UNIVERSITA' COMMERCIALE "LUIGI BOCCONI"

PhD SCHOOL

PhD program in Statistics and Computer Science
Cycle: 36^{th}
Disciplinary Field Code FIS/02

STATISTICAL PHYSICS METHODS FOR NON-CONVEX NEURAL NETWORK MODELS

Advisor: Carlo Lucibello
Co-Advisor: Riccardo Zecchina

PhD Thesis by

Brandon Livio ANNESI

ID number 3133496

2025

Abstract:

In this doctoral thesis I employ the lens of Statistical Physics to study a number of non-convex models of Neural Networks. I start by using the replica method to investigate the loss landscape of a prototypical neural network model, the *Negative Perceptron*, and use the tool of *Linear Mode Connectivity* to describe the connection of different types of solutions. I show that the geometry of such solutions can be described as star-shaped, and numerically verify that such connectivity properties hold for solutions found by algorithms. In the same model, and for the *Tree-Committee Machine*, I study the critical capacity under the full-RSB ansatz, and settle a long standing open problem about the numerical value of such threshold. Comparing it to simulations with Gradient Descent, I observe an algorithmic gap: for some values of the constraint density solutions exists but are not found by the algorithm. I also introduce a transition line that separates a phase where typical states exhibit an Overlap Gap from a phase where no such gap exists, and discuss potential algorithmic implications. Going back to the connectivity properties of the Negative Perceptron, I use the fRSB framework to characterize the disconnection transition. Finally I go beyond the storage setting and study a *Spiked Random Feature Model*, where a low rank correction to the random feature matrix can be learned, in the teacher-student scenario. I observe a detection phenomenon where a minimum amount of data is needed for the student to align its spike with that of the teacher, and compare it to numerical simulations with real datasets.

Contents

List of Figures

0.1 Introduction

In the recent years, AI and in particular Deep Learning have garnered quite a lot of attention. Neural networks have proven to be capable of performing many cognitive task with a surprisingly high level of skill, and the staggering amount of possible applications has led an entire sector to grow at an unprecedented level [Krizhevsky et al., 2012, Vaswani, 2017, Jumper et al., 2021]. As with many of history's scientific revolutions, many of these incredible successes are not yet completely understood. Indeed, a general theory explaining the superiority of deep learning methods, and justifying the many empirical choices that have been made in the years, is somewhat missing. In an attempt to set the theoretical foundations of this newly founded field, many scientific communities are focusing on theoretical machine learning problems.

In this doctoral thesis I employ the tools of statistical physics to study models of neural networks. This approach is not a new one: since the 1980s, the community of Statistical Physics of Disordered Systems has been studying a number of neural network models, applying analytic methods previously developed for the study of spin glasses to the problem of understanding learning with artificial neurons. Since neural networks have earned their place as state of the art models for virtually all machine learning tasks, a number of new questions have emerged. In this thesis we will see how it is possible to address them using such tools. In particular, I will be focusing on questions regarding the non-convexity of the loss landscape, and develop tools to probe such non-convexity analytically.

The document is divided in two parts. In part I, I review some preliminaries that are necessary for the understanding of the rest of the document.

In chapter 1, I give a historical account of the development of connectionist AI, starting from the early days of the perceptron to the recent successes of deep learning. I also give a brief summary of some open questions in the theory of neural networks.

In chapter 2, I describe the tools that Statistical Physics of Disordered Systems has developed, in particular the Replica Method, and see how it is possible to use them to study the simplest models of neural networks. I go over all the details of the calculation of the Gardner volume for the spherical perceptron in the storage setting, as many of the tools and tricks needed for this are employed in the rest of the thesis. Finally, I go over some modern formulations of learning problems, such as the *Negative Perceptron* and the *Hidden Manifold Model,* models that will be studied in the second part of the thesis.

In part II, I present original work on the topic.

In chapter 3, I describe how the replica method can be used to infer the geometry of the landscape of the *Negative Perceptron,* a constraint satisfaction problem that can be seen as a neural network model. In particular, I show how using the tool of *Linear Mode Connectivity,* the

space of solutions of such model can be shown to be *Star-Shaped*, a prediction that has since its formulation been confirmed by a number of other works [Lin et al., 2024, Sonthalia et al., 2024].

In chapter 4, I address the problem of determining the maximum number of random patterns that can be learned by one and two layer networks. In order to do so, I develop an algorithm based on the Replica Symmetry Breaking theory first developed by Parisi for the study of the SK model [Parisi, 1979b], and come up with a numerical method for a precise estimation of the *SAT-UNSAT* transition. I also give an estimate of a new transition line in the phase diagram of the Negative Perceptron, which separates the Gardner phase form the full-RSB phase, and discuss potential algorithmic implications. Using the full-RSB framework developed in this chapter, I also come back to the problem of probing the landscape of the negative perceptron, and get numerical estimates for the linear disconnection transitions. Finally I compare the estimate of the SAT-UNSAT transition with the performance of Gradient Descent as a solver, and suggest that there exists an algorithmic gap between the performance of GD and the existence of solutions.

In chapter 5, I build onto the framework of the Hidden Manifold Model to consider two learning models which are closer to modern day networks used in practice. First, I consider a hidden manifold dataset learned by a two layer network, the *Tree-Committee Machine,* and show how it is possible to calculate the generalization error in the limit of many hidden units. Then I study the *Spiked Random Feature Model,* where a learnable low rank perturbation is added to the fixed (typically random) feature matrix, in connection to a widely used fine-tuning technique called *Low Rank Adaptation* . By considering a teacher-student setting, I show that a minimum amount of data is needed for the student to use the spike and align it to that of the teacher. Finally I show how a similar phenomenon can be observed in a real world fine-tuning task.

Part I

Preliminaries: A Historical Perspective on Artificial Learning, and Statistical Physics Approaches to Learning

Chapter 1

Introduction

In this section I review some of the basic concepts of Artificial Neural Networks, and how they have transformed from their conception in the 1940s to modern day deep networks. I chose to follow the chronological development of these models, as I believe this clearly outlines the difficulties that had to be overcome, and the mindset that gradually shaped into modern day learning tasks. I start with the McCulloch-Pitts neuron from the 1940s, and end with modern day gradient-based learning, describing the many advances and "AI winters" in between. Finally, in the last section of this chapter, I briefly touch upon some open problems regarding the theoretical foundations of machine learning that are somehow related to the topics of the second part.

1.1 The McCulloch-Pitts Neuron

Although identifying a single starting point for scientific ideas is necessarily a simplification, the origins of modern day Neural Networks are conventionally traced back to a 1943 paper by Warren McCulloch and Walter Pitts called "*A logical calculus of the ideas immanent in nervous activity*" [McCulloch and Pitts, 1943]. Their work was inspired by then established view of the brain as a network of neurons which can be in one of two states, active or passive, where the active state is reached when the inputs a neuron receives exceed a certain threshold and the passive state when such inputs are below the threshold. Certainly inspired by the recent development of Information Theory and Turing Computation, their intuition was that the binary nature of such networks could be modeled using Boolean Logic, and thus the complexity of a neural process reduced to a flow of 1s and 0s, which obeyed precise logical rules. The basis of such networks, which is now referred to the McCulloch-Pitts (MCP) neuron, can be seen as a grandfather of the neuron that composes modern day multilayer neural networks. It works by

receiving a set of boolean variables $x_1, \ldots, x_N$, and outputting 1 if the sum of inputs is greater or equal to the threshold, and zero otherwise. If we call $f(x_1, \ldots, x_N)$ the boolean function that is realized by this neuron, the law of its output can be written as

$$f(x_1, \ldots, x_N) = \Theta\left(\sum_{i=1}^{N} x_i - g\right), \tag{1.1}$$

where $\Theta(x)$ is the Heaviside theta, which is 1 if $x > 0$ and 0 otherwise, and g is the threshold. The neuron is thus reduced to a logical gate, and if one includes inhibitory synapses in the model, that have the effect of negating some inputs before being summed, the authors showed that many of the most common (but not all) logic gates such as the AND, OR and NOT could be realized by a MCP neuron. Using this neuronal model as a building block, McCulloch and Pitts developed a theory of neural networks as propositional logic machines, and were able to prove that any computation that can be expressed as propositional logic can be computed by a network of such neurons. In doing so they implicitly made the equivalence between perception and logical reasoning, and set the first brick on the path that would lead to the full automation of perception that modern machine learning tasks such as classification embody.

A second brick on such path was laid again by McCulloch and Pitts with their 1947 paper "*How we know Universals: the Perceptrion of Auditory and Visual Forms*" [Pitts and McCulloch, 1947], where rather than considering the propositional logic realizable by a network of MCP neurons, they focus on the actual act of perception, and how it may realized by one such network. As the title suggests, they study the perception of universals, which they define as simple forms such as squares and circles, and propose some mechanisms by which neural networks can encode translational and rotational invariance. Constrained by the technological advancement of the times, they imagine a device that is capable of such perception, which given a 2 dimensional matrix of binary values, 1 corresponding to a black pixel and 0 to a white one, outputs a 1 if the represented image is recognized and 0 otherwise. After the mechanization of neural circuits, that in the setting described above become logical computers, comes the mechanization of perception itself, reduced to a problem of mapping inputs to outputs. This represents an important paradigm shift from a deductive form of cognition, more in line with parallel lines of AI research such as Symbolic AI, to an inductive one, where information flows from the particular to the general. Although the neuronal model proposed by McCulloch and Pitts was still lacking several of the features that have allowed Machine Learning to achieve many of its great successes, the theoretical framework introduced by the two papers mentioned above can be seen as the basis for the development of connectionist AI, which many years later has given rise to the deep learning we are familiar with.

1.2 The Perceptron

Perhaps the most important element missing in the framework developed by McCulloch and Pitts is the process of learning. The neural networks described above are fixed immutable objects, and the idea that they could self-organize in an error-correcting manner had not yet been incorporated. The first to introduce a neural network model of learning was Arthur Rosenblatt, and his world famous Perceptron.

Rosenblatt had been inspired by the work of Donald O. Hebb, a neuroscientist who's seminal 1949 "The Organization of Behaviour" [Hebb, 2005] introduced the extremely influential and still relevant concept of Hebbian learning. According to this principle, learning is achieved by varying the strength of the connection between neurons, specifically by increasing such connection in neurons that often become active together, an idea that can be summed up with the often cited mantra "Neurons that fire together, wire together". Rosenblatt introduced the idea of the perceptron in an attempt to combine the mechanization of perception described above with the principle of Hebbian learning, and added an important feature to the MCP neuron which is a set of trainable weights $W_1, \dots, W_N$ and bias b [Rosenblatt, 1958]. At its core, the law of the output of a perceptron can be written as

$$f(x_1, \dots, x_N) = \text{sign}\left(\sum_{i=1}^{N} W_i x_i + b\right), \tag{1.2}$$

where $W_1, \dots, W_N$ and b are real numbers that can be viewed respectively as the strength of the connection between each sensory input and the output and the threshold. The advantage of such a formulation is that the act of classification has a clear geometric interpretation: the input $\boldsymbol{x} = (x_1, \dots, x_N) \in \mathbb{R}^N$ becomes a point in N-dimensional space, while the vector $\boldsymbol{W} = (W_1, W_2, \dots, W_N) \in \mathbb{R}^N$ identifies an N-dimensional hyperplane via the identity $\boldsymbol{W} \cdot \boldsymbol{x} + b = 0$. The output of the network depends on which side of the plane, usually referred to as *decision boundary*, the point is. The rigid framework of propositional logic is thus abandoned, and in its place tools coming from vectorial and statistical analysis, more akin to modern day formulations of classifier algorithms, are employed.

The perceptron is the first instance of a classifier algorithm that solves a supervised classification task. In his formulation, the network is shown a set of P inputs and outputs $\mathcal{D} = \{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}, \dots, \{\boldsymbol{x}_P, y_P\}$, where $y_\mu \in \{\pm 1\}$, and adapts its weights in order to give the correct output on a given input. Furthermore, Rosenblatt proposed an algorithm for the learning of such weights, now called the *Perceptron Algorithm*. The algorithm is an iterative procedure that works as follows:

1. Weights are initialized randomly.

2. Patterns are presented one at a time to the network. If the network outputs the desired output, then weights are not changed.

3. If the output is incorrect, the weight vector is updated according to $\boldsymbol{W}^{t+1} = \boldsymbol{W}^t + \eta\,(y - f^t(\boldsymbol{x}))\,\boldsymbol{x}$ where η is a scalar called the learning rate and $f^t(\boldsymbol{x}) = \text{sign}\,(\boldsymbol{W}^t \cdot \boldsymbol{x} + b)$.

4. This is repeated until all patterns are correctly classified.

Under the hypothesis that the dataset is linearly separable, that is that there exists a hyperplane that separates all inputs with label $+1$ from inputs with label -1, it was shown in [Novikoff, 1962] that the algorithm converges to a separating hyperplane in a finite number of steps. Although the actual optimization algorithm used for training modern day networks is different, the basic structure where at each iteration a set of patterns is shown, and depending on the outputs the weights are adjusted, is still in use today.

Rosenblatt's work was mainly funded by Military and Naval institutions. Their aim was to create a technology which could automize the detection of targets, for example the shape of boats and airplanes on radars, and thus were interested in practical applications of the Perceptron. This allowed Rosenblatt to realize a physical machine which used the theoretical principles outlined above. After a number of experiments, the legendary Mark I Perceptron was realized in 1960. Its schematic structure is shown in figure 1.1. As we can see the machine consisted of three types of units:

1. Sensory units, or S-units, transformed a visual pattern obtained by a 20×20 pixel camera in a set of scalars, indicating pixel brightness.

2. Association Units, or A-units, were usual perceptron units where the weight vectors were random and fixed.

3. Response Units, or R-units, were perceptron units with trainable weight vectors.

The number of A-units was 512, while the number of R-units was 8, each corresponding to a specific type of pattern to be recognized. This means that the total number of adjustable parameters was $512 \times 8 = 4096$. Interestingly the A-units, which in modern settings would be called the hidden layer, were connected via random couplings to the R-units, as Rosenblatt believed the connections between the retina and the visual cortex were random. This model of connection would be described in modern terms as a Random Feature Model, which will be described in the next chapter.

Despite Rosenblatt's optimism, the practical performance of the Mark I Perceptron was limited. It could perform extremely simple recognition tasks, such as distinguishing figures with a black square on the right from figures with a square on the left, and simple digits as long

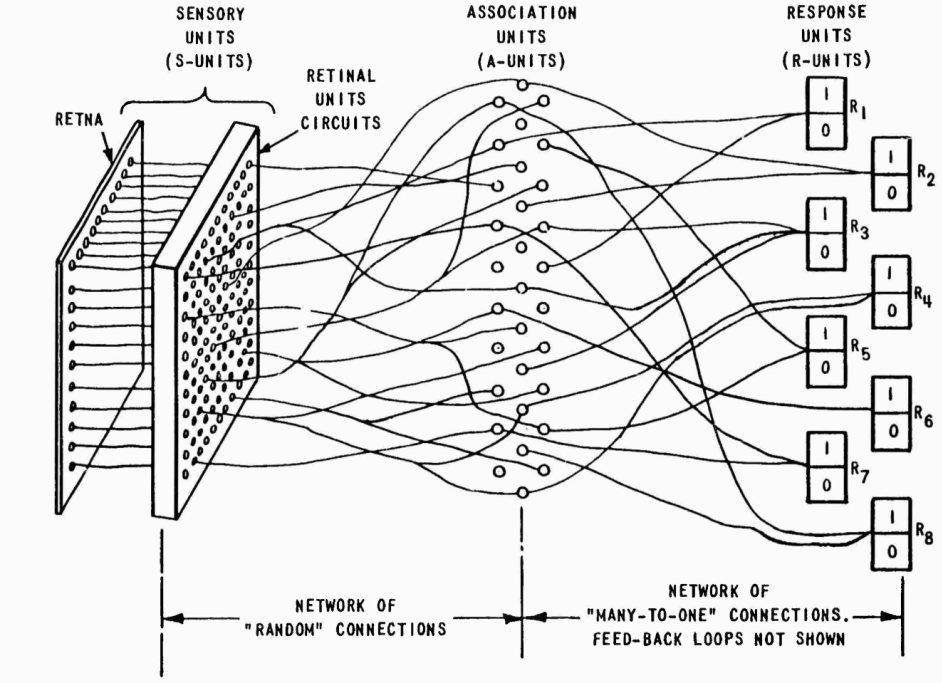

Figure 1.1: Schematic representation of the Mark I Perceptron. Reproduced from the *Mark I Perceptron Operation Manual* (1960).

as they were aligned in the 20×20 matrix. Rosenblatt was aware of these limitations, and in his 1962 book "Principles of Neurodynamics" proposed a number of extensions which he believed would overcome them. Among these, he proposed to extend the perceptron to multiple layers, an idea which would years later revolutionize the field of machine learning, although he didn't propose any algorithms for the training of such networks. Also the 1969 book "Perceptrons: An Introduction to Computational Geometry" by Marvin Minsky and Seymour Papert thoroughly investigated such limitations, and systematically showed that many classification tasks couldn't be performed by a perceptron-like architecture, at least if one took into account the physical limitations of the times. For example, the simple task of distinguishing connected from disconnected figures, could not be performed by a perceptron in which the A-units were only locally connected, that is connected to a small subset of R-units. This book is often cited as the reason for what is referred to as the AI winter, where research in connectionist AI was virtually halted for 10 years, until the first years of the '80s. In a prologue to a later edition of their book, Minsky and Papert argue that the cause for such winter was the lack of basic theories explaining the functioning of the perceptron on some tasks rather than others, and that their work simply gave momentum to this important aspect of neural network research. By focusing on the interaction between the architecture and the learning task, the two were able to rigorously identify a set of tasks which were not realizable with the architectures used at the time. In their words

> "The trouble appeared when perceptrons had no way to represent the knowledge required for solving certain problems. The moral was that one simply cannot learn enough by studying learning by itself; one also has to understand the nature of what one wants to learn. This can be expressed as a principle that applies not only to perceptrons but to every sort of learning machine: No machine can learn to recognize X unless it possesses, at least potentially, some scheme for representing X."

Although their results were limited to one layer networks, it was somehow naively assumed that they would also hold for multilayer networks, thus halting excitement for connectionist AI. Almost 20 years had to pass for a new theoretical result, the *Universal Approximation Theorem*, according to which any function can be implemented by a two layer neural network [Cybenko, 1989], to inspire hope in neural network research. Perhaps, the truth of the matter is that the basic ingredient for practically harnessing the power of multiple layers, backpropagation, still hadn't been discovered.

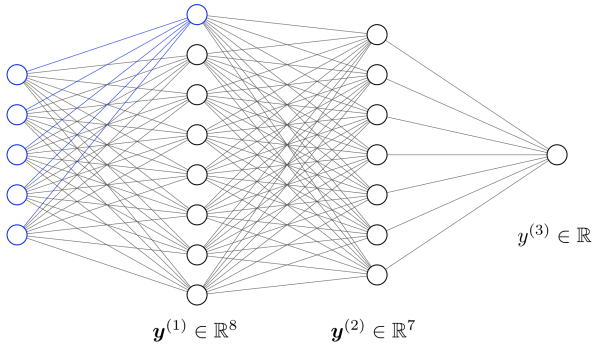

Figure 1.2: An example of a modern fully connected neural network. In blue a single perceptron is highlighted.

1.3 Gradient Based Learning

In his 1962 book "*Principles of Neurodynamics*" Rosenblatt describes the problem of "Back-Propagating Error Correction Procedures" to train multilayer neural networks. He had come to the conclusion that the performance of the perceptron could be improved by training also the connections between the S and A units, and thus starting working on some procedures to train both layers at the same time. His understanding that the signal on how to modify the weights should be backpropagated from the error calculated on the R units was correct, however he did not develop any satisfactory procedure for this backpropagation. Indeed he was considering binary neurons, so couldn't harness the power of gradients. For backpropagation applied to neural networks as we are used to it, we need to wait almost 25 years, when David Rumelhart, Geoffrey Hinton and Ronald Williams outlined such a procedure in an article that appeared in *Nature* in 1986 [Rumelhart et al., 1986] (more or less at the same time also Yann LeCun had come up with a technique which in certain settings reduces to the usual backpropagation [LeCun, 1985]). The idea of backpropagation is very simple, and it can be seen as a direct application of the chain rule. For this reason backpropagation-like techniques had been proposed in many occasions before 1985, although none in the context of neural network learning.

To understand the basic idea behind gradient-based learning and backpropagation, let us consider the simplest multilayer architecture, the *Multilayer Perceptron*. Here, the outputs of a number of N_h perceptrons which act on the same input are concatenated and treated as a new input for the next layer. This vector of outputs is usually referred to as a *hidden layer*, and a neural network can be composed of different hidden layers (the network in figure 1.2 is

composed of two hidden layers for example). The output of a general neural network which is composed of L hidden layers each composed of N_h neurons is

$$y_i^{(1)} = \varphi\Big(\sum_j^N W_{ij}^{(1)} x_j\Big), \tag{1.3}$$

$$y_i^{(2)} = \varphi\Big(\sum_j^{N_h} W_{ij}^{(2)} y_j^{(1)}\Big), \tag{1.4}$$

$$\vdots \tag{1.5}$$

$$y^{out} = \sum_j^{N_h} W_j^{(L+1)} y_j^{(L)}. \tag{1.6}$$

Here $\varphi(x)$ is a general non-linear activation function, which in modern gradient-based settings is usually taken to be continuous, such as $\varphi(x) = \tanh(x)$ or $\varphi(x) = \max(x, 0)$. These non-linearities are a fundamental feature of any multilayer perceptron: without them we could write (using the matrix notation) the output as

$$y^{out} = \underbrace{W^{(L+1)} \cdot W^{(L)} \cdot \dots \cdot W^{(1)}}_{\boldsymbol{A}} \boldsymbol{x} = \boldsymbol{A} \cdot \boldsymbol{x} \qquad W^{(L+1)} \in \mathbb{R}^{N_h}, \tag{1.7}$$

$$W^{(l)} \in \mathbb{R}^{N_h \times N_h} \quad l \in \{2, \dots, L\}, \tag{1.8}$$

$$W^{(1)} \in \mathbb{R}^{N \times N_h}, \tag{1.9}$$

and the action of the whole network could be reduced to a scalar product of the input $\boldsymbol{x}$. This would incur in all the limitations we described above for linear models.

The multilayer perceptron consists of many perceptron stacked onto each other. This particular way of connecting neurons is called a *feed-forward* architecture, as information flows though the network in a given direction, from one hidden layer to the next. It is not the only possible choice, however it has proved to be the most successful in a range of Machine Learning tasks.

The basic idea behind gradient based learning is to define a loss function, which measures how far away we are from our objective, and find a set of weights which minimize it. For example, if we are in the context of supervised learning, our objective is to correctly classify the dataset we are given, and the loss function is a measure of how many mistakes we are making. In all cases the loss function can be written as a sum of functions calculated in each

example-label pair $\{\boldsymbol{x}^\mu, y^\mu\}$

$$\mathcal{L}(\Theta; \mathcal{D}) = \sum_{\mu=1}^{P} V(\Theta, \{\boldsymbol{x}^\mu, y^\mu\}), \tag{1.10}$$

where we have indicated with Θ the set of all parameters of the network. For gradient-based learning it is required that the loss function be differentiable, since we will need to calculate the gradient of this function with respect to the weights. A typical example of a loss function for the task of regression is the *mean-square loss,* defined as

$$\mathcal{L}(\Theta; \mathcal{D}) = \frac{1}{P} \sum_{\mu=1}^{P} \left(y^\mu - y_\Theta(\boldsymbol{x}^\mu) \right)^2, \tag{1.11}$$

while for binary classification, the binary cross entropy is by far the most common choice

$$\mathcal{L}(\Theta; \mathcal{D}) = \frac{1}{P} \sum_{\mu=1}^{P} \log\left(1 + e^{-y^\mu y_\Theta(\boldsymbol{x}^\mu)}\right). \tag{1.12}$$

Now that we have defined a loss function, we need to find a way to minimize it. Perhaps one of the most naive optimization algorithms is gradient descent, which is an iterative procedure which works by repeatedly following the gradient (which coincides with the direction of steepest ascent). In practice, at each iteration we update every parameter of the network according to

$$\theta_i^{t+1} = \theta_i^t - \eta \frac{\partial \mathcal{L}}{\partial \theta_i} \qquad \text{for } i = 1, \ldots, N, \tag{1.13}$$

where η is a (small) scalar called the learning rate, and stop when a certain stopping criterion is met. This simple algorithm has no guarantee of finding the global minimum of the loss function, and is actually quite prone to getting stuck in local minima if the landscape is highly non-convex. Because of the architecture of the multilayer perceptron, $\mathcal{L}(\Theta; \mathcal{D})$ is typically non-convex in the network parameters Θ, and there is thus no guarantee that gradient descent will return a true global minimum of the loss. Indeed this was one of the first concerns regarding gradient-based learning. However, practitioners in the last 30 years have been completely ignoring this issue, and successfully using an optimization algorithm originally designed for convex optimization on a non-convex problem.

For practical reasons that have to do with the efficiency of modern day GPUs, what is used in practice is a slight variation of this algorithm, where the loss function is calculated on a small subset of the dataset called *mini-batches,* that usually consist of tens or hundreds of examples.

This is what goes under the name of *Stochastic Gradient Descent*. At each iteration the gradient is estimated using a small subset of the dataset, and it will thus be noisy. Although this might interfere with the estimation of the direction of steepest descent, the noise has the desirable feature that it helps escaping local minima. Indeed, SGD beyond any expectation has proven to be incredibly successful at optimizing Neural Networks, so much so that a new theoretical understanding of the Landscape of Neural Networks is needed. We will see in the next chapter how understanding the geometry of the landscape of neural networks has become one of the central topics in the field of Theoretical Machine Learning.

There is only one ingredient missing for the formulation of gradient-based learning, and that is an efficient way of calculating the gradient in equation 1.13. This was precisely the intuition of Rumelhart, Hinton and Williams which proposed the following iterative scheme. Consider the derivative with respect to a weight from the last layer. This can be straightforwardly calculated as

$$\frac{\partial \mathcal{L}}{\partial W_i^{(L+1)}} = \frac{\partial \mathcal{L}}{\partial y^{out}} \frac{\partial y^{out}}{\partial W_i^{(L+1)}} = \frac{\partial \mathcal{L}}{\partial y^{out}} y_i^{(L)}. \tag{1.14}$$

If instead we consider a weight from an inner layer (l) we have that

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}} = \sum_k \frac{\partial \mathcal{L}}{\partial y^{out}} \frac{\partial y^{out}}{\partial y_k^{(l)}} \frac{\partial y_k^{(l)}}{\partial W_{ij}^{(l)}} = \frac{\partial \mathcal{L}}{\partial y^{out}} \frac{\partial y^{out}}{\partial y_i^{(l)}} \varphi' \left(\sum_m W_{km}^{(l)} y_m^{(l-1)} \right) y_j^{(l)}. \tag{1.15}$$

The outputs of each neuron $\{y_j^{(l)}\}_{j=1}^{N_h}$ are calculated during the forward pass, and so they can be stored in order to avoid useless computations. The only term that is missing is the derivative $\frac{\partial y^{out}}{\partial y_i^{(l)}}$. But we have that

$$\frac{\partial y^{out}}{\partial y_i^{(l)}} = \sum_k \frac{\partial y^{out}}{\partial y_k^{(l+1)}} \frac{\partial y_k^{(l+1)}}{\partial y_i^{(l)}} = \sum_k \frac{\partial y^{out}}{\partial y_k^{(l+1)}} \varphi' \left(\sum_m W_{km} y_m^{(l)} \right) W_{ki}^{(l+1)}. \tag{1.16}$$

That is this derivative can be expressed as a function of the derivatives $\left\{ \frac{\partial y^{out}}{\partial y_i^{(l+1)}} \right\}_{i=1}^{N_h}$ with respect to the next layer. Backpropagation then is a dynamical programming technique which iteratively calculates these derivatives starting from the last layer and going backwards. This way a number of useless computations are avoided, and the complexity is $O\left((L-1)N^2 + LN\right)$. A gradient step can thus be performed with a forward pass, with which the output y^{out} and the values of each neuron $\{y_i^{(l)}\}_{i,l}$ are calculated, and a backward pass, with which the gradients $\frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}}$ are calculated.

Although all the ingredients for gradient based learning were there in the 1986 paper by Rumelhart, Hinton and Williams, it took a while for deep learning to be accepted as the best alternative for learning. For most of the 1990s and first half of the 2000s other machine learning methods were studied, such as kernel machines and support vector machines, which provided simpler but equally efficient alternatives to the more mysterious and complicated neural networks. In particular for the small datasets of the times, SVMs proved adequate, and much simpler to implement. Perhaps the truth of the matter is that the technological infrastructure wasn't ready for the the calculations that are necessary to train a neural network, and so research in deep learning was relegated to a few research centers scattered around north america. A notable exception is LeNet, the first example of a convolutional neural network developed by Yann LeCun while he worked for Bell Labs, starting from 1988 [LeCun et al., 1998]. There he had access to a large dataset of images of handwritten digits, the famous MNIST dataset, as the US postal service was interested in automating their recognition. Using gradient based learning, he trained a neural network that was capable of recognizing every digit, and in doing so also wrote a compiler which given a neural network architecture would write a c program implementing it. Even though LeNet represented one of the first and most important successes of backpropagation, neural networks still didn't manage to become the dominant paradigm for machine learning. SVMs were still a lot easier to implement and achieved similar performances for the simple tasks of the time, and most importantly could be easily implemented using accessible software. Deep Learning required specific frameworks to be developed, and Yann LeCun's one could not be distributed as it was proprietary of the Bell Labs.

The occasion for neural networks to prove their superiority with respect to other methods came many years later, in the occasion of an annual image recognition contest named ImageNet. The dataset was composed of 1.2 million images divided in 1000 categories, and the organizers challenged anyone to develop an algorithm that could achieve the lowest top-5 error rate, that is the percentage of times the true label is not in the 5 most probable predicted labels. In 2010 the challenge was won by a team from the University of Illinois that used fixed predetermined feature extractors to associate a large vector to every image and use an SVM to classify it, achieving a 28% error rate. In 2012 a team consisting of Geoffrey Hinton, Alex Krizhevsky and Ilya Sutskever decided to participate, and developed a massive convolutional neural network, that they called AlexNet, with more than half a million neurons and 60 million parameters, and trained it on the ImageNet dataset using SGD [Krizhevsky et al., 2012]. AlexNet had a similar architecture with respect to LeNet, however two main advantages set it apart from its simpler cousin: first of all the amount of data the network was trained on was much larger than any previous attempt. Secondly, for the first time Hinton's

team used gpu processors to accelerate the matrix product operations which are at the core of gradient based learning, thus obtaining a huge performance boost. These two factors led the team to obtain a substantial gain with respect to SVM models, achieving a 17% top-5 error rate. Neural networks demonstrated their true potential, and since then have been considered the state of the art models for computer vision.

1.4 Some Open Questions Regarding the Foundations of Machine Learning

In the previous section I described the main types of architectures and protocols that are now commonly used in Machine Learning applications. In many cases, the process that has led to the adoption of certain techniques rather than others has been entirely empirical. Indeed many of the great successes of the recent years have been driven by product optimization rather than deep theoretical analysis, and understanding why some approaches have failed while other have been successful is sometimes hard. Due to this application-driven research that has dominated the field of Machine Learning in the last decade, we are now in the situation that a theory thoroughly explaining the incredible success of deep learning is somewhat lacking. Indeed many questions regarding specific mechanisms by which learning occurs remain unanswered. In this section I will describe a selection of open problems which are related to the topics covered in this thesis.

1.4.1 Gradient Descent on a Non-Convex Loss

One of the many mysteries of deep learning is the interaction between the training algorithm, typically a variant of gradient descent, and the loss function. While first order methods provably find minimizers in convex problems, no such guarantee exists for non-convex functions such as the loss function of a neural network. In fact, Gradient Descent should get stuck in local minima (although this can be avoided using noisy variants such as Stochastic Gradient Descent). Empirically, it has been observed that when the number of parameters is large enough, gradient descent successfully converges to a near-optimal minimum. This has led a large body of research to investigate the geometry and non-convexity of the loss landscape. This topic will be described in further detail in section 3.1.

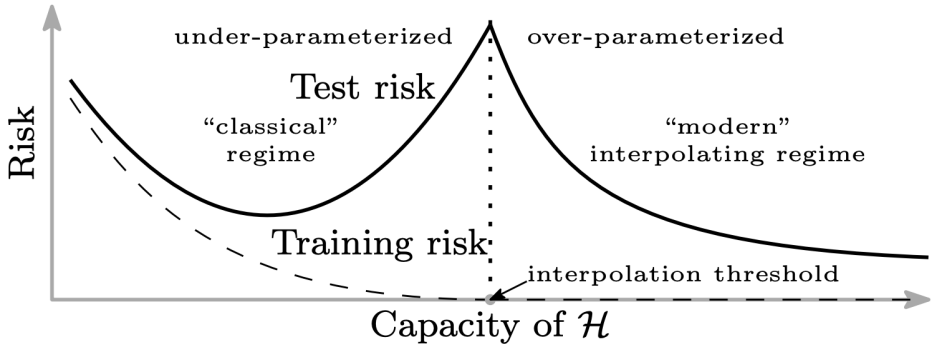

Figure 1.3: A cartoonish view of the Double Descent phenomenon. Image reproduced from [Belkin et al., 2019].

1.4.2 Absence of Overfitting in Overparametrized Neural Networks

Not only is the minimum found by gradient descent near-optimal, the generalization performance of such interpolating and largely overparametrized networks is also remarkable. This is contrary to common Statistical Learning wisdom, according to which when the complexity of the model is large enough to completely fit the training data, then the generalization performance should degrade. This phenomenon goes under the name of "Overfitting", and can be intuitively understood by imagining that the network is not learning the rule that generated the data but is simply memorizing each data point. To avoid overfitting data, many things can be done. By starters, one can always stop the optimization before reaching zero training error: this is usually referred to as *early stopping*. Another common approach is to add a *regularization* term in the loss, for example the $L2$ norm of the weights

$$\mathcal{L}(\Theta; \mathcal{D}) = \sum_{\mu=1}^{P} V(\Theta, \{\boldsymbol{x}^\mu, y^\mu\}) + \lambda \|\Theta\|^2, \tag{1.17}$$

such that by tuning λ it is possible to control the final norm of the minimizer, and avoid that the training dataset is learned perfectly.

Interestingly, many of these procedures are ignored in deep learning. Practitioners use networks that are powerful enough to fit any dataset, and still observe remarkable generalization performance. In an attempt to explain this behaviour, a number of experiments have been performed, where for a fixed dataset, the generalization and training errors are measured as the complexity of the learning model is increased [Nakkiran et al., 2021, Belkin et al., 2019]. What is observed is the following: the training error monotonically decreases until the model is complex enough to fit the dataset, and stays zero from then on. The point in which it becomes zero is usually referred to as the "Interpolation Peak". The test error instead, first decreases

until a local minimum is achieved, then starts increasing until it reaches a local maximum at the interpolation peak, and finally decreases (see Fig. 1.3 for a cartoonish view). This phenomenon is usually referred to as "Double Descent", and can be recovered analytically in simple settings by studying linear models with label noise or random feature models (we will come back to random feature models below) [Mei and Montanari, 2019, Gerace et al., 2020, Hastie et al., 2022, d'Ascoli et al., 2020]. The peak in the generalization error can be understood as an explosion of the variance of the predictor due to the various randomnesses in the learning process[d'Ascoli et al., 2020, Yang et al., 2020, Jacot et al., 2020]. In [d'Ascoli et al., 2020] authors breakdown these various sources of randomness for an analytically tractable model, and show how ensembling can attenuate the peak. In [Gerace et al., 2020] on a similar model the authors observe that the optimal amount of regularization can completely eliminate the interpolation peak.

While the behaviour before the interpolation peak is coherent with the overfitting phenomenon described above, the behaviour after the peak is not. Indeed the decrease of the test error after the peak was not expected nor predicted by Statistical Learning Theory. This has led researchers to come up with concepts such as "Benign Overfitting" and "Implicit Regularization", according to which even without regularization models with large complexity converge toward "simple" solutions, thanks to an implicit bias in commonly used training algorithms such as SGD [He et al., 2019]. These phenomena have been studied in a number of simple models [Bartlett et al., 2021, Chizat and Bach, 2020, Chen et al., 2024, Liang and Rakhlin, 2020], and have given some theoretical evidence supporting the well known empirical fact that "Larger models are better".

1.4.3 Lazy Regime vs Feature Learning

A number of experiments showed that, in particular regimes, neural networks can reach zero error with their weights hardly varying [Li and Liang, 2018, Du et al., 2019]. This observation has led [Chizat et al., 2019] to identify a regime in which neural networks are close to their Taylor expansion

$$f_\Theta(\boldsymbol{x}) \approx f_{\Theta_0}(\boldsymbol{x}) + \nabla_\Theta f\Big|_{\Theta_0} \cdot (\Theta - \Theta_0)\,, \tag{1.18}$$

that goes under the name of *Lazy Training*. In this regime, the authors prove bounds on the distance between the network and its linearized version along training, and show that this phenomenon naturally arises due to an often implicit choice of scaling. The simplest setting in which this scaling can be understood is a two layer network with the following structure

$$f_\Theta(\boldsymbol{x}) = \frac{1}{N^\gamma}\sum_i a_i \varphi\left(\frac{1}{\sqrt{N}}\boldsymbol{W}_i \cdot \boldsymbol{x}\right). \tag{1.19}$$

Authors show that whenever $\gamma < 1$ the network stays in the lazy regime, as opposed to the $\gamma = 1$ case where it is capable of learning features. Writing the expansion of equation 1.19 explicitly, we get that

$$\begin{aligned} f_\Theta(\boldsymbol{x}) - f_{\Theta_0}(\boldsymbol{x}) \approx \frac{1}{N^\gamma}\sum_i (a_i - a_{i,0})\varphi\left(\frac{1}{\sqrt{N}}\boldsymbol{W}_{i,0} \cdot \boldsymbol{x}\right) + \\ + \frac{1}{N^\gamma}\sum_i a_{i,0}\left(\boldsymbol{W}_i - \boldsymbol{W}_{i,0}\right)^T \boldsymbol{x}\, \varphi\left(\frac{1}{\sqrt{N}}\boldsymbol{W}_{i,0} \cdot \boldsymbol{x}\right). \end{aligned} \tag{1.20}$$

This "linearized" network is the sum of two components: the first is referred to as a Random Feature Model (RFM), where the first layer weights are stuck at their (random) initialization, and learning is only performed on the second layer weights, while the second is referred to as the Neural Tangent Kernel (NT) [Jacot et al., 2018]. Inspired by this linearization, a number of works have studied RFMs and NTs as a proxy to understand the lazy regime [d'Ascoli et al., 2020, Ghorbani et al., 2021].

To compare the two regimes, a simple numerical experiment can be performed. Let us define the model

$$\tilde{f}_\Theta(\boldsymbol{x}) = \alpha\left(f_\Theta(\boldsymbol{x}) - f_{\Theta_0}(\boldsymbol{x})\right). \tag{1.21}$$

For the output to stay of order one in the limit $\alpha \gg 1$ we require $\Theta - \Theta_0 \sim 1/\alpha$, so for large α the network will stay close to initialization, and we can therefore perform the linearization of equation 1.19. By varying α then it is possible to switch from one regime to the other. Indeed authors of [d'Ascoli et al., 2020] find that the performance of such network when $\alpha \gg 1$ is comparable to that of a random features model, while in [Chizat et al., 2019] authors observe that for deep networks used in practice performance degrades in the lazy regime. This has been confirmed by a series of numerical works which have identified the capacity of neural networks to learn features as the main driver of the success of deep learning [Malach et al., 2021]. While the lazy regime has been thoroughly analyzed thanks to its equivalence to RFMs and NTs, characterizing the feature learning regime has proved more tricky.

Chapter 2

A Statistical Physics Approach to Learning

As we have seen in the previous chapter, many questions concerning the fundamental mechanisms through which Machine Learning, and in particular Deep Learning, have proven to be successful in a wide range of tasks remain unanswered. In the last decades a number of scientific fields have devoted some attention to such matters, each giving its own frame and contributions. One such field is Physics. As argued in [Zdeborová, 2020] the advantage of the Physical approach is that it lies in between Mathematics, limited by its level of rigor to simple results or unfeasible hypothesis, and Engineering, often times sacrificing deep understanding in favor of effectiveness. Historically Physics, and in particular Statistical Physics, has proceeded by modeling the phenomena it sought to understand, that is by stripping it of the complicated and often irrelevant details, and focusing on the basic mechanisms which give rise to fundamental observable behaviors. This allows the physicist to combine analytic treatment (although not at the level of rigor of Mathematics) and realistic phenomena. In this chapter I review some of the classic Statistical Physics approaches to modeling Neural Networks that have been introduced in the last four decades, a field that goes by the name of Statistical Mechanics of Learning (SMoL), and then focus on more recent developments. In section 2.1, I describe the general setup for framing a problem of learning with an artificial neural network in the language of statistical physics. In section 2.2, I give an example of an analytic calculation that can be performed using a tool that comes from the field of Statistical Mechanics of Disordered Systems, the Replica Method. Many of the novel results in the next part of this document are based on the techniques which are outlined in this section. In section 2.3, I describe a simple variant of the classic learning setup by defining the *margin,* and introduce a model that will be studied in detail in chapters 3 and 4, the *Negative Perceptron.* Finally in section 2.4, I describe a modern formulation of the learning problem, the *Hidden Manifold Model*, which takes into account the intrinsic low dimensionality of the high-dimensional data, and which will come back in chapter 5.

2.1 General Setup

Any machine learning setup consists of three fundamental parts:

1. The *Architecture,* which is another way to say the model we are employing for our learning. If we restrict ourselves to neural networks, then the number of neurons, layers, the type of activation function, the connection between neurons are all factors that influence the information that can be represented, and thus the task that can be learned.

2. The *Data*, that is the dataset from which we are trying to learn. As we saw in the previous chapter connectionist AI is an inductive form of learning, therefore our final trained model cannot contain more information than there was in the training data.

3. The *Algorithm*, that is the learning strategy that we use to train our architecture using the data. In most modern-day settings all training strategies are gradient based, however there are many variations that can significantly vary the final output.

Of course each of these three elements can be thought of as an individual part that will in itself influence the outcome of the learning. However the most interesting, and in many cases still open, questions are regarding their interaction: how can a specific architecture learn properties of a certain dataset? How will a particular optimization strategy work on a particular neural architecture?

In this section I will review how these three parts are modeled in the statistical physics framework. By choosing precisely what elements to include in our final model, it will be possible to understand the interaction between them.

2.1.1 The Architecture

The first element that requires modeling is the network itself. Historically, the first Statistical Physics model of Neural Networks has been Elisabeth Gardner's seminal work [Gardner and Derrida, 1988, Gardner, 1988], in which the authors study a model of a one layer neural network that performs binary classification. That is, a network is completely specified by the N-dimensional vector of its weights $\boldsymbol{W}$, and the output is a label which is parametrized by the numbers $+1$ and -1. The output of the network on an input $\boldsymbol{x} \in \mathbb{R}^N$ can thus be expressed as

$$\hat{y}_{\text{perceptron}}(\boldsymbol{x}) = \text{sign}\left(\frac{\boldsymbol{W}\cdot\boldsymbol{x}}{\sqrt{N}}\right). \tag{2.1}$$

Although it is possible to chose any prior on the weights $\boldsymbol{W}$, historically two main choices have been studied, which are the spherical prior, where weights are normalized such that they lie on

the N-dimensional hypersphere $\|\boldsymbol{W}\|^2 = N$, and the binary prior, where each element of $\boldsymbol{W}$ is ± 1 [Krauth and Mézard, 1989]. As we can see this model is nothing but the perceptron introduced in the previous chapter, that was shown to perform linear classification. An improvement on the complexity of the architecture can be obtained by considering two layer networks. Historically, the first to be introduced are the committee and parity machines [Schwarze and Hertz, 1992, Schwarze, 1993, Monasson and Zecchina, 1995], where the network is specified by K different weight vectors $\boldsymbol{W}_k$, and the output is given by

$$\hat{y}_{\text{committee}}(\boldsymbol{x}) = \text{sign}\left[\frac{1}{\sqrt{K}}\sum_{l=1}^{K}\text{sign}\left(\frac{\boldsymbol{W}_l\cdot\boldsymbol{x}}{\sqrt{N}}\right)\right] \qquad \hat{y}_{\text{parity}}(\boldsymbol{x}) = \text{sign}\left[\prod_{l=1}^{K}\text{sign}\left(\frac{\boldsymbol{W}_l\cdot\boldsymbol{x}}{\sqrt{N}}\right)\right]. \quad (2.2)$$

As we can see, both outputs represent two layer networks, however the only weights that are learned are the first layer weights. The names come from the fact that for the first model the output is given by the majority signs given from each neuron, while for the second by the parity of these outputs. A common variation that has the advantage of simplifying calculations is the *Tree-Committee Machine,* where the first weight vector $\boldsymbol{W}_1 \in \mathbb{R}^{N/K}$ acts on the first N/K components of $\boldsymbol{x}$, the second $\boldsymbol{W}_2$ on the second block of N/K components and so on. This way the number of trainable parameters is still N, exactly like the perceptron, although the nonlinearities guarantee that the model is not linear.

2.1.2 The Data

Up to now, we haven't specified anything about the dataset. Statistical Mechanics of Learning deals with typical properties of learning with random data. As in most theoretical machine learning setups, one supposes that the example-label pair is an *iid* sample from an unknown joint distribution function. The power of SMoL is that it gives us the tools to average complicated observables over this distribution.

Depending on which feature of learning we seek to model, there are different possibilities for choosing the joint distribution of data. In all cases, it is customary to suppose a factorized distribution for the examples: $x_i \overset{iid}{\sim} P(x_i)\ \forall i \in \{1,\dots,N\}$ (we will see later on how more realistic distributions can be analyzed). Typical choices are binary distributions $P(x_i) = \frac{1}{2}\delta(x_i - 1) + \frac{1}{2}\delta(x_i + 1)$ or standard gaussian $P(x_i) = \mathcal{N}(0,1)$. As for the labels, there are two ways of generating them that have historically been studied.

1. The first one is to choose *iid* labels, $y^\mu = \pm 1$ with probability $1/2$. This goes under the name of the **Storage Problem**. This is the most unstructured dataset we can imagine: labels are independent of examples and are completely random, and examples are drawn

from a simple probability distribution which factors over its dimensions. If our task was to model real-world datasets in some way, one could argue that we have miserably failed. However, one might still be interested questions of a are more combinatorial nature. Let us now introduce a key quantity that will be used throughout the whole thesis:

$$\alpha = \frac{P}{N} = \frac{\text{Number of Examples}}{\text{Dimension of Each Example}}. \tag{2.3}$$

One possible question that could be asked is what is the probability that a random dataset is linearly separable as a function of α? And what fraction of volume of weight vectors effectively classify a linearly separable dataset, again as a function of α? Questions such as these can be thoroughly addressed even with this simple model of data. Furthermore, it is possible to define observables that are typically used in actual machine learning scenarios. For example, the quantity $\Theta(y^\mu \hat{y}^\mu)$, where $\Theta(x)$ is the Heaviside Theta, is equal to 0 if the output of the perceptron is correct and 1 otherwise. We can then define the training error as the sum over the whole dataset of this quantity

$$\epsilon_T(\boldsymbol{W}; \mathcal{D}) = \frac{1}{P} \sum_{\mu=1}^{P} \Theta(-y^\mu \hat{y}^\mu). \tag{2.4}$$

The error we have defined is still a function of the weight vector and the dataset. The great power of SMoL is that it allows us to average this observable over both distributions, and give us access to the typical training error.

2. The second possibility is to generate the labels using another network, called the *teacher* network. This is goes under the name of the **Teacher-Student Scenario.** If we call $\tilde{\boldsymbol{W}}$ the weights of this second network, then given an example $\boldsymbol{x}^\mu$, the corresponding label will be $y^\mu = \text{sign}\Big(\sum_i \tilde{W}_i x_i^\mu\Big)$. The task of the original network then, usually referred to as the *student* network, is to align its outputs $\hat{y}^\mu = \text{sign}\Big(\sum_i W_i x_i\Big)$ as much as possible with the actual labels, by looking at the input-output pairs that the teacher has generated. Although we have not improved in the generation of the examples, we have introduced structure in the dataset by generating labels according to a rule. This allows us to model situations and define observables that couldn't be dealt with in the Storage Problem. For example, one quantity which is central in machine learning and does not have a natural interpretation in the storage scenario is the *generalization* or *test error*. This usually indicates the average error a trained network would make in classifying an example which was not in its training dataset. In the teacher-student scenario, it can be easily

written down as

$$\epsilon_g(\boldsymbol{W}, \tilde{\boldsymbol{W}}) = \mathbb{E}_{\boldsymbol{x}}\Theta\big(-y(\boldsymbol{x})\hat{y}(\boldsymbol{x})\big), \tag{2.5}$$

where $\mathbb{E}_{\boldsymbol{x}}$ is the average over a new example.

To finish off, let us comment that the case we have considered here where the teacher network has the same architecture as the student network is normally referred to as a *matching* situation (or bayes-optimal). However, it is also possible to consider *unmatched* scenarios, in which the architecture of the teacher differs from that of the student. We will see examples of this in the following chapters.

2.1.3 The Algorithm

As we have seen learning usually is achieved via *Empirical Risk Minimization*, that is given a fixed dataset we define a loss function over the space of parameters of the network, and we seek to minimize it, together with some regularization penalty:

$$\boldsymbol{\Theta}^* = \arg\min_{\boldsymbol{\Theta}}\{\mathcal{L}(\Theta; \mathcal{D}) + \lambda\|\boldsymbol{\Theta}\|^2\}. \tag{2.6}$$

This minimization can be framed in the language of statistical physics. The link between the two setups is provided by the Gibbs distribution, which given a generic Hamiltonian $\mathcal{H}$ defined over a vector of degrees of freedom $\boldsymbol{\Theta}$ can be written as

$$P(\boldsymbol{\Theta}) = \frac{1}{\mathcal{Z}} e^{-\beta\mathcal{H}(\boldsymbol{\Theta})}, \tag{2.7}$$

where β is a parameter that in physics represents the inverse temperature, which controls how peaked the distribution is around minimizers, and $\mathcal{Z}$ is the normalization constant. If we take the limit $\beta \to \infty$ then it is easy to see that the Gibbs distribution becomes the uniform distribution over all vectors $\boldsymbol{\Theta}$ that achieve zero error on the training set.

If we make the association $\mathcal{H} \leftrightarrow \mathcal{L}$, and introduce a prior over the weights $\mu(\boldsymbol{\Theta}) \propto \exp\big(-\beta\lambda\|\boldsymbol{\Theta}\|^2\big)$, then in the $\beta \to \infty$ limit the Gibbs distribution becomes the uniform distribution over all minimizers of equation (2.6). To make things more concrete, let us consider a specific example of architecture and loss function. The simplest possible choice is to consider a spherical perceptron, defined by its weight vector $\boldsymbol{W}$ and its prior $\mu(\boldsymbol{W}) = \delta\left(\|\boldsymbol{W}\|^2 - N\right)$, and the number of errors loss, which simply counts the number of errors that the perceptron makes on the dataset

$$\mathcal{L}(\Theta; \mathcal{D}) = \sum_{\mu=1}^{P} \Theta\left(-y^\mu \hat{y}_{\boldsymbol{W}}(\boldsymbol{x}^\mu)\right). \tag{2.8}$$

The Gibbs distribution in the zero temperature limit can then be conveniently written as

$$P_{\mathcal{D}}(\boldsymbol{W}) \overset{\beta\to\infty}{\to} \frac{1}{\mathcal{Z}}\mu(\boldsymbol{W})\prod_{\mu=1}^{P}\Theta\Big(-y^{\mu}\hat{y}_{\boldsymbol{W}}(\boldsymbol{x}^{\mu})\Big). \tag{2.9}$$

In other words it is the uniform distribution over all vectors on the sphere which achieve the zero error on the training dataset. From now on we will call these zero-error vectors *solutions*.

