0)(x+A)}{(\bar{q}-q_0)}\hat{y}\} \times \right.$$
$$\left. \left. \times exp\{ i\sqrt{p_1 - p_0 - \frac{(d-d_0)^2}{(\bar{q}-q_0)}} w\hat{y} + -iB\hat{y}\} \right\}^{R-1} \right]^n = \quad (27)$$

(in this passage we integrate in $d\hat{y}$ and shift x of A and rescale it by $\sqrt{\bar{q}-q_0}$)

$$= \int Dz_1 Dz_2 \left[\int Dw Dx g(\sqrt{\bar{q}-q_0}x - A) \right.$$
$$\left. \left\{ \int \frac{dy}{\sqrt{\bar{q}-p_1}} \cdot g(y) \cdot exp\{ -\frac{(y + B - \frac{d-d_0}{\sqrt{\bar{q}-q_0}}x - \sqrt{p_1 - p_0 - \frac{(d-d_0)^2}{(\bar{q}-q_0)}}w)^2}{2(\bar{q}-p_1)} \right\}^{R-1} \right]^n$$

Shift and rescale the y:

$$= \int Dz_1 Dz_2 \left[\int Dw Dx g(\sqrt{\bar{q}-q_0}x - A) \right.$$
$$\left. \left\{ \int Dy \cdot g(\sqrt{\bar{q}-p_1}y - B + \frac{d-d_0}{\sqrt{\bar{q}-q_0}}x + \sqrt{p_1 - p_0 - \frac{(d-d_0)^2}{(\bar{q}-q_0)}}w) \right\}^{R-1} \right]^n$$

This is something like $G_E = 1 + nG'_E + ...$ with

$$G'_E = \int Dz_1 Dz_2 \log \left[\int Dw Dx g(\sqrt{\bar{q}-q_0}x - A) \right.$$
$$\left. \left\{ \int Dy \cdot g(\sqrt{\bar{q}-p_1}y - B + \frac{d-d_0}{\sqrt{\bar{q}-q_0}}x + \sqrt{p_1 - p_0 - \frac{(d-d_0)^2}{(\bar{q}-q_0)}}w) \right\}^{R-1} \right]$$

The final additive contribution to the potential is $\tilde{G}_E = \lim_{R\to 1} \partial_R G'_E$ (we can exchange simultaneously the signs of A, B):

$$\tilde{G}_E = \int \frac{Dz_1 Dz_2}{\int Dx \ g(\sqrt{\bar{q}-q_0}x + A(z_1,z_2))} \int Dw Dx \ g(\sqrt{\bar{q}-q_0}x + A(z_1,z_2)) \times$$
$$\times \log \int Dy \ g \left[\sqrt{\bar{q}-p_1}y + B(z_1,z_2) + \sqrt{p_1 - p_0 - \frac{(d-d_0)^2}{\bar{q}-q_0}}w + \frac{d-d_0}{\sqrt{\bar{q}-q_0}}x \right] \quad (28)$$

In the end:

$$V_{F-P} = extr_{d_0,p_0,p_1} \frac{1}{2}\log(\bar{q}-p_1) + \frac{1}{2}\frac{p_1-\gamma}{\bar{q}-p_1} + \alpha\tilde{G}_E$$

If $g(\cdot) = \theta(\cdot - \kappa)$ the expression for $\tilde{G}$ simplifies to:

$$\tilde{G}_E = \int \frac{Dz_1Dz_2}{H(\frac{\kappa - A(z_1,z_2)}{\sqrt{\bar{q}-q_0}})} \int DwDx\ \theta(x + \frac{A(z_1,z_2)-\kappa}{\sqrt{\bar{q}-q_0}})\times$$

$$\times \log\ H\left(\frac{\kappa - B(z_1,z_2) - \sqrt{p_1 - p_0 - \frac{(d-d_0)^2}{\bar{q}-q_0}}w - \frac{d-d_0}{\sqrt{\bar{q}-q_0}}x}{\sqrt{\bar{q}-p_1}}\right)$$

This expression can be further simplificated with some change of variables and exploiting the properties of Gaussian integrals. For more clarity we introduce intermediate constants:

$$\tilde{G}_E = \int \frac{Dz_1Dz_2}{H(\kappa' - A'(z_1,z_2))} \int DwDx\ \theta(x - \kappa' + A'(z_1,z_2))\times$$

$$\times \log\ H\left(\kappa'' - B'(z_1,z_2) - Cw - C'x\right)$$

Now we rotate w and x:
$x' = \frac{C'}{\sqrt{C^2+C'^2}}x + \frac{C}{\sqrt{C^2+C'^2}}w, x = \frac{C'}{\sqrt{C^2+C'^2}}x' - \frac{C}{\sqrt{C^2+C'^2}}w'$
and send w in -w, so to get

$$\tilde{G}_E = \int \frac{Dz_1Dz_2}{H(\kappa' - A'(z_1,z_2))} \int Dx\ H(\frac{\sqrt{C^2+C'^2}}{C}\{\kappa' - A'\} - \frac{C'}{C}x)\times$$

$$\times \log\ H\left(\kappa'' - B'(z_1,z_2) - \sqrt{C^2+C'^2}x\right)$$

With another rotation of variables: $z_A = \frac{A(z_1,z_2)}{\sqrt{\lambda_1\alpha_+^2+\lambda_2\alpha_-^2}} = \cos\theta z_1 + \sin\theta z_2$, $z_B = \cos\theta z_2 - \sin\theta z_1$:

$$\tilde{G}_E = \int \frac{Dz_ADz_B}{H(\kappa' - az_A)} \int Dx\ H(\frac{\sqrt{C^2+C'^2}}{C}\{\kappa' - az_A\} - \frac{C'}{C}x)\times$$

$$\times \log\ H\left(\kappa'' - b_1\cos\theta z_A + b_1\sin\theta z_B - b_2\cos\theta z_B - b_2\sin\theta z_A - \sqrt{C^2+C'^2}x\right)$$

with $a = \frac{\sqrt{\lambda_1\alpha_+^2+\lambda_2\alpha_-^2}}{\sqrt{\bar{q}-q_0}}, b_1 = \frac{\sqrt{\lambda_1}\beta_+}{\sqrt{\bar{q}-p_1}}, b_2 = \frac{\sqrt{\lambda_2}\beta_-}{\sqrt{\bar{q}-p_1}}$
Collecting the z's yields:

$$\tilde{G}_E = \int \frac{Dz_ADz_B}{H(\kappa' - az_A)} \int Dx\ H(\frac{\sqrt{C^2+C'^2}}{C}\{\kappa' - az_A\} - \frac{C'}{C}x)\times$$

$$\times \log\ H\left(\kappa'' - (b_1\cos\theta + b_2\sin\theta)z_A + (b_1\sin\theta - b_2\cos\theta)z_B - \sqrt{C^2 + C'^2}x\right)$$

We now change variables with the rotation: $x' = \cos\psi x - \sin\psi z_B, x = \cos\psi x' + \sin\psi z'_B)$
with $\sin\psi = \frac{(b_1\sin\theta - b_2\cos\theta)}{\sqrt{(b_1\sin\theta - b_2\cos\theta)^2 + C^2 + C'^2}}, \cos\psi = \frac{\sqrt{C^2+C'^2}}{\sqrt{(b_1\sin\theta - b_2\cos\theta)^2 + C^2 + C'^2}}$:

$$\tilde{G}_E = \int \frac{Dz_A Dx}{H(\kappa' - az_A)} \times$$

$$\times \log\ H\left(\kappa'' - (b_1\cos\theta + b_2\sin\theta)z_A - \sqrt{C^2 + C'^2 + (b_1\sin\theta - b_2\cos\theta)^2}x\right) \times$$

$$\times \int Dz_B\ H(\frac{\sqrt{C^2 + C'^2}}{C}\{\kappa' - az_A\} - \frac{C'}{C}\cos\psi x - \frac{C'}{C}\sin\psi z_B) \times$$

Inverting x:

$$\tilde{G}_E = \int \frac{Dz_A Dx}{H(\kappa' - az_A)} \times$$

$$\times \log\ H\left(\kappa'' - (b_1\cos\theta + b_2\sin\theta)z_A + \sqrt{C^2 + C'^2 + (b_1\sin\theta - b_2\cos\theta)^2}x\right) \times$$

$$\times \int Dz_B\ H(\frac{\sqrt{C^2 + C'^2}}{C}\{\kappa' - az_A\} + \frac{C'}{C}\cos\psi x - \frac{C'}{C}\sin\psi z_B) \times$$

Finally using $\int Dz\, H(\frac{Az+B}{C}) = H(\frac{B}{\sqrt{A^2+C^2}})$ the above expression becomes :

$$\tilde{G}_E = \int \frac{Dz_A Dx}{H(\kappa' - az_A)}\ H(\frac{\frac{\sqrt{C^2+C'^2}}{C}\{\kappa' - az_A\} + \frac{C'}{C}\cos\psi x}{\sqrt{1 + (\frac{C'}{C}\sin\psi)^2}}) \times$$

$$\times \log\ H\left(\kappa'' - (b_1\cos\theta + b_2\sin\theta)z_A + \sqrt{C^2 + C'^2 + (b_1\sin\theta - b_2\cos\theta)^2}x\right)$$

Exploiting the definitions of α, β, λ
$\begin{pmatrix} q_0 & d_0 \\ d_0 & p_0 \end{pmatrix} = \begin{pmatrix} \alpha_+ & \alpha_- \\ \beta_+ & \beta_- \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \alpha_+ & \beta_+ \\ \alpha_- & \beta_- \end{pmatrix} =$
$= \begin{pmatrix} \lambda_1\alpha_+^2 + \lambda_2\alpha_-^2 & \lambda_1\alpha_+\beta_+ + \lambda_2\alpha_-\beta_- \\ \lambda_1\alpha_+\beta_+ + \lambda_2\alpha_-\beta_- & \lambda_1\beta_+^2 + \lambda_2\beta_-^2 \end{pmatrix}$ one gets
$b_1\cos\theta + b_2\sin\theta = \frac{d_0}{\sqrt{q_0}\sqrt{q - p_1}}$
$b_1\sin\theta - b_2\cos\theta = -\frac{\sqrt{q_0 p_0 - d_0^2}}{\sqrt{q_0}\sqrt{q - p_1}}$

$a = \frac{\sqrt{q_0}}{\sqrt{\bar{q}-q_0}}$

$C = \frac{\sqrt{p_1 - p_0 - \frac{(d-d_0)^2}{\bar{q}-q_0}}}{\sqrt{\bar{q}-p_1}}$

$C' = \frac{d-d_0}{\sqrt{\bar{q}-q_0}\sqrt{\bar{q}-p_1}}$

$\sqrt{C^2+C'^2} = \frac{\sqrt{p_1-p_0}}{\sqrt{\bar{q}-p_1}}$

$\sqrt{(b_1\sin\theta - b_2\cos\theta)^2 + C^2 + C'^2} = \frac{\sqrt{q_0 p_1 - d_0^2}}{\sqrt{q_0}\sqrt{\bar{q}-p_1}}$

$\sin\psi = -\frac{\sqrt{q_0 p_0 - d_0^2}}{\sqrt{q_0 p_1 - d_0^2}}$, $\cos\psi = \frac{\sqrt{C^2+C'^2}}{\sqrt{(b_1\sin\theta - b_2\cos\theta)^2 + C^2 + C'^2}} = \frac{\sqrt{q_0}\sqrt{p_1-p_0}}{\sqrt{q_0 p_1 - d_0^2}}$

$\kappa' = \frac{\kappa}{\sqrt{\bar{q}-q_0}}$, $\kappa'' = \frac{\kappa}{\sqrt{\bar{q}-p_1}}$

$\frac{C'}{\sqrt{(b_1\sin\theta - b_2\cos\theta)^2 + C^2 + C'^2}} = \frac{(d-d_0)\sqrt{q_0}}{\sqrt{\bar{q}-q_0}\sqrt{q_0 p_1 - d_0^2}}$

$\frac{\frac{\sqrt{C^2+C'^2}}{C}\{\kappa' - a z_A\} + \frac{C'}{C}\cos\psi x}{\sqrt{1+(\frac{C'}{C}\sin\psi)^2}} = \frac{1}{\sqrt{\bar{q}-q_0}} \frac{\kappa - \sqrt{q_0} z_A + \frac{C'}{\sqrt{C^2+C'^2}}\cos\psi x \sqrt{\bar{q}-q_0}}{\sqrt{\frac{C^2+C'^2\sin^2\psi}{C^2+C'^2}}} =$

$= \frac{1}{\sqrt{\bar{q}-q_0}} \frac{\kappa - \sqrt{q_0} z_A + \frac{C'\sqrt{\bar{q}-q_0}}{\sqrt{(b_1\sin\theta - b_2\cos\theta)^2 + C^2 + C'^2}} x}{\sqrt{1 - \frac{C'^2\cos^2\psi}{C^2+C'^2}}} = \frac{\kappa - \sqrt{q_0} z_A + \frac{(d-d_0)\sqrt{q_0}}{\sqrt{q_0 p_1 - d_0^2}} x}{\sqrt{\bar{q} - q_0 - \frac{(d-d_0)^2 q_0}{q_0 p_1 - d_0^2}}}$

Plugging this stuff inside $\tilde{G}_E$:

$$\tilde{G}_E = \int \frac{Dz_A Dx}{H(\frac{\kappa - \sqrt{q_0} z_A}{\sqrt{\bar{q}-q_0}})} H(\frac{\kappa - \sqrt{q_0} z_A + \frac{(d-d_0)\sqrt{q_0}}{\sqrt{q_0 p_1 - d_0^2}} x}{\sqrt{\bar{q} - q_0 - \frac{(d-d_0)^2 q_0}{q_0 p_1 - d_0^2}}}) \times$$

$$\times \log H\left(\frac{\sqrt{q_0}\kappa - d_0 z_A + \sqrt{q_0 p_1 - d_0^2} x}{\sqrt{q_0}\sqrt{\bar{q} - p_1}}\right)$$

The potential to be optimized over d_0, p_1,is:

$$\mathbb{H}_{FP} = \frac{1}{2}\log(\bar{q} - p_1) + \frac{1}{2}\frac{p_1 - \gamma}{\bar{q} - p_1} + \alpha \int \frac{Dz_A Dx}{H(\frac{\kappa - \sqrt{q_0} z_A}{\sqrt{\bar{q}-q_0}})} H(\frac{\kappa - \sqrt{q_0} z_A + \frac{(d-d_0)\sqrt{q_0}}{\sqrt{q_0 p_1 - d_0^2}} x}{\sqrt{\bar{q} - q_0 - \frac{(d-d_0)^2 q_0}{q_0 p_1 - d_0^2}}}) \times$$

$$\times \log H\left(\frac{\sqrt{q_0}\kappa - d_0 z_A + \sqrt{q_0 p_1 - d_0^2} x}{\sqrt{q_0}\sqrt{\bar{q} - p_1}}\right)$$

with$\gamma = \frac{d-d_0}{(\bar{q}-q_0)^2}(\bar{q}(d+d_0) - 2q_0 d)$. Notice that the dependence on $p_0$has disappeared.

