THÈSE DE DOCTORAT

NNT : 2022UPASP074

# Statistical physics insights on the dynamics and generalisation of artificial neural networks

*Modélisation physique statistique de la dynamique et de la généralisation dans les réseaux de neurones artificiels*

**Thèse de doctorat de l'université Paris-Saclay**

École doctorale n°564 : Physique en Île-de-France (PIF)
Spécialité de doctorat: Physique
Graduate School : Physique, Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Institut de Physique Théorique (Université Paris-Saclay, CNRS, CEA)**, sous la direction de **Lenka ZDEBOROVÁ**, professeur, et le co-encadrement de **Pierfrancesco URBANI**, chargé de recherche.

**Thèse soutenue à Paris-Saclay, le 2 septembre 2022, par**

## Francesca MIGNACCO

### Composition du jury

| | |
|---|---|
| **Leticia CUGLIANDOLO**<br>Professor, Sorbonne Université, Laboratoire de Physique Théorique et Hautes Energies, CNRS, Paris & Institut Universitaire de France. | Présidente |
| **Manfred OPPER**<br>Professor, Centre for Systems Modelling and Quantitative Biomedicine, University of Birmingham. | Rapporteur & Examinateur |
| **David J. SCHWAB**<br>Associate Professor, Initiative for the Theoretical Sciences, Graduate Center, City University of New York. | Rapporteur & Examinateur |
| **Stéphane MALLAT**<br>Professor, College de France, Paris, France & Flatiron Institute, New York, USA. | Examinateur |
| **Lenka ZDEBOROVÁ**<br>Associate Professor, Ecole Polytechnique Fédérale de Lausanne (EPFL). | Directrice de thèse |
| **Pierfrancesco URBANI**<br>CNRS researcher, Université Paris-Saclay, CNRS, CEA, Institut de physique théorique. | Invité |

# Acknowledgements

I want to express my deepest gratitude to my supervisors, Lenka and Pierfrancesco, for introducing me to the joy of discovery. I am extremely grateful to Lenka for her wise guidance: her invaluable criticism, together with her trust in my work, have made me grow a lot during these years. I am deeply indebted to Pierfrancesco for inspiring me with his great scientific curiosity and for his patience in answering all my questions. I also had the pleasure to interact with Florent, who I thank for the fun and interesting discussions.

I am grateful to Manfred Opper and David Schwab for agreeing to review this manuscript. I am also thankful to Leticia Cugliandolo and Stéphane Mallat for accepting to be part of the jury of my thesis defense.

I thank Yue for the interesting collaboration. My most sincere thanks go to my *comité de suivi*, Guilhem and Laura, for their support and precious advice. I thank all the members of the Institut de Physique Théorique. Special thanks go to Grégoire, Camille, and Laure for their kindness and frequent bureaucratic help.

I was lucky to share my Ph.D. journey with a wonderful group of colleagues and friends. I thank Benjamin, Erwan, and Marc for the rare but fun occasions of sharing the office. Thanks to Stefano for his helpful suggestions and for trying to make me do sports. I thank Alia, Antoine, Antonio, Emanuele, Gabriele, Giovanni, Hugo, Jonathan, Luca P., Marylou, Ruben, and Stéphane for the (too short!) good times we have shared. I thank Cédric for explaining me how to prove DMFT. I am particularly grateful to Bruno, Federica, Luca S., and Sebastian for their encouragement and many clarifying explanations. Big thanks go to Maria for all the great moments together in Paris. I also thank Maria and Giovanni for their precious comments on this manuscript. I am really grateful to Elisabetta for our twelve years of friendship and maths, and to Rodrigo for his *Buongiorno* and his insights on AMP. You are the best "Rattoppers" team! I was so happy to be surrounded by a big Italian crew in Paris (with some French/Belgian incursions): thanks to Alberto, Alessandro, Benjamin, Camilla, Chiare, Elena, Flavio, Francesco (Cama), Gaia, Gianmarco, Giulia D., Giulia G. L., Giulia I., Giulio, Hélène, Justine, Leonardo, Lorenzo, Saverio, Vincenzo, Vito. Special thanks to my summer-school dream team: Francesco (Checco) and Matteo.

I want to express my deepest appreciation to Dr. Bonfichi, Dr. Nappi, and all the medical staff at the Hematology and Gynecology departments of S. Matteo hospital in Pavia for following me with professionalism and empathy during these years.

I am thankful to my friends and family in Italy who supported me from afar. Words cannot express my gratitude to my parents, Gloria and Mario, for teaching me the courage of irony and for their unconditional love that helped me in the hard moments. I am grateful to my sister Chiara for always mixing the harshest truth with a great sense of humor. Ringrazio mia nonna Rina per il suo amore un po' apprensivo e per aver sfamato da lontano schiere di ricercatori a sua insaputa.

Finally, I thank Francesco for being on my side every step of the way. He has shared my joy for the smallest achievements and selflessly assisted me whenever I felt lost. I feel so lucky to have embarked on this adventure with you, I look forward to the next one to come.

# Contents

# Acronyms

**AI** Artificial Intelligence. vii, viii, ix

**AMP** Approximate Message Passing. xxv, 36, 41, 42, 43, 44, 56, 86, 87, 104

**ANN** Artificial Neural Network. vii, viii, ix, x, xi, xiv, xv, xvii, xviii, xxv, 2, 3, 8, 9, 12, 53, 54, 57, 58, 74, 86, 105, 107

**BO** Bayes-Optimal. 3, 7, 9, 11, 12, 13, 14, 16, 17, 18, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 39, 40, 41, 42, 43, 44, 45, 71, 72, 73, 103

**CLT** Central Limit Theorem. 52

**CSP** Constraint Satisfaction Problem. xix, 9, 10, 74, 84

**DFA** Direct Feedback Alignment. 107

**DL** Deep Learning. vii, ix, xiii, xiv, xv, 52

**DMFT** Dynamical Mean-Field Theory. xviii, xix, xx, xxiii, xxiv, xxv, 54, 55, 57, 60, 61, 66, 67, 69, 70, 72, 73, 74, 75, 77, 81, 82, 89, 91, 93, 95, 102, 103, 104, 105, 107

**DNN** Deep Neural Network. vii, xi, xii, xiii, xiv, xv, 3, 29, 51, 52, 53, 56, 87

**ERM** Empirical Risk Minimisation. xi, xii, xx, xxv, 3, 4, 9, 12, 13, 18, 19, 24, 30, 33, 34, 36, 37, 40, 41, 44, 45, 50, 58, 74, 103

**FDT** Fluctuation-Dissipation Theorem. xix, xxiii, 24, 75, 76, 77, 78, 79, 80, 105

**GAN** Generative Adversarial Network. x, 104

**GD** Gradient Descent. xi, xii, xix, xxiii, xxiv, 14, 27, 50, 51, 53, 54, 56, 59, 60, 61, 70, 71, 72, 79, 80, 83, 84, 86, 87, 88, 89, 90, 91, 92, 94, 95, 96, 103, 104, 105, 106, 107

**GMM** Gaussian Mixture Model. xviii, xix, xxv, 11, 12, 13, 14, 18, 57, 69, 102, 103, 104, 105

**GPU** Graphic Processing Unit. vii, ix

**i.i.d.** independent and identically distributed. ix, xvii, xx, xxi, xxiv, 5, 9, 11, 16, 33, 55, 61, 64, 74, 77, 86, 88, 104

**MAP** Maximum A Posteriori. 4, 12

**MCMC** Markov-Chain Monte Carlo. 4

**ML** Machine Learning. vii, viii, ix, x, xi, xii, xiii, xv, xviii, xxv, 2, 14, 33, 50, 52, 53, 102, 104, 105

**MLE** Maximum Likelihood Estimator. 4, 5, 12, 14, 26, 27, 32

**MMSE** Minimum Mean Squared Error. 4

**MSE** Mean Squared Error. xii, 5

**p-SGD** Persistent Stochastic Gradient Descent. xix, xxiii, 59, 67, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 84, 88, 91, 92, 93, 95, 96, 97, 102, 103

**RS** Replica Symmetric. 8, 22, 23, 24, 39

**SAT** Satisfiable. 9, 12, 14, 25, 26, 27, 74, 75, 76, 81, 82, 83, 105

**SGD** Stochastic Gradient Descent. xii, xiii, xviii, xix, xxii, xxiii, xxiv, xxv, 50, 51, 52, 53, 54, 56, 57, 59, 60, 61, 67, 70, 72, 73, 74, 75, 76, 77, 78, 79, 80, 84, 86, 87, 88, 91, 92, 93, 94, 95, 97, 102, 103, 104, 105, 106, 107

**SGF** Stochastic Gradient Flow. 60

**SUSY** Super Symmetric. 55, 60, 62, 65

**UNSAT** Unsatisfiable. 9, 13, 24, 25, 26, 74, 75, 76, 78, 79, 80, 83, 103

**VAE** Variational Auto-Encoder. x

# Overview of the thesis

# Motivation and background

*Machine learning* (ML) is a branch of *artificial intelligence* (AI). Nowadays, ML is witnessing an explosive growth of research and investments, largely driven by the success of modern *deep neural networks* (DNNs). Current industrial applications include product recommendation, image recognition, time series prediction, medical diagnosis, natural language processing, and protein folding. The history of *artificial neural networks* (ANNs) dates back to the 40's, with the first models inspired by the biological learning of the human brain (McCulloch & Pitts, 1943; Hebb, 1949). Two distinct waves of disillusionment – the decade-long AI winters – hit the field before convolutional DNNs beated the state-of-the-art image classification methods at the ImageNet challenge in 2012 (Krizhevsky et al., 2012), setting a milestone for the new ML age.

For the past ten years, DNNs have brought about a paradigm shift in computation that resonates at various scales: from a revolution in everyday-life applications to an entirely new toolbox available to scientific research (LeCun et al., 2015). Physics is no exception (Zdeborová, 2017; Carleo et al., 2019). These advances were arguably made possible by the exponential increase in data processing power brought by the development of highly parallelisable Graphic Processing Unit (GPU) processors and have been fueled by the availability of huge amounts of data at unprecedented rates.

However, if on the one hand current DL models have achieved outstanding results in applications, their design still relies heavily on trial-and-error heuristics, which sets the key challenge of building a theoretical framework to ensure the reliability and efficiency of ML systems. This important call has renewed a long-standing research effort to explore the principles underlying the efficient training of ANNs in order to provide theoretical guarantees for practical implementations. Yet, the fundamental open questions that statistician Leo Breiman raised on the theory of ANNs in 1995 (Breiman, 2018) remain to this day largely unanswered:

> *Why don't heavily parameterised neural networks overfit the data?*
>
> *What is the effective number of parameters?*
>
> *Why doesn't backpropagation head for a poor local minima?*
>
> *When should one stop the backpropagation and use the current parameters?*

In order to elucidate the common ground giving rise to the many ML puzzles, in the following we briefly introduce the basic vocabulary and notions of ML with a focus on ANNs. We limit our presentation to those concepts that are strictly necessary to understand the contributions of this thesis. For a thorough overview of the field methods we refer the reader to the books Bishop & Nasrabadi (2006); Goodfellow et al. (2016), and to Mehta et al. (2019) for a comprehensive introduction targeted at a physics audience. In the second section of this chapter, we outline the motivations for inspecting the mysteries of ML theory with the lens of statistical physics.

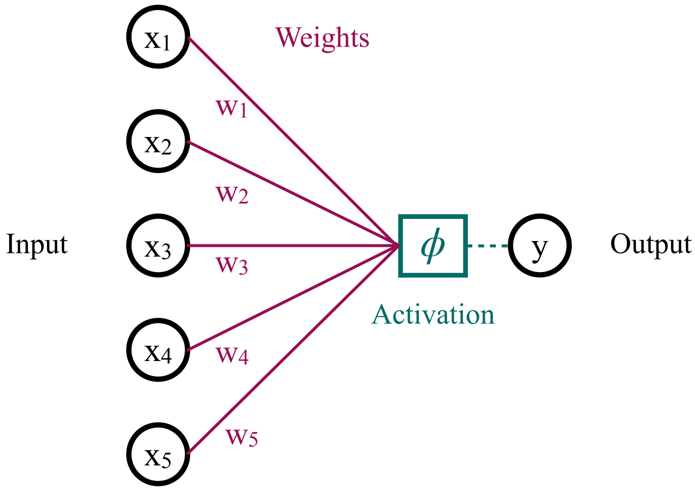


Figure 1 – Pictorial representation of the perceptron model in dimension $d = 5$.

## Machine Learning with Neural Networks 101

The goal of ML is to develop algorithms able to perform a practical task by extracting the necessary information directly from a set of examples. In other words, ML algorithms based on ANNs do not follow a rule-based approach where expert knowledge engineers a list of fixed instructions for the machine, as done instead in the so-called *symbolist* approach to AI. Conversely, ML models are *trained* on large datasets in order to *learn* the relevant *features* underlying a certain task. From an historical point of view, this perspective refers to the *connectionist* approach to AI.

**First ingredient of ANNs: the architecture** — The building block of modern ANNs is the *perceptron*, introduced by Rosenblatt (1958) and inspired by the *formal neuron* previously proposed by McCulloch & Pitts (1943). The perceptron is a simple model defined by $d$ learnable parameters $\boldsymbol{w}$, called *weights*, that maps an input $\boldsymbol{x}$ to an output $\hat{y} = \phi\left(\boldsymbol{w}^\top \boldsymbol{x}\right)$, where $\phi : \mathbb{R} \to \mathbb{R}$ is an (in general non-linear) *activation* function acting component-wise. The classical perceptron was conceived to solve a binary classification task where the output identifies the class membership of the input. In this case, the input is a $d-$dimensional vector and the output is a binary variable, with activation function $\phi(\cdot) = \mathrm{sign}(\cdot)$. A pictorial sketch of the perceptron is illustrated in Figure 1.

In modern jargon, the perceptron is a single-layer feed-forward neural network. Indeed, modern ANNs are built by stacking multiple perceptron units, also called *neurons* or *nodes*, connected by *layers* of learnable weights.[1] The intermediate layers between input and output are called *hidden*, while the *width* is the number of

[1] The term *layer* is used in the literature interchangeably to refer either to the weights or to the nodes, which is sometimes a source of confusion.

neurons in a given layer. The term *feed-forward* refers to the absence of cycles in the connections, which proceed only in the forward direction, from the input through the hidden nodes to the output. If every neuron in a layer is connected to every neuron in the next one, the network is said to be *fully-connected*. A schematic representation of a fully-connected feed-forward multi-layer neural network is depicted in Figure 2. *Depth*, i.e., the presence of multiple layers of weights, is responsible for the great expressivity of modern ANNs, so that the very name of the field was redefined as *deep* learning (DL). Nevertheless, the powerful scalability of perceptron units could not be exploited at its origins due to the lack of practical algorithms able to train such large multi-layer models. A single perceptron machine is instead limited to the realm of linear models and therefore unable to solve even simple tasks as a XOR problem. This pitfall was pointed out by Minsky & Papert (1969), whose criticism dampened the enthusiasm of the first AI research wave, leading to the first of the above-mentioned winters of AI.

Technology filled this algorithmic gap with the advent of GPUs, popularised since the 90's and more than doubling in performance every two years.[2] When large models could finally be trained efficiently, the *architecture*, i.e., the topology of the network, emerged as the first crucial ingredient of the ML pipeline. The architecture of a feed-forward ANN with $L$ layers is expressed by the mapping:

$$\boldsymbol{x} \longmapsto \hat{\boldsymbol{y}} = \phi^{(L)}\left(\boldsymbol{W}^{(L)}\phi^{(L-1)}\left(\dots\phi^{(1)}\left(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{\kappa}^{(1)}\right)\right)\dots + \boldsymbol{\kappa}^{(L)}\right),$$

transforming the input $\boldsymbol{x}$ to the (possibly multi-dimensional) output $\boldsymbol{y}$ and parametrised by the matrices of weights $\{\boldsymbol{W}^{(l)}\}_{l=1}^{L}$ and the thresholds, or *biases*, $\{\boldsymbol{\kappa}^{(l)}\}_{l=1}^{L}$. The zoology of state-of-the-art architectures extends way beyond fully-connected feed-forward networks, encompassing, for instance, convolutional ANNs (LeCun et al., 1989, 1998, 1999) and Long Short-Term Memory recurrent ANNs (Hochreiter & Schmidhuber, 1997). However, in this thesis we only deal with simple feed-forward fully-connected ANNs, hence we do not discuss further more complicated architectures.

**Second ingredient of ANNs: the task —** The crucial discriminant for the choice of the network architecture is the task to be performed. In this regard, as Julius Caesar would put it, "*Ars ML est omnis divisa in partes tres*"[3]: supervised, unsupervised and reinforcement learning. These categories refer to the possible types of input data that the network is presented with.

In the case of *supervised* learning, the dataset $\mathcal{D} = \{(\boldsymbol{x}_\mu, \boldsymbol{y}_\mu)\}_{\mu=1}^{n}$ is made of $n$ pairs of input features $\boldsymbol{x}_\mu \in \mathcal{X} \subseteq \mathbb{R}^d$ and output labels $\boldsymbol{y}_\mu \in \mathcal{Y} \subseteq \mathbb{R}^k$. It is common to assume that the input data and labels are drawn independent and identically distributed (i.i.d.) from the joint probability distribution $\mathrm{P}_{\boldsymbol{x},\boldsymbol{y}}$. The goal of the ANN is to learn the input-output mapping or, equivalently, the conditional probability distribution $\mathrm{P}_{\boldsymbol{y}|\boldsymbol{x}}$. Supervised tasks are essentially of two types: either *classification*, if the output space $\mathcal{Y}$ is discrete, or *regression*, if $\mathcal{Y}$ is continuous. A pictorial representation of these two types of tasks is depicted in Figure 3. The main

[2] This rate is faster than the one predicted by the empirical *Moore's law* and is sometimes referred to as *Huang's law*.

[3] The field of ML is divided into three parts.

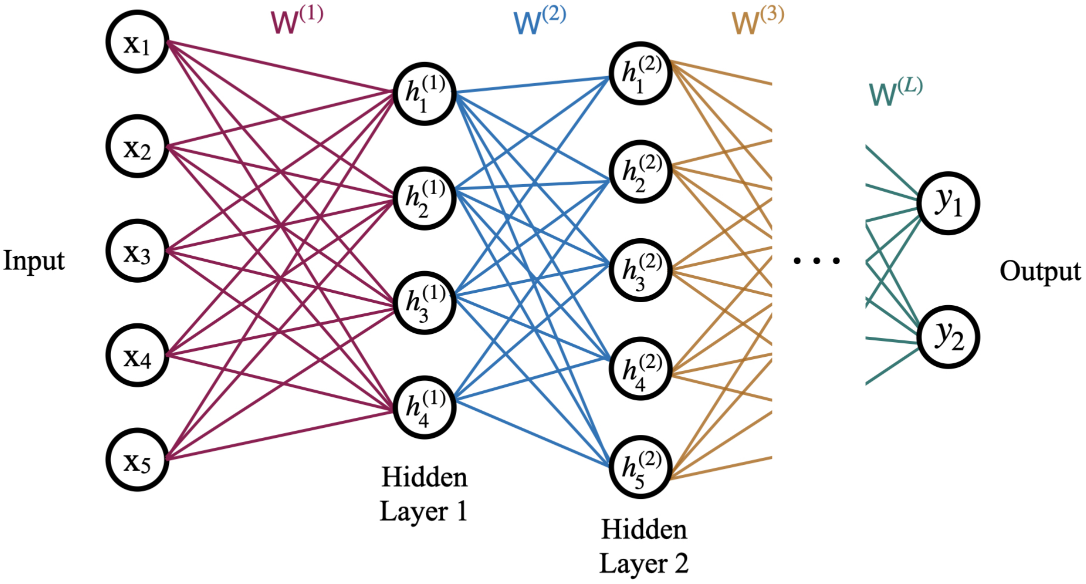


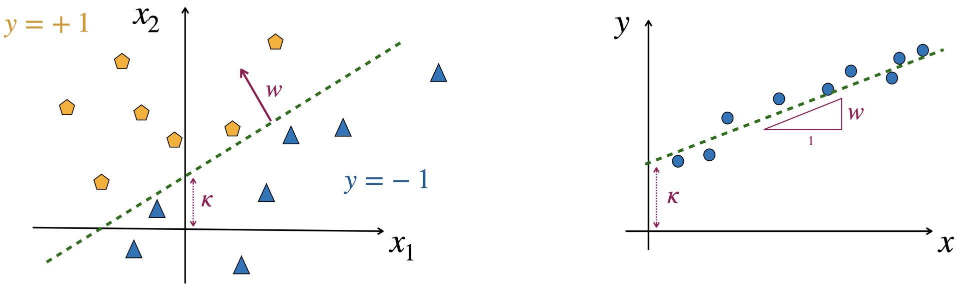
Figure 2 – Pictorial representation of a fully-connected feed-forward neural network with $L$ layers. At each layer $l \in \{1, \dots, L\}$, the hidden units are obtained as: $\boldsymbol{h}^{(l)} = \phi^{(l)}\left(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{\kappa}^{(l)}\right)$, where the activation function acts component-wise and the output is $\hat{\boldsymbol{y}} = \boldsymbol{h}^{(L)}$.

drawback of supervised learning is that it requires *annotation*, i.e., the labeling of input data, which still largely relies on costly human effort.

On the contrary, *unsupervised* learning deals with unlabeled datasets $\mathcal{D} = \{\boldsymbol{x}_\mu\}_{\mu=1}^n$ and aims at extracting relevant information to characterise the underlying probability $\mathrm{P}_{\boldsymbol{x}}$. Classical examples of unsupervised learning tasks are *clustering*, i.e., grouping the data according to their similarity, and *density estimation*, i.e., approximating $\mathrm{P}_{\boldsymbol{x}}$ with the closest among a parametrised family of distributions $\{\mathrm{P}_{\boldsymbol{x}|\boldsymbol{w}}, \boldsymbol{w} \in \mathbb{R}^{d_w}\}$. Recent advanced methods allowing to approximate complex densities are *deep generative models*, such as Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and Variational Auto-Encoders (VAEs) (Kingma & Welling, 2013).

*Reinforcement* learning refers instead to a learning task where the ANN, called *agent* in this context, interacts with the environment via a feedback loop on its actions. At each time, the agent is in a certain state $s_t$ and chooses an action $a_t$ according to a policy $\pi(a, s) = \mathbb{P}\left(a_t = a | s_t = s\right)$ and leading to a state $s_{t+1}$ and a corresponding reward from the interaction with the environment. The goal is to learn the optimal policy. In this manuscript, we only consider supervised learning tasks, and therefore we do not discuss further the other types of tasks.

**Third ingredient of ANNs: the algorithm** — Once we have fixed the task and the architecture, we enter into the core of the ML pipeline, when the *learning* takes place. The *algorithm* specifies the optimisation procedure to find the best possible realisation of the architecture, i.e., the optimal weights and biases, associated to the best performance. In practice, the learning objective measures the degree of error of a given set of weights and biases. Focusing on supervised learning, this error is



(a) **Classification:** Two-dimensional linear classification $\hat{y} = \operatorname{sign}\left(\boldsymbol{w}^\top \boldsymbol{x} + \kappa\right)$ with dimensions $d = 2$, $k = 1$, $n = 14$.

(b) **Regression:** One dimensional linear regression $\hat{y} = wx + \kappa$ with dimensions $d = 1$, $k = 1$, $n = 9$.

Figure 3 – Pictorial representation of two prototypical supervised learning tasks. Both examples are linear tasks, therefore they can be performed by a *perceptron*, or by the readout layer of an ANN, where the inputs $\boldsymbol{x}$ come from the pre-processing of previous layers.

often formalised as the *empirical risk*

$$\hat{\mathcal{R}}\left(\boldsymbol{W}, \mathcal{D}, \ell\right) = \frac{1}{n} \sum_{\mu=1}^{|\mathcal{D}|=n} \ell\left(\hat{\boldsymbol{y}}_{\boldsymbol{W}}\left(\boldsymbol{x}_\mu\right), \boldsymbol{y}_\mu\right) + \lambda\, \Omega\left(\boldsymbol{W}\right).$$

For simplicity, we have incorporated the biases in the weight matrix $\boldsymbol{W}$. The empirical risk $\hat{\mathcal{R}}$ depends on the choice of the *loss* function $\ell : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which introduces further arbitrariness to the optimisation. It is common to add to the risk some form of *regularisation* $\Omega$, an extra penalty that biases the optimisation towards solutions minising a certain complexity. For instance, typical regularisations act on the norm of the weights: $\Omega(\cdot) = \|\cdot\|_2^2$ or *ridge* regularisation (Hoerl & Kennard, 1970), $\Omega(\cdot) = \|\cdot\|_1$ or *lasso* regularisation, and a mixture of the two called *elastic net*. The regularisation relative strength $\lambda \geq 0$, as all the other parameters not directly trained together with the ANN weights, is called an *hyperparameter*.

Empirical risk minimisation (ERM) (Vapnik, 1992) is a widely-common optimisation framework for DNN training. The algorithm defines the rules for the weight updates during the *training* phase, introducing a discrete-time dynamics to the learning problem. The work horses of ML methods are a family of first-order gradient-based algorithms derived as variants of the simple yet effective *gradient descent* (GD) algorithm, which can be described as follows:

At time $t = 0$, initialise $\boldsymbol{W}^{(0)}$, often at random from some prior distribution: $\boldsymbol{W}^{(0)} \sim \mathrm{P}_0$;

At time $0 < t <$`max_steps`, update the weights with GD on the empirical risk:

$$\boldsymbol{W}^{(t+\mathrm{d}t)} \leftarrow \boldsymbol{W}^{(t)} - \mathrm{d}t\, \nabla_{\boldsymbol{W}} \hat{\mathcal{R}}\left(\boldsymbol{W}^{(t)}, \mathcal{D}, \ell\right).$$

The hyperparameter $\mathrm{d}t$, called *learning rate*, measures the length of each time step and must be tuned properly.[4] Stochastic Gradient Descent (SGD) (Robbins & Monro, 1951), a widely adopted variant of GD where only a subset of the data is used at each training step, is extensively discussed in Part 2 of the manuscript. Other tricks accelerating the optimisation of GD include *momentum* (Polyak, 1964) and *Nesterov accelerated gradient* (Nesterov, 1983). Finally, the *back-propagation* algorithm introduced by Rumelhart et al. (1986) and based on the chain-rule derivative, allows to train multi-layer architectures efficiently. For a comprehensive review on training DNNs, see (Bottou, 2010). Some recently introduced biologically-plausible alternatives to back-propagation are discussed in the perspectives of the thesis (see Part 3).

**The generalisation problem** — What crucially distinguishes learning from optimisation is that the latter is only concerned with the minimisation of the objective function. On the contrary, ERM is just the beginning of the story for the ML pipeline. Indeed, the true learning objective would be the *population* risk:

$$\mathcal{R}\left(\boldsymbol{W},\ell\right)=\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathrm{P}_{\boldsymbol{x},\boldsymbol{y}}}\left[\ell\left(\hat{\boldsymbol{y}}_{\boldsymbol{W}}\left(\boldsymbol{x}\right),\boldsymbol{y}\right)\right],$$

which is unaccessible in practical situations. Still, the ultimate goal of learning is not only to perform well on the dataset used for training, but to robustly predict the output from previously unseen data points, a property known as *generalisation*. In other words, we are not looking for *any* global minima of the empirical risk, but for those that generalise well.

In order to test the generalisation properties of an algorithm, it is paramount to allocate a fraction of the data for this purpose. Moreover, given the usually large number of hyperparameters to be optimised on top of the weights, we should also keep some data for this purpose. The dataset is then split into training, validation, and test set, usually of approximately 70%/10%/20% the total size respectively (Goodfellow et al., 2016). The loss averaged on the test set is then used as a finite-size proxy for the population risk.

However, the test loss may not be a good measure for the performance, and other metrics are usually preferred to compute the *generalisation error*. Common choices are instead the average misclassification rate (a.k.a. 0/1 error) for classification:

$$\varepsilon_{\text{gen}}\left(\mathcal{D}\right)=\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathrm{P}_{\boldsymbol{x},\boldsymbol{y}}}\left[\mathbb{1}\left(\hat{\boldsymbol{y}}_{\mathcal{D}}\left(\boldsymbol{x}\right)\neq\boldsymbol{y}\right)\right],$$

where $\mathbb{1}$ denotes the indicator function, and the Mean Squared Error (MSE) for regression:

$$\mathrm{MSE}\left(\mathcal{D}\right)=\frac{1}{2}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathrm{P}_{\boldsymbol{x},\boldsymbol{y}}}\left[\left\|\hat{\boldsymbol{y}}_{\mathcal{D}}\left(\boldsymbol{x}\right)-\boldsymbol{y}\right\|_2^2\right],$$

for a given training dataset $\mathcal{D}$, where the average on previously unseen data points $(\boldsymbol{x},\boldsymbol{y})$ is approximated by the average over the test set. The goal of learning is then

[4] The learning rate can be either fixed or time dependent, in which case one needs to establish an appropriate learning rate schedule $\{(\mathrm{d}t)_t\}_{t\geq 0}$. Celebrated learning rate schedules are *Adagrad* (Duchi et al., 2011) and *Adam* (Kingma & Ba, 2014). Throughout this manuscript, we focus on constant learning rate.

to attain a small *generalisation gap*, i.e., the difference between the population and training errors. Interestingly, the empirical risk profile, or *loss landscape*, navigated by the algorithm is often highly non-convex. Nevertheless, in practice SGD algorithms are able to find minima that generalise well. This is a key mystery, yet to be fully understood, that is further discussed in Part 2.

The basic design principles presented above were known long time before the advent of modern DL (Schmidhuber, 2015) and, despite all the additional complexities of state-of-the-art systems, the building blocks remain the same. However, the precise interplay of the architecture, the task and the algorithm in determining the generalisation performance still defies theoretical understanding (Zdeborová, 2020). The theory is hindered by the huge dimensionality of the problem space where DNNs typically operate, that defies standard mathematical techniques producing counter-intuitive phenomena. The origin of this dauntingly high dimensionality is three-fold: the size of the training set, the dimension of each data point, and most-importantly the number of learnable parameters. Such large parameter regions that we aim at describing are impossible to explore directly, since the number of points required to sample uniformly a given volume grows exponentially with the dimension. This statistical challenge is called the *curse of dimensionality*. A comprehensive knowledge of the role played by each component and its translation into implementation guidelines is the Holy Grail of DL theory, since it would affect dramatically the work of practitioners by saving precious time now devoted to hyperparameter tuning and trial-and-error heuristics.

## The statistical physics perspective

A long history of cross-fertilisation ties the fields of ML and statistical physics together, as testified by the ample collection of classical and new references on the topic (Seung et al., 1992a; Watkin et al., 1993; Opper & Kinzel, 1996; Nishimori, 2001; Engel & Van den Broeck, 2001; Coolen et al., 2005; Bahri et al., 2020; Gabrié, 2020). Indeed, theory and tools from disordered systems and glassy physics are particularly well-suited to study the statistical properties of data-driven learning. Moreover, advanced mean-field methods developed in this context (see, for instance, Opper & Saad (2001) and references therein) offer a good approximation strategy to face the curse of dimensionality. This line of investigation was started in the 80's with the Hopfield model for associative memory (Hopfield, 1982; Amit et al., 1985a,b). The study of feedforward neural networks was pioneered by Gardner with two influential papers (Gardner, 1987, 1988) paving the way for the study of the *capacity*, i.e., the maximum number of training points that can be correctly classified by a perceptron, and the *learning curves*, i.e., the loss and the generalisation error as a function of the training-set size. Gardner and Derrida also initiated the study of the perceptron beyond the capacity limit (Gardner & Derrida, 1988), a direction further pursued in the 90's (Majer et al., 1993; Bouten & Derrida, 1994; Györgyi & Reimann, 2000; Györgyi, 2001).

Statistical physics looks at learning problems as high-dimensional dynamical systems of strongly correlated degrees of freedom in a quenched disorder and aims at providing a statistical description of the observed macroscopic behaviours. This perspective fills the existing gap between the often prohibitive mathematical rigor required by *statistical learning theory*[5] and the product-oriented engineering approach adopted in applications. The recent advances in DL applications renewed the physicists interest in the field, leading to a revival of this approach and a number of interesting new developments that are discussed in the introductions to Parts 1 and 2. Below, we remind some of the crucial ingredients of the statistical physics approach to learning, that are largely exploited in the rest of the manuscript.

**Typical VS worst-case scenario —** ANNs with a single hidden layer can represent any continuous function on a compact subset of $\mathbb{R}^d$. This remarkable *universal approximation* result was first derived by Cybenko (1989) for the sigmoid activation function, and then extended to arbitrary bounded non-constant activations by Hornik (1991). Similar results for DNNs are also available (Lu et al., 2017). However, the existence of the solution does not guarantee that it can be easily found by an algorithm, and another result by Blum & Rivest (1988) indeed states that the training of even very simple ANNs is $\mathcal{NP}-$hard. Learning is therefore computationally intractable for worst-case tasks, yet, in practice, simple first-order gradient-based algorithms can find good solutions.

This open puzzle suggests that the cases of interest lie in a very special subset of all possible tasks. The statistical physics of learning relies on *typical-case analysis*, in contrast with the *worst-case analysis*, which is focused on deriving statistical worst-case bounds on the generalisation gap. The latter is the approach commonly adopted in the realm of statistical learning theory and the Probably Approximately Correct framework introduced by Valiant (1984). These bounds are based on different measures of the *model capacity*[6], such as the Vapnik-Chervonenkis dimension (Vapnik, 1999b) or the more recent Rademacher complexity (Bartlett & Mendelson, 2002). However, these bounds often result in over-pessimistic predictions and fail to capture the quality of the performance observed in practice while training DNNs. On the other hand, one can consider the *average* performance over all possible realisations of training datasets, similarly as an average over quenched disorder in statistical mechanics. Computing averages requires the assumption of a known *generative model* for the data. These averages are particularly meaningful in the limit of high dimensions, or *thermodynamic* limit in the physics language.

**Quenched VS annealed averages —** Throughout this thesis, we focus on *batch learning* settings, where the training set is kept fixed during the whole training phase, while the weights of the ANN evolve in time. Therefore, the data are drawn at random and fixed, inducing a loss landscape that the optimisation algorithm has to navigate. The data thus play the role of *frozen* or *quenched* disorder, that does not change at the time scale of the fast evolving degrees of freedom (the weights), similarly as impurities trapped in a material under fast cooling, whence the name

[5] A theoretical framework drawing from statistics and functional analysis to study the properties of learning algorithms. Classical references are Vapnik (1999a,b).

[6] The capacity of a model quantifies its expressivity or richness. In other words, it refers to the ability of a model to fit a large number of functions.

*quenched* average. On the contrary, we refer to *annealed* averages to indicate averages over random variables that evolve on the fast time scale, on the same footing as the network weights. In general, quenched disorder introduces *frustration* to physical systems, a situation where it is impossible to satisfy all the constraints in the Hamiltonian (our empirical risk) leading to the coexistence of multiple local minima at the same energy level, a distinctive feature of glassy systems.

**The thermodynamic limit —** Training a supervised learning model consists in minimising a cost function that depends on the parameters of the ANN and on the dataset. DNNs usually operate in the *overparametrised* regime, where the number of parameters can reach the order of $10^6 - 10^7$ and largely exceeds the number of training samples, typically around $10^4 - 10^5$ and going up to $10^6 - 10^7$ only for the largest (open) datasets (Wu et al., 2019). In addition, each data point is "fat", lying in dimensions up to $10^6$ for large-size images. Thus, the thermodynamic limit is a good approximation. Crucially, the observables quantifying the performance – e.g., the empirical risk or the errors – enjoy the *self-averaging* property in the thermodynamic limit, meaning that their probability measure concentrates around its typical value. Therefore, in high dimensions, the average case is representative of what is observed in practice for a given dataset.

**Exactly solvable models —** In order to compute these averages we need to introduce the specific form of the distribution that generated the data. The search for realistic data distributions has been surging in recent years, pointing towards very interesting directions for modelling the geometry of real data (Chung et al., 2016, 2018; Mézard, 2017; Goldt et al., 2020; Cohen et al., 2020; Gerace et al., 2020). In parallel, the study of simple models of synthetic tasks is arguably worth per se. Theoretical physics largely relies on models able to capture the essential features of a problem while neglecting the details; the Ising model of ML theory is yet to be found and the hunt for good candidates is currently very active. Solvable models leading to exact solutions are useful for different reasons. First, they provide a controlled setting where experimental observations can be reproduced and established on firmer theoretical ground. Second, they allow to identify universal properties capturing general behaviours and possibly offering a unifying look on the proliferation of experimental observations. Finally, they can help to discriminate between relevant and irrelevant details by revealing missing elements for an accurate description. Exact solutions can thus orient sequential model improvements and foster the development of new analytic tools in a virtuous circle between theory and applications.

In summary, DL is an exploding field of research where outstanding empirical accomplishments coexist with a myriad of theoretical surprises. The research community is actively committed to unlock this "black box" inspecting the structural properties of DNNs (see, e.g., Mallat (2016); Raghu et al. (2017)). In this thesis, we join this research effort offering the tools of statistical physics to grasp some of the mechanisms underlying learning. We recall the *questions* asked by Leo Breiman and notice that we can identify in them two different points of view. Indeed, while the first two questions address more general *static* properties of the network (the final performance, the number of parameters), the last two questions directly concern more specific *dynamical* properties, regarding the algorithmic training procedure. This manuscript is organised along these two main directions of investigation.

# Summary of the contributions

This Ph.D. thesis is inscribed in the search for theoretical foundations of machine learning introduced in the previous chapter. Adopting a statistical physics perspective, we have investigated the generalisation properties and the training dynamics of artificial neural networks (ANNs) via exactly solvable models. We provide here a summary of the main contributions that are covered in this dissertation.

**The models** — Our study focuses on three simple yet prototypical models. The first two describe classification tasks:

- *Binary classification of Gaussian mixtures*: the data are drawn independent and identically distributed (i.i.d.) from two Gaussian clouds centered at the $d-$dimensional vectors $\pm\boldsymbol{w}^*/\sqrt{d} \in \mathbb{R}^d$, while the labels reflect the memberships in the clusters. We also refer to this setting as the *two-cluster dataset*, as opposed to a slight variant, the *three-cluster dataset*, that we adopt as a prototype of non-linear classification task. The latter is still a binary classification of Gaussian mixtures, but the two clouds centered at $\pm\boldsymbol{w}^*/\sqrt{d}$ belong to the same class, while the other class is represented by a Gaussian cloud centered at the origin. The three-cluster dataset is therefore non-linearly separable by definition and allows us to study non-convex loss functions. For both models, we consider learning via single-layer ANNs with activation functions that are expressive enough to perform the corresponding classifications.
- *Multi-class teacher-student classification*: each data sample is drawn i.i.d. from a standard Gaussian distribution in dimension $d$, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$, while the corresponding label is generated by a *teacher* matrix $\boldsymbol{W}^* \in \mathbb{R}^{d\times k}$ as the argmax of the scalar product between the sample and the teacher: $y = \underset{l\in\{1,\dots,k\}}{\operatorname{argmax}} \left(\boldsymbol{x}^\top \boldsymbol{W}_l^*\right)$. The goal of the ANN – also called *student* in this context – is to learn the teacher's weights. Historically, the teacher-student perceptron, originally called "model B", was introduced in Gardner & Derrida (1989). A comprehensive presentation of the teacher-student setting can be found in Nishimori (2001).