By definition the normalization constant can be written as $\mathcal{Z} = \int d\mu(\boldsymbol{W})e^{-\beta\mathcal{H}(\boldsymbol{W})}$. In Physics, this quantity plays such a special role that it has deserved a name of its own: the *Partition Function*. In the $\beta \to \infty$ case, then the expression reduces to

$$\mathcal{Z}(\mathcal{D}) = \int d\mu(\boldsymbol{W})\prod_{\mu=1}^{P}\Theta\Big(-y^{\mu}\hat{y}_{\boldsymbol{W}}(\boldsymbol{x}^{\mu})\Big). \tag{2.10}$$

Because the argument of the integral is 1 if the weight vector correctly classifies all examples (and is of the correct form as specified by the prior) and 0 otherwise, it is easy to give a geometric interpretation to this quantity: it represents the volume of the vectors which correctly classify the whole dataset. However, the volume we have defined is still a function of the dataset, while the theory we seek to develop should describe typical properties seen on generic datasets. The first thing we could do is to average the volume with respect to the distribution of the dataset, and hope that "typical" properties are close enough to average properties in the high-dimensional limit $N \to \infty$. Quantities that enjoy this property in physics are called *self-averaging observables*. We would write

$$\tilde{\mathcal{Z}} = \mathbb{E}_{\mathcal{D}}\mathcal{Z}(\mathcal{D}), \tag{2.11}$$

and perform the integral in a rather straightforward manner. Averages such as this one are referred to as *annealed* averages. Unfortunately, there is no guarantee that they give the correct result. This is because the partition function is *not a self-averaging quantity*. To understand what is going on, we can use some common wisdom that comes from statistical physics, which tells us that *extensive* quantities, that are linear in N, are typically self-averaging. The volume we are considering however, as all volumes typically are, is exponential in the dimension N. We could then define a new quantity as the logarithm of the partition function. In physics, this is usually referred to as the (in this case zero temperature) *free entropy*

$$\Phi = \mathbb{E}_{\mathcal{D}}\log\mathcal{Z}(\mathcal{D}). \tag{2.12}$$

Being the logarithm of an exponential quantity, the free entropy is extensive, and we can thus hope that by averaging it, we might obtain the correct result. This type of average is referred to as a *quenched* average. We will see that indeed, this gives the correct result. However, this comes at a price: rather than averaging the volume directly, we now have to average its logarithm. Unfortunately, performing averages of logarithms is not entirely straightforward. In the next section we will see that one possible method to perform these calculations is the famous *Replica Method.*

2.2 The Replica Method for the Calculation of the Gardner Volume

In this section, we will see how it is possible to calculate the free entropy or *Gardner volume*, named in honor of the person who first performed this calculation. It is defined as

$$\phi = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathcal{D}} \log \mathcal{Z}. \tag{2.13}$$

To perform averages such as $\mathbb{E}_{\mathcal{D}} \log \mathcal{Z}(\mathcal{D})$, we will employ the *replica trick*. The main idea of the method is to use the simple identity

$$\log \mathcal{Z} = \lim_{n\to 0} \frac{\mathcal{Z}^n - 1}{n}. \tag{2.14}$$

Rather than computing the expectation of the logarithm, which we wouldn't know how to do, we can calculate the expectation $\mathbb{E}_{\mathcal{D}} \mathcal{Z}^n$ and then take the limit $n \to 0$. For a generic n, this still seems looks like a dead end, but if we promote n to an integer, then we can surely write

$$\mathcal{Z}^n = \int \prod_{a=1}^{n} d\mu(\boldsymbol{W}_a) \prod_{a=1}^{n} \prod_{\mu=1}^{P} \Theta\Big(- y^\mu \hat{y}_{\boldsymbol{W}_a}(\boldsymbol{x}^\mu)\Big). \tag{2.15}$$

We have written the power as a multiplication of n copies, or *replicas,* of the system, each with the same realization of disorder. For now, the replicas are non-interacting, since the term we have derived them from is factorized. At this point however it is possible to take the average over the dataset, and we will see that this will lead to an interaction between replicas. Let us suppose for example that we are considering the storage problem, with gaussian iid examples and labels which are ± 1 with probability one

$$\mathbb{E}_{\mathcal{D}} \mathcal{Z}^n = \mathbb{E}_x \mathbb{E}_y \int \prod_{a=1}^{n} d\mu(\boldsymbol{W}_a) \prod_{a=1}^{n} \prod_{\mu=1}^{P} \Theta\Big(- y^\mu \text{sign}\big(\frac{1}{\sqrt{N}} \sum_i x_i^\mu W_i^a\big)\Big). \tag{2.16}$$

Before going on let us note a simple fact: the distribution we have chosen for the examples and labels satisfies the properties $P(\boldsymbol{x}^\mu) = P(-\boldsymbol{x}^\mu)$ and $P(y^\mu) = P(-y^\mu)$. For every example x^μ with label $y^\mu = -1$, if we substitute the example-label pair with its opposite, the integrand doesn't change. This effectively means that we can take all labels to be $+1$, and perform one less average

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \mathbb{E}_x \int \prod_{a=1}^{n} d\mu(\boldsymbol{W}_a) \prod_{a=1}^{n}\prod_{\mu=1}^{P} \Theta\Big(-\text{sign}\big(\frac{1}{\sqrt{N}}\sum_i x_i^\mu W_i^a\big)\Big). \tag{2.17}$$

At this point we perform the usual trick in theoretical physics of multiplying the integrand by 1:

$$1 = \int \prod_{\mu a} d\lambda_{\mu a} \prod_{\mu a} \delta\Big(\lambda_{\mu a} - \frac{1}{\sqrt{N}}\sum_i x_i^\mu W_i^a\Big) = \tag{2.18}$$

$$= \int \prod_{\mu a} \frac{d\lambda_{\mu a} d\hat{\lambda}_{\mu a}}{2\pi} e^{i\sum_{\mu a}\lambda_{\mu a}\hat{\lambda}_{\mu a} - i\sum_{\mu a}\hat{\lambda}_{\mu a}\frac{1}{\sqrt{N}}\sum_i x_i^\mu W_i^a}, \tag{2.19}$$

where in the last equality we have used the Fourier representation of the Dirac delta. The expression becomes

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \mathbb{E}_x \int \prod_a d\mu(\boldsymbol{W}_a) \prod_{\mu a} \frac{d\lambda_{\mu a} d\hat{\lambda}_{\mu a}}{2\pi} e^{i\sum_{\mu a}\lambda_{\mu a}\hat{\lambda}_{\mu a} - i\sum_{\mu a}\hat{\lambda}_{\mu a}\frac{1}{\sqrt{N}}\sum_i x_i^\mu W_i^a} \prod_{\mu a}\Theta(-\lambda_{\mu a}). \tag{2.20}$$

Now we are finally in the position to average over the examples, by using the usual gaussian integration laws

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \int \prod_a d\mu(\boldsymbol{W}_a) \prod_{\mu a} \frac{d\lambda_{\mu a} d\hat{\lambda}_{\mu a}}{2\pi} e^{i\sum_{\mu a}\lambda_{\mu a}\hat{\lambda}_{\mu a} - \frac{1}{2}\sum_\mu\sum_{ab}\hat{\lambda}_{\mu a}\hat{\lambda}_{\mu b}\frac{1}{N}\sum_i W_i^a W_i^b} \prod_{\mu a}\Theta(-\lambda_{\mu a}). \tag{2.21}$$

The expression at this point is factorized over the examples, so we can write

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \int \prod_a d\mu(\boldsymbol{W}_a) \Bigg(\prod_a \frac{d\lambda_a d\hat{\lambda}_a}{2\pi} e^{i\sum_a \lambda_a\hat{\lambda}_a - \frac{1}{2}\sum_{ab}\hat{\lambda}_a\hat{\lambda}_b\frac{1}{N}\sum_i W_i^a W_i^b} \prod_a \Theta(-\lambda_a)\Bigg)^{\alpha N}, \tag{2.22}$$

where we have used that $\alpha = P/N$. Finally, we introduce the order parameter $Q_{ab} = \frac{1}{N}\sum_i W_i^a W_i^b$, which represents the interaction between replicas induced by the average over the data. More specifically, for $a \neq b$ it represents the overlap between two different replicas, while for $a = b$ it represents the norm of the weight vector (divided by N). If we are considering a spherical perceptron, then this norm will have to be fixed to 1 via a Lagrange Multiplier, while if we are

considering the binary perceptron it is automatically 1. Again, introducing deltas as we have done above, we get to the final expression

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \int \prod_{ab} dQ_{ab} d\hat{Q}_{ab} e^{nN\phi(Q_{ab},\hat{Q}_{ab})}, \tag{2.23}$$

$$\phi(Q_{ab},\hat{Q}_{ab}) = \frac{1}{n}G_I + \frac{1}{n}G_s + \frac{\alpha}{n}G_E, \tag{2.24}$$

$$G_I = i\sum Q_{ab}\hat{Q}_{ab}, \tag{2.25}$$

$$G_S = \frac{1}{N}\log \int \prod_a d\mu(\boldsymbol{W}_a) e^{-i\sum_{ab}\hat{Q}_{ab}\sum_i W_i^a W_i^b}, \tag{2.26}$$

$$G_E = \log \int \prod_a \frac{d\lambda_a d\hat{\lambda}_a}{2\pi} e^{i\sum_a \lambda_a\hat{\lambda}_a - \frac{1}{2}\sum_{ab}\hat{\lambda}_a\hat{\lambda}_b Q_{ab}} \prod_a \Theta(-\lambda_a). \tag{2.27}$$

We have now reduced a high-dimensional integral to an integral over an $n \times n$ matrix, where we will have to take the $n \to 0$ limit. Moreover, in the limit $N \to \infty$ we can apply Laplace's Method and write $\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n \approx e^{nN\phi(Q_{SP},\hat{Q}_{SP})}$ (here we ignore multiplication factors since we will later take the logarithm and divide by N) where the matrices Q_{SP} and $\hat{Q}_{SP}$ are obtained from the saddle point of ϕ. As we can see the quantity ϕ is composed of three terms: G_I is the interaction term, which connects the values of Q and $\hat{Q}$ at the saddle point, G_S is the entropic term, which measures the volume of configurations that realize a certain overlap matrix Q, and G_E is an energetic term which measures the volume of the configurations which are solutions. To effectively calculate the saddle-point though, we have to make an ansatz on the form of the overlap matrix Q and its Lagrange multiplier $\hat{Q}$. The simplest possible choice is to assume that the overlap is independent of the specific replica couple considered:

$$Q_{ab} = q(1-\delta_{ab}) + 1\delta_{ab}, \tag{2.28}$$

$$\hat{Q}_{ab} = \begin{cases} -i\frac{\hat{q}}{2}(1-\delta_{ab}) + \frac{\hat{h}}{2}\delta_{ab}, & \text{for the Spherical Perceptron} \\ -i\frac{\hat{q}}{2}(1-\delta_{ab}). & \text{for the Binary Perceptron} \end{cases} \tag{2.29}$$

This ansatz is called the *Replica Symmetric* ansatz. We will see that for the problem we are considering it will give the correct results, but in general a more complicated ansatz will be needed. In the spherical case, $\hat{h}$ acts as the Lagrange multiplier which fixes the norm of the weights, while no such parameter is needed for the binary case. Now what is left is to substitute this ansatz in the expression for ϕ, and calculate the saddle points with respect to q,$\hat{q}$ and $\hat{h}$ if we are in the spherical case.

Interaction Term It is a simple matter of exercise to calculate the interaction term by substituting the replica symmetric ansatz in the expression for G_I. This gives

$$G_I = \frac{1}{2}n(n-1)q\hat{q} + \frac{1}{2n}n\hat{h} \tag{2.30}$$

Taking the $n \to 0$ limit and dividing by n we get

$$\mathcal{G}_I = \lim_{n\to 0}\frac{1}{n}G_I = -\frac{1}{2}q\hat{q} + \frac{1}{2}\hat{h} \tag{2.31}$$

Entropic Term For the moment let us stick to the spherical case. First of all we note that the spherical constraint is enforced by the term $\hat{h}$, so the we can ignore the distribution term $d\mu(\boldsymbol{w}_a)$. The expression then factors over the index i, so again we can write it as

$$G_S = \frac{1}{N}\log\left(\int\prod_a dW_a e^{-\sum_{ab}\hat{Q}_{ab}W^aW^b}\right)^N = \log\int\prod_a dW_a e^{-\sum_{ab}\hat{Q}_{ab}W^aW^b}. \tag{2.32}$$

Now substituting our replica symmetric overlap matrix in the expression for G_S we get

$$G_S = \log\int\prod_a dw_a e^{-\frac{1}{2}\hat{q}\left(\sum_a W^a\right)^2 - \frac{\hat{h}-\hat{q}}{2}\sum_a (W^a)^2}. \tag{2.33}$$

Our objective is to factorize the expression over the replica index n. In order to do so, let us introduce the *Hubbard-Stratonovich* transform

$$e^{\frac{1}{2}A^2} = \int Dx\, e^{xA}, \tag{2.34}$$

where $Dx = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}dx$. This property follows directly from the rules of Gaussian integration. It allows us to linearize the term $\left(\sum_a w^a\right)^2$ at the cost of introducing a new gaussian integration variable:

$$G_S = \log\int Dx\prod_a dw_a e^{i\sqrt{\hat{q}}x\sum_a W^a - \frac{\hat{h}-\hat{q}}{2}\sum_a (W^a)^2}. \tag{2.35}$$

We have now achieved our objective: the expression is factorized over a and we can write

$$G_S = \log\int Dx\left(\int dW\, e^{i\sqrt{\hat{q}}xW - \frac{\hat{h}-\hat{q}}{2}W^2}\right)^n. \tag{2.36}$$

Bearing in mind that we will have to divide by n and take the $n \to 0$ limit, we can take do this now and write

$$\mathcal{G}_S = \lim_{n\to 0} \frac{1}{n} G_S = \int Dx \log \int dW\, e^{i\sqrt{\hat{q}}xW - \frac{\hat{h}-\hat{q}}{2}W^2} = \int Dx \log e^{-\frac{\hat{q}x^2}{2(\hat{h}-\hat{q})}} \sqrt{\frac{2\pi}{\hat{h}-\hat{q}}} = \tag{2.37}$$

$$= -\frac{\hat{q}}{2(\hat{h}-\hat{q})} + \frac{1}{2}\log\frac{2\pi}{\hat{h}-\hat{q}}. \tag{2.38}$$

Energetic Term Again, substituting our ansatz for the overlap matrix we have

$$G_E = \log \int \prod_a \frac{d\lambda_a d\hat{\lambda}_a}{2\pi} e^{i\sum_a \lambda_a\hat{\lambda}_a - \frac{1}{2}\left(\sum_a \hat{\lambda}_a\right)^2 q - \frac{1}{2}(1-q)\sum_a(\hat{\lambda}_a)^2} \prod_a \Theta(-\lambda_a). \tag{2.39}$$

Again, using the Hubbard-Stratonovich transform to linearize the quadratic term and factorizing the integral we get

$$G_E = \log \int Dx \prod_a \frac{d\lambda_a d\hat{\lambda}_a}{2\pi} e^{i\sum_a \lambda_a\hat{\lambda}_a - ixq\sum_a \hat{\lambda}_a - \frac{1}{2}\sum_a(\hat{\lambda}_a)^2} \prod_a \Theta(-\lambda_a) = \tag{2.40}$$

$$= \log \int Dx \left(\int \frac{d\lambda d\hat{\lambda}}{2\pi} e^{i\lambda\hat{\lambda} - ixq\hat{\lambda} - \frac{1}{2}(1-q)(\hat{\lambda})^2} \Theta(-\lambda) \right)^n. \tag{2.41}$$

As before we can perform the $n \to 0$ limit and get

$$\mathcal{G}_E = \lim_{n\to 0} \frac{1}{n} G_E = \int Dx \log \int \frac{d\lambda d\hat{\lambda}}{2\pi} e^{i\lambda\hat{\lambda} - ixq\hat{\lambda} - \frac{1}{2}(1-q)(\hat{\lambda})^2} \Theta(-\lambda).$$

Finally, performing the integrals in $d\lambda$ and $d\hat{\lambda}$ we get

$$\lim_{n\to 0} G_E = \int Dx \log \int \frac{d\lambda}{\sqrt{2\pi(1-q)}} e^{-\frac{1}{2}\frac{(\lambda - xq)^2}{1-q}} \Theta(\lambda) = \tag{2.42}$$

$$= \int Dx \log \int D\lambda \Theta(\sqrt{1-q}\lambda + qx) = \int Dx \log H\Big(-\frac{q}{\sqrt{1-q}}x\Big), \tag{2.43}$$

where we have introduced the function

$$H(x) = \int_x^\infty Dy = \frac{1}{2}\mathrm{erfc}\left(\frac{x}{\sqrt{2}}\right). \tag{2.44}$$

Saddle Point Putting everything together, we have that the Gardner volume is

$$\phi = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathcal{D}} \log \mathcal{Z} = \lim_{N\to\infty} \lim_{n\to 0} \frac{1}{N} \mathbb{E}_{\mathcal{D}} \frac{\mathcal{Z}^n - 1}{n}. \tag{2.45}$$

Even though some of the steps might not have seem entirely justified, there are no particularly unreasonable steps in the method up to this point. Now comes the main reason why the replica calculation can only be considered an heuristic method, and cannot be formally justified: to go on in our calculation we must swap the limits in N and n

$$\phi = \lim_{n\to 0} \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathcal{D}} \frac{\mathcal{Z}^n - 1}{n}. \tag{2.46}$$

Applying Laplace's Method then we can write

$$\phi = \lim_{n\to 0} \lim_{N\to\infty} \frac{1}{N} \frac{e^{nN\phi(q^*,\hat{q}^*,\hat{h}^*)} - 1}{n} = \lim_{n\to 0} \phi(q^*, \hat{q}^*, \hat{h}^*), \tag{2.47}$$

where $(q^*, \hat{q}^*, \hat{h}^*)$ are the values of the order parameters at the saddle point, and can be obtained by solving the equations

$$\frac{\partial}{\partial q}\phi = 0 \tag{2.48}$$

$$\frac{\partial}{\partial \hat{q}}\phi = 0 \tag{2.49}$$

$$\frac{\partial}{\partial \hat{h}}\phi = 0 \tag{2.50}$$

This can be easily done numerically. The results are shown in figure 2.1.

As we would expect, the overlap parameter q is increasing as a function of α, and tends to the value 1 for $\alpha \to 2$. This signals the fact that as the pattern to dimension ratio is increased, the Gardner volume decreases in size, and thus solutions are forced to be closer together, effectively leading to a greater overlap. When the overlap reaches the value $q = 1$, the volume has reduced to a point, signaling that beyond this point the space of solutions is empty. We can thus identify $\alpha_c = 2$ as the *critical capacity* or *SAT-UNSAT transition* of the spherical perceptron, beyond which no solution exists. This property was known well before the calculation fo the Gardner volume, thanks to an elegant combinatorial approach developed by Cover in the 1960s [Cover, 1965]. As for the volume ϕ, this decreases as a function of α and reaches the value $-\infty$ for $\alpha = \alpha_c$.

In this section we have seen how the Replica Method can be used to answer questions about the solution space of linear classifiers. We have seen the details of the simplest possible replica

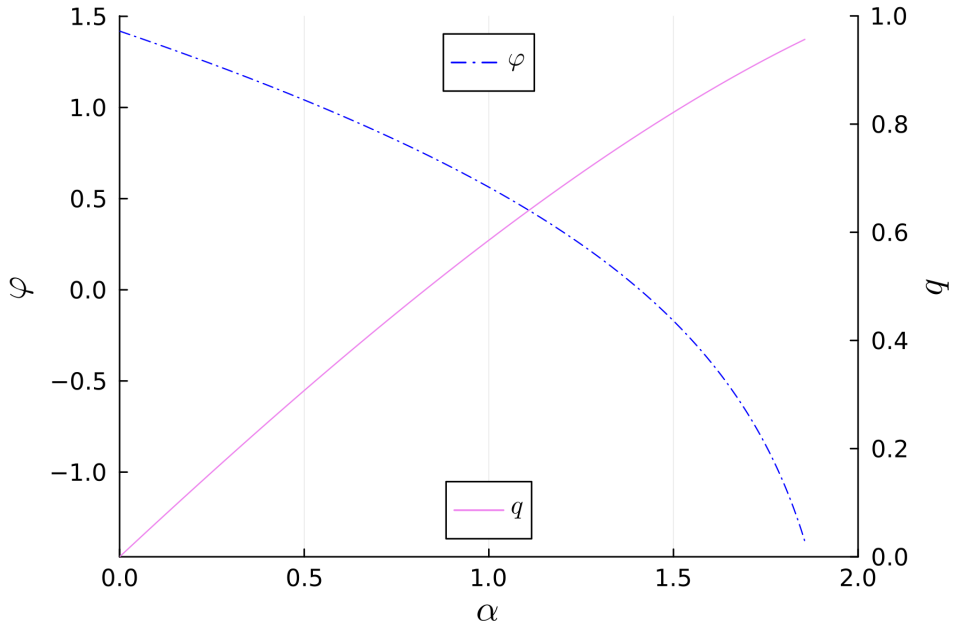

Figure 2.1: The overlap q and free entropy ϕ as a function of α for the spherical perceptron.

calculation, and how it can be used to deduce properties about particular phase transitions. In the next chapter, we will look at a more complex application of the replica method, for the study of the solution landscape of the negative perceptron.

2.3 Adding Robustness: the Margin

In the $N \to \infty$ limit the uniform distribution over solutions defined in equation (2.9) will be exponentially dominated by solutions which enjoy common features. For example, if we sample two solutions uniformly $\boldsymbol{W}_1, \boldsymbol{W}_2 \sim P_{\mathcal{D}}$, then their overlap $\frac{1}{N}\boldsymbol{W}_1 \cdot \boldsymbol{W}_2$ will tend in the $N \to \infty$ limit to a deterministic value, which is equal to the value q determined by the Saddle Point equations (2.48). If we define the stability of a network on a certain pattern as

$$\Delta = y^\mu \frac{1}{\sqrt{N}} \boldsymbol{W} \cdot \boldsymbol{x}^\mu, \tag{2.51}$$

using the replica method it is also possible to calculate the distribution of stabilities of these solutions

$$\rho_{\mathcal{D}}(\Delta) = \langle \frac{1}{P} \sum_{\mu=1}^{P} \delta\left(\Delta - y^\mu \frac{1}{\sqrt{N}} \boldsymbol{W} \cdot \boldsymbol{x}^\mu\right) \rangle, \tag{2.52}$$

where by $\langle . \rangle$ we mean the average with respect to distribution $P_{\mathcal{D}}(\boldsymbol{W})$. Like the free entropy this is a self-averaging quantity, so in the asymptotic limit all samples will share the same profile of stabilities, given by the average $\mathbb{E}_{\mathcal{D}}\rho_{\mathcal{D}}(\Delta)$. One might wonder if solutions that exhibit these properties are the *only* solutions: the answer is of course negative. Solutions that dominate the distribution, and thus enjoy these shared properties, are referred to as *typical* solutions. If one had a sampler for the distribution $P_{\mathcal{D}}$, and looked in the large N limit, these would be the only solutions he would see (with high probability). However, an exponential number of *atypical* solutions still exist, and it has been argued that to understand the functioning of common learning algorithms, these solutions are essential [Baldassi et al., 2015]. One way to analytically access atypical solutions, is to add a robustness requirement to the loss function defined in equation (2.8). This is typically done by adding a margin κ to the loss

$$\mathcal{L}(\Theta; \mathcal{D}) = \sum_{\mu=1}^{P} \Theta\left(-y^{\mu}\hat{y}_{\boldsymbol{W}}(\boldsymbol{x}^{\mu}) + \kappa\right). \tag{2.53}$$

Intuitively, this means that we require not only that all examples with the same label lie on the same side of the decision boundary, but also that they are not too close to it. For example, if the label of a pattern is $y^{\mu} = 1$, then we require the scalar product $\frac{1}{\sqrt{N}}\boldsymbol{W} \cdot \boldsymbol{x}^{\mu} \geq \kappa$. Adding a positive margin thus leads to solutions which are more robust. If we use this loss to define the probability distribution $P_{\mathcal{D},\kappa}(\boldsymbol{W}) \propto \mu(\boldsymbol{W}) \prod_{\mu} \Theta\left(-y^{\mu}\hat{y}_{\boldsymbol{W}}(\boldsymbol{x}^{\mu}) + \kappa\right)$, then samples of this distribution will still correctly classify dataset $\mathcal{D}$. That is, *typical* samples of $P_{\mathcal{D},\kappa}(\boldsymbol{W})$ are *atypical* samples of $P_{\mathcal{D},0}(\boldsymbol{W})$. There are of course other ways to access atypical solutions, but this is a simple and efficient way.

One might wonder how these solutions are organized in the space of all solutions. In [Baldassi et al., 2021] authors show that they are organized in a hierarchical fashion: high margin solutions are surrounded by lower margin solutions, which are in turn surrounded by solutions with even smaller margin.

Although historically applications in Machine Learning have been restricted to positive margins, nothing prevents us from considering $\kappa < 0$. This model goes under the name of the *Negative Perceptron*. Although already studied in [Stojnic, 2013], interest in this model was reignited since it was noted in [Franz and Parisi, 2016] that it is isomorphic to the problem of jamming in infinite dimensions and displays many of the critical properties of the jamming transition of soft matter systems [Liu and Nagel, 2010]. Indeed if we consider a particle in position $\boldsymbol{W}$ of radius σ, lying in a random background of obstacles in positions $\{\boldsymbol{x}_{\mu}\}_{\mu=1}^{P}$, the problem of finding a position such that $\forall \mu \in \{1, ..., P\}\ |\boldsymbol{x}_{\mu} - \boldsymbol{W}| > \sigma$, is isomorphic to finding a solution to the perceptron with $\sigma^2 = 2N + 2\kappa$. As we can see nothing prevents us from considering $\kappa < 0$, and indeed the negative margin case is more interesting. This is

because while the space of solutions for $\kappa \geq 0$ is convex, it becomes non convex for $\kappa < 0$. We will see that this will require a more sophisticated ansatz than the one defined in equations (2.28)-(2.29).

In this thesis however we will use the *Negative Perceptron* mainly as a model of machine learning. Let us start by giving an example where it might make sense to study the perceptron with negative margin. Suppose we are given a dataset $\{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^P$ with binary labels that is linearly separable, that is there exists a decision boundary $\boldsymbol{W}_0$ such that the clouds of points with label -1 are on one side of the boundary and the points with label $+1$ on the other. Let us further suppose that the minimum distance between the points and the decision boundary is $\kappa > 0$. Now let us take a corrupted version of this dataset, where each point $\boldsymbol{x}^\mu$ is corrupted with noise $\boldsymbol{x}_c^\mu = \boldsymbol{x}^\mu + \boldsymbol{\epsilon}^\mu$ with $\|\boldsymbol{\epsilon}^\mu\| < \sigma$. As we can see with high probability the minimum distance between the points of dataset $\{\boldsymbol{x}_c^\mu, y^\mu\}_{\mu=1}^P$ and the decision boundary will become $\kappa - \sigma$, so for $\sigma > \kappa$ we will have that the dataset will not be linearly separable. In this setting then it might be sensible to study the set of decision boundaries with negative margin κ.

Inspired by this setting, in [Montanari et al., 2021] authors study the Negative Perceptron as a model of an overparametrized network. Although the usual concept of overparametrization, where the dimensionality of the data and the size of the dataset are kept fixed while the number of parameters of the network is increased, cannot be modeled by a simple one layer perceptron, the same hard-easy transition they argue can be achieved by decreasing the margin κ. Indeed in the same way that in real world Machine Learning applications as the number of parameters of the network is increased the final loss achieved drops to zero, the same happens for the perceptron as κ is decreased. Understanding this hard-to-easy transition in a simple non-convex model could thus be a proxy for understanding more realistic scenarios. In their work authors present upper and lower bounds for the value of this SAT-UNSAT transition $\alpha_c(\kappa)$, and compare them to the algorithmic threshold of a simple linear programming surrogate of the original non-convex problem. They find that there exists a sizeable gap between the lower bound of the capacity and the threshold of their algorithm, although they also empirically observe that Gradient Descent performs better than their algorithm.

In [Baldassi et al., 2023] authors apply a method, previously used to predict the algorithmic threshold of binary models such as the binary perceptron, to the Negative Perceptron. They find that in this case their method, which will be described in the next chapter, gives a prediction which can be easily overcome with simple Gradient Descent algorithms. Indeed they note that, although their predictions for the SAT-UNSAT transition are only approximate, Gradient Descent finds solution very close to this threshold.

2.4 Adding Structure to Data: the Hidden Manifold Model

In this section we will see how it's possible to consider a more realistic model of data. All results up to this point have considered a factorized input distribution, clearly a very unrealistic assumption for real world datasets which are known to be highly structured. Indeed one of the many questions which are driving research in the foundations of machine learning is the impact of the structure of data on learning. Although the input dimension of data is usually very large, for example in the case of computer vision the number of pixels of an image, common wisdom is that the actual "intrinsic" dimension of each dataset is much lower. In this view, data is supposed to live on a manifold of much lower dimension, embedded in a high dimension space.

Inspired by this view, authors in [Goldt et al., 2020] introduce a new model of data that goes under the name of *Hidden Manifold Model*. They suppose that data is generated by taking a latent vector $\boldsymbol{\xi} \in \mathbb{R}^D$, taken from a factorized distribution $\xi_i \overset{iid}{\sim} P_\xi \ \forall i \in [D]$, and then projecting it to a higher dimensional space according to

$$\boldsymbol{x}^\mu = \varphi\left(\frac{1}{\sqrt{D}} F \boldsymbol{\xi}^\mu\right), \tag{2.54}$$

where $F \in \mathbb{R}^{N\times D}$ is a projection matrix and φ is a non-linear activation function applied component-wise. As we can see, this projection can be interpreted as the action of a fully connected neural network with random weights F. Thanks to this projection then the components of the features $\boldsymbol{x}^\mu$ acquire non-linear correlations, and furthermore the embedding dimension of the input N (which is also the dimension of the weight vector that will classify this input) is separated from the intrinsic dimension D. The authors consider the asymptotic limit where $N, D \to \infty$ with a fixed ratio $\alpha_D = D/N = O(1)$: as we can see this parameter controls the amount of overparametrization of the network. Coherently with the interpretation that the relevant information of each input is encoded in the latent vector, the labels are generated by a linear teacher that acts on the hidden vector $\boldsymbol{\xi}$

$$y^\mu = \varphi\left(\frac{1}{\sqrt{N}} \tilde{\boldsymbol{W}} \cdot \boldsymbol{\xi}^\mu\right). \tag{2.55}$$

The dataset $\mathcal{D} = \{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^P$ is thus a structured dataset, that presents a correlated input vector and mismatch between the space the label is generated on and the space the student actually sees.

Interestingly, the model bears many similarities with a learning technique called *Random Feature Learning*, where data is randomly projected into a different space before being classified with a linear machine. First used by Rosenblatt's Mark I Perceptron, it has been recast in a

modern scenario in the seminal work of Rahimi and Recht, where it was proposed as a trick to speed up kernel learning [Rahimi and Recht, 2007]. This model of learning is analyzed in [Mei and Montanari, 2019, Hastie et al., 2022] where authors consider ridge and ridgeless regression for a random projection matrix. The Hidden Manifold Model goes beyond this setting, as the projection matrix need not be random, and authors even numerically study the case in which the matrix is obtained by training a Generative Adversarial Network (GAN).

The main advantage of this data model is that even though the $\boldsymbol{x}$ are non-gaussian, in the high dimensional limit it can be shown that the joint distribution of outputs on a given pattern

$$y_a = \frac{1}{\sqrt{N}} \boldsymbol{W}_a \cdot \boldsymbol{x}, \tag{2.56}$$

$$v = \frac{1}{\sqrt{N}} \tilde{\boldsymbol{W}} \cdot \boldsymbol{\xi}, \tag{2.57}$$

given the set of weights $\{\boldsymbol{W}_a\}_a$ and $\tilde{\boldsymbol{W}}$ is asymptotically gaussian, with first and second moments

$$\mathbb{E} y_a = \kappa_0, \tag{2.58}$$

$$\mathbb{E} v = 0, \tag{2.59}$$

$$\mathbb{E} y_a y_b - \mathbb{E} y_a \mathbb{E} y_b = (\kappa_2 - \kappa_1^2 - \kappa_0^2) q_{ab}^w + \kappa_1^2 q_{ab}^Z, \tag{2.60}$$

$$\mathbb{E} v v = \tilde{q}, \tag{2.61}$$

$$\mathbb{E} y_a v - \mathbb{E} y_a \mathbb{E} v = \kappa_1 m_a, \tag{2.62}$$

where the scalars $\kappa_0, \kappa_1, \kappa_2$ depend on the activation function and are defined as

$$\kappa_0 = \int Dz \, \varphi(z), \tag{2.63}$$

$$\kappa_1 = \int Dz \, z \, \varphi(z), \tag{2.64}$$

$$\kappa_2 = \int Dz \, \varphi^2(z), \tag{2.65}$$

and the order parameters $q_{ab}^w, q_{ab}^Z, \tilde{q}$ and m_a are defined as

$$q^w_{ab} = \frac{1}{N}\boldsymbol{W}_a \cdot \boldsymbol{W}_b, \tag{2.66}$$

$$q^Z_{ab} = \frac{1}{N}\boldsymbol{W}_a^T \frac{FF^T}{D}\boldsymbol{W}_b, \tag{2.67}$$

$$\tilde{q} = \frac{1}{D}\|\tilde{\boldsymbol{W}}\|^2, \tag{2.68}$$

$$m_a = \frac{1}{N}\boldsymbol{W}_a \frac{1}{\sqrt{D}} F\tilde{\boldsymbol{W}}. \tag{2.69}$$

This result goes under the name of *Gaussian Equivalence Principle (GEP),* and was elevated to the status of theorem in [Goldt et al., 2022] under a set of hypotheses for F, φ and $\boldsymbol{W}, \tilde{\boldsymbol{W}}$. Intuitively, we require the following overlaps $\forall q, p \geq 1\ \forall k_1, \dots, k_p, r_1, \dots, r_q$ to be of order one

$$\frac{1}{\sqrt{N}} \sum_i w_i^{k_1} w_i^{k_2} \cdots w_i^{k_p} F_{ir_1} F_{ir_2} \cdots F_{ir_q} = O(1). \tag{2.70}$$

An intuitive derivation of the above is the following. Consider the variables

$$\boldsymbol{\lambda} = \frac{1}{\sqrt{D}} F\boldsymbol{\xi}. \tag{2.71}$$

For fixed feature matrix F their distribution is gaussian. Let us consider for example $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_D)$ and $F_{ij} \sim N(0,1)$. The variance and covariance of every component are

$$\mathbb{E}\lambda_i^2 = \frac{1}{D}\sum_k F_{ik}^2 = 1 + O\left(\frac{1}{\sqrt{D}}\right), \tag{2.72}$$

$$\mathbb{E}\lambda_i\lambda_j = \frac{1}{D}\sum_k F_{ik}F_{jk} = O\left(\frac{1}{\sqrt{D}}\right) \qquad i \neq j. \tag{2.73}$$

Although different components are correlated, their correlation is small, and this allows us to use the following result:

Lemma 1. *Take gaussian variables u_1 and u_2 with mean zero and covariances*

$$\mathbb{E}u_1^2 = 1 \qquad \mathbb{E}u_2^2 = 1 \qquad \mathbb{E}u_1u_2 = \epsilon m_{12}. \tag{2.74}$$

For two functions f_1 and f_2, define

$$a_i = \mathbb{E}f_i(u_i) \qquad b_i = \mathbb{E}u_i f_i(u_i) \qquad i = 1, 2. \tag{2.75}$$

Then, in the $\epsilon \to 0$ limit, the correlation between $f_1(u_1)f_2(u_2)$ is

$$\mathbb{E}f_1(u_1)f_2(u_2) = a_1a_2 + \epsilon m_{12}b_1b_2 + O(\epsilon^2). \tag{2.76}$$

The proof of this lemma can be found in [Goldt et al., 2020].

Applying this lemma to our case, we can write the covariance

$$\mathbb{E}\varphi(\lambda_i)\varphi(\lambda_j) = \kappa_0^2 + \frac{1}{D}\sum_k F_{ik}F_{jk}\kappa_1^2, \tag{2.77}$$

while for the variance we have simply

$$\mathbb{E}\varphi^2(\lambda_i) = \kappa_2. \tag{2.78}$$

Using the definitions of y_a

$$\mathbb{E}y_a = \kappa_0\frac{1}{\sqrt{N}}\sum_i W_i^a, \tag{2.79}$$

$$\mathbb{E}y_a y_b - \mathbb{E}y_a\mathbb{E}y_b = \frac{1}{N}\sum_{ij} W_i^a W_j^b\left[\mathbb{E}\varphi(\lambda_i)\varphi(\lambda_j) - \mathbb{E}\varphi(\lambda_i)\mathbb{E}\varphi(\lambda_j)\right] = \tag{2.80}$$

$$= \frac{1}{N}\sum_i W_i^a W_i^b\left[\kappa_2 - \kappa_0^2\right] + \frac{1}{N}\sum_{i\neq j} W_i^a W_j^b\frac{1}{D}\sum_k F_{ik}F_{jk}\kappa_1^2 = \tag{2.81}$$

$$= \frac{1}{N}\sum_i W_i^a W_i^b\left[\kappa_2 - \kappa_0^2 - \kappa_1^2\right] + \kappa_1^2\frac{1}{N}\sum_{ij} W_i^a\frac{1}{D}\sum_k F_{ik}F_{jk}W_j^b, \tag{2.82}$$

which is precisely the correlation given in equations (2.66)-(2.69). The correlation $\mathbb{E}\nu y_a$ can be derived in the same manner, but we will skip its derivation.

Interestingly, this is the same distribution we would obtain if the features $\boldsymbol{x}$ were

$$\boldsymbol{x}_{eq} = \kappa_0\mathbf{1} + \kappa_1\frac{1}{\sqrt{D}}F\boldsymbol{\xi} + (\kappa_2 - \kappa_1^2 - \kappa_0^2)\boldsymbol{\zeta}, \tag{2.83}$$

where both $\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbb{I}_D)$ and $\boldsymbol{\zeta} \sim \mathcal{N}(0, \mathbb{I}_N)$. That is, the non-linear features $\boldsymbol{x}$ are equivalent to noisy linear features $\boldsymbol{x}_{eq}$, at least when it comes to the distribution of activations (and therefore other observables such as the test and training error). It is important to note however that we are not implying that $\boldsymbol{x}$ and $\boldsymbol{x}_{eq}$ are equal in distribution, but rather that their first and second moments match, and that all is needed for the distribution of y_a, ν. Indeed the Gaussian Equivalence is nothing but a CLT-like result for the quantity $y = \frac{1}{\sqrt{N}}\boldsymbol{W}\cdot\boldsymbol{x}$ when the correlation among different components of $\boldsymbol{x}$ is small, and holds as long as the student weights $\boldsymbol{W}$ do not align "too much" with the projection matrix F. Equation (2.70) can then be understood

as the condition that the variable $\frac{1}{\sqrt{N}} \sum_i W_i F_{ir}$ and high order tensors stay of order 1 in the thermodynamic limit.

Part II

Novel Applications of Statistical Mechanics of Learning

Chapter 3

The Landscape of the Perceptron

One of the many aspects of Neural Networks that still remains elusive is the ease with which they are trained. The fundamental issue stems from the simple observation that the loss function, defined over the set of parameters, is non-convex, and the fact that neural networks are easily trained with simple variants of first order methods, that have no guarantee of reaching global minima for non-convex functions. One way to approach this problem is to study the landscape of the loss function and understand the geometry of its critical points. In modern day scenarios, in which the number of parameters is in the order of the billions, this space is very high-dimensional. Understanding the geometry of such high-dimensional landscapes is far from trivial. Many intuitive geometrical properties we are used to observing in low-dimensions don't translate to higher-dimensions, where peculiar and counter-intuitive properties often emerge [Milman, 1998, Martiniani and Casiulis, 2023].

In this chapter I study a model of a neural network, the Negative Perceptron, and give an analytic characterization of its training landscape. By looking at the connectivity of its solutions I identify a particular geometry, called *Star-Shaped*, which naturally emerges. The rest of the chapter is organized as follows: in section 3.1, I give a literature review on the topic, in section 3.2, I review the model, and in section 3.3, I show how this particular geometry can be inferred. The results from this chapter are based on a publication [Annesi et al., 2023].

3.1 The Landscape of Neural Networks

In the past eight to ten years, the landscape of Neural Networks has been the object of inquiry for a number of different scientific figures, ranging from Mathematicians, Physicists and ML practitioners. The picture that has emerged is a hybrid one, with different points of view giving different interpretations and results. Any review of the subject must encompass all voices, so

in this section, I review the main results coming from this wide range of fields, trying to give equal weight to every voice.

3.1.1 Optimization of Neural Networks: Local and Global Minima and Saddles

One of the most direct approaches to understanding the interaction between gradient based methods and a landscape would be to look at the relationship between local and global minima. In [Kawaguchi, 2016] the author proves a long standing conjecture, according to which no suboptimal local minima exist for deep linear networks. Inspired by this work, and the empirical observation that training a neural network is "easy", some efforts have been devoted at proving a similar result for non-linear networks, under some overparametrization assumption (since it is also observed that optimizing underparametrized neural networks is not as easy). Ultimately it was shown that this claim is too strong, and authors in [Ding et al., 2019] provided a clear counterexample. A line of work then switched to proving the non-existence of "spurious valleys" [Venturi et al., 2018a] and "suboptimal basins" [Li et al., 2018] for sufficiently overparametrized networks.

A different approach, less rigorous in its nature, comes from the field of Statistical Mechanics of Disordered Systems. Throughout the years, a number of tools based on the Kac-Rice method have emerged to count the number of critical points of a high-dimensional non-convex landscape displaying certain properties [Azaïs and Wschebor, 2009]. These techniques, although technically non-rigorous, have been successfully applied to many well known spin-glass models, and have helped characterize the "roughness" of the landscape [Fyodorov and Tublin, 2019, Ros et al., 2019a,b]. Unfortunately, calculations are limited to simple Gaussian cases, and with a few exceptions [Choromanska et al., 2015, Maillard et al., 2020], cannot be extended to the non-linearities that occur in Learning.

One possibility however is to proceed by comparison: by empirically measuring the phenomena that is known to occur in well studied "glassy" landscapes, for models that are prone to analytical results, it is possible to obtain some information on the landscapes of neural networks. In [Baity-Jesi et al., 2018] authors do precisely this: they compare the dynamics of Neural Networks with the dynamics of the *p-spin* model. By looking at the *mean square displacement*

$$\Delta(t_w, t_w + t) = \frac{1}{M} \sum_{i=1}^{M} (w_i(t_w + t) - w_i(t_w))^2 , \tag{3.1}$$

where the sum is over all M parameters of the neural network, they identify three regimes in the gradient descent dynamics:

1. In the first regime the loss remains approximately constant and the mean square displacement shows no clear dependence on t_w.

2. In the intermediate regime the loss quickly decreases and the mean square displacement plateaus for a time that increases with t_w. This phenomenon goes under the name of ageing, and is a well known characteristic of the dynamics of Disordered Systems.

3. In the final regime the system diffuses around the bottom of the landscape, and again the mean square displacement shows no dependency on t_w.

Although these results do make an important connection between the two landscapes, most notably by observing that ageing occurs in certain phases of the training of a neural network, they also highlight important differences. Indeed in spin-glasses the third regime is not present, as the dynamics gets stuck in high energy-wide basin local minima. Authors explain this discrepancy by suggesting that ageing occurs for different reasons in the two models: in the spin glasses it is caused by barrier crossing of local-minima, while in neural networks by the proliferation of flat directions as optimization proceeds.

As a confirmation of this intuition, authors in [Sagun et al., 2016] look at the spectrum of the Hessian for realistic Neural Networks, both before, during and after training. They observed two interesting facts:

1. The Hessian after training contains a majority of near-zero eigenvalues, indicating a proliferation of flat directions.

2. The spectrum consists mainly of two phases: a first bulk around zero that depends on the architecture of the network, and another part away from the bulk that is data-dependent.

This suggests that rather than thinking of isolated global minima, one should imagine the bottom of the loss landscape as an extremely degenerate and flat manifold. As a rigorous confirmation of this observation, [Cooper, 2018] shows in a rather general setting that the locus of global minima of the loss is not discrete, but rather an $n-d$ dimensional submanifold where n is the number of parameters of the network and d is the dimensionality of the data.

3.1.2 Generalization Performance of Neural Networks: Flatness

Up to now we have been ignoring one fundamental aspect of machine learning: the remarkable performance of networks on new unseen data. Indeed understanding how neural networks

generalize is one of the core tasks of machine learning theory. Again this question can be approached by investigating the landscape of the loss function, and its relation to generalization performance. By now, a wide number of works have reported a general correlation between the flatness of a minimum, typically measured by the trace of the hessian, and the generalization performance of the corresponding network [Liang et al., 2019, Keskar et al., 2016, Petzka et al., 2021, Jiang et al., 2019, Hochreiter and Schmidhuber, 1997]. Authors in [Petzka et al., 2021] give an exhaustive explanation of this phenomenon by introducing a new measure of flatness. Their reasoning goes as follows. First of all they suppose that a neural network function $f(x)$ learns by extracting features $\phi(x) \in \mathbb{R}^m$ and doing a linear projection on these features $f(x) = \boldsymbol{w}^T \phi(x)$, and define *feature robustness* as the robustness of the output with respect to the perturbed of the features $\phi_A(x) = (\mathbb{I} + A)\phi(x)$. They then derive bounds relating this notion to a measure they introduce called *relative flatness*, which can be thought of as a measure of flatness only with respect to the "readout" parameters $\boldsymbol{w}$. Finally they show that under the hypothesis that the distribution of the features induced by the data is smooth and the labels are approximately locally constant, then the generalization gap is bounded by this flatness measure. The exhaustiveness of this approach is such that the claim is solid with respect to attacks such as those pointed out in [Dinh et al., 2017], where it was shown that for deep networks with ReLU activations it is always possible to reparametrize the network in such a way that keeps the generalization error fixed but increases the flatness measure.

One might wonder if it is possible to find a minimal neural network model in which a flatness measure can be calculated analytically. This is precisely what a line of work inspired again by the Disordered Systems has done [Baldassi et al., 2015, 2016, 2021]. Here, authors consider a model of a simple one layer network, a synthetic dataset that consists of iid Gaussian samples, and analytically study the distance between solutions, that is networks that achieve zero training error. They introduce the notion of *Local Entropy*, that given a reference network $\tilde{\boldsymbol{W}}$ can be defined as the log of the volume of all solutions at distance d from the reference

$$S_{\mathcal{D}}(\tilde{\boldsymbol{W}}, d; \kappa) = \frac{1}{N} \log \int d\mu(\boldsymbol{W}) \mathbb{X}_{\mathcal{D}}(\boldsymbol{W}; \kappa) \, \delta\left(\frac{\boldsymbol{W} \cdot \tilde{\boldsymbol{W}}}{N} - (1 - 2d) \right), \tag{3.2}$$

where $\mathbb{X}_{\mathcal{D}}(\boldsymbol{W}; \kappa)$ is the uniform measure over all solutions for a given dataset $\mathcal{D}$ and margin κ. Taking the average with respect to the reference network we obtain the so called *Franz-Parisi Potential*

$$\phi_{FP}(d, \kappa, \tilde{\kappa}) = \mathbb{E}_{\mathcal{D}} \int d\mu(\tilde{\boldsymbol{W}}) \mathbb{X}_{\mathcal{D}}(\tilde{\boldsymbol{W}}; \tilde{\kappa}) S_{\mathcal{D}}(\tilde{\boldsymbol{W}}, d; \kappa), \tag{3.3}$$

a quantity that was first introduced in the field of Spin Glasses in the '80s [Franz and Parisi, 1995]. By varying κ and $\tilde{\kappa}$ and looking at the potential for small d, it is possible to gather information about the distance between solutions, and the organization of such solutions. The authors conclude the following:

1. Solutions are organized in an hierarchical fashion: solutions with large margin are surrounded by solutions with smaller margin, which in turn are surrounded by other solutions with even smaller margin and so on.

2. In some phases of learning there exists a "wide flat minimum", a cluster of solutions of extensive width.

3. In binary models, where the parameters of the network are binary, the disappearance of this cluster coincides with the point in which algorithms stop finding solutions.

4. Solutions found by algorithms belong to this wide flat minimum.

As we can see even in simple one layer models the notion of flatness can be defined and used to explain algorithmic performance. Indeed the whole framework was introduced to resolve an apparent paradox noted in [Braunstein and Zecchina, 2006], that is that although typical solutions for the binary perceptron are isolated, and should thus be hard to find, simple message-passing based algorithms succeed at finding solutions.

These arguments, mainly stemming from the need to explain the incredible generalization performance of neural networks, have nonetheless inspired a number of algorithms for training neural networks which target flatter solutions. Based on the "Local Entropy" approach, authors in [Chaudhari et al., 2019] propose a training algorithm called Entropy-SGD which explicitly inserts the local entropy in the loss function (in such a way that the final solution has high local entropy), and can be used to train deep networks. They report a gain in training loss but comparable generalization error compared to regular SGD. Similar in spirit but different in technical realization, Sharpness Aware Minimization introduced in [Foret et al., 2020], seeks parameters that lie in neighborhoods of uniformly low loss. In this case they do report a substantial gain in generalization performance on a series of benchmark tasks.

3.1.3 Connectivity of Minima

Another approach that has emerged to understand the geometry of the landscape, more empirical in its nature, is to look at so called *Mode Connectivity*. In the typical experiment, given a fixed dataset and architecture two neural networks are trained until convergence, and then joined with some path. If the training loss along this path stays more or less constant the

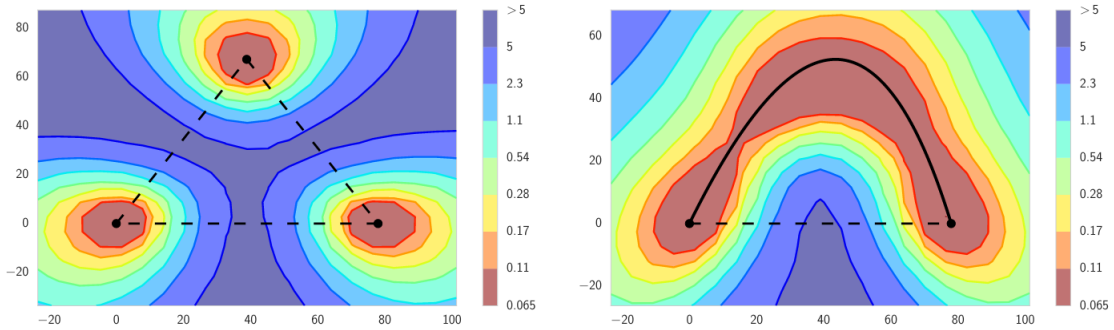

Figure 3.1: The loss landscape between three neural networks. On the left the landscape on the linear plane that joins the three minima, on the right the landscape along the curved path they find that connects them. Reproduced from [Garipov et al., 2018].

two networks are said to be connected, if instead a barrier is encountered then they are disconnected. This methodology has the advantage of being very straightforward, but has nonetheless shown some non-trivial results and applications. Some of the first works in this line of research have been [Draxler et al., 2018, Garipov et al., 2018], where authors noted that although the linear paths connecting two networks present barriers, it is always possible to construct simple polygonal paths with one bend along which both the training and test losses remain (approximately) constant (see figure 3.1). They propose an algorithm for finding such paths, and use this to construct better ensembling procedures. This discovery highlights the fact that again when picturing these minimizers one should think of a low dimensional manifold which connects all minima, rather than distinct valleys. Moreover, it has been recently proposed that barriers between minimizers would actually go to zero if one took into account the permutation symmetry of all hidden neurons [Pittorino et al., 2022, Entezari et al., 2021]. This strong hypothesis is made on the basis of the observation that by using simple "matching" algorithms that find a permutation that makes two networks as similar as possible, then the linear barriers between these matched networks dropped significantly. If this were true then the non-convexity of the landscape would be a simple artifact of a symmetry in the space of parameters. As a confirmation of this intuition, authors in [Frankle and Carbin, 2018, Frankle et al., 2020] study *Linear Mode Connectivity*, where the path between the two minima is linear, and look at how the connectivity of minima evolves throughout training. They perform the following experiment:

1. They train a network for a number e of epochs.

2. They duplicate the network, and train the two copies using different realizations of SGD noise (so different mini-batches).

3. They look at the linear mode connectivity of the final networks as a function of e.

They find that although minima are disconnected when $e = 0$ (a confirmation of what found in [Draxler et al., 2018, Garipov et al., 2018]), they become connected after a small number of epochs, usually at around 1% of the total amount of training. This observation can be explained by imagining that the first few epochs break the permutation symmetry between neurons, and once the "gauge" is fixed all networks belong to the same basin. The authors use this finding to justify a parameter which they use in a pruning algorithm they propose, Iterative Magnitude Pruning, as an indication that the observation is not purely speculative but does have some practical value.

Although most efforts in this direction have been empirical, some theoretical results have been derived to validate this picture. In [Freeman and Bruna, 2016] authors prove that the sub-level sets, that is the sets $F_c = \{\boldsymbol{\theta} | \mathcal{L}(\boldsymbol{\theta}) \leq c\}$, for networks with ReLU-type activations are asymptotically connected, that is given two minima there always exists a path joining them along which the loss increases by a small quantity ϵ which goes to zero as the over-parametrization goes to infinity. However, the rate at which it goes to zero is exponential, that is an exponential number of neurons in the dimension of the input is needed. [Nguyen, 2019] relaxes this condition, and proves that as long as one of the hidden layers of a multilayer network has more neurons than the number of patterns then the sub-level sets are connected. One might wonder if this property of mode connectivity holds for all minima, or only for minima that are found with gradient based algorithms. Authors in [Kuditipudi et al., 2019] show that the latter is true, that is they construct two global minima of an overparametrized network that can be proved to be disconnected. Furthermore, they show that noise stability, that is the property that the network is unaffected if Gaussian noise is injected in the hidden layers, implies mode connectivity for realistic multilayer networks. Finally [Venturi et al., 2018b] showed that as long as the number of neurons for a two layer network is larger than twice the intrinsic dimension of the network, then the sub-level sets are connected. The results stated above are only valid for ReLU-type activations, while [Simsek et al., 2021] proves the connectedness of the global minima manifold for generic differentiable activations functions using sophisticated symmetry arguments.

In this chapter, I consider a specific model of a Neural Network, the Negative Perceptron, and analytically show that, at least for a certain type of solutions, the space of minimizers is Star-Shaped. Following the publication of this paper, a rigorous work came out showing that the star shaped property holds also for linear and two layer ReLU networks in the teacher-student scenario [Lin et al., 2024].

3.2 The Model

In this chapter, we study the *Negative Perceptron*, defined in section 2.3, and look at the connectivity of its solutions. The reason we choose this model is threefold:

1. We require a model with continuous weights, as we will be looking at linear paths between solutions which are only defined for continuous variables.

2. We require the space of solutions to be non-convex. Indeed if it were convex then all vectors along the linear path between solutions would be themselves solutions.

3. We require the model to be analytically treatable.

The simplest model satisfying all three requirements is indeed the negative perceptron. In the following we will consider a random dataset, given by Gaussian inputs $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_N)$ and random binary labels $P(y=1) = P(y=-1) = 1/2$ and look at the landscape of the averaged training error defined in equation 2.4 as a function of the margin κ with which the perceptron is sampled.

3.2.1 Organization of the Space of Solutions

As we did in section 2.3, let us define the $\beta \to \infty$ limit of the Gibbs distribution, which reduces to the uniform distribution over solutions on the N-dimensional hypersphere with a certain margin.

$$p_{\mathcal{D},\kappa}(\boldsymbol{W}) = \frac{1}{\mathcal{Z}_\kappa}\delta\left(\|\boldsymbol{W}\|^2 - N\right)\prod_{\mu=1}^{P}\Theta(\boldsymbol{W}\cdot\boldsymbol{x}^\mu - \kappa\sqrt{N}). \tag{3.4}$$

Here, we have used the same trick of changing the sign of every example with label $y^\mu = -1$, in order to consider only labels $+1$. As we have already explained, this distribution will be exponentially dominated by *typical* solutions. However if we fix the margin with which we measure the loss landscape, say κ_E, and vary the margin of the distribution $p_{\mathcal{D},\kappa}(\boldsymbol{W})$, for $\kappa \neq \kappa_E$ we will have access to the atypical solutions of the landscape.