From now on we turn to the conventions:$\bar{q} \to \tilde{Q}, q_0 \to \tilde{q}, d \to s, d_0 \to \tilde{s}, p_1 \to q, x \to z, z_A \to \tilde{z}$: in the very end the Franz-Parisi entropy is the the extremum of this quantity:

$$\mathbb{H}_{FP} = \frac{1}{2}\log(\tilde{Q} - q) + \frac{1}{2}\frac{q - \gamma}{\tilde{Q} - q} + \alpha \int \frac{Dz D\tilde{z}}{H(\frac{\kappa - \sqrt{\tilde{q}}\tilde{z}}{\sqrt{\tilde{Q} - \tilde{q}}})} H(\frac{\kappa - \sqrt{\tilde{q}}\tilde{z} + \frac{(s-\tilde{s})\sqrt{\tilde{q}}}{\sqrt{\tilde{q}q - \tilde{s}^2}} z}{\sqrt{\tilde{Q} - \tilde{q} - \frac{(s-\tilde{s})^2 \tilde{q}}{\tilde{q}q - \tilde{s}^2}}}) \times$$

$$\times \log\, H\left(\frac{\sqrt{\tilde{q}}\kappa - \tilde{s}\tilde{z} + \sqrt{\tilde{q}q - \tilde{s}^2}z}{\sqrt{\tilde{q}}\sqrt{\tilde{Q} - q}}\right)$$

Deriving we get the saddle point equations for $q, \tilde{s}$:

$$\frac{1}{2}\frac{q-\gamma}{(\tilde{Q}-q)^2} + \alpha\partial_q\tilde{G}_E = 0$$

$$\frac{\tilde{Q}\tilde{s} - \tilde{q}s}{(\tilde{Q}-q)(\tilde{Q}-\tilde{q})^2} + \alpha\partial_{\tilde{s}}\tilde{G}_E = 0$$

with$\gamma = \frac{s-\tilde{s}}{(\tilde{Q}-\tilde{q})^2}(\tilde{Q}(s+\tilde{s}) - 2\tilde{q}s)$. We remark that $\tilde{q}$ is known and given by the Gardner computation.

3.2.3 Asymptotic behaviour

For small distances $s \to 1$ we expect $q \to 1$, since non reference solutions are near to the reference one and than near between themselves. So we expand the potential:

$$\mathbb{H}_{FP} = \frac{1}{2(1-q)}\{1 - \gamma - \alpha\int \frac{DzD\tilde{z}}{H(\frac{\kappa-\sqrt{\tilde{q}}\tilde{z}}{\sqrt{1-\tilde{q}}})}\; H(\frac{\kappa - \sqrt{\tilde{q}}\tilde{z} + \frac{(s-\tilde{s})\sqrt{\tilde{q}}}{\sqrt{\tilde{q}-\tilde{s}^2}}z}{\sqrt{1-\tilde{q} - \frac{(s-\tilde{s})^2\tilde{q}}{\tilde{q}-\tilde{s}^2}}})\times$$

$$\times\left(\frac{\sqrt{\tilde{q}}\kappa - \tilde{s}\tilde{z} + \sqrt{\tilde{q}-\tilde{s}^2}z}{\sqrt{\tilde{q}}}\right)^2 \cdot \theta(\sqrt{\tilde{q}}\kappa - \tilde{s}\tilde{z} + \sqrt{\tilde{q}-\tilde{s}^2}z)\} + O(\log(1-q))$$

Naming $\Xi(s,\tilde{s})$ the coefficient in the curly brackets, we get the limit value of $s, \tilde{s}$ from the saddle point equations:

$$\Xi(s,\tilde{s}) = 0$$

$$\partial_{\tilde{s}}\Xi(s,\tilde{s}) = 0 \tag{29}$$

For $\alpha \to 0 \Rightarrow \tilde{q} = 0,\ s \to -1$, the above equations reduce to $\tilde{s} = 0$. This is compatible with the numerical solution of the equations at decreasing α, see Fig.

Asymptotic behaviours: small distances From $s \to 1, q \to 1$ it follows $\tilde{s} \to \tilde{q}$: configurations very near to typical solutions have mutual distance equal to the distance between typical solutions; this intuition is confirmed numerically, see Fig. 25.

For values of s approaching 1 we show in the graphic $\Xi(\tilde{s})$ ($\kappa = 0.$, $\alpha = 1.3$)

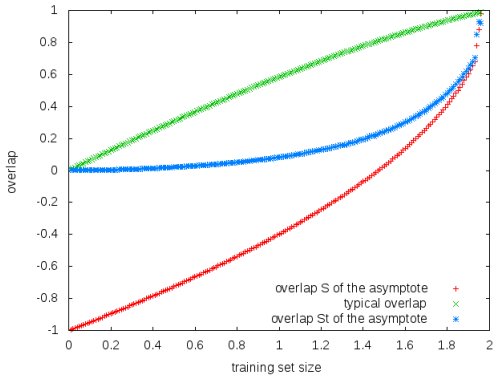

Figure 24: Different overlaps are plotted as a function of α, at $\kappa = 0$: in green the overlap $\tilde{q}$ between typical solutions: it is 0 for $\alpha = 0$ and approaches $\alpha = 2$ linearly. In red the overlap s for the Franz-Parisi entropy and in blue the relative $\tilde{s}$, as computed from eqs. (29). These are solved by minimizing $\Xi(s, \cdot)$ with Brent method and solving the remaining equation $\Xi(s, \tilde{s}(s)) = 0$ with the secant method. Beforehand we had tried 2D Newton method, with poor results.

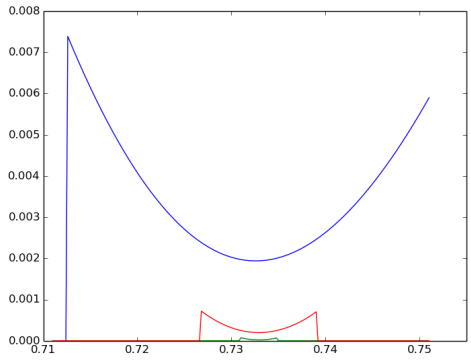

Figure 25: $\Xi(\tilde{s})$ is plotted as function of $\tilde{s}$ for different values of s approaching 1 ($s = 0.999, 0.9999, 0.99999$), with $\kappa = 0.$, $\alpha = 1.3$. Solving the equations (29) for $\Xi(s, \tilde{s})$ consists in finding when the bottom of the "parabola" crosses the $\Xi = 0$ axis. Here the scenary is consistent with $\tilde{s} \to \tilde{q} \simeq 0.733$. We were led to draw this picture by numerical issues when solving the equations for $\Xi(s, \tilde{s})$: actually, notice the shrinking of the definition domain.

Asymptotic behaviours: large distances

Decreasing s below $s = \tilde{q}$, it has been observed that the order parameter q increases in a linear fashion. Above $q \gtrsim 0.98$ (typically for negative s) there is a saturation of the value of q; such saturation we conjecture to be unreal and due to numerical errors (having found an analogous phenomenon in the study of the RS Gardner entropy, see above .

This problem probably comes from the numeric evaluation of the logarithm of a very small number (indeed the denominator of the argument of $\log H$ is $\sqrt{1-q}$. We try to fix this by means of a splitting of the energetic term:

$$\tilde{G}_E \simeq \int \frac{DzD\tilde{z}}{H(\frac{\kappa-\sqrt{\tilde{q}}\tilde{z}}{\sqrt{1-\tilde{q}}})} H(\frac{\kappa - \sqrt{\tilde{q}}\tilde{z} + \frac{(s-\tilde{s})\sqrt{\tilde{q}}}{\sqrt{\tilde{q}q-\tilde{s}^2}}z}{\sqrt{1-\tilde{q}-\frac{(s-\tilde{s})^2\tilde{q}}{\tilde{q}q-\tilde{s}^2}}})\times$$

$$\times\{\log H\left(\frac{\sqrt{\tilde{q}}\kappa - \tilde{s}\tilde{z} + \sqrt{\tilde{q}q-\tilde{s}^2}z}{\sqrt{\tilde{q}}\sqrt{1-q}}\right)\cdot\theta(\delta - \sqrt{\tilde{q}}\kappa - \tilde{s}\tilde{z} + \sqrt{\tilde{q}q-\tilde{s}^2}z)+$$

$$+[-\frac{1}{2}\left(\frac{\sqrt{\tilde{q}}\kappa - \tilde{s}\tilde{z} + \sqrt{\tilde{q}q-\tilde{s}^2}z}{\sqrt{\tilde{q}}\sqrt{1-q}}\right)^2 - \log\left(\sqrt{2\pi}\frac{\sqrt{\tilde{q}}\kappa - \tilde{s}\tilde{z} + \sqrt{\tilde{q}q-\tilde{s}^2}z}{\sqrt{\tilde{q}}\sqrt{1-q}}\right)]\times$$

$$\times\theta(\sqrt{\tilde{q}}\kappa - \tilde{s}\tilde{z} + \sqrt{\tilde{q}q-\tilde{s}^2}z - \delta)\}$$

where in theory $\frac{\delta}{\sqrt{1-q}} \gg 1$, in practice we take $\frac{\delta}{\sqrt{1-q}} = 10$ (indeed $H(10) \approx 8\cdot 10^{-24}$).

3.2.4 Results

The saddle point equations have to be solved to yield $\tilde{s}, q(\alpha, \kappa, s)$ and hence the Franz-Parisi entropy $\mathbb{H}_{\mathbb{FP}}(\alpha, \kappa, s)$. The solution is not completely trivial and exploits the fact that $\tilde{s}, q(\alpha, \kappa, \tilde{q}) = \tilde{q}$: the Franz-Parisi computation at the typical overlap yields the entropy of typical solutions, see Fig. 26 . Studying the 2D extremization problem we found that the definition domain shrinks to a point as $s \to 1$; moreover the saddle point is a minimum which lives in a narrow deep hole very near to the boundary of the definition domain. Derivative based methods (i.e. iterative solution and Newton) turned out to fail because they were attracted in plateaus with small derivative in the middle of the definition domain. Instead, a simple Monte Carlo search of the minimum behaved very efficiently. In fact, we parallelized the algorithm with Julia function `pmap` as the nested integrals of the Franz-Parisi entropy are time expensive.

In Fig. 27 the Franz-Parisi entropy is plotted: notice that decreasing α the entropy increases and that the maximum is taken at $s = \tilde{q}$, as expected. The other interesting feature is the existence of an asymptote at large distances, as also in the discrete case [34]. At small distances, instead, the Franz-Parisi

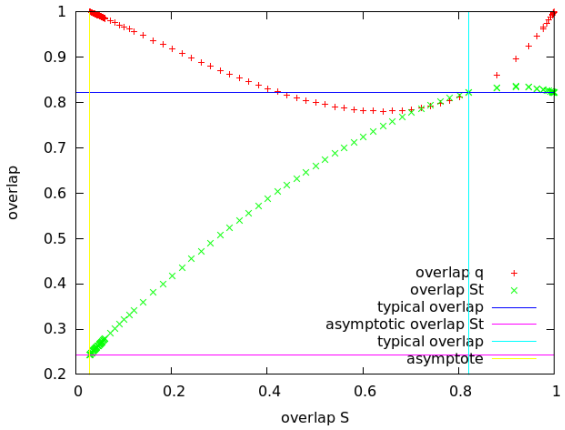

Figure 26: Saddle-point order parameters q and $\tilde{s}$ plotted versus s, for $\kappa = 0., \alpha = 1.5$. Notice the consistency with theoretical predictions: at $s = \tilde{q}$ also $q = \tilde{s} = \tilde{q}$ and at $s \to 1$we have$q \to 1, \tilde{s} \to \tilde{q}$. At large distances the results matches the theoretical asymptotes.

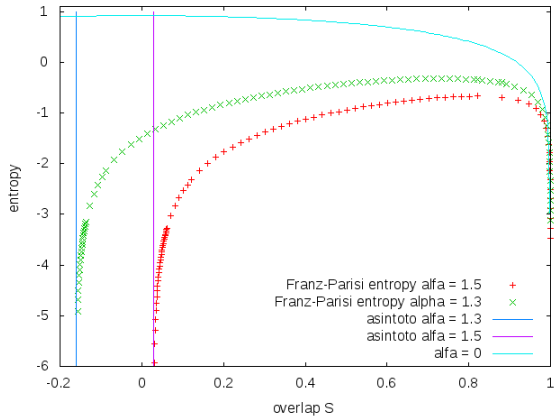

Figure 27: Franz-Parisi entropy as a function of the overlap s for $\alpha = 1.3, 1.5$ and $\kappa = 0$. The value of the asymptotes predicted from eqs. (29) is also reported. The light blue line is the upper limit $\alpha = 0$. For $\kappa \gtrless 0$ the behaviour is qualitatively similar.

entropy saturates the $\alpha = 0$: this does not come as a total surprise as in the continuum each solution has a neighboorhood of solutions around at finite N, but it is not completely trivial because in the thermodynamical limit typical solutions are thought to be on the surface of the solution space.

The bad news is that even in the $\kappa < 0$ region no qualitative difference has been noticed (we have kept below the AT line, as our computation is RS). We do not even report any plot, as they are the same as Fig. 27, but for the fact that, obviously, increasing κ at fixed α, the entropy decreases.