The third one represents a regression task:

- *Sign retrieval*: each data sample $\boldsymbol{x}$ is drawn i.i.d. from a standard Gaussian distribution in dimension $d$, while the corresponding label is the absolute value of the scalar product between the sample and a teacher vector $\boldsymbol{w}^* \in \mathbb{R}^d$: $y = |\boldsymbol{x}^\top \boldsymbol{w}^*|/\sqrt{d}$. This task also belongs to the class of teacher-student problems. Sign retrieval would amount to a simple linear regression if only the signs of the labels were known, whence its name.

**The research questions —** We aim at contextualising and understanding the limitations of training ANNs in high dimensions. We therefore consider the thermodynamic limit where both the number of samples $n$ and the data dimension $d$ tend to infinity, at a fixed rate $\alpha = n/d \sim \mathcal{O}_d(1)$ named *sample complexity.*

On the one hand, we are interested in understanding how the model parameters (the sample complexity, the data structure, the regularisation strength, ...) impact learning and whether, as these parameters change, we can identify phase transitions in the performance, similarly to what observed in statistical mechanics. In particular, it is useful to compare the performance of ANN models to the benchmark provided by the information-theoretically optimal one.

On the other hand, we are interested in investigating the learning dynamics of commonly-used algorithms, such as stochastic gradient descent (SGD). Indeed, while ANNs trained with SGD have achieved impressive performances, the theory behind this practical success is largely unexplained. A consensus has arisen that the answer requires tracking the full trajectory traversed during training, which is highly nontrivial. Indeed, the high dimension of the parameter space defies standard mathematical techniques. Moreover, SGD navigates a non-convex loss landscape following an out-of-equilibrium dynamics with a complicated state-dependent noise, whose characterisation is the subject of intense scrutiny in the ML theory community.

**The results —** The models presented above are used as prototypical high-dimensional examples to explore these research questions.

The binary Gaussian mixture model (GMM) presented above serves us as a working example to discuss and illustrate in a unified fashion many interesting phenomena that are observed in practice. In Article 1, we focus on the static properties of the GMM problem landscape. We study the performance of regularised convex classifiers and provide asymptotic expressions for the train and test errors, derived both from the heuristic replica method and the rigorous Gordon's inequality technique. We then apply our theoretical findings to shed light on the role of the different model parameters on the performance. First, we identify a sharp phase transition in the sample complexity from linear to non-linear separability of the data, whose critical threshold depends on the data structure (the variance of the clusters' noise and the clusters' *unbalance*, i.e., their relative size). At this threshold value, we observe an interpolation "peak" in the generalisation error, similarly as what is observed in practical applications. We also investigate the role of the regularisation strength and observe that regularisation smoothens the interpolation peak until it eventually disappears. We find out that, surprisingly, the information-theoretically optimal performance can be achieved at *infinite* regularisation in the case of balanced clusters. We then show that this peculiar behaviour does not hold anymore as soon as the clusters are unbalanced.

In Article 3, we consider the dynamics of gradient-based training algorithms performing classification of the GMM. We manage to derive the first analytic description of the full trajectory of the learning curves of *mini-batch* SGD, i.e., the realistic case where the available examples are used more than once. To this end, we use dynamical mean-field theory (DMFT) from statistical physics. The result is a closed set of integro-differential equations that must be solved numerically in a self-consistent way. Our numerical solution of the DMFT equations shows excellent agreement with experiments at finite time step, for both convex and non-convex losses and at relatively low dimension $d \approx 10^2 - 10^3$, even though the theory is derived in the thermodynamic limit. A large part of the theory of gradient-based algorithms focuses on the *flow*, i.e., continuous-time limit of the algorithm. However, this limit is not properly defined for SGD, whose flow limit is therefore often treated under approximations as a Langevin-like process. To overcome this problem, we also introduce a variant of the sampling procedure, that we call *persistent* SGD (p-SGD), since in this case all samples are independently endowed with some persistence and spend some typical time in the training mini batch. This *persistent* variant of SGD admits a well-defined continuous-time limit for the sampling procedure, that is encoded by a two-state Markov process. For all discrete time steps, SGD can be recovered from p-SGD with a specific choice of the persistence time without resorting to any approximations. Moreover, p-SGD introduces interesting features to the sampling noise.

In Article 4, we characterise the late-time dynamics and quantify the noise magnitude of SGD and p-SGD. We choose the simple convex setting provided by the binary GMM in order to isolate the algorithmic noise from other possible sources of noise in the dynamics, such as the roughness of the landscape. In the underparametrised regime, where the final training error is positive, the SGD dynamics reaches a stationary state and we define an effective temperature from an effective fluctuation-dissipation theorem (FDT), computed from DMFT. We use the effective temperature to quantify the magnitude of SGD noise as a function of the model parameters. In the overparametrised regime, where the training error vanishes, we measure the noise magnitude of SGD by computing the average distance between two replicas of the system with the same initialisation and two different realisations of the mini-batch sampling. We find that the two noise measures behave similarly as a function of the model parameters. Moreover, we observe that noisier algorithms lead to wider decision boundaries of the corresponding constraint satisfaction problem (CSP).

While SGD seems to outperform its deterministic counterpart (GD) in applications, clear theoretical boundaries on this statement have not been established yet. To address this question, in Article 5 we consider an intrinsically hard problem, the sign retrieval problem presented above, as a benchmark high-dimensional non-convex task to assess how different sources of algorithmic noise affect the generalisation properties. Therefore, we consider GD, SGD, p-SGD and the Langevin algorithm and we perform a series of numerical simulations in order to assess their performance as a function of the model parameters (mini-batch size, persistence time, Langevin temperature). Our experimental findings reveal that, in the considered problem, stochasticity is crucial for generalisation. We also shed light on the qualitative difference between the sources of noise in the algorithms. In particular, we notice that (p-)SGD, due to the particular structure of its noise, has a built-in self-annealing protocol that allows it to outperform GD. We then use informed initialisations, i.e., "warm" starts close to the signal $\boldsymbol{w}^*$, to probe the interplay of the loss landscape with the algorithm. We find that GD can get trapped even very close to the signal, while perfect recovery can be reached starting from less informed initialisations. Moreover, persistence plays a crucial role in avoiding to get stuck in local minima. We then apply DMFT to provide an analytic characterisation of the full trajectory of the algorithms in the high-dimensional limit. We use the theoretical curves as a baseline to show that the observed behaviour is not due to finite-size or finite-learning-rate effects.

A considerable part of modern machine learning practice concerns multi-class classification. However, while the generalisation performance of single-layer teacher-student perceptron on i.i.d. Gaussian inputs and binary labels has been widely studied in high-dimensional learning theory, an analogous analysis for the corresponding multi-class teacher-student perceptron was missing. In Article 2, we fill this gap by deriving and evaluating asymptotic expressions for the errors obtained via ERM and for the information-theoretically optimal performance in the multi-class teacher-student model presented above. We first examine a Rademacher prior for the teacher matrix and we unveil a first-order phase transition in the performance, in analogy with the two-classes case. We then consider a Gaussian teacher prior and we use our theoretical results to explore the performance of ridge-regularised ERM with convex losses. In particular, we discuss two widely-used loss functions: the square and cross-entropy losses. We compare optimally regularised cross-entropy classification to the information-theoretically optimal classifier, and we conclude that for three classes the two are extremely close.

# Synthèse en français

Cette thèse s'inscrit dans la recherche des fondements théoriques de l'apprentissage automatique. Nous avons étudié les propriétés de la généralisation et de la dynamique d'entraînement des réseaux de neurones artificiels à travers des modèles exactement résolubles en utilisant des outils de la physique statistique. Nous présentons ici un résumé des contributions principales qui font l'objet de cette thèse.

**Les modèles —** Notre étude porte sur trois modèles simples mais prototypiques. Les deux premiers décrivent des tâches de classification :

- *Classification binaire des mélanges gaussiens* : les données sont indépendantes et identiquement distribuées (i.i.d.) à partir de deux nuages gaussiens centrés sur les vecteurs $\pm\boldsymbol{w}^*/\sqrt{d} \in \mathbb{R}^d$, tandis que les étiquettes reflètent l'appartenances à un des nuages. Nous considerons aussi un mélange gaussien à trois nuages, que nous adoptons comme prototype de classification non linéaire. Il s'agit toujours d'une classification binaire, mais les deux nuages centré sur $\pm\boldsymbol{w}^*/\sqrt{d}$ appartiennent à la même classe, tandis que l'autre classe est représentée par un troisième nuage centré à l'origine. Ce melange à trois nuages est donc non séparable linéairement par définition et nous permet d'étudier des fonctions de perte non convexes. Pour les deux modèles, nous considérons l'apprentissage des réseaux de neurones artificiels monocouches.

- *Classification multiclasse dans le modèle enseignant-étudiant* : chaque échantillon de données est tiré i.i.d. d'une distribution gaussienne en dimension $d$, $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, tandis que l'étiquette correspondante est générée par une matrice "enseignante" $\boldsymbol{W}^* \in \mathbb{R}^{d\times k}$ comme l'argument du maximum du produit scalaire entre l'échantillon et l'enseignant: $y = \underset{l\in\{1,\ldots,k\}}{\text{argmax}}\left(\boldsymbol{x}^\top \boldsymbol{W}_l^*\right)$. L'objectif du réseau de neurones artificiels – également appelé "étudiant" dans ce contexte – est d'apprendre la matrice $\boldsymbol{W}^*$ de l'enseignant. Du point de vue historique, le perceptron enseignant-élève, initialement appelé « modèle B », était introduit par Gardner & Derrida (1989). Une présentation complète du modèle enseignant-étudiant peut être trouvé dans Nishimori (2001).

Le troisième modèle représente une tâche de régression :

- *Récupération des signes* : chaque échantillon de données $\boldsymbol{x}$ est tiré i.i.d. à partir d'une distribution gaussienne de moyenne zéro et covariance identité, en dimension $d$, tandis que l'étiquette correspondante est la valeur absolue du produit scalaire entre l'échantillon et un vecteur enseignant $\boldsymbol{w}^* \in \mathbb{R}^d$ : $y = |\boldsymbol{x}^\top \boldsymbol{w}^*|/\sqrt{d}$. Cette tâche aussi appartient à la classe des problèmes enseignant-élève. La récupération des signes reviendrait à une simple régression linéaire si les signes des étiquettes étaient connus, d'où son nom.

**Les questions scientifiques —** Nous visons à comprendre les limites de l'entraînement des réseaux de neurones artificiels en haute dimension. À cette fin, nous considerons la limite "thermodynamique" où le nombre d'échantillons $n$ et la dimension de chaque donnée $d$ tendent vers l'infini, à un taux fixé $\alpha = n/d \sim \mathcal{O}_d(1)$.

D'une part, nous visons à comprendre comment les paramètres du modèle (tels que le nombre d'échantillons, la structure des données, la régularisation, ...) impactent l'apprentissage et si nous pouvons identifier des transitions de phase dans la performance tout en ajustant ces paramètres, de manière semblable à ce que l'on observe en mécanique statistique. En particulier, il est utile de comparer les performances des modèles de réseaux de neurones à la performance optimale du point de vue de la théorie de l'information.

D'autre part, nous nous intéressons à l'étude de la dynamique d'apprentissage des algorithmes tels que la descente de gradient stochastique (SGD). En effet, les réseaux de neurones artificiels entraînés avec SGD réalisent des performances impressionnantes dans les applications. Toutefois, la théorie derrière ce succès pratique demeure largement inexpliquée. La réponse nécessite de suivre la trajectoire complète parcourue pendant l'entraînement, ce qui est très compliqué du point de vue de l'analyse. En effet, la dimension élevée de l'espace des paramètres défie les techniques mathématiques traditionnelles. De plus, SGD navigue dans un paysage d'énérgie non convexe suivant une dynamique hors d'équilibre avec un bruit complexe dépendant de sa position.

Les modèles présentés ci-dessus sont utilisés comme exemples prototypiques en grande dimension pour explorer ces questions de recherche.

**Les résultats —** Le modèle de classification binaire de mélanges gaussiens présenté ci-dessus nous sert d'exemple pour discuter de manière unifiée de nombreux phénomènes intéressants qui sont observés dans la pratique.

Dans l'article 1, nous nous concentrons sur les propriétés statiques du paysage d'énérgie du problème. Nous étudions les performances des classificateurs convexes régularisés et calculons des expressions asymptotiques pour les erreurs, dérivées à la fois de la méthode heuristique des répliques et de la technique rigoureuse de l'inégalité de Gordon. Nous appliquons ensuite nos découvertes théoriques pour éclairer le rôle des différents paramètres du modèle. Tout d'abord, nous identifions une transition de phase de séparabilité linéaire à non-séparabilité linéaire des données, en augmentant le nombre d'échantillons par rapport á la dimensionnalité du probléme, dont le seuil critique dépend de la structure des données (la variance du bruit des nuages gaussiens et leur niveau de déséquilibre, c'est-à-dire leur taille relative). A cette valeur seuil, nous observons un "pic" d'interpolation dans l'erreur de généralisation, de la même manière que ce qui est observé dans les applications pratiques. Nous étudions également le rôle de la régularisation et nous découvrons que, de façon surprenante, la performance théoriquement optimale peut être atteinte à une régularisation infinie dans le cas des nuages équilibrés. Nous montrons alors que ce comportement particulier ne tient plus dès que les clusters sont déséquilibrés.

Dans l'article 3, nous considérons la dynamique des algorithmes d'entraînement effectuant la classification binaire des mélanges gaussiens décrits ci-dessus. Nous parvenons à dériver la première description analytique de la trajectoire de l'algorithme "*multi-pass*" SGD, c'est-à-dire le cas réaliste où les exemples disponibles sont utilisés plusieures fois. À cette fin, nous utilisons la théorie dynamique du champ moyen (DMFT) de la physique statistique. Le résultat est un ensemble d'équations intégro-différentielles qui doivent être résolues numériquement de manière auto-cohérente. Notre solution numérique des équations de DMFT montre un accord excellent avec les expériences à pas de temps fini, aussi pour des fonctions de perte non-convexes et en dimension finie ($d \approx 10^2 - 10^3$). Une grande partie de la théorie des algorithmes basés sur les gradients se concentre sur le flux de l'algorithme, c'est-à-dire la limite de temps continu. Cependant, cette limite n'est pas correctement définie pour SGD, dont la limite de flux est donc souvent traitée sous des approximations comme un processus de type Langevin. Afin de surpasser ce problème, nous introduisons également une variante de la procédure d'échantillonnage, que nous appelons SGD *persistant* (p-SGD), puisque dans ce cas tous les échantillons sont indépendamment dotés de persistance et passent un certain temps typique dans le mini-lot d'entraînement. Cette variante persistante de SGD admet une limite de temps continu bien définie pour la procédure d'échantillonnage, qui est répresentée par un processus de Markov à deux états. Pour tous les pas de temps discrets, SGD peut être récupéré à partir de p-SGD avec un certaine choix du temps de persistance sans recourir à aucune approximation. De plus, p-SGD introduit des caractéristiques intéressantes dans le bruit d'échantillonnage de p-SGD.

Dans l'article 4, nous caractérisons la dynamique après longtemps et quantifions l'amplitude du bruit de SGD et p-SGD dans la classification binaire des mélanges gaussiens. Nous choisissons ce problème au paysage de perte convexe afin d'isoler le bruit algorithmique des autres sources de bruit possibles dans la dynamique, comme la rugosité du paysage d'énérgie. Dans le régime sous-paramétrisé, où l'erreur d'entraînement finale est positive, la dynamique de SGD atteint un état stationnaire et on définit une température effective à partir d'un théorème de fluctuation-dissipation (FDT) effectif, calculé à partir de la DMFT. Nous utilisons cette température effective pour quantifier l'amplitude du bruit de SGD en fonction des paramètres du modèle. Dans le régime sur-paramétrisé, où l'erreur d'apprentissage s'annule, nous mesurons l'amplitude du bruit de SGD en calculant la distance moyenne entre deux répliques du système avec la même initialisation et deux réalisations différentes de l'échantillonnage mini-lot. Nous trouvons que les deux mesures de bruit se comportent de manière similaire en fonction des paramètres. De plus, nous observons que les algorithmes plus bruyants conduisent à des solutions plus robustes.

Alors que SGD semble surpasser son homologue déterministe (GD) dans les applications, les limites de cette déclaration n'ont pas encore été établies du point de vue théorique. Pour répondre à cette question, dans l'article 5, nous considérons un problème intrinsèquement difficile, la récupération des signes présentée ci-dessus, comme prototype de tâche non convexe en grande dimension pour évaluer comment différentes sources de bruit algorithmique affectent les propriétés de généralisation. Par conséquent, nous considérons GD, SGD, p-SGD et l'algorithme de Langevin et nous effectuons une série de simulations afin d'évaluer leurs performances en fonction des paramètres du modèle (taille du mini-lot, temps de persistance, température de Langevin). Nos résultats expérimentaux révèlent que, dans le problème considéré, la stochasticité est cruciale pour la généralisation. Nous avons également mis en lumière la différence qualitative entre les sources de bruit dans les algorithmes. En particulier, nous remarquons que le (p-)SGD, en raison de la structure particulière de son bruit, possède un protocole d'auto-recuit qui lui permet de surpasser GD. On utilise alors des initialisations informées, c'est-à-dire, des démarrages proches du signal, $\boldsymbol{w}^*$ pour sonder l'interaction du paysage de perte avec l'algorithme. Nous constatons que GD peut rester bloqué même très près du signal, alors qu'une récupération parfaite est accessible à partir d'initialisations moins renseignées. De plus, la persistance joue un rôle crucial pour éviter de rester coincé dans les minima locaux. Nous appliquons ensuite la DMFT pour fournir une caractérisation analytique de la trajectoire des algorithmes dans la limite de grande dimension. Nous utilisons la courbe théorique comme ligne de base pour montrer que le comportement observé n'est pas dû à des effets de taille finie ou de pas de temps finis.

Une partie considérable de la pratique moderne de l'apprentissage automatique concerne la classification multiclasse. Cependant, alors que la performance de généralisation du perceptron enseignant-élève à une seule couche sur entrées gaussiennes i.i.d. a été largement étudiée dans le cas binaire, la même analyse du perceptron enseignant-élève dans le cas multiclasse était manquante. Dans l'article 2, nous comblons cette lacune en dérivant et en évaluant des expressions asymptotiques pour les erreurs obtenues par la minimisation du risque empirique et pour les performances théoriquement optimales dans le modèle enseignant-élève de classification multiclasse présenté ci-dessus. Nous examinons d'abord un prior de Rademacher pour la matrice de l'enseignant et nous dévoilons une transition de phase du premier ordre dans la performance, en analogie avec le cas des deux classes. Nous considerons alors un prior gaussien pour l'enseignant et nous utilisons nos résultats théoriques pour explorer la performance de la minimisation du risque empirique régularisé avec fonctions de perte convexes. En particulier, nous discutons de deux fonctions de perte largement utilisées : la *cross-entropy* et la *squared loss*. Nous comparons la *cross-entropy* régularisée de manière optimale à la performance optimale en théorie de l'information, et nous concluons que pour trois classes les deux sont extrêmement proches.

# Organisation of the manuscript

The overview of this thesis has served as a warm-up on the key concepts underlying the statistical physics of learning and as an appetizer of some of the relevant open questions. The following dissertation expands on these points and is organised in two main parts: on the *statics* and on the *dynamics* of learning problems. Although these two perspectives are intertwined and contribute synergistically to our understanding of ML theory, we have decided to expose them separately in order to highlight their distinctive methods and questions. This dichotomy deliberately echoes the long-standing one in the physics of glassy systems. Here, we show the particular relevance of these complementary approaches in the context of ML theory.

In Part 1 we focus on ANNs at the end of training. Chapter 1.1 reviews the most common approaches to characterise the properties of the problem landscape and the different learning regimes encountered according to the region of hyper-parameter space where the problem lies. In Chapter 1.2, we introduce the binary Gaussian Mixture Model (GMM) for classification and we derive asymptotic equations to characterise the performance of regularised convex classifiers as well as the optimal one. We use this problem as the occasion to introduce the replica method. We discuss and explain some of the high-dimensional phenomena observed in this setting. In Chapter 1.3, we turn to multi-class classification in the teacher-student perceptron model. We derive asymptotic expressions for the errors obtained via ERM and we discuss its performance in relation to the information-theoretical optimum and to the Approximate Message-Passing Algorithm (AMP).

In Part 2, we turn our focus to the study of the dynamics of training algorithms. In Chapter 2.1, we briefly review the relevant literature on learning dynamics as well as some useful statistical physics methods to study out-of-equilibrium systems, such as dynamical mean-field theory (DMFT). In Chapter 2.2, we show how to track the high-dimensional training dynamics of SGD via DMFT in the GMM for binary classification. Chapter 2.3 explores some directions to characterise the out-of-equilibrium noise introduced by the mini-batch sampling procedure of SGD as a function of the model parameters. In Chapter 2.4, we introduce the sign retrieval problem as a benchmark task to compare different gradient-based algorithms and investigate the role played by stochasticity in navigating high-dimensional non-convex landscapes.

Finally, Part 3 presents some conclusions and perspectives for future work.

# Publications

This thesis manuscript is based on the following articles.

**Publications:**

1. *The role of regularisation in classification of high-dimensional noisy Gaussian mixture*, Francesca Mignacco, Florent Krzakala, Yue M. Lu, and Lenka Zdeborová. *International Conference on Machine Learning*, PMLR, 2020. p. 6874-6883.

2. *Dynamical mean-field theory for stochastic gradient descent in Gaussian mixture classification*, Francesca Mignacco, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborová. *Advances in Neural Information Processing Systems* 33 (2020): 9540-9550. Re-published in the "Machine Learning 2021" Special Issue, *Journal of Statistical Mechanics: Theory and Experiment* 2021, no. 12 (2021): 124008.

3. *Stochasticity helps to navigate rough landscapes: comparing gradient-descent-based algorithms in the phase retrieval problem*, Francesca Mignacco, Pierfrancesco Urbani, Lenka Zdeborová. *Machine Learning: Science and Technology* 2, no. 3 (2021): 035029.

4. *The effective noise of Stochastic Gradient Descent*, Francesca Mignacco, Pierfrancesco Urbani. J. Stat. Mech. (2022) 083405.

**Pre-print:**

5. *Learning curves for the multi-class teacher-student perceptron*, Elisabetta Cornacchia, Francesca Mignacco, Rodrigo Veiga, Cédric Gerbelot, Bruno Loureiro, Lenka Zdeborová. ArXiv preprint *arXiv:2203.12094*. Submitted for publication in *Advances in Neural Information Processing Systems* (2022).

# 1 - The statics of learning problems

## 1.1 - A brief introduction to the statics of learning

In this chapter, we introduce some useful concepts to understand the static properties of learning problems. This description amounts to the characterisation of the problem landscape, induced by the loss function, the architecture and the dataset. The goal is to study the minima of the lanscape, corresponding to the solutions of the optimisation problem presented to the ANN. Therefore, we are focusing on all the possible endpoints of training, regardless the algorithmic trajectory that may have led there.

This introduction is strongly biased towards the statistical physics approach to learning theory described in the *Motivation and background* and by no means aims at a comprehensive review of all possible approaches to the problem. Instead, we focus on the theoretical and methodological approaches that are necessary to understand the results of this thesis. In this regard, the statistical properties of the learning problem can be discussed in the framework of ensemble equilibrium theory and the equal probability Boltzmann principle, which laid the foundation of statistical mechanics. Crucially, as we further discuss in Part 2, such a general principle is still lacking for non-equilibrium theory, introducing a great challenge to the analysis of the learning dynamics.

**A Bayesian perspective on empirical risk minimisation —** The deep connection between the probabilistic approach of statistical mechanics and the study of inference (or learning) problems is naturally understood in a Bayesian probabilistic framework. In a statistical estimation (or ML) problem, the goal is to estimate (or learn) the parameters $\boldsymbol{W}$ from a parametric family of distributions $\mathrm{P}_{\boldsymbol{W}}$ (such as an ANN). To this end, Bayesian statistics (Bayes, 1763) relies on *prior information* $\mathrm{P}_{\boldsymbol{W}^*}$ on the ground truth $\boldsymbol{W}^*$. Notice that, in general, this ground truth can be arbitrarily complicated, but we consider the case in which the parametric family under consideration is expressive enough to include the ground truth.

For simplicity, from now on we specialise the discussion to the case of supervised learning problems. In this case, the ground truth generates the correlations between data and labels.We can therefore write the probability of a certain realisation of the parameters $\boldsymbol{W}$ given the dataset $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y}) = \{(\boldsymbol{x}_\mu, \boldsymbol{y}_\mu)\}_{\mu=1}^n$ making use of Bayes formula:

$$\mathrm{P}\left(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}\right) = \frac{\mathrm{P}\left(\boldsymbol{y}|\boldsymbol{W},\boldsymbol{X}\right)\mathrm{P}_{\boldsymbol{W}^*}(\boldsymbol{W})}{\mathrm{P}(\boldsymbol{X},\boldsymbol{y})}, \tag{1.1.1}$$

where $\mathrm{P}\left(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}\right)$ is called the *posterior* and $\mathrm{P}\left(\boldsymbol{y}|\boldsymbol{W},\boldsymbol{X}\right)$ the conditional *likelihood.* The normalisation factor $\mathrm{P}(\boldsymbol{X},\boldsymbol{y})$ is called *evidence* or – in the statistical physics literature – *partition function*, also denoted by $Z(\boldsymbol{X},\boldsymbol{y})$. The posterior can be equivalently rewritten as

$$\mathrm{P}\left(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}\right) = \frac{1}{Z(\boldsymbol{X},\boldsymbol{y})}\mathrm{e}^{-\beta\mathcal{H}(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y})}, \tag{1.1.2}$$

where the Hamiltonian $\mathcal{H}$ is given by

$$-\beta\mathcal{H}\left(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}\right) = \log \mathrm{P}\left(\boldsymbol{y}|\boldsymbol{W},\boldsymbol{X}\right) + \log \mathrm{P}_{\boldsymbol{W}^*}(\boldsymbol{W}), \tag{1.1.3}$$

so that the posterior clearly corresponds to the *Gibbs* (or *Boltzmann*) distribution of statistical mechanics at inverse temperature $\beta$, where each configuration $\boldsymbol{W}$ is weighted by its Hamiltonian energy rescaled by the temperature $T = 1/\beta$. Tuning the temperature allows us to explore different energy levels, while in the zero-temperature limit $T \to 0$ the Gibbs measure concentrates on the configurations at lowest energy, i.e., the *ground state*. The reader is referred to Nishimori (2001); Grassberger & Nadal (2012); Zdeborová & Krzakala (2016); Advani & Ganguli (2016) for a broader overview on the connection between statistical physics and Bayesian inference.

The empirical risk minimisation (ERM) framework introduced in the *background* chapter can be interpreted as the search for the ground state of a system with Hamiltonian given by the (possibly rescaled) empirical risk $\hat{\mathcal{R}}$, i.e.,

$$\mathcal{H}(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}) = \sum_{\mu=1}^{n} \ell\left(\hat{\boldsymbol{y}}_{\boldsymbol{W}}(\boldsymbol{x}_\mu), \boldsymbol{y}_\mu\right) + \lambda\Omega(\boldsymbol{W}), \tag{1.1.4}$$

where the loss function plays the role of the log-likelihood and the regularisation of the log-prior. Note that, in all realistic cases, neither the true likelihood nor the true prior are known and the empirical risk contains only our best approximations.

This perspective brings useful insights on the design of the estimation (or learning) procedure. In particular, the crucial role played by the prior has recently attracted a great research interest, especially in relation to the *implicit* information induced by the network design. Indeed, in addition to the explicit regularisation term appearing in Eq. (1.1.4) that clearly acts as a prior, for instance enforcing some constraints on the norm of the solution, the specific structure of the architecture is also known to bias the optimisation acting as a prior (see, e.g., Ulyanov et al. (2018) and references therein). Moreover, the optimisation algorithm itself can implicitly bias the dynamics towards solutions that minimise some hidden measure of complexity. This interesting phenomenon is known as *implicit regularisation* (Neyshabur, 2017) and could explain the ability of DNNs to find good solutions without incurring in overfitting despite fitting the training data.

Training an ANN via ERM is just one of the possible procedures – and in many practical settings, the most viable one – to solve the estimation problem presented above. We consider below some other estimators that is of interest for our analysis.

**Bayes-optimal estimator —** Bayes-optimal (BO) estimation corresponds to the idealistic setting where the distributions generating the data and the ground truth are known. Exploiting this additional knowledge, the BO estimator achieves the best possible performance among all estimators having access to the training dataset and therefore provides a theoretically-optimal baseline for practical algorithms. The BO setting enjoys the property that an assignment of $\boldsymbol{W}$ drawn uniformly at random from the posterior distribution becomes statistically equivalent to an assignment drawn from the ground truth distribution:

$$\mathbb{E}\left[f(\boldsymbol{W}_1, \boldsymbol{W}_2)\right] = \mathbb{E}\left[f(\boldsymbol{W}^*, \boldsymbol{W}_3)\right], \tag{1.1.5}$$

for all continuous bounded functions $f$, where $\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3$ are sampled from the posterior distribution and $\boldsymbol{W}^*$ from the ground truth. This is a consequence of the tower property of conditional expectation and results in a number of interesting properties known as *Nishimori conditions* (Opper & Haussler, 1991; Iba, 1999; Nishimori, 2001; Krzakala & Zdeborová, 2011).

**Minimum mean squared error** — The mean squared error

$$\text{MSE}(\hat{\boldsymbol{W}}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}}\left[\|\hat{\boldsymbol{W}} - \boldsymbol{W}\|_F^2\right], \tag{1.1.6}$$

where $\|\cdot\|_F$ stands for the Frobenius norm, is a very natural measure of the performance of the estimator $\hat{\boldsymbol{W}}$, where the average is a compromise justified by the fact that in practice we do not have access to the ground truth parameter $\boldsymbol{W}^*$. The minimum mean squared error (MMSE) estimator is readily obtained by taking the derivative of Eq. (1.1.6) with respect to $\hat{\boldsymbol{W}}$:

$$\hat{\boldsymbol{W}}^{\text{MMSE}} = \underset{\hat{\boldsymbol{W}}}{\text{argmin}}\,\text{MSE}(\hat{\boldsymbol{W}}) = \mathbb{E}_{\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}}\left[\boldsymbol{W}\right]. \tag{1.1.7}$$

Unfortunately, computing the average over the posterior is often intractable in high dimensions and one may resort to Markov-Chain Monte Carlo (MCMC) methods to sample $\text{P}(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y})$. However, sampling methods become prohibitive in very high dimensions. To overcome this difficulty, statistical physics come to the rescue offering very powerful heuristic methods to deal with these high-dimensional averages.

**Maximum a posteriori** — The maximum a posteriori (MAP) is a point-wise estimator maximising directly the posterior distribution

$$\begin{aligned}\hat{\boldsymbol{W}}^{\text{MAP}} &= \underset{\boldsymbol{W}}{\text{argmax}}\,\log \text{P}\left(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}\right) \\ &= \underset{\boldsymbol{W}}{\text{argmin}}\,\mathcal{H}(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}),\end{aligned} \tag{1.1.8}$$

where we see from Eq. (1.1.2) that this is exactly equivalent to the ERM problem.

**Maximum likelihood** — The maximum likelihood estimator (MLE) $\hat{\boldsymbol{W}}^{\text{ML}}$ is one of the most popular statistical estimators. In particular, it is central in the *frequentist* approach to statistical estimation that, at variance with the Bayesian one, focuses on the worst-case scenario. The MLE consists in maximising the probability of observing the given labels, i.e., to restrict the maximisation to the (conditional) likelihood term in Eq. (1.1.1) without assuming any prior information. Since the $n$

samples are taken i.i.d., the estimator can be decomposed as follows:

$$\begin{aligned}\hat{\boldsymbol{W}}^{\mathrm{ML}} &= \underset{\boldsymbol{W}}{\operatorname{argmax}} \frac{1}{n} \sum_{\mu=1}^{n} \log \mathrm{P}\left(\boldsymbol{y}_\mu | \boldsymbol{x}_\mu, \boldsymbol{W}\right) \\ &= \underset{\boldsymbol{W}}{\operatorname{argmax}} \int \mathrm{d}\boldsymbol{x} \int \mathrm{d}\boldsymbol{y}\ \log \mathrm{P}\left(\boldsymbol{y} | \boldsymbol{x}, \boldsymbol{W}\right) \frac{1}{n} \sum_{\mu=1}^{n} \delta(\boldsymbol{y}-\boldsymbol{y}_\mu)\delta(\boldsymbol{x}-\boldsymbol{x}_\mu) \\ &= \underset{\boldsymbol{W}}{\operatorname{argmax}} \int \mathrm{d}\boldsymbol{x} \int \mathrm{d}\boldsymbol{y}\, \hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{y},\boldsymbol{x})\ \log \mathrm{P}\left(\boldsymbol{y} | \boldsymbol{x}, \boldsymbol{W}\right) \\ &= \underset{\boldsymbol{W}}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{x}\sim \hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{x})} \int \mathrm{d}\boldsymbol{y}\, \hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{y}|\boldsymbol{x})\ \log \mathrm{P}\left(\boldsymbol{y} | \boldsymbol{x}, \boldsymbol{W}\right).\end{aligned} \tag{1.1.9}$$

Therefore, the MLE can be interpreted as closing the average statistical distance between the model distribution $\mathrm{P}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})$ and the empirical conditional distribution $\hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{y}|\boldsymbol{x}) = \hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{y},\boldsymbol{x})/\hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{x})$, where $\hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{y},\boldsymbol{x}) = \sum_{\mu=1}^{n} \delta(\boldsymbol{y}-\boldsymbol{y}_\mu)\delta(\boldsymbol{x}-\boldsymbol{x}_\mu)/n$ and $\hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{x}) = \sum_{\mu=1}^{n} \delta(\boldsymbol{x}-\boldsymbol{x}_\mu)/n$. Indeed, the maximum conditional likelihood corresponds to the minimisation with respect to $\boldsymbol{W}$ of the conditional Kullback-Leibler divergence averaged over the empirical data distribution:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}\sim \hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{x})} \left[D_{\mathrm{KL}}(\hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{y}|\boldsymbol{x})||\mathrm{P}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{W}))\right] \\ &\quad= \mathbb{E}_{\boldsymbol{x}\sim \hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{x}),\boldsymbol{y}\sim \hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{y}|\boldsymbol{x})} \left[\log \hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{y}|\boldsymbol{x})\right] \\ &\quad -\mathbb{E}_{\boldsymbol{x}\sim \hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{x}),\boldsymbol{y}\sim \hat{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{y}|\boldsymbol{x})} \left[\log \mathrm{P}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{W})\right],\end{aligned} \tag{1.1.10}$$

that is non-negative everywhere and zero if and only if the two distributions coincide. Interestingly, in the regime where the number of data tends to infinity $n \to \infty$, largely exceeding the number of parameters, the MLE is optimal among all estimators. Indeed, under appropriate conditions, the MLE is *consistent* – i.e., converging to the true value – and *efficient*, since it saturates the Cramer-Rao bound (Rao, 1945; Cramer, 1946) meaning that no other consistent estimator has a lower MSE. However, where the number of data is limited, the MLE incurs in overfitting and regularised estimators are preferable, as we further discuss in Chapter 1.2.

**Mean-field methods —** As discussed above, the huge computational complexity involved in high-dimensional probabilistic modelling calls for alternative methods leading to efficient approximate computations. Celebrated and extensively used examples of such approximations are *mean-field methods*, which have a long history in the statistical physics literature, in particular regarding the study of spin glass models.

In a nutshell, mean-field methods are deterministic methods relying on tools such as Taylor expansions and convex relaxations in order to approximate marginals of the joint probability distributions of a large-scale system by exploiting the dependencies of its highly-coupled degrees of freedom. The structure of these dependencies is crucially clarified by a general framework that associates joint probability distributions with graphs, named *probabilistic graphical modeling*, where the random variables are encoded by the nodes of the graph and their interactions by the edges.

Whenever this graph of interactions is complete, the corresponding model is said to be *fully-connected* or *mean-field*.

The literature on mean-field methods is really vast and the interested reader can find more details in Mézard et al. (1987); Opper & Saad (2001); Mezard & Montanari (2009); Gabrié (2020); Montanari & Sen (2022) and references therein. For an historical overview on spin glass models, we refer the reader to the series of seven expository articles by Anderson (1988b)–Anderson (1990). In the following, we limit ourselves to a brief introduction of the *replica method*, one of the main mean-field methods, used to study analytically spin glass models.

**Self-averaging and the free entropy —** The partition function

$$Z_d(\boldsymbol{X}, \boldsymbol{y}; \beta) = \int \mathrm{d}\boldsymbol{W}\, \mathrm{e}^{-\beta \mathcal{H}(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y})}, \tag{1.1.11}$$

already introduced in Eq. (1.1.2), is a crucial quantity in statistical mechanics since it contains the relevant information on the equilibrium distribution of all the possible configurations of the states of the system. Indeed, $Z_d(\boldsymbol{X}, \boldsymbol{y}; \beta)$ is the *moment generating function* of the Gibbs distribution, since its derivatives with respect to the inverse temperature $\beta$ give rise to the moments of the distribution.

Since the partition function becomes exponentially peaked around the most probable configurations, *large deviation theory* is the suitable mathematical framework to formulate statistical mechanics as a probabilistic theory of high-dimensional correlated systems (Ellis, 2006; Touchette, 2009). We therefore take the logarithm of the partition function, defining the *free entropy* $\phi_d$ and the *free energy* $f_d$ densities:

$$\phi_d(\beta) = \frac{1}{d} \log Z_d(\boldsymbol{X}, \boldsymbol{y}; \beta), \qquad f_d(\beta) = -\frac{1}{\beta d} \log Z_d(\boldsymbol{X}, \boldsymbol{y}; \beta). \tag{1.1.12}$$

These two quantities are interchangeable and we report both definitions since information theory mostly deals with the free entropy while statistical physics usually refers to the free energy. The free energy and the free entropy densities enjoy the *self-averaging* property: in the thermodynamic limit $d \to \infty$, the law of large numbers kicks in and their probability measure concentrates exponentially around its expectation, or typical value

$$\phi_d(\beta) \overset{d\to\infty}{\longrightarrow} \phi(\beta), \qquad f_d(\beta) \overset{d\to\infty}{\longrightarrow} f(\beta). \tag{1.1.13}$$

In high dimensions, this key property allows to obtain general results that are independent from the specific realisation of the disorder. It is important to notice that the partition function itself is *not* self-averaging, and crucially the annealed and quenched averages of its logarithm do not coincide in general. In particular, due to Jensen inequality, the annealed average is an upper bound on the quenched one:

$$\mathbb{E}_{\boldsymbol{X},\boldsymbol{y}} \left[\log Z_d(\boldsymbol{X}, \boldsymbol{y}; \beta)\right] \leq \log \mathbb{E}_{\boldsymbol{X},\boldsymbol{y}} \left[Z_d(\boldsymbol{X}, \boldsymbol{y}; \beta)\right]. \tag{1.1.14}$$

The free entropy (energy) also encodes all the useful information on the system as it is the *cumulant generating function* of the Gibbs measure. If we also consider the *entropy* density

$$\mathcal{S}(e) = \lim_{d\to\infty} -\frac{1}{d} \log \mathrm{P}\left(\mathcal{H}(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{y}) = e\right), \tag{1.1.15}$$

i.e., the rate function of the Gibbs distribution, the Gartner-Ellis theorem (Gärtner, 1977; Ellis, 1984) states that $\mathcal{S}$ is obtained from the free entropy via a Legendre transformation

$$\mathcal{S}(e) = \max_{\beta} \left(\phi(\beta) + \beta\, e\right) \tag{1.1.16}$$

Therefore, the existence of the free entropy implies that the Gibbs measure satisfies a large deviation principle.