The first question one might ask about these atypical solutions, is what are their statistical properties. For example, one might wonder what is the overlap between solutions sampled with different margins

$$p = \mathbb{E}_{\mathcal{D}}\Big\langle \frac{1}{N}\sum_i W_i^1 W_i^2 \Big\rangle_{\kappa_1,\kappa_2}. \tag{3.5}$$

Here with the notation $\left\langle \frac{1}{N}\sum_i W_i^1 W_i^2 \right\rangle_{\kappa_1,\kappa_2}$ we mean that $\boldsymbol{W}^1$ is sampled from $p_{\mathcal{D},\kappa_1}(\boldsymbol{W})$ and $\boldsymbol{W}^2$ is sampled from $p_{\mathcal{D},\kappa_2}(\boldsymbol{W})$. As this overlap will be needed for the main result of this chapter,

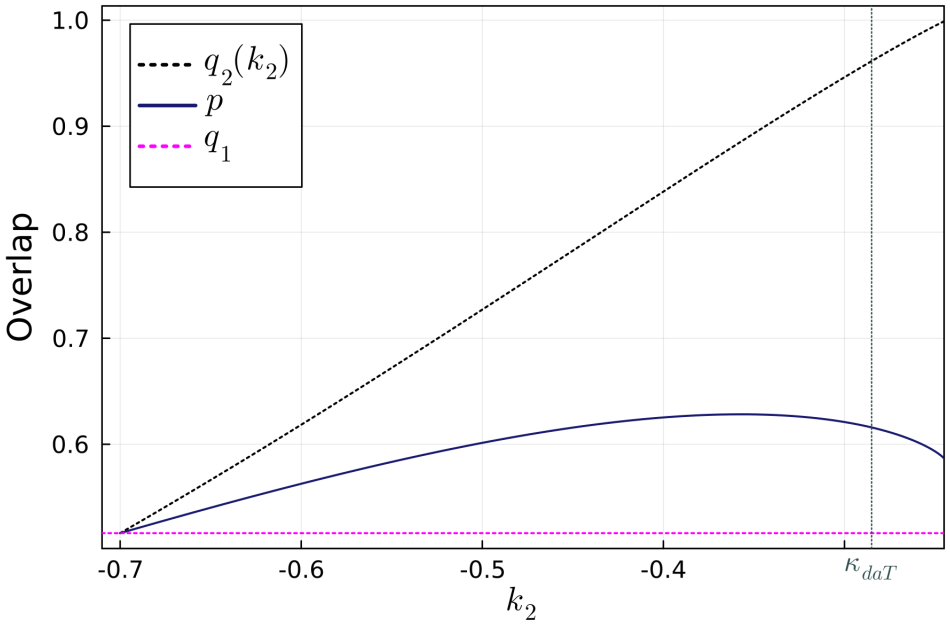

Figure 3.2: As a function of the margin k_2 from bottom to above: typical overlap q_1 between solutions having margin $k_1 = -0.7$ (dashed pink line), typical overlap p between two solutions having respectively margin k_1 and k_2 (full blue line) and typical overlap q_2 between two solutions with margin k_2 (dashed black line). Here $\alpha = 3$. This plot shows the nested overlap structure of the solutions $q_1 < p < q_2$. The dashed vertical line represents the value above which the RS ansatz is wrong.

let us see how one can calculate it using the replica method. To do so, we will look at the Gardner volumes of the problems defined with the two margins. These will give us access to the parameter p, which will appear as the value obtained at a saddle point, as was the case for q in our previous calculation. The details of the calculation are reported in Appendix 3.A. Here we limit ourselves to reporting that the value of p is found by solving the implicit equation

$$p = \alpha\sqrt{(1-q_1)(1-q_2)}\times \\ \times \int \mathcal{D}z_1 \mathcal{D}z_2 \, GH\left(\frac{k_1+\sqrt{q_1}z_1}{\sqrt{1-q_1}}\right) GH\left(\frac{k_2+\frac{p}{\sqrt{q_1}}z_1+\sqrt{q_2-\frac{p^2}{q_1}}z_2}{\sqrt{1-q_2}}\right), \tag{3.6}$$

where we have called $q_1 = \mathbb{E}_{\mathcal{D}}\left\langle \frac{1}{N}\sum_i W_i^1 W_i^2 \right\rangle_{k_1,k_1}$, $q_2 = \mathbb{E}_{\mathcal{D}}\left\langle \frac{1}{N}\sum_i W_i^1 W_i^2 \right\rangle_{k_2,k_2}$, $p = \mathbb{E}_{\mathcal{D}}\left\langle \frac{1}{N}\sum_i W_i^1 W_i^2 \right\rangle_{k_1,k_2}$ and defined $GH(x) \equiv \frac{G(x)}{H(x)}$. The numerical results obtained by solving this equation are shown in figure 3.2. Perhaps non surprisingly, we have that the inequalities $q_1 < p < q_2$ hold.

3.3 Landscape of the Training Error

As we have mentioned above, we will be looking at the landscape of the average training error, defined as

$$\epsilon_T = \mathbb{E}_{\mathcal{D}}\langle \epsilon_T(\boldsymbol{W};\mathcal{D})\rangle = \frac{1}{P}\sum_{\mu=1}^{P}\mathbb{E}_{\mathcal{D}}\langle \Theta(-\frac{1}{\sqrt{N}}\sum_i W_i x_i^\mu + \kappa_E)\rangle_{k_1} = \tag{3.7}$$

$$= \mathbb{E}_{\mathcal{D}}\langle \Theta(-\frac{1}{\sqrt{N}}\sum_i W_i x_i^1 + \kappa_E)\rangle_{k_1}. \tag{3.8}$$

Also here we have introduced a margin κ_E, which can be thought of as the margin of the problem, while the weight vector is sampled from the uniform distribution over solutions with margin k_1. In general the two margins need not be equal.

Averages such as these can also be calculated using the replica method, by using the simple identity

$$\mathcal{Z}^{-1} = \lim_{n\to 0}\mathcal{Z}^{n-1}. \tag{3.9}$$

Indeed we can write

$$\epsilon_T = \mathbb{E}_{\mathcal{D}}\frac{1}{\mathcal{Z}(k_1)}\int d\mu(\boldsymbol{W})\prod_{\mu=1}^{P}\Theta\Big(\frac{1}{\sqrt{N}}\sum_i W_i x_i^\mu - \kappa_E\Big)\Theta\Big(-\frac{1}{\sqrt{N}}\sum_i W_i x_i^1 + k_1\Big) = \tag{3.10}$$

$$= \lim_{n\to 0}\mathbb{E}_{\mathcal{D}}\int\prod_{a=1}^{n} d\mu(\boldsymbol{W}^a)\prod_{\mu a}\Theta\Big(\frac{1}{\sqrt{N}}\sum_i W_i^a x_i^\mu - \kappa_E\Big)\Theta\Big(-\frac{1}{\sqrt{N}}\sum_i W_i^1 x_i^1 + k_1\Big). \tag{3.11}$$

and proceed using the techniques outlined in the previous chapter. In this section however, we will calculate a slightly different quantity.

Consider y weight configurations $\boldsymbol{W}^r \in \mathbb{R}^N$ on the sphere $\|\boldsymbol{W}^r\|^2 = N$ and a vector of interpolation coefficients $\boldsymbol{\gamma} = (\gamma_1,\dots,\gamma_y)$ with $\gamma_r \geq 0$ and such that $\sum_{r=1}^{y}\gamma_r = 1$. We call $\boldsymbol{W}_{\boldsymbol{\gamma}}$ the interpolation

$$\boldsymbol{W}_{\boldsymbol{\gamma}} = \sqrt{N}\frac{\sum_r \gamma_r \boldsymbol{W}^r}{\|\sum_r \gamma_r \boldsymbol{W}^r\|}. \tag{3.12}$$

This can be interpreted as the convex combination between the y vectors $\boldsymbol{W}_1,\dots,\boldsymbol{W}_y$, which is then projected on the N-hypersphere such that it still has the correct norm. For $y = 2$, by varying γ we are going on the geodesic path on the N-dimensional sphere that connects $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$. For $y = 3$ we are considering a geodesic plane, which is a curved surface, and so on. The quantity which we will compute, and will allow us to probe the landscape of the negative

perceptron, is the training error of this interpolated vector $\boldsymbol{W}_\gamma$, when each $\boldsymbol{W}_r$ is sampled from the uniform distribution over solutions with margin k_r

$$E_\gamma = \lim_{N\to+\infty} \mathbb{E}_{\mathcal{D}} \left\langle \Theta\big(-\boldsymbol{W}_\gamma \cdot \boldsymbol{x}^1 + \kappa_E\sqrt{N}\big)\right\rangle_{k_1,\dots,k_y}. \tag{3.13}$$

Using the replica trick with calculations similar to the ones reported above we can write

$$E_\gamma = \lim_{N\to\infty}\lim_{n\to 0}\mathbb{E}_{\mathcal{D}} \int \prod_{a=1}^{n}\prod_{r=1}^{y} d\mu(\boldsymbol{W}^{ar}) \prod_{\mu ar}\Theta\left(\frac{1}{\sqrt{N}}\sum_i W_i^{ar}x_i^\mu - k_r\right)\times \tag{3.14}$$

$$\times\,\Theta\left(-\sum_i W^1_{\gamma,i}x^1_i + \kappa_E\right) =$$

$$= \lim_{N\to\infty}\lim_{n\to 0}\mathbb{E}_{\mathcal{D}} \int \prod_{ar} d\mu(\boldsymbol{W}^{ar}) \prod_{\mu ar}\frac{d\lambda_r^{\mu a} d\hat{\lambda}_r^{\mu a}}{2\pi}\prod_{\mu ar}\Theta(\lambda_r^{\mu a} - k_r)\times \tag{3.15}$$

$$\times \prod_{\mu ar}\Theta\left(-\frac{\sqrt{N}\sum_{r=1}^{y}\gamma_r\lambda_r^{11}}{\|\sum_r \gamma_r \boldsymbol{W}^{1r}\|} + \kappa_E\right) e^{i\sum_{\mu ar}\hat{\lambda}_r^{\mu a}\lambda_r^{\mu a} - i\sum_{\mu ari}\frac{1}{\sqrt{N}}\hat{\lambda}_r^{\mu a}W_{ri}^a x_i^\mu} = \tag{3.16}$$

$$= \lim_{N\to\infty}\lim_{n\to 0}\int \prod_{a<b,rs}\frac{dq_{rs}^{ab}\,d\hat{q}_{rs}^{ab}}{2\pi}\int\prod_{ar}\frac{d\hat{h}_r^a}{2\pi}\, e^{Nn\phi(q,\hat{q},\hat{h})} g(q). \tag{3.17}$$

Again, we have arrived at a form where we can apply Laplace's method, with the only difference that the exponential $e^{Nn\phi}$ is now multiplied by a function

$$g(q) = \frac{\int\prod_{ar}\frac{d\lambda_r^a d\hat{\lambda}_r^a}{2\pi}\prod_{ar}\Theta(\lambda_r^a - k_r)\,\Theta\left(-\frac{\sum_{r=1}^{y}\gamma_r\lambda_r^1}{\sqrt{\sum_{rs}\gamma_r\gamma_s q_{rs}^{a=b}}} + \kappa_E\right) e^{i\sum_{ar}\hat{\lambda}_r^a\lambda_r^a - \frac{1}{2}\sum_{abrs}q_{rs}^{ab}\hat{\lambda}_r^a\hat{\lambda}_s^b}}{\int\prod_{ar}\frac{d\lambda_r^a d\hat{\lambda}_r^a}{2\pi}\prod_{ar}\Theta(\lambda_r^a - k_r)\, e^{i\sum_{ar}\hat{\lambda}_r^a\lambda_r^a - \frac{1}{2}\sum_{abrs}q_{rs}^{ab}\hat{\lambda}_r^a\hat{\lambda}_s^b}}. \tag{3.18}$$

Calling $q_\star$ the matrix of overlaps at the saddle point for ϕ, Laplace's method gives

$$E_\gamma = \lim_{n\to 0}\, g(q_\star), \tag{3.19}$$

i.e. we simply need to calculate the function $g(q)$ in the parameters found from the saddle point.

The calculations for the analytical expression of this observable are reported in Appendix 3.B. Here we limit ourselves to noting that the result can be conveniently refactored as

$$E_\gamma = \Theta\left(f_\gamma(\kappa_E, \{k_r\}_r)\right) I_\gamma\left(\kappa_E, \{k_r\}_r\right), \tag{3.20}$$

where f_γ is a linear function of the margins

$$f_\gamma(\kappa_E, \{k_r\}_r) = \kappa_E\, c_\gamma - \sum_r \gamma_r k_r, \tag{3.21}$$

with $c_\gamma = \sqrt{\sum_{rs} q_{rs}\gamma_r\gamma_s}$, and I_γ is a non-negative function involving $2y$-dimensional Gaussian integrals. Since the expression can evaluate exactly to zero only if the argument of the Θ is negative, this rewriting allows us to analytically determine regions of the simplex at zero energy.

3.3.1 Solutions with Same Margin

To start off, we will consider the case where the vertices of the simplex are all sampled with identical margin $k_r = k$, with $\kappa_E \leq k < \kappa_{dAT}(\alpha)$. One finds that the energy on the projected $(y-1)$—simplex is always strictly greater than zero when $k = \kappa_E$, while extended regions around each vertex fall to zero energy for $k > \kappa_E$. For each value of y one can identify a *coalescence threshold*, $\kappa_y^\star(\kappa_E, \alpha)$, corresponding to the value of the margin above which the entire $(y-1)$—simplex lies at zero energy. In particular, we find the minimum margin $\kappa_2^\star(\kappa_E, \alpha)$ that ensures *linear mode connectivity*. These thresholds are displayed in figure 3.3 as a function of α for $\kappa_E = -0.5$, and satisfy $\kappa_2^\star < \kappa_3^\star < ... < \kappa_\infty^\star$. Above the last *coalescence threshold*, $\kappa_\infty^\star$, the projected convex hull of the entire ensemble of solutions with this margin lies at zero energy: we call this region the geodesically convex component of the manifold of solutions. Although the space of solutions is non-convex, we have identified a large subspace which is convex. By inspecting the distribution of stabilities across the zero-energy manifold, one finds that the geodesic paths encounter different solutions from those of the equilibrium description at the corresponding margin (details in appendix 3.C).

3.3.2 Solutions with Different Margin

We now focus on the connectivity of solutions with different margins. Specifically, we start by considering the geodesic path between a typical solution, $k_1 = \kappa_E$, and a higher margin solution, $k_2 > \kappa_E$. One can show that, for any (κ_E, α) below the dAT transition, there exists a threshold κ_{krn} such that, w.h.p., no energy barrier is encountered along the geodesic path between any solution with $k_2 \geq \kappa_{\text{krn}}$ and any other typical solution with margin $k_1 \geq \kappa_E$. These findings imply that the solution space is *star-shaped*: every typical solution with margin greater or equal than the margin of the problem κ_E is connected to any other such solution by passing through a *kernel* of solutions, i.e. a subset of solutions that are "visible" -- through

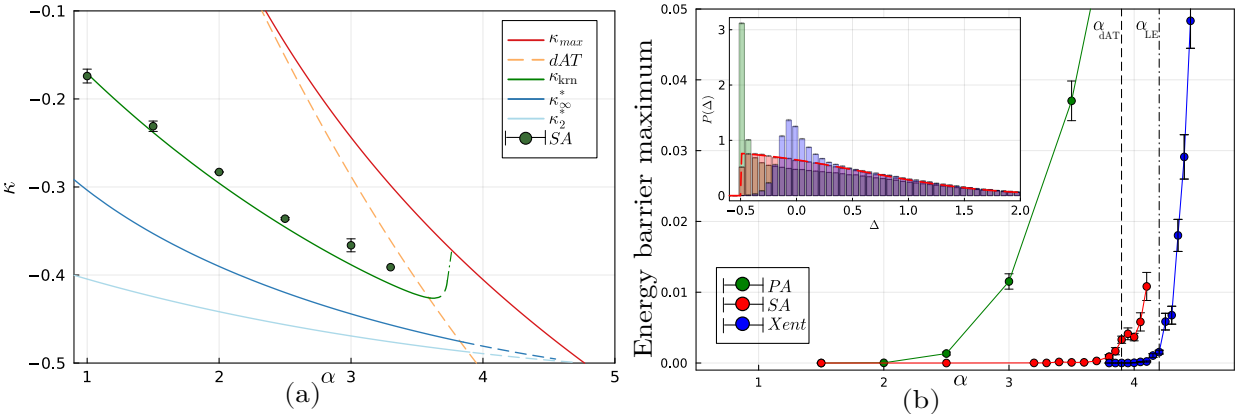

Figure 3.3: (a) **Coalescence threshold lines** at $\kappa_E = -0.5$ (blue and cyan), and the κ_{krn} threshold (green line) as a function of α. In orange the dAT transition line, delimiting the RS-stable region; in red the RS estimate of the κ_{max}, the line beyond which no solution each exists. Because it is beyond the dAT line and it has been calculated using an RS ansatz, it is just an approximation. Points are numerical estimates of the κ_{krn} transition. (b) **Maximum error along the geodesic path** ($y = 2$) connecting numerical solutions found with different algorithms (PA, SA and Xent) with the fBP max-margin solutions. Non-zero energies along the path indicate disconnection in the solution space. The vertical dashed lines denote the values of α_{dAT} and α_{LE} at $\kappa_E = -0.5$. The inset shows stability distributions for PA, SA and Xent at $\alpha = 2$ compared with the theoretical stability distribution of typical solutions (red dashed line).

geodesic paths -- from any typical point of the solution manifold. This geometry is schematically represented in figure 3.4.

The numerical value obtained for κ_{krn} with $\kappa_E = -0.5$ is shown in figure 3.3, together numerical estimates used by sampling the distribution given in equation (3.4).

3.4 Numerics

We compare the properties of solutions found with different solvers on instances of the negative perceptron with $\kappa_E = -0.5$. In particular, in Fig.3.3 (b), we characterize their geodesic connectivity to solutions located in the kernel region, as a function of α. Note that, because of the nested overlap structure, we expect the maximum-margin solutions of the problem to be located in the kernel. Therefore, for obtaining them we employ the focusing-BP (fBP) algorithm, which was shown in [Baldassi et al., 2023] to yield good proxies of the κ_{max} solutions.

Typical solutions instead are approximated by carefully applying Simulated Annealing (SA) on the square hinge loss with margin κ_E and are found to be in good agreement with the theory. Non-zero energy barriers with the fBP solutions seem to appear in close proximity of the dAT transition line, confirming the star-shapedness of typical solutions in the RS stable region.

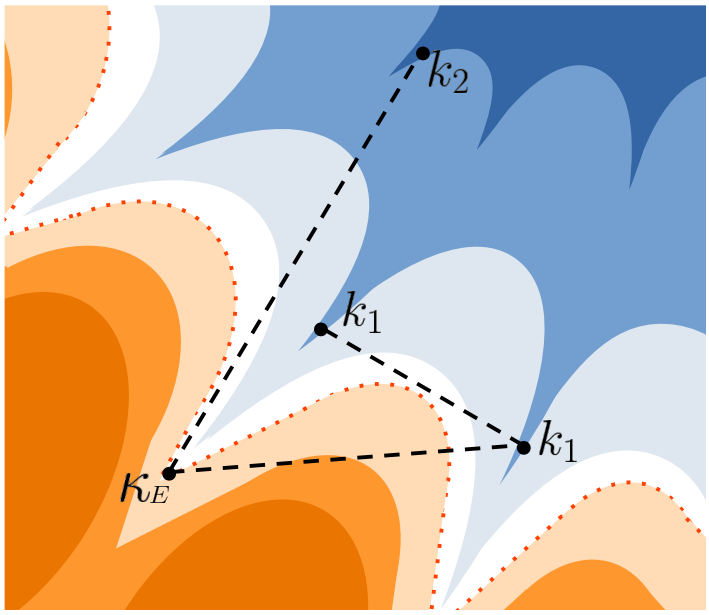

Figure 3.4: Sketch of the solution space of the negative perceptron in the RS phase. The red dotted line represents the border of the connected manifold of solutions for a given margin κ_E (white-blue region). In the orange regions, the configurations have non-zero energy. The solutions that satisfy margins larger than the one of the problem, $\kappa_E < k_1 < k_2$, are organized in a nested structure (darker shades of blue). When a typical solution with margin κ_E is geodesically connected with a solution with margin k_1, an energy barrier (a crossing of the orange region) is observed. However, the k_1 solutions belong to a *geodesically-convex* sub-manifold (the *geodesic path* falls within the white-blue region). Solutions with an even higher margin k_2, located in the *kernel*, are connected to almost any other solution.

SGD on the cross-entropy loss -- the most common optimization objective for this class of problems -- yields robust solutions [Baldassi et al., 2020] with higher average stability than typical, as shown in the inset in Fig.3.3 (b). The geodesic path between independent optimization trajectories, starting from random initialization, shows no energy barriers as soon as the zero-energy region is accessed, revealing an algorithmic bias towards the geodesically convex component of the solution manifold (details in the SM). The disconnection transition with the core solutions (Xent in Fig.3.3 (b)) is delayed with respect to SA, and seems to happen in close proximity of the α_{LE} transition characterized in [Baldassi et al., 2023].

Finally, we implement the classic Perceptron Algorithm (PA). When the learning rate is sufficiently small, this algorithm is able to sample solutions with a large mass of stabilities slightly above threshold (inset of Fig.3.3 (b)), and therefore less robust than typical ones. The disconnection with the core region of the solution manifold is in this case anticipated before the dAT line. Notice that this result is not incompatible with our predictions, since these solutions seem to be sub-dominant in the flat measure over solutions, and cannot be seen through an equilibrium analysis.

These numerical results are consistent with our theoretical picture of a star-shaped space of solutions in the over-parameterized regime, and reveal a progressive disconnection transition that affects different types of solutions according to their degree of robustness.

3.5 Conclusions

In the present chapter, we characterized the connectivity properties of a prototypical model of non-convex neural networks. The theoretical analysis unveiled the presence of a connected manifold of solutions organized in a star-shaped structure. Similar types of structures have been shown to appear in completely unrelated high-dimensional problems ([Zhang and Strogatz, 2021, Martiniani and Casiulis, 2023]). We conjecture that simple mode connectivity may be a universal property of non-convex optimization problems in the over-parameterized regime.

Having limited our analysis to a RS result, we are for the moment unable to say anything about the region beyond the dAT line. We will see in the next chapter how it is possible to do so, and will use these results to talk about the disconnection transition.

Appendix

3.A Calculation of the Overlap between Solutions with Different Margins

The Gardner volume for the combined system of two perceptrons with different margins is

$$\phi(k_1, k_2) = \frac{1}{N}\mathbb{E}_{\mathcal{D}} \log Z(k_1, k_2) \tag{3.22}$$

with $\mathcal{Z}(k_1, k_2) = \mathcal{Z}(k_1)\mathcal{Z}(k_2)$, being

$$\mathcal{Z}(k) \equiv \int d\mu(\boldsymbol{W}) \prod_{\mu=1}^{P} \Theta\left(\frac{1}{\sqrt{N}} \boldsymbol{W} \cdot \boldsymbol{x}^{\mu} - k\right) \equiv \int d\mu(\boldsymbol{W})\, \mathbb{X}(\boldsymbol{W}; k) \tag{3.23}$$

and $d\mu(\boldsymbol{W})$ again is the spherical measure over weights.

Again, thanks to the replica trick, the problem becomes that of computing the average of n independent copies of the system with the same disorder realization of the patterns $\boldsymbol{x}^{\mu}$

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \mathbb{E}_{\mathcal{D}} \int \prod_{ar} \mathrm{d}\mu(\boldsymbol{W}^{ar}) \prod_{\mu ar} \Theta\left(\frac{1}{\sqrt{N}} \sum_i W_i^{ar} x_i^{\mu} - k_r\right), \tag{3.24}$$

where $r = 1, 2$ runs over the two solutions with margin k_r, $a = 1, \ldots, n$ runs over the virtual replicas.

As before we extract the quantity $\lambda_r^{\mu a} = \frac{1}{\sqrt{N}} \sum_i W_i^{ar} x_i^{\mu}$ by using the integral representation of the delta function and this allows us to perform the disorder average on the patterns, obtaining

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \mathbb{E}_{\mathcal{D}} \int \prod_{\mu ar} \frac{d\lambda_r^{\mu a}\, d\hat{\lambda}_r^{\mu a}}{2\pi} \int \prod_{ar} d\mu(\boldsymbol{W}^{ar}) \times \tag{3.25}$$
$$\times \prod_{\mu ar} \Theta\left(\lambda_r^{\mu a} - k_r\right) e^{i\lambda_r^{\mu a}\hat{\lambda}_r^{\mu a}} \prod_{\mu i} e^{-\frac{i}{\sqrt{N}} x_i^{\mu} \sum_{ar} W_i^{ar}\hat{\lambda}_r^{\mu a}} =$$
$$= \int \prod_{\mu ar} \frac{d\lambda_r^{\mu a}\, d\hat{\lambda}_r^{\mu a}}{2\pi} \int \prod_{ar} d\mu(\boldsymbol{W}^{ar}) \times \tag{3.26}$$
$$\times \prod_{\mu ar} \Theta\left(\lambda_r^{\mu a} - k_r\right) e^{i\lambda_r^{\mu a}\hat{\lambda}_r^{\mu a}} \prod_{\mu} e^{-\frac{1}{2N}\sum_{ar,bs}\left(\sum_i W_i^{ar} W_i^{bs}\right)\hat{\lambda}_r^{\mu a}\hat{\lambda}_s^{\mu b}}.$$

By defining appropriate order parameters $q_{rs}^{ab} = \frac{1}{N}\sum_i W_i^{ar} W_i^{bs}$ we can conveniently rewrite the partition function as

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \int \prod_{a<b,rs} \frac{dq_{rs}^{ab}\, d\hat{q}_{rs}^{ab}}{2\pi} \int \prod_{ar} \frac{d\hat{h}_r^a}{2\pi}\, e^{Nn\phi(q,\hat{q},\hat{h})}, \tag{3.27}$$

where we have defined

$$\phi(q,\hat{q},\hat{h}) = -\frac{1}{2n}\sum_{a<b,rs} q_{rs}^{ab}\hat{q}_{rs}^{ab} + \frac{1}{n}\sum_{ar}\hat{h}_r^a + \frac{1}{n}G_S + \frac{\alpha}{n}G_E, \tag{3.28}$$

$$G_S = \ln \int \prod_{ar} dW^{ar}\, e^{\frac{1}{2}\sum_{a<b,rs} W^{ar}W^{bs}\hat{q}_{rs}^{ab} - \sum_{ar}(W^{ar})^2\hat{h}_r^a}, \tag{3.29}$$

$$G_E = \ln \int \prod_{ar} \frac{d\lambda_r^a\, d\hat{\lambda}_r^a}{2\pi} \prod_{ar} e^{i\lambda_r^a\hat{\lambda}_r^a}\, \Theta\left(\lambda_r^a - k_r\right) e^{-\frac{1}{2}\sum_{a<b,rs} q_{rs}^{ab}\hat{\lambda}_r^a\hat{\lambda}_s^b}. \tag{3.30}$$

Replica-Symmetric Ansatz Again let us assume a Replica-Symmetric (RS) ansatz on the order parameters as follows

$$q_{rr}^{ab} = \delta_{ab} + q_r(1-\delta_{ab}) \quad \forall r = s, \tag{3.31}$$
$$q_{rs}^{ab} = R\,\delta_{ab} + p\,(1-\delta_{ab}) \quad \forall r \neq s. \tag{3.32}$$

where q_r represents the typical overlap of a solution with κ_r margin while p and$R$ are respectively the overlaps between the two reference solutions within the same replicas or not. There is no reason in principle to assume p and R to be different since the system is decoupled, however since we are looking for the value of p which would not appear in the calculation if $p = R$, we use the trick of assuming them different and later perform the limit $R \to p$. As

we have already stated, this ansatz will give the correct result only in a certain region of the parameters α and κ.

The calculation follows steps similar to the ones we have seen above, and in the end the free entropy can be written as

$$\phi = \frac{1}{2}\sum_r q_r \hat{q}_r + \sum_r \hat{h}_r + p\,\hat{p} - R\hat{R} + \mathcal{G}_S + \alpha \mathcal{G}_E, \tag{3.33}$$

where

$$\mathcal{G}_S = \lim_{n\to 0}\frac{G_S}{n} = \frac{1}{2}\ln\frac{4\pi^2}{(2\hat{h}_1+\hat{q}_1)(2\hat{h}_2+\hat{q}_2)-(\hat{p}-\hat{R})^2} + \frac{\hat{h}_2\hat{q}_1+(\hat{h}_1+\hat{q}_1)\hat{q}_2+\hat{p}\,(\hat{R}-\hat{p})}{(2\hat{h}_1+\hat{q}_1)(2\hat{h}_2+\hat{q}_2)-(\hat{p}-\hat{R})^2}, \tag{3.34}$$

$$\begin{aligned}\mathcal{G}_E = \lim_{n\to 0}\frac{G_E}{n} = \int \mathcal{D}z_1\,\mathcal{D}z_2 \ln \int \mathcal{D}x\, H\left(\frac{k_1+\sqrt{R-p}\,x+\sqrt{q_1}\,z_1}{\sqrt{1-q_1-R+p}}\right)\times \\ \times H\left(\frac{k_2+\sqrt{R-p}\,x+\frac{p}{\sqrt{q_1}}z_1+\sqrt{q_2-\frac{p^2}{q_1}}\,z_2}{\sqrt{1-q_2-R+p}}\right),\end{aligned} \tag{3.35}$$

with $H(x)$ defined in equation (2.44). By differentiating (3.33) with respect to the order parameters $q_1, q_2, p, R, \hat{q}_1, \hat{q}_2, \hat{h}_1, \hat{h}_2, \hat{p}, \hat{R}$, the saddle point equations take the form of

$$\begin{aligned} &q_1 = -2\frac{\partial \mathcal{G}_S}{\partial \hat{q}_1}, \quad q_2 = -2\frac{\partial \mathcal{G}_S}{\partial \hat{q}_2}, \quad p = -\frac{\partial \mathcal{G}_S}{\partial \hat{p}}, \quad R = \frac{\partial \mathcal{G}_S}{\partial \hat{R}}, \\ &\hat{q}_1 = -2\alpha\frac{\partial \mathcal{G}_E}{\partial q_1}, \quad \hat{q}_2 = -2\alpha\frac{\partial \mathcal{G}_E}{\partial q_2}, \quad \hat{p} = -\alpha\frac{\partial \mathcal{G}_E}{\partial p}, \quad \hat{R} = \alpha\frac{\partial \mathcal{G}_E}{\partial R}, \quad 1+\frac{\partial \mathcal{G}_S}{\partial \hat{h}_1} = 0, \quad 1+\frac{\partial \mathcal{G}_S}{\partial \hat{h}_2} = 0. \end{aligned} \tag{3.36}$$

The conjugated parameters can be solved as functions of the non-conjugated ones. Moreover, if the two solutions sampled with margin κ_1 and κ_2 are free to arrange in the most probable position, we have $p = R$ as already mentioned. This simplifies the conjugated parameters to

$$\begin{aligned} &\hat{q}_1 = \frac{q_1}{(1-q_1)^2}, \qquad \hat{q}_2 = \frac{q_2}{(1-q_2)^2}, \qquad \hat{p} = \frac{p}{(1-q_1)(1-q_2)}, \\ &\hat{R} = \frac{p}{(1-q_1)(1-q_2)}, \quad \hat{h}_1 = \frac{1-2q_1}{2(1-q_1)^2}, \quad \hat{h}_2 = \frac{1-2q_2}{2(1-q_2)^2}. \end{aligned} \tag{3.37}$$

By looking at the expression of the entropy, it is also easy to see that, since the two solutions are sampled independently, in the limit $R \to p$ the expression reduces the sum of the two entropies of single configuration.

$R \to p$ limit and simplification of the saddle point equations. To actually compute the value of p we resort to the saddle point equation $\hat{R} = \alpha \frac{\partial G_E}{\partial R}$ in the limit $R \to p$. Defining

$$\mathcal{Z} \equiv \int \mathcal{D}x\, H\left(\frac{k_1 + \sqrt{R-p}\, x + \sqrt{q_1}\, z_1}{\sqrt{1-q_1-R+p}}\right) \times \tag{3.38}$$
$$\times H\left(\frac{k_2 + \sqrt{R-p}\, x + \frac{p}{\sqrt{q_1}} z_1 + \sqrt{q_2 - \frac{p^2}{q_1}}\, z_2}{\sqrt{1-q_2-R+p}}\right) =$$
$$= \int \mathcal{D}x\, H(a+bx) H(c+dx), \tag{3.39}$$

and integrating by parts once the limit has been replaced in order to remove divergences, we have

$$\frac{\partial G_E}{\partial R} = \frac{1}{\sqrt{(1-q_1)(1-q_2)}} \int \mathcal{D}z_1 \mathcal{D}z_2 \frac{G(a)G(c)}{\mathcal{Z}}, \tag{3.40}$$

since when $R \to p$, then $b, d \to 0$ as

$$\lim_{R\to p} \frac{b}{\sqrt{R-p}} = \frac{1}{\sqrt{1-q_1}}, \tag{3.41}$$
$$\lim_{R\to p} \frac{d}{\sqrt{R-p}} = \frac{1}{\sqrt{1-q_2}}. \tag{3.42}$$

Using equation (3.37) for $\hat{R}$ we therefore get p by solving the following implicit equation

$$p = \alpha \sqrt{(1-q_1)(1-q_2)} \int \mathcal{D}z_1 \mathcal{D}z_2\, GH(a)\, GH(c) = \tag{3.43}$$
$$= \alpha \sqrt{(1-q_1)(1-q_2)} \int \mathcal{D}z_1 \mathcal{D}z_2\, GH\left(\frac{k_1 + \sqrt{q_1}\, z_1}{\sqrt{1-q_1}}\right) \times \tag{3.44}$$
$$\times GH\left(\frac{k_2 + \frac{p}{\sqrt{q_1}} z_1 + \sqrt{q_2 - \frac{p^2}{q_1}}\, z_2}{\sqrt{1-q_2}}\right),$$

where $GH(x) \equiv \frac{G(x)}{H(x)}$.

3.B Calculation of the Training Error along the Geodesic Simplex

In this appendix we will go through the calculations that are necessary to derive the expressions that are reported in section 3.3. As we have seen, the integral we are after is

$$E_\gamma = \lim_{N\to\infty}\lim_{n\to 0}\int \prod_{a<b,rs}\frac{dq_{rs}^{ab}\,d\hat{q}_{rs}^{ab}}{2\pi}\int\prod_{ar}\frac{d\hat{h}_r^a}{2\pi}\,e^{Nn\phi(q,\hat{q},\hat{h})}g(q), \tag{3.45}$$

where

$$g(q)=\frac{\int\prod_{ar}\frac{d\lambda_r^a d\hat{\lambda}_r^a}{2\pi}\prod_{ar}\Theta(\lambda_r^a-k_r)\,\Theta\left(-\frac{\sum_{r=1}^{y}\gamma_r\lambda_r^1}{\sqrt{\sum_{rs}\gamma_r\gamma_s q_{rs}^{a=b}}}+\kappa_E\right)e^{i\sum_{ar}\hat{\lambda}_r^a\lambda_r^a-\frac{1}{2}\sum_{abrs}q_{rs}^{ab}\hat{\lambda}_r^a\hat{\lambda}_s^b}}{\int\prod_{ar}\frac{d\lambda_r^a d\hat{\lambda}_r^a}{2\pi}\prod_{ar}\Theta(\lambda_r^a-k_r)\,e^{i\sum_{ar}\hat{\lambda}_r^a\lambda_r^a-\frac{1}{2}\sum_{abrs}q_{rs}^{ab}\hat{\lambda}_r^a\hat{\lambda}_s^b}}. \tag{3.46}$$

Let us start by making an ansatz on the overlap matrix q_{rs}^{ab}.

3.B.1 General structure of the result in the RS ansatz

In the RS ansatz the tensor q_{rs}^{ab} is identified by two matrices $q_{rs}^{a=b}$ and $q_{rs}^{a\neq b}$. They have the following structure

$$q_{rs}^{aa}=\delta_{rs}+(1-\delta_{rs})\,p_{rs}, \tag{3.47}$$

$$q_{rs}^{a\neq b}\equiv t_{rs}=q_r\delta_{rs}+(1-\delta_{rs})\,p_{rs}. \tag{3.48}$$

We have denoted by q_r, $r\in[y]$ the typical overlap between two solutions having the same margin k_r, $r\in[y]$ and by p_{rs} the typical overlap between a solution having margin k_r and another one having margin k_s, with $r,s\in[y]$ and $r\neq s$. The computation can be carried out by noticing that the term in (3.46) that depends on q_{rs}^{ab} can be written as

$$-\frac{1}{2}\sum_{abrs}q_{rs}^{ab}\hat{\lambda}_r^a\hat{\lambda}_s^b=-\frac{1}{2}\sum_r(1-q_r)\sum_a\left(\hat{\lambda}_r^a\right)^2-\frac{1}{2}\sum_{rs}t_{rs}\sum_{ab}\hat{\lambda}_r^a\hat{\lambda}_s^b= \tag{3.49}$$

$$=-\frac{1}{2}\sum_r(1-q_r)\sum_a\left(\hat{\lambda}_r^a\right)^2-\frac{1}{2}\sum_r\left(\sum_{as}\mathcal{T}_{rs}\hat{\lambda}_s^a\right)^2, \tag{3.50}$$

where $\mathcal{T}_{rs}$ is (r,s) the element of the square root of the matrix t_{rs}. The computation proceeds in a standard way by using a Hubbard-Stratonovich transformation and integrating over $\hat{\lambda}_r^a$. Since the denominator tends to 1 when $n \to 0$ we get

$$E_\gamma = \lim_{n\to 0} \int \prod_r \mathcal{D}x_r \int \prod_{ar} \frac{d\lambda_r^a}{\sqrt{2\pi(1-q_r)}} \times \tag{3.51}$$
$$\times \prod_{ar} \Theta(\lambda_r^a - k_r)\, \Theta\left(\kappa_E c_\gamma - \sum_{r=1}^{y} \gamma_r \lambda_r^1\right) e^{-\frac{1}{2}\sum_{ar} \frac{(\lambda_r^a - \sum_s \mathcal{T}_{rs} x_s)^2}{1-q_r}} =$$
$$= \int \prod_r \mathcal{D}x_r \times \tag{3.52}$$
$$\times \frac{\int \prod_r \mathcal{D}\lambda_r \Theta(\sqrt{1-q_r}\lambda_r + \sum_s \mathcal{T}_{rs} x_s - k_r)\, \Theta\left(\kappa_E c_\gamma - \sum_{r=1}^{y} \gamma_r \left(\sqrt{1-q_r}\lambda_r + \sum_s \mathcal{T}_{rs} x_s\right)\right)}{\prod_r H\left(\frac{k_r - \sum_s \mathcal{T}_{rs} x_s}{\sqrt{1-q_r}}\right)},$$

where we have defined the quantity

$$c_\gamma \equiv \sqrt{\sum_{rs} \gamma_r \gamma_s q_{rs}^{a=b}} = \sqrt{\sum_r \gamma_r^2 + \sum_{r\neq s} p_{rs} \gamma_r \gamma_s}\,. \tag{3.53}$$

Next we can get rid of the theta functions expressing them as integration boundaries. Starting from the

$$E_\gamma = \int \prod_r \mathcal{D}x_r \frac{\int \prod_r \mathcal{D}\lambda_r \prod_r \Theta(\lambda_r - A_r)\, \Theta\left(\frac{\kappa_E c_\gamma - \sum_{rs} \gamma_r \mathcal{T}_{rs} x_s - \sum_{r=1}^{y-1} \sqrt{1-q_r}\gamma_r \lambda_r}{\gamma_y \sqrt{1-q_y}} - \lambda_y\right)}{\prod_r H(A_r)} \tag{3.54}$$
$$= \int \prod_r \mathcal{D}x_r \frac{\int \prod_{r=1}^{y-1} \mathcal{D}\lambda_r \prod_{r=1}^{y-1} \Theta(\lambda_r - A_r)\, \Theta\left(B_y - A_y\right) \int_{A_y}^{B_y} \mathcal{D}\lambda_y}{\prod_r H(A_r)} \tag{3.55}$$
$$= \int \prod_r \mathcal{D}x_r \frac{\int \prod_{r=1}^{y-2} \mathcal{D}\lambda_r \prod_{r=1}^{y-2} \Theta(\lambda_r - A_r)\, \Theta\left(B_{y-1} - A_{y-1}\right) \int_{A_{y-1}}^{B_{y-1}} \mathcal{D}\lambda_{y-1} \int_{A_y}^{B_y} \mathcal{D}\lambda_y}{\prod_r H(A_r)} \tag{3.56}$$
$$= \int \prod_r \mathcal{D}x_r\, \Theta(B_1 - A_1) \frac{\int_{A_1}^{B_1} \mathcal{D}\lambda_1 \cdots \int_{A_y}^{B_y} \mathcal{D}\lambda_y}{\prod_r H(A_r)}, \tag{3.57}$$

where we have defined the following quantities

$$A_r \equiv \frac{k_r - \sum_s \mathcal{T}_{rs} x_s}{\sqrt{1-q_r}}, \qquad r \in [y] \tag{3.58}$$

$$B_l \equiv \frac{\kappa_E c_\gamma - \sum_{r=l+1}^{y} \gamma_r k_r - \sum_{r=1}^{l} \gamma_r \sum_s \mathcal{T}_{rs} x_s - \sum_{r=1}^{l-1} \sqrt{1-q_r}\gamma_r \lambda_r}{\gamma_l \sqrt{1-q_l}} \qquad l \in [y]. \tag{3.59}$$

Notably the term $\Theta(B_1 - A_1) = \Theta\left(\kappa_E c_\gamma - \sum_r \gamma_r k_r\right)$ does not depend on any of the x_r, $r \in [y]$. The innermost integral can be performed analytically, leading to the final expression

$$E_\gamma = \Theta\left(\kappa_E c_\gamma - \sum_r \gamma_r k_r\right) \int \prod_r \mathcal{D}z_r \frac{\int_{A_1}^{B^1} \mathcal{D}\lambda_1 \dots \int_{A_{y-1}}^{B^{y-1}} \mathcal{D}\lambda_{y-1} \left(H(A_y) - H(B_y)\right)}{\prod_r H(A_r)}. \tag{3.60}$$

Therefore, the quantities f_γ and I_γ defined in the main text are

$$f_\gamma(\kappa_E, \{k_r\}_r) \equiv \kappa_E c_\gamma - \sum_r \gamma_r k_r, \tag{3.61}$$

$$I_\gamma\left(\kappa_E, \{k_r\}_r\right) \equiv \int \prod_r \mathcal{D}z_r \frac{\int_{A_1}^{B^1} \mathcal{D}\lambda_1 \dots \int_{A_{y-1}}^{B^{y-1}} \mathcal{D}\lambda_{y-1} \left(H(A_y) - H(B_y)\right)}{\prod_r H(A_r)}. \tag{3.62}$$

3.B.2 Sampling solutions with the same margin k

We consider the case in which all the vertices of the simplex have the same margin, i.e. $k_r = k \geq \kappa_E$, for $r \in [y]$. In this case the matrix t_{rs} defined in equation (3.48) and its square root $\mathcal{T}_{rs}$ contain all equal elements

$$t_{rs} = q, \tag{3.63}$$

$$\mathcal{T}_{rs} = \sqrt{\frac{q}{y}}, \tag{3.64}$$

where we remind that q is the typical overlap between solutions having margin k. In equation (3.60) one can integrate all x_r except one leading to

$$E_\gamma = \Theta\left(\kappa_E c_\gamma - k\right) \int \mathcal{D}z \frac{\int_{A}^{B_1} \mathcal{D}\lambda_1 \dots \int_{A}^{B_{y-1}} \mathcal{D}\lambda_{y-1} \left(H(A) - H(B_y)\right)}{H^y(A)}, \tag{3.65}$$

where

$$A \equiv \frac{\mathcal{T}_{rs} - \sqrt{q}x}{\sqrt{1-q}}, \tag{3.66}$$

$$B_l \equiv \frac{\mathcal{T}_{rs} c_\gamma - \mathcal{T}_{rs} \sum_{r=l+1}^{y} \gamma_r - \sqrt{q}x \sum_{r=1}^{l} \gamma_r - \sqrt{1-q} \sum_{r=1}^{l-1} \gamma_r \lambda_r}{\gamma_l \sqrt{1-q}}, \qquad l \in [y] \tag{3.67}$$

$$c_\gamma = \sqrt{(1-q)\sum_r \gamma_r^2 + q}. \tag{3.68}$$

Notice that since $0 \leq \gamma_r \leq 1$, we have $\sum_r \gamma_r^2 \leq \sum_r \gamma_r = 1$ and consequently $c_\gamma^2 = (1-q)\sum_r \gamma_r^2 + q \leq 1$. Strict equality holds only on the vertices, while c_γ attains its minimum value on the barycenter $\gamma_r \equiv \frac{1}{y}$, for which

$$c^2_{\text{barycenter}} = (1-q)\frac{1}{y} + q. \tag{3.69}$$

Case $k = \kappa_E$ Here we start by analyzing the simplest possible case, i.e. when $k = \kappa_E$. Since $c_\gamma \leq 1$ the argument of the theta function $\kappa_E c_\gamma - \kappa_E$ is always positive when $\kappa_E < 0$ and therefore every point on the simplex has non-vanishing energy. When $\kappa_E > 0$ instead, the argument of the theta function is always negative; when $\kappa_E = 0$, the extremes of integration $A = B_1$; in both cases this means that $E_\gamma = 0$ on the linear interpolation between two solutions. This is consistent with the fact that for $\kappa_E \geq 0$ the space of solutions is convex.

In the left panel panel of Fig. 3.5 we report the training error on the 2-simplex (i.e. a triangle) with solutions having $\kappa_E = -0.5$ placed at its vertices.

Case $k > \kappa_E$ We extend the analysis of the previous subsection to the case $k > \kappa_E$. The condition $\Theta(\kappa_E c_\gamma - k)$ allows us to distinguish between three regimes:

- for $0 \leq \kappa_E \leq k$ the whole convex envelope has zero training error (as said before, the space of solutions is convex)
- for $\kappa_E < k < \kappa_y^\star(\alpha, \kappa_E) < 0$ an extended region in the proximity of each corner has zero training error. In the case $y = 2$ we can extract the extension of this region: the training error is zero if $\gamma \in [0, \gamma_L]$ and$\gamma \in [\gamma_R, 1]$; by symmetry

$$\frac{1}{2} - \gamma_L = \gamma_R - \frac{1}{2} \text{ and } \gamma_{L,R} = \frac{1}{2} \pm \frac{1}{2}\sqrt{1 - 2\left(\frac{1 - \left(\frac{k}{\kappa_E}\right)^2}{1-q}\right)}. \tag{3.70}$$

- If the margin is large enough, (i.e. $k > \kappa_y^\star(\alpha, \kappa_E)$) we can find the whole simplex is at zero training energy, meaning that $\kappa_E c_\gamma - k < 0$ for any value of $\boldsymbol{\gamma}$. we can find interpolations γ with zero training error if the following relation holds: $\kappa_E \leq \frac{k}{c_\gamma}$. In the $y = 2$ case it is easy to find the value $\kappa_2^\star$ by imposing that $\gamma_L = \gamma_R$; we find that $\kappa_2^\star$ is found by searching the value of k that satisfies the following implicit equation

$$1 - \left(\frac{k}{\kappa_E}\right)^2 = \frac{1 - q(k)}{2}, \tag{3.71}$$

  where we have explicited for convenience the dependence of the overlap q on k.
  This argument can be easily extended to generic y by reminding that the minimum value of c_γ is attained on the barycenter $\gamma_r = \frac{1}{y}$; when we increase the value of k by symmetry this is the last point to have non-zero training error. This tells us that if the inequality $\kappa_E c_\gamma - k < 0$ is valid for the barycenter, then the whole simplex will have zero training error. Reminding that κ_E is negative, we have that the condition for which the whole simplex has zero energy is $k > \kappa_E c_{\text{barycenter}} = \kappa_E \sqrt{(1-q)\frac{1}{y} + q} \equiv \kappa_y^\star$ which is consistent with the result for $y = 2$. Since the overlap q increases as the margin k increases, then it easy to see that the thresholds are ordered in y $\kappa_2^\star < \kappa_3^\star < \cdots < \kappa_\infty^\star = \kappa_E \sqrt{q}$.

In the right panel of Fig. 3.5 we report the training error on the 2-simplex (i.e. a triangle) with solutions having $\kappa_2^* < k < \kappa_3^*$ placed at its vertices. In this case, since the solutions are extracted with $k > \kappa_2^*$, the geodesic paths between them (i.e. the edges of the triangle) is at zero energy by definition; however, since $k < \kappa_3^\star$, the 2-simplex presents configurations at non-zero training error near and at its barycenter.

Case $k < \kappa_E$ For $k < \kappa_E$ the manifold presents always configurations with a finite error.

3.B.3 Sampling Solutions with different Margins

In order to further explore the structure of the space of solutions we focus on the case of independent sampling of y solutions with two different margins. For simplicity we consider the first $y - 1$ vertices to be sampled all with margin k_1 and the last vertex with margin k_2. In this case the matrix t_{rs} is of the type

$$q^{a \neq b} = t = \begin{pmatrix} q_1 & \cdots & q_1 & p \\ \vdots & \ddots & \vdots & \vdots \\ q_1 & \cdots & q_1 & p \\ p & \cdots & p & q_2 \end{pmatrix}, \tag{3.72}$$

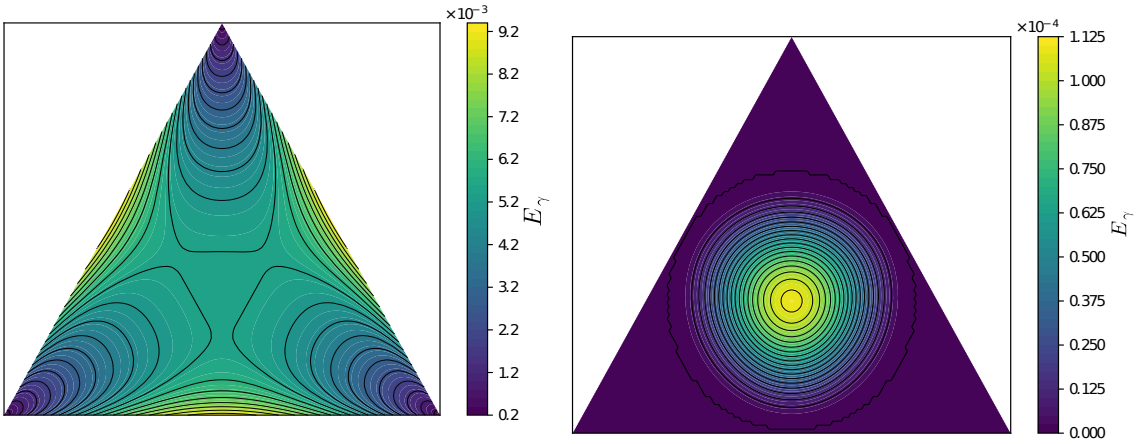

Figure 3.5: Training Error on the interpolation of $y = 3$ soluitons. (Left Panel) All three solution are sampled with $k = \kappa_E$, and are thus typical solutions. (Right Panel) All three solutions are sampled with $\kappa_2^* < k < \kappa_3^*$. As expected, the linear barriers between the solutions are zero, while the plane connecting all three of them shows a bump.

where q_1 and q_y are the typical overlaps between two solutions having respectively margin k_1, k_2 and p is the typical overlap between a solution having margin k_1 and another one with k_2. The square root of t_{rs} has the same form of t_{rs}

$$\mathcal{T} = \begin{pmatrix} \overline{q}_1 & \dots & \overline{q}_1 & \overline{p} \\ \vdots & \ddots & \vdots & \vdots \\ \overline{q}_1 & \dots & \overline{q}_1 & \overline{p} \\ \overline{p} & \dots & \overline{p} & \overline{q}_2 \end{pmatrix}, \tag{3.73}$$

where

$$\overline{q}_1 = \frac{(y-1)q_1 + \sqrt{(y-1)(q_1q_2 - p^2)}}{(y-1)\mathcal{Z}}, \tag{3.74}$$

$$\overline{q}_2 = \frac{q_2 + \sqrt{(y-1)(q_1q_2 - p^2)}}{\mathcal{Z}}, \tag{3.75}$$

$$\overline{p} = \frac{p}{\mathcal{Z}}, \tag{3.76}$$

$$\mathcal{Z} = q_2 + (y-1)q_1 - 2\sqrt{(y-1)(q_1q_2 - p^2)}. \tag{3.77}$$

The term $\sum_s \mathcal{T}_{rs}x_s$ in can be written as

$$\sum_s \mathcal{T}_{rs}x_s = \begin{cases} \overline{q}_1 \sum_{s=1}^{y-1} x_s + \overline{p}x_y, & \text{if } r \in [y-1] \\ \overline{p} \sum_{s=1}^{y-1} x_s + \overline{q}_2 x_y & \text{if } r = y. \end{cases} \tag{3.78}$$

By performing some rotations over the x_r variables and using the identities

$$(y-1)\overline{q}_1^2+\overline{p}^2=q_1, \tag{3.79}$$

$$\overline{p}((y-1)\overline{q}_1+\overline{q}_2)=p, \tag{3.80}$$

$$\sqrt{y-1}\left(\overline{q}_1\overline{q}_2-\overline{p}^2\right)=\sqrt{q_1q_2-p^2}, \tag{3.81}$$

it is possible to perform $y-2$ of the y integrals over x_r. The final result is

$$E_\gamma=\Theta\left(\kappa_E c_\gamma-k_1\sum_{r=1}^{y-1}\gamma_r-k_2\gamma_y\right)\times \tag{3.82}$$
$$\times\int\prod_r \mathcal{D}x_1\,\mathcal{D}x_2\,\frac{\int_{A_1}^{B^1}\mathcal{D}\lambda_1\ldots\int_{A_{y-1}}^{B^{y-1}}\mathcal{D}\lambda_{y-1}\left(H(A_y)-H(B_y)\right)}{\prod_r H(A_r)},$$

where

$$A_r=\begin{cases}\frac{k_r-\sqrt{q_1}x_1}{\sqrt{1-q}}, & r\in[y-1]\\ \frac{k_r-\frac{p}{\sqrt{q_1}}x_1-\sqrt{q_2-\frac{p^2}{q_1}}x_2}{\sqrt{1-q_2}}, & r=y\end{cases} \tag{3.83}$$

$$B_l\equiv\begin{cases}\frac{\kappa_E c_\gamma-k_1\sum_{r=l+1}^{y-1}\gamma_r-k_2\gamma_y-\sqrt{q}x_1\sum_{r=1}^{l}\gamma_r-\sqrt{1-q}\sum_{r=1}^{l-1}\gamma_r\lambda_r}{\gamma_l\sqrt{1-q_l}}, & l\in[y-1]\\ \frac{\kappa_E c_\gamma-\sqrt{q}x_1\sum_{r=1}^{y-1}\gamma_r-\gamma_y\left[\frac{p}{\sqrt{q_1}}x_2+\sqrt{q_2-\frac{p^2}{q_1}}x_2\right]-\sqrt{1-q_1}\sum_{r=1}^{y-1}\gamma_r\lambda_r}{\gamma_y\sqrt{1-q_2}}, & l=y\end{cases} \tag{3.84}$$

and $c_\gamma^2=\sum_r\gamma_r^2+2\gamma_y p\sum_{r=1}^{y-1}\gamma_r+2q_1\sum_{r<s<y}\gamma_r\gamma_s$.