In conclusion, differently from the discrete case, the solutions don' t happen to be pointlike and isolated. Maybe this doesn' t have nothing to do with computational hardness, as there is not a continuous analogous of K-SAT to use as a guide, as far as we know. Maybe this approach is not good for continuous systems and different questions are to be asked. Or maybe the negative perceptron is not a suitable model.

4 Continuous perceptron: out-of-equilibrium analysis

In this Chapter we try to apply the non-equilibrium tools developed in Chapter 2 to the case of the storage problem of the continuous perceptron, with particular emphasis on the negative stability region.

In Section 4.1 we derive, using the replica trick, the saddle point equations for the large deviation measure. In Sec. 4.2 we try to apply the replicate BP approach. Finally, in Sec. 4.3 we investigate the computational hardness of learning comparing GD with replicated GD.

4.1 Constrained Reweighting

In this Section we consider the constrained reweighted system in the 1RSB ansatz and in the limit $y \to \infty$. The reason of this choice is that the y infinite limit allows some simplification in the computations and in [2] it was shown also to yield more reliable results. In the y infinite limit the discrete analysis suggests that a RSB step is required; finally, at $y \to \infty$ the constrained and unconstrained case are equivalent.

We reweight solutions with their local entropy of solutions: the large deviation measure is defined by the partition function:

$$Z(s,\alpha) \equiv \langle \int d\mu(\tilde{W})\ \chi_\xi(\tilde{W})\ \aleph^y(\tilde{W},s)\rangle_\xi \tag{30}$$

We take the quenched average of the relative free-entropy, which is self-averaging:

$$\phi = \lim_{n\to 0}\frac{1}{n}(\langle\{\int d\mu(\tilde{W}_a)\ \chi_\xi(\tilde{W})\ \aleph^y(\tilde{W}_a,s)\}^n\rangle_\xi - 1)$$

where $d\mu(W) = \delta(W^2 - \tilde{Q}N)dW$, $Z = \int d\mu(W)\chi_\xi(W)$ and $\aleph(\tilde{W},s) = \int d\mu(W)\chi_\xi(W)\ \delta(W\cdot\tilde{W} - sN)$. The integration runs over the n $\tilde{W}_a$ variables and over the $n \times y$ W_{ar}'s (we are thinking y as an integer).

Following exactly the same computations as for the Franz-Parisi entropy we get:

$$N\phi = \lim_{n\to 0}\frac{1}{n}\{\int dQ_{ab}d\hat{Q}_{ab}dQ_{ar,b}d\hat{Q}_{ar,b}dQ_{a,bs}d\hat{Q}_{a,bs}dQ_{ar,bs}d\hat{Q}_{ar,bs}\times$$
$$\times e^{N[\sum_{ab}Q_{ab}\hat{Q}_{ab}+\sum_{abr}Q_{ar,b}\hat{Q}_{ar,b}+\sum_{abr}Q_{a,br}\hat{Q}_{a,br}+\sum_{abrs}Q_{ar,bs}\hat{Q}_{ar,bs}]}\times G_E^{\alpha N}G_S^N - 1\} \tag{31}$$

$$G_E = \int dx_a d\hat{x}_a dx_{ar}d\hat{x}_{ar}\prod_a g(x_a)\prod_{ar}g(x_{ar})\ exp - i\left\{\sum_a x_a\hat{x}_a + \sum_{ar}x_{ar}\hat{x}_{ar}\right\}\times$$
$$exp - \frac{1}{2}\left[\sum_{ab}\hat{x}_a\hat{x}_bQ_{ab} + 2\sum_{abr}\hat{x}_{ar}\hat{x}_bQ_{ar,b} + \sum_{abrs}\hat{x}_{ar}\hat{x}_{bs}Q_{ar,bs}\right] \tag{32}$$

$$G_S = \int dW_a dW_{ar} exp\{-\sum_{ab} \hat{Q}_{ab} W_a W_b - \sum_{abr} \hat{Q}_{ar,b} W_{ar} W_b -$$

$$- \sum_{abr} \hat{Q}_{a,br} W_a W_{br} - \sum_{abrs} \hat{Q}_{ar,bs} W_{ar} W_{bs}\}$$

After getting rid of the $\hat{Q}$'s we can express the reweighted free entropy as sum of the entropic and energetic terms:

$$\phi = \lim_{n\to 0} \frac{1}{n} extr_Q \{\frac{1}{2} logDetQ + \alpha \log G_E\}$$

4.1.1 RS ansatz

We start with the RS ansatz, this is useful in order to check the 1RSB results:

$$Q_{aa} = \tilde{Q}, \quad Q_{ab} = \tilde{q}$$

$$Q_{ar,a} = s, \quad Q_{ar,b} = \tilde{s}$$

$$Q_{ar,ar} = \tilde{Q}, \quad Q_{ar,as} = q_1, \quad Q_{ar,bs} = q_0$$

with $\tilde{q} < \tilde{Q} q_0 < q_1 < \tilde{Q}$.

Entropic term

$$\lim_{n\to 0} \frac{1}{2n} logDetQ = \frac{1}{2}\{log(\tilde{Q} - \tilde{q}) + \frac{\tilde{q}}{\tilde{Q} - \tilde{q}} + (y-1)\log[\tilde{Q} - q_1] +$$

$$+ \log[\tilde{Q} - q_1 + y(q_1 - q_0 - (\gamma - \gamma_0))] + y \frac{q_0 - \gamma_0}{\tilde{Q} - q_1 + y(q_1 - q_0 - (\gamma - \gamma_0))}\} \quad (33)$$

where:

$\gamma = \frac{s-\tilde{s}}{(\tilde{Q}-\tilde{q})^2}(\tilde{Q}(s+\tilde{s}) - 2\tilde{q}s)$

$\gamma_0 = -\frac{s-\tilde{s}}{(\tilde{Q}-\tilde{q})^2}(\tilde{q}(s+\tilde{s}) - 2\tilde{Q}\tilde{s})$

$\gamma - \gamma_0 = \frac{(s-\tilde{s})^2}{\tilde{Q}-\tilde{q}}$

With the scaling relations $\tilde{s} \to s - \frac{\delta s}{y}$, $q_1 \to q_0 + \frac{\delta q}{y}$, $\tilde{q} \to \tilde{Q} - \frac{\delta \tilde{q}}{y}$ the infinite y limit is:

$$\lim_{y\to\infty} \frac{1}{2y} logDetQ = \frac{1}{2}\{\log[\tilde{Q} - q_0] + \frac{\tilde{Q}}{\delta\tilde{q}} + \frac{q_0 + \frac{\delta s}{\delta \tilde{q}^2}(\tilde{Q}\delta s - 2\delta\tilde{q}s)}{\tilde{Q} - q_0 + \delta q - \frac{\delta s^2}{\delta\tilde{q}}}\} \quad (34)$$

Energetic term

Part of the computation is identical to the Franz-Parisi one, but for the fact that we keep y finite, so there is some simplification less.

Plugging the RS ansatz in G_E:

$$G_E = \int Dz_1 Dz_2 \left[\int DwDxg(\sqrt{\tilde{Q}-\tilde{q}}x - A) \right.$$

$$\left. \left\{ \int Dt \cdot g(\sqrt{\tilde{Q}-q_1}t - B + \frac{s-\tilde{s}}{\sqrt{\tilde{Q}-\tilde{q}}}x + \sqrt{q_1 - q_0 - \frac{(s-\tilde{s})^2}{(\tilde{Q}-\tilde{q})}}w) \right\}^y \right]^n$$

This is something like $G_E = 1 + nG'_E + ...$ with

$$G'_E = \int Dz_1 Dz_2 \log \left[\int DwDxg(\sqrt{\tilde{Q}-\tilde{q}}x - A) \right.$$

$$\left. \left\{ \int Dt \cdot g(\sqrt{\tilde{Q}-q_1}t - B + \frac{s-\tilde{s}}{\sqrt{\tilde{Q}-\tilde{q}}}x + \sqrt{q_1 - q_0 - \frac{(s-\tilde{s})^2}{(\tilde{Q}-\tilde{q})}}w) \right\}^y \right]$$

If $g(\cdot) = \theta(\cdot - \kappa)$, the expression can be simplified rotating x and w:

$$\tilde{G}_E = \int Dz_1 Dz_2 \log\{ \int Dx \; H(\frac{\sqrt{C^2 + C'^2}}{C}\{\kappa' - A'\} - \frac{C'}{C}x) \times$$

$$\times H^y \left(\kappa'' - B'(z_1, z_2) - \sqrt{C^2 + C'^2}x \right) \}$$

Rotating z_1, z_2 we obtain the final formula:

$$\tilde{G}_E = \int Dz_1 Dz_2 \log\{ \int Dx \; H(\frac{\kappa - \sqrt{\tilde{q} - \frac{\tilde{s}^2}{q_0}}z_1 - \frac{\tilde{s}}{\sqrt{q_0}}z_2 - \frac{s-\tilde{s}}{\sqrt{q_1 - q_0}}x}{\sqrt{\tilde{Q} - \tilde{q} - \frac{(s-\tilde{s})^2}{q_1 - q_0}}}) \times$$

$$\times H^y \left(\frac{\kappa - \sqrt{q_0}z_2 - \sqrt{q_1 - q_0}x}{\sqrt{\tilde{Q} - q_1}} \right) \} \tag{35}$$

Notice also that for $y = 0$ we recover the Gardner RS entropy.

It also useful the formula:

$$\int dx d\hat{x} \theta(x - \kappa) e^{-i(x-B)\hat{x} - \frac{A}{2}\hat{x}^2} = \int dx d\hat{x} \theta(x + B - \kappa) e^{-ix\hat{x} - \frac{A}{2}\hat{x}^2} =$$

$$= \int dx d\hat{x} \theta(\sqrt{A}x + B - \kappa) e^{-ix\hat{x} - \frac{1}{2}\hat{x}^2} = \int Dx \theta(\sqrt{A}x + B - \kappa) = H(\frac{\kappa - B}{\sqrt{A}})$$

4.1.2 1RSB ansatz

We break the symmetry at the level of reference configurations (indices a, b):

$$Q_{a,a} = \tilde{Q},\ Q_{\alpha\beta,\alpha\beta'} = \tilde{q}_1,\ Q_{\alpha\beta,\alpha'\beta'} = \tilde{q}_0$$

$$Q_{a,ar} = s,\ Q_{\alpha\beta,\alpha\beta' r} = \tilde{s}_1,\ Q_{\alpha\beta,\alpha'\beta' r} = \tilde{s}_0$$

$$Q_{ar,ar} = \tilde{Q},\ Q_{ar,as} = q_2,\ Q_{\alpha\beta r,\alpha\beta' s} = q_1,\ Q_{\alpha\beta r,\alpha'\beta' s} = q_0$$

Entropic term

We use $detQ = detQ_{ab} \times \det(Q_{ar,bs} - Q_{ar,b}Q_{ab}^{-1}Q_{ar,b}^T)$: the first factor gives rise to the usual contribution (p.e. eq (81) of [36]):

$$\lim_{n\to 0}\ \frac{1}{2n} logDetQ_{ab} =$$

$$= \frac{1}{2}\{\frac{m-1}{m} log(\tilde{Q} - \tilde{q_1}) + \frac{1}{m}\log[m(\tilde{q}_1 - \tilde{q}_0) + \tilde{Q} - \tilde{q}_1] + \frac{\tilde{q}_0}{m(\tilde{q}_1 - \tilde{q}_0) + \tilde{Q} - \tilde{q}_1}\}$$

This term will give, in the limit $y \to \infty$, the contribution (see below the scaling):

$$\frac{1}{2}\{\frac{1}{x}\log[1 + \frac{x(\tilde{Q} - \tilde{q}_0)}{\delta\tilde{q}}] + \frac{\tilde{q}_0}{x(\tilde{Q} - \tilde{q}_0) + \delta\tilde{q}}\}$$

In order to compute the determinant of a generic 2RSB matrix $Q'_{\alpha\beta r,\alpha'\beta' s}$ we report the eigenvalues with their multiplicities:

$\lambda_1 = \tilde{Q}' - \tilde{q}'_2\,,\ \ \mu_1 = n(y-1)$

$\lambda_2 = \tilde{Q}' + (y-1)\tilde{q'}_2 - y\tilde{q'}_1\,,\ \ \mu_2 = \frac{n}{m}(m-1)$

$\lambda_3 = \tilde{Q}' + (y-1)\tilde{q}'_2 + (m-1)y\tilde{q}'_1 - my\tilde{q}'_0\,,\ \ \mu_3 = \frac{n}{m} - 1$

$\lambda_4 = \tilde{Q}' + (y-1)\tilde{q}'_2 + (m-1)y\tilde{q}'_1 + (n-m)y\tilde{q}'_0\,,\ \ \mu_4 = 1$

The eigenvalues are listed in this order: the eigenvectors of the first one link only the innermost block and so on. The multiplicities sum to ny.(The notation of [5] is $m_1 = y, m_2 = my$.)

Since the $n \to 0$ limit of the Q' matrix is finite (elementwise), it is comfortable to compute first the limit of the log det and then plug in (the limit of) the q''s:

$\lim_{n\to 0}\ \frac{1}{2n}\ =\ \frac{1}{2}\{(y-1)\log\lambda_1 + \frac{m-1}{m}\log\lambda_2 + \frac{1}{m}\log\lambda_3 + \frac{y\tilde{q}'_0}{\lambda_3}\}$

Here:

$\tilde{Q}' = \tilde{Q} - \gamma_2$

$\tilde{q}'_2 = q_2 - \gamma_2$

$\tilde{q}'_1 = q_1 - \gamma_1$

$\tilde{q}'_0 = q_0 - \gamma_0$

where the limit is implied and we do not report the expression for gamma.