**The replica method —** The replica method (Mézard et al., 1987; Dotsenko, 2000) is a powerful tool to compute the average quenched free energy of mean-field models and lies at the heart of this first part of the thesis. The starting point of the computation relies on the *replica trick* – an "obvious identity", as remarked by Anderson (1988a), known by mathematicians at least since Hardy et al. (1952):

$$\begin{aligned} f(\beta) &= -\frac{1}{\beta} \lim_{d\to\infty} \lim_{p\to 0} \frac{\mathbb{E}_{\boldsymbol{X},\boldsymbol{y}}\left[Z_d(\boldsymbol{X},\boldsymbol{y};\beta)^p\right] - 1}{p\, d} \\ &= -\frac{1}{\beta} \lim_{d\to\infty} \lim_{p\to 0} \frac{\partial_p \mathbb{E}_{\boldsymbol{X},\boldsymbol{y}}\left[Z_d(\boldsymbol{X},\boldsymbol{y};\beta)^p\right]}{d}. \end{aligned} \tag{1.1.17}$$

The above expression is perfectly correct, however, at this point we have to proceed with some weird, but necessary, operations. First, we consider $p \in \mathbb{N}$, so that the computation reduces to the much easier average of an integer power of the partition function. This power can be seen as the product of $p$ non-interacting copies or *replicas* of the system $\{\boldsymbol{w}^a\}_{a=1}^p$. Here, for simplicity, we focus on the case where the degrees of freedom are encoded by a vector $\boldsymbol{w} \in \mathbb{R}^d$. The more general case where $\boldsymbol{W} \in \mathbb{R}^{d\times k}$ is a matrix, with $k \sim \mathcal{O}_d(1)$, is considered in Chapter 1.3. Note that, in learning and inference problems where a ground truth $\boldsymbol{w}^*$ is present, the average over the ground truth effectively acts as a $(p+1)^{\text{th}}$ replica (with a different prior distribution, apart from the BO case where the ground truth distribution and the prior match). The average over the disorder can be easily performed at this point, with the important effect of decoupling the degrees of freedom of a given system $\{w_j^a\}_{j=1}^d$, $\forall a = 1,\dots,p$, but coupling different replicas of the system through the order parameters:

$$Q_{ab} = \frac{\boldsymbol{w}^{a\top}\boldsymbol{w}^b}{d}, \qquad m_a = \frac{\boldsymbol{w}^{a\top}\boldsymbol{w}^*}{d}, \qquad \forall a,b = 1\dots,p, \tag{1.1.18}$$

also called *overlap* variables. At this point, the average replicated partition function reads

$$\mathbb{E}_{\boldsymbol{X},\boldsymbol{y}}\left[Z_d(\boldsymbol{X},\boldsymbol{y})^p\right] = \int \mathrm{d}\boldsymbol{Q}\, \mathrm{d}\boldsymbol{m}\; \mathrm{e}^{d\, S(\boldsymbol{Q},\boldsymbol{m})}, \qquad S(\boldsymbol{Q},\boldsymbol{m}) \sim \mathcal{O}_d(1), \tag{1.1.19}$$

thus, for large $d$, we can evaluate it at leading exponential order via a saddle-point method, i.e., extremising the action $S$ with respect to $\boldsymbol{Q}$ and $\boldsymbol{m}$. However, this step requires the exchange of the limits $p \to 0$ and $d \to \infty$, which may not commute. Therefore, in order to compute the limit

$$f(\beta) = \lim_{p\to 0} \frac{1}{p} \operatorname*{extr}_{\boldsymbol{Q},\boldsymbol{m}} S(\boldsymbol{Q},\boldsymbol{m}), \tag{1.1.20}$$

it is necessary to define the infinite dimensional limit of the overlaps in such a way that it is an analytic function in $p$. The first guess that comes to mind, motivated by the symmetry of the action with respect to the permutation of the replicas, is the *replica-symmetric* (RS) ansatz:

$$Q_{ab} := q \qquad \forall a \neq b, \qquad r := Q_{aa}, \quad m := m_a, \quad \forall a = 1, \ldots, p, \tag{1.1.21}$$

where the overlaps between every pair of replicas are statistically equivalent.

This procedure allows us to reduce the high-dimensional problem to the simpler optimisation of the action over a set of few scalar order parameters $q$, $r$, $m$. However, in many scenarios the RS ansatz is found to be unstable leading to unphysical results such as negative entropies (Gardner & Derrida, 1988) and more complex *replica symmetry breaking* ansatz are required. The general solution was found by Parisi in a series of works (Parisi, 1979, 1980, 1983), where he proposed a general scheme to iteratively break the RS in an infinite and continous hierarchy.

A great advantage of the replica method is that its validity extends well beyond quadratic loss functions, where also random matrix theory techniques can be used to study the solution space. Moreover, the method is not restricted to convex loss functions, at variance with currently available rigorous methods such as the Gordon minimax technique mentioned in Chapter 1.2. The generality of the replica method arguably compensates for the drawback of its heuristic nature and can pave the way for the investigation of more and more realistic models of ANNs. We do not dwell any further into the subtleties of the replica method and its applications, which have been already extensively reviewed in many venues (see, e.g., Mézard et al. (1987); Parisi et al. (2020)). For an historical account, see the *History of RSB Interview* (Charbonneau, 2021).

**Information-theoretical and algorithmic phase transitions** — Inspecting the behaviour of the free energy, we can detect information theoretical phase transitions and identify statistical thresholds. Indeed, the overlaps provide a deep interpretation of the structure of the solution space, capturing the typical distance between two solutions and between a typical solution and the ground truth.

Focusing on the Bayes-optimal estimator, we can qualitatively identify three phases:

- *Undetectable phase*: at very low sample complexity, the ground truth is indistinguishable from any other solution and even the optimal estimator is unable to correlate with it.

- *Weak-recovery phase*: above a threshold value of the sample complexity $\alpha \geq \alpha_{\text{weak}}$, the optimal estimator can only partially correlate with the ground truth.

- *Perfect-recovery* or *easy phase*: above the information-theoretical phase transition $\alpha \geq \alpha_{\text{IT}}$, the ground truth becomes a global minimum of the free energy and the optimal estimator can reconstruct it perfectly.

Note that the discussion above only deals with the optimal case, setting thresholds that no algorithm can improve. Indeed, according to the problem, there might be a gap between the information-theoretical easy phase and an algorithmic perfect reconstruction. This region $\alpha_{\mathrm{IT}} \geq \alpha \geq \alpha_{\mathrm{alg}}$ is called *hard phase*. These so-called *statistical-to-computational trade-offs* are currently broadly studied in high-dimensional statistics and inference. An example of this situation is treated in Chapter 1.3. We refer the reader to Percus et al. (2006); Zdeborová & Krzakala (2016); Ricci-Tersenghi et al. (2019) for more details on the topic.

**The teacher-student setting** — The *teacher-student scenario* provides a useful formulation of inference and learning problems: the teacher generates some observations by using some ground truth information and a probabilistic model to generate the data, the student has to recover the ground truth from the data and the observations handed by the teacher.

Starting with the seminal work of Gardner and Derrida (Gardner & Derrida, 1989) the *teacher-student perceptron* is a broadly adopted and studied model for high-dimensional supervised binary (i.e., two-classes) classification. In this model the input data are Gaussian i.i.d. and a single-layer teacher ANN with weights drawn i.i.d. from some distribution generates the labels. A student ANN then uses the input data and labels to *learn* the teacher function. The corresponding generalisation error as a function of the number of samples per dimension $\alpha = n/d$ was first derived using the replica method from statistical physics in the limit $n, d \to \infty$ for a range of teacher weight distributions (Gaussian and Rademacher being the most commonly considered) and for a range of estimators, e.g., BO or ERM with common losses, see the review articles by Seung et al. (1992b); Watkin et al. (1993); Engel & Van den Broeck (2001) and references therein.

Notably, the phase transition in the optimal generalisation error of the teacher-student perceptron with Rademacher teacher weights (Györgyi, 1990; Sompolinsky et al., 1990) is possibly one of the earliest examples of statistical-to-computational trade-off. More recently, these works on the teacher-student perceptron have been put on rigorous ground in Barbier et al. (2019) for the BO estimation, and in Aubin et al. (2020) for ERM with convex losses.

**Constraint satisfaction problems and the SAT-UNSAT transition** — A useful framework to study learning problems is that of *constraint satisfaction problems* (CSPs), such as the $k-$ SAT problem, $q-$ coloring, the traveling salesman problem, and sphere packing. See, e.g., Mezard & Montanari (2009) for an introduction and more examples.

CSPs are a special class of optimisation problems where the goal is to find a configuration of $d$ parameters $\boldsymbol{w}$ that satisfies a given set of constraints $\{C_\mu(\boldsymbol{w})\}_{\mu=1}^{n}$, $n = \alpha d$. The problem is said to be *satisfiable* (SAT) if there exists at least one solution respecting all the constraints and *unsatisfiable* (UNSAT) otherwise. A special case is that of *random* CSPs, where the set of constraints is drawn at random and plays the role of quenched disorder. This analogy has allowed the study of phase transitions in the solution space of CSPs controlled by the constraint density $\alpha$ using statistical physics methods. When the constraint density reaches a critical threshold $\alpha^*$, the space of solutions shrinks to zero and a so-called SAT-UNSAT transition occurs from satisfiability to non-satisfiability. The phenomenology of phase transitions in the solution space of CSPs is actually much richer, the interested reader is referred to Krzakała et al. (2007); Gabrié et al. (2017).

The perceptron problem also belongs to the class of random CSPs with continuous degrees of freedom (Franz et al., 2017), where each sample introduces a new constraint. A crucial point to keep in mind is that the link between constraint-satisfaction and learning problems is limited to the training phase: indeed, the empirical risk can be associated to a constraint-satisfaction landscape. However, the ultimate goal of learning is generalisation and in order to model this phenomenon it is crucial that the constraints involve some correlations between the data and the labels. This is not the case in general for CSPs, where the perceptron is often studied as a prototype random landscape with random labels uncorrelated from the data.

# 1.2 - The learning curves of binary Gaussian mixture classification

High-dimensional statistics where both the dimensionality and the number of samples are large displays highly non-intuitive behaviour. A number of the associated statistical surprises are for example presented in the recent, yet already rather influential papers (Hastie et al., 2022; Sur & Candès, 2019) that analyse high-dimensional *regression* for rather simple models of data. In this chapter, we focus instead on *classification* and we introduce one of the simplest models considered in statistics – the mixture of two Gaussian clusters – as a prototype for this task. We investigate the performance of different estimators and reveal some of the interesting behaviours associated to the high-dimensional regime. In particular, we discuss the surprising effect of the regularisation, that in some cases allows to reach the Bayes-optimal (BO) performance. The following results are based on Article 1. At the same time, this setting serves us as a working example to present the application of the replica method in the context of learning theory.

## 1.2.1 . Introduction to the task

We introduce here the binary Gaussian Mixture Model (GMM) that we also consider in Chapters 2.2 and 2.3. We consider two centroids localised at $\pm\boldsymbol{w}^*/\sqrt{d}$, with $\boldsymbol{w}^* \in \mathbb{R}^d$, and a synthetic training set

$$\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)^\top \in \mathbb{R}^{n\times d}, \quad \text{with binary labels } \; \boldsymbol{y} = (y_1, ..., y_n)^\top \in \{\pm 1\}^n, \tag{1.2.1}$$

where the samples $\boldsymbol{x}_\mu$ are generated as

$$\boldsymbol{x}_\mu = y_\mu \frac{\boldsymbol{w}^*}{\sqrt{d}} + \sqrt{\Delta}\, \boldsymbol{z}_\mu, \qquad \boldsymbol{z}_\mu \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d), \qquad \mu \in \{1, ...n\}\,. \tag{1.2.2}$$

The labels reflect the memberships in the clusters and are drawn i.i.d. with probability $\mathrm{P}(y = +1) = \rho$ and $\mathrm{P}(y = -1) = 1 - \rho$. Therefore, the clusters contain $\rho n$ and $(1 - \rho)n$ points respectively, on average. If $\rho = 0.5$, we say that the clusters are *balanced*, and *unbalanced* otherwise. We draw the centroid $\boldsymbol{w}^*$ either uniformly on the hypersphere of radius $\sqrt{d}$, $\boldsymbol{w}^* \in \mathcal{S}^{d-1}(\sqrt{d})$, or with i.i.d. standard Gaussian components $w^*_j \sim \mathcal{N}(0, 1)$, $\forall j = 1, \ldots, d$. Note that these two choices become equivalent in infinite dimensions. Moreover, due to the statistical isotropy of the samples, without loss of generality we can choose a basis where $\boldsymbol{w}^* = (1, 1, ...1) \in \mathbb{R}^d$.

In all what follows, we consider the high-dimensional setting where the dimension of each point in the dataset is $d \to \infty$ and the size of the training set $n = \alpha d$, at fixed sample complexity $\alpha \sim \mathcal{O}_d(1)$. Similarly, the noise level $\Delta > 0$ and the cluster size parameter $\rho$ are fixed of order one. The factor $\sqrt{d}$ in Eq. (1.2.2) ensures that a classification better than random is possible, yet even the oracle-classifier that knows exactly the centroid $\boldsymbol{w}^*$ only achieves a classification error bounded away from zero. Note that, if the noise level $\Delta$ or the fraction of samples $\alpha$ are small enough, the two Gaussian clouds are linearly separable by an hyperplane. Therefore, as explained in the *Motivation and background* chapter, a single-layer ANN (a perceptron) is enough to perform this task and we can focus on the simplest linear classification machine with output:

$$\hat{y}_\mu(\boldsymbol{w}) = \operatorname{sgn}(\boldsymbol{w}^\top \boldsymbol{x}_\mu/\sqrt{d} + \kappa), \tag{1.2.3}$$

where both the weights $\boldsymbol{w}$ and the bias $\kappa$ have to be learned. In other words, at sufficiently low $\alpha(\Delta, \rho)$, the perceptron classification problem defined above lies in the SAT phase (see Chapter 1.1). The constraints to be satisfied to achieve perfect classification are $y_\mu \left(\boldsymbol{w}^\top \boldsymbol{x}_\mu/\sqrt{d} + \kappa\right) \geq 0$, $\forall \mu = 1, \dots n$. Therefore, the loss is only a function of this product.

We are interested in studying the performance achieved by empirical risk minimisation (ERM) in comparison to Bayes-optimal (BO) estimation.

**Empirical risk minimisation —** We focus on ridge regularised learning performed by ERM, where the empirical risk is given by:

$$\mathcal{L}(\boldsymbol{w}, b){=}\sum_{\mu=1}^{n} \ell\left(y_\mu\left(\tfrac{1}{\sqrt{d}}\boldsymbol{x}_\mu^\top \boldsymbol{w} + \kappa\right)\right) + \frac{1}{2}\lambda\|\boldsymbol{w}\|_2^2, \tag{1.2.4}$$

$\boldsymbol{w}$ and $\kappa$ denote, respectively, the weight vector and the bias to be learned, and $\lambda$ is the tunable strength of the regularisation. While our result holds for any convex loss function $\ell(\cdot)$, we mainly concentrate on the following classic ones:

- the square loss: $\ell(v) = \frac{1}{2}(1 - v)^2$,
- the logistic loss: $\ell(v) = \log\left(1 + \mathrm{e}^{-v}\right)$,
- the hinge loss: $\ell(v) = \max_v\{0, 1 - v\}$,

that we display in Figure 1.2.1. Note that, for the binary GMM model of Eq. (1.2.2) under consideration, the unregularised ($\lambda = 0$) ERM estimator with the logistic loss corresponds to the maximum likelihood estimator (MLE) introduced in Chapter 1.1. This follows directly from the Bayes formula:

$$\log \mathrm{P}(y|\boldsymbol{x}) = \log \frac{\mathrm{P}(\boldsymbol{x}|y)\mathrm{P}(y)}{\sum_{y=\pm 1}\mathrm{P}(\boldsymbol{x}|y)\mathrm{P}(y)} = \log\left(1 + \mathrm{e}^{-c}\right), \tag{1.2.5}$$

where $c = \frac{2}{\Delta}\left(\frac{1}{\sqrt{d}}\boldsymbol{x}^\top\boldsymbol{w}^* + \frac{\Delta}{2}\log\frac{\rho}{1-\rho}\right)$, therefore a simple redefinition of the variables leads to the logistic loss function than turns out to correspond to the MLE (or rather the MAP estimator if one allows the learning of a bias to account for the possibly different cluster sizes).

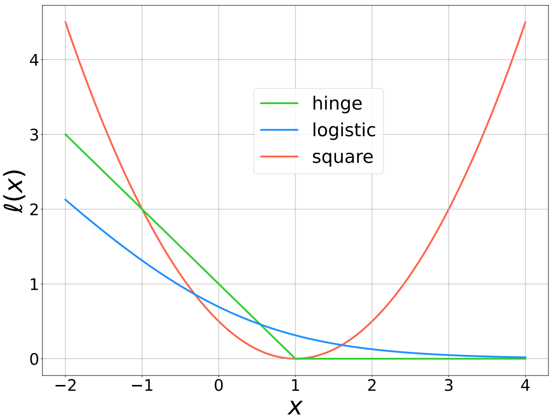


Figure 1.2.1 – Loss functions under consideration.

**Related works** — The unsupervised GMM is a standard problem in statistics (Friedman et al., 2001). The supervised version of the model has generated a recent surge of interest as a prototype for classification tasks. Lelarge & Miolane (2019) have considered the case of balanced clusters and have computed rigorously the BO estimator. We extend this result to arbitrary cluster size $\rho$, which serves us as a baseline for the performance of the estimators obtained via ERM.

The performance of ERM has been studied by Mai & Liao (2019) again for the balanced case and at zero regularisation, under the assumption that the data are not linearly separable, i.e., the problem lies in the UNSAT region of parameter space. The authors have concluded that, in this specific setting, the square loss is a universally-optimal loss function. Our study of regularised losses in the generic unbalanced setting shows that the performance of non-regularised square loss can be drastically improved. The linear separability condition has been studied in the balanced case at zero regularisation by Deng et al. for the logistic loss and by Kini & Thrampoulidis (2020) for the square loss. The effect of data structure on learning a linearly separable rule has been studied already by Marangi et al. (1995) in a model of two clusters of binary input data labeled by a perceptron teacher. In the following, we derive the separability condition as a function of all the problem parameters including arbitrary cluster size.

Our main contribution is the derivation of rigorous asymptotic closed-form expressions for the generalisation and training error in the noisy high-dimensional regime, for any convex loss function, including the effect of regularisation, and for arbitrary cluster size. The proof is based on Gordon's inequality technique (Gordon, 1985; Thrampoulidis et al., 2015). The same formulas are obtained from the heuristic replica theory of statistical physics. Indeed, closely-related models have been studied in the literature with this method (Del Giudice et al., 1989; Franz et al., 1990). We show through numerical simulations that the formulas are extremely accurate even at moderately small dimensions.

Armed with the exact solution, we proceed with a systematic investigation of the effects of regularisation and cluster size. In particular, we discuss how far ERM estimators fall short of the BO one, with surprising conclusions where we illustrate the effect of weak and strong regularisation. In the SAT region, where data are linearly separable, Rosset et al. (2004) prove that all monotone non-increasing loss functions that depend on the margin find a solution maximising the margin. This is indeed exemplified in our model by the fact that below the linear separability transition ($\alpha < \alpha^*(\Delta, \rho)$) the hinge and logistic losses converge to the same test error as the regularisation vanishes. This is related to the implicit regularisation of gradient descent (GD) for the non-regularised minimisation (Soudry et al., 2018b).

The existence of a sharp transition for perfect separability in the model, with and without bias, is interesting in itself. Recently, Sur & Candès (2019) analysed the MLE in high-dimensional logistic regression. While they considered Gaussian data (whereas we study a Gaussian mixture) their results on the existence of the MLE being related to the separability of the data and displaying a sharp phase transition are of the same nature as ours, and similar to earlier works in statistical physics Gardner (1988); Gardner & Derrida (1989); Krauth & Mézard (1989).

All these results show that the binary GMM studied here allows to discuss, illustrate and clarify in a unified fashion many phenomena that are currently the subject of intense scrutiny in high-dimensional statistics and ML.

## 1.2.2 . Generalisation error and Bayes-optimal performance

In this section we show how to derive an asymptotic expression for the generalisation error in high dimensions. As customary in classification tasks, the generalisation error is defined as the average fraction of mislabeled instances

$$\varepsilon_{\text{gen}} = \mathbb{E}_{y_{\text{new}},\boldsymbol{x}_{\text{new}},\boldsymbol{X},\boldsymbol{y}} \left[\mathbb{1}\left(\hat{y}_{\text{new}} \neq y_{\text{new}}\right)\right] = \frac{1}{4}\mathbb{E}_{y_{\text{new}},\boldsymbol{x}_{\text{new}},\boldsymbol{X},\boldsymbol{y}} \left[\left(y_{\text{new}} - \hat{y}_{\text{new}}\right)^2\right], \qquad (1.2.6)$$

where $y_{\text{new}}$ is the label of a new observation $\boldsymbol{x}_{\text{new}}$, and the estimator $\hat{y}_{\text{new}}$ is computed as

$$\hat{y}_{\text{new}} = \text{sign}\left(\frac{\boldsymbol{w}^\top \boldsymbol{x}_{\text{new}}}{\sqrt{d}} + \kappa\right). \qquad (1.2.7)$$

Eq. (1.2.7) holds for every vector $\boldsymbol{w} = \boldsymbol{w}\left(\boldsymbol{X}, \boldsymbol{y}\right)$ and bias $\kappa = \kappa\left(\boldsymbol{X}, \boldsymbol{y}\right)$ computed on the training set $\{\boldsymbol{X}, \boldsymbol{y}\}$. Using that $y_{\text{new}}, \hat{y}_{\text{new}} = \pm 1$, Eq. (1.2.6) can be rewritten as

$$\varepsilon_{\text{gen}} = \frac{1}{2}\left(1 - \mathbb{E}_{y_{\text{new}},\boldsymbol{x}_{\text{new}},\boldsymbol{X},\boldsymbol{y}} \left[\text{sign}\left(\frac{\boldsymbol{w}^\top y_{\text{new}} \boldsymbol{x}_{\text{new}}}{\sqrt{d}} + y_{\text{new}}\, \kappa\right)\right]\right). \qquad (1.2.8)$$

The term $y_{\text{new}}\, \boldsymbol{x}_{\text{new}}$ can be rewritten as

$$y_{\text{new}}\, \boldsymbol{x}_{\text{new}} = y_{\text{new}} \left(y_{\text{new}} \frac{\boldsymbol{w}^*}{\sqrt{d}} + \sqrt{\Delta} \boldsymbol{z}_{\text{new}}\right) = \frac{\boldsymbol{w}^*}{\sqrt{d}} + \sqrt{\Delta}\, \boldsymbol{z}'_{\text{new}}, \qquad (1.2.9)$$

where $\boldsymbol{z}'_{\text{new}} = y_{\text{new}}\boldsymbol{z}_{\text{new}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ is distributed as $\boldsymbol{z}_{\text{new}}$, since $y_{\text{new}}$ and $\boldsymbol{z}_{\text{new}}$ are independent. Therefore, we find

$$\begin{aligned}&\mathbb{E}_{y_{\text{new}},\boldsymbol{x}_{\text{new}},\boldsymbol{X},\boldsymbol{y}}\left[\operatorname{sign}\left(\frac{\boldsymbol{w}^\top y_{\text{new}}\,\boldsymbol{x}_{\text{new}}}{\sqrt{d}} + y_{\text{new}}\,\kappa\right)\right]\\ &= \mathbb{E}_{y_{\text{new}},\boldsymbol{z}'_{\text{new}},\boldsymbol{w}^*,\boldsymbol{X},\boldsymbol{y}}\left[\operatorname{sign}\left(\frac{\boldsymbol{w}^\top\boldsymbol{w}^*}{d} + \sqrt{\frac{\Delta}{d}}\boldsymbol{w}^\top\boldsymbol{z}'_{\text{new}} + y_{\text{new}}\,\kappa\right)\right].\end{aligned} \tag{1.2.10}$$

The estimator $\boldsymbol{w}$ only depends on the training set, hence $\boldsymbol{w}$ and $\boldsymbol{z}'_{\text{new}}$ are independent. We call their rescaled scalar product $\varsigma$, a random variable distributed as a standard Gaussian

$$\varsigma = \frac{1}{||\boldsymbol{w}||}\boldsymbol{w}^\top\boldsymbol{z}'_{\text{new}} \sim \mathcal{N}(0,1)\,. \tag{1.2.11}$$

By averaging over $\varsigma$, we obtain

$$\begin{aligned}&\mathbb{E}_{y_{\text{new}},\boldsymbol{w}^*,\boldsymbol{X},\boldsymbol{y},\varsigma}\left[\operatorname{sign}\left(\frac{\boldsymbol{w}^\top\boldsymbol{w}^*}{d} + \sqrt{\frac{\Delta}{d}}\|\boldsymbol{w}\|\varsigma + y_{\text{new}}\,\kappa\right)\right]\\ &= \mathbb{E}_{y_{\text{new}},\boldsymbol{w}^*,\boldsymbol{X},\boldsymbol{y},\varsigma}\left[\operatorname{sign}\left(\frac{1}{\sqrt{\Delta}}\frac{\boldsymbol{w}^\top}{||\boldsymbol{w}||}\frac{\boldsymbol{w}^*}{\sqrt{d}} + \varsigma + y_{\text{new}}\,\kappa\frac{\sqrt{d}}{\sqrt{\Delta}||\boldsymbol{w}||}\right)\right],\end{aligned} \tag{1.2.12}$$

where we used that $\sqrt{\frac{\Delta}{d}}\|\boldsymbol{w}\| > 0$ to rescale the argument of the sign function. Finally, we obtain

$$\begin{aligned}\varepsilon_{\text{gen}} &= \frac{1}{2}\left(1 - \mathbb{E}_{y_{\text{new}},\boldsymbol{w}^*,\boldsymbol{X},\boldsymbol{y}}\left[\mathbb{P}\left(\varsigma > -\tau\right) - \mathbb{P}\left(\varsigma < -\tau\right)\right]\right)\\ &= \mathbb{E}_{y_{\text{new}},\boldsymbol{w}^*,\boldsymbol{X},\boldsymbol{y}}\left[Q(\tau)\right].\end{aligned} \tag{1.2.13}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}\mathrm{d}t = \frac{1}{2}\operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right)$ is the Gaussian tail function, and we have defined

$$\tau = \frac{\sqrt{d}}{\sqrt{\Delta}||\boldsymbol{w}||}\left(\frac{\boldsymbol{w}^\top\boldsymbol{w}^*}{d} + y_{\text{new}}\,\kappa\right). \tag{1.2.14}$$

Due to concentration of measure, the infinite dimensional limits $m, q$ of the *overlaps*

$$m_d := \frac{\boldsymbol{w}^\top\boldsymbol{w}^*}{d} \overset{d\to\infty}{\longrightarrow} m, \tag{1.2.15}$$

$$q_d := \frac{||\boldsymbol{w}||^2}{d} \overset{d\to\infty}{\longrightarrow} q, \tag{1.2.16}$$

are deterministic. Hence the generalisation error reads

$$\varepsilon_{\text{gen}} = \rho\, Q\Big(\frac{m+\kappa}{\sqrt{\Delta q}}\Big) + (1-\rho)Q\Big(\frac{m-\kappa}{\sqrt{\Delta q}}\Big), \tag{1.2.17}$$

where we remind that $\rho \in (0,1)$ is the probability that $y_{\text{new}} = +1$.

**Bayes-optimal performance** — The BO estimator has access to the $n$ training samples $\{(y_\mu, \boldsymbol{x}_\mu)\}_{\mu=1}^n$ and to the generative model of the data, including the constants $\rho$ and $\Delta$. Crucially, it does not have access to the position of the centroid $\boldsymbol{w}^*$, that can only be estimated from the data. In order to compute the BO error, we consider the distribution of a new data point $\boldsymbol{x}_{\text{new}}$ and the corresponding new label $y_{\text{new}}$, given the estimate $\boldsymbol{w}$ of the true centroid $\boldsymbol{w}^*$, that is given by Bayes formula:

$$\begin{aligned} \mathrm{P}\left(\boldsymbol{x}_{\text{new}}, y_{\text{new}} | \boldsymbol{w}\right) &\propto \mathrm{P}\left(\boldsymbol{x}_{\text{new}} | y_{\text{new}}, \boldsymbol{w}\right) \mathrm{P}_y(y_{\text{new}}) \\ &\propto \exp\left(-\frac{1}{2\Delta}\sum_{i=1}^d \left(x^i_{\text{new}} - \frac{y_{\text{new}} w^i}{\sqrt{d}}\right)^2\right) \mathrm{P}_y(y_{\text{new}}), \end{aligned} \tag{1.2.18}$$

where the symbol "$\propto$" takes into account the normalisation constant. Similarly, the posterior on $\boldsymbol{w}$ given the training set is

$$\begin{aligned} \mathrm{P}\left(\boldsymbol{w} | \boldsymbol{X}, \boldsymbol{y}\right) \propto \mathrm{P}\left(\boldsymbol{X} | \boldsymbol{w}, \boldsymbol{y}\right) \mathrm{P}_{\boldsymbol{w}^*}\left(\boldsymbol{w}\right) \\ \propto \left[\prod_{\mu=1}^n \exp\left(-\frac{1}{2\Delta}\sum_{i=1}^d \left(x^i_\mu - \frac{y_\mu w^i}{\sqrt{d}}\right)^2\right)\right] \exp\left(-\frac{1}{2}\sum_{i=1}^d (w^i)^2\right), \end{aligned} \tag{1.2.19}$$

where we have used the fact that $\boldsymbol{w}^*$ has i.i.d. standard Gaussian components. We would like to find an explicit expression for

$$\mathrm{P}\left(y_{\text{new}} | \boldsymbol{x}_{\text{new}}, \boldsymbol{X}, \boldsymbol{y}\right) \propto \mathbb{E}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}}\left[\mathrm{P}\left(y_{\text{new}}, \boldsymbol{x}_{\text{new}} | \boldsymbol{w}\right)\right], \tag{1.2.20}$$

in order to estimate the new label as

$$\hat{y}_{\text{new}} = \underset{y'=\pm 1}{\operatorname{argmax}} \log \mathrm{P}\left(y' | \boldsymbol{x}_{\text{new}}, \boldsymbol{X}, \boldsymbol{y}\right). \tag{1.2.21}$$

Therefore, we have to compute

$$\begin{aligned} \mathbb{E}_{\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}}\left[\mathrm{P}\left(y_{\text{new}}, \boldsymbol{x}_{\text{new}} | \boldsymbol{w}\right)\right] \propto \\ \mathrm{P}_y\left(y_{\text{new}}\right) \int \left(\prod_{i=1}^d \mathrm{d}w^i \;\; \mathrm{e}^{-\frac{1}{2}(w^i)^2}\right) \prod_{\mu=0}^n \mathrm{e}^{-\frac{1}{2\Delta}\sum_{i=1}^d \left(x^i_\mu - \frac{y_\mu w^i}{\sqrt{d}}\right)^2}, \end{aligned} \tag{1.2.22}$$

where in the product over $\mu$ on the right-hand side we have used the notation $y_0 = y_{\text{new}}$, $\boldsymbol{x}_0 = \boldsymbol{x}_{\text{new}}$. Let us call $I_w$ the integral over $\boldsymbol{w}$ in Eq. (1.2.22):

$$I_w = \prod_{i=1}^d \int \mathrm{d}w^i \, \mathrm{e}^{-\frac{1}{2\Delta}\sum_{\mu=0}^n \left(x^i_\mu - \frac{y_\mu w^i}{\sqrt{d}}\right)^2 - \frac{1}{2}(w^i)^2}. \tag{1.2.23}$$

Computing the integral over $w^i$, we obtain

$$\begin{aligned} I_w &= C\left(\alpha, \Delta, d\right) \prod_{i=1}^d \prod_{\mu=0}^n \mathrm{e}^{-\frac{1}{2\Delta\left(\alpha+\Delta+\frac{1}{d}\right)}\left((\alpha+\Delta)(x^i_\mu)^2 - \frac{\alpha}{n} x^i_\mu y_\mu \sum_{\nu=0}^n x^i_\nu y_\nu\right)} \\ &= C\left(\alpha, \Delta, d\right) \mathrm{e}^{-\frac{1}{2\Delta\left(\alpha+\Delta+\frac{1}{d}\right)}\sum_{i=1}^d\left((\alpha+\Delta)(x^i_{\text{new}})^2 - \frac{\alpha}{n} y_{\text{new}} x^i_{\text{new}} \sum_{\nu=1}^n x^i_\nu y_\nu - \frac{\alpha}{n}(x^i_{\text{new}})^2\right)} \\ &\quad \times \mathrm{e}^{-\frac{1}{2\Delta\left(\alpha+\Delta+\frac{1}{d}\right)}\sum_{\mu=1}^n\sum_{i=1}^d\left((\alpha+\Delta)(x^i_\mu)^2 - \frac{\alpha}{n} y_\mu x^i_\mu \sum_{\nu=1}^n x^i_\nu y_\nu - \frac{\alpha}{n} y_\mu x^i_\mu y_{\text{new}} x^i_{\text{new}}\right)} \\ &= C\left(\alpha, \Delta, d\right) \tilde{C}\left(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x}_{\text{new}}, \alpha, \Delta, d\right) \mathrm{e}^{\frac{\alpha}{\Delta\left(\alpha+\Delta+\frac{1}{d}\right)} y_{\text{new}} \boldsymbol{x}_{\text{new}}^\top \frac{1}{n}\sum_{\mu=1}^n y_\mu \boldsymbol{x}_\mu}, \end{aligned} \tag{1.2.24}$$

where the first two factors $C$ and $\tilde{C}$ contain all the terms that do not depend on $y_{\text{new}}$. Therefore, we find

$$\hat{y}_{\text{new}} = \underset{y=\pm 1}{\text{argmax}} \left[ \frac{\alpha}{\Delta\left(\alpha+\Delta+\frac{1}{d}\right)} y \boldsymbol{x}_{\text{new}}^{\top} \frac{1}{n} \sum_{\mu=1}^{n} y_{\mu} \boldsymbol{x}_{\mu} + \log \mathrm{P}_y\left(y\right) \right]. \tag{1.2.25}$$

Using the fact that $y_{\mu}\boldsymbol{x}_{\mu} = \frac{\boldsymbol{w}^{*}}{\sqrt{d}} + \sqrt{\Delta}\boldsymbol{z}_{\mu}$, $\boldsymbol{z}_{\mu} \sim \mathcal{N}(0, \boldsymbol{I}_d)$ and $\boldsymbol{w}^{*}$ is the true realisation of $\boldsymbol{w}$, the first term in Eq. (1.2.25) in the limit where $n, d \to \infty$ can be rewritten as

$$\frac{1}{n} \sum_{\mu=1}^{n} \boldsymbol{x}_{\text{new}}^{\top} y_{\mu} \boldsymbol{x}_{\mu} \underset{n,d\to\infty}{\longrightarrow} y_{\text{new}} + \sqrt{\Delta\left(1+\frac{\Delta}{\alpha}\right)} z'_{\text{new}}, \tag{1.2.26}$$

where $z'_{\text{new}} \sim \mathcal{N}(0,1)$. Therefore, in the large $d$ limit we find that

$$\hat{y}_{\text{new}} = \underset{y=\pm 1}{\text{argmax}} \left[ \frac{\alpha}{\Delta\left(\alpha+\Delta\right)} y \left( y_{\text{new}} + \sqrt{\Delta\left(1+\frac{\Delta}{\alpha}\right)} z'_{\text{new}} \right) + \log \mathrm{P}_y\left(y\right) \right]. \tag{1.2.27}$$

It is useful to rewrite the generalisation error as

$$\varepsilon_{\text{gen}} = \frac{1}{4} \mathbb{E}_{\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_{\text{new}},y_{\text{new}}} \left[ \left(\hat{y}_{\text{new}} - y_{\text{new}}\right)^2 \right] = \sum_{y_{\text{new}}=-1,1} \mathbb{P}\left(\hat{y}_{\text{new}} \neq y_{\text{new}}\right) \mathrm{P}_y(y_{\text{new}}). \tag{1.2.28}$$

Using Eq. (1.2.27), we can compute

$$\begin{aligned} &\mathbb{P}\left(\hat{y}_{\text{new}} \neq y_{\text{new}}\right) \\ &= \mathbb{P}\left( y_{\text{new}} z'_{\text{new}} < -\sqrt{\frac{\alpha}{\Delta(\alpha+\Delta)}} \left( 1 + \left(1+\frac{\Delta}{\alpha}\right) \frac{\Delta}{2} \log \frac{\mathrm{P}_y(y_{\text{new}})}{\mathrm{P}_y(-y_{\text{new}})} \right) \right). \end{aligned} \tag{1.2.29}$$

If $y_{\text{new}} = 1$, Eq. (1.2.29) gives

$$\mathbb{P}\left(\hat{y}_{\text{new}} \neq 1\right) = Q\left( \frac{\frac{\alpha}{\Delta+\alpha} + \frac{\Delta}{2}\log\frac{\rho}{1-\rho}}{\sqrt{\Delta\frac{\alpha}{\Delta+\alpha}}} \right), \tag{1.2.30}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2} \mathrm{d}t = \frac{1}{2}\mathrm{erfc}\left(\frac{x}{\sqrt{2}}\right)$ is the Gaussian tail function. If $y_{\text{new}} = -1$, Eq. (1.2.29) gives

$$\mathbb{P}\left(\hat{y}_{\text{new}} \neq -1\right) = Q\left( \frac{\frac{\alpha}{\Delta+\alpha} - \frac{\Delta}{2}\log\frac{\rho}{1-\rho}}{\sqrt{\Delta\frac{\alpha}{\Delta+\alpha}}} \right). \tag{1.2.31}$$

Using the fact that $\rho = \mathrm{P}_y(1)$ and $1-\rho = \mathrm{P}_y(-1)$, we obtain

$$\varepsilon_{\text{gen}}^{\text{BO}} = \rho Q\left( \frac{\frac{\alpha}{\Delta+\alpha} + \frac{\Delta}{2}\log\frac{\rho}{1-\rho}}{\sqrt{\Delta\frac{\alpha}{\Delta+\alpha}}} \right) + (1-\rho) Q\left( \frac{\frac{\alpha}{\Delta+\alpha} - \frac{\Delta}{2}\log\frac{\rho}{1-\rho}}{\sqrt{\Delta\frac{\alpha}{\Delta+\alpha}}} \right), \tag{1.2.32}$$

which gives the BO error as a function of the cluster unbalance $\rho$, the noise variance $\Delta$, and the sample complexity $\alpha$.