The argument of the theta function in can be inspected analytically in simple cases. We start from the case $y=2$. In this case $c_\gamma=\gamma^2+(1-\gamma)^2+2p\gamma(1-\gamma)$; the training error on a point on a geodesic, which is parameterized by identified by $\gamma\in[0,1]$, vanishes if $\kappa_E^2\left[\gamma^2+(1-\gamma)^2+2p\gamma(1-\gamma)\right]>k_1\gamma^2+k_2(1-\gamma)^2+2k_1k_2\gamma(1-\gamma)$. If $k_1>\kappa_E$ and $k_2>\kappa_E$ an extended region between $\gamma\in[0,\gamma_L]$ and $\gamma\in[\gamma_R,1]$ have zero training error with

$$\gamma_{L,R}=\frac{k_2(k_2-k_1)-\kappa_E^2(1-p)\pm\kappa_E\sqrt{k_1^2-2k_1k_2p+k_2^2-\kappa_E^2(1-p^2)}}{(k_1-k_2)^2-2\kappa_E^2(1-p)}. \tag{3.85}$$

Imposing $\gamma_L=\gamma_r$ we find a condition for k_2 that we call $\kappa_{\rm krn}(k_1)$

$$\kappa_{\rm krn}(k_1)=k_1p-\sqrt{(1-p^2)(\kappa_E^2-k_1^2)}, \tag{3.86}$$

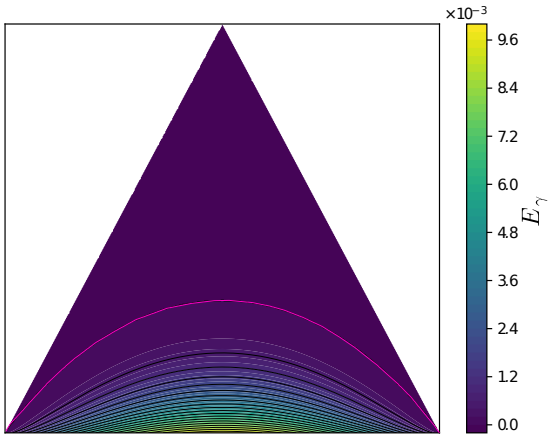

Figure 3.6: Training error on the simplex spanned by $y = 3$ solutions with $\alpha = 1.0$ and $\kappa_E = -0.5$; the two bottom vertices are two typical solutions to the problem, i.e. have margin margin $k_1 = \kappa_E$ whereas the top vertex is sampled with $k_2 = -0.1 > \kappa_{\text{krn}}$. The level curve in purple delimits the zero-energy region on the manifold.

that represents the minimum margin that should be imposed on $\boldsymbol{W}^2$ given the margin k_1 on $\boldsymbol{W}^1$ such that the two solutions are geodesically connected. Notice that the solution with the plus in front of the square root in the previous equation gives values of $\gamma > 1$ so it should be discarded. Since when k_1 increases p increases as well, $\kappa_{\text{krn}}(k_1)$ is a decreasing function of k_1. The maximum value of κ_{krn} is obtained therefore when $k_1 = \kappa_E$, for which we find $\kappa_{\text{krn}} = k_1 p$ which was plotted in the main text. Therefore if the margin of a solution is larger than k_{krn} then we can geodesically connect with every other typical solution with a given margin.

In Fig. 3.6 we report the training error on the simplex spanned by $y = 3$ solutions in the case where the two vertices have been sampled with the same margin of the problem $k_1 = \kappa_E = -0.5$ and the third one (placed on the top) is a robust solution with margin $k_2 > \kappa_E$. Since the two solutions placed at the bottom are typical solutions to the problem, the energy barrier is non vanishing along the geodesic path; however the two solutions are connected by passing through the center of the interior of the simplex. In the same plot, the red line delimits the region of the manifold at zero-energy (above it) from the one at non-zero energy (below). It also represents the minimum length path needed to pass from a typical solution to the other one.

By investigating numerically the training error, one sees that the whole simplex goes to zero as soon as the maximum barrier along the geodesic path goes to zero, i.e. when $k_1 \geq \kappa_{\text{krn}}$. This shows that κ_{krn} does not have a dependence on the number of replicas y as in the case of $\kappa_y^\star$ and that a robust configuration within the kernel of the solution space are also connected, via

a path passing through the simplex, with solutions in the neighborhood of a typical solution. This confirms the overall star-shapedness of the solution space.

3.C Stability Distribution of the Interpolated Solution

In this section, we want to understand which type of configurations are found along the geodesic path between two solutions. To this end, we derive an analytic expression for the stability distribution on the geodesic path connecting two weights $\boldsymbol{W}_\gamma = \sqrt{N}\frac{\gamma \boldsymbol{W}^1 + (1-\gamma)\boldsymbol{W}^2}{\|\gamma \boldsymbol{W}^1 + (1-\gamma)\boldsymbol{W}^2\|}$, with margin k_1 and k_2, i.e.

$$P(\Delta) = \mathbb{E}_\xi \left\langle \delta\Big(\Delta - \frac{\boldsymbol{W}_\gamma \cdot \boldsymbol{\xi}^1}{\sqrt{N}}\Big) \right\rangle_{k_1,k_2} = \tag{3.87}$$

$$= \mathbb{E}_\xi \frac{1}{Z_\xi(k_1,k_2)} \int d\boldsymbol{W}^1 d\boldsymbol{W}^2\, \mathbb{X}_\xi(\boldsymbol{W}^1;k_1)\mathbb{X}_\xi(\boldsymbol{W}^2;k_2)\delta\left(\Delta - \frac{\mathbf{W}_\gamma \cdot \boldsymbol{\xi}^1}{\sqrt{N}}\right). \tag{3.88}$$

By writing $Z^{-1} = \lim_{n\to 0} Z^{n-1}$, one can again write this expression as an integral over a set of n replicas, and to subsequently take the $n \to 0$ limit. Proceeding in a similar fashion to the free entropy computations along the paper, one arrives to the form

$$P(\Delta) = \lim_{n\to 0} \int \prod_{a<b,rs} \frac{dq_{rs}^{ab}\, d\hat{q}_{rs}^{ab}}{2\pi} \int \prod_{ar} \frac{d\hat{h}_r^a}{2\pi} e^{-\frac{N}{2}\sum_{a<b,rs} q_{rs}^{ab}\hat{q}_{rs}^{ab} + N\sum_{ae}\hat{h}_r^a + NG_S + N\alpha G_E} f_\gamma(\{q_{rs}^{ab}\},\Delta), \tag{3.89}$$

where

$$f_\gamma(\{q_{rs}^{ab}\},\Delta) = \int \prod_{ar} \frac{d\lambda_r^a d\hat{\lambda}_r^a}{2\pi} \prod_{ar} \Theta(\lambda_r^a - k_r) \times \tag{3.90}$$

$$\times\, e^{i\sum_{ar}\lambda_r^a\hat{\lambda}_r^a - \frac{1}{2}\sum_{arbs}\hat{\lambda}_r^a q_{rs}^{ab}\hat{\lambda}_s^b} \delta\Big(\Delta - \frac{\gamma\lambda_1^1 + (1-\gamma)\lambda_2^1}{c_\gamma}\Big).$$

In the asymptotic $N \to \infty$ limit, by employing Laplace's method we find that the integral in Eq. (3.89) is given by

$$e^{nN\phi(\{(q_{rs}^{ab})^\star\})} f_\gamma(\{(q_{rs}^{ab})^\star\},\Delta), \tag{3.91}$$

where $\{(q_{rs}^{ab})^\star\}$ are the values which extremize the free entropy. In the $n \to 0$ limit the exponential term in Eq. (3.91) tends to 1, and we are thus left with $P(\Delta) = f_\gamma(\{(q_{rs}^{ab})^\star\},\Delta)$.

By imposing the RS-ansatz given by Eqs. (3.32), we end up with the final formula

$$
\begin{aligned}
P(\Delta) = \sqrt{\frac{\Gamma}{2\pi(1-q_1)}}\frac{c_\gamma}{\gamma}\Theta\Big(c_\gamma\Delta-\gamma k_1-(1-\gamma)k_2\Big)\times \\
\times\int \mathcal{D}z\mathcal{D}y\, e^{\frac{1}{2}\Xi^2-\frac{\left(b_1(y)\gamma+b_2(z,y)(1-\gamma)+\Delta W\right)^2}{2(1-q_1)\gamma^2}}\times \\
\times\frac{H\left(\frac{(b_2(z,y)+k_2)}{\sqrt{\Gamma(1-q_2)}}-\Xi\right)-H\left(\frac{\left(b_2(z,y)(1-\gamma)-\gamma k_1+\Delta c_\gamma\right)}{\sqrt{\Gamma(1-q_2)}(1-\gamma)}-\Xi\right)}{\prod_r H\Big(\frac{b_r+k_r}{\sqrt{1-q_r}}\Big)},
\end{aligned}
\tag{3.92}
$$

where we have introduced the functions

$$b_1(y) = \sqrt{q_1}y, \tag{3.93}$$

$$b_2(z,y) = \frac{p}{\sqrt{q_1}}y+\sqrt{q_2-\frac{p^2}{q_1}}z, \tag{3.94}$$

and the constants

$$\Gamma = \frac{(1-q_1)\gamma^2}{(1-q_1)\gamma^2+(1-q_2)(\gamma-1)^2}, \tag{3.95}$$

$$\Xi = \frac{(1-\gamma)\sqrt{1-q_2}(\|W\|\Delta+\gamma b_1(y)+(1-\gamma)b_2(z,y))}{\gamma\sqrt{1-q_1}\sqrt{(1-\gamma)^2(1-q_2)+\gamma^2(1-q_1)}}, \tag{3.96}$$

$$c_\gamma = \sqrt{\gamma^2+(1-\gamma)^2+2\gamma(1-\gamma)p}. \tag{3.97}$$

Similarly to the case of the stability distribution for typical solutions, and since we are dealing with the error counting loss, there is a value of Δ up to which the distribution is zero. In the case of typical solutions this threshold is simply the margin of the solution κ_E, while in the interpolated case its value is given by

$$\Delta_{eff} = \frac{\gamma k_1+(1-\gamma)k_2}{c_\gamma}, \tag{3.98}$$

which at the extremes of the path reduces precisely to the thresholds of the two typical solutions sampled with margins $\Delta_{eff}|_{\gamma=0} = k_2$ and $\Delta_{eff}|_{\gamma=1} = k_1$, while for each $\gamma \in (0,1)$ could be interpreted as the effective margin of solutions along the path.

In Fig. 3.7 we plot the stability distribution of the solutions along the geodesic for $k_2 = 0.1 > \kappa_{\text{krn}}$ (notice that, coherently with the definition of κ_{krn}, we find that if $k_2 > \kappa_{\text{krn}}$ then $k_1 \leq \Delta_{eff} \leq k_2\ \forall\gamma \in [0,1]$, implying that all the configurations along the geodesic are solutions). Even though the presence of a sharp threshold separating zero from non-zero values

of the stability distribution is reminiscent of the typical solutions case, the functional form of the distribution itself beyond this threshold is qualitatively different from the one of typical solutions at a given margin value k (i.e. it does not follow a truncated Gaussian), implying that the solutions found along the geodesic path are not that of the equilibrium measure at that k, so we do not have control on their sampling.

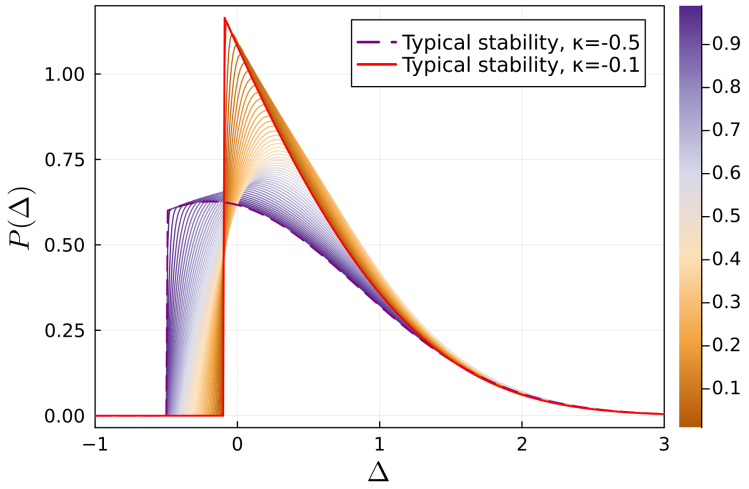

Figure 3.7: Stability distributions of solutions found along the geodesic path between a typical solution at $k_1 = -0.5$ towards a more robust solution at $k_2 = -0.1$, varying the interpolation parameter γ_1 and for $\alpha = 1$. Notice that the geodesic connecting the two solutions in this case is at zero training error. Purple and red lines at the extremes of the curves represent the typical stability distribution for $\kappa_E = -0.5$ and $\kappa_E = -0.1$ respectively.

3.D Numerical Simulations

3.D.1 Numerical Simulation Details

Focusing Belief Propagation Focusing Belief Propagation (fBP) is an heuristic modification of BP that targets flat regions in the solution space [Baldassi et al., 2016]. In the negative perceptron problem the solution obtained with fBP are a good approximation of the κ_{max} solutions (see Baldassi et al. [2023]). In the fBP simulations (see [Baldassi et al., 2020] for implementation details) we used $y = 10$ replicas and an increasing coupling constant γ with 15 exponentially spaced values in the range $[0.1, 10]$. For each value of γ we initialized fBP with the messages obtained at the previous coupling value. We then run it until the maximum

absolute difference between messages was less than 10^{-5} or until it reached a maximum number of 500 messages updates.

Simulated Annealing In order to sample solutions as uniformly as possible from the solution space, and thus capture the typical solutions, we employ Simulated Annealing (SA) on the quadratic Hinge loss, defined as follows:

$$\ell(\Delta_{\kappa_E}) \equiv \Delta^2_{\kappa_E}\Theta(-\Delta_{\kappa_E}), \tag{3.99}$$

where $\Delta^\mu_{\kappa_E} = y^\mu(w\cdot\xi^\mu) - \kappa_E\|w\|$. Optimizing the Hinge loss instead of the number of errors function makes the sampling much faster as the SA dynamics benefits from the high-energy solutions organization, while the zero-energy manifold is unaltered. The weight vectors w are initialized on the unit sphere. At each step, we propose a move in a random direction $w' = w + \eta$, with $\eta = \epsilon\mathcal{N}(0,1)$, and $\epsilon = 5\times10^{-3}$ in the simulations. The weights are updated with probability $p = \min\left(1, e^{-\beta\Delta\ell}\right)$, where $\Delta\ell$ is the loss difference between w and w'. After N move proposals we increase the inverse temperature as $\beta \leftarrow \beta + d\beta$, with $\beta_0 = 2000$ and $d\beta = 2$. The simulation is interrupted whenever a solution is found or when the maximum number of 200000 allowed updates per weight is reached. As it was previously noted in **?**, the scaling with the instance size N of the convergence times of SA is worst-case exponential, and this fact limits our analysis to small and medium sizes $N \leq 4000$.

Gradient Descent on cross-entropy. We employed a standard gradient descent optimization strategy. Notice that the standard cross-entropy/logistic loss can be adapted to the negative perceptron problem by adding the corresponding margin κ_E in the exponential term of the sigmoid:

$$\ell(\Delta_{\kappa_E}) = \log(1 + e^{-\Delta_{\kappa_E}}). \tag{3.100}$$

The weights are initialized uniformly on the unit sphere, and optimized with a batchsize 200 and learning rate 1, with a maximum number of allowed epochs equal to 20000. Early stopping is implemented, interrupting the trajectory as soon as a zero energy configuration is found.

Perceptron Algorithm. We implemented the standard Perceptron Algorithm (PA) (see e.g. Engel and Van den Broeck [2001]). For each pattern, if the stability Δ^μ on the μ -th pattern is greater than zero, then the weights are updated as $w \leftarrow w + 2\eta\xi^\mu$ where η is the learning rate. After each weight update, we project the weight vector onto the unit sphere (in order to avoid a huge increase of the norm and a slowing down of the optimization progress). Notice that the PA algorithm is equivalent to a gradient descent optimization on the Hinge loss with unit batch size, and similar results can be obtained also with larger batch sizes provided the learning rate is lowered. By varying the learning rate, we were able to obtain PA solutions with a wide range of stability distribution (see Fig. 3.8 and the discussion in the following Section).

3.D.2 Sampling bias of the Perceptron Algorithm as a function of the learning rate

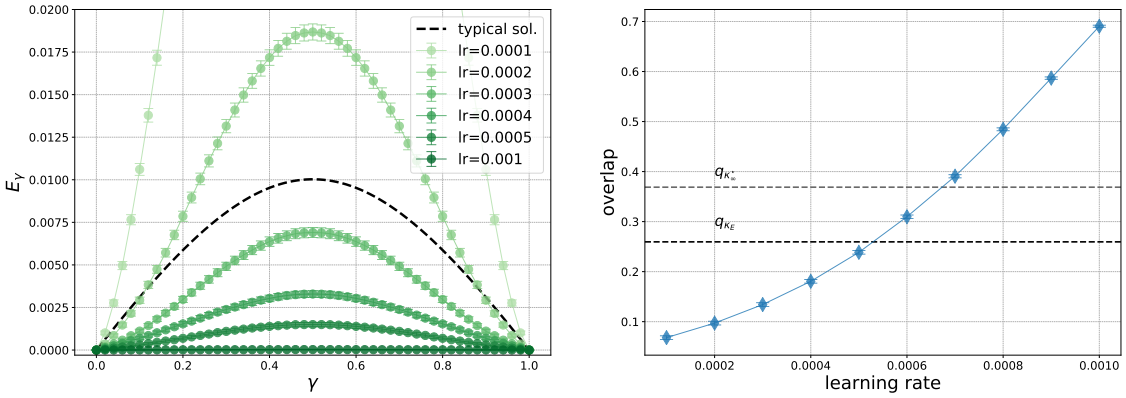

Figure 3.8: (Left) Maximum fraction of errors along the geodesic path connecting two solutions obtained with PA at $\alpha = 1$ and $\kappa_E = 0.5$ using different learning rates at $N = 1000$. Increasing the learning rate PA is able to find geodesically connected solutions. We also plot as a reference the barrier between typical solutions (dashed line). (Right) Average overlap between PA solutions as a function of the learning rate. Horizontal dashed lines are q_{κ_E} and $q^\star_\infty$ at $\alpha = 1$. When PA finds solutions with higher energy barrier with respect to typical solutions, they are further on average w.r.t. typical solutions (smaller overlap). As soon as the geodesic path connecting PA solutions has zero energy, the solutions have average overlap comparable with the overlap at the coalescence threshold.

Finally, we investigate the dependence of the sampling bias of PA as a function of the learning rate employed in the dynamics. As commented above and in the main text, when the learning rate is sufficiently small, this algorithm is able to find solutions that display an atypical distribution of stabilities peaked around the minimum margin allowed in the problem. These solutions are typically found very close to initialization and have an extremely small overlap with one another. This overlap is much smaller than that of the typical solutions of the problem, which numerically dominate the flat measure over solutions. While at low constraint densities these solutions have no barriers with respect to the fBP solutions, they instead show non-zero barriers between each other at any α.

Interestingly, the type of solution sampled by PA can be completely altered by increasing the learning rate employed for the optimization. As shown in the left panel of Fig. 3.8, on the one hand, the quicker optimization trajectories corresponding to higher learning rates find pairs of solutions separated by lower and lower energy barriers, crossing at a certain value the typical energy barrier between the equilibrium solutions (dashed black line). On the other hand, the reciprocal overlap increases up to values that are much larger than that of the equilibrium

solutions and comparable with that of the solutions in the kernel regions at margin $\kappa^{\star}_{\infty}$, as shown in the right panel of Fig. 3.8.

Chapter 4

Breaking the Replica Symmetry

Up to now all calculations and results have been derived in the *Replica Symmetric* ansatz. When using such hypothesis, we tacitly assume that the relevant order parameters concentrate, that is if we sample two solutions $\boldsymbol{W}^1, \boldsymbol{W}^2$ from the distribution $P_{\mathcal{D},\kappa}(\boldsymbol{W})$, their overlap in the asymptotic limit will concentrate to a value

$$\frac{1}{N}\boldsymbol{W}^1 \cdot \boldsymbol{W}^2 \underset{N\to\infty}{\to} \mathbb{E}_{\mathcal{D}} \lim_{N\to\infty} \frac{1}{N}\langle \boldsymbol{W}^1 \cdot \boldsymbol{W}^2 \rangle = q, \tag{4.1}$$

that can be obtained from the saddle point equations. This is guaranteed to happen if the space of solutions is convex. However when we introduce non-convexity, for example considering a negative margin, there is no such guarantee, and indeed for a portion of the phase diagram concentration will not hold. In these phases a more complicated ansatz for the overlap matrix will be needed. In this chapter, we will see how the "Replica Symmetry" will need to be broken, and what results can be derived using this "broken" ansatz. We will consider two continuous models of Neural Networks, the *Tree-Committee Machine* [Barkai et al., 1990, Engel et al., 1992, Baldassi et al., 2019, Franz et al., 2019] with arbitrary non-linearity and the *Spherical Negative Perceptron*, both of which are known to have non-convex spaces of solutions. We settle a long-standing open problem about the numerical value of the SAT-UNSAT threshold. Previous estimates were all derived under the *Replica Symmetric* (RS) and *1-step Replica Symmetry Breaking* (1RSB) assumption, both of which only provide an approximation to the actual value.

Furthermore, we identify a new phase transition line between the *Full-Replica Symmetry Breaking* (fRSB) and the Gardner phase in the negative perceptron, where typical solutions develop a so called *Overlap Gap* [Gamarnik, 2021]. We discuss this in connection to recently developed algorithms based on Approximate Message Passing [El Alaoui and Sellke, 2022, Montanari, 0], which provably finds solutions conditioned on the absence of this Overlap Gap.

The rest of the chapter is organized as follows. In section 4.1, I review some important works that have studied these models. In Section 4.2, I precisely define the models and learning tasks we are interested in, namely the classification of random patterns and labels. In Section 4.3, I summarize the main steps in the analytical calculation we performed. In Section 4.4, I introduce a simple method through which we are able to compute the exact SAT/UNSAT threshold of those models with high precision. In Section 4.5, I study the transition to the Gardner phase starting from the fRSB phase, and propose an empirical method for the numerical estimation of this threshold. In Section (4.6), I discuss where commonly used algorithms such as Gradient Descent are able to find solutions. Finally, in Section (4.7), I go back to the problem of linear connectivity in the fRSB phase of the negative perceptron. Most results of this chapter can be found in [Annesi et al., 2024].

4.1 Related Works

Previous works on the tree-committee machine in the large width limit with sign [Barkai et al., 1990, 1992, Engel et al., 1992] and other non-linear activation functions such as ReLU [Baldassi et al., 2019, Zavatone-Veth and Pehlevan, 2021] have only characterized the SAT/UNSAT transition in the Replica Symmetric (RS) or 1-step Replica Symmetry Breaking (1RSB) approximation. Recently, using fully-lifted random duality theory techniques, Refs. [Stojnic, 2024b,a] obtained results compatible with RS, 1RSB and 2-steps RSB approximations. One of the goals of the present chapter is to compute *exaclty* α_c in the infinite-width limit regime.

The negative perceptron model has been thoroughly studied in connection to jamming in high dimensions [Franz and Parisi, 2016, Franz et al., 2017]. Along these lines of research, some efforts have been devoted at understanding the universality class of this SAT-UNSAT (in this contest, *jamming*) transition [Charbonneau et al., 2014, Franz et al., 2017], and identifying further models that belong to this same universality class [Franz et al., 2019, Sclocchi and Urbani, 2022]. In this context the critical exponents of the model were computed [Franz et al., 2017] and were shown to be exactly the same as those observed in the jamming of spheres in large dimensions [Charbonneau et al., 2014]. Recently, the tree-committee machine with several activation functions and the parity machine with a finite number of hidden units K [Franz et al., 2019] have also been shown to pertain to the same universality class.

From the optimization point of view, imposing a negative margin is necessary in order to obtain a non-convex model: for $\kappa \geq 0$ indeed the space of solutions is convex and algorithms are able to reach capacity, which can be obtained exactly using an RS ansatz [Gardner, 1988, Gardner and Derrida, 1989]. For $\kappa < 0$ the space of solution is instead non-convex [Malatesta, 2023] and, in the overparameterized regime $\alpha \ll 1$, we have seen above that it is *star-shaped*

[Annesi et al., 2023]. From the point of view of algorithmic dynamics, at present it is difficult to compare the algorithmic threshold with the capacity transition since, as in the case of the committee machine, we only know approximations [Baldassi et al., 2023] or upper bounds [Montanari et al., 2021] to the true value of the latter.

In [El Alaoui and Sellke, 2022], the authors develop an algorithm called incremental Approximate Message Passing (iAMP), originally devised in [Montanari, 0] for approximating the ground state of the Sherrington-Kirkpatrick model. Interestingly, this algorithm can be proven to reach capacity, provided that the typical states exhibit no overlap gap, i.e. the overlap distribution of typical states is with a connected support. This is what we refer in the rest of the paper as a no overlap gap condition (nOG). Notice that the nOG condition is different from the no Overlap Gap Property (nOGP) introduced by Gamarnik [Gamarnik, 2021] and that was connected to algorithmic hardness for stable algorithms. The OGP condition indeed requires that there exist q_1 and q_2 with $q_1 < q_2$ such that all pairs of solutions are at most at an overlap q_1 or at least an overlap q_2, i.e. there exists a "gap" since one cannot find solutions in the interval $q \in [q_1, q_2]$. As such, the OG condition we have introduced here is weaker with respect to the OGP since the last refers to all possible pairs of solutions, whereas the first one only to the typical ones. As a consequence the OGP implies the OG (while the nOG implies nOGP). A paradigmatic example showing the difference between OG and OGP is the binary perceptron: it has been shown that for any constraint density $\alpha > 0$ the problem exhibits OG (i.e. typical states are gapped) [Huang and Kabashima, 2014, Abbe et al., 2021], but it is conjectured to display the OGP only above a critical value of the constrained density $\alpha > \alpha_{OGP} \simeq 0.77$ [Baldassi et al., 2015, 2021].

In the present paper we identify all the regions in the (κ, α) phase diagram that satisfy the nOG and we compute a new transition line separating a nOG from an overlap gapped phase for the typical states.

4.2 The Model

The models that we study in this chapter are the *Spherical Negative Perceptron*, defined above, and the *Tree-Committee Machine,* a neural network with one hidden layer having non-overlapping receptive fields and fixed second layer weights. The architecture of the latter is depicted in Fig. 4.1. Mathematically, given a N-dimensional input vector $\boldsymbol{x}$, its output is computed as

$$\hat{y} = \text{sign}\left(\frac{1}{\sqrt{K}} \sum_{l=1}^{K} c_l \, \varphi(\tau_l) \right), \tag{4.2}$$

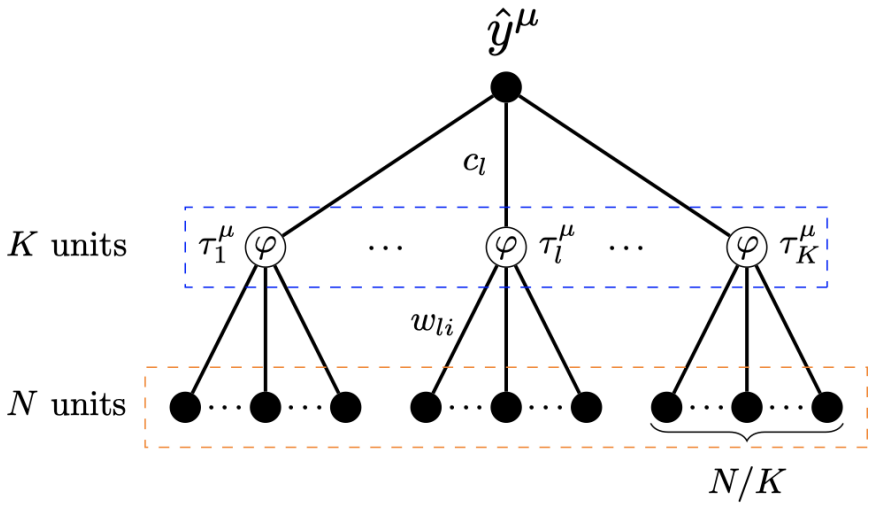

Figure 4.1: Tree-committee-machine architecture.

where K is the width of the hidden layer, c_l are the weights of the second layer and τ_l is the l-th receptive field, given by

$$\tau_l \equiv \sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w_{li} x_{li}, \tag{4.3}$$

where w_{li}, $i \in \left[\frac{N}{K}\right]$, $l \in [K]$ are the N weights of the first layer, $\varphi(\bullet)$ is a generic activation function and we have used the notation $x_{li} = x_{i+l\frac{N}{K}}$. We will consider in the following the case of *spherical weights*: individually $w_{li} \in \mathbb{R}$, but each branch of the weights is constrained to live on the N/K-dimensional sphere of radius $\sqrt{N/K}$,

$$\sum_{i=1}^{N/K} w_{li}^2 = \frac{N}{K}, \qquad l \in [K]\,. \tag{4.4}$$

The weights of the second layer will be considered fixed to $c_l = \pm 1$ or to $c_l = 1$ respectively depending if the activation function φ is odd or not. Another choice could be to impose all the c_l to be 1 and subtract a bias term $\sqrt{K}b$ inside the sign of equation (4.2), so that the preactivation of the output has zero mean. Notice that in the case of the identity activation function $\varphi(h) = h$ and for $K = 1$, we recover the *perceptron* architecture.

Again, we consider the *storage* problem, where the training set $\mathcal{D} = \{\boldsymbol{x}^\mu, y^\mu\}_{\mu=1}^P$ is composed of inputs distributed as i.i.d. standard normal Gaussian variables $x_i^\mu \sim \mathcal{N}(0,1)$, $\forall \mu, i$ and the corresponding label will be $y^\mu = \pm 1$ with equal probability. We consider learning with a

margin, as outlined in the previous chapters, and will be interested in particular in the SAT-UNSAT threshold $\alpha_c(\kappa)$ as a function of this margin.

4.3 Accessing the entropy of solutions via the replica method

Using the same tools that have been described in the previous part, we start by writing the expression for the Gardner volume

$$\mathcal{Z}_{\mathcal{D}} = \int d\mu(\boldsymbol{w}) \prod_{\mu=1}^{P} \Theta\left(\Delta^{\mu}(\boldsymbol{w};\kappa)\right) = \int d\mu(\boldsymbol{w})\, \mathbb{X}_{\mathcal{D}}(\boldsymbol{w};\kappa) \ . \tag{4.5}$$

We are interested in computing the average log-volume of solutions, i.e. the entropy of the system

$$\phi = \lim_{N\to\infty} \mathbb{E}_{\mathcal{D}} \frac{1}{N} \ln \mathcal{Z}_{\mathcal{D}} \ , \tag{4.6}$$

Using the replica method to take the average of the log, the order parameters

$$q_l^{ab} \equiv \frac{K}{N} \sum_{i=1}^{N/K} w_{li}^a w_{li}^b \, , \quad a < b \in [n] \, , \, l \in [K] \, , \tag{4.7}$$

naturally appear. They represent the overlap between the same hidden unit of two independent replicas of the systems. The overlap between different hidden units does not contribute because they are connected to non-overlapping, uncorrelated portion of the input. We enforce the definition (4.7) by using delta functions and their integral representations; this will in turn introduce the conjugated parameters $\hat{q}_l^{ab}$ with $a \leq b \in [n]$, $l \in [K]$. Notice that we need also the diagonal conjugated overlaps $\hat{q}_l^{aa}$ in order to enforce the spherical constraint in equation (4.4). In the end we get the following representation of the averaged replicated partition function

$$\overline{Z_{\mathcal{D}}^n} = \int \prod_{\substack{a<b \\ l}} dq_l^{ab} \prod_{\substack{a\leq b \\ l}} d\hat{q}_l^{ab} \, e^{NS(\boldsymbol{q},\hat{\boldsymbol{q}})} \, , \tag{4.8}$$

where we have defined

$$S(\boldsymbol{q},\hat{\boldsymbol{q}}) \equiv G_S(\boldsymbol{q},\hat{\boldsymbol{q}}) + \alpha G_E(\boldsymbol{q}) \,, \tag{4.9}$$

$$G_S(\boldsymbol{q},\hat{\boldsymbol{q}}) \equiv \frac{1}{2K}\sum_{ab}\sum_{l} q_l^{ab}\hat{q}_l^{ab} - \frac{1}{2K}\sum_{l=1}^{K}\ln\det\hat{\boldsymbol{q}}_l \,, \tag{4.10}$$

$$G_E(\boldsymbol{q}) \equiv \ln \mathbb{E}_y \int \prod_{la}\frac{d\lambda_l^a d\hat{\lambda}_l^a}{2\pi}\prod_a e^{-\beta\ell\left(\frac{y}{\sqrt{K}}\sum_{l=1}^{K} c_l\,\varphi(\lambda_l^a)-\kappa\right)} e^{i\sum_{la}\lambda_l^a\hat{\lambda}_l^a - \frac{1}{2}\sum_{ab,l} q_l^{ab}\hat{\lambda}_l^a\hat{\lambda}_l^b} \,, \tag{4.11}$$

we have understood that $q_l^{aa}=1$ because of the spherical constraint (4.4) and we have used the notation $\bar{f}=\mathbb{E}_{\mathcal{D}}f$. The conjugated parameters satisfy saddle point equations that can be explicitly solved: $q_l^{ab}=\left[\hat{\boldsymbol{q}}_l^{-1}\right]^{ab}$. Therefore the averaged replicated partition function can be written more compactly as

$$\overline{Z_{\mathcal{D}}^n} = \int \prod_{\substack{a<b\\ l}} dq_l^{ab}\, e^{NS(\boldsymbol{q})} \,, \tag{4.12}$$

where

$$S(\boldsymbol{q}) \equiv G_S(\boldsymbol{q}) + \alpha G_E(\boldsymbol{q}) \,, \tag{4.13}$$

$$G_S(\boldsymbol{q}) \equiv \frac{1}{2K}\sum_{l=1}^{K}\ln\det\boldsymbol{q}_l \,, \tag{4.14}$$

$$G_E(\boldsymbol{q}) \equiv \ln \mathbb{E}_y \int \prod_{la}\frac{d\lambda_l^a d\hat{\lambda}_l^a}{2\pi}\prod_a e^{-\beta\ell\left(\frac{y}{\sqrt{K}}\sum_{l=1}^{K} c_l\,\varphi(\lambda_l^a)-\kappa\right)} e^{i\sum_{la}\lambda_l^a\hat{\lambda}_l^a - \frac{1}{2}\sum_{ab,l} q_l^{ab}\hat{\lambda}_l^a\hat{\lambda}_l^b} \,. \tag{4.15}$$

Notice that we recover the perceptron by imposing $\varphi(x)=x$ and $K=1$. We write both the entropic and energetic terms for later convenience

$$G_S(\boldsymbol{q}) = \frac{1}{2}\ln\det\boldsymbol{q} \,, \tag{4.16}$$

$$G_E(\boldsymbol{q}) \equiv \ln \mathbb{E}_y \int \prod_{a}\frac{d\lambda^a d\hat{\lambda}^a}{2\pi}\prod_a e^{-\beta\ell(y\lambda^a-\kappa)} e^{i\sum_a\lambda^a\hat{\lambda}^a - \frac{1}{2}\sum_{ab} q^{ab}\hat{\lambda}^a\hat{\lambda}^b} = \tag{4.17}$$

$$= \ln \mathbb{E}_y\, e^{\frac{1}{2}\sum_{ab} q^{ab}\frac{\partial^2}{\partial h^a\partial h^b}}\prod_a e^{-\beta\ell(yh^a-\kappa)}\Big|_{h^a=0} \,. \tag{4.18}$$

In the last step we have integrated over the $\hat{\lambda}^a$ variables and used the following set of identities

$$\int \prod_a \frac{d\lambda^a}{\sqrt{2\pi \det \boldsymbol{q}}} e^{-\frac{1}{2}\sum_{ab}\left[\boldsymbol{q}^{-1}\right]^{ab}\hat{\lambda}^a\hat{\lambda}^b} \prod_a g(\lambda^a) = \int \prod_a D\lambda^a \prod_a g\left(\sum_b \left[\sqrt{\boldsymbol{q}}\right]^{ab}\lambda^b\right) = \tag{4.19}$$

$$= \int \prod_a D\lambda^a \, e^{\sum_{ab}\left[\sqrt{\boldsymbol{q}}\right]^{ab}\lambda^b \frac{d}{dh_a}} \prod_a g\left(h_a\right)\Big|_{h_a=0} = e^{\frac{1}{2}\sum_{ab} q^{ab}\frac{d^2}{dh_a dh_b}} \prod_a g\left(h_a\right)\Big|_{h_a=0}, \tag{4.20}$$

where $g(\bullet)$ is a generic function and $D\lambda \equiv \frac{d\lambda}{\sqrt{2\pi}} e^{-\frac{\lambda^2}{2}}$. We have also used the notation $\sqrt{\boldsymbol{q}}$ to denote the square root of the symmetric (and therefore positive semidefinite) overlap matrix q^{ab}.

4.3.1 Large width limit

The large number of hidden units limit can be performed before imposing the ansatz over the replica indices of the overlap matrix q_l^{ab}. An important point to notice in this regard is that since the weights are not overlapping and have access to uncorrelated portions of the input, clearly q_l^{ab} must be independent on l on average. We can exploit this to simplify notably the entropic and energetic terms. The entropic term is easy and it reads

$$G_S(\boldsymbol{q}) = \frac{1}{2}\ln \det \boldsymbol{q} \ . \tag{4.21}$$

In the energetic term (4.15) we have instead to use the central limit theorem on the variable $u_a = \frac{1}{\sqrt{K}}\sum_{l=1}^K c_l \, \varphi(\lambda_l^a)$. This can be done extracting the variables u_a from the loss function in (4.15) via n delta functions, inserting their integral representations, Taylor expanding at second order and re-exponentiating. Performing those steps and using identity (4.19) we get

$$G_E(\boldsymbol{q}) \equiv \ln \mathbb{E}_y \int \prod_a \frac{du_a d\hat{u}_a}{2\pi} e^{i\sum_a u_a\hat{u}_a} \int \prod_{la} \frac{d\lambda_l^a d\hat{\lambda}_l^a}{2\pi} \prod_a e^{-\beta\ell(yu_a-\kappa)} \tag{4.22}$$

$$\times \, e^{i\sum_{la}\lambda_l^a\hat{\lambda}_l^a - \frac{1}{2}\sum_{ab,l} q_l^{ab}\hat{\lambda}_l^a\hat{\lambda}_l^b - i\sum_a \hat{u}_a \frac{1}{\sqrt{K}}\sum_{l=1}^K c_l\,\varphi(\lambda_l^a)} \tag{4.23}$$

$$= \ln \mathbb{E}_y \int \prod_a \frac{du_a d\hat{u}_a}{2\pi} e^{i\sum_a u_a\hat{u}_a} \prod_a e^{-\beta\ell(yu_a-\kappa)} e^{-i\sum_a \hat{u}_a M_a - \frac{1}{2}\sum_{ab}\Delta_{ab}\hat{u}_a\hat{u}_b} \tag{4.24}$$

$$= \ln \mathbb{E}_y \, e^{\frac{1}{2}\sum_{ab}\Delta_{ab}\frac{\partial^2}{\partial h_a \partial h_b}} \prod_a e^{-\beta\ell(y(M_a+h_a)-\kappa)}\Big|_{h_a=0}, \tag{4.25}$$

where M_a and Δ_{ab} represents respectively the mean and the covariance matrix of the variable u_a, i.e.

$$M_a \equiv m_c \int \prod_a \frac{d\lambda_l^a d\hat{\lambda}_l^a}{2\pi} e^{i\sum_a \lambda_l^a \hat{\lambda}_l^a - \frac{1}{2}\sum_{ab} q^{ab}\hat{\lambda}_l^a \hat{\lambda}_l^b} \varphi(\lambda_l^a) \equiv m_c \left\langle \varphi(\lambda^a) \right\rangle \ , \tag{4.26}$$

$$\Delta_{ab} \equiv \sigma_c \left[\left\langle \varphi(\lambda^a)\varphi(\lambda^b) \right\rangle - \left\langle \varphi(\lambda^a) \right\rangle \left\langle \varphi(\lambda^b) \right\rangle \right] , \tag{4.27}$$

with $m_c \equiv \frac{1}{\sqrt{K}} \sum_{l=1}^K c_l$ and $\sigma_c \equiv \frac{1}{K} \sum_{l=1}^K c_l^2$. They can also be written more compactly using identity (4.19) as

$$M_a \equiv m_c \, e^{\frac{1}{2}\sum_{ab} q^{ab} \frac{\partial^2}{\partial s_a \partial s_b}} \varphi(s_a) \Big|_{s_a=0} \ , \tag{4.28}$$

$$\Delta_{ab} \equiv \sigma_m \, e^{\frac{1}{2}\sum_{ab} q^{ab} \frac{\partial^2}{\partial s_a \partial s_b}} \varphi(s_a)\varphi(s_b) \Big|_{s_a=0} - m_c^2 M_a M_b \ . \tag{4.29}$$

Notice that in our case the mean M_a is always vanishing: if the activation function is odd indeed $\langle \varphi(\lambda^a) \rangle = 0$, whereas if the activation function is even $m_c = 0$ since $c_l = \pm$ with equal probability in order to prevent the model to have a bias towards positive or negative labels. We therefore get the following integral representation of the model in the large K limit:

$$\overline{Z_{\mathcal{D}}^n} = \int \prod_{a<b} dq^{ab} \, e^{NS(\boldsymbol{q})} \ , \tag{4.30}$$

$$S(\boldsymbol{q}) = \frac{1}{2} \ln \det \boldsymbol{q} + \alpha \ln \left(\mathbb{E}_y \, e^{\frac{1}{2}\sum_{ab} \Delta_{ab} \frac{\partial^2}{\partial h_a \partial h_b}} \prod_a e^{-\beta \ell(y h_a - \kappa)} \Big|_{h_a=0} \right) . \tag{4.31}$$

Notice that this expression is exactly equal in form to that of the perceptron model, see equation (4.3); the only difference is that instead of having the matrix q^{ab} we have an effective order parameter Δ_{ab} which is a function through $\varphi(\cdot)$ of q^{ab}. This has been evidenced for the first time in [Baldassi et al., 2019]. The quantity Δ_{ab} is also exactly identical to the so-called Neural Network Gaussian Process (NNGP) kernel [Zavatone-Veth and Pehlevan, 2022] that appears as the covariance matrix of the function implemented by a neural network at initialization (i.e. with random weights) in the infinite width limit and given two different inputs [Williams, 1996]. Here, the only difference is that this quantity does not depend on the overlap between those two inputs, but it depends instead on the average overlap q^{ab} between two different replicas of the weights extracted from the Gibbs measure.

Saddle point equations

In the large N limit, the averaged replicated partition function in equation (4.3.1) is dominated by the saddle points of the action $S(\boldsymbol{q})$. The entropy of the system can be therefore written as

$$\phi = \lim_{n\to 0} \max_{\boldsymbol{q}} \frac{S(\boldsymbol{q})}{n} \,. \tag{4.32}$$

The stationary points of the action can be obtained by imposing that the first derivative of the action vanishes. This set of $\frac{n(n-1)}{2}$ saddle point equations read, in the large width limit, as

$$q_{cd}^{-1} = -\alpha \frac{d\Delta_{cd}}{dq_{cd}} \frac{\mathbb{E}_y \, e^{\frac{1}{2}\sum_{ab}\Delta_{ab}\frac{\partial^2}{\partial h_a \partial h_b}} \frac{\partial^2}{\partial h_c \partial h_d} \prod_a e^{-\beta\ell(y h_a - \kappa)} \Big|_{h_a=0}}{\mathbb{E}_y \, e^{\frac{1}{2}\sum_{ab}\Delta_{ab}\frac{\partial^2}{\partial h_a \partial h_b}} \prod_a e^{-\beta\ell(y h_a - \kappa)} \Big|_{h_a=0}} \,, \qquad c < d \in [n] \tag{4.33}$$

where

$$\frac{d\Delta_{ab}}{dq^{ab}} = e^{\frac{1}{2}\sum_{cd} q^{cd}\frac{\partial^2}{\partial s_c \partial s_d}} \frac{\partial\varphi(s_a)}{\partial s_a}\frac{\partial\varphi(s_b)}{\partial s_b}\Big|_{s_a=0} \,. \tag{4.34}$$

4.3.2 Full Replica Symmetry Breaking ansatz and variational formulation

In order to solve the saddle point equations in the small n limit, one needs to impose some type of ansatz on the structure of the replica overlap matrix q^{ab}. Here we consider the most general type of ansatz, the k-steps Replica Symmetry Breaking (k-RSB) ansatz [Parisi, 1979b,a, Mézard et al., 1987], in which it is assumed that the overlap matrix assumes the $k+1$ values $q_0, q_1, \ldots, q_k$. Defining the set of integers $1 = m_k \leq m_{k-1} \leq \cdots \leq m_0 \leq m_{-1} \equiv n$ with m_{s-1} divisible by m_s for $s = 0, \ldots, k-1$ the overlap matrix q^{ab} is written in the k-RSB ansatz as

$$q^{ab} = q_0 + \sum_{s=0}^{k} (q_{s+1} - q_s) I^{ab}_{n,m_s}, \tag{4.35}$$

where I^{ab}_{n,m_s} is the (a,b) element of a block matrix of size $n \times n$ whose blocks have size $m_s \times m_s$ and contains all ones and all zeros respectively inside and outside the blocks. We have understood in the previous equation that $q_{k+1} = 1$.

In the following we will use the square bracket notation $[\bullet]_s$ to denote the operation of extracting step $s+1$ from the k-step RSB matrix in its argument, i.e., for example, $\left[q^{ab}\right]_s = q_s$. As we show in appendix 4.B also the NNGP kernel Δ_{ab} assumes a k-RSB form with the same block structure of q^{ab}; in addition the $s+1$-th step of Δ_{ab} is given by a simple function of the

$(s+1)$-th step of the matrix q^{ab}

$$[\Delta_{ab}]_s = \int Dx \left[\int Dy\, \varphi\left(\sqrt{q_s}\, x + \sqrt{1-q_s}\, y\right)\right]^2 \equiv \Delta(q_s). \tag{4.36}$$

We report in appendix 4.B the expression of the entropic and energetic term in the small n-limit for the k-step RSB ansatz.

In the small n limit, the parameterization (4.35) is equivalent to requiring that the matrix q^{ab} is parameterized by a stepwise function $q(x)$ in the interval $x \in [0,1]$

$$q(x) = q_s\,, \qquad x \in [m_{s-1}, m_s)\,, \qquad s = 0, \dots, k\,. \tag{4.37}$$

In the large number of steps limit $q(x)$ tends to a continuous function and so does the NNGP kernel function $\Delta(q)$. This is what is called full-RSB ansatz (fRSB). When we are in the fRSB phase, we expect the $q(x)$ that maximises the free energy [4.3.1] to have the following shape: for $x \in [0, x_m)$ and $x \in [x_M, 1)$, $q(x)$ is constant and equal to q_m and q_M respectively, while for $x \in [x_m, x_M]$ it is a continuous monotonic function of x. In the Replica Symmetric (RS) phase instead, we expect the $q(x)$ to be constant and equal to a single value q.

Although the function $q(x)$ is not of easy interpretation, it is connected to a fundamental quantity, namely the probability distribution of the overlap between two samples of the uniform measure over solutions

$$P(q) = \overline{\int d\mu(\boldsymbol{w}^1) d\mu(\boldsymbol{w}^2)\, \mathbb{X}_{\mathcal{D}}(\boldsymbol{w}^1;\kappa)\, \mathbb{X}_{\mathcal{D}}(\boldsymbol{w}^2;\kappa)\, \delta\left(q - \frac{K}{N}\sum_{i=1}^{N/K} w^1_{li} w^2_{li}\right)}. \tag{4.38}$$

Indeed it can be shown that if we denote by $x(q)$ the inverse function of $q(x)$, then $P(q) = \frac{dx(q)}{dq}$ (or in other word $x(q)$ is the CDF of $P(q)$). Performing the continuous limit, the fRSB entropy can be written as

$$\phi = \mathcal{G}_S + \alpha \mathcal{G}_E, \tag{4.39}$$

$$\mathcal{G}_S \equiv \lim_{n\to 0} \frac{G_S}{n} = \frac{1}{2}\left[\ln(1-q_M) + \frac{q_m}{\lambda_m} + \int_{q_m}^{q_M} \frac{dq}{\lambda(q)}\right], \tag{4.40}$$

$$\mathcal{G}_E \equiv \lim_{n\to 0} \frac{G_E}{n} = \int dh\, \mathcal{N}_{\Delta(q_m)-\Delta(0)}(h)\, f(q_m, h), \tag{4.41}$$

having indicated with $\mathcal{N}_\sigma(h) \equiv \frac{e^{-\frac{h^2}{2\sigma}}}{\sqrt{2\pi\sigma}}$ and by $\lambda(q)$ the continuous limit of the eigenvalues of a k-RSB matrix (see also section 4.A.2), i.e.

$$\lambda(q) = \int_q^1 dq' x(q') . \tag{4.42}$$

The function of two variables f in the energetic term satisfies the following partial differential equation (PDE) [Parisi, 1980b, Duplantier, 1981]

$$f(q_M, h) = \ln \int dz \, \mathcal{N}_{\Delta(1)-\Delta(q_M)}(z+h) \, e^{-\beta\ell(z-\kappa)}, \tag{4.43}$$

$$\dot{f}(q,h) = -\frac{1}{2}\dot{\Delta}(q)\left[f''(q,h) + x(q) f'(q,h)^2\right], \tag{4.44}$$

having denoted with a upper dot the derivative with respect to q and with a prime the derivative with respect to h. The second equation (4.44) is a slight variation to the Parisi's equation which is obtained in the case $\Delta(q) = 1$ i.e. in the linear activation (perceptron) case. Notice that for both the error counting loss and the quadratic hinge loss, the initial condition, equation (4.43), can be explicitly solved analytically; in particular, in the large β limit, in both cases one has

$$f(q_M, h) = \ln H\left(\frac{\kappa + h}{\sqrt{\Delta(1) - \Delta(q_M)}}\right), \tag{4.45}$$

where $H(x) \equiv \frac{1}{2}\mathrm{Erfc}\left(\frac{x}{\sqrt{2}}\right)$. The saddle point equations in the continuous limit are difficult to derive differentiating equation (4.41) with respect to $x(q)$, because f depends implicitly on $x(q)$ through equation (4.44). As suggested in [Sommers and Dupont, 1984] we can remove this dependence by using Lagrange's method [Sommers and Dupont, 1984, Duplantier, 1981]

$$\phi_{\text{var}} = \frac{1}{2}\left[\ln(1-q_M) + \frac{q_m}{\lambda_m} + \int_{q_m}^{q_M} \frac{dq}{\lambda(q)}\right] + \alpha \int dz \, \mathcal{N}_{\Delta(q_m)-\Delta(0)}(z) \, f(q_m, z) \tag{4.46}$$

$$- \alpha \int_{-\infty}^{+\infty} dh \, P(q_M, h)\left[f(q_M, h) - \ln \int dz \, \mathcal{N}_{\Delta(1)-\Delta(q_M)}(z+h) \, e^{-\beta\ell(z-\kappa)}\right] \tag{4.47}$$

$$+ \alpha \int_0^1 dq \int_{-\infty}^{+\infty} dh \, P(q,h)\left[\dot{f}(q,h) + \frac{\dot{\Delta}(q)}{2}\left(f''(q,h) + x(q) f'(q,h)^2\right)\right]. \tag{4.48}$$

Deriving ϕ_{var} with respect to $x(q)$ we get the saddle point equations in the continuous limit

$$\frac{q_m}{\lambda_m^2} + \int_{q_m}^{q} \frac{dp}{\lambda^2(p)} = \alpha \dot{\Delta}(q) \int dh \, P(q,h) f'(q,h)^2 . \tag{4.49}$$

Differentiating with respect to $f(q_m,h)$ and $f(q,h)$ we get that the function P satisfies a PDE of the Fokker-Planck type

$$P(q_m,h) = \mathcal{N}_{\Delta(q_m)-\Delta(0)}(h), \tag{4.50}$$

$$\dot{P}(q,h) = \frac{\dot{\Delta}(q)}{2}\Big[P''(q,h) - 2x(q)\big(P(q,h)f'(q,h)\big)'\Big]. \tag{4.51}$$

which can be shown to be equal to the continuous limit of iteration rule given in appendix, equation (4.159).

We show in appendix 4.B how to solve equations (4.3.2) and (4.49) numerically by writing them in a discretized version that correspond to a finite number k of steps of RSB. Once they are solved for a particular guessed value of $q(x)$ in the interval $[x_m, x_M]$, the updated $q(x)$ can be computed from equation (4.49).

4.3.3 Instability of the ansatz

The continuous limit and the variational formulation of the saddle point described above can be also useful as a tool to derive equations describing the instability of the ansatz itself. In order to do that we need to derive equation (4.49) written in terms of x,

$$\frac{q_m}{\lambda_m^2} + \int_{x_m}^{x} dy\, \frac{\dot{q}(y)}{\lambda^2(y)} = \alpha\dot{\Delta}(q(x)) \int dh\, P(x,h) f'(x,h)^2, \tag{4.52}$$

with respect to x. We use the identity

$$\frac{\partial}{\partial x}\int dh\, P(x,h)\, g(x,h) = \int dh\, P(x,h)\, \Omega(x,h)\, g(x,h), \tag{4.53}$$

where $\Omega(x,h)$ is the differential operator

$$\Omega(x,h) = \frac{\partial}{\partial x} + \frac{\dot{\Delta}}{2}\frac{dq}{dx}\left(\frac{\partial^2}{\partial h^2} + 2x f'(x,h)\frac{\partial}{\partial h}\right). \tag{4.54}$$

Deriving equation (4.52) with respect to x once, *assuming that* $\frac{dq}{dx} \neq 0$ (i.e. x is considered to be in the interval $[x_m, x_M]$) and using Parisi's equation (4.3.2) we have

$$\frac{1}{\lambda^2(q)} = \alpha\ddot{\Delta}(q) \int dh\, P(q,h) f'(q,h)^2 + \alpha\dot{\Delta}^2(q) \int dh\, P(q,h) f''(q,h)^2\ . \tag{4.55}$$

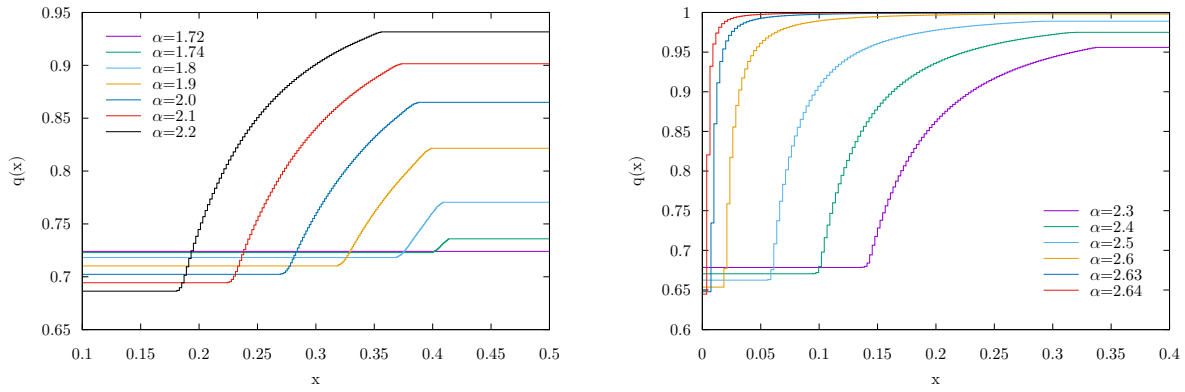

Figure 4.2: Overlap $q(x)$ for the infinite-width tree-committee machine, with ReLU non-linearity near the onset of RSB which happens at $\alpha_{\text{dAT}} \sim 1.7212$ Baldassi et al. [2019] (left panel), and near the critical capacity regime (right panel).