Energetic term

The energetic term produces equation (B.59) of [2]:

$$\tilde{G}_E = \frac{\log G_E}{n} = \frac{1}{m}\int Dz_0 Dz_0' \log\left(\int Dz_1 Dz_1' \left[\int Dz_2 H^y(A(z_0,z_1,z_2)) \times L(z_0,z_0',z_1,z_1',z_2)\right]^m\right)$$

$$= \frac{1}{m}\int Dz_0 Dz_0' \log\left(\int Dz_1 Dz_1' \left[\int Dz_2 H^y(A(z_0,z_1,z_2)) \times L(z_0,z_0',z_1,z_1',z_2)\right]^m\right)$$

with L given in (B.60) where $f(\cdot) = \theta(\cdot - \kappa)$. We take here the limit $y \to \infty$that reduces to neglecting L and recover the G_E of the unconstrained case:

$$\lim_{y\to\infty} \tilde{G}_E = \frac{1}{m}\int Dz_0 \log\left(\int Dz_1 \left[\int Dz_2 H^y(A(z_0,z_1,z_2))\right]^m\right) \tag{36}$$

$A(z_0,z_1,z_2)$ can be evinced from (B.24,B.25), adding $-\kappa$ in the θ-function and reabsorbing it by means of the translation of $\lambda^{\beta,a} \to \lambda^{\beta,a} + \kappa$ that in the final expression propagates as (the sign is adjusted with $z_0 \to -z_0$)

$$A(z_0,z_1,z_2) = \frac{\kappa + \sqrt{q_0}z_0 + \sqrt{q_1 - q_0}z_1 + \sqrt{q_2 - q_1}z_2}{\sqrt{\tilde{Q} - q_2}}$$

Notice that only the order parameters of kind $Q_{ar,bs}$enter this expression.

The consistency with the RS result eq (35) is verified by dropping the H function here (without proof), rotating the z's so to get $B = \sqrt{q_0}z$ and setting $m = 1, q_2 = q_1$.

Infinite y limit

We anticipate that in the limit $y \to \infty$ the order parameters relative to the reference configurations will disappear (obvious from the weights in the partition function)

As in [2] we adopt the scaling relation

$$m \to \frac{x}{y}$$

$$q_2 \to q_1 + \frac{\delta q}{y}$$

$$\tilde{s}_1 \to s - \frac{\delta s}{y}$$

$$\tilde{q}_1 \to \tilde{Q} - \frac{\delta \tilde{q}}{y}$$

It is useful to rearrange:
$\lambda_1 = \tilde{Q} - q_2$
$\lambda_2 = \tilde{Q} - q_2 + y(q_2 - q_1 - (\gamma_2 - \gamma_1))$
$\lambda_3 = \tilde{Q} - q_2 + y(q_2 - q_1 - (\gamma_2 - \gamma_1)) + my(q_1 - q_0 - (\gamma_1 - \gamma_0))$
and now consider the asymptotic behaviour of the various terms:

$$\gamma_2 - \gamma_1 = \frac{(s - \tilde{s}_1)^2}{\tilde{Q} - \tilde{q}_1} \sim \frac{\delta s^2}{y\delta\tilde{q}}$$

$$\gamma_1 - \gamma_0 = ... \sim O(1)$$

$$\gamma_0 = ... \sim O(1)$$

and notice that the eigenvalues are order $O(1)$ and $\lambda_3 = \lambda_2 + x(q_1 - q_0 - (\gamma_1 - \gamma_0)) \sim O(1)$):

$\lim_{y\to\infty} \frac{1}{2y}\{(y-1)\log\lambda_1 + \frac{m-1}{m}\log\lambda_2 + \frac{1}{m}\log\lambda_3 + \frac{y\tilde{q}_0'}{\lambda_3}\} \simeq \frac{1}{2}\{\log\lambda_1 - \frac{1}{x}\log\lambda_2 + \frac{1}{x}\log\lambda_3 + \frac{\tilde{q}_0'}{\lambda_3}\}$
Summing also the contribution of the first determinant:

$$G_S^\infty = \frac{1}{2}\{\frac{\tilde{q}_0}{x(\tilde{Q} - \tilde{q}_0) + \delta\tilde{q}} + \frac{1}{x}\log[1 + \frac{x(\tilde{Q} - \tilde{q}_0)}{\delta\tilde{q}}] + \log(\tilde{Q} - q_1) +$$

$$+\frac{1}{x}\log[1 + \frac{x(q_1 - q_0 - (\gamma_1 - \gamma_0))}{\tilde{Q} - q_1 + (\delta q - \frac{\delta s^2}{\delta\tilde{q}}) + x(q_1 - q_0 - (\gamma_1 - \gamma_0))}] +$$

$$+\frac{q_0 - \gamma_0}{\tilde{Q} - q_1 + (\delta q - \frac{\delta s^2}{\delta\tilde{q}}) + x(q_1 - q_0 - (\gamma_1 - \gamma_0))}\}$$

This result can be checked: if we set $q_1 \to q_0, \tilde{s}_0 \to s - \frac{\delta s}{y}, \tilde{q}_0 \to \tilde{Q} - \frac{\delta\tilde{q}}{y}$ (in fact it is sufficient $q_1 \to q_0, \tilde{s}_0 \to s, \tilde{q}_0 \to \tilde{Q}$) and send $y \to \infty$ we recover eq. (34) .

The energetic term is, by means of the saddle point method:

$$\tilde{G}_E^\infty = \frac{1}{x}\int Dz_0 \log\left(\int Dz_1 e^{xB(z_0,z_1)}\right) \tag{37}$$

where $B(z_0, z_1) = \max_{z_2}\{-\frac{z_2^2}{2} + \log H\left(\frac{\kappa + \sqrt{q_0}z_0 + \sqrt{q_1 - q_0}z_1 + \sqrt{\delta\tilde{q}}z_2}{\sqrt{\tilde{Q} - q_1}}\right)\}$
The RS semplification occurs if $q_1 = q_0$: $\tilde{G}_E^{\infty,RS} = \int Dz_0 \max_{z_2}\{-\frac{z_2^2}{2} + \log H\left(\frac{\kappa + \sqrt{q_0}z_0 + \sqrt{\delta\tilde{q}}z_2}{\sqrt{\tilde{Q} - q_2}}\right)\}$
The experience gathered in [2] suggests that the solving the saddle point equations for this system may be quite troublesome. We posticipated this task, and finally we gave it up, as we think it isn' t worth the effort, see below.

4.2 Replicated BP

In the previous Section we tried to compute the formal energy of the system described by the free entropy:

$$\phi(s,y) = \left\langle \log \left(\int d\mu(\tilde{W}) \chi(\tilde{W}\cdot\xi)\, \aleph^y(\tilde{W},s) \right) \right\rangle$$

with (the logarithm of) $\aleph(\tilde{W},s) = \int d\mu(W)\chi_\xi(W)\ \delta(W\cdot\tilde{W} - sN)$ (local entropy) playing the role of an energy and y of an inverse temperature. Instead of using the replica method, we can take y integer and approximate the free entropy with the Bethe free-energy of a reference system coupled to y copies of the system, to be evaluated using belief propagation or some variation. For N large enough, the quenched disorder can be thought to be implemented in the single sample, so we neglect the average over ξ. In this way we can estimate the local entropy (at finite y), that we were not able to compute in the previous section, by derivation of the free entropy:

$$\mathcal{S}_{\mathcal{I}} = -\frac{1}{N}\partial_y \phi$$

as the usual thermodynamics relation.

Another expedient is relaxing the hard $\delta(W\cdot\tilde{W} - sN)$ and spherical constraints introducing a soft constraint as a local energetic term. In the end, we want to do message-passing on the probabilistic graphical model with variable nodes $\tilde{W}, W_a$, see Fig. 28, and partition function:

$$Z(s,y) = \int d\tilde{W} d\mu W_a\, e^{H(\tilde{W}) + \sum_a H(W_a) + \gamma \sum_a \tilde{W} W_a} \tag{38}$$

On this grafical model we write the BP equations (7). The starting point is to treat the y "replicas" in a symmetric way, requiring they obey the same probability distribution (that depends on i). Then we assume a gaussian form for the marginal of the synaptic weights and the messages and consider their scaling behaviour: in fact, due to the spherical constraint, we expect the $\tilde{W}_i, W_i$ to be distributed with both mean and variance $O(1)$:

$$p(\tilde{W}_i) \propto \exp\{\frac{\tilde{m}_i}{\tilde{\rho}_i}\tilde{W}_i - \frac{\tilde{W}_i^2}{2\tilde{\rho}_i}\} \tag{39}$$

$$p(W_i) \propto \exp\{\frac{m_i}{\rho_i}W_i - \frac{W_i^2}{2\rho_i}\} \tag{40}$$

From the BP formula for the marginal we infer the scaling of the messages ingoing the variable nodes from the factor nodes relative to the input patterns:

$$\tilde{\nu}_{\xi\to i}(\tilde{W}_i) \propto \exp\{\frac{\tilde{m}_{\xi\to i}}{\tilde{\rho}_{\xi\to i}}\tilde{W}_i - \frac{\tilde{W}_i^2}{2\tilde{\rho}_{\xi\to i}}\}$$

$$\nu_{\xi\to i}(W_i) \propto \exp\{\frac{m_{\xi\to i}}{\rho_{\xi\to i}}W_i - \frac{W_i^2}{2\rho_{\xi\to i}}\}$$

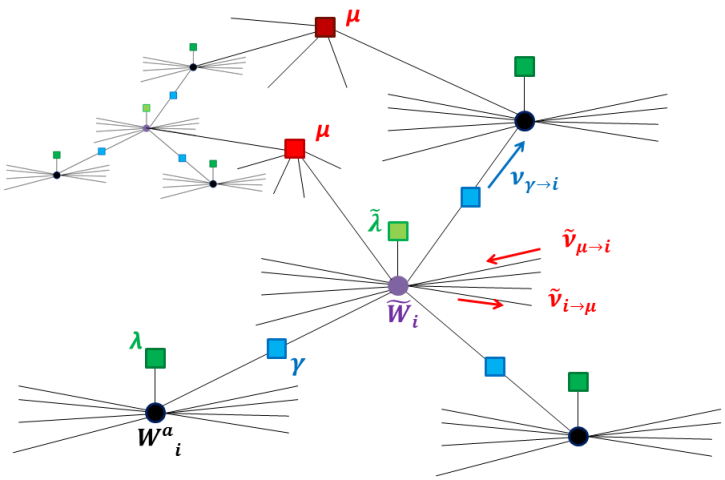

Figure 28: A slice of the factor of the model considered in equation (38). Only the local neighbourhood of site i is shown: in purple the node variables corresponding to $\tilde{W}_i$ and in black those of W_i^a (here $y = 3$). The green boxes represent the local field that sets the L^2 normalization. The blue boxes are the elastic interactions between the reference and the real replicas. Finally, the interrupted lines connect each variable node to a factor corresponding to a pattern (red boxes). The interaction between different sites (see small site j slice on upper left) is mediated uniquely by the pattern constraints. Some sample messages are also reported.

where m is $O(N^{1/2})$ and ρ $O(N)$. Notice that the moments m, ρ are the derivatives of the messages computed in $W = 0$ (and the same for the tilded). We only add that is *not* true that the messages are Gaussian distributed (they have a tail...) but in the N large limit only the first and second moments are important.

We now compute the marginals:

$$p(\tilde{W}_i) \propto \exp\{\tilde{W}_i \sum_{\xi} \frac{\tilde{m}_{\xi\to i}}{\tilde{\rho}_{\xi\to i}} - \frac{\tilde{W}_i^2}{2}(\tilde{\lambda} + \sum_{\xi} \tilde{\rho}^{-1}_{\xi\to i})\} \times \left[\tilde{\nu}_{\gamma\to i}(\tilde{W}_i)\right]^y$$

$$p(W_i) \propto \exp\{W_i \sum_{\xi} \frac{m_{\xi\to i}}{\rho_{\xi\to i}} - \frac{W_i^2}{2}(\lambda + \sum_{\xi} \rho^{-1}_{\xi\to i})\} \times \nu_{\gamma\to i}(W_i)$$

We have also imposed the spherical constraint here by means of the Lagrange multipliers $\tilde{\lambda}, \lambda$, which are $O(1)$ so that $\tilde{W}_i, W_i$ be $O(1)$: the constraint can be thought of as a single-variable factor node.

From comparison with 39 (must look at the W-dependence) we get the BP equations for the moments:

$$\frac{\tilde{m}_i}{\tilde{\rho}_i} = \sum_{\xi} \frac{\tilde{m}_{\xi\to i}}{\tilde{\rho}_{\xi\to i}} + y\frac{\tilde{m}_{\gamma\to i}}{\tilde{\rho}_{\gamma\to i}}$$

$$\tilde{\rho}_i^{-1} = \tilde{\lambda} + \sum_{\xi} \tilde{\rho}^{-1}{}_{\xi\to i} + y\tilde{\rho}^{-1}_{\gamma\to i}$$

$$\frac{m_i}{\rho_i} = \sum_{\xi} \frac{m_{\xi\to i}}{\rho_{\xi\to i}} + \frac{m_{\gamma\to i}}{\rho_{\gamma\to i}}$$

$$\rho_i^{-1} = \lambda + \sum_{\xi} \rho^{-1}_{\xi\to i} + \rho^{-1}_{\gamma\to i}$$