**The Hebb estimator** — In the setting under consideration, the BO performance turns out to be realised efficiently by the following simple estimator, akin to applying the Hebb's rule (Hebb, 1949):

$$\hat{\boldsymbol{w}}^{\text{Hebb}} = \frac{1}{\alpha} \sum_{\mu=1}^{n} y_\mu \frac{\boldsymbol{x}_\mu}{\sqrt{d}}, \tag{1.2.33}$$

when plugged into the estimation rule in Eq. (1.2.3), as known for the case of the teacher-student perceptron (Engel & Van den Broeck, 2001). This result has already been shown in Lelarge & Miolane (2019) for the case of balanced clusters. Note that, in the case of balanced clusters, the Hebb estimator is unbiased by definition, since the noise has zero mean.

In the more interesting case of non-balanced mixture of Gaussians, one further needs to optimise the intercept $\kappa$ in the linear fit. Since the minimiser of the generalisation error with respect to the bias is unique, this parameter can be optimised in a number of ways, including gradient descent or cross validation. The optimal bias $\hat{\kappa}$ is obtained from the minimisation of the generalisation error in Eq. (1.2.17) with respect to $\kappa$, at fixed $m, q$:

$$\hat{\kappa} = \underset{\kappa}{\text{argmin}} \;\; \varepsilon_{\text{gen}}(q, m, \kappa) = \frac{q}{m} \frac{\Delta}{2} \log\left(\frac{\rho}{1-\rho}\right). \tag{1.2.34}$$

Substituting Eq. (1.2.33) in the definition of $q, m$ in Eq. (1.2.16), we obtain that the values of $m$ and $q$ associated to the plugin estimator are

$$m = 1, \qquad q = \left(1 + \tfrac{\Delta}{\alpha}\right). \tag{1.2.35}$$

Hence, the generalisation error of the plug-in estimator is

$$\begin{aligned} \varepsilon_{\text{gen}}^{\text{plugin}} &= \mathbb{P}\left(y_{\text{new}}\left(\frac{1}{\sqrt{d}}\boldsymbol{x}_{\text{new}}^\top \hat{\boldsymbol{w}}^{\text{Hebb}} + \hat{\kappa}\right) < 0\right) \\ = \mathbb{P}&\left(y_{\text{new}} z'_{\text{new}} < -\sqrt{\frac{\alpha}{\Delta(\alpha+\Delta)}}\left(1 + y_{\text{new}}\left(1 + \frac{\Delta}{\alpha}\right)\frac{\Delta}{2}\log\frac{\rho}{1-\rho}\right)\right), \end{aligned} \tag{1.2.36}$$

where we have used Eq. (1.2.26) in the last equality. The probability in Eq. (1.2.36) is the same as in Eq. (1.2.29). Thus, the plug-in estimator achieves the BO error. Since there exists a plug-in estimator that reaches the BO performance, it is particularly interesting to see how the ones obtained by ERM compare with the optimal result.

## 1.2.3 . The learning curves of empirical risk minimisation via the replica method

In this section, we illustrate how to derive the learning curves[1] for the binary GMM of classification via the replica method introduced in Chapter 1.1. This derivation is based on Franz et al. (2017); Urbani (2018). We refer to Article 1 for the rigorous derivation obtained via Gordon's inequality techniques.

[1] An unpublished variant of this computation has been derived in September 2019 together with Federica Gerace and Bruno Loureiro.

**The Gibbs measure —** We recast the ERM framework in the statistical physics language, as previously explained in Chapter 1.1, by writing the Gibbs distribution

$$\mathrm{P}_\beta\left(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}\right) = \frac{1}{Z_d(\boldsymbol{X},\boldsymbol{y})}\prod_{\mu=1}^{n}\exp\left(-\beta\ell\left(y_\mu\frac{\boldsymbol{w}^\top\boldsymbol{x}_\mu}{\sqrt{d}}\right)+\kappa\right)\prod_{i=1}^{d}\exp\left(-\frac{\beta\lambda}{2}w_i^2\right) \tag{1.2.37}$$

and recalling that ERM is recovered in the ground-state limit $\beta\to\infty$ ($T=1/\beta\to 0$).

**Average free energy density —** To characterise the typical performance of the algorithm, we compute the average free energy density $f_\beta$ in the infinite dimensional limit:

$$f_\beta = -\frac{1}{\beta}\lim_{d\to\infty}\frac{\langle\log Z_d(\boldsymbol{X},\boldsymbol{y})\rangle_\beta}{d}, \tag{1.2.38}$$

where from now on we use the square brackets to indicate the average over the disorder (the dataset) $\langle\cdot\rangle_\beta = \mathbb{E}_{\boldsymbol{X},\boldsymbol{y}}\left[\cdot\right]$ performed at inverse temperature $\beta$. In order to compute it, we start from the replica trick:

$$f_\beta = -\frac{1}{\beta}\lim_{d\to\infty}\lim_{p\to 0}\frac{\partial_p\langle Z_d(\boldsymbol{X},\boldsymbol{y})^p\rangle_\beta}{d}. \tag{1.2.39}$$

The $p^{\text{th}}$ moment of the partition function is

$$\langle Z_d\left(\boldsymbol{X},\boldsymbol{y}\right)^p\rangle_\beta = \left\langle\int\left(\prod_{a=1}^{p}\prod_{j=1}^{d}\mathrm{d}w_j^a\right)\mathrm{e}^{-\beta\sum_{a=1}^{p}\left[\sum_{\mu=1}^{n}\ell\left(y_\mu\frac{\boldsymbol{w}^{a\top}\boldsymbol{x}_\mu}{\sqrt{d}}+\kappa\right)+\frac{\lambda}{2}\|\boldsymbol{w}^a\|^2\right]}\right\rangle_\beta, \tag{1.2.40}$$

where we have replaced the $p^{\text{th}}-$power with a product over $p$ *replicas* of the system, indicised by $a=1,\dots p$. It is useful to define the auxiliary variables

$$r_\mu^a = \boldsymbol{w}^{a\top}\boldsymbol{x}_\mu/\sqrt{d}, \qquad \forall\mu=1,\dots n, \tag{1.2.41}$$

by means of the Fourier representation of the Dirac $\delta$-function: $\delta(x)=\int_{-\infty}^{+\infty}\frac{\mathrm{d}z}{2\pi}e^{ixz}$. We obtain

$$\begin{aligned}\langle Z_d\left(\boldsymbol{X},\boldsymbol{y}\right)^p\rangle_\beta = &\left\langle\int\left(\prod_{a,j}\mathrm{d}w_j^a\right)\left(\prod_{a,\mu}\frac{\mathrm{d}r_\mu^a\mathrm{d}\hat{r}_\mu^a}{2\pi}\right)\right.\\ &\left.\mathrm{e}^{-\beta\sum_{a,\mu}\ell\left(y_\mu r_\mu^a+\kappa\right)-\frac{\beta\lambda}{2}\sum_{a,j}(w_j^a)^2}\times\exp\left(\mathrm{i}\sum_{\mu,a}\hat{r}_\mu^a\left[r_\mu^a-\frac{\boldsymbol{w}^{a\top}\boldsymbol{x}_\mu}{\sqrt{d}}\right]\right)\right\rangle_\beta.\end{aligned} \tag{1.2.42}$$

We can now average over the Gaussian vectors $\boldsymbol{z}_\mu$, $\mu=1,...n$, and find

$$\begin{aligned}\langle Z_d\left(\boldsymbol{X},\boldsymbol{y}\right)^p\rangle_\beta \propto &\left\langle\int\left(\prod_{a,j}\mathrm{d}w_j^a\right)\left(\prod_{a,\mu}\frac{\mathrm{d}r_\mu^a\mathrm{d}\hat{r}_\mu^a}{2\pi}\right)\mathrm{e}^{\sum_{a,\mu}\left[\mathrm{i}\hat{r}_\mu^a r_\mu^a-\beta\ell\left(y_\mu r_\mu^a+\kappa\right)\right]-\frac{\beta\lambda}{2}\sum_{a,j}(w_j^a)^2}\right.\\ &\left.\times\prod_{\mu=1}^{\alpha d}\exp\left(-\frac{\Delta}{2}\sum_{a,b=1}^{n}\hat{r}_\mu^a\hat{r}_\mu^b\frac{\boldsymbol{w}^{a\top}\boldsymbol{w}^b}{d}-\sum_{a=1}^{n}\mathrm{i}y_\mu\hat{r}_\mu^a\frac{\boldsymbol{w}^{a\top}\boldsymbol{w}^*}{d}\right)\right\rangle_\beta,\end{aligned} \tag{1.2.43}$$

where now the brackets $\langle\cdot\rangle_\beta$ indicate the average over the labels $y_\mu$ and the ground truth $\boldsymbol{w}^*$, and we have used the symbol "$\propto$" to indicate that we are neglecting multiplicative constant terms. Notice that Eq. (1.2.43) depends on the weights only through the two order parameters:

$$Q_{ab} = \frac{\boldsymbol{w}^{a\top}\boldsymbol{w}^b}{d} \qquad \forall a \le b, \tag{1.2.44}$$

$$m_a = \frac{\boldsymbol{w}^{a\top}\boldsymbol{w}^*}{d} \qquad \forall a, \tag{1.2.45}$$

that we have introduced in Chapter 1.1 and that naturally appear from the average over the disorder, decoupling the degrees of freedom in each system but coupling different replicas. We refer to $Q_{ab}$ (Eq. (1.2.44)) as *self-overlap* and to $m_a$ (Eq. (1.2.45)) as *magnetisation*. Therefore, we can transform the integral over $\boldsymbol{w}$ into an integral over $\boldsymbol{Q}$, $\boldsymbol{m}$ via the Jacobian:

$$\begin{aligned} J\left(\boldsymbol{Q},\boldsymbol{m}\right) &= d^{\frac{p(p+1)}{2}} \int \left(\prod_{a,j} \mathrm{d}w_j^a \, \mathrm{e}^{-\frac{\beta\lambda}{2}(w_j^a)^2}\right) \\ \times \prod_{a\le b} \delta\left(dQ_{ab} - \boldsymbol{w}^{a\top}\boldsymbol{w}^b\right) &\prod_a \delta\left(dm_a - \boldsymbol{w}^{a\top}\boldsymbol{w}^*\right) \\ &= d^{\frac{p(p+1)}{2}} \int \left(\prod_{a,j} \mathrm{d}w_j^a \, \mathrm{e}^{-\frac{\beta\lambda}{2}(w_j^a)^2}\right) \\ \times \mathrm{e}^{\sum_{a\le b} \mathrm{i}\hat{Q}_{ab}\left(dQ_{ab} - \boldsymbol{w}^{a\top}\boldsymbol{w}^b\right) + \sum_a \mathrm{i}\hat{m}_a\left(dm_a - \boldsymbol{w}^{a\top}\boldsymbol{w}^*\right)}&. \end{aligned} \tag{1.2.46}$$

We can now compute the Gaussian integral over the weights $\boldsymbol{w}$, obtaining

$$\begin{aligned} &\langle Z_d\left(\boldsymbol{X},\boldsymbol{y}\right)^p\rangle_\beta \propto \\ &\left\langle \int \left(\prod_{a,\mu} \frac{\mathrm{d}r_\mu^a \mathrm{d}\hat{r}_\mu^a}{2\pi}\right) \left(\prod_{a\le b} \frac{\mathrm{d}Q_{ab}\mathrm{d}\hat{Q}_{ab}}{2\pi}\right) \left(\prod_a \frac{\mathrm{d}m_a \mathrm{d}\hat{m}_a}{2\pi}\right) \mathrm{e}^{\sum_{a,\mu}\left[\mathrm{i}\hat{r}_\mu^a r_\mu^a - \beta\ell\left(y_\mu r_\mu^a + \kappa\right)\right]} \right. \\ &\times \left[\prod_{\mu=1}^n \mathrm{e}^{-\frac{\Delta}{2}\sum_{a,b=1}^n \hat{r}_\mu^a \hat{r}_\mu^b Q_{ab} - \sum_{a=1}^n \mathrm{i} y_\mu \hat{r}_\mu^a m_a}\right] \exp\left(d\sum_{a\le b} \mathrm{i}\hat{Q}_{ab}Q_{ab} + d\sum_{a=1}^n \mathrm{i}\hat{m}_a m_a\right) \\ &\left. \times \exp\left(-\frac{d}{2}\log\det\left(\mathrm{i}\tilde{\boldsymbol{Q}} + \frac{d}{2}\sum_{a,b}\mathrm{i}\tilde{Q}_{ab}^{-1}\hat{m}_a\hat{m}_b\right)\right)\right\rangle_\beta , \end{aligned} \tag{1.2.47}$$

where we have defined $\tilde{Q}_{ab} = (1+\delta_{a,b})\hat{Q}_{ab} - \mathrm{i}\delta_{a,b}\lambda/T$ ($\delta_{a,b}$ being the Kronecker delta) and we have used that $\|\boldsymbol{w}^*\|^2 = d$. Integrating over $\hat{\boldsymbol{m}}$, we find

$$\begin{split}\langle Z_d\left(\boldsymbol{X},\boldsymbol{y}\right)^p\rangle_\beta \propto \Bigg\langle \int \left(\prod_{a,\mu}\frac{\mathrm{d}r_\mu^a \mathrm{d}\hat{r}_\mu^a}{2\pi}\right)\left(\prod_{a\leq b}\mathrm{d}Q_{ab}\right)\left(\prod_a \mathrm{d}m_a\right)\mathrm{e}^{\sum_{a,\mu}\left[\mathrm{i}\hat{r}_\mu^a r_\mu^a - \beta\ell\left(y_\mu r_\mu^a + \kappa\right)\right]} \\ \times\left[\prod_{\mu=1}^{\alpha d}\exp\left(-\frac{\Delta}{2}\sum_{a,b=1}^n \hat{r}_\mu^a\hat{r}_\mu^b Q_{ab} - \sum_{a=1}^n \mathrm{i}y_\mu\hat{r}_\mu^a m_a\right)\right] \\ \times\underbrace{\left(\prod_{a\leq b}\mathrm{d}\hat{Q}_{qb}\right)\exp\left(d\sum_{a\leq b}\mathrm{i}\hat{Q}_{ab}Q_{ab} - \frac{d}{2}\log\det\left(\mathrm{i}\tilde{\boldsymbol{Q}}\right) - \frac{d}{2}\sum_{a,b}\mathrm{i}\tilde{Q}_{ab}m_a m_b\right)}_{\mathcal{I}}\Bigg\rangle_\beta .\end{split} \tag{1.2.48}$$

Let us consider the integral over $\hat{\boldsymbol{Q}}$, that we have called $\mathcal{I}$. We can perform a saddle-point approximation for large $d$ by extremising the exponent in $\mathcal{I}$ with respect to the $\widehat{Q}_{ab}$ variables:

$$\frac{\partial}{\partial\widehat{Q}_{ab}}\left\{\frac{\mathrm{i}}{2}\sum_{a,b}\left(\tilde{Q}_{ab} + \mathrm{i}\beta\lambda\delta_{a,b}\right)Q_{ab} - \frac{1}{2}\log\det(\mathrm{i}\tilde{Q}) - \frac{\mathrm{i}}{2}\sum_{a,b} m_a\tilde{Q}_{ab}m_b\right\} = 0. \tag{1.2.49}$$

Eq. (1.2.49) implies

$$\mathrm{i}Q_{ab} - \tilde{Q}_{ab}^{-1} - \mathrm{i}m_a m_b = 0. \tag{1.2.50}$$

Hence, the solution of the saddle-point equation is

$$\tilde{\boldsymbol{Q}} = -\mathrm{i}\boldsymbol{U}^{-1}, \tag{1.2.51}$$

where $U_{ab} = Q_{ab} - m_a m_b$. The average partition function becomes

$$\begin{split}\langle Z_d\left(\boldsymbol{X},\boldsymbol{y}\right)^p\rangle_\beta \propto \Bigg\langle \int \left(\prod_{a\leq b}\mathrm{d}Q_{ab}\right)\left(\prod_a \mathrm{d}m_a\right)\mathrm{e}^{\frac{d}{2}\log\det(\boldsymbol{U}) - \frac{\beta\lambda d}{2}\sum_a Q_{aa}} \\ \times\left(\prod_{a,\mu}\frac{\mathrm{d}r_\mu^a\mathrm{d}\hat{r}_\mu^a}{2\pi}\right)\left[\prod_{\mu=1}^{\alpha d}\mathrm{e}^{-\frac{\Delta}{2}\sum_{a,b=1}^n \hat{r}_\mu^a\hat{r}_\mu^b Q_{ab} + \sum_{a=1}^n\left[\mathrm{i}\hat{r}_\mu^a(r_\mu^a - y_\mu m_a) - \beta\ell\left(y_\mu r_\mu^a + \kappa\right)\right]}\right]\Bigg\rangle .\end{split} \tag{1.2.52}$$

In order to compute the integral over the variables $\hat{r}_\mu^a$ and $r_\mu^a$, it is useful to rewrite the Gaussian factor $\mathcal{F} = \exp\left(-\frac{\Delta}{2}\sum_{a,b,\mu}\hat{r}_\mu^a\hat{r}_\mu^b Q_{ab}\right)$ as a differential operator acting on the product over the replicas' index $c$:

$$\mathcal{F} = \prod_{\mu=1}^n \exp\left(\frac{\Delta}{2}\sum_{a,b=1}^p Q_{ab}\frac{\partial^2}{\partial h_a^\mu \partial h_b^\mu}\right)\left(\prod_{c=1}^p \exp\left(-\mathrm{i}\hat{r}_\mu^c h_c^\mu\right)\right)\Bigg|_{h_c^\mu = 0}. \tag{1.2.53}$$

Hence the integral over $\hat{r}_\mu^a$, $r_\mu^a$ is equal to

$$\begin{split}\prod_{\mu=1}^n \exp\left(\frac{\Delta}{2}\sum_{a,b=1}^p Q_{ab}\frac{\partial^2}{\partial h_a^\mu\partial h_b^\mu}\right) \\ \times\left(\int\prod_{c=1}^p\frac{\mathrm{d}r_\mu^c\mathrm{d}\hat{r}_\mu^c}{2\pi}\exp\left(-\beta\ell\left(y_\mu r_\mu^c + \kappa\right) + \mathrm{i}\hat{r}_\mu^c(r_\mu^c - h_c^\mu - y_\mu m_c)\right)\Bigg|_{h_c^\mu=0}\right).\end{split} \tag{1.2.54}$$

Note that the input-output correlations are only between a training sample and the corresponding label, i.e., within the same $\mu$, therefore it is possible to factorise over $\mu$. Finally, we can use the Fourier representation of the Dirac $\delta-$function and rewrite

$$\langle Z_d\left(\boldsymbol{X},\boldsymbol{y}\right)^p\rangle_\beta \propto \int \left(\prod_{a\leq b} \mathrm{d}Q_{ab}\right)\left(\prod_{a}\mathrm{d}m_a\right) \mathrm{e}^{\frac{d}{2}\log\det(\boldsymbol{U})-\frac{\beta\lambda d}{2}\sum_a Q_{aa}+\alpha d\log\mathcal{Z}}, \tag{1.2.55}$$

where

$$\mathcal{Z} = \mathbb{E}_y\left[\exp\left(\frac{\Delta}{2}\sum_{a,b=1}^{p} Q_{ab}\frac{\partial^2}{\partial h_a\partial h_b}\right)\left(\prod_{c=1}^{p}\mathrm{e}^{-\beta\ell(yh_c+ym_c+\kappa)}\Big|_{h_c=0}\right)\right]. \tag{1.2.56}$$

Therefore, in the large $d$ limit the action is dominated by its saddle point. We can evaluate it by the Laplace method (Wong, 1989):

$$\langle Z_d\left(\boldsymbol{X},\boldsymbol{y}\right)^p\rangle \simeq \exp\left(dS\left(\left\{Q^*_{ab}\right\},\left\{m^*_a\right\}\right)\right), \tag{1.2.57}$$

where we have defined the *replicated action* $S$

$$S\left(\left\{Q_{ab}\right\},\left\{m_a\right\}\right) = \frac{1}{2}\log\det\left(\boldsymbol{U}\right) - \frac{\beta\lambda}{2}\sum_a Q_{aa} + \alpha\log\mathcal{Z}, \tag{1.2.58}$$

and $\left\{Q^*_{ab}\right\},\left\{m^*_a\right\}$ are the solutions of the saddle-point equations

$$\frac{\partial}{\partial Q_{ab}}S\left(\{Q_{ab}\},\{m_a\}\right)\Big|_{\{Q^*_{ab}\},\{m^*_a\}} = 0 \qquad \forall a\leq b, \tag{1.2.59}$$

$$\frac{\partial}{\partial m_a}S\left(\{Q_{ab}\},\{m_a\}\right)\Big|_{\{Q^*_{ab}\},\{m^*_a\}} = 0 \qquad \forall a. \tag{1.2.60}$$

**Replica symmetric ansatz** — We look for a subspace of solutions with a symmetry among all the replicas. To this end, we introduce a replica symmetric (RS) ansatz defined as follows

$$Q_{ab} = r\delta_{ab} + q(1-\delta_{ab}), \quad m_a = m. \tag{1.2.61}$$

The first term in the action is then

$$\begin{aligned}\frac{1}{2}\log\det\boldsymbol{U} =& \frac{1}{2}\log\det\begin{bmatrix} r-m^2 & q-m^2 & \dots & q-m^2\\ q-m^2 & r-m^2 & \dots & q-m^2\\ q-m^2 & \dots & \dots & r-m^2\end{bmatrix}\\ =& \frac{1}{2}\log\left[(r-q)^{p-1}(r-q+p(q-m^2))\right],\end{aligned} \tag{1.2.62}$$

using the matrix determinant lemma. The second term is simply $-p\beta\lambda r/2$, and the third term is

$$\begin{aligned}&\alpha\log\mathcal{Z} =\\ \alpha\log\mathbb{E}_y&\left[\mathrm{e}^{\frac{\Delta q}{2}\left(\sum_a\frac{\partial}{\partial h_a}\right)^2}\prod_c \mathrm{e}^{\frac{\Delta(r-q)}{2}\frac{\partial^2}{\partial h_c^2}}\exp\left(-\frac{1}{T}\ell\left(yh_c+ym_c+\kappa\right)\right)\Big|_{h_c=0}\right].\end{aligned} \tag{1.2.63}$$

We now use the following two identities (Nishimori, 2001; Urbani, 2018):

$$\exp\left(\frac{\omega}{2}\frac{\partial^2}{\partial h^2}\right)g(h)=\int_{-\infty}^{+\infty}\frac{dz}{\sqrt{2\pi\omega}}\mathrm{e}^{-\frac{z^2}{2\omega}}g(h-z)=\gamma_\omega * g(h), \tag{1.2.64}$$

where $\gamma_\omega \sim \mathcal{N}(0,\omega)$, and

$$\left(\sum_{a=1}^{n}\frac{\partial}{\partial h_a}\right)^{\tau} g(h_1,...,h_t)\bigg|_{\{h_c=h\}}=\frac{\partial^\tau}{\partial h^\tau}g(h,...,h), \tag{1.2.65}$$

to rewrite

$$\alpha\log\mathcal{Z}=\alpha\log\mathbb{E}_y\left[\gamma_{\Delta q} * \left[\gamma_{\Delta(r-q)} * \exp\left(-\frac{1}{T}\ell\left(yh+ym+\kappa\right)\right)\right]^p\bigg|_{h=0}\right]. \tag{1.2.66}$$

Under the RS ansatz, the expression of the action is thus

$$\begin{aligned}S(r,q,m)=&\frac{1}{2}\left[(p-1)\log(r-q)+\log(r-q+p(q-m^2))\right]-p\frac{\beta\lambda r}{2}\\&+\alpha\log\mathbb{E}_y\left[\gamma_{\Delta q} * \left[\gamma_{\Delta(r-q)} * \exp\left(-\beta\ell\left(yh+ym+\kappa\right)\right)\right]^p\bigg|_{h=0}\right].\end{aligned} \tag{1.2.67}$$

In the limit $p\to 0$ :

$$\begin{aligned}S(r,q,m)\simeq p\tilde{S}(r,q,m)=p\Bigg(&\frac{1}{2}\log(r-q)+\frac{q-m^2}{2(r-q)}-\beta\frac{\lambda r}{2}\\+\alpha\mathbb{E}_y&\left[\gamma_{\Delta q} * \log\left(\gamma_{\Delta(r-q)} * \exp\left(-\beta\ell\left(yh+ym+\kappa\right)\right)\right)\bigg|_{h=0}\right]\Bigg).\end{aligned} \tag{1.2.68}$$

Finally, by setting to zero the derivatives of the action with respect to $r$, $q$ and $m$, we obtain the following saddle-point equations in the RS-ansatz :

$$\begin{aligned}\frac{1}{(r-q)}&=\\=\beta\lambda-\alpha\Delta\int_{-\infty}^{+\infty}\frac{\mathrm{d}h}{\sqrt{2\pi\Delta q}}e^{-h^2/2\Delta q}\mathbb{E}_y&\left[\frac{\partial^2}{\partial h^2}\log\left(\gamma_{\Delta(r-q)} * \mathrm{e}^{-\beta\ell(yh+ym+\kappa)}\right)\right],\end{aligned} \tag{1.2.69}$$

$$\begin{aligned}\frac{q-m^2}{(r-q)^2}&=\\\alpha\Delta\int_{-\infty}^{+\infty}\frac{\mathrm{d}h}{\sqrt{2\pi\Delta q}}e^{-h^2/2\Delta q}\mathbb{E}_y&\left[\left(\frac{\partial}{\partial h}\log\left(\gamma_{\Delta(r-q)} * \mathrm{e}^{-\beta\ell(yh+ym+\kappa)}\right)\right)^2\right],\end{aligned} \tag{1.2.70}$$

$$\frac{m}{(r-q)}=\alpha\int_{-\infty}^{+\infty}\frac{\mathrm{d}h}{\sqrt{2\pi\Delta q}}e^{-h^2/2\Delta q}\mathbb{E}_y\left[\frac{\partial}{\partial m}\log\left(\gamma_{\Delta(r-q)} * \mathrm{e}^{-\beta\ell(yh+ym+\kappa)}\right)\right]. \tag{1.2.71}$$

**Low temperature expansion —** In this paragraph, we show how to track the performance of ERM by looking at the ground state of the system. In order to compute Eqs. (1.2.69)–(1.2.71) in the $\beta \to \infty$ ($T = 1/\beta \to 0$) limit, we focus on the UNSAT phase, where the solution is unique due to the convexity of the loss functions under consideration. Since in this case the volume of possible solutions shrinks to zero, we can use the ansatz: $q \simeq r - \chi T$ as $T \to 0$, where $\chi > 0$ is constant with respect to $T$. This ansatz reflects the fluctuation-dissipation theorem FDT for equilibrium systems that is further discussed in Chapter 2.3. The inner convolution in Eqs. (1.2.69)-(1.2.71) can be rewritten as

$$\log\left(\gamma_{\Delta\chi T} * \exp\left(-\frac{1}{T}\ell\left(yh+\kappa\right)\right)\right) \underset{T\to 0}{\simeq} -\frac{1}{T} V_{\text{eff}}\left(z^*|h,m,\chi\right), \tag{1.2.72}$$

where we have changed variable $h \leftarrow yh - m - y\kappa$ and $V_{\text{eff}}$ and $z^*$ are respectively, the *Moreau envelope* (Parikh et al., 2014; Bauschke et al., 2011):

$$V_{\text{eff}}\left(z^*|h,m,\chi\right) = \min_z \left(\frac{1}{2\chi}(z-h)^2 + \ell(z)\right), \tag{1.2.73}$$

and the *proximal map*:

$$z^*(h,y,\chi) = \text{argmin}_z \left(\frac{1}{2\chi}(z-h)^2 + \ell(z)\right), \tag{1.2.74}$$

which is unique since the empirical risk is strictly convex in the UNSAT phase and given by the implicit relation

$$z^* = h - \chi\ell'(z^*). \tag{1.2.75}$$

We can then rewrite the final equations in the zero-temperature limit:

$$\begin{aligned} \frac{1}{\chi} &= \lambda + \frac{\alpha}{\chi}\left(1 - \mathbb{E}_{y,h}\left[\partial_h z^*(h,y,\chi)\right]\right) \\ \frac{q-m^2}{\chi^2} &= \frac{\alpha}{\Delta}\mathbb{E}_{y,h}\left[(\ell'(z^*))^2\right], \\ \frac{m}{\chi} &= \frac{\alpha}{\Delta}\mathbb{E}_{y,h}\left[\ell'(z^*)\right], \end{aligned} \tag{1.2.76}$$

where the dependence on $y$ is hidden in $h \sim \mathcal{N}(m + y\kappa, \Delta q)$, and $y = \pm 1$, $\mathrm{P}(y = +1) = \rho \in (0,1)$. Finally, noting that at zero temperature the free energy equals the energy, the equation for the bias can be simply obtained by extremising the action with respect to $\kappa$:

$$\mathbb{E}_{y,h}\left[y(z^* - h)\right] = 0. \tag{1.2.77}$$

The above set of equations must be evaluated for each of the losses under consideration, which is done in the appendix of Article 1. In the case of the square loss, the solution is analytic. However, in general and for instance in the case of logistic and hinge losses, the above set of equations must be solved self-consistently. Note that, in this case, given the uniqueness of the solution, the RS ansatz is always stable (Franz et al., 2017).

**The critical value for the separability phase transition —** In order to derive the separability (SAT-UNSAT) transition from Eqs. (1.2.76), it is convenient to define the random variable $u^*(h, y, \chi) = \ell'(z^*(h, y, \chi))$, where $z^*$ is given by Eq. (1.2.75). Notice that this definition is intuitive if we rewrite the convex loss function $\ell$ as a Legendre transformation:

$$\ell(v) = \max_u \{vu - \tilde{\ell}(u)\}, \tag{1.2.78}$$

where the convex conjugate $\tilde{\ell}(u)$ is defined as

$$\tilde{\ell}(u) = \max_v \{uv - \ell(v)\}. \tag{1.2.79}$$

It is straightforward to see that $\tilde{\ell}'(u^*) = h - \chi u^* = z^*$. Given that $h \sim \mathcal{N}(m + y\kappa, \Delta q)$, it follows that the cumulant distribution function of $u^*$ is given by

$$\mathbb{P}(u^* \leq u) = \rho\, Q\left(\frac{\tilde{\ell}'(u) + \chi u - m - \kappa}{\sqrt{\Delta q}}\right) + (1-\rho)\, Q\left(\frac{\tilde{\ell}'(u) + \chi u - m + \kappa}{\sqrt{\Delta q}}\right), \tag{1.2.80}$$

where again $Q(\cdot)$ denotes the distribution function of a standard normal random variable. We consider convex and monotonically decreasing loss functions, with $\ell(+\infty) = \ell'(+\infty) = 0$. It follows that $\ell'(-\infty) < u^* < 0$. We can now use the identity

$$\mathbb{E}_{y,h}[(u^*)^2] = (-2) \int_{\ell'(-\infty)}^{0} \mathrm{d}u\, u\, \mathbb{P}(u^* \leq u) \tag{1.2.81}$$

to rewrite the second of Eqs. (1.2.76) as

$$\begin{aligned} 1 - \frac{m^2}{q} &= \\ \frac{\alpha\chi^2}{\Delta q} \int_0^{-\ell'(-\infty)} \mathrm{d}u\, 2u \Bigg[\rho\, Q\left(\frac{\tilde{\ell}'(-u) - \chi u - m - \kappa}{\Delta q}\right) & \\ + (1-\rho)\, Q\left(\frac{\tilde{\ell}'(-u) - \chi u - m + \kappa}{\Delta q}\right)\Bigg]. & \end{aligned} \tag{1.2.82}$$

We now introduce the following rescaled variables: $\tilde{\chi} = \chi/\sqrt{q}$, $\varrho := m/\sqrt{q}$, $\tilde{\kappa} = \kappa/\sqrt{q}$, that control the performance as can be easily seen from the generalisation error in Eq. (1.2.17). With the new definitions and rescaling the integration variable $u$ by $\tilde{\chi}$, we find

$$1 - \varrho^2 = \frac{\alpha}{\Delta} \mathcal{S}(\tilde{\chi}, q, \varrho, \tilde{\kappa}), \tag{1.2.83}$$

where

$$\mathcal{S}(\tilde{\chi}, q, \varrho, \tilde{\kappa}) = \frac{\alpha}{\Delta} \int_0^{-\tilde{\chi}\ell'(-\infty)} \mathrm{d}u\, 2u \left[\rho\, Q\left(\frac{\tilde{\ell}'(-u)}{\Delta q} + \frac{-\tilde{\chi}u - \varrho - \tilde{\kappa}}{\Delta}\right)\right. \tag{1.2.84}$$

$$\left. + (1-\rho) Q\left(\frac{\tilde{\ell}'(-u)}{\Delta q} + \frac{-\tilde{\chi}u - \varrho + \tilde{\kappa}}{\sqrt{\Delta}}\right)\right] \tag{1.2.85}$$

Notice that, for any fixed $\tilde{\chi}$ and $\varrho$, the function $\mathcal{S}$ is monotonically decreasing as we increase $q$. Moreover,

$$\lim_{q\to\infty} \mathcal{S}(\tilde{\chi}, q, \varrho, \tilde{\kappa}) = \mathcal{S}^*(\tilde{\chi}, \varrho, \tilde{\kappa}) := \int_0^{-\tilde{\chi}\ell'(-\infty)} \mathrm{d}u\, 2u \left[\rho\, Q\left(\frac{-\tilde{\chi}u - \varrho - \tilde{\kappa}}{\Delta}\right)\right. \tag{1.2.86}$$

$$\left. +(1-\rho) Q\left(\frac{-\tilde{\chi}u - \varrho + \tilde{\kappa}}{\sqrt{\Delta}}\right)\right]. \tag{1.2.87}$$

Clearly, $\mathcal{S}^*(\tilde{\chi}, \varrho, \tilde{\kappa})$ is monotonic with respect to $\tilde{\chi}$, and it has a finite limit as $\tilde{\chi} \to \infty$, i.e.,

$$\lim_{\tilde{\chi}\to\infty} \mathcal{S}^*(\tilde{\chi}, \varrho, \tilde{\kappa}) = \Delta \int_0^{\infty} \mathrm{d}u\, u^2 \left[\rho\, f\left(u + \frac{\varrho + \tilde{\kappa}}{\sqrt{\Delta}}\right) + (1-\rho)\, f\left(u + \frac{\varrho - \tilde{\kappa}}{\sqrt{\Delta}}\right)\right], \tag{1.2.88}$$

where $f$ is the probability density function of $\mathcal{N}(0,1)$. An implication of this limit being finite is that the solution $\tilde{\chi}$ of Eq. (1.2.83) tends to $\infty$ as $q \to \infty$ if

$$\alpha < \frac{\Delta(1-\varrho^2)}{\mathcal{S}^*(\infty, \varrho)}. \tag{1.2.89}$$

Remembering the definition of the ansatz we have used for the UNSAT phase: $q = r - \chi T$, we can already guess that the divergence of $\tilde{\chi}$ reveals that the solution is not unique anymore and Eq. (1.2.89) defines the SAT phase. Moreover, it follows from the definition of $z^*$ (Eq. (1.2.75)) that, as $\chi \to \infty$, $\ell'(z^*) \to 0$ and thus $\ell(z^*) \to 0$. Consequently,the average training error vanishes in this region. The transition is thus given by maximising the right-hand side of Eq. (1.2.89):

$$\begin{aligned} \alpha^* &= \max_{0\le\varrho\le1,\kappa} \zeta(\varrho, \kappa), \\ \zeta(\varrho, b) &= \frac{1-\varrho^2}{\int_0^\infty \mathrm{d}u\, u^2 \left[\rho\, f\left(u + \frac{\varrho}{\sqrt{\Delta}} - \kappa\right) + (1-\rho)\, f\left(u + \frac{\varrho}{\sqrt{\Delta}} + \kappa\right)\right]}. \end{aligned} \tag{1.2.90}$$

This characterisation can be interpreted as follows: if there exists a $\varrho$ that satisfies Eq. (1.2.89), then as we move along the "ray" of constant slope $\varrho = m/\sqrt{q}$, the training loss vanishes.

## 1.2.4 . The observed high-dimensional phenomena

In this section, we evaluate the above formulas and investigate the dependence of the test error on the regularisation strength $\lambda$, the sample complexity $\alpha$, the noise variance $\Delta$ and the cluster size $\rho$.

Keeping in mind that the minimisation of the non-regularised logistic loss corresponds to the MLE in the considered model, we pay particular attention to it as a benchmark for the performance of the most commonly used method in statistics. Another important benchmark is the BO performance, that provides a threshold that no algorithm can improve.

**Weak and strong regularisation** — Figure 1.2.2 summarises how the regularisation strength $\lambda$ and the cluster size $\rho$ influence the generalisation performance. The left panel displays the balanced case $\rho = 0.5$, while the right panel shows the unbalanced one at $\rho = 0.2$. Let us recall that $\alpha^*$ is defined as the value such that for $\alpha < \alpha^*(\Delta, \rho)$ the problem lies in the SAT phase, therefore a solution exists and the training loss vanishes. In other words, the data are linearly separable. In the left panel of Figure 1.2.2, we depict (in green) the performance of the non-regularised logistic loss, a.k.a. the MLE. For $\alpha > \alpha^*(\Delta, \rho)$, the training data are not linearly separable and the minimum training loss is bounded away from zero. For $\alpha < \alpha^*(\Delta, \rho)$ the data are linearly separable, in which case properly speaking the MLE is ill-defined (Sur & Candès, 2019), the curve that we depict is the limiting value reached as $\lambda \to 0^+$. The points are the results of simulations with a standard `scikit-learn` (Pedregosa et al., 2011) package. As shown by Soudry et al. (2018b), even though the logistic estimator does not exist, GD actually converges to the max-margin solution in this case, or equivalently to the minimal norm solution that classifies all samples correctly, a phenomenon called “implicit regularisation” that is well illustrated here.

Figure 1.2.2 further depicts (in purple) the performance of the BO error given by Eq. (1.2.32). We have also evaluated the performance of both logistic and square losses at optimal value of regularisation parameter $\lambda$. This is where the balanced case (left panel) differs *crucially* from the unbalanced one (right panel). While in the high-dimensional limit of the balanced case the optimal regularisation diverges to infinity $\lambda_{\text{opt}} \to \infty$ and the corresponding error matches exactly the BO one, in the unbalanced case $0 < \lambda_{\text{opt}} < \infty$ and the error for both losses is bounded away from the BO one for any $\alpha > 0$. We give below a fully analytic argument for the perhaps unexpected property of achieving Bayes-optimality at $\lambda_{\text{opt}} \to \infty$ and $\rho = 0.5$ valid for any loss that has finite 2nd derivative at the origin.