This equation computed at the k-RSB level will give us a prediction of the ansatz instability, i.e. the value of α for which the chosen ansatz does not hold anymore. In the appendix we show how this expression reproduces the de Almeida-Thouless (dAT) instability [de Almeida and Thouless, 1978] when equation (4.55) is evaluated with a Replica Symmetric (RS) ansatz ($q(x) = q$ for any $x \in [0,1]$), and the so-called Gardner transition line [Gardner, 1985] when evaluated using a one-step RSB ansatz.

4.3.4 Breaking point update

From the numerical point of view, even the breaking points x_m and x_M need to be found. An update equation for each one of them can be obtained [Crisanti and Rizzo, 2002] deriving equation (4.55) with respect to x. Again assuming $\frac{dq}{dx} \neq 0$ and solving for x

$$x = \frac{\lambda(x)}{2} \frac{\int dh\, P(x,h) \left[\dddot{\Delta} f'(x,h)^2 + 3\dot{\Delta}\ddot{\Delta} f''(x,h)^2 + \dot{\Delta}^3 f'''(x,h)^2 \right]}{\int dh\, P(x,h) \left[\ddot{\Delta} f'(x,h)^2 + \dot{\Delta}^2 f''(x,h)^2 + \lambda(x)\dot{\Delta}^3 f''(x,h)^3 \right]}, \tag{4.56}$$

which in the case of the identity activation function $\Delta(q) = q$ reduces to [Franz et al., 2017]

$$x = \frac{\lambda(x)}{2} \frac{\int dh\, P(x,h) f'''(x,h)^2}{\int dh\, P(x,h) [f''(x,h)^2 + \lambda(x) f''(x,h)^3]}. \tag{4.57}$$

Once equations (4.3.2), (4.3.2) and (4.49) are solved for a guess of x_m and x_M, they can be updated using equation (4.56); the whole process is iterated until convergence is reached. We refer to the appendix 4.B for an in-depth discussion of the numerical procedure used.

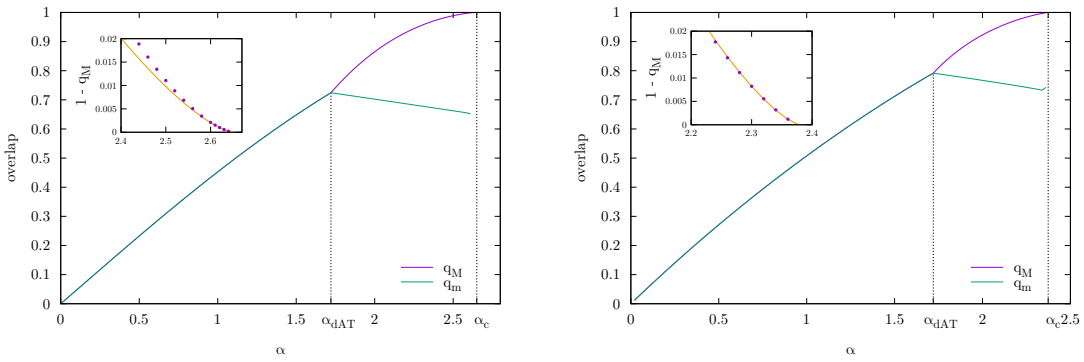

Figure 4.3: Minimum and maximal overlap q_m and q_M as a function of α in the case of the ReLU (left panel) and Erf activation functions (right) with $\kappa = 0$. For $\alpha \leq \alpha_{dAT}$, the RS ansatz is correct so $q_m = q_M$. For $\alpha \to \alpha_c$ we have that $q_M \to 1$. (Inset) We show that q_M scales as a power law, see equation (4.60), with an exponent $\sigma \simeq 1.4157$. Dots are exact numerical solutions, lines are power-law fits.

In Fig. 4.2 we show the resulting plots of $q(x)$ for several values of α starting from the onset of RSB at $\alpha_{\rm dAT}$ in the case of the ReLU activation function $\mathrm{ReLU}(z) = \max(0, z)$.

4.4 Exact determination of the SAT/UNSAT transition

In order to determine the SAT/UNSAT transition, a possible strategy is to perform the $q_M \to 1$ limit inside the fRSB equations. This has been performed in [Charbonneau et al., 2014, Franz et al., 2017], in order to determine the critical exponents of jamming. However the resulting equations are not easy to analyze numerically. Here we adopt another simpler approach that consists in evaluating an observable whose behavior near the SAT/UNSAT transition can be analytically predicted.

This observable is called the *reduced pressure* and it is proportional to the derivative of the free entropy with respect to the margin

$$\tilde{p} = -\frac{1}{\alpha}\frac{\partial \phi}{\partial \kappa}. \tag{4.58}$$

The name "pressure" comes from the fact that when we differentiate the free energy with respect to the volume one gets the pressure: in the sphere packing interpretation of the negative perceptron problem, a variation with respect to κ is indeed equivalent to a change of the particle volume [Franz and Parisi, 2016]. We refer the reader to appendix 4.C for a connection of the reduced pressure to the stability distribution. Reminding that the evolution equations for the

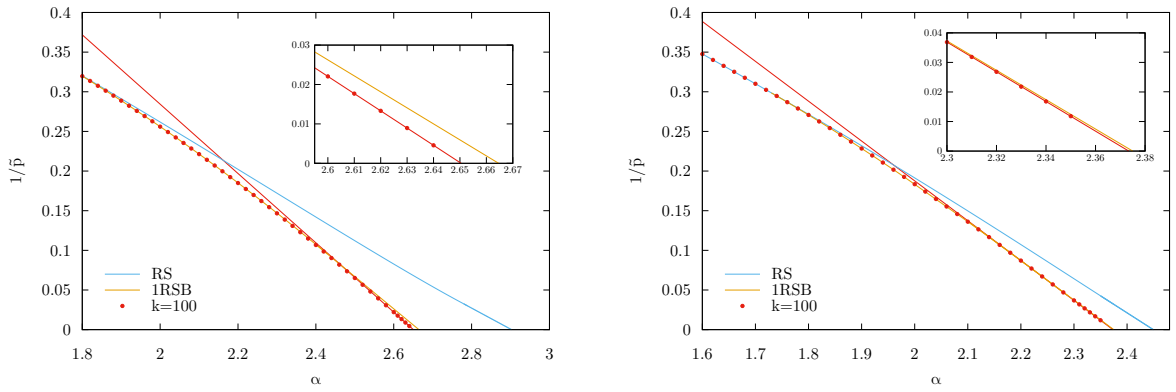

Figure 4.4: Inverse reduced pressure as a function of the constraint density α in the case of the infinite-width tree-committee machine, with ReLU (left panel) and Erf (right) activation functions with $\kappa = 0$. The blue and orange lines represents RS and 1RSB predictions. The red dots represent the solutions obtained by using $k = 100$ steps of RSB. For $\alpha \to \alpha_c$ the inverse reduced pressure scales as $\tilde{p}^{-1} \sim \alpha - \alpha_c$. The red line represents a fit to the k-RSB data near the critical capacity.

functions $\tilde{f}(q_m, h) = f(x_m, -h-\kappa)$ and P are independent on κ, one gets

$$\tilde{p} = -\frac{1}{\alpha}\frac{\partial \phi}{\partial \kappa} = -\int dh\, P(q_m, h) f'(x_m, h) = -\int dh\, P(q_M, h) f'(q_M, h). \tag{4.59}$$

Now we use the (not yet mathematically proven) fact that upon approaching the SAT/UNSAT transition, $\tilde{p}$ scales as [Franz et al., 2017, Parisi et al., 2020]

$$\tilde{p} \propto \frac{1}{\alpha_c - \alpha}. \tag{4.60}$$

We show in Fig. 4.4 a validation of this scaling from the numerical solution of our fRSB equation. We applied this strategy to the non-linear two-layer networks defined in section 4.2 with zero margin, $\kappa = 0$. We show in Fig. 4.4 the inverse reduced pressure as a function of α for the ReLU and Erf activations computed using $k = 100$ steps of RSB; a linear fit to the numerical data is also presented. We show for comparison also the inverse reduced pressure computed at the RS and 1RSB level. In Table (4.1) we summarize our findings for the value of the SAT/UNSAT transition for several activation functions. We also report the constraint density where RSB effects arise and the SAT/UNSAT transition computed in the RS and 1RSB approximations (whose derivations can be found respectively in appendix 4.B.4 and 4.B.5).

	ReLU	Tanh	Erf	Swish
$\alpha_{\rm dAT}$	1.721195	1.7530	1.71995	1.805634
$\alpha_c^{\rm RS}$	$\frac{2\pi}{\pi-1} \simeq 2.934$	2.3556	2.4514	2.42416
$\alpha_c^{\rm 1RSB}$	2.66428	2.306265	2.37499	2.3855699
$\alpha_c^{\rm fRSB}$	2.6504(5)	2.3049(0)	2.3733(5)	2.3838(3)

Table 4.1: dAT and exact SAT/UNSAT transition for some activation functions with $\kappa = 0$. We also show for comparison the SAT/UNSAT transition computed in the RS approximation.

4.5 Gardner phase in the negative perceptron and the no Overlap Gap condition

In this section, we focus on the case of the Negative Perceptron. While in two-layer networks a non-convexity is already present due to the non-linear activation function of the hidden layer, in the case of the perceptron one needs to achieve non-convexity by using a negative margin $\kappa < 0$. We will thus be concerned with the whole (κ, α) phase diagram, while in the previous section we limited ourselves to the $\kappa = 0$ case. In subsection 4.5.1 we remind the full phase diagram of the model, whereas in subsection 4.5.2 we unveil the presence of a line separating two phases, where typical states respectively have or do not have an overlap gap. We refer to appendix 4.B.5 the phase diagram of the tree-committee machine with ReLU activation.

4.5.1 Phase Diagram

Depending on the value of the load α and the margin κ, the model exhibits a variety of phases, the boundaries of which were calculated in [Franz et al., 2017]. In the appendix we sketch how these lines can be estimated, while here we summarise what the phases are, and what type of $q(x)$ we expect in each phase. A plot of the phase diagram is reported in Figure 4.5.

For $\alpha < \alpha_{dAT}$, the RS solution is stable, and we thus expect $q(x)$ to be constant. Increasing α above α_{dAT} we enter different phases depending on the value of κ:

- For $\kappa_{1RSB} < \kappa < 0$, the system goes into a fRSB phase, which we have described above, through a continuous phase transition.

- For $\kappa_{RFOT} < \kappa < \kappa_{1RSB}$ the system passes into a 1RSB phase, always through a continuous phase transition, before entering at larger value of α into a fRSB phase. In the 1RSB phase, $q(x)$ is a stepwise function, with $q(x) = q_0$ for $x < m$ and $q(x) = q_1$ for $x \geq m$.

- For $\kappa < \kappa_{RFOT}$ the system goes into a sequence of phase transitions that are also encountered in infinite-dimensional theories of glasses and that are known as Random First

Order Transitions (RFOT). Firstly, (before RS becomes unstable), for $\alpha_{\text{dyn}} < \alpha < \alpha_K$ the system enters a "*dynamical 1RSB*" phase: although the free energy is equal to that found using an RS ansatz, the equilibrium measure decomposes into an exponential number of pure states. This corresponds to having an 1RSB phase with $m = 1$. Further increasing α above α_K, we cross the *Kauzmann line*, indicating the onset of a 1RSB phase with $m < 1$. Finally, for $\alpha > \alpha_G$ the system enters a *Gardner phase*, where the $q(x)$ exhibits both a 1RSB-like discontinuity at $x = m$, and an fRSB-like continuous part for $x_m \leq x \leq x_M$, with $m \leq x_m$.

4.5.2 Gardner Phase, Overlap Gap and Algorithmic Implications

It is natural to wonder where the boundary between the fRSB and Gardner phase lies, as this has important algorithmic consequences. Indeed, Refs. [Montanari, 0, El Alaoui et al., 2021] analyzed an algorithm called *Incremental AMP* (iAMP) which provably finds a solution in the whole SAT phase, provided that the distribution of overlaps of typical states has connected support. Throughout the paper we called this the *No Overlap Gap* condition (nOG). This property holds in the fRSB phase, however it does not in the Gardner phase (nor in the 1RSB phase). The boundary between these phases could thus act as an algorithmic threshold, at least for iAMP.

Our contribution is thus to give a numerical estimate of this line, which we call α_{1+fRSB}. Rather than looking at the $q(x)$ directly, we use a more precise criterion. Starting in the fRSB phase for a suitable fixed value of κ, we look at the derivative of $q(x)$, which can be calculated analytically in the region $[x_m, x_M]$ (see appendix 4.D). Then we increase the value of α; α^{1+fRSB} corresponds to the first point where $\dot{q}(x_m)$ becomes negative. Solutions with negative derivative are unphysical, so they signal a discontinuity in the function, which corresponds to a gap in the overlap distribution. More details and several plots of $q(x)$ and $\dot{q}(x)$ near the transition to the Gardner phase are reported in Appendix 4.D. Notice that a similar criterion was used in [Franz et al., 2017] to determine the numerical value of κ_{1RSB}.

4.6 Numerical Simulations

In this section, we compare our estimates of the critical capacity with the performance of Gradient Descent (GD), a first-order optimization method which is a variant of the most widely used optimization algorithm for neural networks, Stochastic Gradient Descent (SGD).

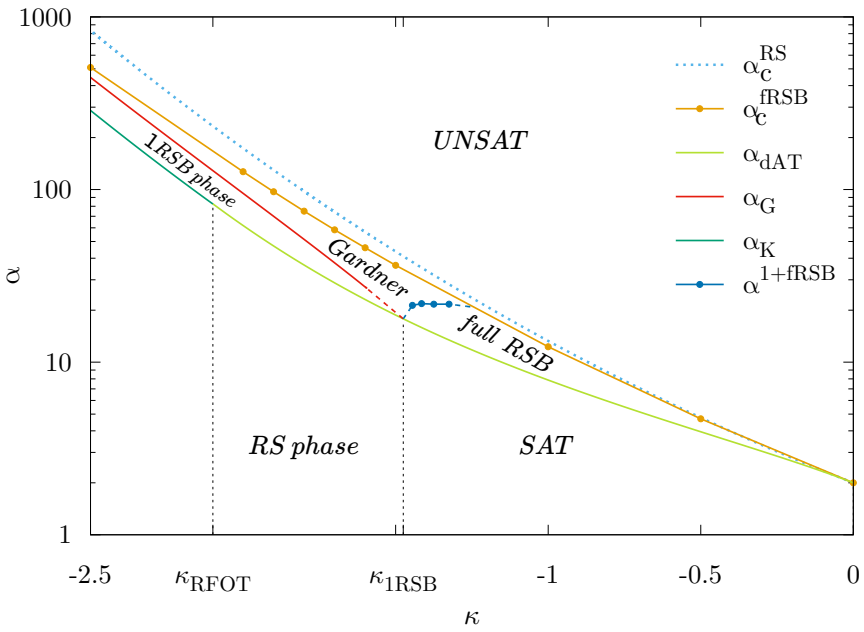

Figure 4.5: Phase Diagram of the Negative Perceptron. The dynamical transition line $\alpha_{dyn}(\kappa)$ that exists for $\kappa < \kappa_{\mathrm{RFOT}}$ is not displayed for clarity reasons, but it can be found in Baldassi et al. [2023]. Dashed lines represent linear interpolations of the Gardner and 1+fRSB transitions to their intersections with the dAT line which happens at $\kappa = \kappa_{1RSB}$. The dotted line represents the critical capacity evaluated with the RS ansatz.

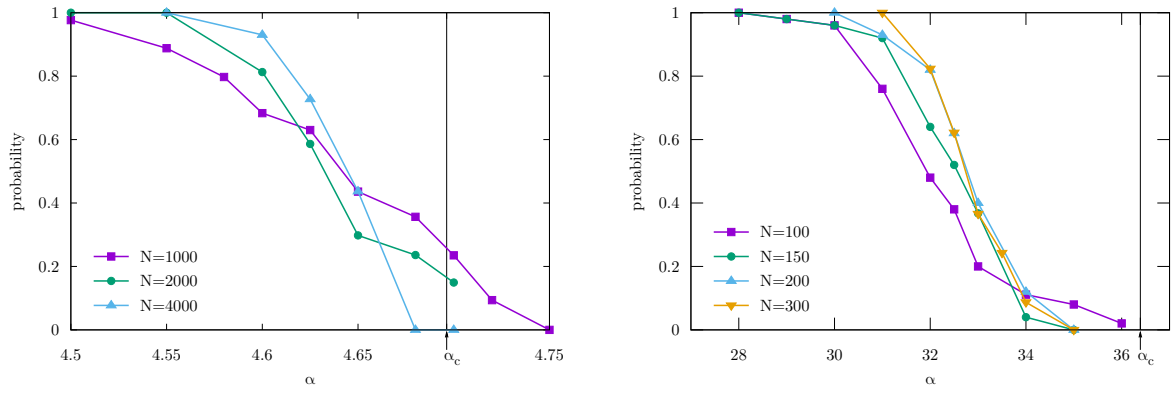

Figure 4.6: Probability of finding solutions using GD on the cross entropy loss (4.62) versus α for the negative perceptron with $\kappa = -0.5$, with sizes $N = 1000, 2000, 4000$ (left panel) and with $\kappa = -1.5$ for $N = 100, 150, 200$ and 300. In the GD simulations we have fixed the learning rate $\eta = 1$ and the maximum number of training epochs to $2 \cdot 10^6$. The vertical black line represents the exact value of the SAT/UNSAT transition.

In order to find a solution using GD we used a (differentiable) loss function $\mathcal{L}(\boldsymbol{w})$

$$\mathcal{L}(\boldsymbol{w}) = \sum_{\mu=1}^{\alpha N} \ell(\Delta^{\mu}(\boldsymbol{w};\kappa)), \tag{4.61}$$

where $\ell(\bullet)$ is a loss function per pattern. Generically $\ell(\bullet)$ is chosen to be small if the stability of each pattern in the training set is large and large otherwise. Commonly used loss functions are

$$\ell(x) = \frac{1}{\gamma}\ln(1+e^{-\gamma x}), \tag{4.62}$$

$$\ell(x) = \frac{x^2}{2}\Theta(-x), \tag{4.63}$$

that are called respectively the *cross entropy* and *quadratic hinge* loss.

A solution is found by running the following iterative scheme

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta\nabla_w \mathcal{L}(\boldsymbol{w}), \tag{4.64}$$

until all constrains $\Delta^{\mu}(\boldsymbol{w};\kappa) \geq 0$ for $\mu = 1,\ldots,P$ are satisfied. In this model particular attention needs to be paid to the norm, since we are studying the set of solutions subject to the constraint given in equation (4.4), and the dynamics given by equation (4.64) will not keep the weights normalized as we want. There are two ways to deal with this:

- Introduce a normalization step after every GD update.
- Keep the norm free to vary, and normalize it when the number of errors is calculated.

When training a tree-committee machine we have empirically observed that the first method leads to a larger probability of finding a solution, while for the negative perceptron the second method works best.

In figure 4.6 we show the probability of finding a solution for a the negative perceptron as a function of α at fixed $\kappa = -0.5, -1.5$ and for several values of N. As we can see, as N increases, the transition between non-solvable and solvable problems becomes sharper. This transition, however, clearly happens at values of α below the critical capacity, thus implying that there is an algorithmic gap. Similar conclusions can be drawn in the tree-committee machine case with ReLU activation functions, see Figure 4.7.

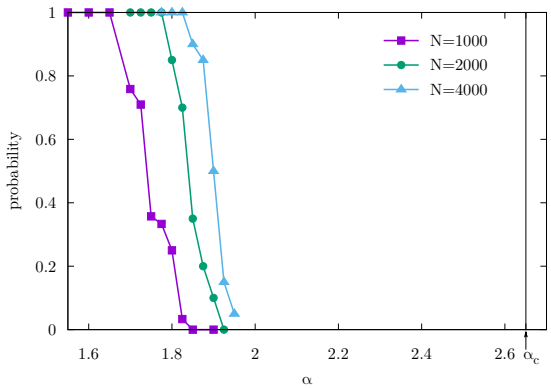

Figure 4.7: Probability of finding solutions using GD on the cross entropy loss (4.62) versus α for the tree-committee machine with a ReLU activation function. Here we have used $K = 100$ and sizes $N = 1000, 2000, 4000$. In the GD simulations we have fixed the learning rate $\eta = 1$ and the maximum number of training epochs to 10^5. The vertical black line represents the exact value of the SAT/UNSAT transition.

4.7 Linear Mode Connectivity in the Presence of fRSB

In this section, we will see how the fRSB calculations outlined in the previous sections can be applied to the problem of the connectivity of solutions of the negative perceptron. Indeed, all results presented in the previous chapter were derived in the *Replica Symmetric* hypothesis, which is known to not hold for all $\alpha > \alpha_{dAT}(\kappa)$. Beyond this threshold, we will have to calculate equation (3.17) using a general KRSB ansatz given by equation (4.35). There is however a subtlety which is not present in the RS case. When we calculate the training error in the general KRSB ansatz for solutions sampled with the same margin k we get

$$E_\gamma = \lim_{n\to 0} \int \prod_{ab} dq_{ab} d\hat{q}_{ab} e^{N\left(\frac{1}{2}\sum_{ab} q_{ab}\hat{q}_{ab} + G_S + \alpha G_E\right)} g(q), \tag{4.65}$$

$$g(q) = \frac{\int \prod_a d\lambda_a d\hat{\lambda}_a e^{i\sum_a \lambda_a\hat{\lambda}_a - \frac{1}{2}\sum_{ab}\hat{\lambda}_a q_{ab}\hat{\lambda}_b} \prod_a \Theta\left(\lambda_a - \kappa_E\right)\Theta\left(-\frac{\sum_\alpha^y \gamma_\alpha\lambda_\alpha}{\sqrt{\sum_{\alpha\beta}^y \gamma_\alpha\gamma_\beta q_{\alpha\beta}}} + k\right)}{\int \prod_a d\lambda_a d\hat{\lambda}_a e^{i\sum_a \lambda_a\hat{\lambda}_a - \frac{1}{2}\sum_{ab}\hat{\lambda}_a q_{ab}\hat{\lambda}_b} \prod_a \Theta\left(\lambda_a - \kappa_E\right)}. \tag{4.66}$$

As we have already mentioned, we can apply laplace's method and say that $E_\gamma = \lim_{n\to 0} g(q^*)$ where q^* is the matrix of overlaps at the saddle point. However for a KRSB ansatz, there will exist many saddle points with the same free energy, given by the various permutations obtained by relabeling of the replicas, which correspond to a permutation of rows and columns in the overlap matrix. To get the correct result then we will have to sum over all these permutations

$$E_\gamma = \frac{1}{n(n-1)} \sum_{\pi \in \Pi} g\left(\pi(q^*)\right), \tag{4.67}$$

where with Π we have indicated the set of all these permutations. In the following we will consider $y = 2$, that is we will look at linear paths between solutions. Since the only term that depends on the specific permutation of q in $g(q)$ is $\sum_{\alpha\beta}^{2} \gamma_\alpha \gamma_\beta q_{\alpha\beta}$, we will consider the resulting expression for the various permutations (with their corresponding multiplicity)

$$\sum_{a,b=1}^{2} \gamma_a \gamma_b q_{ab} = \begin{cases} (\gamma_1^2 + \gamma_2^2) + 2\gamma_1\gamma_2 q_0 = c_0^2, & \text{with multiplicity } n(n-m_1) \\ (\gamma_1^2 + \gamma_2^2) + 2\gamma_1\gamma_2 q_1 = c_1^2, & \text{with multiplicity } n(m_1-m_2) \\ \vdots & \\ (\gamma_1^2 + \gamma_2^2) + 2\gamma_1\gamma_2 q_K = c_K^2 & \text{with multiplicity } n(m_K-1). \end{cases} \tag{4.68}$$

With a slight abuse of notation we will write $\left[\sum_{a,b=1}^{2} \gamma_a \gamma_b q_{ab}\right]_s = c_s^2$, and for each $s \in [K]$ will write the corresponding expression for $\left[E_\gamma\right]_s$, which can be interpreted as the error between states that lie at a distance q_s on the ultrametric tree. Plugging in the KRSB ansatz we get that the solution can be written in terms of the following iterations

$$\xi_K(x,y) = \int_{A(x)}^{C(x,y)} D\lambda_1 \Big(H(A(y)) - H(B(x,y)) \Big), \tag{4.69}$$

$$\xi_{k-1}(x,y) = \int Dz Dz' \Big(\psi_k(x + z\sqrt{q_k - q_{k-1}}) \Big)^{m_k/m_{k+1}-1} \Big(\psi_k(y + z'\sqrt{q_k - q_{k-1}}) \Big)^{m_k/m_{k+1}-1} \times \tag{4.70}$$

$$\times\, \xi_k(x + z\sqrt{q_k - q_{k-1}}, y + z'\sqrt{q_k - q_{k-1}}), \qquad k = K, \ldots, l+1$$

$$\xi_l(y) = \int Dz Dz' \Big(\psi_{l+1}(x + z\sqrt{q_{l+1} - q_l}) \Big)^{m_{l+1}/m_{l+2}-1} \Big(\psi_{l+1}(y + z'\sqrt{q_{l+1} - q_l}) \Big)^{m_{l+1}/m_{l+2}-1} \tag{4.71}$$

$$\times\, \xi_{l+1}(y + z\sqrt{q_{l+1} - q_l}, y + z'\sqrt{q_{l+1} - q_l}),$$

$$\xi_{k-1}(y) = \int Dz \Big(\psi_k(y + z\sqrt{q_k - q_{k-1}}) \Big)^{m_k/m_{k+1}-1} \xi_k(y + z\sqrt{q_k - q_{k-1}}), \qquad k = l, \ldots, 1 \tag{4.72}$$

$$\left[E_\gamma\right]_l = \Theta\Big(\kappa_E c_l - k\Big)(m_{l+1} - m_l) \int Dz_0 \Big(\psi_0(\sqrt{q_0} z_0) \Big)^{-1} \xi_0(\sqrt{q_0} z_0), \tag{4.73}$$

where

$$A(x) = \frac{k-x}{\sqrt{1-q_K}}, \tag{4.74}$$

$$B(x,y) = \frac{-\sqrt{1-q_K}\gamma_1\lambda_1 - (\gamma_1 x + \gamma_2 y) + \kappa_E c_l}{\sqrt{1-q_K}\gamma_2}, \tag{4.75}$$

$$C(x,y) = \frac{-(\gamma_1 x + \gamma_2 y) + \kappa_E c_l + \gamma_2 y - \gamma_2 k}{\sqrt{1-q_K}\gamma_1}. \tag{4.76}$$

In the $K \to \infty$ limit we obtain the fRSB expression for the training error along the path $E_\gamma(x)$, and by differentiating the inverse function $x\left(E_\gamma\right)$ we can obtain the probability distribution $P\left(E_\gamma\right)$. As we can see, the quantity E_γ is not self averaging anymore, and thus the definition of the coalescence thresholds κ_y^* given in the previous chapter loses meaning. However we can define the upper and lower limits of $\kappa_y^{*,\text{up}}$ and $\kappa_y^{*,\text{lo}}$, such that for $\kappa > \kappa_y^{*,\text{up}}$ the $(y-1)$-dimensional simplex of solutions sampled with margin k is at zero energy, and $\kappa < \kappa_y^{*,\text{lo}}$ the simplex at energy greater than zero. To compute these bounds for the $y = 2$ case we can leverage the fact that, as in the RS case, the expression for $\left[E_\gamma\right]_l$ is multiplied by a heaviside theta $\Theta\Big(\kappa_E c_l - k\Big)$ which easily tells us when the error will be zero. Using the fact that the c_l will be ordered, and the fact that the error will be maximum for $\gamma_1 = \gamma_2 = \frac{1}{2}$, then we can say that $\kappa_y^{*,\text{up}}$ is the value of k for which $\kappa_E c_0 - k = 0$, and $\kappa_y^{*,\text{lo}}$ the value for which $\kappa_E c_K - k = 0$. As we did for κ_∞^*, we can consider the limit $y \to \infty$ and calculate upper and lower thresholds $\kappa_\infty^{*,\text{up}}$ and $\kappa_\infty^{*,\text{lo}}$, that will be given by the values for which $\kappa_E\sqrt{q_0} - k = 0$ and $\kappa_E\sqrt{q_K} - k = 0$ respectively.

In figure 4.8 we show the values of these upper and lower bounds for $y = 2$ and $y = \infty$, along with the critical capacity derived in the 1RSB ansatz (which is close enough to the fRSB estimate). As we can see the upper limit intersects the capacity in points α_2^d and α_∞^d. Intuitively, this means that for $\alpha > \alpha_2^d$ solutions which are surely linear mode connected stop existing, and for $\alpha > \alpha_\infty^d$ solutions which surely form a convex subspace stop existing. This highlights the gradual breaking down of the connectivity between solutions as the critical capacity is approached.

4.8 Conclusions

In the present chapter we studied the storage problem for two prototypical neural network models, the *Negative Perceptron* and the *Tree-Committee Machine*. Using the replica method, we determined the saddle-point equations that the order parameters need to satisfy, for arbitrary

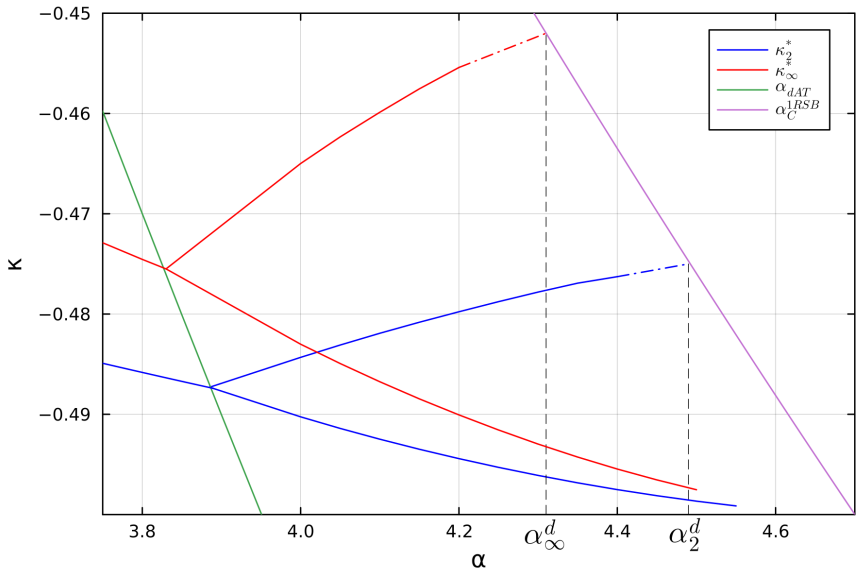

Figure 4.8: Upper and lower branches of κ_2^* and κ_∞^* in the fRSB phase. Here $\kappa_E = -0.5$, and the green and purple lines are the dAT line and critical capacity in the 1RSB ansatz respectively

(negative) margin κ for the first and for arbitrary activation function φ for the latter. Focusing on the *Full-RSB* region of the phase space, we solved these equations numerically using a k-RSB ansatz with large k, and used the solutions to compute several observables. By performing a linear fit of the inverse reduced pressure near the SAT/UNSAT threshold we were able to give a high precision numerical estimate of this transition.

For the negative perceptron we determined another novel phase transition between a *fRSB* and *Gardner* phase, and gave a numerical estimate of the value of this threshold. We discussed the *no Overlap Gap condition*, according to which the support of the distribution $P(q)$ of typical states is connected, and identified the boundaries of validity of this property in the phase diagram. The authors of [El Alaoui and Sellke, 2022] recently proposed an algorithm, *iAMP*, which provably finds solutions under the *nOG* hypothesis. We have showed that this hypothesis does not hold in the *Gardner* phase. This could indicate that this transition acts as an algorithmic threshold for this model.

We also compared our estimates of the SAT-UNSAT threshold with the performance of *Gradient Descent*. In all cases analyzed we have given evidence that Gradient Descent stops finding solutions before the exact SAT/UNSAT threshold that we computed, thus suggesting the presence of an algorithmic gap.

Finally, we used the kRSB ansatz to analyze the disconnection transition in the negative perceptron. We found out that perhaps unsurprisingly, close to the SAT-UNSAT transition surely connected solutions gradually stop existing. It would be interesting to study also how the

kernel transition κ_{krn} evolves beyond the dAT line, as it would give information about the disconnection of the star-shape.

Appendix

4.A Properties of k-RSB and fRSB matrices

4.A.1 Eigenvalues

We derive the eigenvalues of a fRSB matrix by iteration starting from the RS case and moving to the 1 and 2RSB case. For the sake of generality we will suppose the matrix q^{ab} is parameterized by the value it attains on its diagonal q_d and the (step) functions corresponding to values out of the diagonal: $q^{ab} \to \{q_d, q(x)\}$.

- A RS matrix can be decomposed as a sum of two matrices

$$q^{ab} = (q_d - q)\delta_{ab} + q, \tag{4.77}$$

that commute between each other, so they can be simultaneously diagonalized. An $n \times n$ matrix with all elements equal to q has $n-1$ degenerate zero eigenvalues and one eigenvalue equal to nq. We therefore get two eigenvalues

$$\lambda_{-1} = q_d - q + nq\,, \qquad d_{-1} = 1 \tag{4.78}$$

$$\lambda_0 = q_d - q\,, \qquad d_0 = n-1 \tag{4.79}$$

- A 1RSB matrix can be expressed as the sum of 3 terms

$$q^{ab} = (q_d - q_1)\delta_{ab} + (q_1 - q_0)I_{ab}^{m_0} + q_0, \tag{4.80}$$

where $I_{ab}^{m_0}$ is the $n \times n$ matrix having elements equal to 1 inside the blocks of size m_0 located around the diagonal and 0 otherwise. Again all the three matrices commute with each other and can be simultaneously diagonalized. Each of the n/m_0 blocks of $I_{ab}^{m_0}$ has all equal elements equal to 1, therefore it has $\frac{n}{m_0}(m_0 - 1)$ eigenvalues equal to 0 and

$\frac{n}{m_0}$ equal to m_0. We therefore have the following eigenvalues

$$\lambda_{-1} = q_d - q_1 + m_0(q_1 - q_0) + nq_0\,, \qquad d_{-1} = 1 \tag{4.81}$$

$$\lambda_0 = q_d - q_1 + m_0(q_1 - q_0)\,, \qquad d_0 = \frac{n}{m_0} - 1 = n\left(\frac{1}{m_0} - \frac{1}{n}\right) \tag{4.82}$$

$$\lambda_1 = q_d - q_1 \qquad d_1 = \frac{n}{m_0}(m_0 - 1) = n\left(1 - \frac{1}{m_0}\right). \tag{4.83}$$

- A 2RSB matrix is decomposed as

$$q^{ab} = (q_d - q_2)\delta_{ab} + (q_2 - q_1)I_{ab}^{m_1} + (q_1 - q_0)I_{ab}^{m_0} + q_0, \tag{4.84}$$

repeating the same argument as above we have

$$\lambda_{-1} = q_d - q_2 + m_1(q_2 - q_1) + m_0(q_1 - q_0) + nq_0\,, \tag{4.85}$$

$$\lambda_0 = q_d - q_2 + m_1(q_2 - q_1) + m_0(q_1 - q_0)\,, \tag{4.86}$$

$$\lambda_1 = q_d - q_2 + m_2(q_2 - q_1)\,, \tag{4.87}$$

$$\lambda_2 = q_d - q_2\,, \tag{4.88}$$

with degeneracies respectively

$$d_{-1} = 1, \tag{4.89}$$

$$d_0 = \frac{n}{m_0} - 1 = n\left(\frac{1}{m_0} - \frac{1}{n}\right), \tag{4.90}$$

$$d_1 = \frac{n}{m_0}(m_0 - 1) - \frac{n}{m_1}(m_1 - 1) = n\left(\frac{1}{m_1} - \frac{1}{m_0}\right), \tag{4.91}$$

$$d_2 = n\left(1 - \frac{1}{m_1}\right). \tag{4.92}$$

- Generalizing to a k-RSB matrix we have

$$\lambda_s = \sum_{i=s}^{k} m_i(q_{i+1} - q_i)\,, \qquad d_s = n\left(\frac{1}{m_s} - \frac{1}{m_{s-1}}\right), \qquad s = -1\,, \ldots\,, k \tag{4.93}$$

where we have defined $m_k = 1$, $q_{k+1} = q_d$ and $m_{-1} = n \to 0$, $q_{-1} = 0$, $m_{-2} = \infty$. Notice also that in the small n limit $\lambda_{-1} = \lambda_0$.

In the continuous limit the eigenvalues become a function of x:

$$\lambda(x) = \int_{q(x)}^{1} dq' x(q') = q_d - xq(x) - \int_x^1 dy\, q(y). \tag{4.94}$$

As in the previous sections, we will denote by λ_m and λ_M the values of λ corresponding to the minimum q_m and a maximum q_M value of the overlap, i.e.

$$\lambda_m = q_d - \int_0^1 dx\, q(x), \tag{4.95}$$

$$\lambda_M = q_d - q_M. \tag{4.96}$$

4.A.2 Inverse

Since k-RSB matrices form a group, the inverse element $p_{ab} = (q^{-1})_{ab}$ must be an element of the group. Therefore the functional form of the eigenvalues is the same as the one derived before. Moreover, we know that the eigenvalues are simply $1/\lambda_s$ with $s = 0, \dots, k$. We therefore have

$$\sum_{i=s}^{k} m_i (p_{i+1} - p_i) = \frac{1}{\sum_{i=s}^{k} m_i (q_{i+1} - q_i)}. \tag{4.97}$$

Those are $k+1$ equations in $k+1$ unknowns. They can be solved iteratively; first of all taking the $i = k$ index we get

$$p_d = p_k + \frac{1}{q_d - q_k}. \tag{4.98}$$

By subtracting the $(s-1)$-th and the s-th equations we get the recursion

$$p_s = p_{s-1} + \frac{1}{m_{s-1}} \left[\frac{1}{\sum_{i=s-1}^{k} m_i (q_{i+1} - q_i)} - \frac{1}{\sum_{i=s}^{k} m_s (q_{i+1} - q_i)} \right] \tag{4.99}$$

$$= p_{s-1} + \frac{1}{m_{s-1}} \left(\frac{1}{\lambda_{s-1}} - \frac{1}{\lambda_s} \right) = p_{s-1} - \frac{q_s - q_{s-1}}{\lambda_{s-1}\lambda_s}, \qquad s = 0, \dots, k \tag{4.100}$$

Iterating we get that the inverse of a k-RSB matrix elements are given by

$$p_s = -\frac{q_0}{\lambda_0^2} - \sum_{i=1}^{s} \frac{q_i - q_{i-1}}{\lambda_{i-1}\lambda_i} \tag{4.101}$$

$$p_d = \frac{1}{q_d - q_k} - \frac{q_0}{\lambda_0^2} - \sum_{i=1}^{k} \frac{q_i - q_{i-1}}{\lambda_{i-1}\lambda_i} \tag{4.102}$$

In the $k \to \infty$ limit we therefore get

$$\lim_{k\to\infty} p_s = p(x) = -\frac{q_m}{\lambda_m^2} - \int_0^x dx\, \frac{\dot{q}(s)}{\lambda^2(s)} = -\frac{q_m}{\lambda_m^2} - \int_{q_m}^q \frac{dq'}{\lambda^2(q')}, \tag{4.103}$$

$$\lim_{k\to\infty} p_d = \frac{1}{q_d - q_M} - \frac{q_m}{\lambda_m^2} - \int_0^1 dx\, \frac{\dot{q}(s)}{\lambda^2(s)} = \frac{1}{\lambda_M} - \frac{q_m}{\lambda_m^2} - \int_0^1 \frac{dq'}{\lambda^2(q')}. \tag{4.104}$$

Notice how the right hand side of the first equation above is equivalent to the left hand side of the saddle point equation (4.49).

4.A.3 Log of the determinant

Having computed the eigenvalues of a generic k-RSB matrix with diagonal elements q_d, we are now ready to compute the log of the determinant, which appears in the entropic term, see for example (4.131). We are interested as usual in the limit $n \to 0$. We have

$$\lim_{n\to 0} \frac{1}{n} \ln\det \boldsymbol{q} = \lim_{n\to 0} \frac{1}{n} \sum_{i=-1}^{k} d_i \ln \lambda_i = \lim_{n\to 0} \left[\sum_{i=-1}^{k} \frac{1}{m_i} \ln \lambda_i - \sum_{i=-1}^{k-1} \frac{1}{m_i} \ln \lambda_{i+1} \right] = \tag{4.105}$$

$$= \ln(q_d - q_k) + \frac{q_0}{\lambda_0} + \sum_{i=0}^{k-1} \frac{1}{m_i} \ln \frac{\lambda_i}{\lambda_{i+1}} = \tag{4.106}$$

$$= \ln(q_d - q_k) + \frac{q_0}{\lambda_0} + \sum_{i=0}^{k-1} \frac{1}{m_i} \ln\left(1 + \frac{m_i(q_{i+1} - q_i)}{\lambda_{i+1}}\right). \tag{4.107}$$

When k is large $q_i - q_{i-1}$ is small, so that, in the continuous limit, we get

$$\lim_{k\to\infty} \lim_{n\to 0} \frac{1}{n} \ln\det \boldsymbol{q} = \ln(q_d - q_M) + \frac{q_m}{\lambda_m} + \int_{x_m}^{x_M} dx\, \frac{\dot{q}(x)}{\lambda(x)}. \tag{4.108}$$

4.A.4 Asymptotic Behaviour of $f(m_l, h)$

We start from the recursion relation in the case of the number of error loss function

$$f(m_k, h) = f(x_M, h) = \ln \int dz\, \mathcal{N}_{\Delta(1)-\Delta(q_k)}(z+h)\, e^{-\beta\ell(z-\kappa)} = \ln H\left(\frac{\kappa + h}{\sqrt{\Delta(1) - \Delta(q_k)}}\right), \tag{4.109}$$

$$f(m_s, h) = \frac{1}{m_s} \ln \int dz\, \mathcal{N}_{\Delta(q_{s+1})-\Delta(q_s)}(z-h) e^{m_s f(m_{s+1}, z)} \qquad s = k-1, \ldots, 0. \tag{4.110}$$

We know that

$$\ln H(x) \simeq -\frac{x^2}{2} - \ln x - \frac{1}{2}\ln(2\pi), \qquad \text{as } x \to +\infty \tag{4.111}$$

whereas it goes exponentially to 0 as $x \to -\infty$. Therefore

$$\ln H\left(\frac{\kappa+h}{\sqrt{\Delta(1)-\Delta(q_k)}}\right) = \begin{cases} -\frac{(\kappa+h)^2}{2(\Delta(1)-\Delta(q_k))} \equiv -\frac{(\kappa+h)^2}{2\Lambda_k}, & h \to +\infty \\ O(e^{-h^2}), & h \to -\infty \end{cases} \tag{4.112}$$

where $\Lambda_k \equiv \Delta(1)-\Delta(q_k)$. Similarly we will define the quantities

$$\Lambda_s = \sum_{i=s}^{k} m_i(\Delta(q_{i+1})-\Delta(q_i)) = \Lambda_{s+1} + m_s(\Delta(q_{s+1})-\Delta(q_s)), \qquad s=-1,\dots,k \tag{4.113}$$

which will appear naturally in the following, and which represent the eigenvalues of the effective order parameter matrix Δ_{ab}.

The asymptotic behavior of $f(m_k,h)$ at at $h \to \pm\infty$ will induce a similar one for the functions $f(m_s,h)$ with $s = k-1,\dots,0$. Let's start with the case $s = k-1$. We have for $h \to \infty$

$$f(m_{k-1},h) = \frac{1}{m_{k-1}} \ln \int dz\, \mathcal{N}_{\Delta(q_k)-\Delta(q_{k-1})}(z)\, e^{m_{k-1} f(m_k, z+h)} \tag{4.114}$$

$$= \frac{1}{m_{k-1}} \ln \int_{-h}^{\infty} dz\, \mathcal{N}_{\Delta(q_k)-\Delta(q_{k-1})}(z)\, e^{-\frac{m_{k-1}(\kappa+z+h)^2}{2(\Delta(1)-\Delta(q_k))}} \tag{4.115}$$

$$\simeq \frac{1}{m_{k-1}} \ln \int_{-h}^{\infty} dz\; e^{-\frac{z^2}{2(\Delta(q_k)-\Delta(q_{k-1}))} - \frac{m_{k-1}(\kappa+z+h)^2}{2(\Delta(1)-\Delta(q_k))}} \tag{4.116}$$

$$\simeq -\frac{(\kappa+h)^2}{2(\Delta(1)-\Delta(q_k))} + \frac{1}{m_{k-1}} \ln \int_{-h}^{\infty} dz\; e^{-a_k z^2 - b_k z}, \tag{4.117}$$

where we have neglected low order terms in h and defined the quantities

$$a_k \equiv \frac{1}{2}\left(\frac{m_{k-1}}{\Delta(1)-\Delta(q_k)} + \frac{1}{\Delta(q_k)-\Delta(q_{k-1})}\right) = \frac{m_{k-1}\Lambda_{k-1}}{2\Lambda_k(\Lambda_{k-1}-\Lambda_k)}, \tag{4.118}$$

$$b_k \equiv \frac{m_{k-1}(\kappa+h)}{\Delta(1)-\Delta(q_k)} = \frac{m_{k-1}(\kappa+h)}{\Lambda_k}. \tag{4.119}$$

Using the identity

$$\int_{\gamma}^{+\infty} dz\, e^{-\alpha z^2-\beta z} = \sqrt{\frac{\pi}{\alpha}}\, e^{\frac{\beta^2}{4\alpha}} H\left(\frac{\beta+2\alpha\gamma}{\sqrt{2\alpha}}\right), \tag{4.120}$$

and noticing that the argument of the H function $b_k - 2a_k h = -\frac{h}{q_k - q_{k-1}} \to -\infty$ we have

$$f(m_k, h) \simeq -\frac{(\kappa+h)^2}{2(\Delta(1)-\Delta(q_k))} + \frac{1}{m_{k-1}} \ln\left[e^{\frac{b_k^2}{4a_k}} H\left(\frac{b_k - 2a_k h}{\sqrt{2a_k}} \right) \right] = \tag{4.121}$$

$$= -\frac{(\kappa+h)^2}{2(\Delta(1)-\Delta(q_k))} + \frac{b_k^2}{4m_{k-1}a_k} = \tag{4.122}$$

$$= -\frac{(\kappa+h)^2}{2\Lambda_k} + \frac{m_{k-1}(\kappa+h)^2(\Lambda_{k-1}-\Lambda_k)}{2\Lambda_k\Lambda_{k-1}} = \tag{4.123}$$

$$- \frac{(\kappa+h)^2}{2\left(\Delta(1)-\Delta(q_k)+m_{k-1}(\Delta(q_k)-\Delta(q_{k-1}))\right)} = \tag{4.124}$$

$$\equiv -\frac{(\kappa+h)^2}{2\Lambda_k}. \tag{4.125}$$

Iterating we get, for $s = 0, \dots, k$

$$f(m_s, h) \simeq -\frac{(\kappa+h)^2}{2\Lambda_s}, \qquad \text{as } h \to +\infty \tag{4.126}$$

Notice that since $\Delta(q_{s+1}) \geq \Delta(q_s)$, then $\Lambda_s \geq \Lambda_{s+1}$ for all $s = 0, \dots, k$; this tells us that $f(m_s, h)$ diverges slower to $-\infty$ for $h \to +\infty$ with respect to $f(m_{s+1}, h)$. Similarly one finds that

$$f(m_s, h) \simeq O(e^{-h^2}) \qquad \text{as } h \to -\infty. \tag{4.127}$$

4.B k-steps Replica Symmetry Breaking ansatz

In this first appendix we derive the expressions of the entropic and energetic term for finite number of breakings of Replica Symmetry [Parisi, 1979a,b, 1980a], and we mention how we have solved the corresponding saddle point equations. We remind that we call the $k+1$ values assumed by the matrix q^{ab} as q_0, q_1, …, q_k, and the block sizes respectively as m_0, m_1, …, m_{k-1}. We will use the square bracket notation $[\bullet]_s$ to denote the operation of extracting step $s+1$ from the k-step RSB matrix in its argument, i.e., for example, $\left[q^{ab}\right]_s = q_s$.

4.B.1 Entropic potential

Imposing the k-RSB structure on the overlap matrix q^{ab} will enable us to perform the small n limit

$$\phi = \max_{\boldsymbol{q}} \mathcal{S}(\boldsymbol{q}), \tag{4.128}$$

$$\mathcal{S}(\boldsymbol{q}) = \mathcal{G}_S(\boldsymbol{q}) + \alpha \mathcal{G}_E(\boldsymbol{q}) \equiv \lim_{n\to 0} \frac{G_S(\boldsymbol{q})}{n} + \alpha \lim_{n\to 0} \frac{G_E(\boldsymbol{q})}{n}. \tag{4.129}$$

In order to compute the entropic term $\mathcal{G}_S(\boldsymbol{q})$ one needs to compute the eigenvalues of a generic k-RSB matrix and the corresponding multiplicities. In appendix 4.A.1 we show that there are $k+2$ eigenvalues λ_s with multiplicities d_s, $s = -1, 0, \dots, k$ which read

$$\lambda_s = \sum_{i=s}^{k} m_i (q_{i+1} - q_i), \qquad d_s = n\left(\frac{1}{m_s} - \frac{1}{m_{s-1}}\right), \qquad s = -1, \dots, k \tag{4.130}$$

In the previous equations we have used the definitions $m_k = 1$, $q_{k+1} = 1$ and $m_{-1} = n$, $q_{-1} = 0$, $m_{-2} = \infty$. Once the eigenvalues are known, one can compute the entropic term, which consists in computing the log of the determinant of $\boldsymbol{q}$ in the small n limit. We show in appendix 4.A.3 that it reads

$$\mathcal{G}_S(\boldsymbol{q}) = \lim_{n\to 0} \frac{1}{2n} \ln\det \boldsymbol{q} = \frac{1}{2}\ln(1-q_k) + \frac{q_0}{2\lambda_0} + \sum_{i=0}^{k-1} \frac{1}{2m_i} \ln\left(1 + \frac{m_i(q_{i+1}-q_i)}{\lambda_{i+1}}\right). \tag{4.131}$$

4.B.2 Infinite width energetic potential

Effective order parameters and entropy

If we impose a k-RSB ansatz on q^{ab}, also the effective order parameters Δ_{ab}

$$\Delta_{ab} = e^{\frac{1}{2}\sum_{ab} q^{ab} \frac{\partial^2}{\partial s_a \partial s_b}} \varphi(s_a)\varphi(s_b)\Big|_{s_a=0}, \tag{4.132}$$

will be a k-steps RSB matrix with the same block size as q^{ab}. It is easy to show that the $s+1$ step of Δ_{ab} is

$$\left[\Delta^{ab}\right]_s = \int Dz_0 \dots Dz_s \left[\int Dz_{s+1} \dots Dz_{k+1}\, \varphi\left(\sum_{l=0}^{k+1} \sqrt{q_l - q_{l-1}} z_l\right)\right]^2 = \tag{4.133}$$

$$= \int Dx \left[\int Dy\, \varphi\left(\sqrt{q_s}\, x + \sqrt{1-q_s}\, y\right)\right]^2 \equiv \Delta(q_s), \tag{4.134}$$

i.e. it depends on q_s only. We have used in the second line the fact that the sum of Gaussian random variables is still Gaussian distributed (or equivalently, we have performed several 2-dimensional rotations over the variables $z_0 \dots z_s$ and $z_{s+1}, \dots, z_{k+1}$). Notice that the previous expression can also be written as a two dimensional Gaussian integral

$$\Delta(q_s) = \int \frac{d\boldsymbol{h}}{2\pi\sqrt{\det \boldsymbol{\mathcal{C}}}}\, e^{-\frac{1}{2}\boldsymbol{h}^T \boldsymbol{\mathcal{C}}^{-1}\boldsymbol{h}}\, \varphi(h_1)\,\varphi(h_2), \tag{4.135}$$

where

$$\boldsymbol{\mathcal{C}} = \begin{pmatrix} 1 & q_s \\ q_s & 1 \end{pmatrix}, \tag{4.136}$$

therefore showing that our effective order parameters are also equivalent to the NNGP kernel appearing in neural networks at initialization or in the lazy regime. In the following we will also need the indices $l = -1$ and $l = k+1$ in order to write down the expression of the entropy; consistently with the notation $q_{-1} \equiv 0$ and $q_{k+1} \equiv 1$, they can be found by substituting them in (4.133), i.e.

$$\Delta(q_{-1}) = \Delta(0) = \left[\int Dx\, \varphi(x)\right]^2, \tag{4.137}$$

$$\Delta(q_{k+1}) = \Delta(1) = \int Dx\, \varphi^2(x)\,. \tag{4.138}$$

Given those definitions the energetic term reads

$$\mathcal{G}_E = \frac{1}{m_0}\int Dz_0 \ln \int Dz_1 \left[\dots \left[\int Dz_{k+1}\, e^{-\beta\ell\left(\sqrt{\Delta(1)-\Delta(q_k)}z_{k+1} - \sum_{s=0}^{k}\sqrt{\Delta(q_s)-\Delta(q_{s-1})}z_s - \kappa\right)} \right]^{\frac{m_{k-1}}{m_k}} \dots \right]^{\frac{m_0}{m_1}}. \tag{4.139}$$

The energetic term can be written more compactly defining a discrete set of functions $f(m_s, h)$, with $s = 0, \dots, k$, that satisfy the iterative rule

$$f(m_k, h) = \ln \int dz\, \mathcal{N}_{\Delta(1)-\Delta(q_k)}(z+h)\, e^{-\beta\ell(z-\kappa)} \tag{4.140}$$

$$f(m_s, h) = \frac{1}{m_s}\ln \int dz\, \mathcal{N}_{\Delta(q_{s+1})-\Delta(q_s)}(z-h) e^{m_s f(m_{s+1}, z)}, \qquad s = k-1, \dots, 0 \tag{4.141}$$

where $\mathcal{N}_\sigma(z) \equiv \frac{e^{-\frac{z^2}{2\sigma}}}{\sqrt{2\pi\sigma}}$. Notice how the iteration rule for $\tilde{f}(m_0, h) \equiv f(m_0, -h-\kappa)$ does not explicitly depend on κ (this the function that is actually used in [Franz et al., 2017]). Notice

that in error counting loss, which we focus on in this paper, $\ell(x) = \Theta(-x)$ the integral in the initial condition for f can be explicitly solved, giving

$$f(m_k, h) = \ln H_\beta\left(\frac{\kappa + h}{\sqrt{\Delta(1) - \Delta(q_k)}}\right), \tag{4.142}$$

where $H_\beta(x) \equiv e^{-\beta} + (1 - e^{-\beta})H(x)$ and $H(x) \equiv \int_x^\infty Dy = \frac{1}{2}\mathrm{Erfc}\left(\frac{x}{\sqrt{2}}\right)$. The energetic term therefore can be expressed in terms of $f(m_0, h)$ as

$$\mathcal{G}_E = \int dh \mathcal{N}_{\Delta(q_0)-\Delta(0)}(h) f(m_0, h). \tag{4.143}$$

Effective order parameters for some activation functions

We list here the expressions of the effective order parameters for some activation functions of interest

- $\varphi(x) = x$, in the case of the identity activation we get back the perceptron case

$$\Delta(q) = q. \tag{4.144}$$

- $\varphi(x) = \mathrm{sign}(x)$ [Barkai et al., 1990, 1992, Engel et al., 1992]

$$\Delta(q) = 1 - \frac{2}{\pi}\arccos(q). \tag{4.145}$$

- $\varphi(x) = \mathrm{ReLU}(x) = \max(0, x)$

$$\Delta(q) = \frac{\sqrt{1-q^2}}{2\pi} + \frac{q}{\pi}\arctan\left(\sqrt{\frac{1+q}{1-q}}\right). \tag{4.146}$$

- $\varphi(x) = \mathrm{Erf}(\gamma x)$

$$\Delta(q) = 1 - \frac{2}{\pi}\arccos\left(\frac{2\gamma q}{1+2\gamma}\right). \tag{4.147}$$

Alternative approach

One can find (4.139) directly imposing the k-RSB ansatz on finite width version of the energetic term, which reads

$$\mathcal{G}_E = \frac{1}{m_0}\mathbb{E}_y \int \prod_l Dz_l^0 \times \tag{4.148}$$
$$\times \ln \int \prod_l Dz_l^1 \left[\dots \left[\int \prod_l Dz_l^{k+1} e^{-\beta \ell\left(\frac{y}{\sqrt{K}} \sum_{l=1}^K c_l \, \varphi\left(\sum_{s=0}^{k+1} \sqrt{q_s - q_{s-1}} z_l^s \right) - \kappa \right)} \right]^{\frac{m_{k-1}}{m_k}} \dots \right]^{\frac{m_0}{m_1}}.$$

We can now use the central limit theorem repeatedly on this expression to perform the large K limit. We specialize here for simplicity to the number of error loss with $\beta \to \infty$, but the argument can be trivially generalized to generic loss functions. The innermost K-dimensional integrals can be simplified as

$$\int \prod_l Dz_l^{k+1} \Theta\left(\frac{y}{\sqrt{K}} \sum_{l=1}^K c_l \, \varphi\left(\sum_{s=0}^{k+1} \sqrt{q_s - q_{s-1}} z_l^s \right) - \kappa \right) \simeq \tag{4.149}$$
$$\simeq \int Dz^{k+1} \Theta\left(y M^{(0)} + \sqrt{\Delta^{(0)}} z^{k+1} - \kappa \right) = H\left(\frac{\kappa + y M^{(0)}}{\sqrt{\Delta^{(0)}}} \right), \tag{4.150}$$

where $M^{(0)}$ and $\Delta^{(0)}$ are respectively the mean and the variance with respect to variables z^{k+1}

$$M^{(0)} \equiv \frac{1}{\sqrt{K}} \sum_{l=1}^K c_l \int Dh \, \varphi\left(\sum_{s=0}^k \sqrt{q_s - q_{s-1}} z_l^s + \sqrt{1 - q_k} h \right), \tag{4.151}$$
$$\Delta^{(0)} \equiv \frac{1}{K} \sum_{l=1}^K c_l^2 \left[\int Dh \, \varphi^2 \left(\sum_{s=0}^k \sqrt{q_s - q_{s-1}} z_l^s + \sqrt{1 - q_k} h \right) \right. \tag{4.152}$$
$$\left. - \left(\int Dh \, \varphi\left(\sum_{s=0}^k \sqrt{q_s - q_{s-1}} z_l^s + \sqrt{1 - q_k} h \right) \right)^2 \right].$$

Iterating the procedure k times we have

$$\mathcal{G}_E \equiv = \frac{1}{m_0}\mathbb{E}_y \int Dz_0 \ln \int Dz_1 \times \tag{4.153}$$
$$\times \left[\dots \left[\int Dz_k \, H^{m_{k-1}} \left(\frac{\kappa + y M + \sum_{s=0}^k \sqrt{\Delta(q_s) - \Delta(q_{s-1})} z_s}{\sqrt{\Delta(1) - \Delta(q_k)}} \right) \right]^{\frac{m_{k-2}}{m_{k-1}}} \dots \right]^{\frac{m_0}{m_1}}$$

where $\Delta(q)$ is the same kernel function defined in (4.133) and the mean term is

$$M \equiv m_c \int Dx\, \varphi(x), \tag{4.154}$$

where $m_c = \frac{1}{\sqrt{K}} \sum_l c_l$.