The messages of type $i \to \xi$ are recovered from the moments leaving out the ξ addend from the first sum:

$$\frac{\tilde{m}_{i\to\xi}}{\tilde{\rho}_{i\to\xi}} = \frac{\tilde{m}_i}{\tilde{\rho}_i} - \frac{\tilde{m}_{\xi\to i}}{\tilde{\rho}_{\xi\to i}}$$

$$\frac{m_{i\to\xi}}{\rho_{i\to\xi}} = \frac{m_i}{\rho_i} - \frac{m_{\xi\to i}}{\rho_{\xi\to i}}$$

$$\tilde{\rho}^{-1}_{i\to\xi} = \tilde{\rho}_i^{-1} - \tilde{\rho}^{-1}_{\xi\to i}$$

$$\rho^{-1}_{i\to\xi} = \rho_i^{-1} - \rho^{-1}_{\xi\to i}$$

Notice the different orders of the terms. Similarly the messages of type $i \to \gamma$:

$$\frac{\tilde{m}_{i\to\gamma}}{\tilde{\rho}_{i\to\gamma}} = \frac{\tilde{m}_i}{\tilde{\rho}_i} - \frac{\tilde{m}_{\gamma\to i}}{\tilde{\rho}_{\gamma\to i}}$$

$$\frac{m_{i\to\gamma}}{\rho_{i\to\gamma}} = \frac{m_i}{\rho_i} - \frac{m_{\gamma\to i}}{\rho_{\gamma\to i}}$$

$$\tilde{\rho}^{-1}_{i\to\gamma} = \tilde{\rho}^{-1}_i - \tilde{\rho}^{-1}_{\gamma\to i}$$

$$\rho^{-1}_{i\to\gamma} = \rho^{-1}_i - \rho^{-1}_{\gamma\to i}$$

Now we have to manipulate the second set of BP equations:
with $\tilde{u}_{\gamma\to i}(\tilde{W}_i) \propto \int dW_i e^{\gamma W_i \tilde{W}_i} u_{i\to\gamma}(W_i) \Rightarrow$

$$\frac{\tilde{m}_{\gamma\to i}}{\tilde{\rho}_{\gamma\to i}} = \gamma m_{i\to\gamma}$$

$$\tilde{\rho}^{-1}_{\gamma\to i} = -\gamma^2 \rho_{i\to\gamma}$$

and similarly:

$$\frac{m_{\gamma\to i}}{\rho_{\gamma\to i}} = \gamma \tilde{m}_{i\to\gamma}$$

$$\rho^{-1}_{\gamma\to i} = -\gamma^2 \tilde{\rho}_{i\to\gamma}$$

The messages from the patterns are instead given by:

$$\tilde{\nu}_{\xi\to i}(\tilde{W}_i) \propto \int \prod_{j\neq i} d\tilde{W}_j \, \theta(\tilde{W}\cdot\xi - \sqrt{N}\kappa) \, \exp\{\sum_{j\neq i} \frac{\tilde{m}_{j\to\xi}}{\tilde{\rho}_{j\to\xi}} \tilde{W}_j - \sum_{j\neq i} \tilde{\rho}^{-1}_{j\to\xi} \frac{\tilde{W}_j^2}{2}\}$$

$$\propto \int_{\sqrt{N}\kappa}^{+\infty} d\kappa' \int \prod_{j\neq i} d\tilde{W}_j \, \delta(\tilde{W}\cdot\xi - \kappa') \, \exp\{\sum_{j\neq i} \frac{\tilde{m}_{j\to\xi}}{\tilde{\rho}_{j\to\xi}} \tilde{W}_j - \sum_{j\neq i} \tilde{\rho}^{-1}_{j\to\xi} \frac{\tilde{W}_j^2}{2}\}$$

exploiting that $\xi_i = \pm 1$:

$$\tilde{\nu}_{\xi\to i}(\tilde{W}_i) \propto$$

$$\propto \int_{\sqrt{N}\kappa}^{+\infty} d\kappa' \int \prod_{j\neq i} d\tilde{W}_j \, \delta(\tilde{W}_i\xi_i + \sum_j \tilde{W}_j - \kappa') \, \exp\{\sum_{j\neq i} \xi_j \frac{\tilde{m}_{j\to\xi}}{\tilde{\rho}_{j\to\xi}} \tilde{W}_j - \sum_{j\neq i} \tilde{\rho}^{-1}_{j\to\xi} \frac{\tilde{W}_j^2}{2}\}$$

As the inner integral is the probability that the sum $\sum_j X_j$ of independent gaussian RVs be equal to $\kappa' - \tilde{W}_i\xi_i$:

$$\tilde{\nu}_{\xi\to i}(\tilde{W}_i) \propto \int_{\sqrt{N}\kappa}^{+\infty} d\kappa' \, \exp - \frac{(\kappa' - \tilde{W}_i\xi_i - \sum_{j\neq i}\xi_j\tilde{m}_{j\to\xi})^2}{2\sum_{j\neq i}\tilde{\rho}_{j\to\xi}}$$

$$\propto H(\frac{\sqrt{N}\kappa - \tilde{W}_i\xi_i - \sum_{j\neq i}\xi_j\tilde{m}_{j\to\xi}}{\sqrt{\sum_{j\neq i}\tilde{\rho}_{j\to\xi}}}),$$

hence by derivation ($\frac{\tilde{m}_{\xi\to i}}{\tilde{\rho}_{\xi\to i}} = \partial_{\tilde{W}_i}\log\tilde{\nu}_{\xi\to i}(\tilde{W}_i)|_{\tilde{W}_i=0}$ and $\tilde{\rho}^{-1}_{\xi\to i} = -\partial^2_{\tilde{W}_i}\log\tilde{\nu}_{\xi\to i}(\tilde{W}_i)|_{\tilde{W}_i=0}$):

$$\frac{\tilde{m}_{\xi\to i}}{\tilde{\rho}_{\xi\to i}} = \xi_i\, g(\sqrt{N}\kappa - \sum_{j\neq i}\xi_j\tilde{m}_{j\to\xi}, \sum_{j\neq i}\tilde{\rho}_{j\to\xi})$$

$$\tilde{\rho}^{-1}_{\xi\to i} = g^2(\sqrt{N}\kappa - \sum_{j\neq i}\xi_j\tilde{m}_{j\to\xi}, \sum_{j\neq i}\tilde{\rho}_{j\to\xi}) -$$

$$-\frac{\sqrt{N}\kappa - \sum_{j\neq i}\xi_j\tilde{m}_{j\to\xi}}{\sum_{j\neq i}\tilde{\rho}_{j\to\xi}} g(\sqrt{N}\kappa - \sum_{j\neq i}\xi_j\tilde{m}_{j\to\xi}, \sum_{j\neq i}\tilde{\rho}_{j\to\xi})$$

where $g(a,b) = -\partial_a \log H(a/\sqrt{b}) = \frac{\frac{e^{-\frac{a^2}{2b}}}{\sqrt{2\pi b}}}{H(a/\sqrt{b})}$. Similarly:

$$\frac{m_{\xi\to i}}{\rho_{\xi\to i}} = \xi_i\, g(\sqrt{N}\kappa - \sum_{j\neq i}\xi_j m_{j\to\xi}, \sum_{j\neq i}\rho_{j\to\xi})$$

$$\rho^{-1}_{\xi\to i} = g^2(\sqrt{N}\kappa - \sum_{j\neq i}\xi_j m_{j\to\xi}, \sum_{j\neq i}\rho_{j\to\xi}) -$$

$$-\frac{\sqrt{N}\kappa - \sum_{j\neq i}\xi_j m_{j\to\xi}}{\sum_{j\neq i}\rho_{j\to\xi}} g(\sqrt{N}\kappa - \sum_{j\neq i}\xi_j m_{j\to\xi}, \sum_{j\neq i}\rho_{j\to\xi})$$

Notice that $\sqrt{N}\kappa - \sum_{j\neq i}\xi_j m_{j\to\xi} \sim O(\sqrt{N})$, $\sum_{j\neq i}\rho_{j\to\xi} \sim O(N) \Rightarrow g \sim O(N^{-1/2}) \Rightarrow \rho_{\xi\to i} \sim O(N)$, as it should be. Notice the this second set of BP equations is nothing else than that of a single perceptron, as it should be.

4.2.1 Order parameters

We can recover the typical overlaps found also in the replica treatment (notice that for the law of large numbers this quantities have variance $O(N^{-1/2})$, ie. they are self-average):

$$\tilde{q} = \frac{\left\langle \tilde{W} \right\rangle^2}{N} = \frac{\sum_i \tilde{m}_i^2}{N}$$

$$s = \frac{\left\langle \tilde{W} \cdot W_1 \right\rangle}{N}$$

$$\tilde{s} = \frac{\left\langle \tilde{W} \right\rangle \cdot \langle W_1 \rangle}{N} = \frac{\sum_i \tilde{m}_i m_i}{N}$$

$$q_1 = \frac{\langle W_1 \cdot W_2 \rangle}{N}$$

$$q_0 = \frac{\langle W_1 \rangle \cdot \langle W_2 \rangle}{N} = \frac{\sum_i m_i^2}{N}$$

Moreover we should check the soft spherical constraints:

$$\tilde{Q} = \frac{\left\langle \tilde{W}^2 \right\rangle}{N} = \frac{\sum_i \tilde{\rho}_i}{N}$$

$$Q = \frac{\left\langle W^2 \right\rangle}{N} = \frac{\sum_i \rho_i}{N}$$

These can be implemented directly in the replica expression of the entropy (the scaling of the entropy under rescaling of the W's is a shift of a constant plus a rescaling of K).

S and q_1 must be computed exploiting the usual strategy for the marginal of more variables in BP, eq. 8 , in particular:

$$p(\tilde{W}_i, W_i) \propto e^{\gamma \tilde{W}_i W_i} \tilde{\nu}_{i\to\gamma}(\tilde{W}_i) \nu_{i\to\gamma}(W_i)$$

hence: $\left\langle \tilde{W}_i W_i \right\rangle = \frac{\int d\tilde{W}_i dW_i \, \tilde{W}_i W_i p(\tilde{W}_i, W_i)}{\int d\tilde{W}_i dW_i \, p(\tilde{W}_i, W_i)}$.

Shifting the W and exploiting that a gaussian of matrix A_{ij} has moments $(A^{-1})_{ij}$:

$$\left\langle \tilde{W}_i W_i \right\rangle =$$

$$= \rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma} \frac{(\frac{m_{i\to\gamma}}{\rho_{i\to\gamma}} + \frac{\tilde{m}_{i\to\gamma}}{\tilde{\rho}_{i\to\gamma}}\tilde{\rho}_{i\to\gamma}\gamma)(\frac{\tilde{m}_{i\to\gamma}}{\tilde{\rho}_{i\to\gamma}} + \frac{m_{i\to\gamma}}{\rho_{i\to\gamma}}\rho_{i\to\gamma}\gamma) - \gamma(-1 + \rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma}\gamma^2)}{(-1 + \rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma}\gamma^2)^2} =$$

$$= \frac{(\tilde{\rho}_{i\to\gamma}^{-1}\frac{m_{i\to\gamma}}{\rho_{i\to\gamma}} + \frac{\tilde{m}_{i\to\gamma}}{\tilde{\rho}_{i\to\gamma}}\gamma)(\rho_{i\to\gamma}^{-1}\frac{\tilde{m}_{i\to\gamma}}{\tilde{\rho}_{i\to\gamma}} + \frac{m_{i\to\gamma}}{\rho_{i\to\gamma}}\gamma) - \gamma(-\frac{1}{\rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma}} + \gamma^2)}{(-\frac{1}{\rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma}} + \gamma^2)^2}$$

Notice the symmetry between the tilded and not messages. The moment between replicas is obtained from the distribution:

$$p(W_i^1, W_i^2) \propto \int d\tilde{W}_i \, e^{\gamma \tilde{W}_i W_i^1} e^{\gamma \tilde{W}_i W_i^2} \nu_{i\to\gamma}(W_i^1)\nu_{i\to\gamma}(W_i^2) \frac{\tilde{\nu}_{i\to\gamma}(\tilde{W}_i)}{\tilde{\nu}_{\gamma\to i}(\tilde{W}_i)}$$

with $\frac{\tilde{\nu}_{i\to\gamma}(\tilde{W}_i)}{\tilde{\nu}_{\gamma\to i}(\tilde{W}_i)} \propto \exp\{-(\tilde{\rho}_i^{-1} - 2\tilde{\rho}_{\gamma\to i}^{-1})\frac{\tilde{W}_i^2}{2} + (\frac{\tilde{m}_i}{\tilde{\rho}_i} - 2\frac{\tilde{m}_{\gamma\to i}}{\tilde{\rho}_{\gamma\to i}})\tilde{W}_i\}$. After integration it is clear that the final result $\left\langle W_i^1 W_i^2 \right\rangle$ has the same form as $\left\langle \tilde{W}_i W_i \right\rangle$ above with the replacings: $\gamma \to \frac{\gamma^2}{\tilde{\rho}_i^{-1} - 2\tilde{\rho}_{\gamma\to i}^{-1}}$, $\frac{m_{i\to\gamma}}{\rho_{i\to\gamma}} \to \frac{\tilde{m}_{i\to\gamma}}{\tilde{\rho}_{i\to\gamma}} \to m_{i\to\gamma} + \gamma \frac{\frac{\tilde{m}_i}{\tilde{\rho}_i} - 2\frac{\tilde{m}_{\gamma\to i}}{\tilde{\rho}_{\gamma\to i}}}{\tilde{\rho}_i^{-1} - 2\tilde{\rho}_{\gamma\to i}^{-1}}$, $\rho_{i\to\gamma}^{-1} \to \tilde{\rho}_{i\to\gamma}^{-1} \to \rho_{i\to\gamma}^{-1} - \frac{\gamma^2}{\tilde{\rho}_i^{-1} - 2\tilde{\rho}_{\gamma\to i}^{-1}}$

4.2.2 Bethe free entropy

In the follow we will consider all the messages normalized. The BP estimate for the free entropy (Bethe free entropy) is given by eq. (9):

$$F_{Bethe} = F(\vec{\nu}) = \sum_{\mu\in F} F_\mu + \sum_{v\in V} F_v - \sum_{V\times F} F_{(v,\mu)}$$

The local field contibution can be absorbed in the vertex terms [10]. The factor contibution is:

$$\sum_\mu F_\mu(\overrightarrow{\nu}) = \sum_\xi (\tilde{F}_\xi + yF_\xi) + y\sum_i F_{\gamma(i)}$$

where

$$F_\xi = \log \int d^N W \, \theta(W\cdot\xi - \sqrt{N}\kappa) \prod_i \nu_{i\to\xi}(W_i) = \log H(\frac{\kappa\sqrt{N} - \sum_i m_{i\to\xi}\xi_i}{\sqrt{\sum_i \rho_{i\to\xi}}})$$

$$\tilde{F}_\xi = \log H(\frac{\kappa\sqrt{N} - \sum_i \tilde{m}_{i\to\xi}\xi_i}{\sqrt{\sum_i \tilde{\rho}_{i\to\xi}}})$$