**On the Bayes-optimality of infinite regularisation in the balanced-clusters case** — We start by considering the square loss. At $\rho = 0.5$, it is straightforward to check from Eq. (1.2.33) that the bias $\kappa = 0$ and the generalisation error is given by Eq. (1.2.17) and reads

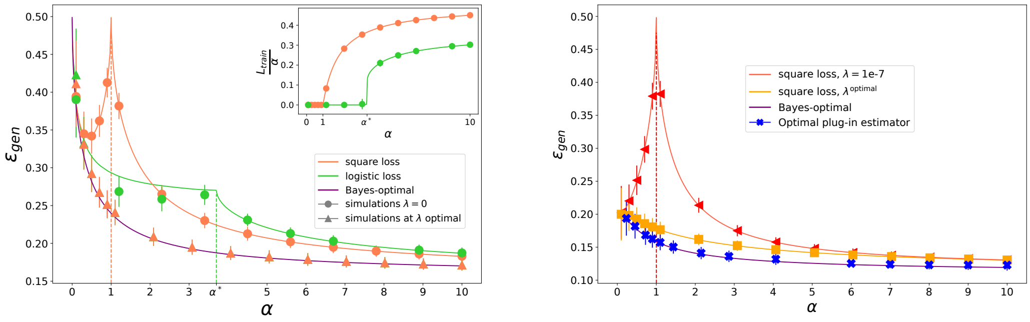
$$\varepsilon_{\text{gen}} = Q\left(\frac{m}{\sqrt{\Delta q}}\right), \tag{1.2.91}$$

where $m$ and $q$ are obtained via Eqs. (1.2.76), evaluated at $\rho = 0.5$. The BO error for this problem is given by Eq. (1.2.32) and reads

$$\varepsilon_{\text{gen}}^{\text{BO}} = Q\left(\frac{\alpha}{\Delta(\Delta + \alpha)}\right). \tag{1.2.92}$$

Therefore, in order to reach Bayes-optimality we need a weight vector $\boldsymbol{w}$ with an overlap $m$ and squared norm $q$ such that

$$\frac{m}{\sqrt{q}} = \sqrt{\frac{\alpha}{\Delta + \alpha}} \tag{1.2.93}$$



(a) **Equal cluster size, $\rho = 0.5$.** It is possible to tune the regularisation strength $\lambda$ in order to reach the optimal performance. The inset displays the training loss as a function of $\alpha$. The training loss is close to zero up to the interpolation transition $\alpha^*$.

(b) **Unequal cluster size, $\rho = 0.2$.** The BO error cannot be achieved by the optimally regularised losses under consideration. Instead, the BO performance can be reached by the optimal plug-in defined by Eq. (1.2.33) (simulations at $d = 5000$).

Figure 1.2.2 – Generalisation error $\varepsilon_{\text{gen}}$ as a function of the sample complexity $\alpha$ at optimal and low regularisation ($\lambda = 10^{-7}$) and fixed noise variance $\Delta = 1$. The dashed vertical lines mark the interpolation thresholds. The error achieved by the square and logistic losses is compared to the BO one. We compare our theoretical findings with numerical simulations at dimension $d = 1000$, marked by the symbols.

By using Eq. (1.2.76) at $\rho = 0.5$, Eq. (1.2.93) can be rewritten as

$$\frac{\Delta q}{(1-m)^2} = 0. \tag{1.2.94}$$

Eq. (1.2.94) is verified by the fixed point equations only at $\lambda \to \infty$. Indeed in this limit we find that:

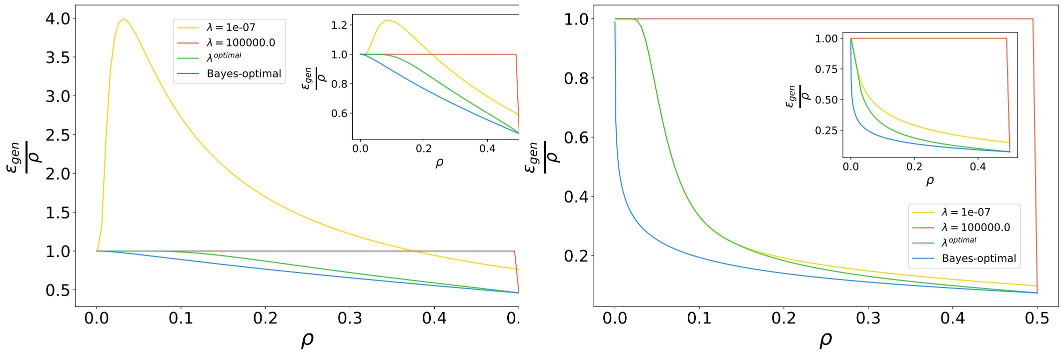
$$\chi = \frac{\Delta}{\lambda} + o\left(\lambda^{-1}\right), \quad m = \frac{\alpha}{\lambda} + o\left(\lambda^{-1}\right), \quad q = \frac{\alpha}{\lambda^2}(\Delta + \alpha) + o\left(\lambda^{-2}\right), \tag{1.2.95}$$

so that

$$\frac{m}{\sqrt{q}} \xrightarrow{\lambda \to \infty} \sqrt{\frac{\alpha}{\Delta + \alpha}}. \tag{1.2.96}$$

Therefore, as $\lambda$ grows and the $\ell_2-$norm of the weight vector goes to zero, the vector aligns itself optimally to the hidden one and the generalisation error becomes optimal. A similar scaling holds for the hinge loss as can be checked via Eqs. (1.2.76) (in particular, using Eqs. D.16-18 of Appendix D in Article 1). We can then see why this remains correct for any twice differentiable loss $\ell(\cdot)$ with finite second derivative at the origin. As long as the $\ell_2-$norm vanishes when $\lambda \to \infty$, we can expand

$$\ell\left(y\boldsymbol{x}^\top \boldsymbol{w}\right) = \ell(0) + y\boldsymbol{x}^\top \boldsymbol{w}\ell'(0) + \frac{1}{2}\left(\boldsymbol{x}^\top \boldsymbol{w}\right)^2 \ell''(0) + o(q). \tag{1.2.97}$$



(a) $\alpha = 1.2$, $\Delta = 1$.

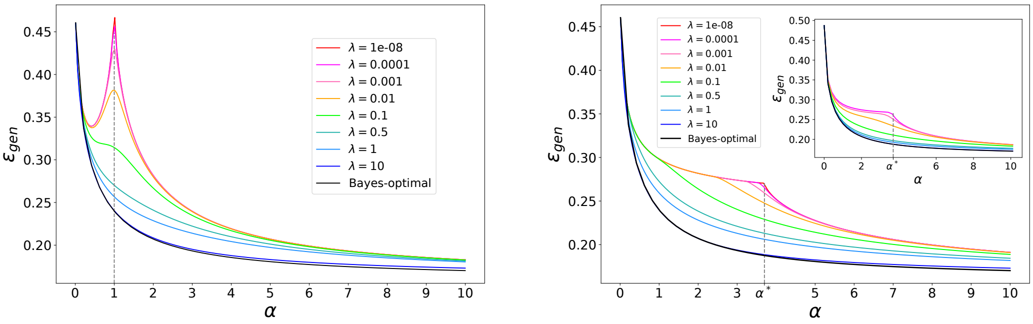
(b) $\alpha = 7$, $\Delta = 0.3$.

Figure 1.2.3 – Generalisation error $\varepsilon_{\text{gen}}$ as a function of the cluster size $\rho$ for the square loss, compared to the BO performance. The insets display the same figure for the hinge loss. The vertical axis is rescaled by $\rho$ for visibility purposes. The error is computed at low ($\lambda = 10^{-7}$), high ($\lambda = 10^5$) and optimal regularisation. We observe that Bayes-optimality at infinite regularisation holds strictly at $\rho = 0.5$.

If $\ell'(0) < 0$ and $\ell''(0) > 0$, which hold for the logistic loss that is of interest in this problem, the loss behaves like the square one. This is the origin of the peculiar behaviour of Bayes-optimality observed at $\lambda \to \infty$ for the balanced case $\rho = 0.5$. We observe numerically that this peculiar behaviour is not valid anymore as soon as $\rho \neq 0.5$, as shown in Figure 1.2.3 where the generalisation error is plotted as a function of $\rho$ at zero, infinite and optimal regularisation for the square and hinge losses.

**Regularisation and the interpolation peak** — In Figure 1.2.4 we depict the dependence of the generalisation error on the regularisation strength $\lambda$ in the balanced case $\rho = 0.5$ for the square, hinge, and logistic losses. The curves at small regularisation show the interpolation peak at $\alpha^* = 1$ for the square loss and $\alpha^*$ for all the losses that go to zero whenever the data are linearly separable. We observe a smooth disappearance of the peak as regularisation is added, similarly as what has been observed in other models that present the interpolation peak (Hastie et al., 2022; Mei & Montanari, 2022) in the case of the square loss. Here we thus show that a similar phenomenon arises with the logistic and hinge losses as well, this is interesting since the same phenomenon has been observed in DNNs using a logistic/cross-entropy loss (Geiger et al., 2019; Nakkiran et al., 2021). In fact, as the regularisation increases, the error gets better in this model with equal-size clusters, and the BO error is reached at large regularisation.

**Max-margin and weak regularisation** — Figure 1.2.5a illustrates the generic property that all monotone non-increasing loss functions converge to the max-margin solution for linearly separable data as $\lambda \to 0^+$ (Rosset et al., 2004). Figure 1.2.5a depicts a very slow convergence towards this result as a function of the regularisation parameter $\lambda$ for the logistic loss. While for $\alpha > \alpha^*$ the performance of both



(a) **Square.**

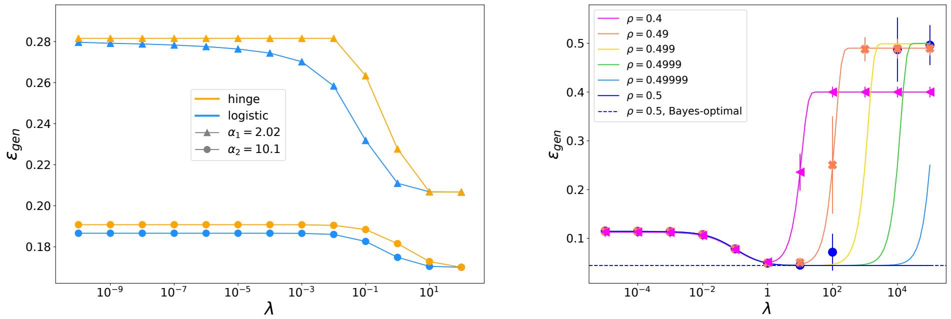
(b) **Hinge** (main) and **logistic** (inset).

Figure 1.2.4 – Generalisation error $\varepsilon_{\text{gen}}$ as a function of the sample complexity $\alpha$ for different values of the regularisation strength $\lambda$, at fixed cluster variance $\Delta = 1$ and balanced cluster size $\rho = 0.5$. The performance of ERM is compared to the BO one. If the two clusters have the same size,the BO error can be reached by increasing the regularisation. The regularisation smoothens the curves and makes the “kink” disappear in all cases.

the hinge and logistic losses is basically indistinguishable from the asymptotic one already at $\log\lambda \approx -3$, for $\alpha < \alpha^*$ the convergence of the logistic loss still did not happen even at $\log\lambda \approx 10$.

**Cluster sizes and regularisation** — In Figure 1.2.5b we study in greater detail the dependence of the generalisation error both on the regularisation $\lambda$ and $\rho$, as $\rho \to 0.5$. We see that the optimality of $\lambda \to \infty$ holds strictly at $\rho = 0.5$, and at any $\rho$ close to 0.5 the error at $\lambda \to \infty$ is very large and there is a well-delimited region of $\lambda$ for which the error is close to (but strictly above) the BO error. As $\rho \to 0.5$ this interval is getting longer and longer until it diverges at $\rho = 0.5$. It needs to be stressed that this result is asymptotic, holding only when $d, n \to \infty$ while $\alpha = n/d$ is fixed. The finite-size fluctuations cause that the finite size system behaves rather as if $\rho$ was close but not equal to 0.5, and at finite size if we set $\lambda$ arbitrarily large then we reach a high generalisation error. We instead need to optimise the value of $\lambda$ for finite sizes, for instance by cross-validation.

**Separability phase transition** — The position of the interpolation threshold when data become linearly separable has a well-defined limit in the high-dimensional regime as a function of the sample complexity. The kink in generalisation indeed occurs at a value $\alpha^*$ when the training loss of logistic and hinge losses goes to zero (while for the square loss the peak appears at $\alpha^* = 1$ when the system of $n$ linear equations with $d$ parameters become solvable). The position of $\alpha^*$, given by Eq. (1.2.90), is shown in Figure 1.2.6 as a function of the noise variance $\Delta$ and for different values of $\rho$. For very large cluster variance, the data become random and hence $\alpha^* = 2$ for balanced clusters, as famously derived in the classical work



(a) **Hinge** (orange) and **logistic** (blue) losses. We take $\Delta = 1$, $\rho = 0.5$ and two different values of $\alpha$: $\alpha_1 = 2$, $\alpha_2 = 10$. As $\lambda \to 0^+$, the error of the two losses approaches the same value if the data are separable ($\alpha_1 < \alpha^*$), although we observe a very slow convergence. The error values reached as $\lambda \to 0^+$ are not the same if the data are not separable ($\alpha_2 > \alpha^*$). At large $\lambda$, the error of both losses approaches the BO, for all values of $\alpha$.

(b) **Square** loss.We take $\Delta = 0.3$, $\alpha = 2$ and different values of $\rho$ close to 0.5. At all $\rho < 0.5$, the error exhibits a minimum at finite $\lambda = \lambda^*$ and reaches a plateau at $\lambda > \lambda^*$. The value of the error at the plateau is $\varepsilon_{\text{gen}} = \min\{\rho, 1-\rho\}$, i.e., the error attained by the greedy strategy assigning all points to the larger cluster. The symbols mark simulations for $\rho = 0.4$ ($d = 1000$) and $\rho = 0.49, 0.5$ ($d = 10000$). Due to the finite dimensionality, effectively $\rho < 0.5$ in the numerics. and the error always plateaus in simulations.

Figure 1.2.5 – Generalisation error $\varepsilon_{\text{gen}}$ as a function of the regularisation strength $\lambda$.

by Cover (1965). When $\rho < 0.5$, however, it is easier to separate linearly the data points and the limiting value of $\alpha^*$ gets larger and differs from Cover's. For finite $\Delta$, the two Gaussian distributions become distinguishable, and the data acquires structure. Consequently, the critical threshold $\alpha^*$ is growing as the correlations make data easier to be linearly separated, similarly as described in Sur & Candès (2019).

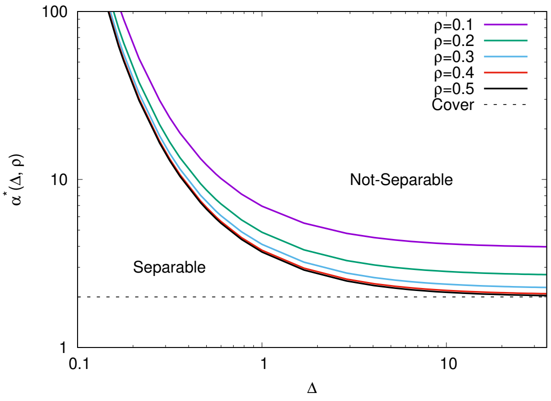


Figure 1.2.6 – Critical value of the sample complexity $\alpha = \alpha^*$ at which the linear separability transition occurs, as a function of the noise variance $\Delta$ and for different values of the cluster size $\rho$. Similarly as in the case of Gaussian data (Sur & Candès, 2019), the MLE does not exist on the left of the curve. The lines mark the transition from linearly separable to non-linearly separable data, that depends on the data structure ($\Delta$ and $\rho$). The horizontal dashed line marks the threshold $\alpha^* = 2$ found by Cover (1965).

# 1.3 - The learning curves of multi-class teacher-student classification

Modern ML classification tasks most often involve multiple classes, e.g., 10 for classification on the MNIST (Deng, 2012) or CIFAR10 (Krizhevsky et al., 2010) datasets, or even 1000 for ImageNet (Deng et al., 2009). Multi-class classification is therefore ubiquitous in practical applications, yet theory most commonly focuses on binary classification models – such as the one considered in the previous chapter – that are more easily amenable to exact characterisation.

In this chapter, we consider the teacher-student perceptron – broadly studied as a high-dimensional model of binary classification since the seminal work by Gardner & Derrida (1989) – and we generalise it to encompass multi-class classification. The following presentation is based on Article 2.

## 1.3.1 . Introduction to the task

We consider a multi-class classification problem where the training data $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n \times d}$ are composed of $n$ $d-$dimensional i.i.d. standard Gaussian samples, where $x_{\mu j} \sim \mathcal{N}(0,1)$, $\forall j \in \{1, \dots, d\}$, $\forall \mu \in \{1, \dots, n\}$. The corresponding labels are $\boldsymbol{Y} = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_n)^\top \in \{0,1\}^{n \times k}$, each representing the one-hot[1] encoding of one of $k$ possible classes. In particular, we assume that the labels are generated by a *teacher* matrix $\boldsymbol{W}^* = (\boldsymbol{w}_1^*, \dots, \boldsymbol{w}_k^*) \in \mathbb{R}^{d \times k}$ as

$$y_{\mu l} = \begin{cases} 1 & \text{if} \quad l = \underset{h \in \{1,\dots,k\}}{\text{argmax}} \left( \boldsymbol{x}_\mu^\top \boldsymbol{w}_h^* \right) \\ 0 & \text{otherwise} \end{cases}, \qquad \forall \mu \in \{1, \dots, n\}. \tag{1.3.1}$$

In the following, we denote the output channel as $\phi_{\text{out}}(\boldsymbol{v}) := \mathrm{e}_{\underset{l \in [k]}{\text{argmax}}(\boldsymbol{v}_l)} \in \{0,1\}^k$, where $\mathrm{e}_h$ denotes the standard one-hot vector with the $h^{\text{th}}$ site equal to 1 and all other entries equal to zero. The teacher matrix $\boldsymbol{W}^*$ is drawn with i.i.d. entries either from a standard Gaussian $w_{il}^* \sim \mathcal{N}(0,1)$ or a Rademacher distribution $w_{il}^* = \pm 1$ with equal probability. Note that for $k = 2$ this problem corresponds to the well-studied teacher-student perceptron problem with binary labels (Gardner & Derrida, 1989; Engel & Van den Broeck, 2001).

Similarly as in Chapter 1.2, we are interested in the problem of learning the teacher-target function in the high-dimensional setting, where $n, d \to \infty$ at a fixed sample complexity $\alpha = n/d$, under two estimation procedures: empirical risk minimisation (ERM) and Bayes-optimal (BO) estimation.

**Prior dimensional reduction —** It is useful to notice that the above problem can be easily mapped from $k$ to $k-1$ dimensions. The intuition is exactly the same

[1] A one-hot vector has all entries equal to zero but one that is equal to one.

of the binary perceptron: the knowledge of $k-1$ components of the one-hot label representation $\boldsymbol{y}$ is enough to determine the remaining component by exclusion. Nevertheless, for $k>2$ shifting the weights in order to reproduce this structure introduces additional correlations that must be taken into account.

We recall that $\boldsymbol{W}^*$ is a $d \times k$ matrix, and denote by $\boldsymbol{w}^*_l$ , $1 \leq l \leq k$, its columns, each corresponding to a different class. Notice that the label $\boldsymbol{y} = \mathrm{e}_{\mathrm{argmax_l(\{w_l^{*\top} x\}_{l\in[k]})}}$ given by Eq. (1.3.1) of a data point $\boldsymbol{x}$ can be equivalently expressed by taking the $k^{\text{th}}-$component, i.e. $\boldsymbol{w}^{*\top}_k \boldsymbol{x}$, as a reference for comparison and setting

$$\tilde{\boldsymbol{w}}^*_h \leftarrow \boldsymbol{w}^*_h - \boldsymbol{w}^*_k \qquad \text{for all } 1 \leq h \leq k, \tag{1.3.2}$$

so that $\tilde{\boldsymbol{w}}^*_k = \boldsymbol{0}$, and the problem is reduced to $k-1$ dimensions. We then replace $\boldsymbol{W}^*$ by $\tilde{\boldsymbol{W}}^* \in \mathbb{R}^{d\times(k-1)}$. Denoting by $\boldsymbol{1}_k$ the $k$-dimensional vector with all entries equal to 1, we present schematically in Figure 1.3.1 the prior reduction. For simplicity, in the following we present all the expressions in the original $k-$dimensional space. However, we take into account the above mapping whenever we need to evaluate explicitly the prior term.

**Empirical risk minimisation** — In the first case, the statistician (or student) is given only the training data $(\boldsymbol{X}, \boldsymbol{Y})$ and has to learn the teacher weights $\boldsymbol{W^*}$ with a multi-class perceptron model $\hat{\boldsymbol{y}}(\boldsymbol{x}) = \phi_{\text{out}}\left(\boldsymbol{W}^\top \boldsymbol{x}\right)$ by ERM over the training set:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}\in\mathbb{R}^{d\times k}}{\text{argmin}} \left[\mathcal{H}(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y}) + \frac{\lambda}{2}\|\boldsymbol{W}\|_F^2\right], \tag{1.3.3}$$

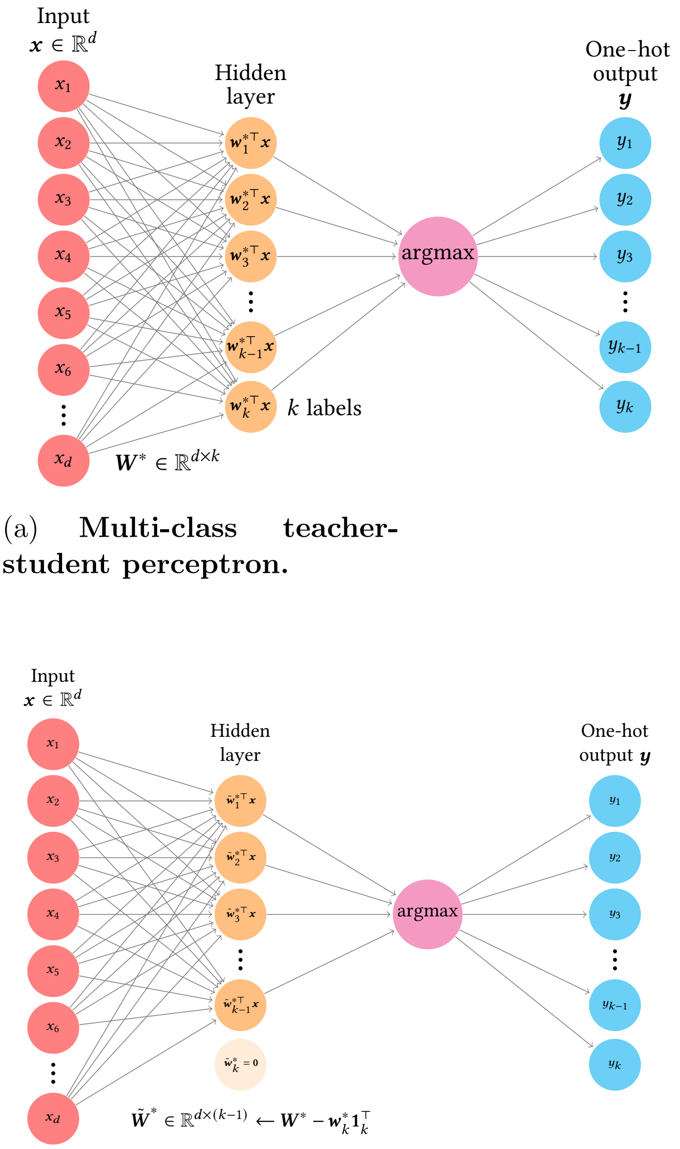
with $\mathcal{H}(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y}) = \sum_{\mu=1}^n \ell\left(\boldsymbol{W}^\top \boldsymbol{x}_\mu, \boldsymbol{y}_\mu\right)$ and ridge regularisation of strength $\lambda$, where $\|\cdot\|_F$ is the Frobenius norm. The loss function $\ell$ accounts for the performance of the weight vector $\boldsymbol{W}$ over a single training point. Two widely used loss functions for multi-class classification are the cross-entropy loss: $\ell(\boldsymbol{z},\boldsymbol{y}) = -\sum_{l=1}^k y_l \cdot \ln\left(e^{z_l}/\sum_{l=1}^k e^{z_l}\right)$ and the square loss: $\ell(\boldsymbol{z},\boldsymbol{y}) = (\boldsymbol{z}-\boldsymbol{y})^\top(\boldsymbol{z}-\boldsymbol{y})/2$.

**Bayes-optimal estimator** — In the BO setting, as explained in Chapters 1.1 and 1.2, the student has access not only to the training data but also on prior knowledge on the teacher weight distribution $\mathrm{P}_{\boldsymbol{W}^*}$ and on the model generating the inputs and the labels as in Eq. (1.3.1). In the teacher-student setting under consideration, where labels are generated by a noiseless channel, the BO estimator for the label $\boldsymbol{y}_{\text{new}}$ of a previously unseen data point $\boldsymbol{x}_{\text{new}}$ can be computed directly from the BO estimator $\hat{\boldsymbol{W}}_{\text{BO}}$ of the teacher weights as $\hat{\boldsymbol{y}}_{\text{new}} = \phi_{\text{out}}(\hat{\boldsymbol{W}}_{\text{BO}}^\top \boldsymbol{x}_{\text{new}})$. The matrix $\hat{\boldsymbol{W}}_{\text{BO}}$ is the minimiser of the mean-squared error with respect to the ground-truth $\boldsymbol{W^*}$, i.e.,

$$\hat{\boldsymbol{W}}_{\text{BO}} = \underset{\mathrm{W}}{\text{argmin}}\, \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{W}^*}\|\boldsymbol{W} - \boldsymbol{W}^*\|_F^2 = \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{W}^*,\boldsymbol{W}|(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{W}^*)}\left[\boldsymbol{W}\right]. \tag{1.3.4}$$

Note that computing explicitly the BO estimator requires computing the posterior distribution,

$$\mathrm{P}\left(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y}\right) = \frac{1}{Z_d}\prod_{l=1}^k P_{\boldsymbol{W}^*}\left(\boldsymbol{w}_l\right)\prod_{\mu=1}^n \delta\left(\boldsymbol{y}_\mu - \phi_{\text{out}}(\boldsymbol{W}^\top \boldsymbol{x}_\mu)\right). \tag{1.3.5}$$



(a) **Multi-class teacher-student perceptron.**

(b) **Prior reduction.**

Figure 1.3.1 – Schematic representation of the multi-class classification problem defined in Eq. (1.3.1). The knowledge of $k-1$ components of the one-hot label representation $\boldsymbol{y}$ is enough to determine the remaining component. Nevertheless for $k > 2$, shifting the weights in order to reproduce this structure introduces additional correlations that must be taken into account.

which is in general unfeasible in high dimensions. However, as we shall see, its performance can be characterised exactly in such limit. A key quantity in our derivation is the *free entropy density*:

$$\Phi = \lim_{d\to\infty} \frac{1}{d} \mathbb{E}_{\boldsymbol{X},\boldsymbol{W}^*} \ln Z_d, \tag{1.3.6}$$

where we remind that the partition function $Z_d$ is the normalisation of the posterior. In the BO setting, the free entropy density is closely related to the mutual information density between the labels and the weights, see Barbier et al. (2019) for an explicit discussion of this connection.

The generalisation performance of different optimisation strategies is measured via the misclassification rate (a.k.a. 0/1 error), as commonly done for classification (see also Chapter 1.2):

$$\varepsilon_{\text{gen}}(\alpha) = \mathbb{E}_{\boldsymbol{x}_{\text{new}},\boldsymbol{X},\boldsymbol{W}^*} \mathbb{1}\left[\hat{\boldsymbol{y}}\left(\hat{\boldsymbol{W}}(\alpha)\right) \neq \boldsymbol{y}_{\text{new}}\right], \tag{1.3.7}$$

where $\boldsymbol{x}_{\text{new}}$ is a previously unseen data point and $\boldsymbol{y}_{\text{new}}$ the corresponding label, generated by the teacher as in Eq. (1.3.1). Similarly, the estimator $\hat{\boldsymbol{y}}$ is generated by the weight matrix $\hat{\boldsymbol{W}}$, which in turn depends on the training set. We compare the performance obtained via ERM to the one of the BO estimator from Eq. (1.3.4). Note that Eq. (1.3.7) for the BO error can be written as

$$\begin{aligned}\varepsilon_{\text{gen}}^{\text{BO}} &= \frac{1}{2}\mathbb{E}_{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{x},\boldsymbol{W}^*}\|\phi_{\text{out}}(\boldsymbol{W}^{*\top}\boldsymbol{x}) - \phi_{\text{out}}(\langle\boldsymbol{W}^\top\boldsymbol{x}\rangle)\|_2^2 \\ &= 1 - \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y},\boldsymbol{x},\boldsymbol{W}^*}\left[\phi_{\text{out}}(\boldsymbol{W}^{*\top}\boldsymbol{x})^\top\phi_{\text{out}}(\hat{\boldsymbol{W}}_{\text{BO}}^\top\,\boldsymbol{x})\right],\end{aligned} \tag{1.3.8}$$

where for brevity $\boldsymbol{x} = \boldsymbol{x}_{\text{new}}$ and $\langle\cdot\rangle = \mathbb{E}_{\boldsymbol{W}|(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{W}^*)}$, and we have used that $\|\phi_{\text{out}}(\cdot)\|_2^2 \equiv 1$. Since the distribution of $\boldsymbol{x}_{\text{new}}$ is rotationally invariant, the averaged quantity $\mathbb{E}_{\boldsymbol{S},\boldsymbol{x}_{\text{new}},\boldsymbol{W}^*}\left[\,\phi_{\text{out}}(\boldsymbol{W}^{*\top}\,\boldsymbol{x}_{\text{new}})^\top\,\phi_{\text{out}}(\hat{\boldsymbol{W}}_{\text{BO}}^\top\,\,\boldsymbol{x}_{\text{new}})\right]$ only depends on the correlation between $\boldsymbol{W}^*$ and $\hat{\boldsymbol{W}}_{\text{BO}}$, which as we show later concentrates to the maximiser of the free entropy (1.3.6) in the high-dimensional limit.

Closely related settings, such as high-dimensional multi-class classification with Gaussian mixture data (Loureiro et al., 2021; Wang et al., 2021; Kini & Thrampoulidis, 2021; Thrampoulidis, 2020) were recently reported, while the generalisation to multi-class classification for the teacher-student setting is still missing. In the following we show how to fill this gap.

The main technical difficulty of analysing the teacher-student perceptron with $k > 2$ classes is that the corresponding closed-form formulas are given in terms of a set of coupled self-consistent equations on $(k-1)\times(k-1)$ dimensional *matrix variables* (the *order parameters* introduced in Chapter 1.1), involving $(k-1)$-dimensional integrals. This poses some challenges in both the mathematical proof and the numerical evaluation of their solution. We can overcome these difficulties by building on recent works with similar matrix structure, notably the committee machine (Aubin et al., 2019; Barbier, 2021) and the supervised $k$-cluster Gaussian mixture classification (Loureiro et al., 2021).

The heuristic replica method allows to derive a generic set of equations covering both the BO and the ERM cases. The rigorous proof for the BO case is given in Aubin et al. (2019); Barbier (2021) based on an interpolation argument. The ERM case, proven in Article 2, adds the difficulty of non Bayes optimality to the matrix valued problem. This prevents the use of both interpolation methods as in Aubin et al. (2019) or convex Gaussian comparison inequalities, see e.g. Thrampoulidis et al. (2018). Those difficulties are handled by employing a similar proof strategy as in Loureiro et al. (2021, 2022), which leverages on the rigorous analysis of matrix-valued approximate message passing (AMP) iterations. Although the planted model considered here is more elaborate than in Loureiro et al. (2021, 2022), the present problem is also amenable to a matrix-valued AMP iteration by decomposing the data matrix into two parts aligned and orthogonal to the subspace spanned by the columns of the teacher weights. We refer the reader to Article 2 for the details on the proof, that is not discussed in this manuscript.

## 1.3.2 . The learning curves via the replica method

Here we sketch the main points of the derivation of the learning curves via the replica method. We consider a general setting where the student has access to a prior distribution $P_w$ over the teacher weights and a model distribution $P_{\text{out}}$, which can be the true ones or not. This formulation encompasses both the BO and non BO settings. As we shall see in the following, ERM can be seen as a special case of the latter. The posterior distribution of the student weights is given by

$$P\left(\{\boldsymbol{w}_l\}_{l=1}^k|\boldsymbol{X},\boldsymbol{Y}\right) = \frac{1}{Z_d}\prod_{l=1}^k P_w(\boldsymbol{w}_l)\prod_{\mu=1}^n P_{\text{out}}(\boldsymbol{y}_\mu|\{h_{\mu l}\}_{l=1}^k) \tag{1.3.9}$$

where we have defined $h_{\mu l} = \boldsymbol{w}_l^\top \boldsymbol{x}_\mu/\sqrt{d}$. The partition function is then

$$Z_d = \int_{\mathbb{R}^{d\times k}} \mathrm{d}\boldsymbol{W}\prod_{l=1}^k P_w(\boldsymbol{w}_l)\prod_{\mu=1}^n P_{\text{out}}(\boldsymbol{y}_\mu|\{h_{\mu l}\}_{l=1}^k). \tag{1.3.10}$$

By using the *replica trick*, we can compute the free entropy in the high-dimensional limit as

$$\Phi := \lim_{d\to\infty}\Phi_d := \lim_{d\to\infty}\frac{1}{d}\mathbb{E}_{\boldsymbol{X},\boldsymbol{W}^*}\ln Z_d \approx \lim_{d\to\infty}\lim_{p\to 0}\frac{1}{d}\partial_p\mathbb{E}_{\boldsymbol{X},\boldsymbol{W}^*}Z_d^p. \tag{1.3.11}$$

We can then rewrite the average in Eq. (1.3.11) as

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{X},\boldsymbol{W}^*}Z_d^p &= \mathbb{E}_{\boldsymbol{X},\boldsymbol{W}^*}\left[\int_{\mathbb{R}^{d\times k}}\mathrm{d}\boldsymbol{W}\prod_{l=1}^k P_w(\boldsymbol{w}_l)\prod_{\mu=1}^n P_{\text{out}}(\boldsymbol{y}_\mu|\{h_{\mu l}\}_{l=1}^k)\right]^p && (1.3.12)\\
&= \mathbb{E}_{\boldsymbol{X},\boldsymbol{W}^*}\left[\prod_{a=1}^p\int_{\mathbb{R}^{d\times k}}\mathrm{d}\boldsymbol{W}^a\prod_{l=1}^k P_w(\boldsymbol{w}_l^a)\prod_{\mu=1}^n P_{\text{out}}\left(\boldsymbol{y}_\mu|\{h_{\mu l}^a\}_{l=1}^k\right)\right] && (1.3.13)\\
&= \mathbb{E}_{\boldsymbol{X}}\int_{\mathbb{R}^{n\times k}}\mathrm{d}\boldsymbol{Y}\prod_{a=0}^p\left[\int_{\mathbb{R}^{d\times k}}\mathrm{d}\boldsymbol{W}^a\prod_{l=1}^k P_w^a(\boldsymbol{w}_l^a)\prod_{\mu=1}^n P_{\text{out}}^a\left(\boldsymbol{y}_\mu|\{h_{\mu l}^a\}_{l=1}^k\right)\right], && (1.3.14)
\end{aligned}$$

where above we have renamed $\boldsymbol{W}^0 = \boldsymbol{W}^*$. In order to account for both the BO and non-BO cases, we keep the distinction between teacher and student distributions by adding an index $a$ to prior and model distributions. In what follows, $P_w^0 = P_{\boldsymbol{W}^*}$ and $P_{\text{out}}^0 = P_{\text{out}}^*$ refer to the teacher, while $P_w^{a>0} = P_w$ and $P_{\text{out}}^{a>0} = P_{\text{out}}$ to the student. Let us denote the covariance tensor of the $h_{\mu l}^a$ as

$$\mathbb{E}[h_{\mu l}^a h_{\nu l'}^b] = \delta_{\mu\nu}Q_{bl'}^{al}, \tag{1.3.15}$$

$$Q_{bl'}^{al} = \frac{1}{d}\sum_{i=1}^d w_{il}^a w_{il'}^b, \tag{1.3.16}$$

with $\boldsymbol{Q}_a^b \in \mathbb{R}^{k\times k}$. We can rewrite the above as

$$\begin{aligned}\mathbb{E}_{\boldsymbol{X},\boldsymbol{W}^*} Z_d^p &= \mathbb{E}_{\boldsymbol{X}} \int_{\mathbb{R}^{n\times k}} \mathrm{d}\boldsymbol{Y} \prod_{a=0}^{p} \left[ \int_{\mathbb{R}^{d\times k}} \mathrm{d}\boldsymbol{W}^a \prod_{l=1}^{k} P_w^a(\boldsymbol{w}_l^a) \prod_{\mu=1}^{n} P_{\text{out}}^a \left(\boldsymbol{y}_\mu | \{h_{\mu l}^a\}_{l=1}^k\right)\right] \\ &= \prod_{(a,l);(b,l)} \int_{\mathbb{R}} \mathrm{d}Q_{bl'}^{al}\; I_{\text{prior}}(\{Q_{bl'}^{al}\})\; I_{\text{channel}}(\{Q_{bl'}^{al}\}),\end{aligned} \tag{1.3.17}$$

where we have denoted

$$I_{\text{prior}}(\{Q_{bl'}^{al}\}) = \prod_{a=0}^{p} \int_{\mathbb{R}^{d\times k}} \mathrm{d}\boldsymbol{W}^a \left[\prod_{l=1}^{k} P_w^a(\boldsymbol{w}_l^a)\right] \prod_{(a,l);(b,l')} \delta\left(Q_{bl'}^{al} - \frac{1}{d}\sum_{i=1}^{d} w_{il}^a w_{il'}^b\right), \tag{1.3.18}$$

$$\begin{aligned}I_{\text{channel}}(\{Q_{bl'}^{al}\}) &= \int_{\mathbb{R}^{n\times K}} \mathrm{d}\boldsymbol{Y} \prod_{a=0}^{p} \int_{\mathbb{R}^{d\times k}} \mathrm{d}h^a \left[\prod_{a=0}^{p}\prod_{\mu=1}^{n} P_{\text{out}}^a(\boldsymbol{y}_\mu | h_\mu^a)\right] \\ &\times \exp\left(-\frac{n}{2}\ln\det\boldsymbol{Q} - \frac{nk(p+1)}{2}\ln 2\pi - \frac{1}{2}\sum_{\mu=1}^{n}\sum_{a,b}\sum_{l,l'} h_{\mu l}^a (Q^{-1})_{bl'}^{al} h_{\mu l'}^b\right),\end{aligned} \tag{1.3.19}$$

and we have introduced both the definitions of the overlaps $\{Q_{bl'}^{al}\}$ and the local fields $\{h_{\mu l}^a\}$. We can introduce the Fourier representation of the Dirac $\delta-$functions in the prior term $I_{\text{prior}}$ and rewrite

$$\mathbb{E}Z_d^p = \prod_{(a,l);(b,l')} \int_{\mathbb{R}^2} \frac{\mathrm{d}Q_{bl'}^{al}\, \mathrm{d}\hat{Q}_{bl'}^{al}}{2\pi} \exp\left(d\, H(\boldsymbol{Q},\hat{\boldsymbol{Q}})\right), \tag{1.3.20}$$

where we have defined

$$H(\boldsymbol{Q},\hat{\boldsymbol{Q}}) := \frac{1}{2}\sum_{a=0}^{p}\sum_{l,l'} Q_{al'}^{al}\hat{Q}_{al'}^{al} - \frac{1}{2}\sum_{a\neq b}\sum_{l,l'} Q_{bl'}^{al}\hat{Q}_{bl'}^{al} + \ln I(\{\hat{Q}_{bl'}^{al}\}) + \alpha \ln J(\{Q_{bl'}^{al}\}), \tag{1.3.21}$$

and the auxiliary functions:

$$I(\{\hat{Q}_{bl'}^{al}\}) = \prod_{a=0}^{p}\int_{\mathbb{R}^k} \mathrm{d}\boldsymbol{w}^a\, P_w^a(\boldsymbol{w}^a) \exp\left(-\frac{1}{2}\sum_{a=0}^{p}\sum_{l,l'} w_l^a \hat{Q}_{al'}^{al} w_{l'}^a + \frac{1}{2}\sum_{a\neq b}\sum_{l,l'} w_l^a \hat{Q}_{bl'}^{al} w_{l'}^b\right), \tag{1.3.22}$$

$$J(\{Q_{bl'}^{al}\}) = \int_{\mathbb{R}^k} \mathrm{d}\boldsymbol{y} \prod_{a=0}^{p}\int_{\mathbb{R}^k} \frac{\mathrm{d}\boldsymbol{h}^a}{(2\pi)^{k(p+1)/2}} \frac{P_{\text{out}}^a(\boldsymbol{y}|\boldsymbol{h}^a)}{\sqrt{\det\boldsymbol{Q}}} \exp\left(-\frac{1}{2}\sum_{a,b}\sum_{l,l'} h_l^a (Q^{-1})_{bl'}^{al} h_{l'}^b\right). \tag{1.3.23}$$