4.B.3 Saddle point equations

The aim of this section is to write the saddle point equations

$$q_{cd}^{-1} = -\alpha \frac{\partial \Delta_{cd}}{\partial q_{cd}} \frac{e^{\frac{1}{2}\sum_{ab} \Delta_{ab} \frac{\partial^2}{\partial h_a \partial h_b}} \frac{\partial^2}{\partial h_c \partial h_d} \prod_a e^{-\beta \ell(h_a - \kappa)} \Big|_{h_a=0}}{e^{\frac{1}{2}\sum_{ab} \Delta_{ab} \frac{\partial^2}{\partial h_a \partial h_b}} \prod_a e^{-\beta \ell(h_a - \kappa)} \Big|_{h_a=0}} \equiv -\alpha \frac{\partial \Delta_{cd}}{\partial q_{cd}} M_{cd}, \tag{4.155}$$

$$\frac{\partial \Delta_{cd}}{\partial q_{cd}} = e^{\frac{1}{2}\sum_{ab} q^{ab} \frac{\partial^2}{\partial s_a \partial s_b}} \frac{\partial \varphi(s_c)}{\partial s_c} \frac{\partial \varphi(s_d)}{\partial s_d} \Big|_{s_a=0}, \tag{4.156}$$

in the k-RSB ansatz in a compact form suitable for numerical evaluations. In the k-RSB ansatz, $\frac{\partial \Delta_{cd}}{\partial q_{cd}}$, $(q^{-1})_{cd}$ and M_{cd} will be k-RSB matrices as well. Therefore, in order to compute the update for the overlap q_s, we need to compute the matrix elements $\left[\boldsymbol{q}^{-1}\right]_s$, $[\boldsymbol{M}]_s = M_s$ and $\left[\frac{\partial \Delta_{cd}}{\partial q_{cd}}\right]_s$ with $s = 0, \dots, k$. We start from $\left[\frac{\partial \Delta_{cd}}{\partial q_{cd}}\right]_s$ which is

$$\left[\frac{\partial \Delta_{cd}}{\partial q_{cd}}\right]_s = \int Dx \left[\int Dy\, \varphi'\left(\sqrt{q_s}\, x + \sqrt{1-q_s}\, y\right)\right]^2 = \dot{\Delta}(q_s), \qquad s = 0, \dots, k \tag{4.157}$$

having denoted by a dot the derivative with respect to q. The matrix elements of M_{cd} instead can be written as

$$M_s = \int dh\, P(m_s, h) f'(m_s, h)^2, \qquad s = 0, \dots, k \tag{4.158}$$

where we have denoted with a prime a derivative with respect to h. P is instead found by the following iteration rule

$$P(m_{-1},h) = \delta(h), \tag{4.159}$$

$$P(m_0,h) = e^{m_{-1}f(m_0,h)} \int dz\, \mathcal{N}_{\Delta(q_0)-\Delta(q_{-1})}(z-h)\, P(m_{-1},z)\, e^{-m_{-1}f(m_{-1},z)} = \tag{4.160}$$

$$= \mathcal{N}_{\Delta(q_0)-\Delta(0)}(h), \tag{4.161}$$

$$P(m_l,h) = e^{m_{l-1}f(m_l,h)} \times \tag{4.162}$$

$$\times \int dz\, \mathcal{N}_{\Delta_{q_l}-\Delta_{q_{l-1}}}(z-h)\, P(m_{l-1},z)\, e^{-m_{l-1}f(m_{l-1},z)}, \qquad l = 1,\dots,k \tag{4.163}$$

which is the same as Sherrington Kirkpatrick (SK) model, apart for the effective order parameters.

Finally we can get the update for the steps $q_s, s = 0,\dots,k$ by computing the inverse elements of the computed matrix $p_s \equiv -\alpha\dot{\Delta}(q_s)M_s, s = 0,\dots,k$. The inverse elements of a generic k-RSB matrix with diagonal elements $p_{k+1} \equiv p_d$ are reported in section 4.A.2.

However in order to use those results, we need to know what is the diagonal value assumed by the k-RSB matrix $\boldsymbol{p}$, i.e. $p_{k+1} = p_d$. This can be computed knowing that the corresponding diagonal value of the overlap matrix $\boldsymbol{q}$ is $q_{k+1} = q_d = 1$. Therefore we can find p_d by exploiting equations (4.101)-(4.102); in the end one has to solve the implicit equation

$$1 = \frac{1}{p_d - p_k} - \frac{p_0}{\hat{\lambda}_0^2} - \sum_{s=0}^{k-1} \frac{p_{s+1}-p_s}{\hat{\lambda}_s \hat{\lambda}_{s+1}}, \tag{4.164}$$

where $\hat{\lambda}_s$ are the eigenvalues of the matrix p

$$\hat{\lambda}_s \equiv \sum_{i=s}^{k} m_i (p_{i+1} - p_i), \qquad s = 0,\dots,k \tag{4.165}$$

Once p_d is computed we can find the corresponding values of q_s, using the recursions

$$q_s = q_{s-1} - \frac{p_s - p_{s-1}}{\hat{\lambda}_{s-1}\hat{\lambda}_s}, \qquad s = 0,\dots,k \tag{4.166}$$

as derived in section 4.A.2.

Summary

To summarize, in order to solve the k-RSB saddle point equations, we use the following procedure. We start with an initial guess for q_s, $s = 0,\dots,k$ and a starting value for the minimal value and maximal value of x, $x_m = m_0$ $x_M = m_{k-1}$. We generate a grid of $k-2$ points between x_m and x_M, given by $m_1 < \cdots < m_{k-2}$; the grid need not to be necessary equispaced. Then

1. Compute the effective order parameters $\Delta(q_s)$ and their derivatives $\dot{\Delta}(q_s)$ for $s = 0,\dots,k$ using respectively (4.133), (4.157).

2. Compute $f(m_s,h)$ for $s = k,\dots,0$ using (4.140) and $P(m_s,h)$ for $s = 0,\dots,k$ using equations (4.159).

3. Compute M_s using (4.158) and then $p_s = -\alpha\dot{\Delta}(q_s)M_s$ with $s = 0,\dots,k$.

4. Compute p_d by solving the implicit equation (4.164).

5. Use relations (4.166) to get a new estimate of q_s from p_s.

6. Repeat points 1-5 until convergence.

7. Update the value of the minimal and maximal breaking-point evaluating (4.56) respectively in m_0 and m_{k-1}. Generate a new grid of values of $k-2$ points between x_m and x_M, given by $m_1 < \cdots < m_{k-2}$ and compute the values of q_s, $s = 0,\dots,k$, interpolating the $q(x)$ obtained at point 6.

8. Repeat points 1-7 until convergence.

Once convergence is reached, we can compute all the observable of interest, in particular the free entropy as

$$\begin{aligned}\phi =& \frac{1}{2}\ln(1-q_k) + \frac{q_0}{2\lambda_0} + \sum_{s=0}^{k-1}\frac{1}{2m_s}\ln\left(1+\frac{m_s(q_{s+1}-q_s)}{\lambda_{s+1}}\right) + \\ &+ \alpha\int dh\,\mathcal{N}_{\Delta(q_0)-\Delta(0)}(h)\,f(m_0,h)\,.\end{aligned} \tag{4.167}$$

In the left panel of Fig. 4.9, in the case $\varphi(h) = \mathrm{erf}(h)$, $\kappa = 0$ and $\alpha = 2.3$, we show the plot of $q(x)$ before and after updating the breaking points for the first time. On the right panel we show the corresponding update of the breaking points. It is evident that the convergence on the breaking point is reached very rapidly and most of the situations only two repetitions of points 1-5 are needed.

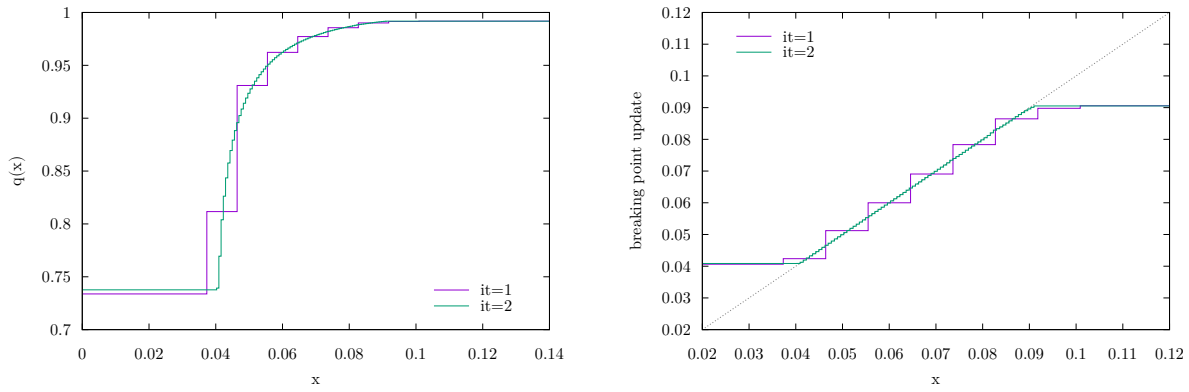

Figure 4.9: Left panel: $q(x)$ as a function of x before the first and second update of the breaking points (respectively violet and green line) as described in the text. Right panel: breaking point update (i.e. the right hand side of equation (4.56)) as a function of x. Here we have used $\varphi(h) = \mathrm{erf}(h)$ with $\kappa = 0$ and $\alpha = 2.3$. We initialized the code with $x_m = 0.001$ and $x_M = 0.9$ and we used $k = 100$. Only two iterations are sufficient to get a very precise estimate of x_m and x_M, i.e. the points where green line departs from the identity (dashed).

4.B.4 Replica Symmetric Ansatz

Entropy and saddle point equations

In the Replica Symmetric (RS) approximation we have, in the infinite β limit the following entropy

$$\phi = \mathcal{G}_S + \alpha \mathcal{G}_E, \tag{4.168}$$

$$\mathcal{G}_S = \frac{1}{2(1-q)} + \frac{1}{2}\ln(1-q), \tag{4.169}$$

$$\mathcal{G}_E = \int Dz_0 \ln H\left(\frac{\kappa + \sqrt{\Delta(q)-\Delta(0)}\, z_0}{\sqrt{\Delta(1)-\Delta(q)}}\right). \tag{4.170}$$

The corresponding saddle point equation for the overlap q reads

$$\frac{q}{2(1-q)^2} = -\alpha \frac{\partial \mathcal{G}_E}{\partial q} = \tag{4.171}$$

$$= \alpha \dot{\Delta}(q) \int Dz_0 \left[\frac{d}{dz} \ln H\left(\frac{z}{\sqrt{\Delta(1)-\Delta(q)}}\right)\Bigg|_{z=\kappa+\sqrt{\Delta(q)-\Delta(0)}z_0}\right]^2 = \tag{4.172}$$

$$= \frac{\alpha \dot{\Delta}(q)}{\Delta(1)-\Delta(q)} \int Dz_0\, GH^2\left(\frac{\kappa + \sqrt{\Delta(q)-\Delta(0)}\, z_0}{\sqrt{\Delta(1)-\Delta(q)}}\right). \tag{4.173}$$

dAT instability

Applying the RS ansatz on (4.52) will allow us to derive the instability of the RS ansatz itself, know as dAT instability. In this case $\lambda(q) = 1-q$ and the solution to the PDEs in equations (4.44) and (4.51) is trivial

$$P(q,h) = N_{\Delta(q)-\Delta(0)}(h), \tag{4.174}$$

$$f(q,h) = \ln H\left(\frac{\kappa+h}{\sqrt{\Delta(1)-\Delta(q)}}\right). \tag{4.175}$$

Inserting those identities in (4.55) we get

$$\begin{aligned}\frac{1}{(1-q)^2} =& \alpha\ddot{\Delta}(q)\int Dh\left[\frac{d}{dz}\ln H\left(\frac{z}{\sqrt{\Delta(1)-\Delta(q)}}\right)\Big|_{z=\kappa+\sqrt{\Delta(q)-\Delta(0)}h}\right]^2 + \qquad (4.176)\\ &+\alpha\dot{\Delta}^2(q)\int Dh\left[\frac{d^2}{dz^2}\ln H\left(\frac{z}{\sqrt{\Delta(1)-\Delta(q))}}\right)\Big|_{z=\kappa+\sqrt{\Delta(q)-\Delta(0)}h}\right]^2\\ =&\frac{\alpha\ddot{\Delta}(q)}{\Delta(1)-\Delta(q)}\int Dh\, GH^2\left(\frac{\kappa+\sqrt{\Delta(q)-\Delta(0)}h}{\sqrt{\Delta(1)-\Delta(q)}}\right)+ \qquad (4.177)\\ &+\frac{\alpha\dot{\Delta}^2(q)}{(\Delta(1)-\Delta(q))^2}\int Dh\,\mathcal{W}^2\left(\frac{\kappa+\sqrt{\Delta(q)-\Delta(0)}h}{\sqrt{\Delta(1)-\Delta(q)}}\right),\end{aligned}$$

where

$$\mathcal{W}(z) \equiv \frac{d^2}{dz^2}\ln H(z) = -\frac{d}{dz}GH(z) = GH(z)\left(z - GH(z)\right). \tag{4.178}$$

SAT/UNSAT transition in the RS approximation

To find the SAT/UNSAT transition in the RS approximation we have to perform the $q \to 1$ limit. As evinced in [Baldassi et al., 2019], in most of the activation functions, the kernel $\Delta(q)$ scales as

$$\Delta(q) \simeq \Delta(1) - \dot{\Delta}(1)\delta q, \tag{4.179}$$

with $\delta q = 1-q$.

Using the fact that $\ln H(x) \simeq -\frac{1}{2}\ln(2\pi) - \ln x - \frac{x^2}{2}$ as $x \to \infty$, retaining only the most divergent terms we get

$$\int Dz_0 \ln H\left(\frac{\kappa + \sqrt{\Delta(q)-\Delta(0)}z_0}{\sqrt{\Delta(1)-\Delta(q)}}\right) \simeq \tag{4.180}$$

$$\simeq \int_{-\frac{\kappa}{\sqrt{\Delta(1)-\Delta(0)}}}^{+\infty} Dz_0 \left[\frac{1}{2}\ln\delta q - \frac{\left(\kappa + \sqrt{\Delta(1)-\Delta(0)}z_0\right)^2}{2\dot{\Delta}(1)\delta q}\right] = \tag{4.181}$$

$$= \frac{1}{2}\ln(\delta q) H\left(\tilde{x}(\kappa)\right) - \frac{B(\kappa)}{2\dot{\Delta}(1)\delta q}, \tag{4.182}$$

where we have defined the quantities

$$\tilde{x}(\kappa) = -\frac{\kappa}{\sqrt{\Delta(1)-\Delta(0)}}, \tag{4.183}$$

$$B(\kappa) = \kappa\sqrt{\Delta(1)-\Delta(0)}G\left(\tilde{x}(\kappa)\right) + \left(\kappa^2 + \Delta(1) - \Delta(0)\right)H\left(\tilde{x}(\kappa)\right). \tag{4.184}$$

The free energy is

$$\phi = \frac{1}{2\delta q} + \frac{1}{2}\ln\delta q + \frac{\alpha}{2}\left(\ln(\delta q) H\left(\tilde{x}(\kappa)\right) - \frac{B(\kappa)}{\dot{\Delta}(1)\,\delta q}\right). \tag{4.185}$$

The derivative with respect to δq is

$$2\frac{\partial\phi}{\partial\delta q} = \frac{1}{\delta q} - \frac{1}{\delta q^2} + \alpha\left(\frac{H\left(\tilde{x}(\kappa)\right)}{\delta q} + \frac{B(\kappa)}{\dot{\Delta}(1)\,\delta q^2}\right) = 0\,. \tag{4.186}$$

In the critical capacity limit, i.e. $\alpha = \alpha_c^{\mathrm{RS}} - \delta\alpha$ we have that δq scales linearly in $\delta\alpha$, $\delta q = C\delta\alpha$. We get

$$2\frac{\partial\phi}{\partial\delta q} = \frac{1}{C\delta\alpha} - \frac{1}{C^2\delta\alpha^2} + (\alpha_c - \delta\alpha)\left[\frac{H\left(\tilde{x}(\kappa)\right)}{C\delta\alpha} + \frac{B(\kappa)}{\dot{\Delta}(1)\,C^2\delta\alpha^2}\right] \tag{4.187}$$

$$= \frac{1}{C\delta\alpha}\left[1 + \alpha_c H\left(\tilde{x}(\kappa)\right) - \frac{B(\kappa)}{C\dot{\Delta}(1)}\right] + \frac{1}{C^2\delta\alpha^2}\left[\alpha_c\frac{B(\kappa)}{\dot{\Delta}(1)} - 1\right] = 0\,. \tag{4.188}$$

The first term gives the scaling of δq, the second gives us the critical capacity in terms of the margin

$$\alpha_c^{\mathrm{RS}} = \frac{\dot{\Delta}(1)}{B(\kappa)}. \tag{4.189}$$

Notice that imposing (4.189) is equivalent to impose that the divergence $1/\delta q$ in the entropy (4.185) is eliminated at the critical capacity (and the free energy correctly goes to $-\infty$ in that limit). In particular, in the zero margin case we get that $B(\kappa=0)=\frac{\Delta(1)-\Delta(0)}{2}$ and therefore

$$\alpha_c^{\mathrm{RS}}=\frac{2\Delta'(1)}{\Delta(1)-\Delta(0)}=\frac{2\int Dh\,\varphi'(h)^2}{\int Dh\,\varphi^2(h)-\left(\int Dh\,\varphi(h)\right)^2}, \tag{4.190}$$

as was previously derived in [Baldassi et al., 2019].

4.B.5 1RSB ansatz

Entropy

In the 1RSB approximation and in the error counting loss case the entropy reads

$$\phi=\mathcal{G}_S+\alpha\mathcal{G}_E, \tag{4.191}$$

$$\mathcal{G}_S=\frac{1}{2}\left(\frac{q_0}{1-q_1+m(q_1-q_0)}+\frac{m-1}{m}\ln(1-q_1)+\frac{1}{m}\ln(1-q_1+m(q_1-q_0))\right), \tag{4.192}$$

$$\mathcal{G}_E=\frac{1}{m}\int Dz_0\,\ln\int Dz_1\,H^m\left(\frac{\kappa-\sqrt{\Delta(q_0)-\Delta(0)}z_0-\sqrt{\Delta(q_1)-\Delta(q_0)}z_1}{\sqrt{\Delta(1)-\Delta(q_1)}}\right). \tag{4.193}$$

Gardner Transition

In the 1RSB case usually the instability to the fRSB type of ansatz (Gardner transition) develops at q_1. Imposing a 1RSB ansatz in (4.55) and evaluating it in $q=q_1$ we get

$$\frac{1}{(1-q_1)^2}= \tag{4.194}$$

$$=\frac{\alpha\ddot{\Delta}(q)}{\Delta(1)-\Delta(q_1)}\int Dz_0\,\frac{\int Dz_1\,H^m(\mathcal{A}(z_0,z_1))\,GH^2(\mathcal{A}(z_0,z_1))}{\int Dz_1\,H^m(\mathcal{A}(z_0,z_1))} \tag{4.195}$$

$$+\frac{\alpha\dot{\Delta}^2(q)}{(\Delta(1)-\Delta(q_1))^2}\int Dz_0\,\frac{\int Dz_1\,H^m(\mathcal{A}(z_0,z_1))\,\mathcal{W}^2(\mathcal{A}(z_0,z_1))}{\int Dz_1\,H^m(\mathcal{A}(z_0,z_1))}, \tag{4.196}$$

where

$$\mathcal{A}(z_0,z_1)\equiv\frac{\kappa+\sqrt{\Delta(q_0)-\Delta(0)}z_0+\sqrt{\Delta(q_1)-\Delta(q_0)}z_1}{\sqrt{\Delta(1)-\Delta(q_1)}}. \tag{4.197}$$

We plot the Gardner transition for the committee machine with the ReLU activation function in Figure 4.10.

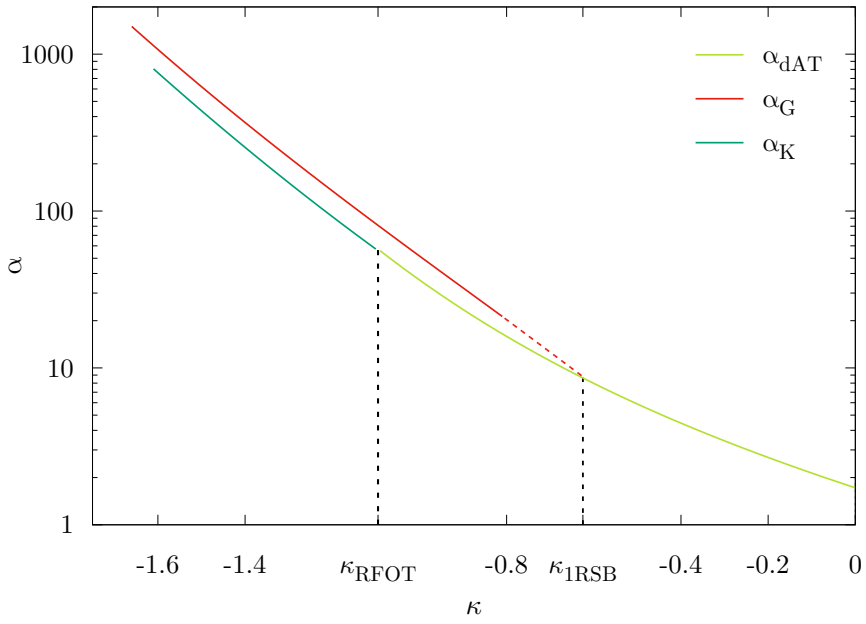

Figure 4.10: Plot of the dAT (eq. (4.176)), Gardner (eq. (4.194)) and Kautzmann transition lines as a function of κ for the committee machine in the large width limit with the ReLU activation function.

SAT/UNSAT transition in the 1RSB approximation

In order to compute the SAT/UNSAT transition in the 1RSB approximation, one needs to perform the limit $q_1 \to 1$ with $m = \tilde{m}(1-q_1) \to 0$ [Barkai et al., 1990, 1992, Baldassi et al., 2019]. Therefore we express all in terms of m by using $\delta q_1 = 1 - q_1 = \frac{m}{\tilde{m}}$ obtaining

$$\phi = \frac{1}{2m}\left[m \ln\left(\frac{m}{\tilde{m}}\right) + \ln\left(1 - m + \tilde{m}(1-q_0)\right) + \frac{\tilde{m} q_0}{1 - m + \tilde{m}(1-q_0)} + 2m\alpha \mathcal{G}_E \right]. \tag{4.198}$$

In the limit $m \to 0$, we need to assure that the entropy goes to $-\infty$, so we need to impose that the coefficient of first order expansion of the free energy (which is of order $1/m$) vanishes. This is equivalent to impose that at the SAT/UNSAT transition

$$\ln\left(1 + \tilde{m}(1-q_0)\right) + \frac{\tilde{m} q_0}{1 + \tilde{m}(1-q_0)} + 2\alpha_c \mathcal{F}(\kappa; q_0, \tilde{m}) = 0, \tag{4.199}$$

or

$$\alpha_c = \frac{\ln\left(1 + \tilde{m}(1-q_0)\right) + \frac{\tilde{m} q_0}{1 + \tilde{m}(1-q_0)}}{2\mathcal{F}(\kappa; q_0, \tilde{m})}, \tag{4.200}$$

where

$$\mathcal{F}(\kappa;q_0,\tilde{m}) = \lim_{m\to 0}\int Dz_0 \times \tag{4.201}$$
$$\times \ln \int Dz_1\, H^m\left(\frac{\kappa-\sqrt{\Delta(q_0)-\Delta(0)}z_0-\sqrt{\Delta(1)-\Delta(q_0)}z_1}{\sqrt{\dot{\Delta}(1)\frac{m}{\tilde{m}}}}\right).$$

Expanding the $H(x)$ function at large arguments $H(x)\simeq G(x)/x$ and performing the integral over z_1 one gets

$$\mathcal{F}(\kappa;q_0,\tilde{m}) = \tag{4.202}$$
$$= \int Dz_0 \ln\left[H\left(\frac{\kappa+\sqrt{\Delta(q_0)-\Delta(0)}z_0}{\sqrt{\Delta(1)-\Delta(q_0)}}\right)+\right. \tag{4.203}$$
$$\left.+\frac{\sqrt{\dot{\Delta}(1)}\,e^{-\frac{\tilde{m}\left(\kappa+\sqrt{\Delta(q_0)-\Delta(0)}z_0\right)^2}{2\left(\dot{\Delta}(1)+\left(\Delta(1)-\Delta(q_0)\right)\tilde{m}\right)}}}{\sqrt{\dot{\Delta}(1)+(\Delta(1)-\Delta(q_0))\,\tilde{m}}}H\left(-\sqrt{\frac{\dot{\Delta}(1)}{\Delta(1)-\Delta(q_0)}}\frac{\kappa+\sqrt{\Delta(q_0)-\Delta(0)}z_0}{\sqrt{\dot{\Delta}(1)+(\Delta(1)-\Delta(q_0))\,\tilde{m}}}\right)\right].$$

4.C Observables

Once the saddle point equations are solved, we can use the solutions not only to compute the entropy, but also other observables of interest.

4.C.1 Distribution of Stabilities

An observable of interest is the so called distribution of stability $\mathcal{P}(h)$, i.e.

$$\hat{\mathcal{P}}(h) \equiv \frac{1}{P}\sum_{\mu=1}^{P}\delta\left(h-\Delta^{\mu}(\boldsymbol{w};\kappa)\right), \tag{4.204}$$
$$\mathcal{P}(h) = \overline{\left\langle\hat{\mathcal{P}}(h)\right\rangle}, \tag{4.205}$$

this quantity, also called "gap probability distribution" in the context of the jamming of hard spheres [Franz and Parisi, 2016], quantifies in which fashion the constraints of the training set are satisfied. In the context of machine learning it has been recognized that well-generalizing solutions have a stability distribution that is small and flat around zero [Baldassi et al., 2021, 2022]; those kind of solutions can be found by biasing the measure towards flat regions [Baldassi et al., 2019].

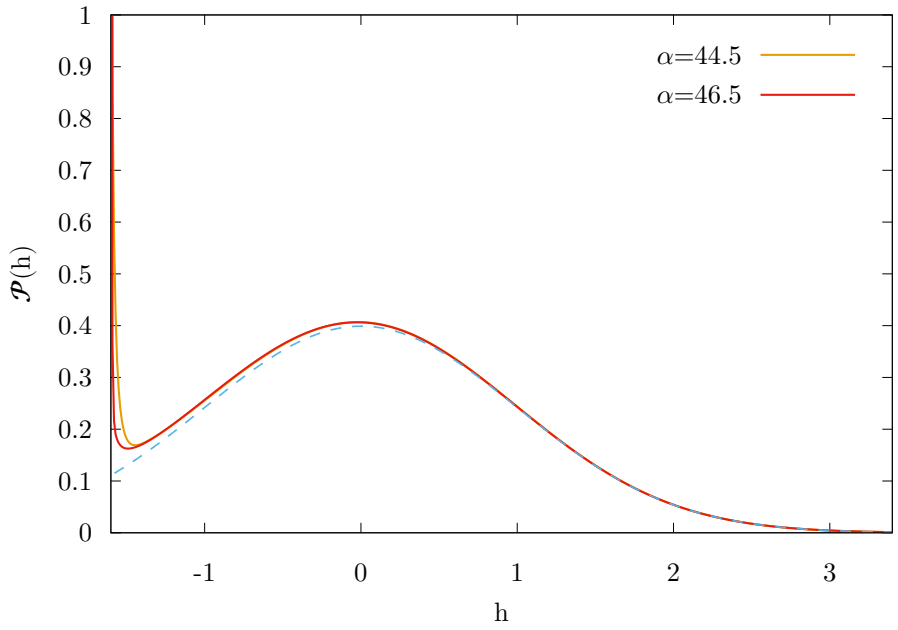

Figure 4.11: Stability distribution for $\kappa = -1.6$ and two values of α. As $\alpha \to \alpha_c$ the distribution develops a power law behavior around small stabilities $h \sim \kappa$. We show in the dashed blue line a standard normal Gaussian distribution for comparison.

We can easily compute this distribution by rewriting the partition function as

$$Z = \int d\mu(\boldsymbol{w})\, e^{-\beta \sum_\mu \ell(\Delta^\mu(\boldsymbol{w};\kappa))} = \int d\mu(\boldsymbol{w})\, e^{P \int dh\, \hat{\mathcal{P}}(h)[-\beta \ell(h)]}. \tag{4.206}$$

The stability distribution can be obtained by taking a derivative of the free entropy with respect to the loss function, i.e.

$$\mathcal{P}(h) = -\frac{1}{\alpha\beta}\frac{\partial \phi}{\partial \ell(h)} = e^{-\beta\ell(h)} \int dz\, P(q_M, z)\, \mathcal{N}_{\Delta_1 - \Delta_{q_M}}(h + z + \kappa) e^{-f(q_M, z)}. \tag{4.207}$$

A generic observable $\mathcal{O}$ of the stability h, can be therefore easily expressed as an integral over the stability distribution

$$\langle \mathcal{O} \rangle \equiv \int dh\, \mathcal{P}(h)\, \mathcal{O}(h) = \int dh\, \mathcal{O}(h)\, e^{-\beta \ell(h)} \int dz\, P(q_M, z)\, \mathcal{N}_{\Delta(1)-\Delta(q_M)}(h + z + \kappa) e^{-f(q_M, z)} \tag{4.208}$$

$$= \int dz\, P(q_M, z) e^{-f(q_M, z)} \int dh\, \mathcal{O}(h)\, e^{-\beta \ell(h)}\, \mathcal{N}_{\Delta(1)-\Delta(q_M)}(h + z + \kappa) \tag{4.209}$$

$$= \int dz\, P(q_M, z) \frac{\int dh\, \mathcal{O}(h)\, e^{-\beta \ell(h)}\, \mathcal{N}_{\Delta(1)-\Delta(q_M)}(h + z + \kappa)}{\int dh\, e^{-\beta \ell(h)}\, \mathcal{N}_{\Delta(1)-\Delta(q_M)}(h + z + \kappa)}. \tag{4.210}$$

As an example, the fraction of violated constraints z can be obtained by using the observable $\mathcal{O}(h) = \Theta(-h)$, i.e.

$$z = \int_{-\infty}^{0} dh\, \mathcal{P}(h). \tag{4.211}$$

4.C.2 Pressure

The average stability of violated constraints or "pressure" [Franz et al., 2017] can be obtained by using as observable $\mathcal{O}(h) = -h\Theta(-h)$, which gives

$$p = -\int_{-\infty}^{0} dh\, \mathcal{P}(h) h = -\int dz\, P(q_M, z) \frac{\int_{-\infty}^{0} dh\, h\, e^{-\beta \ell(h)}\, \mathcal{N}_{\Delta(1)-\Delta(q_M)}(h + z + \kappa)}{\int dh\, e^{-\beta \ell(h)}\, \mathcal{N}_{\Delta(1)-\Delta(q_M)}(h + z + \kappa)}. \tag{4.212}$$

In the SAT phase, by definition $\mathcal{P}(h) = 0$ for $h < 0$, therefore the pressure (and also the fraction of violated constraints) vanishes. However one can study how it tends to zero with the temperature. For the number of error loss this decays exponentially to 0 with β going to infinity. For the quadratic hinge loss it vanishes linearly to zero with $T = 1/\beta$; in the SAT region; indeed

$$p = -T \int dz\, P(q_M, z) \frac{\mathcal{N}_{\Delta(1)-\Delta(q_M)}(z + \kappa) \int_{-\infty}^{0} dh\, h\, e^{-\frac{h^2}{2}}}{\int_0^{\infty} dh\, \mathcal{N}_{\Delta(1)-\Delta(q_M)}(h + z + \kappa)} = \tag{4.213}$$

$$= -T \int dz\, P(q_M, z) f'(q_M, z)\,. \tag{4.214}$$

Using the property

$$\frac{d}{dx} \int_0^1 dx\, P(x, h) f'(x, h) = 0\,. \tag{4.215}$$

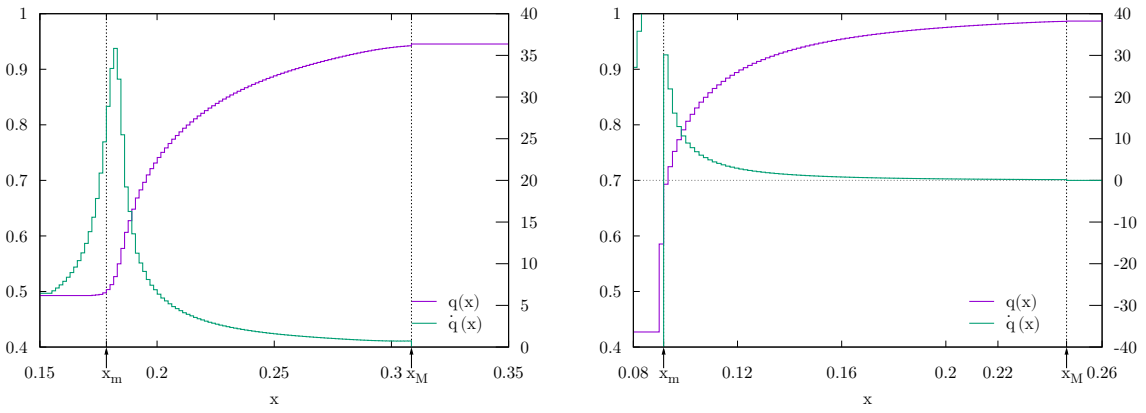

Figure 4.12: Behavior of $q(x)$ (violet) and $\dot{q}(x)$ (green) as a function of x in the phase where typical states do not possess any gap (left panel, $\kappa = -1.27$ and $\alpha \simeq 18$) and a phase where they possess a gap (right panel, $\kappa = -1.4$ and $\alpha \simeq 26.7$). When there is no gap $\dot{q}$ is always positive in the range $x \in [x_m, x_M]$. A gap instead appears for a fixed κ at a value of $\alpha = \alpha^{1+fRSB}(\kappa)$ where for $x \to x_m$, the denominator of (4.217) becomes zero, signaling an infinite derivative of $q(x)$. For $\alpha > \alpha^{1+fRSB}$, the denominator suddenly becomes negative at $x = x_m$.

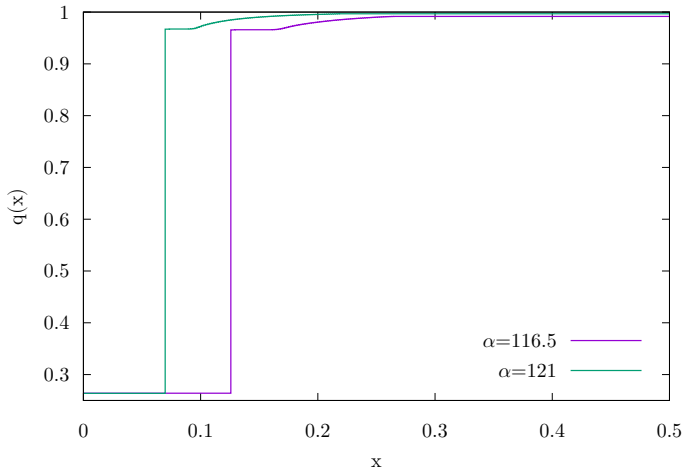

Figure 4.13: $q(x)$ deep in the Gardner phase (here $\kappa = -2.0$), where one can see clearly that the point m where there is a jump is distinct with x_m.

one gets

$$p = -T \int dz\, P(q_M, z) f'(q_M, z) = -T \int dz\, P(q_m, z) f'(q_m, z)\,. \tag{4.216}$$

The "reduced pressure" presented in the main text is therefore related to the pressure by $p = T\tilde{p}$.

4.D Equation for $\dot{q}(x)$ and the transition to the overlap gapped phase

Higher order derivatives of the saddle point equation can give information to derivatives of $q(x)$ in the interval $[x_m, x_M]$. For example, deriving twice equation (4.55) and solving for $\frac{dq}{dx}$, we get

$$\frac{dq}{dx} = \frac{\frac{1}{\lambda^3(x)} + \alpha\dot{\Delta}^3 \int dh\, P(x,h) f''(x,h)^3}{\frac{\alpha}{2}\int dh\, P(x,h)\mathcal{B}(x,h) - \frac{3x^2}{\lambda^4(x)}}, \tag{4.217}$$

where

$$\begin{aligned}\mathcal{B}(x,h) = 6\dot{\Delta}^4 x^2 f''^4 + \dot{\Delta}^4 f''''^2 - 12\dot{\Delta}^4 x f'' f'''^2 + \ddddot{\Delta} f'^2 \\ + \left(3\ddot{\Delta}^2 + 4\dot{\Delta}\dddot{\Delta}\right) f''^2 + 6\ddot{\Delta}\dot{\Delta}^2\left(f'''^2 - 2x f''^3\right),\end{aligned} \tag{4.218}$$

which in the case $\Delta(q) = q$ reduces to

$$\frac{dq}{dx} = \frac{\frac{1}{\lambda^3(x)} + \alpha \int dh\, P(x,h) f''(x,h)^3}{\frac{\alpha}{2}\int dh\, P(x,h)[6x^2 f''(x,h)^4 + f''''(x,h)^2 - 12x f''(x,h) f'''(x,h)^2] - \frac{3x^2}{\lambda^4(x)}}. \tag{4.219}$$

As we have described in the main text, we used equation (4.219) to evaluate the transition between the fRSB phase (no overlap gap phase), to the Gardner phase (which is overlap gapped). Indeed the transition is signaled by the divergence of the derivative of $q(x)$ at $x = x_m$, see Figure 4.12. If then one moves in a region (κ, α) deep in the Gardner phase (i.e. for κ very negative and α large) one can see that the point where the $q(x)$ has a jump (i.e. for $x = m$) becomes visibly distinct and lower than x_m, see Figure 4.13. Similar transitions have been seen in [Rizzo, 2013], even if in a slightly different setting.

Chapter 5

Adding Signal and Structure to the Data

All the results from the previous chapters where derived under the hypothesis of a factorized input distribution $P(\boldsymbol{x}) = \prod_{i=1}^{N} P(x_i)$ with random labels, a setting that goes under the name of the *Storage Problem*. As I have already mentioned in the first part, this bears no resemblance to the actual datasets used in practice. It is a setting still worth investigating, as it was observed that modern day neural networks have the capacity to fit also these types of unstructured datasets [Zhang et al., 2021], however one of the most important features of neural networks, that is the capability to generalize to new unseen data, cannot be modeled with this approach. In this chapter then we will see how it is possible to consider a more realistic model of data, both by considering inputs with (weakly) correlated components, and by considering a teacher network with a certain structure that labels the inputs. I will build onto the formalism of the *Hidden Manifold Model* introduced in section 2.4. In section 5.1, I review the relevant literature. In section 5.2, I analyze how the *Tree-committee Machine* performs on a dataset generated by a Hidden Manifold in the limit of many hidden neurons. In section 5.3, instead I allow for a low-rank perturbation to the first layer weights that can be learned, and study the amount of data necessary for the teacher's spike to be recovered by the student.

5.1 Related Works

The performance of a number of famous learning tasks, including ridge and ridgeless regression and max-margin learning, with random features in the proportional regime, where the number of examples P is proportional to the number of parameters N, have been characterized in a number of works [Hastie et al., 2022, Montanari et al., 2019, Mei and Montanari, 2022]. The picture that has emerged is clear: random features behave like linear noisy features, therefore cannot perform better than linear models. These results have been extended by a line of work

which introduced the *Hidden Manifold Model*, already described in the introduction, where the feature matrix is allowed to be deterministic, as long as a certain alignment condition is satisfied (see equation 2.70). In [Goldt et al., 2020] authors consider the case of online gradient descent in the teacher-student scenario for 2 layer architectures, while full batch learning for general loss functions and feature matrices is studied in [Gerace et al., 2020]. The validity of Gaussian Equivalence is extended to pre-trained generative networks in [Goldt et al., 2022], using both analytical and numerical evidence.

Going beyond the proportional regime, [Ghorbani et al., 2021] considers learning a polynomial teacher in two asymptotic regimes, first when the limit of number of examples P going to infinity is taken before the limit of number of hidden neurons N and input dimensionality D going to infinity, and second when the $N \to \infty$ is taken before the $P, D \to \infty$ limits. They prove that in the first regime, when $D^{1+\delta} < N < D^{2-\delta}$ for some $\delta > 0$, then the test error achieved by a random feature model is lower bounded by the error achieved by a linear network, while in the second regime, when $P < D^{2-\delta}$ the same lower bound holds. [Aguirre-López et al., 2024] instead uses statistical physics methods in the asymptotic regime $P \sim D^K$ $N \sim D^L$ to derive a closed form expression for the generalization error of a RFM student with a polynomial teacher.

One other possibility for escaping the linear regime is to consider a feature matrix that is trained for one Gradient Descent step. In [Ba et al., 2022] authors study ridge regression with such feature matrix, and consider two cases: one in which the learning rate is small, such that the correction to the pre-activations is perturbative, and one in which the learning rate is large enough that the correction to the pre-activations is of the same order as the original one. By showing that the correction given by the gradient step is low rank, and extending the Gaussian Equivalence theorem to this case, they show that the first scenario, which is the closest to that of section 5.3, is not enough to escape the linear regime, while the second is sufficient to outperform the best linear model. Inspired by this setting, [Cui et al., 2024] studies how this one pass trained feature matrix (with large learning rate) can learn a non-linear function of the inputs in the case where the gradient step is big.

Interestingly, this low rank perturbation resembles a widely used technique for fine-tuning of large neural networks, Low Rank Adaptation (LoRA) [Hu et al., 2021]. Indeed it was observed that it is sufficient to add very low rank, often rank two or three, trainable matrices to all frozen weight matrices to match, and sometimes exceed, the performance of full fine-tuning. Although this technique has become standard in Large Language Models and Image Generation Models, a theoretical picture which explains its great performance is somewhat lacking. Some notable exceptions are [Malladi et al., 2023], where the authors show that LoRA is approximately equal to full fine-tuning in the Lazy Regime, and [Zeng and Lee, 2023] where explicit conditions on the minimum rank needed to learn a target model are derived.

5.2 Hidden Manifold Committee Machine

In this section, we consider a *Tree-Committee Machine*, that is a two layer neural network with non-overlapping receptive fields and fixed second layer weights, that has already been introduced in section 4.2. We study its performance this time on data generated by a hidden manifold model, that is

$$\boldsymbol{x}^\mu = \varphi\left(\frac{1}{\sqrt{D}} F \boldsymbol{\xi}^\mu\right), \qquad \boldsymbol{\xi}^\mu \sim \mathcal{N}(0, \mathbb{I}_D) \tag{5.1}$$

$$y^\mu = \text{sign}\left(\frac{1}{\sqrt{D}} \tilde{\boldsymbol{w}} \cdot \boldsymbol{\xi}^\mu\right), \tag{5.2}$$

for $\mu \in [P]$. The output of the network is given by equation (4.2).