(this should be valid for the perceptron too)

$$F_{\gamma(i)} = \log \int d\tilde{W}_i dW_i \, e^{\gamma\tilde{W}_i W_i} \tilde{\nu}_{i\to\gamma}(\tilde{W}_i)\nu_{i\to\gamma}(W_i) =$$

$$= \log \frac{\exp \gamma \frac{\tilde{m}_{i\to\gamma}^2 \rho_{i\to\gamma}\gamma + m_{i\to\gamma}^2 \tilde{\rho}_{i\to\gamma}\gamma + 2\tilde{m}_{i\to\gamma}m_{i\to\gamma}}{2(1-\rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma}\gamma^2)}}{\sqrt{1-\rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma}\gamma^2}} =$$

$$= \gamma \frac{\tilde{m}_{i\to\gamma}^2 \rho_{i\to\gamma}\gamma + m_{i\to\gamma}^2 \tilde{\rho}_{i\to\gamma}\gamma + 2\tilde{m}_{i\to\gamma}m_{i\to\gamma}}{2(1-\rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma}\gamma^2)} - \frac{1}{2}\log(1-\rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma}\gamma^2)$$

The second contribution relates to the variable nodes:

$$\sum_{v\in V} F_v = y\sum_i F_i + \sum_i \tilde{F}_i$$

where:

$$F_i = \log \int dW_i \prod_{\xi} \nu_{\xi\to i}(W_i)\nu_{\gamma\to i}(W_i) e^{-\lambda\frac{W_i^2}{2}} =$$

(this is the integral of a product of normalized gaussians

$\int dx \prod_i \frac{\exp -\frac{(x-m_i)^2}{2\rho_i}}{\sqrt{2\pi\rho_i}} = \sqrt{\frac{2\pi}{\sum_i \rho_i^{-1}}} \frac{\exp\{-\frac{1}{2}\sum_i \frac{m_i^2}{\rho_i} + \frac{1}{2}\frac{(\sum_i \frac{m_i}{\rho_i})^2}{\sum_i \rho_i^{-1}}\}}{\prod_i \sqrt{2\pi\rho_i}}$)

$$F_i = -\frac{1}{2}\sum_{\xi}\frac{m_{\xi\to i}^2}{\rho_{\xi\to i}} - \frac{1}{2}\frac{m_{\gamma\to i}^2}{\rho_{\gamma\to i}} + \frac{1}{2}\frac{(\sum_\xi \frac{m_{\xi\to i}}{\rho_{\xi\to i}} + \frac{m_{\gamma\to i}}{\rho_{\gamma\to i}})^2}{(\sum_\xi \rho_{\xi\to i}^{-1} + \rho_{\gamma\to i}^{-1} + \lambda)} -$$

$$-\frac{1}{2}(\log(\sum_\xi \rho_{\xi\to i}^{-1} + \rho_{\gamma\to i}^{-1} + \lambda) + \sum_\xi \log\rho_{\xi\to i} + \log\rho_{\gamma\to i}) - \frac{M}{2}\log(2\pi) =$$

$$= -\frac{1}{2}\sum_\xi \frac{m_{\xi\to i}^2}{\rho_{\xi\to i}} - \frac{1}{2}\frac{m_{\gamma\to i}^2}{\rho_{\gamma\to i}} + \frac{1}{2}\frac{m_i^2}{\rho_i} - \frac{1}{2}(-\log\rho_i + \sum_\xi \log\rho_{\xi\to i} + \log\rho_{\gamma\to i}) - \frac{M}{2}\log(2\pi)$$

Similarly:

$$\tilde{F}_i = \log\int d\tilde{W}_i \prod_\xi \nu_{\xi\to i}(\tilde{W}_i)\left[\nu_{\gamma\to i}(\tilde{W}_i)\right]^y e^{-\tilde{\lambda}\frac{\tilde{W}_i^2}{2}} =$$

$$= -\frac{1}{2}\sum_\xi \frac{\tilde{m}_{\xi\to i}^2}{\tilde{\rho}_{\xi\to i}} - \frac{1}{2}y\frac{\tilde{m}_{\gamma\to i}^2}{\tilde{\rho}_{\gamma\to i}} + \frac{1}{2}\frac{\tilde{m}_i^2}{\tilde{\rho}_i} - \frac{1}{2}(-\log\tilde{\rho}_i + \sum_\xi \log\tilde{\rho}_{\xi\to i} + y\log\tilde{\rho}_{\gamma\to i}) -$$

$$-\frac{(M+y-1)}{2}\log(2\pi)$$

The last contribution comes from the edges:

$$\sum_{V\times F} F_{(v,\mu)} = y\sum_i \tilde{F}_{i,\gamma} + y\sum_i F_{i,\gamma} + \sum_i\sum_\xi \tilde{F}_{i,\xi} + y\sum_i\sum_\xi F_{i,\xi}$$

$$F_{i,\gamma} = \log\int dW_i\, \nu_{\gamma\to i}(W_i)\nu_{i\to\gamma}(W_i) =$$

$$= -\frac{1}{2}\frac{m_{\gamma\to i}^2}{\rho_{\gamma\to i}} - \frac{1}{2}\frac{m_{i\to\gamma}^2}{\rho_{i\to\gamma}} + \frac{1}{2}\frac{m_i^2}{\rho_i} - \frac{1}{2}(-\log\rho_i + \log\rho_{\gamma\to i} + \log\rho_{i\to\gamma}) - \frac{1}{2}\log(2\pi)$$

$$\tilde{F}_{i,\gamma} = -\frac{1}{2}\frac{\tilde{m}^2_{\gamma\to i}}{\tilde{\rho}_{\gamma\to i}} - \frac{1}{2}\frac{\tilde{m}^2_{i\to\gamma}}{\tilde{\rho}_{i\to\gamma}} + \frac{1}{2}\frac{\tilde{m}^2_{i}}{\tilde{\rho}_{i}} - \frac{1}{2}(-\log\tilde{\rho}_i + \log\tilde{\rho}_{\gamma\to i} + \log\tilde{\rho}_{i\to\gamma}) - \frac{1}{2}\log(2\pi)$$

$$F_{i,\xi} = \log\int dW_i\, \nu_{\xi\to i}(W_i)\nu_{\xi\to i}(W_i) =$$

$$= -\frac{1}{2}\frac{m^2_{\xi\to i}}{\rho_{\xi\to i}} - \frac{1}{2}\frac{m^2_{i\to\xi}}{\rho_{i\to\xi}} + \frac{1}{2}\frac{m^2_{i}}{\rho_{i}} - \frac{1}{2}(-\log\rho_i + \log\rho_{\xi\to i} + \log\rho_{i\to\xi}) - \frac{1}{2}\log(2\pi)$$

$$\tilde{F}_{i,\xi} = -\frac{1}{2}\frac{\tilde{m}^2_{\xi\to i}}{\tilde{\rho}_{\xi\to i}} - \frac{1}{2}\frac{\tilde{m}^2_{i\to\xi}}{\tilde{\rho}_{i\to\xi}} + \frac{1}{2}\frac{\tilde{m}^2_{i}}{\tilde{\rho}_{i}} - \frac{1}{2}(-\log\tilde{\rho}_i + \log\tilde{\rho}_{\xi\to i} + \log\tilde{\rho}_{i\to\xi}) - \frac{1}{2}\log(2\pi)$$

4.2.3 Local entropy

The above Bethe free entropy F_{Bethe} is an estimate for the free entropy F of the system $\tilde{W}$ of inverse temperature y and energy $-E(\tilde{W},\tilde{\lambda},\lambda,\gamma) = -\frac{\tilde{\lambda}}{y}\tilde{W}^2 + \log\aleph(\tilde{W},\gamma) = -\frac{\tilde{\lambda}}{y}\tilde{W}^2 + \log\int dW\,\chi_\xi(W)\, e^{-\lambda W^2+\gamma W\tilde{W}} \simeq N(-\frac{1}{2}\frac{\tilde{\lambda}}{y}\tilde{Q} + S_{loc}(\gamma,\lambda,\tilde{\lambda}) - \frac{1}{2}\lambda Q + \gamma s)$ where we have used the saddle point method meaning that the last equality is the saddle-point energy of the large-deviation $\tilde{W}$'s , while S_{loc} denotes the *local entropy*, i.e. $e^{NS_{loc}}$ is the number of solutions around such super-solutions. To each Lagrange parameter $\gamma,\lambda,\tilde{\lambda}$ it corresponds a typical overlap $s, Q, \tilde{Q}$ so that it is possible to invert these relation and get $S_{loc}(s,Q,\tilde{Q}) \simeq \frac{-E_{bethe}(\tilde{\lambda}(\tilde{Q}),\lambda(Q),\gamma(s))}{N} + \frac{1}{2}\frac{\tilde{\lambda}(\tilde{Q})\tilde{Q}}{y} + \frac{1}{2}\lambda(Q)Q - \gamma(s)s$. In practice one adjusts $\lambda,\tilde{\lambda}$ in running-time so to keep $Q,\tilde{Q}=1$.

For $y\to\infty$ $-yE_{Bethe} = F_{Bethe}$ [21] because one admits that the ground states are present in non-exponential number, while for finite y:

$$S_{loc}(y,s) \equiv \frac{1}{N}\left\langle\log\aleph(s)\right\rangle_{s,y} = -\frac{1}{N}\left\langle E\right\rangle_{s,y} = \frac{1}{N}\partial_y\log Z(s,y) = \frac{1}{N}\partial_y F_{Bethe}(s,y)$$

where, as discussed above:

$$\frac{1}{N}F_{Bethe}(s,y,Q,\tilde{Q}) = \frac{F_{bethe}(\tilde{\lambda}(\tilde{Q}),\lambda(Q),\gamma(s))}{N} + \frac{1}{2}\tilde{\lambda}(\tilde{Q})\tilde{Q} + \frac{1}{2}y\lambda(Q)Q - y\gamma(s)s$$

so that:

$$S_{loc}(y,s) = \frac{1}{N}\partial_y F_{Bethe}(\tilde{\lambda}(\tilde{Q}),\lambda(Q),\gamma(s),y) + \frac{1}{2}\lambda(Q)Q - \gamma(s)s$$

The local entropy at $y=0$ is the Franz-Parisi potential.

[21] $free\,entropy \equiv \log Z = S - \frac{1}{T}E$

Now the derivative $\partial_y F_{Bethe}(\tilde{\lambda}(\tilde{Q}), \lambda(Q), \gamma(s), y)$ has to be computed keeping in mind that $\frac{\delta F_{Bethe}}{\delta m} = 0$ for the very definition of BP equations[22]. So we can perform the explicit derivative of the Bethe free entropy with respect to y:

$$\partial_y F_{Bethe}(\tilde{\lambda}, \lambda, \gamma, y) = \sum_{\xi} F_{\xi} + \sum_{i} F_{\gamma(i)} +$$

$$+ \sum_{i} F_i + \sum_{i} \partial_y \tilde{F}_i +$$

$$- \sum_{i} F_{i,\gamma} - \sum_{i} \tilde{F}_{i,\gamma} - \sum_{i,\xi} F_{i,\xi}$$

with

$$\partial_y \tilde{F}_i = \partial_y \log \int d\tilde{W}_i \prod_{\xi} \tilde{\nu}_{\xi \to i}(\tilde{W}_i) \left[\tilde{\nu}_{\gamma \to i}(\tilde{W}_i)\right]^y e^{-\tilde{\lambda}\frac{\tilde{W}_i^2}{2}} =$$

$$= \frac{\int d\tilde{W}_i \prod_{\xi} \tilde{\nu}_{\xi \to i}(\tilde{W}_i) \log \tilde{\nu}_{\gamma \to i}(\tilde{W}_i) \left[\tilde{\nu}_{\gamma \to i}(\tilde{W}_i)\right]^y e^{-\tilde{\lambda}\frac{\tilde{W}_i^2}{2}}}{\int d\tilde{W}_i \prod_{\xi} \tilde{\nu}_{\xi \to i}(\tilde{W}_i) \left[\tilde{\nu}_{\gamma \to i}(\tilde{W}_i)\right]^y e^{-\tilde{\lambda}\frac{\tilde{W}_i^2}{2}}} = \left\langle \log \tilde{\nu}_{\gamma \to i}(\tilde{W}_i) \right\rangle_{\tilde{m}_i, \tilde{\rho}_i} =$$

$$= -\frac{\tilde{\rho}_i + \tilde{m}_i^2 - 2\tilde{m}_i \tilde{m}_{\gamma \to i} + \tilde{m}_{\gamma \to i}^2}{2\tilde{\rho}_{\gamma \to i}} - \frac{1}{2}\log \tilde{\rho}_{\gamma \to i} - \frac{1}{2}\log 2\pi$$

Cancellations

There are many cancellations due to the last three terms:

$$\partial_y F_{Bethe}(\tilde{\lambda}, \lambda, \gamma, y) = \sum_{\xi} F_{\xi} + \sum_{i} F_{\gamma(i)} +$$

$$+ \sum_{i} F_i + \sum_{i} \partial_y \tilde{F}_i +$$

$$- \sum_{i} F_{i,\gamma} - \sum_{i} \tilde{F}_{i,\gamma} - \sum_{i,\xi} F_{i,\xi}$$

with:

$$F_{\xi} = \log H\left(\frac{\kappa\sqrt{N} - \sum_i m_{i \to \xi}\xi_i}{\sqrt{\sum_i \rho_{i \to \xi}}}\right)$$

$$F_{\gamma(i)} = \gamma \frac{\tilde{m}_{i \to \gamma}^2 \rho_{i \to \gamma}\gamma + m_{i \to \gamma}^2 \tilde{\rho}_{i \to \gamma}\gamma + 2\tilde{m}_{i \to \gamma} m_{i \to \gamma}}{2(1 - \rho_{i \to \gamma}\tilde{\rho}_{i \to \gamma}\gamma^2)} - \frac{1}{2}\log(1 - \rho_{i \to \gamma}\tilde{\rho}_{i \to \gamma}\gamma^2)$$

[22] and also $\partial_\lambda F \cdot \partial_y \lambda = Q \cdot \partial_y \lambda$ so that the two cancel out, and the same for the other Lagrange multipliers.