We observe that, upon exchanging the limits in $d$ and $p$, the high-dimensional limit of the free entropy can be computed via a saddle-point method:

$$\Phi = \lim_{d\to\infty} \mathbb{E}_{\boldsymbol{X},\boldsymbol{W}^*} \ln Z_d = \lim_{p\to 0^+} \operatorname{extr}_{\boldsymbol{Q},\hat{\boldsymbol{Q}}} \left[H(\boldsymbol{Q},\hat{\boldsymbol{Q}})\right]. \tag{1.3.24}$$

**Replica symmetric ansatz —** In order to progress in the calculation, we assume that the extremum in Eq. (1.3.24) is attained at $\{\boldsymbol{Q}, \hat{\boldsymbol{Q}}\}$ described by a replica symmetric (RS) ansatz. We distinguish between the BO and non-BO cases. Note that in the BO case we can drop the $a-$index from the prior and model distributions. In the BO setting we make the following ansatz:

$$Q^{al}_{al'} = Q^*_{ll'}, \qquad \hat{Q}^{al}_{al'} = \hat{Q}^*_{ll'}, \qquad \forall a = 0, ..p, \, \forall l, l' \leq k, \tag{1.3.25}$$

$$Q^{al}_{bl'} = q_{ll'}, \qquad \hat{Q}^{al}_{bl'} = \hat{q}_{ll'}, \qquad \forall a \neq b, \forall l, l' \leq k. \tag{1.3.26}$$

In the non BO setting we make the following ansatz:

$$Q^{al}_{al'} = Q^0_{ll'}, \qquad \hat{Q}^{al}_{al'} = \hat{Q}^0_{ll'}, \qquad \forall a = 1, ..p, \, \forall l, l' \leq K \tag{1.3.27}$$

$$Q^{al}_{bl'} = q_{ll'}, \qquad \hat{Q}^{al}_{bl'} = \hat{q}_{ll'}, \qquad \forall a \neq b, \; a, b = 1, ...p, \, \forall l, l' \leq k \tag{1.3.28}$$

$$Q^{0l}_{al'} = m_{ll'}, \qquad \hat{Q}^{0l}_{al'} = \hat{m}_{ll'}, \qquad \forall a = 1, ...p, \, \forall l, l' \leq k \tag{1.3.29}$$

$$Q^{0l}_{0l'} = Q^*_{ll'}, \qquad \hat{Q}^{0l}_{0l'} = \hat{Q}^*_{ll'}, \qquad \forall l, l' \leq k \tag{1.3.30}$$

We do not report the derivation of the update equations in the RS ansatz, that can be found in full detail in Article 2. Here, we only write the final expression of the saddle-point equations. In the BO setting the update equations are given by:

$$\begin{aligned} \boldsymbol{q} &= \mathbb{E}_{\boldsymbol{\xi}} \left[ \mathcal{Z}^*_w(\hat{\boldsymbol{q}}^{1/2}\boldsymbol{\xi}, \hat{\boldsymbol{q}}) \, \boldsymbol{f}^*_w(\hat{\boldsymbol{q}}^{1/2}\boldsymbol{\xi}, \hat{\boldsymbol{q}}) \, \boldsymbol{f}^*_w(\hat{\boldsymbol{q}}^{1/2}\boldsymbol{\xi}, \hat{\boldsymbol{q}})^\top \right], \\ \hat{\boldsymbol{q}} &= \alpha \mathbb{E}_{\boldsymbol{y},\boldsymbol{\xi}} \left[ \mathcal{Z}^*_{\text{out}}(\boldsymbol{y}; \boldsymbol{q}^{1/2}\boldsymbol{\xi}, \boldsymbol{Q}^* - \boldsymbol{q}) \, \boldsymbol{f}^*_{\text{out}}(\boldsymbol{y}; \boldsymbol{q}^{1/2}\boldsymbol{\xi}, \boldsymbol{Q}^* - \boldsymbol{q}) \, \boldsymbol{f}^*_{\text{out}}(\boldsymbol{y}; \boldsymbol{q}^{1/2}\boldsymbol{\xi}, \boldsymbol{Q}^* - \boldsymbol{q})^\top \right], \end{aligned} \tag{1.3.31}$$

where $\boldsymbol{\xi}$ denotes a $k-$dimensional standard Gaussian variable $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_k)$.

In the non-BO setting, we define for simplicity: $\boldsymbol{V} = \boldsymbol{Q}^0 - \boldsymbol{q}$, $\hat{\boldsymbol{V}} = \hat{\boldsymbol{Q}}^0 + \hat{\boldsymbol{q}}$, and we find

$$\begin{aligned} \boldsymbol{m} &= \mathbb{E}_{\boldsymbol{\xi}} \left[ \mathcal{Z}^*_w \times \boldsymbol{f}^*_w(\hat{\boldsymbol{m}}\hat{\boldsymbol{q}}^{-1/2}\boldsymbol{\xi}, \hat{\boldsymbol{m}}^T\hat{\boldsymbol{q}}^{-1}\hat{\boldsymbol{m}}) \, \boldsymbol{f}_w(\hat{\boldsymbol{q}}^{1/2}\boldsymbol{\xi}, \hat{\boldsymbol{V}})^\top \right], \\ \boldsymbol{q} &= \mathbb{E}_{\boldsymbol{\xi}} \left[ \mathcal{Z}^*_w(\hat{\boldsymbol{m}}\hat{\boldsymbol{q}}^{-1/2}\boldsymbol{\xi}, \hat{\boldsymbol{m}}^T\hat{\boldsymbol{q}}^{-1}\hat{\boldsymbol{m}}) \, \boldsymbol{f}_w(\hat{\boldsymbol{q}}^{1/2}\boldsymbol{\xi}, \hat{\boldsymbol{V}}) \boldsymbol{f}_w(\hat{\boldsymbol{q}}^{1/2}\boldsymbol{\xi}, \hat{\boldsymbol{V}})^\top \right], \\ \boldsymbol{V} &= \mathbb{E}_{\boldsymbol{\xi}} \left[ \mathcal{Z}^*_w(\hat{\boldsymbol{m}}\hat{\boldsymbol{q}}^{-1/2}\boldsymbol{\xi}, \hat{\boldsymbol{m}}^T\hat{\boldsymbol{q}}^{-1}\hat{\boldsymbol{m}}) \partial_\gamma \boldsymbol{f}_w(\hat{\boldsymbol{q}}^{1/2}\boldsymbol{\xi}, \hat{\boldsymbol{V}}) \right], \\ \hat{\boldsymbol{m}} &= \alpha \, \mathbb{E}_{\boldsymbol{y},\boldsymbol{\xi}} \left[ \mathcal{Z}^*_{\text{out}} \, \boldsymbol{f}^*_{\text{out}}(\boldsymbol{y}, \boldsymbol{m}\boldsymbol{q}^{-1/2}\boldsymbol{\xi}, \boldsymbol{Q}^* - \boldsymbol{m}^\top\boldsymbol{q}^{-1}\boldsymbol{m}) \, \boldsymbol{f}_{\text{out}}(\boldsymbol{y}, \boldsymbol{q}^{1/2}\boldsymbol{\xi}, \boldsymbol{V})^\top \right], \\ \hat{\boldsymbol{q}} &= \alpha \, \mathbb{E}_{\boldsymbol{y},\boldsymbol{\xi}} \left[ \mathcal{Z}^*_{\text{out}}(\boldsymbol{y}, \boldsymbol{m}\boldsymbol{q}^{-1/2}\boldsymbol{\xi}, \boldsymbol{Q}^* - \boldsymbol{m}^\top\boldsymbol{q}^{-1}\boldsymbol{m}) \, \boldsymbol{f}_{\text{out}}(\boldsymbol{y}, \boldsymbol{q}^{1/2}\boldsymbol{\xi}, \boldsymbol{V}) \, \boldsymbol{f}_{\text{out}}(\boldsymbol{y}, \boldsymbol{q}^{1/2}\boldsymbol{\xi}, \boldsymbol{V})^\top \right], \\ \hat{\boldsymbol{V}} &= -\alpha \, \mathbb{E}_{\boldsymbol{y},\boldsymbol{\xi}} \left[ \mathcal{Z}^*_{\text{out}}(\boldsymbol{y}, \boldsymbol{m}\boldsymbol{q}^{-1/2}\boldsymbol{\xi}, \boldsymbol{Q}^* - \boldsymbol{m}^\top\boldsymbol{q}^{-1}\boldsymbol{m}) \partial_w \boldsymbol{f}_{\text{out}}(\boldsymbol{y}, \boldsymbol{q}^{1/2}\boldsymbol{\xi}, \boldsymbol{V}) \right], \end{aligned} \tag{1.3.32}$$

where in both settings we have made use of the following definitions. For $\boldsymbol{w} \in \mathbb{R}^k$, let

$$Q_w(\boldsymbol{w}; \boldsymbol{\gamma}, \boldsymbol{\Lambda}) \equiv \frac{P_w(\boldsymbol{w})}{\mathcal{Z}_w(\boldsymbol{\gamma}, \boldsymbol{\Lambda})} e^{-\frac{1}{2}\boldsymbol{w}^\top\boldsymbol{\Lambda}\boldsymbol{w} + \boldsymbol{\gamma}^\top\boldsymbol{w}}, \tag{1.3.33a}$$

with

$$\boldsymbol{f}_w(\boldsymbol{\gamma}, \boldsymbol{\Lambda}) \equiv \partial_{\boldsymbol{\gamma}} \log \mathcal{Z}_w(\boldsymbol{\gamma}, \boldsymbol{\Lambda}) = \mathbb{E}_{Q_w}[\boldsymbol{w}], \tag{1.3.33b}$$

and for $\boldsymbol{z} \in \mathbb{R}^k$, let

$$Q_{\text{out}}(\boldsymbol{z}; \boldsymbol{y}, \boldsymbol{\omega}, \boldsymbol{V}) \equiv \frac{P_{\text{out}}(\boldsymbol{y}|\boldsymbol{w})}{\mathcal{Z}_{\text{out}}(\boldsymbol{y}, \boldsymbol{\omega}, \boldsymbol{V})} \frac{e^{-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\omega})^\top \boldsymbol{V}^{-1}(\boldsymbol{z}-\boldsymbol{\omega})}}{\sqrt{\det(2\pi \boldsymbol{V})}} , \tag{1.3.33c}$$

with

$$\boldsymbol{f}_{\text{out}}(\boldsymbol{y}, \boldsymbol{\omega}, \boldsymbol{V}) \equiv \partial_{\boldsymbol{\omega}} \log \mathcal{Z}_{\text{out}}(\boldsymbol{y}, \boldsymbol{\omega}, \boldsymbol{V}) = \boldsymbol{V}^{-1} \mathbb{E}_{Q_{\text{out}}}[\boldsymbol{z} - \boldsymbol{\omega}] , \tag{1.3.33d}$$

where the definitions of $\boldsymbol{f}_w^*, \boldsymbol{f}_{\text{out}}^*$ are identical, provided that $P_w, P_{\text{out}}$ are replaced by $P_{\boldsymbol{W}^*}, P_{\text{out}}^*$. The functions $\mathcal{Z}_w$ and $\mathcal{Z}_{\text{out}}$ are given by

$$\mathcal{Z}_w(\boldsymbol{\gamma}, \boldsymbol{\Lambda}) = \int_{\mathbb{R}^k} \mathrm{d}\boldsymbol{w} P_w(\boldsymbol{w}) e^{-\frac{1}{2}\boldsymbol{w}^\top \boldsymbol{\Lambda} \boldsymbol{w} + \boldsymbol{\gamma}^\top \boldsymbol{w}} , \tag{1.3.34a}$$

$$\mathcal{Z}_{\text{out}}(\boldsymbol{y}; \boldsymbol{\omega}, \boldsymbol{V}) = \int_{\mathbb{R}^k} d\boldsymbol{z} \frac{e^{-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\omega})^\top \boldsymbol{V}^{-1}(\boldsymbol{z}-\boldsymbol{\omega})}}{\sqrt{\det(2\pi \boldsymbol{V})}} P_{\text{out}}(\boldsymbol{y}|\boldsymbol{z}) , \tag{1.3.34b}$$

and $\mathcal{Z}_w^*, \mathcal{Z}_{\text{out}}^*$ are defined in the exact same way provided that the student distributions $P_w, P_{\text{out}}$ are replaced by the teacher distributions $P_w^*, P_{\text{out}}^*$. The explicit expressions of the auxiliary functions depend on the choice of the teacher and student distributions. For the evaluation of the expressions in the special cases under consideration, see Appendix C of Article 2.

**Generalisation error** — In the high-dimensional limit the asymptotic generalisation error associated to the ERM estimator (1.3.3) can be expressed only as a function of the parameters $(\boldsymbol{m}, \boldsymbol{q})$ obtained by solving the self-consistent equations (1.3.32):

$$\varepsilon_{\text{gen}} = \mathrm{P}_{(\boldsymbol{\nu}, \boldsymbol{\mu}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})} \left( \phi_{\text{out}}(\boldsymbol{\mu}) \neq \phi_{\text{out}}(\boldsymbol{\nu}) \right) , \tag{1.3.35}$$

where $\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{Q}^* & \boldsymbol{m} \\ \boldsymbol{m} & \boldsymbol{q} \end{bmatrix}$. As one can expect from the discussion in the previous section, the BO error is obtained by a similar, but simpler expression depending only on the overlap $\boldsymbol{q}$, given by Eqs. (1.3.31):

$$\varepsilon_{\text{gen}} = \mathrm{P}_{\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_k)} \left( \phi_{\text{out}}(\boldsymbol{q}^{1/2} \boldsymbol{\xi}) \neq \phi_{\text{out}}(\boldsymbol{Q}^{*1/2} \boldsymbol{\xi}) \right) . \tag{1.3.36}$$

## 1.3.3 . The Approximate Message Passing algorithm

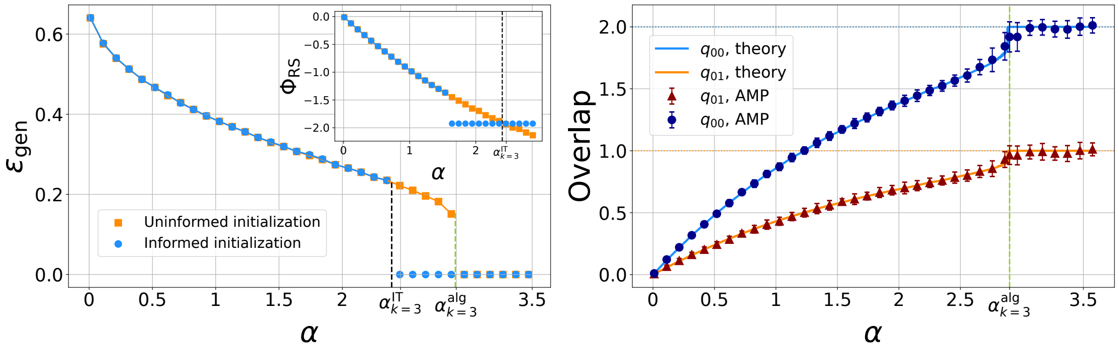
In order to illustrate our theoretical results for the performance of the BO (Eq. (1.3.4)) and ERM (Eq. (1.3.3)) estimators, we would like to compare our asymptotic expressions for the generalisation error with simulations. On one hand, the regularised empirical risk defined in Eq. (1.3.3) is strongly convex, and therefore it can be readily minimised with any descent-based algorithm such as gradient descent or stochastic gradient descent. Indeed, in the ERM simulations that follow we employ out-of-the-box multi-class solvers from `scikit-learn` (Pedregosa et al., 2011) to assess our theoretical result from Eqs. (1.3.32). On the other hand, explicitly computing the BO estimator requires sampling from the posterior, an operation which is prohibitively costly in high-dimensions. Instead, we employ an *Approximate Message Passing* (AMP) algorithm to efficiently approximate the posterior marginals. AMP has several interesting properties which make it a popular tool in the study of random problems. First, it is proven to be optimal among a class of random estimation problems by Celentano et al. (2020), and for this reason it is widely used as a benchmark to assess algorithmic complexity. Second, it admits a set of scalar *state evolution equations* allowing to track its performance in high-dimensions (Javanmard & Montanari, 2013).

For the BO estimation problem considered here, AMP follows the well-known AMP algorithm for generalised linear estimation Donoho et al. (2009); Rangan (2011), which takes advantage of the high-dimensional limit $d \to \infty$ by approximating the posterior distribution in Eq. (1.3.5) by a multivariate Gaussian distribution through a belief propagation procedure expanded in powers of $d^{-1}$. The difference is that the estimators $\hat{\boldsymbol{w}}_j$ are $k$-dimensional vectors and their variances $\hat{\boldsymbol{C}}_j$ are $k \times k$ dimensional matrices, with $j = 1, \ldots, d$. The channel and prior update functions, $\boldsymbol{f}_{\text{out}}$ and $\boldsymbol{f}_{\boldsymbol{w}}$, respectively, are defined in the previous chapter. For a detailed derivation of the algorithm, see Article 2 and Aubin (2020).

Several versions of this $k$-fold AMP and the associated state evolution appeared in previous works, e.g., Aubin et al. (2019). It can be shown that the state evolution equations associated to the AMP algorithm for BO estimation coincide exactly with the self-consistent Eqs. (1.3.31) presented in the previous chapter starting from an *uninformed initialisation* $\boldsymbol{q}_0 \approx \boldsymbol{0}$ Aubin et al. (2019). This interesting property implies that when the extremisation problem in Eq. (1.3.24) has only one extremiser, AMP provides an exact approximation to the BO estimator in the high-dimensional limit. Instead, when there are more than one maxima in Eq. (1.3.24), AMP converges to an estimator with overlap $\boldsymbol{q}$ closest to the uninformed initial condition. If this is not the global maximum, this corresponds to a situation where AMP differs from the BO estimator. Since AMP provides a bound on the performance of first-order algorithms, this situation is an example of an *algorithmic hard phase*, where it is conjectured that the statistical optimal performance cannot be achieved by algorithms running in time $\sim O(d^2)$. We have implemented the AMP algorithm for $k = 3$ classes using the mapping presented above, which makes the estimators $(k-1)$-dimensional vectors and their variances $(k-1)\times(k-1)$ dimensional matrices. For more details on the algorithmic implementation, see Article 2.

## 1.3.4 . The results for $k = 3$ classes

In this section we discuss the consequences of our theoretical results for the particular case of $k = 3$ classes and compare them with numerical simulations. We investigate the dependence of the generalisation error on the sample complexity $\alpha$. First, we consider the case of Rademacher teacher weights and show that a first-order phase transition arises in the BO performance. Then, we turn to the case of Gaussian teacher weights and explore the role of the regularisation strength $\lambda$ in approaching the BO performance with ERM.



(a) Generalisation error $\varepsilon_{\rm gen}$ as a function of the sample complexity $\alpha$ evaluated via Eqs. (1.3.31). The orange points mark the error that would be asymptotically reached by the randomly initialised AMP. The blue points mark the BO error. The inset depicts the corresponding free entropies as a function of $\alpha$, their crossing locating the information-theoretic transition to perfect generalisation at $\alpha^{\rm IT}_{k=3} \approx 2.45$. AMP reaches perfect generalisation starting from $\alpha^{\rm alg}_{k=3} \approx 2.89$.

(b) Diagonal ($q_{00}$) and anti-diagonal ($q_{01}$) entries of the self-overlap matrix as a function of $\alpha$ in the BO setting. The full lines mark the fixed points of Eqs. (1.3.31), the symbols represent the result obtained by the AMP algorithm described in Section 1.3.3 averaged over 20 runs.

Figure 1.3.2 – **AMP for Rademacher teacher priot with $k = 3$ classes.**

**Bayes-optimal performance for Rademacher teacher** — The main difference between Gaussian and Rademacher teacher is that in the second case perfect generalisation is achievable at finite sample complexity, in line with the results known for the two-classes case of Györgyi (1990); Sompolinsky et al. (1990); Seung et al. (1992b). To compute the optimal information-theoretical performance, we have evaluated the global extremum of the replica free entropy. To this end, we have run the replica saddle-point iterations Eqs. (1.3.31) with both uninformed and informed initialisations and computed the free entropy in Eq. (1.3.24) of the fixed points (if distinct) reached by the two initialisations. In Figure 1.3.2 we report the generalisation error corresponding to the fixed points reached by the two initialisations, along with their corresponding free entropy in the inset. We found that indeed, for Rademacher teacher weights, the generalisation error decreases continuously for $\alpha \leq \alpha^{\rm IT}_{k=3} \approx 2.45$, and then jumps to zero for all $\alpha > \alpha^{\rm IT}_{k=3}$. From a statistical physics perspective, this discontinuous transition in the error corresponds to a *first-order phase transition* associated to the discontinuous appearance of a second extremum associated to perfect learning in the free energy potential.

The state evolution of the AMP algorithm is equivalent to gradient descent on the free energy potential (Eq. (1.3.24)) starting from an uninformed random initialisation. Therefore, the appearance of a second extremum away from zero implies that AMP is not able to achieve the BO statistical performance. Since

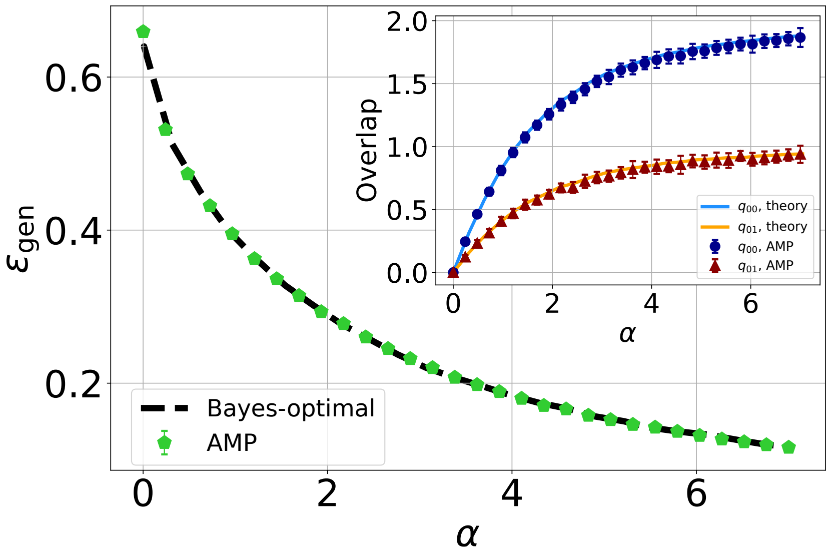
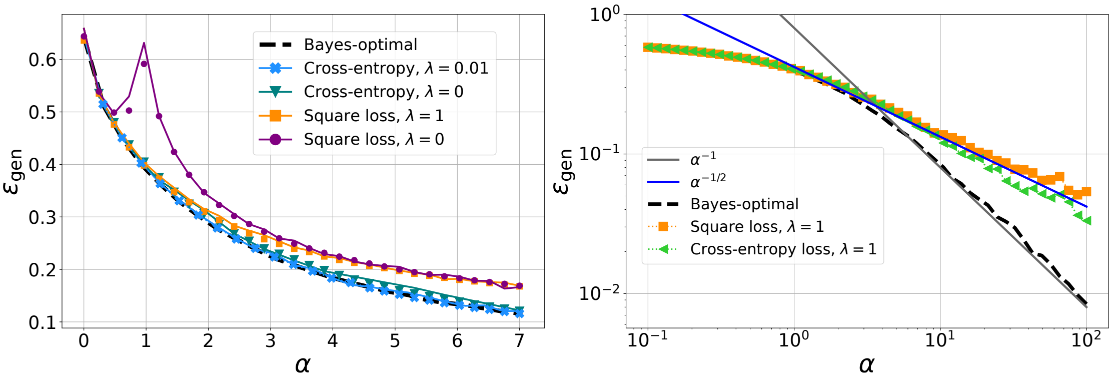


Figure 1.3.3 – **AMP for Gaussian teacher prior with $k = 3$ classes:** Generalisation error $\varepsilon_{\mathrm{gen}}$ as a function of the sample complexity $\alpha$. The performance of AMP (averaged over 20 runs), computed from Eq. (1.3.36), is marked by the green symbols (error bars are smaller than the symbols). The dashed black line indicates the BO error. The inset displays the diagonal ($q_{00}$) and anti-diagonal ($q_{01}$) entries of the self-overlap matrix as a function of $\alpha$ in the BO setting. The full lines mark the fixed points of Eqs. (1.3.31), while the symbols represent the result obtained from the AMP algorithm described in Section 1.3.3 averaged over 20 runs.

AMP is conjectured to be optimal among first-order methods (Celentano et al., 2020), this result is an example of a fundamental *statistical-to-algorithmic gap* in this problem. For $\alpha > \alpha_{k=3}^{\mathrm{alg}} \approx 2.89$, we observe that the uninformed minimum disappears, and we can check that this coincides with the sample complexity at which AMP is able to achieve zero generalisation error from random initialisation. This marks the algorithmic threshold, i.e., the sample complexity beyond which perfect generalisation is reachable algorithmically efficiently.

Our findings thus suggest the existence of an algorithmic *hard phase* for $\alpha_{k=3}^{\mathrm{IT}} < \alpha < \alpha_{k=3}^{\mathrm{alg}}$, where the theoretically optimal performance is not reachable by efficient algorithms. We note here the comparison with the canonical perceptron with Rademacher teacher weights and two classes, where the same thresholds are well known to be $\alpha_{k=2}^{\mathrm{IT}} = 1.249$, $\alpha_{k=2}^{\mathrm{algo}} = 1.493$ (Györgyi, 1990; Sompolinsky et al., 1990; Barbier et al., 2019). Naturally, these values are roughly twice smaller than the ones for $k = 3$ since for $k$ classes the teacher has $k - 1$ independent $d$-dimensional binary elements that need to be recovered in order to reach perfect generalisation. Comparing more precisely the values for $k = 3$ and also their difference, all are slightly smaller than the double of the values for $k = 2$.

**Bayes-optimal performance for Gaussian teacher —** Figures 1.3.3, 1.3.4 and 1.3.5 summarise our results for the case of Gaussian teacher weights. The BO error,



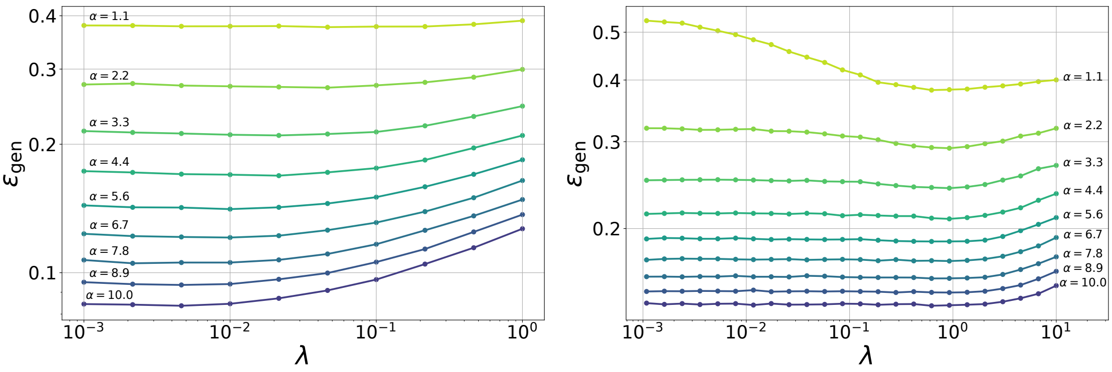
(a) Generalisation error $\varepsilon_{\rm gen}$ as a function of the sample complexity $\alpha$. The black dashed line depicts the BO performance. Full lines mark the performance of ERM with cross-entropy (blue) and square loss (orange), each computed at optimised ridge regularisation ($\lambda = 0.01$ and $\lambda = 1$ respectively, see Figure 1.3.5) from the fixed points of Eqs. (1.3.3). The symbols mark the results from numerical simulations at dimension $d = 1000$, averaged over 250 seeds. We also plot the performance of simulations at zero regularisation and theory at $\lambda \to 0^+$, for both cross-entropy (dark green) and square loss (purple).

(b) **Large$-\alpha$ behaviour.** Generalisation error as a function of $\alpha$. We plot our theoretical predictions at large $\alpha$, in log-log scale for visibility purposes. The black dashed line marks the BO error, the symbols mark the error of ERM at fixed regularisation $\lambda = 1$.

Figure 1.3.4 – **BO and ERM performances for Gaussian teacher weights.**

computed from Eq. (1.3.8), is depicted by the dashed black line in both figures and is a smooth, monotonically-decreasing function of the sample complexity $\alpha$. Interestingly, for Gaussian teacher weights, the BO-AMP algorithm – described in Section 1.3.3 and marked by the green symbols in Figure 1.3.3 – achieves the BO performance. This is highly non-trivial: computing the BO estimator usually requires sampling from the posterior distribution of the weights given the data, and therefore can be prohibitively costly in the high-dimensional regime considered here. For Gaussian weights AMP provides an exact approximation of the posterior marginals in quadratic time in the input dimension.

**Approaching Bayes-optimality with ERM —** Instead, how does ERM compare to the BO estimator? Note that the empirical risk in Eq. (1.3.3) is convex, and therefore, at variance with the posterior estimation, this problem can be readily simulated using descent-based algorithms such as stochastic gradient descent. The generalisation error obtained by ERM is plotted in Figure 1.3.4 as a function of the sample complexity. The full lines depict our theoretical predictions for the learning curves while the symbols mark the results from numerical simulations performed



(a) **Cross-entropy loss:** Generalisation error $\varepsilon_{\text{gen}}$ as a function of the regularisation strength $\lambda$, at fixed sample complexity $\alpha$. Different values of $\alpha$ are depicted with different colours. The curves are the result of numerical simulations performed at dimension $d = 1000$, averaged over 250 instances. We conclude that for these values of $\alpha$ the optimal $\lambda$ is close to 0.01.

(b) **Square loss:** Generalisation error $\varepsilon_{\text{gen}}$ as a function of the regularisation strength $\lambda$, at $\alpha$. Different values of $\alpha$ are depicted with different colours. The curves are the result of numerical simulations performed at dimension $d = 1000$, averaged over 250 seeds. We conclude that for these values of $\alpha$ the optimal $\lambda$ is close to 1.

Figure 1.3.5 – **The role of regularisation in ERM for $k = 3$ classes.**

at finite dimension $d = 1000$. We find excellent agreement between the two. For both cross-entropy and square losses, we show the performance achieved without regularisation ($\lambda = 0$) and with naively-optimised $\lambda$, obtained by cross-validation, in Figure 1.3.5. Interestingly, we find that the optimally-regularised cross-entropy loss achieves a close-to-optimal performance, while the square loss maintains a finite gap with respect to the BO error even at fine-tuned regularisation strength. Similar results were obtained for the two-classes teacher student perceptron Aubin et al. (2020). The fact that regularised cross-entropy minimisation is so close to optimal also in multi-class classification is remarkable and the generality of this finding is worth further investigation.

**Large–$\alpha$ behaviour —** Figure 1.3.4b considers again a Gaussian teacher prior and explores the behaviour of the generalisation error at large sample complexity. The BO performance is depicted in black and decays as $1/\alpha$ in the large$-\alpha$ regime. On the other hand, the performance obtained by ERM at fixed $\lambda$ displays a slower decay $\alpha^{-1/2}$. This is again compatible with the behaviour observed in the two-classes case Aubin et al. (2020). It remains to be analysed whether for $k > 2$ the optimally regularised ERM achieves the $1/\alpha$ rate as it does for the two classes.

**The role of regularisation —** Figure 1.3.5 further illustrates the role played by ridge regularisation. We plot the generalisation error as a function of the regularisation strength $\lambda$ at fixed sample complexity $\alpha$ for the cross-entropy (1.3.5a) and the square loss (1.3.5b). Different curves represent different values of sample complexity. We observe that the optimal regularisation depends only very mildly on the sample complexity $\alpha$ for this range of values of $\alpha$.

# Article 1

## The role of regularization in classification of high-dimensional noisy Gaussian mixture

Francesca Mignacco, Florent Krzakala, Yue M. Lu, and Lenka Zdeborová.

International Conference on Machine Learning, PMLR, 2020. p. 6874-6883.

☞ Proceedings of the 37th International Conference on Machine Learning, PMLR 119:6874-6883
☞ ArXiv preprint: *arXiv:2002.11544*

### Abstract

We consider a high-dimensional mixture of two Gaussians in the noisy regime where even an oracle knowing the centers of the clusters misclassifies a small but finite fraction of the points. We provide a rigorous analysis of the generalization error of regularized convex classifiers, including ridge, hinge and logistic regression, in the high-dimensional limit where the number $n$ of samples and their dimension $d$ go to infinity while their ratio is fixed to $\alpha = n/d$. We discuss surprising effects of the regularization that in some cases allows to reach the Bayes-optimal performances. We also illustrate the interpolation peak at low regularization, and analyze the role of the respective sizes of the two clusters.

# Article 2

## Learning curves for the multi-class teacher-student perceptron

Elisabetta Cornacchia, Francesca Mignacco, Rodrigo Veiga, Cédric Gerbelot,
Bruno Loureiro, Lenka Zdeborová.

☞ ArXiv preprint *arXiv:2203.12094*

**Abstract**

One of the most classical results in high-dimensional learning theory provides a closed-form expression for the generalisation error of binary classification with the single-layer teacher-student perceptron on i.i.d. Gaussian inputs. Both Bayes-optimal estimation and empirical risk minimisation (ERM) were extensively analysed for this setting. At the same time, a considerable part of modern machine learning practice concerns multi-class classification. Yet, an analogous analysis for the corresponding multi-class teacher-student perceptron was missing. In this manuscript we fill this gap by deriving and evaluating asymptotic expressions for both the Bayes-optimal and ERM generalisation errors in the high-dimensional regime. For Gaussian teacher weights, we investigate the performance of ERM with both cross-entropy and square losses, and explore the role of ridge regularisation in approaching Bayes-optimality. In particular, we observe that regularised cross-entropy minimisation yields close-to-optimal accuracy. Instead, for a binary teacher we show that a first-order phase transition arises in the Bayes-optimal performance.

# 2 - The dynamics of learning problems

# 2.1 - A brief introduction to the dynamics of learning

In this chapter, we introduce some useful methods and concepts to investigate the dynamical properties of learning algorithms. We remind that the goal of the algorithm is to search for a solution $\hat{\boldsymbol{W}}$ that minimises the empirical risk $\mathcal{H}$, given the data $\boldsymbol{X}$ and the corresponding labels $\boldsymbol{Y}$. This is the empirical risk minimisation (ERM) problem introduced in *Motivation and background* and Chapter 1.1:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmin}}\, \mathcal{H}\left(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}\right) = \underset{\boldsymbol{W}}{\operatorname{argmin}} \sum_{\mu=1}^{n} \ell\left(\hat{\boldsymbol{y}}_{\boldsymbol{W}}\left(\boldsymbol{x}_\mu\right), \boldsymbol{y}_\mu\right) + \lambda\, \Omega\left(\boldsymbol{W}\right), \tag{2.1.1}$$

where $\ell(\cdot)$ is a loss function accounting for the per-sample error and $\Omega$ is an explicit regularisation function weighted by the hyperparameter $\lambda \geq 0$.

The simplest way to attack the minimisation in Eq. (2.1.1) is to employ a general-purpose algorithm such as full-batch gradient descent (GD):

$$\boldsymbol{W}^{(0)} \sim \mathrm{P}_0\,, \qquad \boldsymbol{W}^{(t+\mathrm{d}t)} \leftarrow \boldsymbol{W}^{(t)} - \mathrm{d}t\, \nabla_{\boldsymbol{W}} \mathcal{H}\left(\boldsymbol{W}^{(t)}|\boldsymbol{X},\boldsymbol{y}\right), \tag{2.1.2}$$

where d$t$ is the time-step or learning rate and $\boldsymbol{W}^{(t)}$ marks the realisation of the weights at time $t$. However, calculating the gradient of the loss function brings a great computational burden, since it requires the evaluation of the current state of the weights on the full training set.