As we have done above, we start by calculating the free entropy

$$\phi = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathcal{D}} \log \mathcal{Z} = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathcal{D}} \log \int \prod_{i=1}^{N} \prod_{l=1}^{K} dW_{il} P_w(\{w_{il}\}_{il}) e^{-\beta \sum_\mu \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}^\mu), y^\mu)}, \tag{5.3}$$

where we have used the notation w_{il} to indicate the ith weight of the lth block (so $i \in [N/K]$, $l \in [K]$). As we have proceeded above we use the replica trick to take the average of the log, and are thus reduced to computing the average replicated partition function

$$\mathbb{E}_{\mathcal{D}} \mathcal{Z}^n = \int \prod_{ail} dw^a_{il} P_w(w^a_{il}) e^{-\beta \sum_{\mu a} \ell(\{\lambda^{a\mu}_l\}_l, u^\mu)} \times \tag{5.4}$$

$$\times \prod_\mu \underbrace{\mathbb{E}_{\boldsymbol{\xi}} \left[\prod_l \delta\left(\lambda^{a\mu}_l - \sqrt{\frac{K}{N}} \sum_{i=1}^{N/K} w^a_{ik} \varphi\left(\frac{1}{\sqrt{D}} \sum_{k=1}^{D} F^l_{ik} \xi^\mu_k\right)\right) \delta\left(u^\mu - \frac{1}{\sqrt{D}} \sum_{i=1}^{D} \tilde{w}_i \xi^\mu_i\right)\right]}_{P\left(\{\lambda^{a\mu}_l\}_{al}, u^\mu\right)}.$$

As we can see we are again in the condition to apply the *Gaussian Equivalence Principle,* with the only difference that now we have $K \times n$ activations λ^a_l. So the joint distribution of u and $\{\lambda^a_l\}_{l,a}$ (the distribution over the indeces μ factorizes) can be written as (using the notation $Y_0 \equiv u$ and $Y_{l+Ka} = \lambda^a_l$)

$$P\left(u, \{\lambda^a_l\}_{la}\right) = \frac{1}{(2\pi)^{Kn/2} \sqrt{2\pi \det \Sigma}} e^{-\frac{1}{2}(Y-m)^T \Sigma^{-1} (Y-m)}, \tag{5.5}$$

where

$$\Sigma = \begin{pmatrix} 1 & M_l^a \\ M_l^a & Q_{lm}^{ab} \end{pmatrix}, \qquad m_0 = 0, \qquad m_{l+Ka} = \rho_l^a, \tag{5.6}$$

$$Q_{lm}^{ab} = \mathbb{E}_\xi\left[\lambda_l^a \lambda_m^b\right] - \mathbb{E}_\xi\left[\lambda_l^a\right]\mathbb{E}_\xi\left[\lambda_m^b\right] = \kappa_*^2 q_l^{ab}\delta_{lm} + \kappa_1^2 p_{lm}^{ab}, \tag{5.7}$$

$$M_l^a = \mathbb{E}_\xi\left[\lambda_l^a u\right] = \kappa_1 r_l^a, \tag{5.8}$$

$$\rho_l^a = \mathbb{E}_\xi\left[\lambda_l^a\right] = \kappa_0\sqrt{\frac{K}{N}}\sum_i w_{il}^a, \tag{5.9}$$

$$q_l^{ab} = \frac{K}{N}\sum_{i=1}^{N/K} w_{il}^a w_{il}^b, \tag{5.10}$$

$$s_{kl}^a = \sqrt{\frac{K}{N}}\sum_{i=1}^{N/K} w_{il}^a F_{ki}^l, \tag{5.11}$$

$$p_{lm}^{ab} = \frac{1}{D}\sum_{k=1}^{D} s_{kl}^a s_{km}^b, \tag{5.12}$$

$$r_l^a = \frac{1}{D}\sum_{k=1}^{D} s_{kl}^a \tilde{w}_k. \tag{5.13}$$

To simplify notation, we will use boldface to indicate a K-dimensional vector, so $\lambda_l^a \to \boldsymbol{\lambda}^a$. We will also consider an odd activation function, such that $\kappa_0 = 0$, and choose a spherical prior for the weights of the network such that

$$\sum_{i=1}^{N/K} w_{il}^2 = \frac{N}{K}. \tag{5.14}$$

Using the fourier representation of the Gaussian distribution, the average partition function can written as

$$\mathbb{E}_\xi \mathcal{Z}^n = \int d\tilde{\boldsymbol{w}} P_{\tilde{w}}(\tilde{\boldsymbol{w}}) \prod_{ai} d\boldsymbol{w}_i^a \prod_\mu du^\mu d\hat{u}^\mu \prod_{a\mu} d\boldsymbol{\lambda}^{a\mu} d\boldsymbol{\lambda}^{a\mu}\, P_w(\boldsymbol{w}^a) e^{-\beta\sum_{\mu a}\ell(\lambda^{a\mu},u^\mu)} \times \tag{5.15}$$

$$\times \prod_\mu \frac{1}{\sqrt{2\pi\det\Sigma}} e^{iu_\mu\hat{u}_\mu - \frac{\hat{u}_\mu^2}{2} - \frac{1}{2}\sum_{ab}(\hat{\boldsymbol{\lambda}}^{a\mu})^T Q^{ab}\hat{\boldsymbol{\lambda}}^{b\mu} + i\sum_a \hat{\boldsymbol{\lambda}}^{a\mu}\cdot\boldsymbol{\lambda}^{a\mu} - \sum_a \hat{u}^\mu \boldsymbol{\lambda}^{a\mu}\cdot\boldsymbol{M}_a}. \tag{5.16}$$

Introducing delta functions for the order parameters and averaging over the projection matrix $F_{ij}^l \sim \mathcal{N}(0,1)$ we get to the expression

$$\mathbb{E}\mathcal{Z}^n = \int dQ\, d\hat{Q}\, dr\, d\hat{r} d\rho\, d\hat{\rho}\; e^{Nn\phi_{var}}, \tag{5.17}$$

$$\begin{aligned}\phi_{var} =& -\frac{1}{2n}\alpha_D \sum_{ablm} i\hat{p}^{ab}_{lm} p^{ab}_{lm} - \frac{1}{2n}\frac{1}{K}\sum_{abl} i\hat{q}^{ab}_l q^{ab}_l - \frac{\alpha_D}{n}\sum_{al} i\hat{r}_{al} r_{al} \\ &+ \frac{1}{n}G_{SS} + \frac{\alpha_D}{n}G_{SE} + \frac{\alpha}{n}G_E,\end{aligned} \tag{5.18}$$

$$\begin{aligned}G_E =& \log \int \prod_a \frac{d\hat{\boldsymbol{\lambda}}^a d\boldsymbol{\lambda}^a}{(2\pi)^K}\frac{d\hat{u}du}{2\pi} e^{-\beta\sum_a \Theta\left(-\mathrm{sign}(u)\sum_l c_l \varphi(\lambda^a_l)\right) + i\hat{u}u + \sum_{al} i\hat{\boldsymbol{\lambda}}^a\cdot\boldsymbol{\lambda}^a} \times \\ &\times e^{-\frac{1}{2}(\hat{u})^2 - \frac{1}{2}\sum_{ab}\left(\hat{\boldsymbol{\lambda}}^a\right)^T Q^{ab}\hat{\boldsymbol{\lambda}}^b - \sum_a \hat{u}M^a\cdot\hat{\boldsymbol{\lambda}}^a},\end{aligned} \tag{5.19}$$

$$G_{SS} = \frac{1}{K}\sum_l \log \int \prod_a dw^a\; e^{\frac{1}{2}\sum_{ab} i\hat{q}^{ab}_l w^a w^b}, \tag{5.20}$$

$$G_{SE} = \log \int D\tilde{w} \prod_a \frac{d\hat{\boldsymbol{s}}^a d\boldsymbol{s}^a}{2\pi}\; e^{\sum_{al} i\hat{\boldsymbol{s}}^a\cdot\boldsymbol{s}^a - \frac{1}{2}\sum_{abl} q^{ab}_l \hat{s}^a_l \hat{s}^b_l + \frac{1}{2}\sum_{ablm} i(\boldsymbol{s}^a)^T \hat{p}_{ab}\boldsymbol{s}^b + i\tilde{w}\sum_a \hat{\boldsymbol{r}}^a\cdot\boldsymbol{s}^a}. \tag{5.21}$$

Since we will be interested in the $K \to \infty$ limit, we consider a Replica Symmetric ansatz which already contains the first order expansions in K^{-1}

$$p^{ab}_{lm} = \begin{cases} p_d + \frac{p_c}{K}, & a = b,\, l = m \\ \frac{p_A}{K}, & a = b,\, l \neq m \\ p_B + \frac{p_E}{K}, & a \neq b,\, l = m \\ \frac{p_0}{K}, & a \neq b,\, l \neq m \end{cases} \qquad r_a = \frac{r}{\sqrt{K}} \qquad q^{ab}_l = \begin{cases} 1 & a = b \\ q_0 & a \neq b \end{cases} \tag{5.22}$$

and likewise for hat variables. Details of the calculation are provided in appendix 5.A. The variational free energy, which still needs to be extremised, is

$$\begin{aligned}\phi_{var} \underset{n\to 0}{\to} & -\frac{1}{2}\alpha_D K\left(\hat{p}_d + \frac{\hat{p}_C}{K}\right)\left(p_d + \frac{p_C}{K}\right) - \frac{1}{2}\alpha_D K(K-1)\frac{\hat{p}_A p_A}{K^2} + \frac{1}{2}\alpha_D K\left(\hat{p}_B + \frac{\hat{p}_E}{K}\right)\left(p_B + \frac{p_E}{K}\right) + \\ &+ \frac{1}{2}\alpha_D K(K-1)\frac{\hat{p}_O p_O}{K^2} + \frac{1}{2}\hat{q}_1 + \frac{1}{2}\hat{q}_0 q_0 - \alpha_D \hat{r} r + \alpha\mathcal{G}_E + \mathcal{G}_{SS} + \alpha_D\mathcal{G}_{SE}.\end{aligned} \tag{5.23}$$

with

$$\mathcal{G}_{SS} = \frac{1}{2}\log(\frac{2\pi}{\hat{q}_1+\hat{q}_0}) + \frac{1}{2}\frac{\hat{q}_0}{\hat{q}_1+\hat{q}_0}, \tag{5.24}$$

$$\mathcal{G}_{SE} = \log\left[(q_1-q_0)\Big(\hat{p}_B-\hat{p}_d+\frac{1}{q_1-q_0}\Big)^{-K/2}\sqrt{1+\frac{(\hat{p}_O-\hat{p}_A)}{\hat{p}_B-\hat{p}_d+\frac{1}{q_1-q_0}}}\right] - \frac{1}{2}K\frac{q_0}{q_1-q_0} \tag{5.25}$$

$$+\frac{1}{2}K\frac{(\hat{p}_B+\frac{q_0}{(q_1-q_0)^2}+\frac{\hat{r}^2}{K})}{\hat{p}_B-\hat{p}_d+\frac{1}{q_1-q_0}} - \frac{1}{2}K\frac{\frac{\hat{p}_O-\hat{p}_A}{K}(\hat{p}_B+\frac{q_0}{(q_1-q_0)^2}+\frac{\hat{r}^2}{K})-(K-1)\frac{\hat{p}_O+\hat{r}^2}{K}(\hat{p}_A-\hat{p}_O)}{(\hat{p}_B-\hat{p}_d+\frac{1}{q_1-q_0})(\hat{p}_B-\hat{p}_d+\frac{\hat{p}_A-\hat{p}_O}{K}(K-1)+\frac{1}{q_1-q_0})}, \tag{5.26}$$

$$\mathcal{G}_E = 2\int D\eta\, H\left(-\frac{\sqrt{\frac{2}{\pi Q_d}}M}{\sqrt{\frac{2}{\pi}\frac{Q_0-M^2}{Q_d}+1-\frac{4}{\pi}\arctan\Big(\sqrt{\frac{Q_d-Q_B}{Q_B+Q_d}}\Big)}}\eta\right)\times \tag{5.27}$$

$$\times\log H_\beta\left(\frac{-\sqrt{\frac{2Q_0}{\pi Q_d}+1-\frac{4}{\pi}\arctan\Big(\sqrt{\frac{Q_d-Q_B}{Q_B+Q_d}}\Big)}\eta}{\sqrt{\frac{4}{\pi}\arctan\Big(\sqrt{\frac{Q_d-Q_B}{Q_B+Q_d}}\Big)+\frac{2}{\pi}\frac{Q_A-Q_O}{Q_d}}}\right).$$

Here we used $\varphi = \text{sign}$, we already performed the $K\to\infty$ limit in the expression for $\mathcal{G}_E$ and we defined

$$H_\beta(x) = e^{-\beta} + (1-e^{-\beta})H(x), \tag{5.28}$$

($H(x)$ is defined in equation (2.44)). Also let us introduce the order parameters

$$Q_d = \kappa_*^2 + \kappa_1^2 p_d, \tag{5.29}$$

$$Q_A = \kappa_1^2 p_A, \tag{5.30}$$

$$Q_B = \kappa_*^2 q_0 + \kappa_1^2 p_B, \tag{5.31}$$

$$Q_O = \kappa_1^2 p_O. \tag{5.32}$$

Note that the expression for $\mathcal{G}_E$ does not depend on the p_C and p_E terms used in the ansatz, so when writing the saddle point equations with respect to p_C and p_E we get

$$\hat{p}_d + \hat{p}_C/K = O(K^{-1}), \tag{5.33}$$

$$\hat{p}_B + \hat{p}_E/K = O(K^{-1}). \tag{5.34}$$

From this we get that in the asymptotic $K\to\infty$ limit $\hat{p}_d = \hat{p}_B = 0$. Plugging this in the saddle point equations with respect to $\hat{p}_C$ and $\hat{p}_E$:

$$p_d + \frac{p_C}{K} = 2\frac{\partial G_{SE}}{\partial \hat{p}_C} = \frac{1+(2\hat{p}_B - \hat{p}_d)(q_0-1)^2}{(1+(\hat{p}_d-\hat{p}_B)(q_0-1))^2} + O(K^{-1}), \tag{5.35}$$

$$p_B + \frac{p_E}{K} = -2\frac{\partial G_{SE}}{\partial \hat{p}_E} = \frac{q_0+\hat{p}_B(q_0-1)^2}{(1+(\hat{p}_d-\hat{p}_B)(q_0-1))^2} + O(K^{-1}), \tag{5.36}$$

we find that $p_B = q_0$ and $p_d = 1$. Plugging these in equations (5.29)-(5.31)we get that $Q_d = 1$ and $Q_B = q_0$. Plugging these values into the variational free energy

$$\begin{aligned}\phi_{var} \underset{n\to 0, K\to\infty}{\to} &-\frac{1}{2}\alpha_D\hat{p}_C - \frac{1}{2}\alpha_D\hat{p}_A p_A + \frac{1}{2}\alpha_D\hat{p}_E q_0 + \frac{1}{2}\alpha_D\hat{p}_O p_O \\ &+ \frac{1}{2}\hat{q}_1 + \frac{1}{2}\hat{q}_0 q_0 - \alpha_D \hat{r} r + \alpha\mathcal{G}_E + \mathcal{G}_{SS} + \alpha_D\mathcal{G}_{SE}.\end{aligned} \tag{5.37}$$

This expression now needs to be numerically extremised, and the values of the order parameters at the saddle point can be used to calculate a number of observables.

For example, the *Generalization Error* can be calculated using the replica method like we did in chapter 3.3 for the training error

$$\epsilon_g = \lim_{N\to\infty} \mathbb{E}_{\mathcal{D}, y^{new}, \boldsymbol{x}^{new}} \Big\langle \Theta\left(y^{new} f_\theta(\boldsymbol{x}^{new})\right) \Big\rangle_\theta = \tag{5.38}$$

$$= \lim_{N\to\infty} \lim_{n\to 0} \mathbb{E}_{\mathcal{D}, y^{new}, \boldsymbol{x}^{new}} \mathcal{Z}^{n-1} \int \prod_i d\boldsymbol{w}_i \, e^{-\beta \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}^\mu), y^\mu)} \Theta\left(y^{new} f_\theta(\boldsymbol{x}^{new})\right). \tag{5.39}$$

Using Laplace's method we get

$$\epsilon_g = \int du\, d\boldsymbol{\lambda}\, P(u,\boldsymbol{\lambda})\, \Theta\Big(-u\frac{1}{\sqrt{K}}\sum_l c_l \sigma(\lambda_l)\Big) = \frac{1}{\pi}\text{arcotg}\left(\frac{\sqrt{\frac{2}{\pi}}M_*}{\sqrt{Q_{d*} + \frac{2}{\pi}(Q_{A*} - M_*^2)}}\right), \tag{5.40}$$

where with the expression M_* we mean the order parameter calculated in the saddle point given by the extremisation of equation (5.37). In figure 5.1 we show the generalization error as a function of α for two values of α_D in the $\beta \to \infty$ limit.

5.3 Spiked Hidden Manifold

In this section, we consider a teacher and student neural architecture for a regression task with one hidden layer. We denote with $\mathbf{x} \in \mathbb{R}^D$ the input data and with N the dimension of the

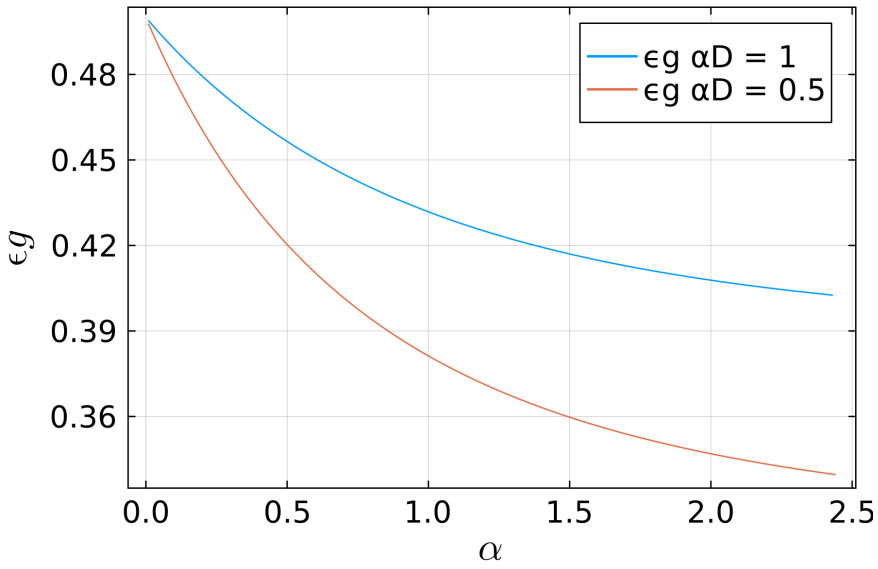

Figure 5.1: Generalization error for a tree-committee machine with $\varphi = \text{sign}$ as a function of α for fixed α_D in the $\beta \to \infty$ limit.

hidden layer of neurons. The first layer of weights is a matrix $F \in \mathbb{R}^{N\times D}$ and the second layer is given by the vector $\mathbf{w} \in \mathbb{R}^N$. The predictions are given by

$$\hat{y} = \frac{1}{\sqrt{N}}\mathbf{w} \cdot \varphi\left(\frac{1}{\sqrt{D}}F\mathbf{x}\right), \tag{5.41}$$

where $\varphi(\cdot)$ is the activation function.

We consider here a generalization of the Random Feature Model (RFM). In the RFM model, the matrix F is fixed to its random initialization and only the weights $\mathbf{w}$ are trained. In our model instead, we allow a low-rank correction to the first layer to be learned. In particular, we take F of the form

$$F = Z + \frac{\mathbf{u}\mathbf{v}^T}{\sqrt{N}}, \tag{5.42}$$

and assume $Z_{ij} \sim \mathcal{N}(0,1)$. The low rank correction is given by the learnable vectors $\mathbf{u} \in \mathbb{R}^N$ and $\mathbf{v} \in \mathbb{R}^D$. Note that, with this scaling, the contribution to the pre-activation vector $\frac{1}{\sqrt{D}}F\mathbf{x}$ given by the spike is of smaller order with respect to that given by the random feature projection. For the correction to be relevant then we need to require an alignment of $\boldsymbol{u}$ and $\boldsymbol{w}$, so that $\frac{1}{N}\sum_i w_i u_i \sim \mathcal{O}(1)$. Although we are considering a partly learnable feature matrix, because of this choice of scaling we cannot say we are exploring the feature learning regime. We call $\theta = (\mathbf{w}, \mathbf{u}, \mathbf{v})$ the set of parameters. We compactly write the student prediction as $\hat{y} = f_\theta(\mathbf{x})$.

We consider a teacher-student setting, where a teacher network denoted by parameters $\tilde{\theta} = (\tilde{\mathbf{w}}, \tilde{\mathbf{u}}, \tilde{\mathbf{v}})$, given a set of patterns $\{\boldsymbol{x}_\mu\}_{\mu=1}^P$, generates a set of labels according to $y_\mu = f_{\tilde{\theta}}(\boldsymbol{x}_\mu)$. In this section, we will always consider *iid* Gaussian patterns, $\mathbf{x}^\mu \sim \mathcal{N}(0, \mathbb{I}_D)$.

Given the dataset $\mathcal{D} = \{(\mathbf{x}^\mu, y^\mu)\}_{\mu=1}^P$, the student network learns by Empirical Risk Minimization, that is by minimizing

$$\theta = \arg\min_{\hat{\theta}} \sum_\mu \ell(f_{\hat{\theta}}(\mathbf{x}^\mu), y^\mu) + \lambda \|\hat{\theta}\|^2. \tag{5.43}$$

where $\ell(.,.)$ is the per-example loss functions, and λ regulates the amount of regularization. Given that we are considering the task of regression, we will be using the *Mean Square Error* $\ell(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$. We are interested in the training and generalization errors

$$\epsilon_t = \mathbb{E}_{\mathcal{D}} \Big\langle \frac{1}{P} \sum_\mu \ell\left(y_{\boldsymbol{\theta}}(\boldsymbol{x}^\mu), y\right) \Big\rangle, \tag{5.44}$$

$$\epsilon_g = \mathbb{E}_{\mathcal{D}, \{\boldsymbol{x}, y\}} \Big\langle \frac{1}{2} \Big(y - f_{\hat{\theta}}(\boldsymbol{x}) \Big)^2 \Big\rangle, \tag{5.45}$$

where the $\langle . \rangle$ represent averages with respect to the student defined in equation (5.43), and the $\mathbb{E}_{\{\boldsymbol{x}, y\}}$ in the generalization error indicates an average with respect to a new input-output pair not present in the training dataset. We are interested in the limits $N, D, P \to +\infty$ of the above quantities with fixed ratios $\alpha = \frac{P}{N}$ and $\alpha_D = \frac{D}{N}$.

An important observation is that the student has access to the fixed first layer matrix Z of the teacher, but not the spike nor the first layer weights. As mentioned before, we will chose a prior for the teacher weights such that the overlap $\tilde{q}^{wu} = \tilde{\mathbf{w}} \cdot \tilde{\mathbf{u}}/N$ is of order one. However, the student will not have such alignment in its prior, so although we are considering matching architectures, we are not in the Bayesian-Optimal case.

Note that the interpretation of this model is twofold. On one hand it can be seen as a model of fine tuning via the LoRA technique described in section 5.1: the dataset generated by the teacher with its rank-one spike in the feature matrix represents the new task, and the student needs to fine-tune to this new task using the spike and readout weights. On the other it can be seen as an inference task: we can ask how much data is necessary for the student to realize the dataset is generated by a teacher with a spike in its feature matrix, and align its spike to the one of the teacher.

5.3.1 Replica Analysis

In this section, we give an heuristic derivation of the replica calculation, and leave the full details to the appendix.

We want to calculate the quantity

$$\phi = \lim_{N\to\infty} \frac{1}{N}\mathbb{E}_{\mathcal{D}}\mathcal{Z}, \qquad \mathcal{Z} = \log \int d\boldsymbol{\theta}\, e^{-\beta \sum_{\mu} \ell\left(f_{\theta}(\boldsymbol{x}^{\mu}), f_{\tilde{\theta}}(\boldsymbol{x}^{\mu})\right) + \lambda \|\boldsymbol{\theta}\|^2}, \tag{5.46}$$

To take the average over the log we use the replica trick and thus have to average the "replicated" partition function $\mathcal{Z}^n$

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \mathbb{E}_{\boldsymbol{x}} \int \prod_{a=0}^{n} d\boldsymbol{\theta}_a\, e^{-\beta \sum_{a\mu} \ell\left(f_{\theta_a}(\boldsymbol{x}^{\mu}), f_{\theta_0}(\boldsymbol{x}^{\mu})\right) + \lambda \sum_{a=1}^{n} \|\boldsymbol{\theta}_a\|^2} P(\boldsymbol{\theta}_0), \tag{5.47}$$

where we have labeled the teacher network with the index $a = 0$, and used the expression $\mathbb{E}_{\boldsymbol{x}}$ to indicate the average over all patterns $\{\boldsymbol{x}_{\mu}\}_{\mu}$. Now, writing delta functions for the output of the network

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \int \prod_{\mu a} dy_{\mu a} \prod_{a} d\boldsymbol{\theta}_a\, e^{-\beta \sum_{a\mu} \ell\left(y_{\mu a}, y_{\mu 0}\right) + \lambda \sum_{a=1}^{n} \|\boldsymbol{\theta}_a\|^2} P(\boldsymbol{\theta}_0) \times \tag{5.48}$$

$$\times \prod_{\mu} \underbrace{\mathbb{E}_{\boldsymbol{x}} \prod_{a} \delta\left(y_{\mu a} - f_{\theta_a}(\boldsymbol{x}^{\mu})\right)}_{p\left(\{y_{\mu a}\}_a | \{\boldsymbol{\theta}_a\}_a\right)}.$$

Because each input-output pair $\{\boldsymbol{x}_{\mu}, y_{\mu}\}$ is *iid*, the distribution of the outputs will be the same for every pattern, so we just have to calculate the distribution $p\left(\{y_a\}_a | \{\boldsymbol{\theta}_a\}_a\right)$. In appendix 5.B we show that this distribution is Gaussian, with mean 0 and a certain correlation matrix Σ. Here we limit ourselves to giving an heuristic derivation of the correlation matrix.

As explained in [Goldt et al., 2020], Gaussian Equivalence is expected to hold if the term

$$\frac{1}{\sqrt{N}} \sum_{i} w_i F_{ij} = O(1). \tag{5.49}$$

As we can see for our choice of F we have that

$$\frac{1}{\sqrt{N}} \sum_{i} w_i F_{ij} = \frac{1}{\sqrt{N}} \sum_{i} w_i Z_{ij} + \frac{v_j}{N} \sum_{i} w_i u_i, \tag{5.50}$$

which is of order one. When Gaussian Equivalence holds, the error obtained by a linear network trained on random features $\varphi\left(\frac{1}{\sqrt{D}} F \boldsymbol{x}^{\mu}\right)$ is the same as that trained on the inputs

$$\phi^{e}(\boldsymbol{x}) = \kappa_0 \mathbf{1} + \kappa_1 \frac{1}{\sqrt{D}} F \boldsymbol{x}^{\mu} + \kappa_* \xi^{\mu}, \tag{5.51}$$

where $\mathbf{1}$ is the all ones vector, $\boldsymbol{\xi}^\mu \sim \mathcal{N}(0, \mathbb{I}_N)$ is Gaussian noise, and the scalars $\kappa_0, \kappa_1, \kappa_*$ are

$$\kappa_0 = \int Dz\, \varphi(z), \tag{5.52}$$

$$\kappa_1 = \int Dz\, z\varphi(z), \tag{5.53}$$

$$\kappa_2 = \int Dz\, z^2\varphi(z), \tag{5.54}$$

$$\kappa_* = \sqrt{\kappa_2 - \kappa_1^2 - \kappa_0^2}. \tag{5.55}$$

Applying this equivalent model to our setting, we get that

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \kappa_0 \frac{\mathbf{w}\cdot\mathbf{1}}{\sqrt{N}} + \kappa_1 \frac{\mathbf{w}^T Z \mathbf{x}}{\sqrt{ND}} + \kappa_* \frac{\mathbf{w}\cdot\boldsymbol{\xi}}{\sqrt{N}} + \kappa_1 \frac{\mathbf{w}\cdot\mathbf{u}}{N}\frac{\mathbf{v}\cdot\boldsymbol{x}}{\sqrt{D}}. \tag{5.56}$$

Let us from now on make the assumption $\kappa_0 = 0$. The correlation between iid replicas of the distribution is given by

$$\Sigma_{ab} = \mathbb{E}_{\boldsymbol{\xi},\boldsymbol{x}} f_{\boldsymbol{\theta}_a}(\boldsymbol{x}) f_{\boldsymbol{\theta}_b}(\boldsymbol{x}) = \kappa^* q_{ab}^w + \kappa_1^2 q_{ab}^Z + \kappa_1^2 q_a^{wu} q_{ab}^v q_b^{wu} + \frac{\kappa_1^2}{\sqrt{\alpha_D}}(q_{ba}^{wv} q_a^{wu} + q_{ab}^{wv} q_b^{wu}), \tag{5.57}$$

where we have defined the overlaps

$$q_{ab} = \frac{\mathbf{w}_a \cdot \mathbf{w}_b}{N}, \tag{5.58}$$

$$q_{ab}^Z = \frac{\mathbf{w}_a^T Z^T Z \mathbf{w}_b}{ND}, \tag{5.59}$$

$$q_{ab}^v = \frac{\mathbf{v}_a \cdot \mathbf{v}_b}{D}, \tag{5.60}$$

$$q_{ab}^{wv} = \frac{\mathbf{w}_a^T Z \mathbf{v}_b}{N\sqrt{D}}, \tag{5.61}$$

$$q_a^{wu} = \frac{\mathbf{w}_a \cdot \mathbf{u}_a}{N}. \tag{5.62}$$

Now substituting our expression for the output distribution in equation (5.48) we get

$$\mathbb{E}_{\mathcal{D}} \mathcal{Z}^n = \mathbb{E}_Z \int \prod_{\mu a} dy_{\mu a} \prod_{a=0}^{n} d\boldsymbol{\theta}_a\, e^{-\beta \sum_{a\mu} \ell\left(y_{\mu a}, y_{\mu 0}\right) + \lambda \sum_{a=1}^n \|\boldsymbol{\theta}_a\|^2} P(\boldsymbol{\theta}_0) \prod_{\mu} \mathcal{N}\left(\{y_{\mu a}\}_a; \mathbf{0}, \Sigma_{ab}\right) \tag{5.63}$$

Introducing order parameters defined in equations (5.58) - (5.62) we get

$$\mathbb{E}_{\mathcal{D}}\mathcal{Z}^n = \int \prod_{ab} dq_{ab}\, d\hat{q}_{ab}\, dq^Z_{ab}\, d\hat{q}^Z_{ab}\, dq^v_{ab}\, d\hat{q}^v_{ab}\, dq^{wv}_{ab}\, d\hat{q}^{wv}_{ab} \prod_a dq^{wu}_a\, d\hat{q}^{wu}_a\, e^{nN\phi}, \tag{5.64}$$

$$\phi = \frac{1}{n}\left(G_I + G_S + \alpha G_E\right), \tag{5.65}$$

$$G_I = \sum_a q^{wu}_a \hat{q}^{wu}_a + \alpha_D \sum_{ab} q^v_{ab}\hat{q}^v_{ab} + \sum_{ab} q^w_{ab}\hat{q}^w_{ab} + \sum_{ab} q^Z_{ab}\hat{q}^Z_{ab} + \sum_{ab} q^{wv}_{ab}\hat{q}^{wv}_{ab}, \tag{5.66}$$

$$G_S = \frac{1}{N}\log \mathbb{E}_Z \int \prod_{a=0}^{n} d\boldsymbol{w}_a d\boldsymbol{u}_a d\boldsymbol{v}_a\, P(\boldsymbol{\theta}_0) e^{-\sum_{ab} \boldsymbol{w}_a^T\left(\hat{q}^{ab}_w \mathbb{I}_N + \hat{Q}^{ab}_Z \Omega\right)\boldsymbol{w}_b - \sum_a \hat{q}^a_{wu} \boldsymbol{w}_a^T \boldsymbol{u}^a} \times \tag{5.67}$$

$$\times\, e^{-\sum_{ab} \hat{q}^{wv}_{ab} \boldsymbol{w}_a^T \frac{Z\boldsymbol{v}_b}{\sqrt{D}} - \sum_{ab} \hat{q}^{ab}_v \boldsymbol{v}_a^T \boldsymbol{v}_b + \lambda \sum_{a=1}^n \left(\|\boldsymbol{w}_a\|^2 + \|\boldsymbol{u}_a\|^2 + \|\boldsymbol{v}_a\|^2\right)},$$

$$G_E = \log \int \prod_{a=0}^{n} \frac{dy^a}{\sqrt{2\pi}} \frac{1}{\sqrt{\det(\Sigma)}} e^{-\frac{1}{2}\sum_{ab} y_a \Sigma^{-1}_{ab} y_b - \beta \sum_{a=1}^n \ell\left(y^0, y^a\right)}, \tag{5.68}$$

where we have defined the matrix $\Omega = \frac{ZZ^T}{D}$.
Next, we need to make an assumption on the ansatz of our order parameters. We will work in the *Replica-Symmetric* ansatz, that is we will assume for every matrix q_{ab} and $\hat{q}_{ab}$ the form

$$q_{ab} = \begin{cases} (q + \delta q)\delta_{ab} + q(1 - \delta_{ab}), & a, b \geq 0 \\ m, & a = 0 \text{ xor } b = 0 \end{cases} \tag{5.69}$$

and for every vector $q^{wu}_a = q^{wu}$ for $a \geq 1$ and $q^{wu}_0 = \tilde{q}^{wu}$. Special attention needs to payed to the q^{wv}_{ab} order parameter: indeed the value for $a = 0$ need not be equal to the value for $b = 0$, so we will use two values m^{vw} and m^{wv}. Note that the replica symmetric assumption is not a priori justified, since we are working in a mismatched scenario (the student prior is different from the teacher prior), so we will have to check a posteriori if it holds.

Using Laplace's method and taking the $n \to 0$ limit we reach the final expression

$$\phi = \underset{\substack{\hat{q}, \hat{q}^Z, \hat{q}^v, \hat{q}^{wv}, \hat{q}^{wu} \\ q, q^Z, q^v, q^{wv}, q^{wu}}}{\text{extr}} \left\{\mathcal{G}_I + \mathcal{G}_S + \mathcal{G}_E\right\}, \tag{5.70}$$

where for all three functions $\mathcal{G}_{I,S,E} = \lim_{n\to 0} \frac{G_{I,S,E}}{n}$. The extremisation is over 27 scalar order parameters. In appendix 5.C we derive explicit forms for these functions for a choice of the loss function $\ell(.,.)$, while here we limit ourselves to showing the results.

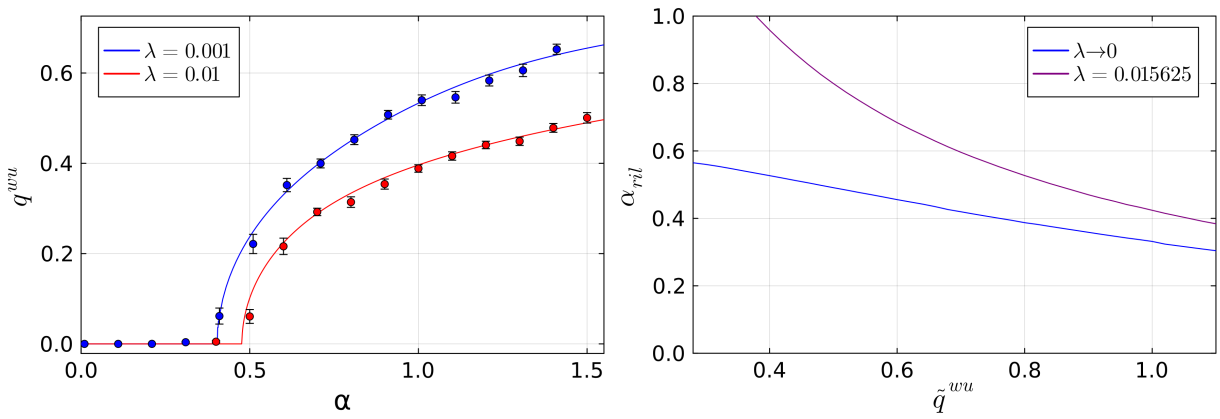

Figure 5.2: On the left, the student q_{wu} overlap as a function of α when $\tilde{q}_{wu} = 0.8$ for different values of λ. Points with relative error bars are given by simulations using full batch gradient descent with $D = 300$, trained for a total of 10^6 epochs. Each point is averaged over 20 runs. On the right α_{ril} as a function of $\tilde{q}_{wu}$ both for finite λ and in the limit $\lambda \to 0$. In both plots $\alpha_D = 0.5$.

5.3.2 Results

As specified above, we choose a prior for the teacher which aligns $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{u}}$ by a quantity $\tilde{q}^{wu} = \tilde{\mathbf{w}} \cdot \tilde{\mathbf{u}}/N$, while we put no such prior on the student. The student will have to thus infer the presence of this alignment from the data. In figure 5.2 (left) we plot the student overlap between $\mathbf{w}$ and $\mathbf{u}$ as a function α. As we can see, it takes a minimum quantity of data α_{ril} for this overlap to become greater than zero. Because of the scaling we choose for the spike, for $\alpha < \alpha_{ril}$ the student effectively acts as if no low rank correction were present, while for $\alpha > \alpha_{ril}$ the student starts to align in a such a way to use it. The value of this detection threshold α_{ril} depends on the strength of the teacher spike $\tilde{q}^{wu}$, the dimensionality ratio between first and second layer α_D and the regularization strength λ. By empirically noticing that the function $\alpha_{ril}(\lambda)$ goes linearly to a value $\alpha_{ril}(0)$ for $\lambda \to 0$, it is also possible to extract the zero-regularization limit by performing a linear fit. Interestingly, in this limit the same qualitative picture as the finite λ case holds, highlighting the fact that this detection transition is not a trivial consequence of the balancing of loss minimization and regularization. Indeed, in figure 5.2 (right) we plot the value of this threshold α_{ril} as a function of $\tilde{q}^{wu}$ for fixed α_D, both in the finite λ and $\lambda \to 0$ case.

In figure 5.3 we plot the generalization and training errors as a function of α, for a student network with and without the rank-one correction. As we can see both observables coincide until the threshold α_{ril}, a trivial consequence of the fact that in this scaling regime when $\boldsymbol{w}$ and $\boldsymbol{u}$ aren't aligned, the low rank correction has no effect on the output. For $\alpha > \alpha_{ril}$, the two networks show different behaviours. The spike-less student exhibits the well established

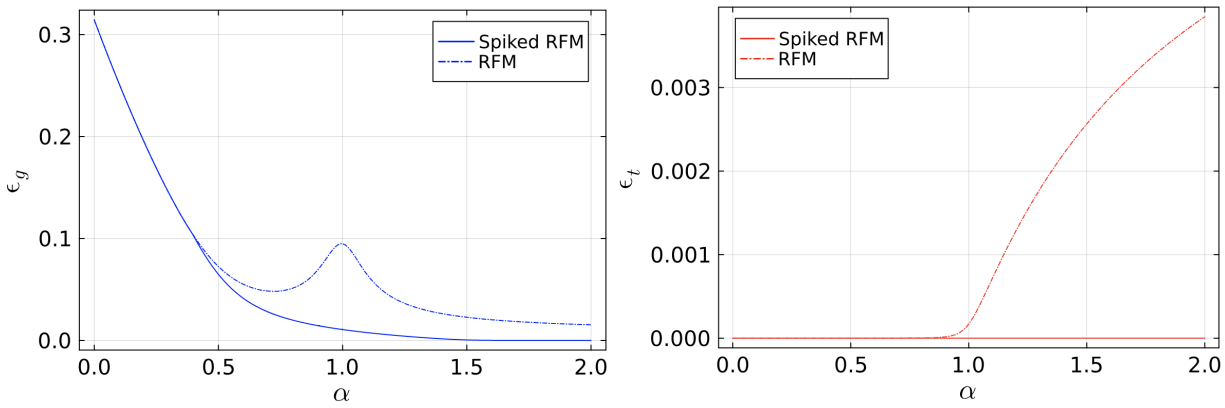

Figure 5.3: Training (right) and Generalization (left) Errors for Regression on a dataset generated by a Spiked RFM teacher when the student is also a Spiked RFM (solid line) and when it is just a RFM (dashed line). Here $\lambda = 10^{-4}$, $\alpha_D = 0.5$ and $\tilde{q}_{wu} = 0.8$.

Double Descent, where the generalization error has a peak at $\alpha = 1$, a consequence of the fact that the last term in equation 5.56 for the teacher output is seen by the spike-less student as noise. The same is not observed for the spike-full student, where instead the training error always remains close to zero while the generalization error drops to zero at $\alpha = 1+\alpha_D$. Indeed, this is to be expected since in this case the architectures of teacher and student are matched (although the priors are different), and the number of parameters to fit is $N + D$ (the $\boldsymbol{u}$ needs only to be aligned with $\boldsymbol{w}$), which according to traditional statistical learning wisdom requires $P = N + D$ examples to be perfectly learned.

5.3.3 Experiments on Real Datasets

In the previous section we saw that for Gaussian data in the teacher-student scenario, it takes a minimum amount of data for the the network to use its Low Rank correction to the feature matrix. A natural question is whether this detection phenomenology can be observed in real world problems. To verify this, we perform a simple experiment on the MNIST dataset, that can be divided in two steps:

- We pretrain a two layer neural network on the MNIST dataset, on a regression task, i.e. for a picture representing a handwritten 2, the output of the network has to be as close as possible to the value 2 (in reality we divide labels by 10 for numeric reasons, so for a picture of the digit 2 the label is 0.2). We use the MSE loss with small regularization, and use digits 1 through 6.

- We then train a Spiked RFM defined in equation (5.41), where the random feature matrix is equal to the first layer of the pretrained network. We thus train the first layer weights

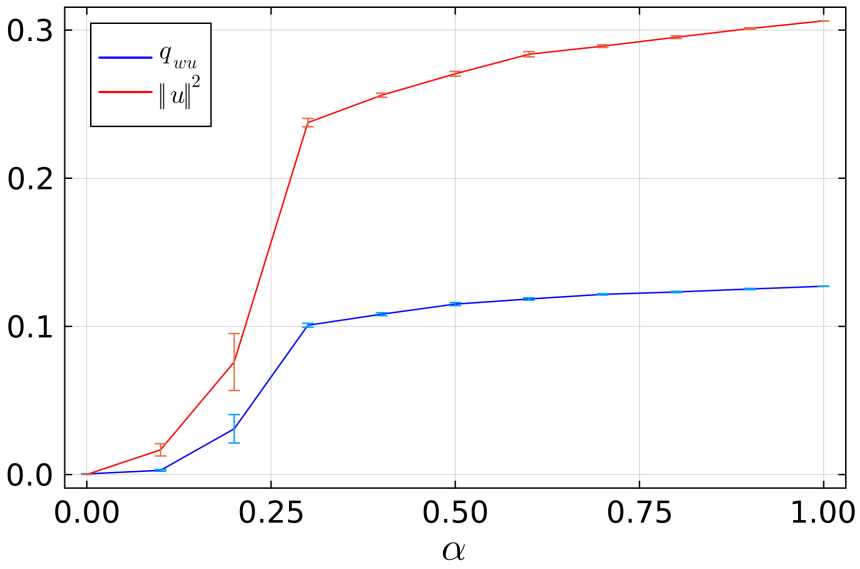

Figure 5.4: Alignment between $\boldsymbol{u}$ and $\boldsymbol{w}$ and norm of $\boldsymbol{u}$ as a function of α, the fraction of number 7 patterns used for the training in the fine-tuning task. We trained a Spiked RFM $N = 1000$, $D = 784$ (the input size of MNIST), tanh activation function and with F taken from the pretraining. For the optimization we used SGD with $\lambda = 10^{-5}$ for 2000 epochs. Each point is an average over 10 runs.

$\boldsymbol{w}$, and the spikes $\boldsymbol{u}$ and $\boldsymbol{v}$. As is standard in LoRA applications, we initialize the spike with $\boldsymbol{v}_{init} = \mathbf{0}$ and the second layer weights $\boldsymbol{w}$ with the weights of the pretrained model. The network is trained on the same task as before, with the difference that now we add a fraction α of the number 7 digits.

Clearly, the network "fine-tunes" by aligning $\boldsymbol{v}$ with a prototype of the 7 digit, and $\boldsymbol{u}$ with $\boldsymbol{w}$ so that the perturbation is of the right order. In figure 5.4 we show the norm of $\boldsymbol{u}$ and the alignment q_{wu} as a function of α. As we can see two regimes can be identified: a first regime for low α, where the network doesn't use the spike, and a regime for higher α where the alignment q_{wu} becomes greater than zero. Again we observe the same phenomenon: it takes a minimum amount of data for the network to use its spike to fine-tune. There is however an important difference with respect to the Gaussian case: this phenomenon disappears in the limit of zero regularization. Although the value of λ used in the experiments is not unrealistic ($\lambda = 10^{-5}$), without regularization this detection phenomenon completely disappears.

5.4 Conclusions

In this chapter, we have studied two models that go beyond the storage setting. First we have seen how a two layer network, the *Tree-Committee Machine*, performs on data generated in the hidden manifold framework, in the limit of many hidden units. Although the only difference with respect to the original *Hidden Manifold Model* is the architecture of the network after the random feature projection, this represents an important modeling shift as the *Tree-Committe Machine* is non-convex, a feature which is known to hold for deep networks.

Then, we have studied a setting which goes beyond the random feature model, by adding a rank-one learnable perturbation of the feature matrix. Although because of our choice of scaling we are still the lazy regime, this represents an important step to understanding learning in multilayer neural networks. By analyzing a teacher-student scenario as a model for fine-tuning using the LoRA technique, we have seen how such spike in a teacher can be detected by the student only beyond a certain threshold α_{ril}, and how double descent is mitigated when this spike is detected. Finally, we have observed a similar phenomenon in a real world problem involving fine-tuning using LoRA.

Appendix

5.A Replica Calculation for the Hidden Manifold Tree-Committee Machine

In this section, we will go over the details of the replica calculation that leads to the expressions cited in section 5.2. The expressions for the $\mathcal{G}_{SS}$ and $\mathcal{G}_{SE}$ can be derived straightforwardly applying the techniques outlined in the previous chapters, so we will not derive them here. The expression for $\mathcal{G}_E$ instead requires more attention.

We start with

$$G_E = \log \int \prod_a \frac{d\hat{\boldsymbol{\lambda}}^a d\boldsymbol{\lambda}^a}{(2\pi)^K} \frac{d\hat{u}du}{2\pi} e^{-\beta \sum_a \Theta\left(-\text{sign}(u)\frac{1}{\sqrt{K}}\sum_l c_l \varphi(\lambda_l^a)\right) + i\hat{u}u + i\sum_a (\boldsymbol{\lambda}^a)^T \hat{\boldsymbol{\lambda}}^a} \times \\ \times e^{-\frac{1}{2}(\hat{u})^2 - \frac{1}{2}\sum_{ab}\hat{\boldsymbol{\lambda}}^a Q^{ab} \hat{\boldsymbol{\lambda}}^b - \hat{u}\frac{\mu_1 r}{\sqrt{K}}\sum_a (\hat{\boldsymbol{\lambda}}^a)^T \mathbf{1}_K}, \tag{5.71}$$

where $\forall a, b \in [n]\, Q^{ab} \in \mathbb{R}^{K\times K}$, and with $\mathbf{1}_K$ we indicate the all ones vector in dimension K. Let us call

$$Q_{lm}^{ab} = \begin{cases} R_{lm}, & a = b \\ Q_{lm}, & a \neq b \end{cases} \tag{5.72}$$

$$R_{lm} = \begin{cases} Q_D + Q_C/K, & if \quad l = m \\ Q_A/K, & if \quad l \neq m \end{cases} \tag{5.73}$$

$$Q_{lm} = \begin{cases} Q_B + Q_E/K, & if \quad l = m \\ Q_O/K. & if \quad l \neq m \end{cases} \tag{5.74}$$

Then, integrating in $\hat{u}$

$$G_E = \log \int \prod_a \frac{d\hat{\boldsymbol{\lambda}}^a d\boldsymbol{\lambda}^a}{(2\pi)^K} \frac{du}{\sqrt{2\pi}} e^{-\beta \sum_a \Theta\left(-\text{sign}(u)\frac{1}{\sqrt{K}}\sum_l c_l \varphi(\lambda_l^a)\right) + \sum_a i(\boldsymbol{\lambda}^a)^T \hat{\boldsymbol{\lambda}}^a - \frac{1}{2}\sum_a \hat{\boldsymbol{\lambda}}^a (R-Q)\hat{\boldsymbol{\lambda}}^b} \times \\ \times e^{-\frac{1}{2}\sum_{ab}\hat{\boldsymbol{\lambda}}^a Q \hat{\boldsymbol{\lambda}}^b - \frac{1}{2}\left(u + i\frac{\mu_1 r}{\sqrt{K}}\mathbf{1}_K^T(\sum_a \hat{\boldsymbol{\lambda}}^a)\right)^2}. \tag{5.75}$$

Substituting the values of matrix Q

$$G_E = \log \int \prod_a \frac{d\hat{\boldsymbol{\lambda}}^a d\boldsymbol{\lambda}^a}{(2\pi)^K} Du\, e^{-\beta \sum_a \Theta\left(-\text{sign}(u)\frac{1}{\sqrt{K}}\sum_l c_l \varphi(\lambda_l^a)\right) + i\sum_a (\boldsymbol{\lambda}^a)^T \hat{\boldsymbol{\lambda}}^a - \frac{1}{2}(\mathbf{1}_K^T \sum_a \boldsymbol{\lambda}^a)^2 \left(\frac{Q_O}{K} - \frac{\mu_1^2 r^2}{K}\right)} \times \\ \times e^{-\frac{1}{2}\sum_a \hat{\boldsymbol{\lambda}}^a (R-Q)\hat{\boldsymbol{\lambda}}^a - \frac{1}{2}\sum_{ab}\hat{\boldsymbol{\lambda}}^a \cdot \hat{\boldsymbol{\lambda}}^b Q_B - iu\frac{\mu_1 r}{\sqrt{K}}\mathbf{1}_K^T(\sum_a \hat{\boldsymbol{\lambda}}^a)}. \tag{5.76}$$

Performing a standard and a vector Hubbard-Stratonovich transform ($e^{-\frac{1}{2}\boldsymbol{v}^T A \boldsymbol{v}} = \int D\boldsymbol{\xi} e^{i\boldsymbol{v}^T A^{1/2}\boldsymbol{\xi}}$), we can decouple replicas

$$G_E = \log \int D\boldsymbol{\xi} D\eta Du \times \\ \times \left(\int \frac{d\hat{\boldsymbol{\lambda}} d\boldsymbol{\lambda}}{(2\pi)^K} e^{-\beta\Theta\left(-\text{sign}(u)\frac{1}{\sqrt{K}}\sum_l c_l \varphi(\lambda_l^a)\right) + i(\boldsymbol{\lambda})^T \hat{\boldsymbol{\lambda}} + i\eta \mathbf{1}_K^T \hat{\boldsymbol{\lambda}} \sqrt{\frac{Q_O}{K} - \frac{\mu_1^2 r^2}{K}} - \frac{1}{2}\hat{\boldsymbol{\lambda}}(R-Q)\hat{\boldsymbol{\lambda}} - i\boldsymbol{\xi}^T \hat{\boldsymbol{\lambda}}\sqrt{Q_B} - iu\frac{\mu_1 r}{\sqrt{K}}\mathbf{1}_K^T \hat{\boldsymbol{\lambda}}} \right)^n. \tag{5.77}$$

Performing the integral in $\hat{\boldsymbol{\lambda}}$

$$G_E = \log \int D\boldsymbol{\xi} D\eta Du \times \\ \times \left(\int Dz D\boldsymbol{\lambda} e^{-\beta\Theta\left[-\text{sign}(u)\frac{1}{\sqrt{K}}\sum_l c_l \varphi\left(\sqrt{Q_d - Q_B}\lambda_l - z\sqrt{\frac{Q_A - Q_O}{K}} + \xi_l\sqrt{Q_B} - \eta\sqrt{\frac{Q_O - M^2}{K}} + \frac{uM}{\sqrt{K}}\right)\right]} \right)^n. \tag{5.78}$$

Taylor expanding the nonlinearity, and taking the $n \to 0$ limit

$$\mathcal{G}_E = \lim_{n\to 0}\frac{G_E}{n} = \int D\boldsymbol{\xi} D\eta Du \log \int Dz D\boldsymbol{\lambda} \times \\ \times e^{-\beta\Theta\left(-\text{sign}(u)\frac{1}{\sqrt{K}}\sum_l c_l \varphi\left(\sqrt{Q_d - Q_B}\lambda_l + \xi_l\sqrt{Q_B}\right) - \text{sign}(u)\frac{1}{K}\sum_l c_l \varphi'\left(\sqrt{Q_d - Q_B}\lambda_l + \xi_l\sqrt{Q_B}\right)\left(-z\sqrt{Q_A - Q_O} - \eta\sqrt{Q_O - M^2} + uM\right)\right)}. \tag{5.79}$$

Let us now define

$$G = \frac{1}{\sqrt{K}} \sum_l c_l \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B}\Big), \tag{5.80}$$

$$F = \frac{1}{K} \sum_l c_l \varphi'\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B}\Big). \tag{5.81}$$

In the large K limit, F will concentrate to a deterministic value while G will still fluctuate. Indeed introducing delta functions and using their spectral definition, and calling I the argument of the log in equation (5.79)

$$I = \int dF d\hat{F} dG d\hat{G} Dz D\boldsymbol{\lambda} e^{-\beta\Theta\Big(-\text{sign}(u)G - \text{sign}(u)F\Big(-z\sqrt{Q_A - Q_O} - \eta\sqrt{Q_O - M^2} + uM\Big)\Big) + iM\hat{M} + iF\hat{F}} \times \tag{5.82}$$
$$\times\, e^{-i\hat{G}\frac{1}{\sqrt{K}}\sum_l c_l \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l\sqrt{Q_B}\Big) - i\hat{F}\frac{1}{K}\sum_l c_l \varphi'\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l\sqrt{Q_B}\Big)}.$$

Now expanding the exponentials to second order and averaging with respect to $\boldsymbol{\lambda}$

$$I \approx \int dF d\hat{F} dG d\hat{G} Dz D\boldsymbol{\lambda} e^{-\beta\Theta\Big(-\text{sign}(u)G - \text{sign}(u)F\Big(-z\sqrt{Q_A - Q_O} - \eta\sqrt{Q_O - M^2} + uM\Big)\Big) + iM\hat{M} + iF\hat{F}} \times \tag{5.83}$$
$$\times \prod_l \left(1 - i\hat{G}\frac{1}{\sqrt{K}} c_l \langle \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l\sqrt{Q_B}\Big)\rangle_\lambda - \frac{1}{2}\hat{G}^2 \frac{1}{K} c_l^2 \langle \varphi^2\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l\sqrt{Q_B}\Big)\rangle_\lambda\right) \times$$
$$\times \prod_l \left(1 - i\hat{F}\frac{1}{K} c_l \langle \varphi'\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l\sqrt{Q_B}\Big)\rangle_\lambda\right).$$

Now re-exponentiating and integrating in $d\hat{F}$ and $d\hat{G}$

$$I = \int DGDz e^{-\beta\Theta\Big(-\text{sign}(u)(\sigma G + \mu) - \text{sign}(u)F\Big(-z\sqrt{Q_A - Q_O} - \eta\sqrt{Q_O - M^2} + uM\Big)\Big)}, \tag{5.84}$$

with

$$\mu = \frac{1}{\sqrt{K}} \sum_l c_l \langle \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B} \Big) \rangle_\lambda, \tag{5.85}$$

$$\sigma^2 = \frac{1}{K} \sum_l c_l^2 \Big(\langle \varphi^2\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B} \Big) \rangle_\lambda - \langle \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B} \Big) \rangle_\lambda^2 \Big), \tag{5.86}$$

$$F = \frac{1}{K} \sum_l c_l \langle \varphi'\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B} \Big) \rangle_\lambda. \tag{5.87}$$

Using the identities

$$\int Dx\, e^{-\beta\Theta(ax+b)} = H_\beta\left(\frac{b}{|a|} \right), \tag{5.88}$$

$$\int Dx\, H(a + bx) = H\left(\frac{a}{\sqrt{1+b^2}} \right), \tag{5.89}$$

we can integrate in G and z

$$\mathcal{G}_E = \int D\boldsymbol{\xi} D\eta Du \log H_\beta \left(\frac{\text{sign}(u) F(\boldsymbol{\xi}) \Big(\eta \sqrt{Q_O - M^2} - uM \Big) - \text{sign}(u)\mu(\boldsymbol{\xi})}{\sqrt{\sigma^2(\boldsymbol{\xi}) + (Q_A - Q_O)F^2(\boldsymbol{\xi})}} \right). \tag{5.90}$$

The last integral to perform is in $D\boldsymbol{\xi}$. Again, we introduce delta functions for μ, σ^2 and F. The only term that will fluctuate is μ, so we can write directly

$$\mathcal{G}_E = \int D\mu D\eta Du \log H_\beta \left(\frac{\text{sign}(u) F\Big(\eta \sqrt{Q_O - M^2} - uM \Big) - \text{sign}(u)(\Delta\mu + \bar{\mu})}{\sqrt{\sigma^2 + (Q_A - Q_O)F^2}} \right), \tag{5.91}$$

with

$$\bar{\mu} = \frac{1}{\sqrt{K}} \sum_l c_l \langle\langle \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B}\Big)\rangle_\lambda\rangle_\xi, \tag{5.92}$$

$$\Delta^2 = \frac{1}{K} \sum_l c_l^2 \Bigg(\langle\langle \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B}\Big)\rangle_\lambda^2\rangle_\xi - \langle\langle \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B}\Big)\rangle_\lambda\rangle_\xi^2 \Bigg), \tag{5.93}$$

$$\sigma^2 = \frac{1}{K} \sum_l c_l^2 \Big(\langle\langle \varphi^2\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B}\Big)\rangle_\lambda\rangle_\xi - \langle\langle \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B}\Big)\rangle_\lambda^2\rangle_\xi \Big), \tag{5.94}$$

$$F = \frac{1}{K} \sum_l c_l \langle\langle \varphi'\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B}\Big)\rangle_\lambda\rangle_\xi. \tag{5.95}$$

Let us choose $\varphi(x) = \text{sign}(x)$ and $c_l = 1 \ \forall l \in [K]$. Then

$$\bar{\mu} = \langle\langle \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B}\Big)\rangle_\lambda\rangle_\xi = \langle \varphi(\sqrt{Q_d} z)\rangle_z = 0, \tag{5.96}$$

$$\Delta^2 = \langle\langle \varphi\Big(\sqrt{Q_d - Q_B}\lambda_l + \xi_l \sqrt{Q_B}\Big)\rangle_\lambda^2\rangle_\xi = \left\langle \left(-1 + 2H\left(-\sqrt{\frac{Q_B}{Q_d - Q_B}}\xi \right) \right)^2 \right\rangle_\xi = \tag{5.97}$$

$$= -1 + \frac{4}{\pi} \arctan\Big(\sqrt{\frac{Q_d + Q_B}{Q_d - Q_B}}\Big) = 1 - \frac{4}{\pi} \arctan\Big(\sqrt{\frac{Q_d - Q_B}{Q_B + Q_d}}\Big), \tag{5.98}$$

$$\sigma^2 = \frac{4}{\pi} \arctan\Big(\sqrt{\frac{Q_d - Q_B}{Q_B + Q_d}}\Big), \tag{5.99}$$

$$F = \langle \varphi'(\sqrt{Q_d} z)\rangle_z = \sqrt{\frac{2}{\pi Q_d}}. \tag{5.100}$$

Finally performing rotations in u, μ and η we get to the final expression reported in the main text

$$\mathcal{G}_E = 2 \int D\eta\, H\left(-\frac{\sqrt{\frac{2}{\pi Q_d}} M}{\sqrt{\frac{2}{\pi}\frac{Q_0 - M^2}{Q_d} + 1 - \frac{4}{\pi}\arctan\Big(\sqrt{\frac{Q_d - Q_B}{Q_B + Q_d}}\Big)}} \eta \right) \times \tag{5.101}$$

$$\times \log H_\beta \left(\frac{-\sqrt{\frac{2Q_0}{\pi Q_d} + 1 - \frac{4}{\pi}\arctan\Big(\sqrt{\frac{Q_d - Q_B}{Q_B + Q_d}}\Big)}\,\eta}{\sqrt{\frac{4}{\pi}\arctan\Big(\sqrt{\frac{Q_d - Q_B}{Q_B + Q_d}}\Big) + \frac{2}{\pi}\frac{Q_A - Q_O}{Q_d}}} \right).$$

5.B Conditional Gaussian Equivalence

In this section, we will show that the distribution $p(\{y_a\}_a|\{\boldsymbol{\theta}_a\}_a)$ defined in equation (5.48) is Gaussian, and will calculate the value of its correlation matrix. For the sake of clarity we will use the notation $\bar{y} = \{y_a\}_{a=0}^n$ for all variables that depend on the replica index a.
First off we introduce the variables

$$g_{\mu a} = \frac{\mathbf{v}^a \cdot \mathbf{x}^\mu}{\sqrt{D}}, \tag{5.102}$$

which conditionally on the parameters of the network $\bar{\boldsymbol{\theta}}$ are Gaussian distributed with mean zero and covariance matrix $\Sigma^v_{ab} = q^v_{ab}$. In the following, all distributions will be conditioned on $\bar{\boldsymbol{\theta}}$, so for notation's sake we won't explicitly write it.
We claim that if we consider the conditional distribution

$$p(\bar{y}|\bar{g}) = \int d\boldsymbol{x}\, p(\bar{y}|\mathbf{x},\bar{g})\, p(\mathbf{x}|\bar{g}), \tag{5.103}$$

the conditional GET applies, that is this distribution is again Gaussian. To see this, let us calculate the distribution of $\mathbf{x}$ conditioned on $\bar{g}$. We have that

$$\mathbb{E}_{\mathbf{x}|\bar{g}} x_j = \frac{1}{\sqrt{D}} \bar{g}^T (\bar{q}^v)^{-1} \bar{v}_j, \tag{5.104}$$

$$\mathbb{E}_{\mathbf{x}|g} x_j x_k - \mathbb{E}_{\mathbf{x}|g} x_j \mathbb{E}_{\mathbf{x}|g} x_k = \delta_{jk} - \frac{1}{D} \bar{v}_j (q^v)^{-1} \bar{v}_k. \tag{5.105}$$