$$F_i = -\frac{1}{2}\sum_{\xi}\frac{m_{\xi\to i}^2}{\rho_{\xi\to i}} - \frac{1}{2}\frac{m_{\gamma\to i}^2}{\rho_{\gamma\to i}} + \frac{1}{2}\frac{m_i^2}{\rho_i} - \frac{1}{2}(-\log\rho_i + \sum_{\xi}\log\rho_{\xi\to i} + \log\rho_{\gamma\to i}) - \frac{M}{2}\log(2\pi)$$

$$\partial_y \tilde{F}_i = -\frac{\tilde{\rho}_i + \tilde{m}_i^2 - 2\tilde{m}_i\tilde{m}_{\gamma\to i} + \tilde{m}_{\gamma\to i}^2}{2\tilde{\rho}_{\gamma\to i}} - \frac{1}{2}\log\tilde{\rho}_{\gamma\to i} - \frac{1}{2}\log 2\pi$$

$$F_{i,\gamma} = -\frac{1}{2}\frac{m_{\gamma\to i}^2}{\rho_{\gamma\to i}} - \frac{1}{2}\frac{m_{i\to\gamma}^2}{\rho_{i\to\gamma}} + \frac{1}{2}\frac{m_i^2}{\rho_i} - \frac{1}{2}(-\log\rho_i + \log\rho_{\gamma\to i} + \log\rho_{i\to\gamma}) - \frac{1}{2}\log(2\pi)$$

$$\tilde{F}_{i,\gamma} = -\frac{1}{2}\frac{\tilde{m}_{\gamma\to i}^2}{\tilde{\rho}_{\gamma\to i}} - \frac{1}{2}\frac{\tilde{m}_{i\to\gamma}^2}{\tilde{\rho}_{i\to\gamma}} + \frac{1}{2}\frac{\tilde{m}_i^2}{\tilde{\rho}_i} - \frac{1}{2}(-\log\tilde{\rho}_i + \log\tilde{\rho}_{\gamma\to i} + \log\tilde{\rho}_{i\to\gamma}) - \frac{1}{2}\log(2\pi)$$

$$F_{i,\xi} = -\frac{1}{2}\frac{m_{\xi\to i}^2}{\rho_{\xi\to i}} - \frac{1}{2}\frac{m_{i\to\xi}^2}{\rho_{i\to\xi}} + \frac{1}{2}\frac{m_i^2}{\rho_i} - \frac{1}{2}(-\log\rho_i + \log\rho_{\xi\to i} + \log\rho_{i\to\xi}) - \frac{1}{2}\log(2\pi)$$

Final expression of the local entropy

$$\partial_y F_{Bethe}(\tilde{\lambda}, \lambda, \gamma, y) = \sum_{\xi} F_{\xi} + \sum_{i} F_{\gamma(i)} +$$

$$+ \sum_{i} F_i' + \sum_{i} \partial_y \tilde{F}_i +$$

$$- \sum_{i} F_{i,\gamma}' - \sum_{i} \tilde{F}_{i,\gamma}' - \sum_{i,\xi} F_{i,\xi}'$$

with

$$F_{\xi} = \log H\left(\frac{\kappa\sqrt{N} - \sum_i m_{i\to\xi}\xi_i}{\sqrt{\sum_i \rho_{i\to\xi}}}\right)$$

$$F_{\gamma(i)} = \gamma\frac{\tilde{m}_{i\to\gamma}^2\rho_{i\to\gamma}\gamma + m_{i\to\gamma}^2\tilde{\rho}_{i\to\gamma}\gamma + 2\tilde{m}_{i\to\gamma}m_{i\to\gamma}}{2(1 - \rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma}\gamma^2)} - \frac{1}{2}\log(1 - \rho_{i\to\gamma}\tilde{\rho}_{i\to\gamma}\gamma^2)$$

$$F_i' = 0$$

$$\partial_y \tilde{F}_i' = -\frac{\tilde{\rho}_i + \tilde{m}_i^2 - 2\tilde{m}_i\tilde{m}_{\gamma\to i}}{2\tilde{\rho}_{\gamma\to i}}$$

$$F_{i,\gamma}' = -\frac{1}{2}\frac{m_{i\to\gamma}^2}{\rho_{i\to\gamma}} - \frac{1}{2}\log\rho_{i\to\gamma} - \frac{1}{2}\log(2\pi)$$

$$\tilde{F}'_{i,\gamma} = -\frac{1}{2}\frac{\tilde{m}^2_{i\to\gamma}}{\tilde{\rho}_{i\to\gamma}} + \frac{1}{2}\frac{\tilde{m}^2_i}{\tilde{\rho}_i} - \frac{1}{2}(-\log\tilde{\rho}_i + +\log\tilde{\rho}_{i\to\gamma})$$

$$F'_{i,\xi} = -\frac{1}{2}\frac{m^2_{i\to\xi}}{\rho_{i\to\xi}} + \frac{1}{2}\frac{m^2_i}{\rho_i} + \frac{1}{2}\log\rho_i - \frac{1}{2}\log\rho_{i\to\xi}$$

4.2.4 Results

We run the above version of replicated BP (namely it is analogue to a focusing BP but for the fact that the center has not been traced out). Unluckily, only for $\kappa = 0$ we managed to make the algorithm converge. Actually, for $\kappa = 0$, only the direction of the synaptic weight vector matters, and the norm is completely controlled by a balanced interplay of the γ and λ driven interactions. The problems arise with $\kappa \lesseqgtr 0$: here the norm matters for the pattern constraint satisfaction and while the spherical constraint is set in a soft way, the energy of a pattern is a 0 or $+\infty$ function, corresponding to the fact that we are at zero temperature. In the discrete case these problems didn' t arise. The fix would be considering the finite temperature system, but this goes beyond the scope of this work.

We report the results at zero stability. The procedure is starting with $\gamma = 0$ and run BP until convergence; start from the converged messages increasing or decreasing γ and run BP again, to explore all the range of distances. Indeed, the $\gamma = 0$ case is equivalent to standard BP (uncoupled real replicas) and yields the Gardner entropy. Increasing γ means tuning on smaller distances. Still, there are convergence problems out of a range of distances, in particular we observe that the messages stop converging. We think this is once again due to the fact that at such non-typical overlaps there are few solutions and the system prefers to minimize the number of violated patterns irrespective of the soft Lagrange interactions.

We report two series of results in the range of distances where the algorithm converged, see Fig. 29:

- at $y \to 0$ the local entropy should reduce to the Franz-Parisi potential; in practice we expect the replicated BP estimate to be reliable only in the large y limit (this is observed also in the discrete case)
- $y = 3$: the result is reasonable (the Franz-Parisi entropy is a lower bound), but we cannot say if the local entropy stands above the Franz-Parisi entropy due to large finite-size effect errors (fluctuations in different samples)

In the end, the general approach may be promising, the main troubles coming from:

- the zero temperature issue, which can be fixed modifying the BP equations (we have sketch the computation and it is feasible in theory)

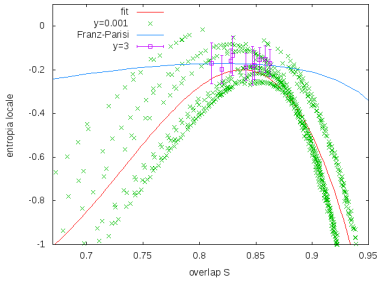

Figure 29: Local entropy estimated with replicated BP at $y = 0.001$ and $y = 3$; $\alpha = 1.5, \kappa = 0, N = 600$. For small y the approach is expected to fail, and actually only the region $s \simeq s_{typ}$ ($\gamma = 0$, decouple system hence y independent) seems acceptable. In green we have reported the data taken in different simulations in order to give an idea of how big the fluctuations are in different samples. At $y = 3.$, instead, the approach is theoretically consistent and in the region in which the meassages converge the local entropy estimate follows the Fran-Parisi entropy (which is a lower bound), see purple points, which are obtained averaging over 10 simulations the experimental points $(s(\gamma), S_{loc}(\gamma)$ for different values of γ

- the geometry of the space; we don' t know if this is a real problem but one possible picture is that, if two replicas live in two different minima separated by an energy barrier, their interaction seems quite different from the discrete case

4.3 Numerical experiments

Seen the not promising results obtained so far, we turned to numerical experiments to see where the numerical hardness arises and if with the strategies proposed in Section 2.2, the more adaptable being replicating known algorithms, we could solve this problem.

We started with Gradient Descent (GD), recall eq. (1). Actually, it is known in the deep learning literature [25, 40] that simple GD fails in the training of neural networks, mainly for the following reasons:

- it gets trapped in local minima with high training error or it slows down in plateaus [42]
- even when it achieves low training error, it yields poor generalization properties with respect to simple strategies as SGD
- in practice one deals with large networks and enormous training sets of correlated patterns, so computing the gradient at each step using the whole training set is unfeasible and not convienient

In particular, in [40] they introduced a replicated version of SGD and compare its performances with existing algorithms on a 6 layer network. In their language[23], y workers are subject to the potential generated by the patterns and a are coulpled to a master by an elastic interaction. Apart from reasons connected to the parallelization of the code, the intuition behind this method is threefold:

- an entropic argument: more workers provide a better exploration of the energetic landscape
- the single workers may be attracted towards local minima, but the best (deeper) minimum should provide a stronger attraction
- in the final phase of convergence, having more workers should speed up the collapse in the minimum, due to an average over fluctuations (indeed the algorithm is called *Elastic Averaging SGD* (EASGD))

It is not clear if the success of this procedure has anything to do with the existence of dense clusters of solutions or something like that. Another question concerns the role of stochasticity in SGD: one may think that the noise selects the more robust solutions, so that already SGD may be connected to an out-of-equilibrium landscape.

Said this, we tried GD for the storage problem in the continuous perceptron, expecting the GD to get trapped in local minima, at least in the $\kappa < 0$ region where the space of solutions is no more convex. However, we were surprised to see that no computational hardness seems to be present even in the fullRSB region.

In particular, we focused on the region $\kappa = -1, \alpha = 11.$ and at this point a simple GD with a proper choice of the learning, see Appendix , rate succeeds in 100% of cases. Notice that at $\kappa = -1$ the storage capacity is expected to be at around $\alpha_c \sim 13$ while the AT line is at about $\alpha \simeq 7$. [43]. For greater α it is not possible to make statements due to the stretching of convergence times. The numerical results suggest that with probability 1 all local minima are also solutions. We have not been able to prove this conjecture. Nevertheless, we mention that recently some papers have been published [14, 44] which claim that in certain regimes in deep learning all local minima are global ones or are low energy ones [45].

In Fig. 30 we plot the mutual overlap between solutions found with GD: in practice we fix a sample and repeat GD from 120 different initialization points, until we find a solution. We kept in memory the corresponding vector and computed the overlap between all pairs of solutions. Such overlaps are plotted in the histogram. The scenario seems to be robust in N; the number of iterations needed to find a solution doesn't seem to scale in N but depends only on α (while the complexity of the single iteration is order $O(\alpha N^2)$.

[23] actually one of their main concern is having an highly distributed implementation of the algorithm

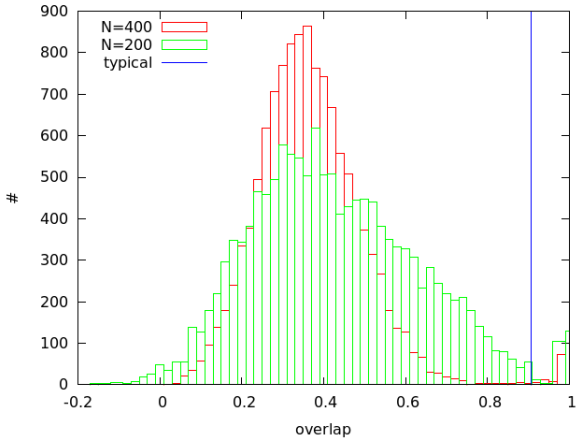

Figure 30: Overlaps between pairs of the 120 solutions found with GD over the same sample, with random initialization points, for $\kappa = -1, \alpha = 11.$ and $N = 200, 400$. The small peak near 1 is maybe due to solutions in the same connected component and the scaling with N suggests that the algotithmic overlap is a self-averaging quantity.

4.3.1 Replicated GD

Even though the storage problem doesn' t seem hard, we implemented replicated GD in order to understand if the solutions found with GD and replicated GD show the same features.

Replicating GD consists in dealing with y replicas of the system and at each iteration to update each copy with the usual GD rule plus an elastic term with coupling constant $\gamma(t)$.

In order to keep things under control, we studied different scaling properties of the dynamics of the algorithm, monitoring in time the (average) overlap between different replicas (used γ constant or step-like for simplicity), see Figs. 31,32.

Once checked the scalings, we studied the statistical properties of solutions. Making the histogram of the overlaps between pairs of solutions, no qualitative difference is observed, see Fig. 33.

4.3.2 Discussion

The overall results seem to suggest that the continuous perceptron does not show the same features as the discrete one. Even though both models can be treated with a very simialr replica computation, their dynamical behaviour is totally different. The discrete perceptron is in fact a spin glass model, numerically hard to solve with energy minimization based methods. The continuous perceptron doesn' t show this hardness and can be solved with a simple gradient descent.

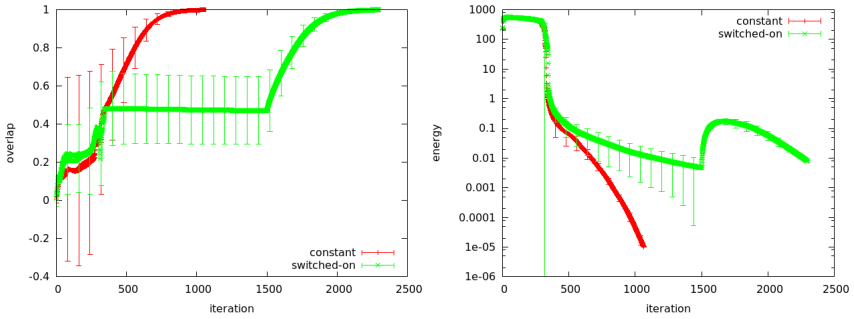

Figure 31: Left panel: Typical trajectory of the overlap between replicas vs time. As overlap we have taken the average of the overlaps of one replicas with the others. In the red dataset we have kept γ constant in time, while in the green one we have turned it on after 1500 iterations. Here $\gamma = 0.001$, $\alpha = 10., \kappa = -1., N = 300$. The green plot suggests that after about 300 iterations each free replica selects a basin of attraction. Right panel: Average energy (quadratric loss function) of the replicas versus time for the same two simulations. Notice that in the green simulation at the accension of the interaction the replicas come out of their basin of attraction and converge to the same region.