An efficient alternative emerged with the introduction of the stochastic gradient descent (SGD) algorithm (Robbins & Monro, 1951; Bottou, 1999, 2010). SGD, at variance with GD, approximates the gradient by evaluating it only on a *mini batch* – a small subset of the training set – which is changed at each step of the dynamics:

$$\boldsymbol{W}^{(0)} \sim \mathrm{P}_0\,, \qquad \boldsymbol{W}^{(t+\mathrm{d}t)} \leftarrow \boldsymbol{W}^{(t)} - \mathrm{d}t\, \tilde{\nabla}^{B}_{\boldsymbol{W}} \mathcal{H}\left(\boldsymbol{W}^{(t)}|\boldsymbol{X},\boldsymbol{y}\right), \tag{2.1.3}$$

where

$$\tilde{\nabla}^{B}_{\boldsymbol{W}} \mathcal{H}\left(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{y}\right) = \sum_{\mu\in\mathcal{B}} \nabla_{\boldsymbol{W}} \ell\left(\hat{\boldsymbol{y}}_{\boldsymbol{W}}\left(\boldsymbol{x}_\mu\right), \boldsymbol{y}_\mu\right) + \lambda\, \nabla_{\boldsymbol{W}} \Omega\left(\boldsymbol{W}\right) \tag{2.1.4}$$

indicates the approximated gradient computed on the (time-dependent) mini batch $\mathcal{B} \subseteq \{1,\dots,n\}$ of cardinality $B = |\mathcal{B}|$. The usual practical procedure is to partition the dataset into mini batches, that are parsed one by one until all have been used. At this point one training *epoch* has passed, the samples are shuffled and the procedure is repeated.

Quite surprisingly, in practical applications simple local optimisation methods – variants of the SGD algorithm – are able to find near-optimal solutions despite the non-convexity of the loss landscape and the curse of dimensionality. Indeed, over-parametrised neural networks trained by SGD, that can perfectly fit even random data, do not incur in over-fitting on real data. Instead, they can achieve excellent performances on previously unseen data (Zhang et al., 2017). Therefore, understanding how SGD can navigate so efficiently the high-dimensional non-convex loss landscape is one of the central problems in ML theory.

**Implicit regularisation and the role of initialisation** — A popular attempt to explain the success of SGD consists in showing that the loss landscape itself is simple, without spurious local minima, i.e., configurations that minimise the empirical risk but perform poorly on the test set. However, some empirical evidence instead leads to the conclusion that the loss landscape of state-of-the-art DNNs actually has spurious local (or even global) minima and SGD is able to find them with ad hoc initialisations (Safran & Shamir, 2017; Liu et al., 2019). Still, the SGD algorithm, initialised at random and with little use of explicit regularisation, leads to good generalisation properties in practice, a phenomenon commonly referred to as *implicit bias* or *implicit regularisation*.

Theoretical guarantees on this phenomenon have been derived in the case of separable data in a linear setup both for GD (Soudry et al., 2018a) and SGD (Nacson et al., 2019) depending on the loss function. In the case of square loss, both algorithms converge to the global solution that is closer to the initialisation in squared distance. In the case of logistic loss – and all strictly decreasing loss functions with exponential tail – the dynamics is biased towards the max-margin classifier regardless of the initialisation. The study of implicit bias in GD has been recently extended to multiplicative parametrisations (Gunasekar et al., 2018), deep linear networks (Ji & Telgarsky, 2019) and homogeneous networks (Lyu & Li, 2019). Moreover, Woodworth et al. (2020) show that the crossover between the "lazy learning" regime of *neural tangent kernel* (Jacot et al., 2018), where the features do not change, and the *feature learning* regime can be tuned by the scale of initialisation.

In summary, while it has commonly been observed that SGD outperforms GD in practical applications (Keskar et al., 2017; He et al., 2019), theoretical results in support of this claim remain sparse (Abbe & Sandon, 2020; HaoChen et al., 2020) and the machine learning community is actively working to bridge this gap. In the next section, we provide a non-exhaustive list of some relevant alternative methods to study the dynamics of gradient-based optimisation methods.

## 2.1.1 . The theory of gradient-based learning algorithms

A recent stream of works is aiming at characterising the nature of the noise introduced by SGD and identifying its properties in order to understand how they correlate with the final generalisation performance. In other words, the purpose of this line of research is to investigate the implicit bias introduced by the most commonly adopted gradient-based algorithms to the training of DNNs. The nature of SGD noise is hard to grasp, due to its complicated structure resulting from the interplay of the architecture, the data distribution, the loss and the mini-batch sampling procedure. We briefly list some of the most common alternative approaches to tackle the problem.

**Langevin-like approximations** — A very active research direction models SGD as a discretisation of a stochastic differential equation, i.e., via the Langevin-like dynamics

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{W}^{(t)}}{\mathrm{d}t} &= -\nabla_{\boldsymbol{W}}\mathcal{H}\left(\boldsymbol{W}^{(t)}|\boldsymbol{X},\boldsymbol{y}\right)+\xi(t),\\ \xi(t) &= \left(\nabla_{\boldsymbol{W}}-\tilde{\nabla}^{B}_{\boldsymbol{W}}\right)\mathcal{H}\left(\boldsymbol{W}^{(t)}|\boldsymbol{X},\boldsymbol{y}\right),\end{aligned} \tag{2.1.5}$$

where the noise $\xi(t)$ has zero mean and is assumed to be Gaussian by invoking the central-limit theorem (CLT). This type of analysis has been adopted by several works (Hu et al., 2019; Li et al., 2017; Mandt et al., 2017; Chaudhari & Soatto, 2018; Cheng et al., 2020), with some differences in how to model the structure of the noise, the finite time step and the batch size. The importance of modeling anisotropic noise was highlighted in Zhu et al. (2019), while Jastrzebski et al. (2017) stress that the *ratio* between learning rate and batch size is an important quantity controlling the dynamics. The interplay between SGD noise and the loss curvature has been studied in Wei & Schwab (2019); Thomas et al. (2020). The authors of Pesme et al. (2021) show that better generalisation properties of SGD in diagonal linear networks are related to slower convergence.

However, this approach leads to a stochastic differential equation involving a hybrid, ill-defined continuous-time limit (Yaida, 2018), where the learning rate is sent to zero $\mathrm{d}t \to 0^+$ in the dynamics, but at the same time it is kept finite in the noise variance.

**$\alpha$-stable Lévy process description** — The validity of the CLT in this context has been questioned in Li et al. (2021) and challenged by a set of experiments (Simsekli et al., 2019; Şimşekli et al., 2019; Martin & Mahoney, 2019) suggesting that in fact the SGD noise may be responsible for Lévy flights in the phase space of the weights during training. These results are at the basis of the current search for further theoretical understanding of the possible motivations underlying the observed "big jumps" (Hodgkinson & Mahoney, 2021; Gurbuzbalaban et al., 2021).

**Wide flat minima and generalisation** — The connection between the geometry of the solution space and generalisation properties is the object of intense investigation in the ML theory community. A line of works in this direction focuses on the *flatness* of the loss minima and how it affects the algorithmic bias. Alternative measures of flatness have been proposed, the two most studied being the *local entropy*, measuring the low-loss volume surrounding a minimiser in weight space, and the average increase in the loss profile around a minimiser, which is related to the Hessian around a minimiser. Both measures have been found to correlate with generalisation (Jiang et al., 2019; Baldassi et al., 2019, 2020) and with each other (Pittorino et al., 2021), and efficient training algorithms that search for flat regions have been proposed (Baldassi et al., 2016; Chaudhari et al., 2019). On the other hand, it is known that DNNs trained with ReLU[1] activations are invariant with respect to weight rescaling (Dinh et al., 2017), which complicates the understanding of generalisation in terms of flatness.

[1] The Rectified Linear Unit (ReLU) (Nair & Hinton, 2010) is a piece-wise linear function: $\mathrm{ReLU}(x) = \max\{0, x\}$ and is arguably the most popular activation function for DL applications.

Tracking analytically the whole trajectory of the algorithm without resorting to approximations on the update rule remains an arduous task, certainly for the state-of-the-art DNNs trained on real datasets. A detailed description of the whole trajectory taken by SGD without resorting to approximations has been obtained only in several special cases.

**(Deep) linear networks —** First such case are linear networks where the dynamics of full-batch GD has been analysed using random matrix theory techniques in shallow networks already in, e.g., Baldi et al. (1990); Baldi & Chauvin (1991); Le Cun et al. (1991); Krogh & Hertz (1992); Baldi & Hornik (1995); Bös & Opper (1997); Bös & Opper (1998) and references therein. The analysis has then been extended to the case of deep linear networks in Saxe et al. (2013); Advani et al. (2020). It is important to notice that, despite the linearity of their input-output map, the dynamics of deep linear networks is still non-linear. This line of works has led to very interesting insights about the dynamics, for instance regarding the role of initialisation, early stopping and weight decay. However, linear networks lack the expressivity of the non-linear ones and the large-time behaviour of the algorithm can be obtained with a simple spectral algorithm.

**Online SGD —** Another case where the trajectory of the algorithm was understood in detail is the *one-pass* or *online* SGD in the case of shallow neural networks. The term “online” refers to a sampling procedure where the network uses a fresh example at each time step to approximate the gradient. Therefore, in this case there is no notion of landscape and no distinction between training and test error.

A first line of works investigating this limit focuses on two-layer networks with a finite number $k$ of hidden units, in a teacher-student setting with synthetic Gaussian input data. The study of single or two-layer networks trained by online SGD has a long history in the statistical physics literature (Kinzel & Rujan, 1990; Kinouchi & Caticha, 1992; Copelli & Caticha, 1995; Biehl & Schwarze, 1995; Riegler & Biehl, 1995). In their seminal works, Saad & Solla (1995c,b,a) derived a deterministic description of the stochastic gradient updates via a set of ODEs for the overlap variables (already introduced in Chapter 1.1), holding in the infinite dimensional limit where both the number of samples $n$ and dimensions $d$ diverge at fixed rate $\alpha = n/d$ and fixed number of hidden units $k \sim \mathcal{O}_d(1)$, while crucially the learning rate scales as $1/d$. This result led to a series of important contributions in the statistical physics literature (see, e.g., Vicente et al. (1998); Saad (2009)).

Recently, Goldt et al. (2019) proved that this description is asymptotically exact and paved the way for a new stream of works addressing current open questions in ML theory, e.g., modeling the structure of real data (Goldt et al., 2020, 2022), direct feedback alignment (Refinetti et al., 2021a), continual learning (Lee et al., 2021, 2022), curriculum learning (Saglietti et al., 2021), shallow autoencoders (Refinetti & Goldt, 2022).

Another interesting line of research recently provided insights on the behaviour of online SGD for two-layer ANNs in the limit of infinitely-wide hidden layer (Rotskoff & Vanden-Eijnden, 2018; Mei et al., 2018; Chizat & Bach, 2018; Sirignano & Spiliopoulos, 2020). These works have shown that in this setting the optimisation can be mapped to a convex problem in the space of the distributions of the hidden-layer weights and the dynamics can be written in terms of a closed set of partial differential equations. This approach is known as *mean-field*[2] or *hydrodynamic* limit of neural networks. Remarkably, this result implies that ANNs in the hydrodynamic limit converge globally to perfect learning as soon as enough data are available and the learning rate is properly scaled.

The recent work of Veiga et al. (2022) reconciles the two online-learning regimes described above, considering arbitrary learning rate and a general range of hidden-layer width.

**Dynamical mean-field theory for SGD —** Tracking the trajectory of multi-pass SGD in the realistic case where the training samples are reused multiple times is a central result of this thesis. We have obtained this characterisation via the dynamical mean-field theory (DMFT) formalism (Mézard et al., 1987; Georges et al., 1996; Parisi et al., 2020). It consists in a closed set of integro-differential equations that track the full trajectory of stochastic gradient-based algorithms in the high-dimensional limit and for generic non-convex losses. The method will be discussed in more detail in Chapter 2.2. This derivation extends the one reported in Agoritsas et al. (2018) for the non-convex perceptron model (Franz et al., 2017) with random inputs and random labels, motivated there as a model of glassy phases of hard spheres. Interestingly, the DMFT equations tracking the high-dimensional limit of GD-flow have been proven rigorously in the recent work of Celentano et al. (2021). While we relegate the discussion on DMFT for modelling ANN training to the next chapter, we present a brief overview on the broader applications of this method to study mean-field spin glasses and high-dimensional statistics problems in the next section.

## 2.1.2 . Dynamical mean-field theory in the disordered systems literature

DMFT has a long history in the disordered systems physics literature, where it has been applied to study the Langevin dynamics of mean-field spin glasses using just a small number of relevant order parameters, starting from (Sompolinsky & Zippelius, 1982; Crisanti & Sompolinsky, 1987; Kirkpatrick & Thirumalai, 1987). In a nutshell, DMFT allows to reduce the description of a high-dimensional disordered system of strongly correlated degrees of freedom to a set of integro-differential equations for low-dimensional overlap parameters, where the disorder has been integrated out at the price of adding memory to the system via two-time quantities.

The DMFT equations can be derived using at least two equivalent methods:

[2] The term *mean-field* has been used in several contexts in machine learning (Poole et al., 2016; Schoenholz et al., 2017; Yang et al., 2019; Mei et al., 2019; Gilboa et al., 2019; Novak et al., 2019). Note that the term in the aforementioned works refers to a variety of approximations and concepts. In this thesis we use it with the same meaning as in Mézard et al. (1987); Georges et al. (1996); Parisi et al. (2020) as discussed in Chapter 1.1. Most importantly, the term mean-field in our case has nothing to do with the width of an eventual hidden layer.

the *dynamical cavity approach* (Mézard et al., 1987) and the *Martin-Siggia-Rose-Janssen-De Dominicis formalism* (Martin et al., 1973; Janssen, 1976; De Dominicis, 1978) for path integrals, that is the one adopted in this thesis in its supersymmetric (SUSY) version (J. Kurchan, 1992; Kurchan, 2002; Zinn-Justin, 2002). Depending on the structure of the Hamiltonian, the precise form of the final equations can differ. For the simplest class of DMFT equations, we start from the dynamics of a high-dimensional system of $d$ coupled degrees of freedom $\boldsymbol{w}(t) = \{w_j(t)\}_{j=1}^d$ and we end up with a pair of closed integro-differential equations[3] for the correlation $C(t,t') = \langle \sum_{j=1}^d w_j(t) w_j(t') \rangle / d$ and the linear response function $R(t,t') = \langle \sum_{j=1}^d \delta w_j(t)/\delta H_j(t')|_{H=0} \rangle$, where $H_j(t)$ is a local field coupled to the $j^{\text{th}}$ degree of freedom.

The celebrated *$p-$spin spherical model*, described by the disordered long-range $p-$body Hamiltonian:

$$\mathcal{H}(\boldsymbol{w}) = - \sum_{i_1 < i_2 < \ldots < i_p} J_{i_1 i_2 \ldots i_p} w_{i_1} w_{i_2} \ldots w_{i_p}, \qquad \sum_{j=1}^d w_j^2 = d, \tag{2.1.6}$$

falls in this category (see Sompolinsky & Zippelius (1982) for $p = 2$, Kirkpatrick & Thirumalai (1987); Crisanti & Sompolinsky (1987) for $p > 2$). The term $J_{i_1 i_2 \ldots i_p}$ in Eq. (2.1.6) denotes a rank$-p$ symmetric tensor in dimension $d$ whose components are either drawn i.i.d. from a standard Gaussian distribution (*random case*) or generated by an hidden signal $\boldsymbol{w}^*$: $J_{i_1 i_2 \ldots i_p} = w^*_{i_1} w^*_{i_2} \ldots w^*_{i_p}$ (*planted case*). The mean-field $p-$spin has served as a prototypical model for the glass transition to explore the connection between static and dynamic properties (Crisanti & Sommers; Crisanti et al., 1993). The out-of-equilibrium dynamics of the model was solved by Cugliandolo & Kurchan (1993a). Their equations resulted in a key development of the understanding of the relaxation dynamics of glassy systems (Bouchaud et al., 1996, 1998). Moreover, while in general DMFT is a heuristic statistical physics method, it has been amenable to a rigorous proof in this case (Arous et al., 1997; Ben Arous et al., 2006).

In general, and also in the case of the perceptron model considered in this thesis, the DMFT equations cannot be closed on correlation and response functions and instead involve memory kernels that must be determined from a one-dimensional[4] stochastic process in a self-consistent way (see, e.g., Opper & Diederich (1992); Maimbourg et al. (2016); Agoritsas et al. (2018); Pearce et al. (2020); Roy et al. (2020)). This procedure will be further discussed in Chapter 2.2. We also refer the interested reader to Cugliandolo (2002) for a more comprehensive introduction.

**Exploring the connection between statics and dynamics in mixed $p-$spin models** — Recently, mixed $p-$spin models, i.e., models described by combinations of the

[3] Usually referred to as Crisanti-Horner-Sommers-Cugliandolo-Kurchan (CHSCK) equations in the literature.

[4] For variants of the problem such as two-layer networks with a finite number $k$ of hidden units or a system of $k$ coupled replicas, the self-consistent process is $k-$dimensional.

Hamiltonian in Eq. (2.1.6):

$$\mathcal{H}(\boldsymbol{w}) = -\sum_{p\in P} \alpha_p \sum_{i_1<i_2<\ldots<i_p} J_{i_1 i_2 \ldots i_p} w_{i_1} w_{i_2} \ldots w_{i_p}, \tag{2.1.7}$$

controlled by the mixture coefficients $\{\alpha_p\}_{p\in P}$, have been adopted as prototypes of hard optimisation problems in order to investigate the interplay of the energy landscape and the asymptotic gradient-descent dynamics.

In Folena et al. (2020), the authors consider a mixture of random $p-$spin models, revealing that the relaxation clearly differs from the pure $p-$spin model in the fact that the asymptotic states keep memory of the initial condition and the final energy is a decreasing function of the temperature at which the initial configuration thermalises.

Mannelli et al. (2020a) have studied the mixed model: $P = \{p_1 = 2, p_2 > 2\}$ generated by a planted signal, also known as *spiked matrix-tensor model*, to assess the performance of the Langevin algorithm at high-dimensional noisy inference, finding that its algorithmic threshold is suboptimal with respect to the one given by the AMP algorithm. In Mannelli et al. (2019); Sarao Mannelli et al. (2019), the authors extend the analysis to gradient flow and develop a quantitative theoretical framework to explain how GD can find good minima despite the presence of exponentially-many bad local minima, by combining the dynamical equations with the Kac-Rice analysis of the stationary points of the landscape.

However, the spiked matrix-tensor model does not offer a natural way to study the SGD algorithm or to explore the difference between training and test errors. In particular, this model does not allow for the study of the so-called interpolating regime, where the loss function is optimised to zero while the test error remains positive. As such, its landscape is intrinsically different from supervised learning problems since in the former the spurious minima proliferate at high values of the loss while the good ones lie at the bottom of the landscape. Instead, DNNs have both spurious and good minima at 100% training accuracy and their landscape resembles much more the one of continuous constraint satisfaction problems (Franz et al., 2017, 2019a).

# 2.2 - Dynamical mean-field theory for stochastic gradient descent

In this chapter, we present how to extend the dynamical mean-field theory (DMFT) to analyse in a closed form the learning dynamics of the multi-pass SGD algorithm in the high-dimensional Gaussian mixture model (GMM) for binary classification introduced in Chapter 1.2, and in a non-linearly-separable variant. This formalism allows us to explore the performance of the algorithm as a function of the problem parameters in a prototype non-convex loss landscape with interpolating regimes and a large generalisation gap.

## 2.2.1 . Introduction to the task

We consider a training set made of $n$ points:

$$\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n\times d} \quad \text{with binary labels } \boldsymbol{y} = (y_1, \dots, y_n)^\top \in \{\pm 1\}^n, \tag{2.2.1}$$

and two different GMMs for the data:

- The *two-cluster dataset* is the same data model introduced in Section 1.2.1 in the special case of balanced clusters, i.e., drawn with equal probability $\mathrm{P}(y_\mu = 1) = \mathrm{P}(y_\mu = -1) = 1/2$, $\forall \mu = 1, \dots, n$, and

  $$\boldsymbol{x}_\mu = y_\mu \frac{\boldsymbol{w}^*}{\sqrt{d}} + \sqrt{\Delta}\boldsymbol{z}_\mu \,. \tag{2.2.2}$$

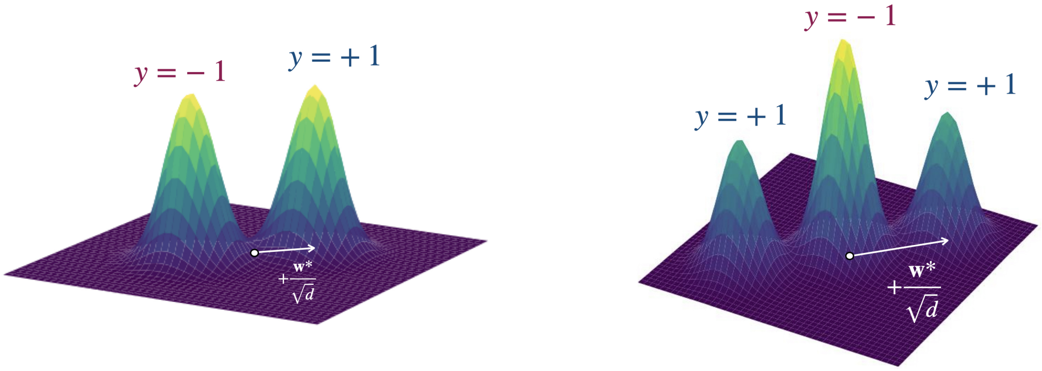
  As already discussed in Chapter 1.2, if the noise level $\Delta$ and/or the sample complexity $\alpha$ are small enough, the two Gaussian clouds are linearly separable by an hyperplane, and a single-layer ANN is enough to perform the classification task. We thus consider learning with the simplest prediction rule $\hat{y}_\mu(\boldsymbol{w}) = \mathrm{sign}\left(\boldsymbol{x}_\mu^\top \boldsymbol{w}\right)$.

- The *three-cluster dataset* is again a binary classification task but on a GMM that is not linearly separable anymore:

  $$\boldsymbol{x}_\mu = c_\mu \frac{\boldsymbol{w}^*}{\sqrt{d}} + \sqrt{\Delta}\boldsymbol{z}_\mu, \quad \text{with } y_\mu = 2c_\mu^2 - 1, \tag{2.2.3}$$

  and

  $$c_\mu = \begin{cases} +1 & \text{with prob. } \frac{1}{4} \\ -1 & \text{with prob. } \frac{1}{4} \\ 0 & \text{with prob. } \frac{1}{2} \end{cases}, \qquad \forall \mu = 1, \dots, n. \tag{2.2.4}$$


$y = -1$
$y = +1$
$+\frac{\mathbf{w}^*}{\sqrt{d}}$
$y = -1$
$y = +1$
$y = +1$
$+\frac{\mathbf{w}^*}{\sqrt{d}}$

(a) Two-cluster dataset.

(b) Three-cluster dataset.

Figure 2.2.1 – Pictorial representation of the Gaussian mixture datasets under consideration.

Therefore, in this case the data come from a mixture of three clouds of Gaussian points, with two external clouds centered at $\pm\boldsymbol{w}^*/\sqrt{d}$ and one centered at the origin. In order to fit the data, we consider a single-layer ANN with the *door activation* function, defined as

$$\hat{y}_\mu(\boldsymbol{w}) = \operatorname{sign}\left(\left(\frac{\boldsymbol{x}_\mu^\top \boldsymbol{w}}{\sqrt{d}}\right)^2 - L^2\right). \tag{2.2.5}$$

For simplicity, we will fix the onset parameter $L$, that in principle can be learned as well, for instance by cross-validation.

A pictorial representation of the data models under consideration is shown in Figure 2.2.1.

Similarly as in previous chapters, we consider the empirical risk minimisation (ERM) framework, with empirical risk given by

$$\mathcal{H}(\boldsymbol{w}) = \sum_{\mu=1}^{n} \ell\left(y_\mu\, \phi\left(\frac{\boldsymbol{x}_\mu^\top \boldsymbol{w}}{\sqrt{d}}\right)\right) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2, \tag{2.2.6}$$

where we have added a ridge regularisation term of strength $\lambda$. The activation function $\phi(\cdot)$ is given by

$$\phi(z) = \begin{cases} x & \textit{linear} \text{ for the two-cluster dataset} \\ x^2 - L^2 & \textit{door} \text{ for the three-cluster dataset} \end{cases}. \tag{2.2.7}$$

## 2.2.2 . The training algorithms

In this section, we introduce the stochastic training algorithms that will be the object of study of this and the following chapters. We first start by writing the definitions of the algorithmic updates in discrete time.

**Full-batch gradient descent —** The discrete dynamics of full-batch GD is given by the weights update:

$$\begin{aligned} w_j(t+\mathrm{d}t) &= w_j(t) - \mathrm{d}t\,\left[\partial_{w_j}\mathcal{H}\left(\boldsymbol{w}\right) + \lambda w_j(t)\right] \\ = w_j(t) - \mathrm{d}t &\left[\sum_{\mu=1}^{n}\Lambda'\left(y_\mu, \frac{\boldsymbol{w}(t)^\top \boldsymbol{x}_\mu}{\sqrt{d}}\right)\frac{x_{\mu,j}}{\sqrt{d}} + \lambda w_j(t)\right], \quad \forall j = 1,\dots,n, \end{aligned} \tag{2.2.8}$$

where $\mathrm{d}t > 0$ is the time step and we have introduced the function $\Lambda(y,h) = \ell\left(y\phi\left(h\right)\right)$ with a prime indicating the derivative with respect to $h$, i.e., $\Lambda'(y,h) = y\ell'\left(y\phi\left(h\right)\right)\phi'\left(h\right)$. We consider a Gaussian initialisation of the weight vector $\boldsymbol{w}(0) \sim \mathcal{N}(\boldsymbol{0}, R\,\boldsymbol{I}_d)$, where $R > 0$ is a parameter that tunes the average norm of the weight vector at the beginning of the dynamics.

In the following, we consider different ways to add stochasticity to the dynamics.

**Multi-pass stochastic gradient descent —** We study multi-pass SGD, where the samples are reused multiple times during training. We consider the case where mini batches are sampled with replacement with size $B = \mathtt{b}n$, $\mathtt{b} \in (0,1]$ at each time step. If we introduce a set of binary variables $s_\mu(t) \in \{0,1\}$, $\mu = 1,...,n$, to select which samples are used compute the approximate gradient, then in the large $d$ limit the vanilla-SGD algorithm (sampling with replacement) is equivalent to

$$\begin{aligned} & w_j(t+\mathrm{d}t) = \\ w_j(t) - \mathrm{d}t & \left[\sum_{\mu=1}^{n} s_\mu(t)\Lambda'\left(y_\mu, \frac{\boldsymbol{w}(t)^\top \boldsymbol{x}_\mu}{\sqrt{d}}\right)\frac{x_{\mu,j}}{\sqrt{d}} + \lambda w_j(t)\right], \quad \forall j = 1,\dots,n, \end{aligned} \tag{2.2.9}$$

where we draw

$$s_\mu(t) = \begin{cases} 1 & \text{with probability } \mathtt{b} \\ 0 & \text{otherwise} \end{cases} \tag{2.2.10}$$

independently at each time step. However, the continuous-time limit $\mathrm{d}t \to 0^+$ is not well-defined in this case.

**Persistent stochastic gradient descent —** We define an alternative *persistent* version of the SGD discrete-time process for the variables $s_\mu(t)$. We call the resulting algorithm persistent-SGD (p-SGD). In Chapter 5, we will show that this choice of mini-batch sampling can result in a performance improvement in some cases. The sampling vector is initialised as

$$s_\mu(t=0) = \begin{cases} 1 & \text{w.p. } \mathtt{b} \\ 0 & \text{otherwise} \end{cases} \tag{2.2.11}$$

and updated according to

$$\begin{aligned} \mathrm{Prob}\left(s_\mu(t+\mathrm{d}t) = 1 | s_\mu(t) = 0\right) &= \\ 1 - \mathrm{Prob}\left(s_\mu(t+\mathrm{d}t) = 0 | s_\mu(t) = 0\right) &= \frac{\mathrm{d}t}{\tau}, \\ \mathrm{Prob}\left(s_\mu(t+\mathrm{d}t) = 0 | s_\mu(t) = 1\right) &= \\ 1 - \mathrm{Prob}\left(s_\mu(t+\mathrm{d}t) = 1 | s_\mu(t) = 1\right) &= \frac{1-\mathtt{b}}{\mathtt{b}\tau}\mathrm{d}t. \end{aligned} \tag{2.2.12}$$

This sampling process has the advantage of being well-defined in the continuous-time limit. Indeed, each sampling variable $s_\mu(t)$ follows a two-state Markov jump process with exponentially-distributed transition times and inhomogeneous switching rates: $r_{0\to1}^{s_\mu(t)} = 1/\tau$ ("activation" rate), $r_{1\to0}^{s_\mu(t)} = (1-\mathtt{b})/\mathtt{b}\tau$ ("deactivation" rate). The value of $\tau > 0$ indicates the average time spent by each sample out of the training mini-batch, and we will call it *persistence time*. The average time spent in the training mini batch by each sample is $\tau\mathtt{b}/(1-\mathtt{b})$. If we set $\tau = \mathrm{d}t/\mathtt{b}$, we recover the vanilla-SGD algorithm. Note that, in this setting, there are two parameters controlling the stochasticity of the algorithm: the mini-batch size $\mathtt{b}$ and the persistence time $\tau$.

**Langevin dynamics** — A different kind of stochastic dynamics is provided by the Langevin algorithm at temperature $T$, widely studied in physics:

$$w_j(t+\mathrm{d}t) = w_j(t) - \mathrm{d}t\left[\partial_{w_j}\mathcal{H}\left(\boldsymbol{w}\right) + \lambda w_j(t)\right] + \mathrm{d}t\,\varsigma_j(t), \quad \forall j = 1,...d. \tag{2.2.13}$$

The random vector $\boldsymbol{\varsigma}(t)$ is Gaussian white noise:

$$\begin{aligned} \langle \varsigma_j(t)\rangle &= 0, & \forall j = 1,...d, \\ \langle \varsigma_i(t)\varsigma_j(t')\rangle &= 2T\,\delta_{ij}\,\delta(t-t'), & \forall i,j = 1,...d. \end{aligned} \tag{2.2.14}$$

Note that by setting $\mathtt{b} = 1$ in Eq. (2.2.9) or $T = 0$ in Eq. (2.2.13) we recover the full-batch GD algorithm.

**Stochastic gradient flow** — In the following, as done in Article 3, we write the DMFT equations for the continuous-time dynamics defined by the $\mathrm{d}t \to 0^+$ limit. We will then discuss the discrete-time case. For simplicity, the flow dynamics of SGD and Langevin can be regrouped in the following *stochastic gradient flow* (SGF) equations

$$\dot{w}_j(t) = -\left[\sum_{\mu=1}^{n} s_\mu(t)\Lambda'\left(y_\mu, \frac{\boldsymbol{w}(t)^\top\boldsymbol{x}_\mu}{\sqrt{d}}\right)\frac{x_{\mu,j}}{\sqrt{d}} + \lambda w_j(t)\right] + \varsigma_j(t), \tag{2.2.15}$$

$\forall j = 1,...d$, where *gradient flow* – the continuous-time limit of GD – is recovered by setting $\mathtt{b} = 1$ and $T = 0$.

## 2.2.3 . Dynamical mean-field theory for SGD

We will now analyse the SGF in the infinite-size limit $n, d \to \infty$ at fixed $\alpha = n/d$, $\mathtt{b}$ and $\tau$ of order one. To this end, we use DMFT from statistical physics of disordered systems. As already mentioned in Chapter 2.1, there are at least two ways to write the DMFT equations. One is by using field-theoretical techniques, the other is to employ a dynamical version of the so-called *cavity method* (Mézard et al., 1987). Here we opt for the first option that is generally very compact and immediate. We use a supersymmetric (SUSY) representation to derive the DMFT equations (J. Kurchan, 1992; Agoritsas et al., 2018), leading to a computation that resembles very much a *static* treatment of the Gibbs measure of the problem (Kurchan, 2002), as the ones carried out in Chapters 1.2 and 1.3 of this thesis.

The derivation based on the cavity method is detailed in Agoritsas et al. (2018). The main differences of the present work with respect to Agoritsas et al. (2018) are that here we consider multi-pass SGD and that our dataset is structured while in Agoritsas et al. (2018) the derivation was done for full-batch GD, random i.i.d. inputs and random labels, i.e., a case where learning is impossible and we cannot investigate the generalisation error and its properties.

The starting point of the DMFT is the dynamical partition function

$$
\begin{aligned}
Z_{\mathrm{dyn}} = &\int\limits_{\boldsymbol{w}(0)=\boldsymbol{w}^{(0)}} \mathcal{D}\left[\boldsymbol{w}(t)\right] \\
\times \prod_{j=1}^{d} \delta &\left[-\dot{w}_j(t) - \lambda w_j(t) - \sum_{\mu=1}^{n} s_\mu(t)\Lambda'\left(y_\mu, \frac{\boldsymbol{w}(t)^\top \boldsymbol{x}_\mu}{\sqrt{d}}\right)\frac{x_{\mu,j}}{\sqrt{d}} + \zeta_j(t)\right],
\end{aligned}
\tag{2.2.16}
$$

where $\int_{\boldsymbol{w}(0)\,=\,\boldsymbol{w}^{(0)}}\mathcal{D}\left[\boldsymbol{w}(t)\right]$ stands for the measure over the dynamical trajectories starting from $\boldsymbol{w}^{(0)}$ and following the dynamics given by Eq. (2.2.15). Note that we only fix the initial condition, while the endpoint of the dynamics is free. Therefore, the trajectory is unique due to the causality of the dynamics. Moreover, in our case the definition of the discrete-time updates implies the use of the Itô convention. Thus, the determinant of the Jacobian of the change of variables from $\boldsymbol{\zeta}$ to $\boldsymbol{w}$ is equal to one (Cugliandolo & Lecomte, 2017) and we do not have to introduce fermionic fields to compute it (Zinn-Justin, 2002; Cugliandolo, 2002).

Since $Z_{\mathrm{dyn}} = \mathbb{E}[Z_{\mathrm{dyn}}] = 1$ (it is just an integral of a Dirac delta function) (Dominicis, 1976) one can average directly $Z_{\mathrm{dyn}}$ over the training set, the initial condition, the Langevin noise and the stochastic processes of $s_\mu(t)$:

$$
\begin{aligned}
Z_{\mathrm{dyn}} = &\left\langle \int \left[\frac{\mathrm{d}\boldsymbol{w}^{(0)}}{(2\pi)^{\frac{d}{2}}} e^{-\frac{1}{2}\|\boldsymbol{w}^{(0)}\|_2^2}\right] \int \mathcal{D}\boldsymbol{\zeta}(t) \int_{\boldsymbol{w}(0)=\boldsymbol{w}^{(0)}} \mathcal{D}\boldsymbol{w}(t) \right.\\
&\left. \times \prod_{j=1}^{d} \delta\left[-\dot{\mathrm{w}}_j(t) - \lambda \mathrm{w}_j(t) - \sum_{\mu=1}^{n} s_\mu(t)\Lambda'\left(y_\mu, \frac{\boldsymbol{w}(t)^\top \boldsymbol{x}_\mu}{\sqrt{d}}\right)\frac{\mathrm{x}_{\mu,j}}{\sqrt{d}} + \zeta_j(t)\right]\right\rangle,
\end{aligned}
\tag{2.2.17}
$$

where the brackets $\langle\cdot\rangle$ stand for the average over $s_\mu(t)$, $y_\mu$ and the realisation of the noise in the training set. The averages over the initial condition and the Langevin noise are written explicitly. Note that we choose an initial condition that is Gaussian, but we could have chosen a different probability measure over the initial configuration of the weights. The equations can be generalised to other initial conditions as soon as they do not depend on quenched random variables that enter in the SGD dynamics and their distribution is separable. Since the initial condition is uncorrelated with the disorder, there is no need to use the replica trick (Houghton et al., 1983). After a few steps of algebra, we obtain

$$
Z_{\mathrm{dyn}} = \left\langle \int \mathcal{D}\boldsymbol{w}(t)\mathcal{D}\hat{\boldsymbol{w}}(t)\, e^{S_{\mathrm{dyn}}} \right\rangle,
\tag{2.2.18}
$$

where we have defined

$$\begin{aligned} S_{\rm dyn} = \sum_{j=1}^{d} \int_0^{+\infty} {\rm d}t\, {\rm i}\hat{w}_j(t) \big(-\dot{w}_j(t) - \lambda w_j(t) \\ - \sum_{\mu=1}^{n} s_\mu(t) \Lambda' \left( y_\mu, \frac{\boldsymbol{w}(t)^\top \boldsymbol{x}_\mu}{\sqrt{d}} \right) \frac{{\rm x}_{\mu,j}}{\sqrt{d}} - {\rm i}T\, \hat{w}_j(t) \Big) . \end{aligned} \tag{2.2.19}$$

and we have introduced a set of conjugate fields $\hat{\boldsymbol{w}}(t)$ to produce the integral representation of the Dirac $\delta-$function.