Introducing the pre-activations

$$\boldsymbol{\lambda}^a = \frac{1}{\sqrt{D}} Z\mathbf{x} + \frac{1}{\sqrt{N}} g_a \mathbf{u}^a, \tag{5.106}$$

we have that their first moment is at leading order

$$\mathbb{E}_{\mathbf{x}|\bar{g}}\, \lambda_i^a = \frac{1}{\sqrt{N}} M_i^a + O\left(\frac{1}{D}\right), \tag{5.107}$$

$$M_i^a = g_a u_i^a + \frac{1}{\sqrt{D\alpha_D}} \sum_{bc} \sum_j g_b (\bar{q}^v)^{-1}_{bc} Z_{ij} v_j^c. \tag{5.108}$$

Their second moment instead

$$\mathbb{E}_{\mathbf{x}|\bar{g}}\,\lambda_i^a\lambda_j^b - \mathbb{E}_{\mathbf{x}|\bar{g}}\,\lambda_i^a\mathbb{E}_{\mathbf{x}|\bar{g}}\lambda_j^b = \frac{1}{D}\sum_k Z_{ik}Z_{jk} - \frac{1}{D^2}\sum_{cd}(\sum_k Z_{ik}v_k^c)(q^v)^{-1}_{cd}(\sum_k Z_{jk}v_k^d). \tag{5.109}$$

By the law of large numbers variables λ_{ia} are Gaussian. Furthermore, their covariance is of order 1 if $i = j$ and of order $1/\sqrt{N}$ for $i \neq j$. This weak correlation structure, together with the fact that the first moments of the pre-activations are small, allows us to apply the GET. Indeed, the post-activations can always be taylor expanded as $\varphi(\lambda_i^a) \approx \varphi(\hat{\lambda}_i^a) + \varphi'(\hat{\lambda}_i^a)\frac{M_{ia}}{\sqrt{N}}$, where the $\hat{\lambda}_i^a = \lambda_i^a - \mathbb{E}_{\mathbf{x}|\bar{g}}\lambda_i^a$ The GET can be applied to these Gaussian variables. For example the first moment of the post-activations is

$$\mathbb{E}_{\mathbf{x}|\bar{g}}\varphi(\lambda_i^a) = \kappa_0 + \kappa_1\frac{1}{\sqrt{N}}M_i^a + O(\frac{1}{D}). \tag{5.110}$$

From this we get that the first moment of the output is

$$\mathbb{E}_{\mathbf{x}|\bar{g}}\; y^a = \mathbb{E}_{\bar{g}}\frac{1}{\sqrt{N}}\sum_i w_i^a\,\mathbb{E}_{\mathbf{x}|\bar{g}}\varphi(\lambda_i^a) = \tag{5.111}$$

$$= \kappa_0\left(\frac{1}{\sqrt{N}}\sum_i w_i^a\right) + \kappa_0'\left(\frac{1}{N}\sum_i w_i^a M_i^a\right) + O\left(\frac{1}{\sqrt{D}}\right). \tag{5.112}$$

Proceeding in the same way for the second moments, for $i \neq j$ we have that to leading order

$$\mathbb{E}_{\mathbf{x}|\bar{g}}\varphi(\lambda_i^a)\varphi(\lambda_j^b) = \mathbb{E}\varphi(\hat{\lambda}_i^a + \frac{1}{\sqrt{N}}M_i^a)\mathbb{E}\varphi(\hat{\lambda}_j^b + \frac{1}{\sqrt{N}}M_j^b) + \tag{5.113}$$

$$+\left(\frac{1}{D}(ZZ^T)_{ij} - \frac{1}{D}\sum_{cd}\sum_{kl}\frac{Z_{ik}v_k^c}{\sqrt{D}}\frac{Z_{jl}v_l^d}{\sqrt{D}}(q^v)^{-1}_{cd}\right)\times$$

$$\times\,\mathbb{E}\hat{\lambda}_i^a\varphi(\hat{\lambda}_i^a + \frac{1}{\sqrt{N}}M_i^a)\mathbb{E}\hat{\lambda}_j^b\varphi(\hat{\lambda}_j^b + \frac{1}{\sqrt{N}}M_j^b) \approx$$

$$= \kappa_0^2 + \left(\frac{1}{D}(ZZ^T)_{ij} - \frac{1}{D}\sum_{cd}\sum_{kl}\frac{Z_{ik}v_k^c}{\sqrt{D}}\frac{Z_{jl}v_l^d}{\sqrt{D}}(q^v)^{-1}_{cd}\right)\kappa_1^2. \tag{5.114}$$

For $i = j$ instead

$$\mathbb{E}_{\mathbf{x}|\bar{g}}\varphi(\lambda_i^a)\varphi(\lambda_i^b) = \int D\hat{\lambda}\,\varphi(\hat{\lambda}+\frac{1}{\sqrt{N}}M_i^a)\varphi(\hat{\lambda}+\frac{1}{\sqrt{N}}M_i^b) = \tag{5.115}$$

$$= \int D\hat{\lambda}\left(\varphi(\hat{\lambda})+\varphi'(\hat{\lambda})\frac{1}{\sqrt{N}}M_i^a\right)\left(\varphi(\hat{\lambda})+\varphi'(\hat{\lambda})\frac{1}{\sqrt{N}}M_i^b\right) = \tag{5.116}$$

$$= \int D\hat{\lambda}\varphi^2(\hat{\lambda}) + \int D\hat{\lambda}\varphi(\hat{\lambda})\varphi'(\hat{\lambda})\frac{M_i^a+M_i^b}{\sqrt{N}} \approx \kappa_2. \tag{5.117}$$

The second moments of $\bar{y}$ are then given by:

$$Cov(y)_{ab} = \mathbb{E}_{\mathbf{x}|\bar{g}}y^a y^b - (\mathbb{E}_{\mathbf{x}|\bar{g}}y^a)(\mathbb{E}_{\mathbf{x}|\bar{g}}y^b) \tag{5.118}$$

$$= \frac{1}{N}\sum_{ij} w_i^a w_j^b \left[\left(\mathbb{E}_{\mathbf{x}|\bar{g}}\varphi(\lambda_i^a)\varphi(\lambda_j^b)\right) - \left(\mathbb{E}_{\mathbf{x}|\bar{g}}\varphi(\lambda_i^a)\right)\left(\mathbb{E}_{\mathbf{x}|\bar{g}}\varphi(\lambda_j^b)\right)\right] = \tag{5.119}$$

$$= \frac{1}{N}\sum_{i} w_i^a w_i^b \left(\kappa_2 - \kappa_0^2\right) + \tag{5.120}$$

$$+ \frac{1}{N}\sum_{i\neq j} w_i^a w_j^b \left(\frac{1}{D}(ZZ^T)_{ij} - \frac{1}{D}\sum_{cd}\sum_{kl}\frac{Z_{ik}v_k^c}{\sqrt{D}}\frac{Z_{jl}v_l^d}{\sqrt{D}}(q^v)^{-1}_{cd}\right)\kappa_1^2 =$$

$$= q^w_{ab}(\kappa_2 - \kappa_0^2 - \kappa_1^2) + \tag{5.121}$$

$$+ \frac{1}{N}\sum_{i,j} w_i^a w_j^b \left(\frac{1}{D}(ZZ^T)_{ij} - \frac{1}{D}\sum_{cd}\sum_{kl}\frac{Z_{ik}v_k^c}{\sqrt{D}}\frac{Z_{jl}v_l^d}{\sqrt{D}}(q^v)^{-1}_{cd}\right)\kappa_1^2 =$$

$$= \kappa_*^2 q_{ab} + \kappa_1^2 q^Z_{ab} - \frac{\kappa_1^2}{\alpha_D}\sum_{cd} q^{wv}_{ac}(q^v)^{-1}_{cd}q^{wv}_{bd}. \tag{5.122}$$

Putting everything together, we have that

$$P(\bar{y}\,|\,\bar{g}) = \mathcal{N}(\bar{y};\bar{\mu}(\bar{g}),\Sigma), \tag{5.123}$$

where the mean and covariance matrix are given by

$$\Sigma_{ab} = \kappa^* q^w_{ab} + \kappa_1^2 Q^Z_{ab} - \frac{\kappa_1^2}{\alpha_D} \sum_{cd} q^{wv}_{ac} (q_v^{-1})_{cd} q^{wv}_{bd}, \tag{5.124}$$

$$\mu_a(\bar{g}) = \kappa_0 \left(\frac{1}{\sqrt{N}} \sum_i w_i^a \right) + \kappa_1 \left(\frac{1}{N} g_a \sum_i w_i^a u_i^a \right) + \tag{5.125}$$
$$+ \frac{\kappa_1}{\sqrt{\alpha_D}} \left(\frac{1}{N\sqrt{D}} \sum_{bc} g_b (q_v^{-1})_{bc} \sum_{ji} w_i^a Z_{ij} v_j^c \right).$$

The full distribution $P(\bar{y})$ is obtained by marginalizing with respect to g. Since this quantity only enters the mean, the distribution will still be Gaussian. The first moment is easily derived

$$\mathbb{E}\, y_a = \kappa_0 \left(\frac{1}{\sqrt{N}} \sum_i w_i^a \right) \tag{5.126}$$

For the second moment we have instead

$$Cov(\bar{y}) = \mathbb{E}_{\bar{g}} \mathbb{E}_{\boldsymbol{x}|\bar{g}} y^a y^b - \mathbb{E}_{\bar{g}} \mathbb{E}_{\boldsymbol{x}|\bar{g}} y^a \mathbb{E}_{\bar{g}} \mathbb{E}_{\boldsymbol{x}|\bar{g}} y^b = \mathbb{E}_{\bar{g}} Cov(\bar{y}|\bar{g}) + Cov_{\bar{g}}(\bar{\mu}) = \tag{5.127}$$

$$= \kappa^* q^w_{ab} + \kappa_1^2 Q^Z_{ab} - \frac{\kappa_1^2}{\alpha_D} \sum_{cd} q^{wv}_{ac} (q_v^{-1})_{cd} q^{wv}_{bd} + \tag{5.128}$$

$$+ \kappa_1^2 \mathbb{E}_{\bar{g}} \Bigg[\frac{1}{N^2} \sum_i w_i^a \Big(g_a u_i^a + \frac{1}{\sqrt{D\alpha_D}} \sum_{cfj} g_c (\bar{q}_v^{-1})_{cf} Z_{ij} v_j^f \Big) \times$$
$$\times \sum_i w_i^b \Big(g_b u_i^b + \frac{1}{\sqrt{D\alpha_D}} \sum_{dej} g_d (\bar{q}_v)^{-1}_{de} Z_{ij} v_j^e \Big) \Bigg] =$$

$$= \kappa^* q^w_{ab} + \kappa_1^2 Q^Z_{ab} + \kappa_1^2 q^{wu}_a q^v_{ab} q^{wu}_b + \frac{\kappa_1^2}{\sqrt{\alpha_D}} (q^{wv}_{ba} q^{wu}_a + q^{wv}_{ab} q^{wu}_b). \tag{5.129}$$

5.C Replica Calculation for the Spiked Random Features Model

In this appendix we give explicit formulas and derivations for quantities introduced in the main text that can be calculated with the replica method.

5.C.1 Free Entropy

Let us start from equations (5.64). Substituting the RS ansatz in the entropic term G_S

$$G_S = \frac{1}{Nn}\log \mathbb{E}_{\boldsymbol{\theta}_0,F}\int \prod_{a=1}^{n} d\boldsymbol{v}_a d\boldsymbol{w}_a d\boldsymbol{u}_a \left(\frac{\lambda}{2\pi}\right)^{nN}\left(\frac{\lambda}{2\pi}\right)^{nD/2}\times \tag{5.130}$$
$$\times\, e^{-\frac{\lambda}{2}\sum_a \|\boldsymbol{u}_a\|^2 - \frac{\hat{\delta q}_v+\lambda}{2}\sum_a \|\boldsymbol{v}_a\|^2 - \hat{m}_v \sum_a (\boldsymbol{v}_0)^T \boldsymbol{v}^a - \frac{1}{2}\hat{q}_v \sum_{ab} \boldsymbol{v_a}\cdot\boldsymbol{v}_b - \frac{\delta\hat{q}_{wv}}{2}\sum_a \boldsymbol{w}_a^T \frac{Z\boldsymbol{v}_a}{\sqrt{D}}}\times$$
$$\times\, e^{-\frac{\hat{q}_{wv}}{2}\sum_{ab}\boldsymbol{w}_a^T\frac{Z\boldsymbol{v}_b}{\sqrt{D}} - \frac{\hat{m}_{wv}}{2}\sum_a \boldsymbol{w}_0^T\frac{Z\boldsymbol{v}_a}{\sqrt{D}} - \frac{\hat{m}_{vw}}{2}\sum_a \boldsymbol{w}_a^T\frac{Z\boldsymbol{v}_0}{\sqrt{D}} - \frac{1}{2}\sum_a (\boldsymbol{w}^a)^T((\delta\hat{q}_w+\lambda)\mathbb{I}_N + \delta\hat{q}_Z\Omega)\boldsymbol{w}^a}$$
$$\times\, e^{-\frac{1}{2}\sum_{ab}\boldsymbol{w}_a(\hat{q}_w\mathbb{I}_N+\hat{q}_Z\Omega)\boldsymbol{w}_b - \sum_a\left(\boldsymbol{w}^0\right)^T(\hat{m}_w\mathbb{I}_N+\hat{m}_Z\Omega)\boldsymbol{w}^a - \sum_a \hat{q}_{wu}(\boldsymbol{w}^a)^T\boldsymbol{u}^a}.$$

We decouple replicas by performing a vector Hubbard-Stratonovich transformation $e^{-\frac{\hat{q}_v}{2}\sum_{ab}\boldsymbol{v}_a\cdot\boldsymbol{v}_b} = \int D\boldsymbol{\xi} e^{i\sqrt{\hat{q}_v}\boldsymbol{\xi}\cdot\sum_a \boldsymbol{v}_a}$, and integrate in variables $\boldsymbol{v}_a$

$$G_S = \frac{1}{Nn}\log \mathbb{E}_{\boldsymbol{\theta}_0,F}\int D\boldsymbol{\xi}\prod_{a=1}^{n} d\boldsymbol{w}_a \left(\frac{\lambda}{2\pi}\right)^{nN/2}\left(\frac{\lambda}{\delta\hat{q}_v+\lambda}\right)^{nD/2}\times \tag{5.131}$$
$$\times\, e^{\frac{1}{2(\delta\hat{q}_v+\lambda)}\sum_a \|\hat{m}_v\boldsymbol{v}_0 + i\sqrt{\hat{q}_v}\boldsymbol{\xi} + \frac{\hat{m}_{wv}}{2}\frac{Z^T}{\sqrt{D}}\boldsymbol{w}_0 + \frac{\delta\hat{q}_{wv}}{2}\frac{Z^T}{\sqrt{D}}\boldsymbol{w}_a + \frac{\hat{q}_{wv}}{2}\sum_b \frac{Z^T}{\sqrt{D}}\boldsymbol{w}_b\|^2}\times$$
$$\times\, e^{-\frac{\hat{m}_{vw}}{2}\sum_a \boldsymbol{w}_a^T\frac{Z\boldsymbol{v}_0}{\sqrt{D}} - \frac{1}{2}\sum_a(\boldsymbol{w}^a)^T\left((\delta\hat{q}_w+\lambda-\frac{\hat{q}_{wu}^2}{\lambda})\mathbb{I}_N + \delta\hat{q}_Z\Omega\right)\boldsymbol{w}^a - \frac{1}{2}\sum_{ab}\boldsymbol{w}_a(\hat{q}_w\mathbb{I}_N+\hat{q}_Z\Omega)\boldsymbol{w}_b - \sum_a\left(\boldsymbol{w}^0\right)^T(\hat{m}_w\mathbb{I}_N+\hat{m}_Z\Omega)\boldsymbol{w}^a}.$$

Ignoring n^2 terms, and performing a second Hubbard-Stratonovich transform we get to

$$G_S = \frac{\alpha_D}{2}\log\left(\frac{\lambda}{\hat{\delta q}_v+\lambda}\right) + \frac{1}{n}\log \mathbb{E}_{\boldsymbol{\theta}_0,F}\int D\boldsymbol{\xi} D\boldsymbol{\zeta}\left(\int d\boldsymbol{w}\left(\frac{\lambda}{2\pi}\right)^{N/2}\times\right. \tag{5.132}$$
$$\times\, e^{\frac{1}{2(\hat{\delta q}_v+\lambda)}\|m_v\boldsymbol{v}_0 + i\sqrt{q_v}\boldsymbol{\xi} + \frac{\hat{m}_{wv}}{2}\frac{Z^T}{\sqrt{D}}\boldsymbol{w}_0\|^2 - \frac{1}{2}(\boldsymbol{w})^T\left((\delta\hat{q}_w+\lambda-\frac{\hat{q}_{wu}^2}{\lambda})\mathbb{I}_N + \left(\delta\hat{q}_Z - \frac{\delta\hat{q}_{wv}^2}{4(\hat{\delta q}_v+\lambda)}\right)\Omega\right)\boldsymbol{w}}\times$$
$$\left.\times\, e^{\left[\frac{\delta\hat{q}_{wv}m_v}{2(\hat{\delta q}_v+\lambda)}\boldsymbol{v}_0\frac{Z^T}{\sqrt{D}} + i\frac{\delta\hat{q}_{wv}\sqrt{q_v}}{2(\hat{\delta q}_v+\lambda)}\boldsymbol{\xi}\frac{Z^T}{\sqrt{D}} - \left(\boldsymbol{w}^0\right)^T\left(\hat{m}_w\mathbb{I}_N + \left(\hat{m}_Z - \frac{\delta\hat{q}_{wv}\hat{m}_{wv}}{4(\hat{\delta q}_v+\lambda)}\right)\Omega\right) + i\boldsymbol{\zeta}\left(\hat{q}_w\mathbb{I}_N + \left(\hat{q}_Z - \frac{\delta\hat{q}_{wv}\hat{q}_{wv}}{2(\hat{\delta q}_v+\lambda)}\right)\Omega\right)^{1/2} - \frac{\hat{m}_{vw}}{2}\frac{\boldsymbol{v}_0 Z^T}{\sqrt{D}}\right]\boldsymbol{w}}\right)^n.$$

Performing the last integral in $\boldsymbol{w}$, and taking the $n\to 0$ limit we get to the final expression

$$
\begin{aligned}
\mathcal{G}_S =& \lim_{n\to 0}\frac{G_S}{n} = \\
=& \frac{\alpha_D}{2}\log\left(\frac{\lambda}{\hat{\delta q}_v+\lambda}\right)+\frac{1}{2N}\mathbb{E}_F\,\mathrm{Tr}\log\frac{\lambda}{\delta\hat{q}_w+\lambda-\frac{\hat{q}_{wu}^2}{\lambda}+\left(\delta\hat{q}_Z-\frac{\delta\hat{q}_{wv}^2}{4(\hat{\delta q}_v+\lambda)}\right)\Omega}+ \\
&+\frac{\hat{m}_{wv}^2}{8(\hat{\delta q}_v+\lambda)}\mathbb{E}_F\frac{1}{N}\mathrm{Tr}\,\Omega+\mathbb{E}_F\frac{1}{2N}\mathrm{Tr}\frac{\left(\left(\frac{\delta\hat{q}_{wv}m_v}{2(\hat{\delta q}_v+\lambda)}-\frac{\hat{m}_{vw}}{2}\right)^2-\frac{\delta\hat{q}_{wv}^2 q_v}{4(\hat{\delta q}_v+\lambda)^2}\right)\Omega}{(\delta\hat{q}_w+\lambda-\frac{\hat{q}_{wu}^2}{\lambda})+\left(\delta\hat{q}_Z-\frac{\delta\hat{q}_{wv}^2}{4(\hat{\delta q}_v+\lambda)}\right)\Omega}+ \\
&+\frac{\alpha_D}{2}\frac{m_v^2-q_v}{\hat{\delta q}_v+\lambda}+\mathbb{E}_F\frac{1}{2N}\mathrm{Tr}\frac{\left(\hat{m}_w+\left(\hat{m}_Z-\frac{\delta\hat{q}_{wv}\hat{m}_{wv}}{4(\hat{\delta q}_v+\lambda)}\right)\Omega\right)^2-\left(\hat{q}_w+\left(\hat{q}_Z-\frac{\delta\hat{q}_{wv}\hat{q}_{wv}}{2(\hat{\delta q}_v+\lambda)}\right)\Omega\right)}{(\delta\hat{q}_w+\lambda-\frac{\hat{q}_{wu}^2}{\lambda})+\left(\delta\hat{q}_Z-\frac{\delta\hat{q}_{wv}^2}{4(\hat{\delta q}_v+\lambda)}\right)\Omega}.
\end{aligned}
\tag{5.133}
$$

The last average to perform is with respect to the random projection matrix F. As we can see, it only appears in the matrix $\Omega = FF^T/D$, which in turn appears only under the trace operator. This allows us to make the substitution

$$
\frac{1}{N}\mathrm{Tr}\,f(\Omega)\to\mathbb{E}_\rho f(\rho). \tag{5.134}
$$

where ρ is distributed according to the Marchenko-Pastur law.
Now we calculate the energetic term.

$$
G_E = \frac{1}{n}\log\int\prod_{a=0}^{n}\frac{dy_a}{\sqrt{2\pi}}\frac{1}{\sqrt{\det(\Sigma)}}e^{-\frac{1}{2}\sum_{ab}y_a(\Sigma^{-1})_{ab}y_b-\beta\sum_{a=1}^{n}\ell\left(y^0,y^a\right)} \tag{5.135}
$$

$$
= \frac{1}{n}\log\int\prod_{a=0}^{n}\frac{dy_a d\lambda_a}{2\pi}e^{-\frac{1}{2}\sum_{ab}\lambda_a^T\Sigma_{ab}\lambda_a+i\sum_a\lambda_a y_a-\beta\sum_{a=1}^{n}\ell\left(y^0,y^a\right)}. \tag{5.136}
$$

If we call

$$
\begin{cases}
\Sigma_d = \kappa^*(q_w+\delta q_w)+\kappa_1^2(q_Z+\delta q_Z)+\kappa_1^2(q_v+\delta q_v)q_{wu}^2+2\frac{\kappa_1^2}{\sqrt{\alpha_D}}q_{wu}(q_{wv}+\delta q_{wv}),\\
\Sigma_0 = \kappa^* q_w+\kappa_1^2 q_Z+\kappa_1^2 q_v q_{wu}^2+2\frac{\kappa_1^2}{\sqrt{\alpha_D}}q_{wu}q_{wv},\\
\Sigma_m = \kappa^* m_w+\kappa_1^2 m_Z+\kappa_1^2 q_v q_{wu}\tilde{q}_{wu}+2\frac{\kappa_1^2}{\sqrt{\alpha_D}}\left(m_{wv}q_{wu}+m_{vw}\tilde{q}_{wu}\right),\\
\tilde{\Sigma} = \kappa^*\tilde{q}_w+\kappa_1^2\tilde{q}_Z+\kappa_1^2\tilde{q}_v\tilde{q}_{wu}^2+2\frac{\kappa_1^2}{\sqrt{\alpha_D}}\tilde{q}_{wu}\tilde{q}_{wv},
\end{cases}
\tag{5.137}
$$

then

$$G_E = \frac{1}{n}\log\int\prod_{a=0}^{n}\frac{dy_a d\lambda_a}{2\pi}e^{-\frac{1}{2}\lambda_0^2\tilde{\Sigma}-\Sigma_m\sum_a\lambda_0\lambda_a-\frac{1}{2}(\Sigma_d-\Sigma_0)\sum_{a=1}^{n}\lambda_a^2-\frac{1}{2}\Sigma_0\left(\sum_a\lambda_a\right)^2}\times \\ \times e^{i\sum_{a=0}\lambda_a y_a-\beta\sum_{a=1}^{n}\ell\left(y^0,y^a\right)}. \tag{5.138}$$

Performing once again a Hubbbard-Stratonovich transformation and taking the $n\to 0$ limit we get

$$\mathcal{G}_E = \lim_{n\to 0}\frac{G_E}{n} = \int Dy_0\,Dz\log\int\frac{dy}{\sqrt{2\pi(\Sigma_d-\Sigma_0)}}\,e^{-\frac{1}{2(\Sigma_d-\Sigma_0)}\left(y-\frac{\Sigma_m}{\sqrt{\tilde{\Sigma}}}y_0-\sqrt{\Sigma_0-\frac{\Sigma_m^2}{\tilde{\Sigma}}}z\right)^2-\beta\ell(y_0,y)}. \tag{5.139}$$

From now one we consider a specific loss function, namely the MSE defined above, and plug it in equation (5.139). Thanks to the quadratic form of the loss the integral in dy is Gaussian and can be done analytically, giving

$$\mathcal{G}_E = \int Dy_0 Dz\left[-\frac{1}{2}\log\left(1+\beta(\Sigma_d-\Sigma_0)\right)-\frac{1}{2}\frac{\left(\left(\frac{\Sigma_m}{\tilde{\Sigma}}-1\right)\sqrt{\tilde{\Sigma}}y_0+\sqrt{\Sigma_0-\frac{\Sigma_m^2}{\tilde{\Sigma}}}z\right)^2}{\Sigma_d-\Sigma_0+\frac{1}{\beta}}\right]. \tag{5.140}$$

Performing the last Gaussian averages with respect to y_0 and z we get to the final expression

$$\mathcal{G}_E = -\frac{1}{2}\log\left(1+\beta(\Sigma_d-\Sigma_0)\right)-\frac{1}{2}\frac{\Sigma_0+\tilde{\Sigma}-2\Sigma_m}{\Sigma_d-\Sigma_0+\frac{1}{\beta}}. \tag{5.141}$$

5.C.2 Observables

In this section, we will derive explicit formulas for the Training and Generalization errors, defined in equations (5.44) and (5.45).
The first is obtained by simply noting that

$$\epsilon_t = -\frac{\partial\phi}{\partial\beta}. \tag{5.142}$$

The only term that depends explicitly on β is the energetic term, so we easily get

$$\epsilon_t = -\frac{1}{2}\frac{\Sigma_d-\Sigma_0}{1+\beta(\Sigma_d-\Sigma_0)}-\frac{1}{2}\frac{\Sigma_0+\tilde{\Sigma}-2\Sigma_m}{\left(\Sigma_d-\Sigma_0+\frac{1}{\beta}\right)^2}\frac{1}{\beta^2}. \tag{5.143}$$

The generalization error instead can be obtained using the distribution $P(\bar{y})$ derived above. Indeed we have that

$$\epsilon_g = \int dy\, dy_0\, P(y, y_0)\, \frac{1}{2}(y - y_0)^2 . \tag{5.144}$$

The distribution is Gaussian, so the second moments are easily derived

$$\epsilon_g = \frac{\Sigma_d + \tilde{\Sigma} - 2\Sigma_m}{2}. \tag{5.145}$$

5.C.3 The $\beta \to \infty$ Limit

As we can see from equation 5.46, taking the $\beta \to \infty$ limit after we have done the rescaling $\lambda \to \beta\lambda$ concentrates the measure on the Empirical Risk Minimizer. In this limit, the order parameters have well established scalings, so the following rescalings have to be performed for their limit to be well defined

$$\hat{q} \to \beta^2 \hat{q}, \qquad \hat{m} \to \beta \hat{m}, \qquad \delta\hat{q} \to \beta\, \delta\hat{q}, \tag{5.146}$$

$$q \to q, \qquad m \to m, \qquad \delta q \to \delta q/\beta, \tag{5.147}$$

$$\phi \to \phi/\beta. \tag{5.148}$$

With these scalings, both the entropic and energetic terms can be straightforwardly obtained

$$\begin{aligned} \frac{\mathcal{G}_S}{\beta} \underset{\beta\to\infty}{\to} \frac{\alpha_D}{2} \frac{\hat{m}_v^2 - \hat{q}_v}{\delta\hat{q}_v + \lambda_v} + \frac{1}{8}\mathbb{E}_\rho \left[\frac{\left(\left(\hat{m}_{wv} - \frac{\delta\hat{q}_{wv}\hat{m}_v}{\delta\hat{q}_v + \beta\lambda_v} \right)^2 - \frac{\delta\hat{q}_{wv}^2 \hat{q}_v}{(\delta\hat{q}_v + \lambda_v)^2} \right)\rho}{\delta\hat{q}_w + \lambda_w - \frac{\hat{q}_{wu}^2}{\lambda_u} + \left(\delta\hat{Q}_Z - \frac{\delta\hat{q}_{wv}^2}{4(\delta\hat{q}_v + \lambda_v)} \right)\rho} \right] + \\ + \frac{\hat{m}_{wv}^2}{8(\delta\hat{q}_v + \lambda_v)} \mathbb{E}_\rho\left[\rho\right] + \frac{1}{2}\mathbb{E}_\rho \left[\frac{\left(\hat{m}_w + \left(\hat{m}_Z - \frac{\delta\hat{q}_{wv}\hat{m}_{wv}}{4(\delta\hat{q}_v + \lambda_v)} \right)\rho \right)^2 - \left(\hat{q}_w + \left(\hat{Q}_Z - \frac{\delta\hat{q}_{wv}\hat{q}_{wv}}{2(\delta\hat{q}_v + \lambda_v)} \right)\rho \right)}{\left(\delta\hat{q}_w + \lambda_w - \frac{\hat{q}_{wu}^2}{\lambda_u} + \left(\delta\hat{Q}_Z - \frac{\delta\hat{q}_{wv}^2}{4(\delta\hat{q}_v + \lambda_v)} \right)\rho \right)} \right], \end{aligned} \tag{5.149}$$

$$\frac{\mathcal{G}_E}{\beta} \underset{\beta\to\infty}{\to} -\frac{1}{2}\frac{\Sigma_0 + \tilde{\Sigma} - 2\Sigma_m}{\Sigma_d - \Sigma_0 + 1}, \tag{5.150}$$

Finally, the Generalization error stays the same, while the training error becomes

$$\epsilon_t \underset{\beta\to\infty}{\to} \frac{1}{2}\frac{\Sigma_0 + \tilde{\Sigma} - 2\Sigma_m}{(1 + \Sigma_d - \Sigma_0)^2}. \tag{5.151}$$

Chapter 6

Conclusions and Future Directions

In this thesis we have used the methods of statistical physics, in particular the replica method, to study a number of models of neural networks. Although the reasons to study each model are diverse, a central theme to all cases considered is the non-convexity of the loss as a function of the parameters of the model. Indeed many simple one-layer models that have been studied in the literature are known to be convex, and thus cannot be faithful models of more complex deep learning phenomena which are connected to non-convexity. Considering a negative margin, as we did in chapter 3, considering two layer networks, as we did in chapter 4, and studying two layer models in which weights can be learned on both layers, as we did in chapter 5, allows us to investigate particular aspects of this non-convexity in neural networks.

In chapter 3 this non-convexity was investigated directly, by looking at the linear connectivity between minimizers of the loss. Indeed once we acknowledge the non-convexity of the landscape of neural networks, this opens up a plethora of questions about its geometry: how non-convex is this landscape? How connected are the minimizers? Do particular geometries emerge from this non-convexity? The tool of Linear Mode Connectivity allowed us to probe such landscape, and to infer a particular type of geometry known as *star-shaped*. Most notably, this geometry was then studied and observed in more complex models of neural networks [Lin et al., 2024] and even in real neural networks [Sonthalia et al., 2024]. This is precisely the role that theory should have on experiments: inferring structures and making predictions that can be verified. There are many interesting extensions that can be thought of. For example studying a teacher-student model would allow us to look at the connectivity not only of the training error but also of the generalization error.

In chapter 4 non-convexity appeared as a feature which required sophisticated tools to be dealt with. Indeed the full-RSB framework is required precisely due to this non-convexity, and phenomena such as the non-concentration of the overlap are interesting consequences. Thanks to these tools we were able to numerically estimate a threshold which hadn't yet been

determined, the storage capacity, both for the Negative Perceptron and for the Tree-Committee Machine with arbitrary activation function. This required developing a numerical method to solve the full-RSB equations, and a numeric procedure to obtain a precise estimate of the threshold. Thanks to the former, we were able to distinguish two phases in the phase diagram of the Negative Perceptron, the Gardner phase and the full-RSB phase, and to give a rough estimate of the line separating the two. We argued that such line might have some important algorithmic consequences for AMP-based algorithms that can be provably find solutions in the full-RSB phase but not in the Gardner phase. Thanks to the precise estimates of the SAT-UNSAT transition, we were also able to compare the performance of Gradient Descent as a solver to the actual existence of solutions. We found that a small but non-zero algorithmic gap exists between the maximum constraint density α at which GD can find solutions and the actual storage capacity α_C. Finally we came back to the problem of linear connectivity, and used the full-RSB machinery to analyze the disconnection in the fRSB phase. We found that as α is increased the connectivity of a set of y solutions gradually breaks down: there exists thresholds $\alpha^d_{y_1} < \cdots < \alpha^d_{y_n}$ with $y_1 > \cdots > y_n$ such that for $\alpha > \alpha^d_y$ there exists no κ such that the simplex generated by y solutions sampled with that margin is at energy zero. An interesting extension would be to look at the full-RSB error along the linear paths between solutions with different margins. This would enable us to determine the breaking down of the star-shaped geometry.

Finally, in chapter 5, non-convexity appeared as a direct consequence of the types of models studied. The study of the Tree-Committee Machine on the hidden manifold data allowed us to look at the generalization performance of a non-convex model in this setting. The Spiked Random Feature Model instead set a first stone on the path that leads to understanding models of network where learning happens in both layers. Although with the scaling we chose we are not investigating the feature learning regime, it is still worth to investigate how the learning of such spike occurs in a teacher-student scenario. We found that even in the zero regularization limit, there exists a minimum amount of data for the network to align its spike with that of the teacher, and numerically estimated this threshold α_{ril} as a function of the strength of the alignment in the teacher. Having observed this detection-like phenomenon with gaussian data, we then ran experiments on fine-tuning task with actual datasets, and found a similar behaviour for the alignment of the spike. A natural question is what happens in the feature learning regime. This can be addressed in our model by choosing a different scaling for the spike, such that the perturbation to the features due to the spike is of the same order as that due to the random feature matrix. Although technically more challenging, this would offer a glimpse into the world of learned features, and could be connected to a number of works that have explored this regime [Cui et al., 2024, Ba et al., 2022].

Bibliography

Emmanuel Abbe, Shuangping Li, and Allan Sly. Proof of the contiguity conjecture and lognormal limit for the symmetric perceptron. *arXiv preprint arXiv:2102.13069*, 2021.

Fabián Aguirre-López, Silvio Franz, and Mauro Pastore. Random features and polynomial rules. *arXiv preprint arXiv:2402.10164*, 2024.

Brandon L Annesi, Enrico M Malatesta, and Francesco Zamponi. Exact full-rsb sat/unsat transition in infinitely wide two-layer neural networks. *arXiv preprint arXiv:2410.06717*, 2024.

Brandon Livio Annesi, Clarissa Lauditi, Carlo Lucibello, Enrico M. Malatesta, Gabriele Perugini, Fabrizio Pittorino, and Luca Saglietti. Star-shaped space of solutions of the spherical negative perceptron. *Phys. Rev. Lett.*, 131:227301, Nov 2023. doi: 10.1103/PhysRevLett.131.227301. URL `https://link.aps.org/doi/10.1103/PhysRevLett.131.227301`.

Jean-Marc Azaïs and Mario Wschebor. *Level sets and extrema of random processes and fields*. John Wiley & Sons, 2009.

Jimmy Ba, Murat A Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. *Advances in Neural Information Processing Systems*, 35:37932–37946, 2022.

Marco Baity-Jesi, Levent Sagun, Mario Geiger, Stefano Spigler, Gérard Ben Arous, Chiara Cammarota, Yann LeCun, Matthieu Wyart, and Giulio Biroli. Comparing dynamics: Deep neural networks versus glassy systems. In *International Conference on Machine Learning*, pages 314–323. PMLR, 2018.

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, 2015.

Carlo Baldassi, Christian Borgs, Jennifer T Chayes, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes. *Proceedings of the National Academy of Sciences*, 113(48):E7655–E7662, 2016.

Carlo Baldassi, Enrico M. Malatesta, and Riccardo Zecchina. Properties of the geometry of solutions and capacity of multilayer neural networks with rectified linear unit activations. *Phys. Rev. Lett.*, 123:170602, Oct 2019. doi: 10.1103/PhysRevLett.123.170602.

Carlo Baldassi, Fabrizio Pittorino, and Riccardo Zecchina. Shaping the learning landscape in neural networks around wide flat minima. *Proceedings of the National Academy of Sciences*, 117(1):161–170, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1908636117.

Carlo Baldassi, Clarissa Lauditi, Enrico M Malatesta, Gabriele Perugini, and Riccardo Zecchina. Unveiling the structure of wide flat minima in neural networks. *Physical Review Letters*, 127 (27):278301, 2021.

Carlo Baldassi, Clarissa Lauditi, Enrico M Malatesta, Rosalba Pacelli, Gabriele Perugini, and Riccardo Zecchina. Learning through atypical phase transitions in overparameterized neural networks. *Physical Review E*, 106(1):014116, 2022.

Carlo Baldassi, Enrico M. Malatesta, Gabriele Perugini, and Riccardo Zecchina. Typical and atypical solutions in nonconvex neural networks with discrete and continuous weights. *Phys. Rev. E*, 108:024310, Aug 2023. doi: 10.1103/PhysRevE.108.024310. URL `https://link.aps.org/doi/10.1103/PhysRevE.108.024310`.

E Barkai, D Hansel, and I Kanter. Statistical mechanics of a multilayered neural network. *Physical review letters*, 65(18):2312, 1990.

E. Barkai, D. Hansel, and H. Sompolinsky. Broken symmetries in multilayered perceptrons. *Phys. Rev. A*, 45:4146–4161, Mar 1992. doi: 10.1103/PhysRevA.45.4146. URL `https://link.aps.org/doi/10.1103/PhysRevA.45.4146`.

Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. Deep learning: a statistical viewpoint. *Acta numerica*, 30:87–201, 2021.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1903070116.

Alfredo Braunstein and Riccardo Zecchina. Learning by message passing in networks of discrete synapses. *Physical review letters*, 96(3):030201, 2006.

Patrick Charbonneau, Jorge Kurchan, Giorgio Parisi, Pierfrancesco Urbani, and Francesco Zamponi. Fractal free energy landscapes in structural glasses. *Nature communications*, 5 (1):1–6, 2014. doi: 10.1038/ncomms4725.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124018, 2019.

Feng Chen, Daniel Kunin, Atsushi Yamamura, and Surya Ganguli. Stochastic collapse: How gradient noise attracts sgd dynamics towards simpler subnetworks. *Advances in Neural Information Processing Systems*, 36, 2024.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on learning theory*, pages 1305–1338. PMLR, 2020.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204. PMLR, 2015.

Yaim Cooper. The loss landscape of overparameterized neural networks. *arXiv preprint arXiv:1804.10200*, 2018.

Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 1965.

A. Crisanti and T. Rizzo. Analysis of the ∞-replica symmetry breaking solution of the sherrington-kirkpatrick model. *Phys. Rev. E*, 65:046137, Apr 2002. doi: 10.1103/PhysRevE.65.046137. URL `https://link.aps.org/doi/10.1103/PhysRevE.65.046137`.

Hugo Cui, Luca Pesce, Yatin Dandi, Florent Krzakala, Yue M Lu, Lenka Zdeborová, and Bruno Loureiro. Asymptotics of feature learning in two-layer networks after one gradient-step. *arXiv preprint arXiv:2402.04980*, 2024.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Stèphane d'Ascoli, Maria Refinetti, Giulio Biroli, and Florent Krzakala. Double trouble in double descent : Bias and variance(s) in the lazy regime, 2020.

J R L de Almeida and D J Thouless. Stability of the Sherrington-Kirkpatrick solution of a spin glass model. *Journal of Physics A: Mathematical and General*, 11(5):983, may 1978. doi: 10.1088/0305-4470/11/5/028.

Tian Ding, Dawei Li, and Ruoyu Sun. Sub-optimal local minima exist for almost all over-parameterized neural networks. *Journal of Environmental Sciences (China) English Ed*, 2019.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pages 1309–1318. PMLR, 2018.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019.

B. Duplantier. Comment on parisi's equation for the sk model for spin glasses. *Journal of Physics A: Mathematical and General*, 14(1):283, 1981. doi: 10.1088/0305-4470/14/1/027. URL `http://stacks.iop.org/0305-4470/14/i=1/a=027`.

Ahmed El Alaoui and Mark Sellke. Algorithmic pure states for the negative spherical perceptron. *Journal of Statistical Physics*, 189(2):27, 2022. doi: https://doi.org/10.1007/s10955-022-02976-6.

Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. Optimization of mean-field spin glasses. *The Annals of Probability*, 49(6):2922–2960, 2021. doi: 10.1214/21-AOP1519.

A. Engel, H. M. Köhler, F. Tschepke, H. Vollmayr, and A. Zippelius. Storage capacity and learning algorithms for two-layer neural networks. *Phys. Rev. A*, 45:7590–7609, May 1992. doi: 10.1103/PhysRevA.45.7590. URL `https://link.aps.org/doi/10.1103/PhysRevA.45.7590`.

Andreas Engel and Christian Van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, 2001.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020.

Silvio Franz and Giorgio Parisi. Recipes for metastable states in spin glasses. *Journal de Physique I*, 5(11):1401–1415, 1995.

Silvio Franz and Giorgio Parisi. The simplest model of jamming. *Journal of Physics A: Mathematical and Theoretical*, 49(14):145001, feb 2016. doi: 10.1088/1751-8113/49/14/145001.

Silvio Franz, Giorgio Parisi, Maksim Sevelev, Pierfrancesco Urbani, and Francesco Zamponi. Universality of the SAT-UNSAT (jamming) threshold in non-convex continuous constraint satisfaction problems. *SciPost Phys.*, 2:019, 2017. doi: 10.21468/SciPostPhys.2.3.019. URL `https://scipost.org/10.21468/SciPostPhys.2.3.019`.

Silvio Franz, Sungmin Hwang, and Pierfrancesco Urbani. Jamming in multilayer supervised learning models. *Phys. Rev. Lett.*, 123: 160602, Oct 2019. doi: 10.1103/PhysRevLett.123.160602. URL `https://link.aps.org/doi/10.1103/PhysRevLett.123.160602`.

C Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. *arXiv preprint arXiv:1611.01540*, 2016.

Yan V Fyodorov and Rashel Tublin. Counting stationary points of the loss function in the simplest constrained least-square optimization. *arXiv preprint arXiv:1911.12452*, 2019.

David Gamarnik. The overlap gap property: A topological barrier to optimizing over random structures. *Proceedings of the National Academy of Sciences*, 118(41):e2108492118, 2021. doi: https://doi.org/10.1073/pnas.2108492118.

E. Gardner. Spin glasses with p-spin interactions. *Nuclear Physics B*, 257:747–765, 1985. ISSN 0550-3213. doi: https://doi.org/10.1016/0550-3213(85)90374-8. URL https://www.sciencedirect.com/science/article/pii/0550321385903748.

Elizabeth Gardner. The space of interactions in neural network models. *Journal of Physics A: Mathematical and General*, 21(1):257–270, jan 1988. doi: 10.1088/0305-4470/21/1/030.

Elizabeth Gardner and Bernard Derrida. Optimal storage properties of neural network models. *Journal of Physics A: Mathematical and General*, 21(1):271–284, jan 1988. doi: 10.1088/0305-4470/21/1/031.

Elizabeth Gardner and Bernard Derrida. Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, 22(12):1983–1994, jun 1989. doi: 10.1088/0305-4470/22/12/004.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

Federica Gerace, Bruno Loureiro, Florent Krzakala, Marc Mezard, and Lenka Zdeborova. Generalisation error in learning with random features and the hidden manifold model. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3452–3462. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/gerace20a.html.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. 2021.

Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X*, 10(4):041044, 2020.

Sebastian Goldt, Bruno Loureiro, Galen Reeves, Florent Krzakala, Marc Mézard, and Lenka Zdeborová. The gaussian equivalence of generative models for learning with shallow neural networks. In *Mathematical and Scientific Machine Learning*, pages 426–471. PMLR, 2022.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50(2):949, 2022.

Fengxiang He, Tongliang Liu, and Dacheng Tao. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. *Advances in neural information processing systems*, 32, 2019.

Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Haiping Huang and Yoshiyuki Kabashima. Origin of the computational hardness for learning with binary synapses. *Phys. Rev. E*, 90:052813, Nov 2014. doi: 10.1103/PhysRevE.90.052813.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract

Arthur Jacot, Berfin Simsek, Francesco Spadaro, Clément Hongler, and Franck Gabriel. Implicit regularization of random feature models. In *International Conference on Machine Learning*, pages 4631–4640. PMLR, 2020.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Kenji Kawaguchi. Deep learning without poor local minima. *Advances in neural information processing systems*, 29, 2016.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Werner Krauth and Marc Mézard. Storage capacity of memory networks with binary couplings. *Journal de Physique*, 50(20):3057–3066, 1989. doi: /10.1051/jphys:0198900500200305700.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Rohith Kuditipudi, Xiang Wang, Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Rong Ge, and Sanjeev Arora. Explaining landscape connectivity of low-cost solutions for multilayer nets. *Advances in neural information processing systems*, 32, 2019.

Yann LeCun. Une procedure d'apprentissage ponr reseau a seuil asymetrique. *Proceedings of Cognitiva 85*, pages 599–604, 1985.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Dawei Li, Tian Ding, and Ruoyu Sun. On the benefit of width for neural networks: Disappearance of bad basins. *arXiv preprint arXiv:1812.11039*, 2018.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in neural information processing systems*, 31, 2018.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. 2020.

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd international conference on artificial intelligence and statistics*, pages 888–896. PMLR, 2019.

Zhanran Lin, Puheng Li, and Lei Wu. Exploring neural network landscapes: Star-shaped and geodesic connectivity. *arXiv preprint arXiv:2404.06391*, 2024.

Andrea J Liu and Sidney R Nagel. The jamming transition and the marginally jammed solid. *Annu. Rev. Condens. Matter Phys.*, 1(1):347–369, 2010.

Antoine Maillard, Gérard Ben Arous, and Giulio Biroli. Landscape complexity for the empirical risk of generalized linear models. In *Mathematical and Scientific Machine Learning*, pages 287–327. PMLR, 2020.

Eran Malach, Pritish Kamath, Emmanuel Abbe, and Nathan Srebro. Quantifying the benefit of using differentiable learning over tangent kernels. In *International Conference on Machine Learning*, pages 7379–7389. PMLR, 2021.

Enrico M Malatesta. High-dimensional manifold of solutions in neural networks: insights from statistical physics. *arXiv preprint arXiv:2309.09240*, 2023.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *International Conference on Machine Learning*, pages 23610–23641. PMLR, 2023.

Stefano Martiniani and Mathias Casiulis. When you can't count, sample! computable entropies beyond equilibrium from basin volumes. *Papers in Physics*, 15:150001–150001, 2023.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, n/a(n/a), 2019. doi: https://doi.org/10.1002/cpa.22008.

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4):667–766, 2022.

Marc Mézard, Giorgio Parisi, and Miguel Angel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.

V Milman. Surprising geometric phenomena in high-dimensional convexity theory. In *European Congress of Mathematics: Budapest, July 22–26, 1996 Volume II*, pages 73–91. Springer, 1998.

Rémi Monasson and Riccardo Zecchina. Learning and generalization theories of large committee-machines. *Modern Physics Letters B*, 9(30):1887–1897, 1995.

Andrea Montanari. Optimization of the Sherrington–Kirkpatrick Hamiltonian. *SIAM Journal on Computing*, 0(0):FOCS19–1–FOCS19–38, 0. doi: 10.1137/20M132016X. URL `https://doi.org/10.1137/20M132016X`.

Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. The generalization error of max-margin linear classifiers: Benign overfitting and high dimensional asymptotics in the overparametrized regime. *arXiv preprint arXiv:1911.01544*, 2019.

Andrea Montanari, Yiqiao Zhong, and Kangjie Zhou. Tractability from overparametrization: The example of the negative perceptron. *arXiv preprint arXiv:2110.15824*, 2021. URL https://arxiv.org/abs/2110.15824.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021. doi: 10.1088/1742-5468/ac3a74.

Quynh Nguyen. On connected sublevel sets in deep learning. In *International conference on machine learning*, pages 4790–4799. PMLR, 2019.

Albert BJ Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622. New York, NY, 1962.

G. Parisi. Infinite number of order parameters for spin-glasses. *Phys. Rev. Lett.*, 43:1754–1756, Dec 1979a. doi: 10.1103/PhysRevLett.43.1754. URL https://link.aps.org/doi/10.1103/PhysRevLett.43.1754.

Giorgio Parisi. Toward a mean field theory for spin glasses. *Physics Letters A*, 73(3): 203 – 205, 1979b. ISSN 0375-9601. doi: 10.1016/0375-9601(79)90708-4. URL http://www.sciencedirect.com/science/article/pii/0375960179907084.

Giorgio Parisi. A sequence of approximated solutions to the s-k model for spin glasses. *Journal of Physics A: Mathematical and General*, 13(4):L115, 1980a. doi: 10.1088/0305-4470/13/4/009.

Giorgio Parisi. The order parameter for spin glasses: a function on the interval 0-1. *Journal of Physics A: Mathematical and General*, 13(3):1101, 1980b. doi: 10.1088/0305-4470/13/3/042. URL http://stacks.iop.org/0305-4470/13/i=3/a=042.

Giorgio Parisi, Pierfrancesco Urbani, and Francesco Zamponi. *Theory of simple glasses: exact solutions in infinite dimensions*. Cambridge University Press, 2020.

Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. *Advances in neural information processing systems*, 34: 18420–18432, 2021.

Fabrizio Pittorino, Antonio Ferraro, Gabriele Perugini, Christoph Feinauer, Carlo Baldassi, and Riccardo Zecchina. Deep networks on toroids: removing symmetries reveals the structure of flat regions in the landscape geometry. In *International Conference on Machine Learning*, pages 17759–17781. PMLR, 2022.

Walter Pitts and Warren S McCulloch. How we know universals the perception of auditory and visual forms. *The Bulletin of mathematical biophysics*, 9:127–147, 1947.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pages 1177–1184, 2007. URL https://papers.nips.cc/paper/2007/hash/013a006f03dbc5392effeb8f18fda755-Abstract

Tommaso Rizzo. Replica-symmetry-breaking transitions and off-equilibrium dynamics. *Phys. Rev. E*, 88:032135, Sep 2013. doi: 10.1103/PhysRevE.88.032135. URL https://link.aps.org/doi/10.1103/PhysRevE.88.032135.

Valentina Ros, Gerard Ben Arous, Giulio Biroli, and Chiara Cammarota. Complex energy landscapes in spiked-tensor and simple glassy models: Ruggedness, arrangements of local minima, and phase transitions. *Physical Review X*, 9(1):011003, 2019a.

Valentina Ros, Giulio Biroli, and Chiara Cammarota. Complexity of energy barriers in mean-field glassy systems. *Europhysics Letters*, 126(2):20003, 2019b.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

Henry Schwarze. Learning a rule in a multilayer neural network. *Journal of Physics A: Mathematical and General*, 26(21):5781, 1993.

Henry Schwarze and John Hertz. Generalization in a large committee machine. *Europhysics Letters*, 20(4):375, 1992.

Antonio Sclocchi and Pierfrancesco Urbani. High-dimensional optimization under nonconvex excluded volume constraints. *Physical Review E*, 105(2):024134, 2022.

Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In *International Conference on Machine Learning*, pages 9722–9732. PMLR, 2021.

H J Sommers and W Dupont. Distribution of frozen fields in the mean-field theory of spin glasses. *Journal of Physics C: Solid State Physics*, 17(32):5785, nov 1984. doi: 10.1088/0022-3719/17/32/012. URL https://dx.doi.org/10.1088/0022-3719/17/32/012.

Ankit Sonthalia, Alexander Rubinstein, Ehsan Abbasnejad, and Seong Joon Oh. Do deep neural network solutions form a star domain? *arXiv preprint arXiv:2403.07968*, 2024.

Mihailo Stojnic. Negative spherical perceptron. *arXiv preprint arXiv:1306.3980*, 2013.

Mihailo Stojnic. Exact capacity of the wide hidden layer treelike neural networks with generic activations. *arXiv preprint arXiv:2402.05719*, 2024a.

Mihailo Stojnic. Fixed width treelike neural networks capacity analysis–generic activations. *arXiv preprint arXiv:2402.05696*, 2024b.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Luca Venturi, Afonso Bandeira, and Joan Bruna. Neural networks with finite intrinsic dimension have no spurious valleys. *arXiv preprint arXiv:1802.06384*, 15, 2018a.

Luca Venturi, Afonso S Bandeira, and Joan Bruna. Spurious valleys in two-layer neural network optimization landscapes. *arXiv preprint arXiv:1802.06384*, 2018b.

Christopher Williams. Computing with infinite networks. In M.C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1996. URL https://proceedings.neurips.cc/paper$_f$iles/paper/1996/file/ae5e3ce40e0404a45ecacaaf05e*Paper.pdf*.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. In *International Conference on Machine Learning*, pages 10767–10777. PMLR, 2020.

Jacob A. Zavatone-Veth and Cengiz Pehlevan. Activation function dependence of the storage capacity of treelike neural networks. *Phys. Rev. E*, 103:L020301, Feb 2021. doi: 10.1103/PhysRevE.103.L020301. URL https://link.aps.org/doi/10.1103/PhysRevE.103.L020301.

Jacob A. Zavatone-Veth and Cengiz Pehlevan. On Neural Network Kernels and the Storage Capacity Problem. *Neural Computation*, 34(5):1136–1142, 04 2022. ISSN 0899-7667. doi: 10.1162/neco$_{a0}$1494.*URL*https://doi.org/10.1162/neco_a_01494.

Lenka Zdeborová. Understanding deep learning is also a job for physicists. *Nature Physics*, 16 (6):602–604, 2020.

Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. *arXiv preprint arXiv:2310.17513*, 2023.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, February 2021. ISSN 0001-0782. doi: 10.1145/3446776.

Yuanzhao Zhang and Steven H Strogatz. Basins with tentacles. *Physical Review Letters*, 127(19): 194101, 2021.