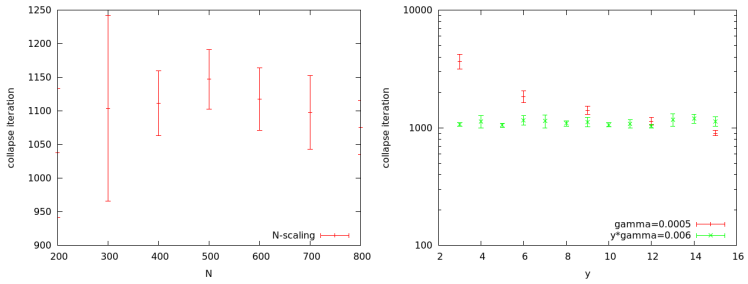

Figure 32: Scaling of the dynamics with the parameters. In the left panel the number of iterations required to have the replicas with mutual overlaps greater than 0.999 ("collapse iteration") are plotted versus the system size, averaged over 10 samples and at $y = 6, \gamma = 0.001$. The parameters of the algorithm are properly normalized in N: the typical time scales of the dynamics do not depend on N. In the right panel the collapse iteration is plotted versus y both at fixed $\gamma = 0.0005$ (red line, exponential-like scaling) and with $\gamma = 0.006/y$. The more numerous the replicas, the faster they collapse in the same point and, moreover, as suggested from the large deviation measure, only the product $y * \gamma$ matters.

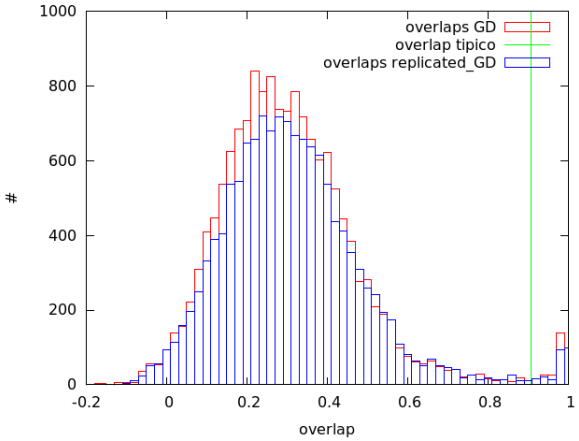

Figure 33: Overlaps between pairs of solutions found with GD versus replicated GD, at $\alpha = 11., \kappa = -1., N = 200, y = 7$. No qualitative difference.

The space of solutions is very different too: in the discrete case the typical solution are isolated, in the continuous case for continuity this is not true. The geometrical structure of the sphere seems also to play a role, with the spherical constraint being a key element.

We had chosen the negative perceptron because it was quite analytically treatable and hoped it was complex enough to show the precursor features of deeper learning problems. The results obtained suggest that if the success of deep neural networks is somehow connected to dense regions of "desirable" configurations, this property should arise from the deep nature of the architecture . The same idea of "desirable" may need to be rethought: maybe in the continuous case we have to seek for dense regions of configurations with low training error and not necessarily solutions, and it is not clear if it is the volume occupied by these configurations to be relevant or the volume spanned by (in which they are dense enough).

We have some arguments that the situation may be more interesting with deeper architectures. As above mentioned, we have some reasons to hope that the flexible "replicated algorithm" strategy may yield better performances [40]. Already in simple two-layer networks (with no artificial negative stability) the space of solutions is highly non-trivial [19, 18]: for the tree-like committee machine, for example, the intermediate output $\vec{\tau} = \{\tau_1, ..., \tau_2\}$, $\tau_i = \pm 1$ is called *internal representation* and is used to label the (domains of) solutions (solution means $sign(\sum_i \tau_i) > 0$). In the generalization problem for fully connected committees, both the discrete and continuous case show an interesting transition from a symmetric phase with unspecialized hidden units to a symmetry broken phase with specialized hidden units [17]. At the present time we are working at simulations on tree-like committee machines, to understand if the training problem is computationally hard.

We don' t know if the large deviation approach has nothing to do with deep learning, but we think that it is worth investing some time in investigating this issue.

Conclusions

These months in Torino have been an amazing research experience for me. I have joined a well structured group with a longstanding tradition in the physics of spin-glasses and optimization problems. Last year was for the group a really fruitful period: with an elegant conjecture they were able to describe the out-of-equilibrium behaviour of efficient learning routines on discrete synapses and then sketch a simple and very flexible strategy to design new algorithms. The generality of the approach opens several perspectives and a lot of work will be required to investigate the innumerable possible research lines (see paragraph 2.3).

I had to study a lot of advanced tools in statistical mechanics (BP and replica method in primis) and finally was directed to work on one of the main projects of the group: extending the achivements obtained in the discrete case to the continuous one, starting from the simplest possible scenario. In the end, the negative perceptron turned out to be unsuitable for our purposes: it has no computational hardness and the negative stability is only source of troubles. We now turn our hopes to deeper architectures (see discussion 4.3.2). All the same, the work done on the perceptron will be useful for a twofold reason: as a training for future technical computations and as a precursor of the possible problems to be encountered dealing with continuous synapses (the main question being: what is the continuum analogous of the attractive dense clusters of solutions?).

I did a lot of complex computations and learned one major lesson: always start from the simple stuff and redo what has already been done; this will give you the right momentum and will allow you to check difficult equations in simple regimes.

Appendix

A. Gradient Descent Code

We report a readable version of the Julia code of the GD algorithm. The important remarks are:

- the energy is defined as the sum over the patterns of the squared deviation from the threshold (corresponding to $r = 2$ in eq. (2))
- a small extra stability ϵ (typicall $\epsilon = 10^{-3}, 10^{-4}$) helps
- the synaptic weights are normalized at each step, and the trasversal component of the gradient is taken

```
######## GD ########
function GDescent(sist::Sistema,  ϵ::Float64)

    W0=Array(Float64,sist.N) ### random initialization
    for i in 1:sist.N
         W0[i]=rand_normal(0.,1.) ### of each synapsis with a Gaussian weight
    end

    pars = SGDpar(batch_size,ϵ,0.4, Array(Float64,sist.N))
    pars.W = deepcopy(W0)  ## in julia by default the array are treated by reference
    W = deepcopy(W0)    ## deepcopy is required to copy the VALUES of the array
    Δ,it,E_vera,E = 1.,0,1,0.

    ∇W = Array(Float64, sist.N) #array containing the gradient
    ∇T = Array(Float64, sist.N) #array containing the transversal component of the gradient

    while it <20000

        ∇Told = deepcopy(∇T)
        E, E_vera = gradiente!(∇W, pars, sist) ### compute the gradient and the energy
        ########## update rule of the learning rate #############
        if it < 10000
            pars.η=max(1./(sqrt(1+it/30)), 0.2*log(it/600))
        else
            pars.η=0.5
        end
        ######################################
        ∇T=∇W-pars.W*(pars.W⋅∇W/sist.N)   #transversal component of the gradient
        W = pars.W - pars.η*(∇T)         ## GD update rule
        W = W/sqrt((W⋅W)/sist.N)        ## normalization
        Δ = norm(W-pars.W)/sqrt(sist.N)    ## update magnitude
        pars.W = deepcopy(W)
        ## intermediate prints
        println(it, "  ", E, "   ", E_vera,"   ", Δ, "   ", pars.η, "   ", ∇W⋅∇W/sist.N )

        if E_vera == 0   ####### solution condition
            println("ok in $it passi")
            return (deepcopy(pars.W), it, E_vera)
        end
        it+=1

    end
    return (deepcopy(pars.W), it, E_vera)
end
```

The complexity of each iteration is completely contained in the gradiente! function and is $O(\alpha N^2)$, corresponding to a loop over each site and each pattern:

```
63  function gradiente!(gradiente, pars::SGDpar, sist::Sistema)
64
65      E = 0.
66      E_vera = 0
67      soglia = (sist.K+pars.ϵ)*sqrt(sist.N)
68      soglia_vera = (sist.K)*sqrt(sist.N)
69
70      rangeμ = randsubseq([1:sist.M;], pars.batch_size)
71      prodotto = 0.
72
73      for i in 1:sist.N
74              gradiente[i] =0.
75      end
76
77      for μ in 1:length(rangeμ)
78          prodotto = 0.
79          for i in 1:sist.N
80              prodotto+=sist.ξ[i,rangeμ[μ]]*pars.W[i]
81          end
82
83          E+= (prodotto>soglia ? 0. : (prodotto-soglia)^2/sist.N  )
84          E_vera += (prodotto>soglia_vera ? 0 : 1 )
85          for i in 1:sist.N
86                  gradiente[i] += (prodotto>soglia ? 0. : 2*sist.ξ[i,rangeμ[μ]]*(prodotto-soglia)/sist.N  )
87          end
88      end
89
90      return E, E_vera
```

Acknowledgements

I thank my colleagues and friends: these months of research together have been real fun. In particular, I am grateful to Riccardo Zecchina and Sergio Caracciolo, who have made this experience possible. Finally, a special praise for Carlo Lucibello, the best tutor one could hope for.

References

[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, sep 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):023301, 2016.

[3] C. Baldassi, C. Borgs, J. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina. Unreasonable Effectiveness of Learning Neural Nets: Accessible States and Robust Ensembles. *arXiv:1605.06444*, 2016.

[4] S.F. Edwards and P.W. Anderson. Theory of spin glasses. *Journal of Physics F: Metal Physics*, 5:965–974, 1975.

[5] A.Virasoro M.Mezard, G.Parisi. *Spin Glass Theory and Beyond*, volume 2016. feb 2016.

[6] M. Mezard and G. Parisi. Replicas and optimization. *Journal de Physique Lettres*, 46(17), 1985.

[7] E. Gardner. Optimal storage properties of neural network models. *J. Phys. A. Math. Gen.*, 21, 1988.

[8] C. Van den Broeck A. Engel. *Statistical Mechanics of Learning*. 2001.

[9] M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

[10] M. Mezard and A. Montanari. *Information,Physics, and Computation*. 2009.

[11] J. Yedidia. Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 51(7), 2005.

[12] D. Bayati, M. Shah and M. Sharma. A Simpler Max-Product Maximum Weight Matching Algorithm and the Auction Algorithm. *IEEE International Symposium on Information Theory*, 2006.

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

[14] Ian Goodfellow Yoshua Bengio and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

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

[17] John Hertz and Holm Schwarze. Generalization in large committee machines. *Physica A: Statistical Mechanics and its Applications*, 200(1):563 – 569, 1993.

[18] R. Monasson and R. Zecchina. Weight Space Structure and Internal Representations: A Direct Approach to Learning and Generalization in Multilayer Neural Networks. *Physical Review Letters*, 75:2432–2435, September 1995.

[19] R. Monasson and D. O'Kane. Domains of solutions and replica symmetry breaking in multilayer neural networks. *EPL (Europhysics Letters)*, 27(2):85, 1994.

[20] W. Krauth and M. Mezard. Storage capacity of memory networks with binary couplings. *Journal de Physique*, 50(20), 1989.

[21] L. Zdeborová. *Statistical Physics of Hard Optimization Problems*. PhD thesis, PhD Thesis, 2008, 2008.

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

[23] G. O'Connor, D. Wittenberg, and S. Wang. Graded bidirectional synaptic plasticity is composed of switch-like unitary events. *Proceedings of the National Academy of Sciences*, 102(7), 2005.

[24] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or -1. *arXiv:1602.02830v3*, 052313, 2016.

[25] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *Aistats*, 9, 2010.

[26] S. Franz and G. Parisi. The simplest model of jamming. *Journal of Physics A: Mathematical and Theoretical*, 49(14), 2016.

[27] M. Bouten. Replica symmetry instability in perceptron models. *J. Phys. A. Math. Gen.*, 27(1594), 1994.

[28] D. Sherrington and S. Kirkpatrick. Solvable model of a spin-glass. *Physics Review Letters*, 35(26):1792–1796, 1975.

[29] R. L. de Almeida and D. J. Thouless. Stability of the Sherrington-Kirkpatrick solution of a spin glass model. *J. Phys. A*, 11:983, 1978.

[30] G. Parisi. Toward a mean field theory for spin glasses. *Physics Review Letters*, 37(3), 1979.

[31] G. Györgyi. Beyond storage capacity in a single model neuron: Continuous replica symmetry breaking. *Journal of Statistical Physics*, 101(1/2):679–702, 2000.

[32] C. Caracciolo, S. Lucibello, G. Parisi, and G. Sicuro. Scaling hypothesis for the Euclidean bipartite matching problem. *Phys. Rev. E*, 90(012118), 2014.

[33] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

[34] Haiping Huang and Yoshiyuki Kabashima. Origin of the computational hardness for learning with binary synapses. *Phys. Rev. E*, 90:052813, Nov 2014.

[35] S. Franz and G. Parisi. Recipes for Metastable States in Spin Glasses. *Journal de Physique I*, 5:1401–1415, November 1995.

[36] T. Castellani and A. Cavagna. Spin-glass theory for pedestrians. *Journal of Statistical Mechanics: Theory and Experiment*, 5:05012, May 2005.

[37] L. Zdeborová and F. Krzakala. Generalization of the cavity method for adiabatic evolution of Gibbs states. , 81(22):224205, June 2010.

[38] H. Sompolinsky, N. Tishby, and H. S. Seung. Learning from examples in large neural networks. *Phys. Rev. Lett.*, 65:1683–1686, Sep 1990.

[39] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6), 1953.

[40] S. Zhang, A. Choromanska, and Y. LeCun. Deep learning with Elastic Averaging SGD. *ArXiv e-prints*, December 2014.

[41] J. J. W. H. Sørensen, M. K. Pedersen, M. Munch, P. Haikka, J. H. Jensen, T. Planke, M. G. Andreasen, M. Gajdacz, K. Mølmer, A. Lieberoth, and J. F. Sherson. Exploring the quantum speed limit with computer games. , 532:210–213, April 2016.

[42] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural Comput.*, 10(2):251–276, February 1998.

[43] Silvio Franz and Giorgio Parisi. The simplest model of jamming. *ArXiv e-prints*, pages 1–8, jan 2015.

[44] K. Kawaguchi. Deep Learning without Poor Local Minima. *ArXiv e-prints*, May 2016.

[45] Anna Choromanska, Mikael Henaff, Michaël Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surface of multilayer networks. *CoRR*, abs/1412.0233, 2014.