**SUSY formulation —** The dynamical action $S_{\rm dyn}$ in Eq. (2.2.19) can be rewritten in a SUSY form, by extending the time coordinate to include two Grassmann coordinates[1] $\theta$ and $\bar{\theta}$, i.e., $t_a \to a = (t_a, \theta_a, \bar{\theta}_a)$. The dynamic variable $\boldsymbol{w}(t_a)$ and the auxiliary variable $\hat{w}(t_a)$ are encoded together in a super-field

$$\boldsymbol{w}(a) = \boldsymbol{w}(t_a) + {\rm i}\, \theta_a \bar{\theta}_a \hat{\boldsymbol{w}}(t_a). \tag{2.2.20}$$

In the following, the term "super" refers to any quantity involving both commuting and anticommuting variables. The introduction of these mathematical objects will help us in the calculations. From the properties of Grassmann variables (Zinn-Justin, 2002):

$$\begin{aligned} &\theta^2 = \bar{\theta}^2 = \theta\bar{\theta} + \bar{\theta}\theta = 0, \\ &\int {\rm d}\theta = \int {\rm d}\bar{\theta} = 0, \qquad \int {\rm d}\theta\, \theta = \int {\rm d}\bar{\theta}\, \bar{\theta} = 1, \\ &\partial_\theta g(\theta) = \int {\rm d}\theta\, g(\theta) \quad \text{for a generic function } g, \end{aligned} \tag{2.2.21}$$

it follows that

$$\int {\rm d}a\, f\left(\boldsymbol{w}(a)\right) = \int_0^{+\infty} {\rm d}t_a\, {\rm i}\hat{\boldsymbol{w}}(t_a)^\top \nabla_{\boldsymbol{w}} f\left(\boldsymbol{w}(t_a)\right). \tag{2.2.22}$$

Note that, as a consequence of the properties of Grassmann algebra, a function of Grassman variables can only be linear. We can use Eq. (2.2.22) to rewrite $S_{\rm dyn}$. We obtain

$$S_{\rm dyn} = -\frac{1}{2} \int {\rm d}a{\rm d}b\, \mathcal{K}(a,b) \boldsymbol{w}(a)^\top \boldsymbol{w}(b) - \sum_{\mu=1}^{n} \int {\rm d}a\, s_\mu(a)\, \Lambda\left(y_\mu, h_\mu(a)\right), \tag{2.2.23}$$

[1] Grassmann anticommuting variables were first presented by Herman G. Grassmann (1809-1877). *Supermathematics*, i.e., the use of commuting and anticommuting variables on equal footing, and its important applications to physics were introduced by Felix A. Berezin (1931-1980). His most important results are the Berezin integral over Grassmann anticommuting variables and the Berezinian, i.e., the generalisation of the Jacobian. From this construction it follows that the integral of a Grassman variable is equal to its derivative. An extended introduction and more examples on the use of Grassmann variables can be found in the physics books Efetov (1983); Zinn-Justin (2002) and the mathematics books Berezin (1987); DeWitt (1992).

where we have defined $h_\mu(a) \equiv \boldsymbol{w}(a)^\top \boldsymbol{x}_\mu/\sqrt{d}$ and we have defined the kernel $\mathcal{K}(a,b)$ such that

$$\begin{gathered} -\frac{1}{2}\int \mathrm{d}a\mathrm{d}b\,\mathcal{K}(a,b)\boldsymbol{w}(a)^\top \boldsymbol{w}(b) = \\ \sum_{j=1}^{d}\int_0^{+\infty} \mathrm{d}t\,\mathrm{i}\hat{w}_j(t)\left(-\dot{w}_j(t) - \lambda w_j(t) - \mathrm{i}T\hat{w}_j(t)\right)\,, \end{gathered} \tag{2.2.24}$$

which is given by

$$\begin{aligned} &\mathcal{K}(a,b) = -2T\delta(t_a - t_b) - \theta_a\bar{\theta}_a\partial_{t_a}\delta(t_b - t_a) - \theta_b\bar{\theta}_b\partial_{t_b}\delta(t_a - t_b) + \lambda\delta(a,b), \\ &\delta(a,b) = \delta(t_a - t_b)(\theta_a\bar{\theta}_a - \theta_b\bar{\theta}_b). \end{aligned} \tag{2.2.25}$$

By inserting the definition of $h_\mu(a)$ in the partition function, we have

$$\begin{aligned} Z_{\text{dyn}} = \Bigg\langle \int & \mathcal{D}\boldsymbol{w}(a)\mathcal{D}h_\mu(a)\mathcal{D}\hat{h}_\mu(a) \\ & \exp\left[-\frac{1}{2}\int \mathrm{d}a\mathrm{d}b\,\mathcal{K}(a,b)\boldsymbol{w}(a)^\top\boldsymbol{w}(b) - \sum_{\mu=1}^{n}\int \mathrm{d}a\, s_\mu(a)\,\Lambda\left(y_\mu, h_\mu(a)\right)\right] \\ & \exp\left[\sum_{\mu=1}^{n}\int \mathrm{d}a\,\mathrm{i}\,\hat{h}_\mu(a)\left(h_\mu(a) - \frac{\boldsymbol{w}(a)^\top\boldsymbol{x}_\mu}{\sqrt{d}}\right)\right]\Bigg\rangle\,. \end{aligned} \tag{2.2.26}$$

Let us consider the last factor in the integral in (2.2.26). We can perform the average over the random vectors $\boldsymbol{z}_\mu \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$, denoted by an overline, as

$$\begin{aligned} &\overline{\exp\left[\sum_{\mu=1}^{n}\int \mathrm{d}a\,\mathrm{i}\,\hat{h}_\mu(a)\left(h_\mu(a) - \frac{\boldsymbol{w}(a)^\top\boldsymbol{x}_\mu}{\sqrt{d}}\right)\right]} \\ &= \overline{\exp\left[\sum_{\mu=1}^{n}\int \mathrm{d}a\,\mathrm{i}\,\hat{h}_\mu(a)\left(h_\mu(a) - c_\mu m(a) - \sqrt{\frac{\Delta}{d}}\boldsymbol{w}(a)^\top\boldsymbol{z}_\mu\right)\right]} \\ &= \exp\left[\sum_{\mu=1}^{n}\int \mathrm{d}a\,\mathrm{i}\,\hat{h}_\mu(a)\left(h_\mu(a) - c_\mu m(a)\right)\right. \\ &\left. -\frac{\Delta}{2}\sum_{\mu=1}^{n}\int \mathrm{d}a\,\mathrm{d}b\,Q(a,b)\hat{h}_\mu(a)\hat{h}_\mu(b)\right]\,, \end{aligned} \tag{2.2.27}$$

where we have defined the *dynamical* overlap variables

$$\begin{aligned} m(a) &= \frac{1}{d}\boldsymbol{w}(a)^\top\boldsymbol{w}^*, \\ Q(a,b) &= \frac{1}{d}\boldsymbol{w}(a)^\top\boldsymbol{w}^*(b). \end{aligned} \tag{2.2.28}$$

By inserting the definitions of $m(a)$ and $Q(a,b)$ in the partition function, we obtain

$$Z_{\text{dyn}} = \int \mathcal{D}\boldsymbol{Q}\,\mathcal{D}\boldsymbol{m}\; e^{dS(\boldsymbol{Q},\boldsymbol{m})}, \tag{2.2.29}$$

where $\boldsymbol{Q} = \{Q(a,b)\}_{a,b}$, $\boldsymbol{m} = \{m(a)\}_a$ and

$$\begin{aligned}
S(\boldsymbol{Q},\boldsymbol{m}) &= \frac{1}{2}\log\det\left(Q(a,b) - m(a)m(b)\right) - \frac{1}{2}\int \mathrm{d}a\mathrm{d}b\,\mathcal{K}(a,b)Q(a,b) + \alpha\log\mathcal{Z},\\
\mathcal{Z} &= \left\langle \int \mathcal{D}h(a)\mathcal{D}\hat{h}(a)\,\exp\left[-\frac{\Delta}{2}\int \mathrm{d}a\mathrm{d}b\;Q(a,b)\hat{h}(a)\hat{h}(b)\right.\right.\\
&\quad \left.\left. + \int \mathrm{d}a\, i\hat{h}(a)\left(h(a) - cm(a)\right) - \int \mathrm{d}a\, s(a)\,\Lambda\left(y, h(a)\right)\right]\right\rangle ,
\end{aligned} \tag{2.2.30}$$

where the term $\log\det\left(Q(a,b) - m(a)m(b)\right)$ is due to the change of variables from $\boldsymbol{w}(a)$ to $\boldsymbol{Q}(a,b)$, $\boldsymbol{m}(a)$, that is analogous to the one performed for the static replica computation in Chapter 1.2. We have used that the samples are i.i.d. and removed the index $\mu = 1,...n$. The brackets denote the average over the random variable $c$, that has the same distribution as the $c_\mu$, over $y$, distributed as $y_\mu$, and over the random sampling process of $s(t)$. If we perform the translation $Q(a,b) \leftarrow Q(a,b) + m(a)m(b)$, we obtain

$$\begin{aligned}
S(\boldsymbol{Q},\boldsymbol{m}) &= \frac{1}{2}\log\det Q(a,b) - \frac{1}{2}\int \mathrm{d}a\mathrm{d}b\,\mathcal{K}(a,b)\left(Q(a,b) + m(a)m(b)\right) + \alpha\log\mathcal{Z},\\
\mathcal{Z} &= \left\langle \int \mathcal{D}h(a)\mathcal{D}\hat{h}(a)\, e^{S_{\text{loc}}}\right\rangle ,
\end{aligned} \tag{2.2.31}$$

where the effective local action $S_{\text{loc}}$ is given by

$$\begin{aligned}
S_{\text{loc}} &= -\frac{\Delta}{2}\int \mathrm{d}a\mathrm{d}b\;Q(a,b)\hat{h}(a)\hat{h}(b) - \frac{\Delta}{2}\left(\int \mathrm{d}a\,\hat{h}(a)m(a)\right)^2\\
&\quad + \int \mathrm{d}a\,\mathrm{i}\hat{h}(a)\left(h(a) - cm(a)\right) - \int \mathrm{d}a\, s(a)\,\Lambda\left(y, h(a)\right).
\end{aligned} \tag{2.2.32}$$

Performing a Hubbard-Stratonovich transformation

$$\exp\left[-\frac{\Delta}{2}\left(\int \mathrm{d}a\,\hat{h}(a)m(a)\right)^2\right] = \int \frac{\mathrm{d}h_0}{\sqrt{2\pi}}\mathrm{e}^{-\frac{h_0^2}{2}}\,\exp\left(\mathrm{i}\sqrt{\Delta}\,h_0\int \mathrm{d}a\,\hat{h}(a)m(a)\right) \tag{2.2.33}$$

and a set of transformations on the fields $h(a)$ and $h_0$:

$$h(a) \leftarrow h(a) + m(a)(c + h_0), \quad h(a), h_0 \leftarrow \sqrt{\Delta}h(a), \sqrt{\Delta}h_0, \tag{2.2.34}$$

we obtain that we can rewrite $\mathcal{Z}$ as

$$\begin{aligned}
\mathcal{Z} &= \left\langle \int \frac{\mathrm{d}h_0}{\sqrt{2\pi}}e^{-\frac{h_0^2}{2}}\int \mathcal{D}h(a)\mathcal{D}\hat{h}(a)\,\exp\left[-\frac{1}{2}\int \mathrm{d}a\mathrm{d}b\;Q(a,b)\hat{h}(a)\hat{h}(b)\right.\right.\\
&\quad \left.\left. + \int \mathrm{d}a\,\mathrm{i}\hat{h}(a)h(a) - \int \mathrm{d}a\, s(a)\,\Lambda\left(y, \sqrt{\Delta}h(a) + m(a)(c + \sqrt{\Delta}h_0)\right)\right]\right\rangle\\
&= \left\langle \int \frac{\mathrm{d}h_0}{\sqrt{2\pi}}e^{-\frac{h_0^2}{2}}\int \mathcal{D}h(a)\mathcal{D}\hat{h}(a)\mathrm{e}^{S_{\text{loc}}}\right\rangle
\end{aligned} \tag{2.2.35}$$

**Saddle-point equations —** We are interested in the large $d$ limit of $Z_{\text{dyn}}$, in which, according to Eq. (2.2.29), the partition function is dominated by the saddle-point value of $S(\boldsymbol{Q}, \boldsymbol{m})$:

$$\begin{cases} \left.\dfrac{\delta S(\boldsymbol{Q}, \boldsymbol{m})}{\delta Q(a,b)}\right|_{(\boldsymbol{Q},\boldsymbol{m})=(\tilde{Q},\tilde{m})} = 0 \\ \\ \left.\dfrac{\delta S(\boldsymbol{Q}, \boldsymbol{m})}{\delta m(a)}\right|_{(\boldsymbol{Q},\boldsymbol{m})=(\tilde{Q},\tilde{m})} = 0 \end{cases} . \tag{2.2.36}$$

The solution for the self-overlap $\tilde{Q}(a,b)$ is obtained from the equation

$$-\mathcal{K}(a,b) + Q^{-1}(a,b) + \frac{2\alpha}{\mathcal{Z}} \frac{\delta \mathcal{Z}}{\delta Q(a,b)} = 0. \tag{2.2.37}$$

The saddle-point equation for $\tilde{m}(a)$ is instead

$$-\int \mathrm{d}b\, \mathcal{K}(a,b) m(b) + \frac{\alpha}{\mathcal{Z}} \frac{\delta \mathcal{Z}}{\delta m(a)} = 0. \tag{2.2.38}$$

**Self-consistent effective stochastic process —** At this point, we have obtained that the path integral is dominated by the saddle point of the SUSY dynamical action $S(\boldsymbol{Q}, \boldsymbol{m})$, computed in Eqs. (2.2.37)-(2.2.38). However, these equations still depend on the averages over $h(a)$ and $h_0$ contained in $\mathcal{Z}$. The trick to proceed is to write an effective stochastic process for the variable $h(t)$, such that the corresponding SUSY dynamical action would be exactly the effective local action $S_{\text{loc}}$ in Eq. (2.2.35).

It can be shown by exploiting the Grassmann structure of Eqs. (2.2.37)-(2.2.38) that they lead to a self consistent stochastic process described by

$$\dot{h}(t) = -\tilde{\lambda}(t)h(t) - \sqrt{\Delta} s(t) \Lambda'\left(y, r(t)\right) + \int_0^t \mathrm{d}t' M_R(t,t') h(t') + \xi(t), \tag{2.2.39}$$

There are several sources of stochasticity in Eq. (2.2.39). First, one has a dynamical noise $\xi(t)$ that is Gaussian distributed and characterised by the correlations

$$\langle \xi(t) \rangle = 0, \qquad \langle \xi(t)\xi(t') \rangle = M_C(t,t') + T\,. \tag{2.2.40}$$

Furthermore, the starting point $h(0)$ of the stochastic process is random and distributed according to

$$P(h(0)) = e^{-h(0)^2/(2R)}/\sqrt{2\pi R}\,. \tag{2.2.41}$$

Moreover, one has to introduce a quenched Gaussian random variable $h_0$ with mean zero and variance one to model the initialisation. We recall that the random variable $c = \pm 1$ with equal probability in the two-cluster model, while $c = 0, \pm 1$ in the three-cluster one. The variable $y(c)$ is therefore $y(c) = c$ in the two-cluster case, and is given by Eq. (2.2.3) in the three-cluster one. Finally, one has a dynamical stochastic process $s(t)$ whose statistical properties are specified in Eq. (2.2.12). The magnetisation $m(t)$ is obtained from the following deterministic differential equation

$$\partial_t m(t) = -\lambda m(t) - \mu(t), \qquad m(0) = 0^+\,. \tag{2.2.42}$$

The stochastic process for $h(t)$, the evolution of $m(t)$, as well as the statistical properties of the dynamical noise $\xi(t)$ depend on a series of auxiliary functions and kernels that must be computed self-consistently and are given by

$$\begin{aligned}
\tilde{\lambda}(t) &= \lambda + \hat{\lambda}(t),\ \hat{\lambda}(t) = \alpha\Delta \left\langle s(t)\Lambda''\left(y(c), r(t)\right)\right\rangle, \\
\mu(t) &= \alpha \left\langle s(t)\left(c+\sqrt{\Delta}h_0\right)\Lambda'\left(y(c), r(t)\right)\right\rangle, \\
M_C(t,t') &= \alpha\Delta \left\langle s(t)s(t')\Lambda'\left(y(c), r(t)\right){}'\left(y(c), r(t')\right)\right\rangle, \\
M_R(t,t') &= \alpha\Delta \frac{\delta}{\delta P(t')}\langle s(t)\Lambda'(y(c), r(t))\rangle\Big|_{P=0}.
\end{aligned} \tag{2.2.43}$$

In Eq. (2.2.43) the brackets denote the average over all the sources of stochasticity in the self-consistent stochastic process, and thus capture the information about the interaction of the dynamical weights with .the dataset, the initialisation and the algorithmic noise. Therefore one needs to solve the stochastic process in a self-consistent way. Note that $P(t)$ in Eq. (2.2.39) is set to zero and we need it only to define the kernel $M_R(t,t')$. The memory kernel $M_R(t,t')$ can also be expressed as

$$M_R(t,t') = \alpha\Delta\langle s(t)\Lambda''\left(y(c), r(t)\right)\mathcal{T}(t,t')\rangle, \tag{2.2.44}$$

where $\mathcal{T}(t,t') = \delta h(t)/\delta P(t')$ satisfies:

$$\begin{aligned}
\dot{\mathcal{T}}(t,t') = -\tilde{\lambda}(t)\mathcal{T}(t,t') - \sqrt{\Delta}s(t)\Lambda''(y, r(t))\left(\mathcal{T}(t,t') - \delta(t-t')\right) \\
+ \int_{t'}^{t} \mathrm{d}s M_R(t,s)\mathcal{T}(s,t'),
\end{aligned} \tag{2.2.45}$$

which is the expression that we use in practice to solve the system. The set of Eqs. (2.2.39), (2.2.42) and (2.2.43) can be solved by an iterative algorithm, described below.

**Numerical solution of the DMFT equations —** To integrate numerically the DMFT equations, we proceed by iterations:

1. We start from a random guess of the kernels, that we use to sample several realisations of the stochastic process of Eq. (2.2.39);

2. We compute the averages over these multiple realisations to obtain the updates of the auxiliary functions and kernels in Eq. (2.2.43), along with the magnetisation (2.2.42);

3. We use these new guesses to sample again multiple realisations of the stochastic process;

4. We repeat steps 2. and 3. until the kernels reach a fixed point.

As in all iterative solutions of fixed point equations, it is natural to introduce some damping in the update of the kernels to avoid wild oscillations. Note that the DMFT fixed point equations are deterministic, hence at given initial condition the solution is unique. Indeed, the kernels computed by DMFT are causal and a simple integration scheme of the equations is just extending them progressively in time starting from their initial value, which is completely deterministic given the initial condition for the stochastic process. This procedure has been first implemented in Eissfeller & Opper (1992, 1994) and recently developed further in other applications (see, e.g., Roy et al. (2019); Manacorda et al. (2020)). However, DMFT has a long tradition in condensed matter physics Georges et al. (1996) where more involved algorithms have been developed.

In order to solve Eqs. (2.2.39), (2.2.42) and (2.2.43), we need to discretise time. In the following Section 2.2.4, in order to compare our theoretical predictions with numerical simulations, we will take a simple uniform time grid, with the time step in the DMFT equal to the learning rate in the simulations. In the time-discretised DMFT, this allows us to extract the variables $s(t)$ either from SGD or p-SGD. In the former case this provides an SGD-inspired discretisation of the DMFT equations, which is exact also in discrete time provided that the weight increments do not have higher-order terms than $\mathcal{O}(\mathrm{d}t)$. The convergence of subsequent iterations for the learning curves obtained with this numerical procedure is illustrated in Figure 2.2.2a for the average magnetisation and Figure 2.2.2b for the average loss function. Notice that, since we have taken the thermodynamic limit $d \to \infty$ at fixed time horizon, the DMFT equations provide theoretical predictions for the finite-time properties of the infinite-dimensional system.

**Correlation and response functions** — Once the self-consistent stochastic process is solved, from the solution $Q(a,b)$ of the saddle-point Eqs. (2.2.36), we can obtain the equations for the dynamical correlation function $C(t,t') = \sum_j w_j(t)w_j(t')/d$ and the response function $R(t,t') = \sum_j \delta w_j(t)/\delta H_j(t')/d$. We consider the linear response regime, where $R(t,t')$ controls the variations of the weights when their dynamical evolution is affected by an infinitesimal local field $H_i(t)$. Coupling a local field $H_i(t)$ to each variable $w_i(t)$ changes the loss function as follows: $\mathcal{H}\left(\boldsymbol{w}(t)\right) \to \mathcal{H}\left(\boldsymbol{w}(t)\right) - \sum_{i=1}^d H_i(t)w_i(t)$, resulting in an extra term $H_i(t)$ to the right hand side of Eq. (2.2.15). We then consider the limit $H_i(t) \to 0$. Indeed, we can write the closure relation

$$\begin{aligned}\delta(a,b) &= \int \mathrm{d}c\, Q^{-1}(a,c)Q(c,b) \\ &= \int \mathrm{d}c\, \left[\mathcal{K}(a,c) - \mathcal{M}(a,c)\right] Q(c,b) + \hat{\lambda}(a)Q(a,b).\end{aligned} \tag{2.2.46}$$

Now we can express the overlap explicitly in time and Grassmann coordinates

$$\begin{aligned}Q(a,b) &= \frac{1}{d}\boldsymbol{w}(a)^\top \boldsymbol{w}(b) \\ &= C(t_a,t_b) - m(t_a)m(t_b) + \theta_a\bar{\theta}_a R(t_b,t_a) + \theta_b\bar{\theta}_b R(t_a,t_b),\end{aligned} \tag{2.2.47}$$

where we remind that we have performed the change of variable $Q(a,b) \to Q(a,b) + m(a)m(b)$. Plugging the definition of $\mathcal{K}(a,b)$ (Eq. (2.2.25)) and Eq. (2.2.43) in Eq.

(2.2.46), we find

$$\begin{aligned} \delta(t_a-t_b)(\theta_a\bar\theta_a-\theta_b\bar\theta_b) = \\ -2TR(t_b,t_a)+\partial_{t_a}C(t_a,t_b)-\partial_{t_a}m(t_a)m(t_b)+\lambda\left(C(t_a,t_b)-m(t_a)m(t_b)\right) \\ -\int \mathrm{d}t_c\left[M_C(t_a,t_c)R(t_b,t_c)+M_R(t_a,t_c)\left(C(t_b,t_c)-m(t_b)m(t_c)\right)\right] \\ +\theta_a\bar\theta_a\left[\partial_{t_a}R(t_b,t_a)+\lambda R(t_b,t_a)\right]-\theta_a\bar\theta_a\int \mathrm{d}t_c\, M_R(t_c,t_a)R(t_b,t_c) \\ +\theta_b\bar\theta_b\left[\partial_{t_a}R(t_a,t_b)+\lambda R(t_a,t_b)-\int \mathrm{d}t_c\, M_R(t_a,t_c)R(t_c,t_b)\right] \\ +\hat\lambda(t_a)\left(C(t_a,t_b)-m(t_a)m(t_b)+\theta_a\bar\theta_a R(t_b,t_a)+\theta_b\bar\theta_b R(t_a,t_b)\right). \end{aligned} \tag{2.2.48}$$

We can derive the equations for correlation and response from the scalar and Grassmann terms (the terms in $\theta_a\bar\theta_a$ and $\theta_b\bar\theta_b$ result in the same contribution):

$$\begin{aligned} \partial_t C(t',t) =& -\tilde\lambda(t)C(t,t')+2TR(t',t)+\int_0^t \mathrm{d}s\, M_R(t,s)C(t',s)+\int_0^{t'} \mathrm{d}s\, M_C(t,s)R(t',s) \\ & -m(t')\left(\int_0^t \mathrm{d}s\, M_R(t,s)m(s)+\mu(t)-\hat\lambda(t)m(t)\right) \qquad \text{if } t\neq t', \\ \frac{1}{2}\,\partial_t C(t,t) =& -\tilde{}(t)C(t,t)+\int_0^t \mathrm{d}s\, M_R(t,s)C(t,s)+\int_0^t \mathrm{d}s\, M_C(t,s)R(t,s) \\ \partial_t R(t,t') =& -\tilde\lambda(t)R(t,t')+\delta(t-t')+\int_{t'}^t \mathrm{d}s\, M_R(t,s)R(s,t'), \end{aligned} \tag{2.2.49}$$

where we have used Eq. (2.2.42) in the first of Eqs. (2.2.49) . It is interesting to note that the second of Eqs. (2.2.49) controls the evolution of the norm of the weight vector $C(t,t)$ and even if we set $\lambda=0$ we get that it contains an effective regularisation $\hat\lambda(t)$ that is dynamically self-generated (Soudry et al., 2018a). At this point, the numerical solution of the equations for correlation and response is straightforward due to causality: we can integrated them in one forward pass in time, since at each step the all the required quantities are already known.

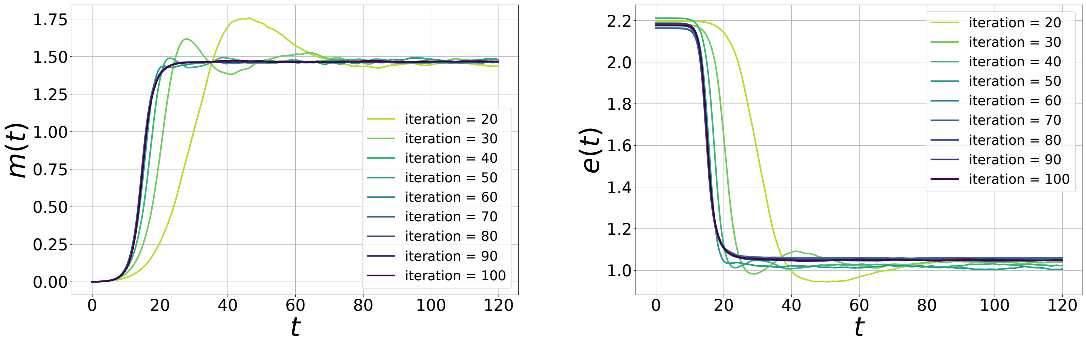
**Dynamics of the loss and the generalisation error —** Once the solution for the self-consistent stochastic process is found, one can get several interesting quantities. First, one can look at the training loss, which can be obtained as

$$e(t)=\alpha\langle\Lambda(y,r(t))\rangle, \tag{2.2.50}$$

where again the brackets denote the average over the realisation of the stochastic process in Eq. (2.2.39). The training accuracy is given by

$$a(t)=1-\langle\Theta(-y\phi(r(t)))\rangle, \tag{2.2.51}$$

where $\Theta(\cdot)$ is the Heaviside step function and we remind that $\phi$ is the activation function given by Eq. (2.2.7). By definition, the accuracy is equal to one as soon



(a) Different iterations for the averaged magnetisation computed via Eq. (2.2.42).

(b) Different iterations for the averaged loss computed via Eq. (2.2.50).

Figure 2.2.2 – We plot the time curves for different iterations of the numerical procedure to compute the DMFT solution. We consider the p-SGD algorithm at $\mathsf{b} = 1/\tau = 0.3$, $\Delta = 0.05$, $\alpha = 3$, $L = 0.707$, and $\mathrm{d}t = 0.2$, for the non-linearly separable setting of the three-cluster GMM model. Each curve corresponds to an iteration of the algorithm over a fixed time window. Different iterations are marked in different colors.

as all vectors in the training set are correctly classified. Finally, one can compute the generalisation error. At any time step, it is defined as the fraction of mislabeled instances:

$$\varepsilon_{\text{gen}}(t) = \frac{1}{4}\mathbb{E}_{\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_{\text{new}},y_{\text{new}}}\left[\left(y_{\text{new}} - \hat{y}_{\text{new}}\left(\boldsymbol{w}(t)\right)\right)^2\right], \tag{2.2.52}$$

where $\{\boldsymbol{X},\boldsymbol{y}\}$ is the training set, $\boldsymbol{x}_{\text{new}}$ is an unseen data point and $\hat{y}_{\text{new}}$ is the estimator for the new label $y_{\text{new}}$. The dependence on the training set here is hidden in the weight vector $\boldsymbol{w}(t) = \boldsymbol{w}(t,\boldsymbol{X},\boldsymbol{y})$. In the two-cluster case we have computed the error in Chapter 1.2:

$$\varepsilon_{\text{gen}}(t) = \frac{1}{2}\text{erfc}\left(\frac{m(t)}{\sqrt{2\Delta\, C(t,t)}}\right). \tag{2.2.53}$$

For the door activation trained on the three-cluster dataset we obtain

$$\varepsilon_{\text{gen}}(t) = \frac{1}{2}\text{erfc}\left(\frac{L}{\sqrt{2\Delta C(t,t)}}\right) + \frac{1}{4}\left(\text{erf}\left(\frac{L-m(t)}{\sqrt{2\Delta C(t,t)}}\right) + \text{erf}\left(\frac{L+m(t)}{\sqrt{2\Delta C(t,t)}}\right)\right). \tag{2.2.54}$$

The derivation of the above expression is very similar to that of Eq. (1.2.17) and can be found in Article 3.

## 2.2.4 . Results

In this section, we compare the theoretical curves resulting from the solution of the DMFT equations derived in Section 2.2.3 to numerical simulations. First

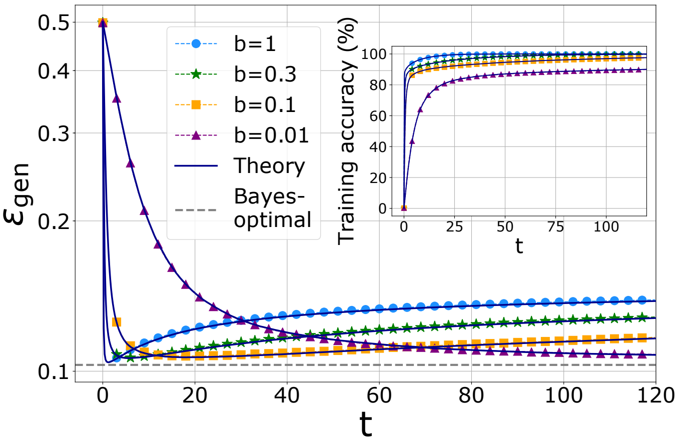
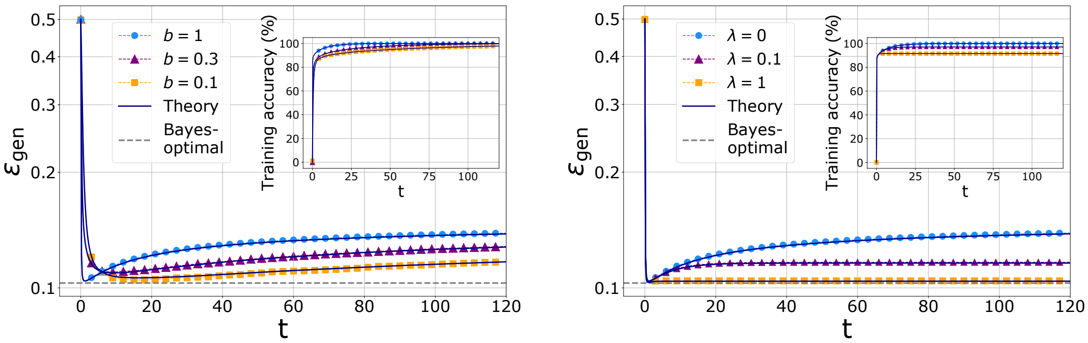


Figure 2.2.3 – Generalisation error as a function of the training time for the vanilla SGD algorithm at finite learning rate d$t$ = 0.2 and sample complexity $\alpha = 2$ for the two-clusters case. The line marks the theoretical predictions from DMFT, while the symbols mark the numerical simulations, performed at $d$ = 500. We consider different values of the batch size $\mathtt{b}$ = $0.01, 0.1, 0.3, 1$ (GD). The dashed line marks the Bayes-optimal error from Chapter 1.2. The inset shows the training accuracy.

of all, this analysis shows that our theory is indeed able to capture the learning dynamics of SGD even in discrete time and finite dimension. In Figure 2.2.3, we plot the generalisation error and the training accuracy as a function of time for SGD, comparing numerical simulations at finite dimension $d = 500$ and learning rate d$t = 0.2$ to the theoretical prediction obtained via DMFT with an analogous discretisation. We find an excellent agreement between the two. Moreover, we gain insight into the learning dynamics of SGD and its dependence on the various control parameters in the two models under consideration.

Figure 2.2.4a shows the learning dynamics of the p-SGD algorithm in the two-cluster model without regularisation $\lambda = 0$. We clearly see a good match between the numerical simulations and the theoretical curves obtained from DMFT, notably also for small values of batch size $\mathtt{b}$ and dimension $d = 500$. The figure shows that there exist regions in control parameter space where p-SGD is able to reach 100% training accuracy, while the generalisation error is bounded away from zero. Figure 2.2.4b illustrates the role of regularisation in the same model trained with full-batch gradient descent, presenting that regularisation has a similar influence on the learning curve as small batch size but without the dynamical slowing down incurred by p-SGD. The influence of the batch size $\mathtt{b}$ and the regularisation $\lambda$ for the three-cluster model is shown in Figure 2.2.5. We see an analogous effect as for the two-clusters. In the inset of Figure 2.2.5, we show the norm of the weights as a function of the training time. Both with the smaller mini-batch size and larger regularisation the norm is small, testifying further that the two play a similar role in this case.

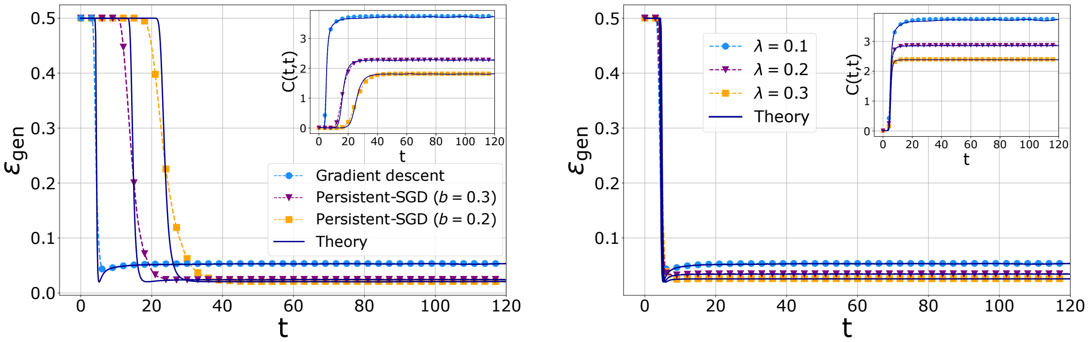
One difference between the two-cluster an the three-cluster models we observe concerns the behaviour of the generalisation error at small times. Actually, for the three-cluster model, good generalisation is reached because of finite-size effects.



(a) p-SGD at fixed $\lambda = 0, 1/\tau = 0.6$ and different batch sizes $\mathtt{b} = 1$ (GD) , $0.3, 0.1$.

(b) Full-batch GD ($\mathtt{b} = 1$) at different regularisation strengths $\lambda = 0, 0.1, 1$.

Figure 2.2.4 – Generalisation error $\varepsilon_{\text{gen}}$ as a function of the training time $t$ in the two-cluster model, with $\alpha = 2, \Delta = 0.5$. The continuous lines mark the numerical solution of DMFT equations, while the symbols are the results of simulations at dimension $d = 500$, learning rate $\mathrm{d}t = 0.2$, and initialisation variance $R = 0.01$. The insets show the training accuracy as a function of the training time. The dashed grey lines mark the BO error from the formula computed in Article 1.



(a) p-SGD at $\lambda = 0.1, 1/\tau = \mathtt{b}$ and different batch sizes $\mathtt{b} = 1$ (GD) , $0.2, 0.3$.

(b) Full-batch GD ($\mathtt{b} = 1$) at different regularisation strengths $\lambda = 0.1, 0.2, 0.3$.

Figure 2.2.5 – Generalisation error $\varepsilon_{\text{gen}}$ as a function of the training time $t$ in the three-cluster model, at fixed $\alpha = 3, \Delta = 0.05, L = 0.7$. The continuous lines mark the numerical solution of DMFT equations, while the symbols represent simulations at $\mathrm{d}t = 0.2, R = 0.01$, and $d = 1000$.

Indeed, the corresponding loss function displays a $\mathbb{Z}_2$ symmetry according to which for each local minimum $\boldsymbol{w}$ there is another one $-\boldsymbol{w}$ with exactly the same properties. Note that this symmetry is inherited from the activation function $\phi$ in Eq. (2.2.7), which is even. This implies that if $d \to \infty$, the generalisation error would not move away from 0.5 in finite time. However, when $d$ is large but finite, at time $t = 0$ the weight vector has a finite projection on the centroid $\boldsymbol{w}^*$ which is responsible for the dynamical symmetry breaking and eventually for a low generalisation error at long times. In order to obtain an agreement between the theory and simulations, we initialise $m(t)$ in the DMFT equations with its corresponding finite-$d$ average

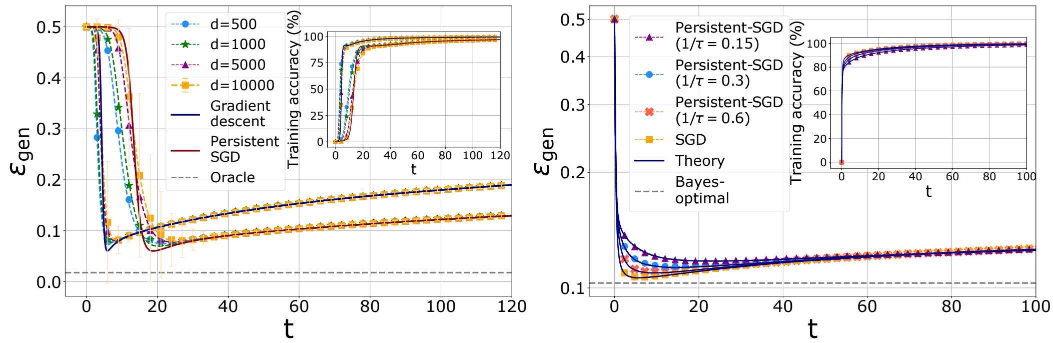
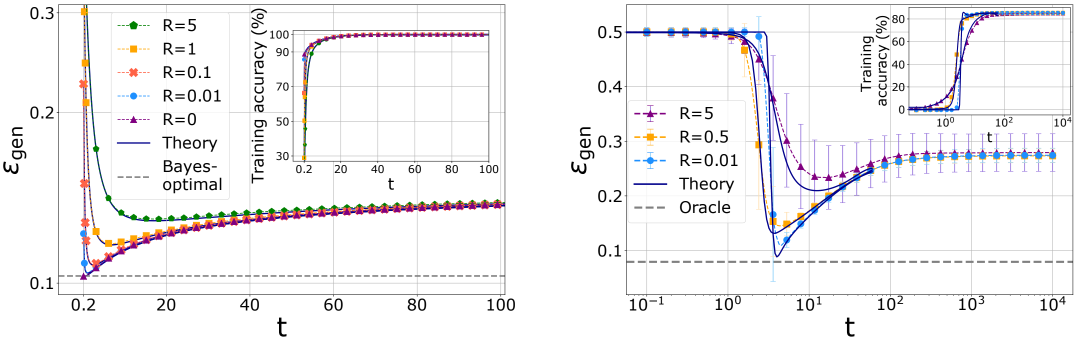


Figure 2.2.6 – **Left:** Generalisation error as a function of the training time for full-batch gradient descent and p-SGD with $1/\tau = \mathtt{b} = 0.3$ in the three-cluster model, at fixed $\alpha = 2$, $\Delta = 0.05$, $L = 0.7$ and $\lambda = 0$. The continuous lines mark the numerical solution of DMFT equations, the symbols represent simulations at $\mathrm{d}t = 0.2$, $R = 1$, and increasing dimension $d = 500, 1000, 5000, 10000$. Error bars are plotted for $d = 10000$. The dashed lines mark the oracle error (see supplementary material). **Right:** Generalisation error as a function of the training time for p-SGD with different activation rates $1/\tau = 0.15, 0.3, 0.6$ and vanilla SGD in the two-cluster model, both with $\mathtt{b} = 0.3$, $\alpha = 2$, $\Delta = 0.5$, $\lambda = 0$, $\mathrm{d}t = 0.2$, $R = 0.01$. The continuous lines mark the numerical solution of DMFT equations, while the symbols represent simulations at $d = 500$. The dashed lines mark the BO error from Chapter 1.2. In each panel, the inset displays the training accuracy as a function of time.



(a) **Two-cluster model** at fixed $\alpha = 2, \Delta = 0.5, \lambda = 0, \mathrm{d}t = 0.2, d = 500$. The dashed line marks the BO error computed in Chapter 1.2. The $y-$axis is cut for better visibility.

(b) **Three-cluster model** at fixed $\alpha = 3, \Delta = 0.1, \lambda = 0, \mathrm{d}t = 0.1, d = 1000$. The dashed line marks the oracle error whose computation can be found in Article 3.

Figure 2.2.7 – Generalisation error $\varepsilon_{\text{gen}}$ as a function of training time $t$ for full-batch GD at different values of the initialisation variance $R$. The continuous lines mark the numerical solution of DMFT equations, while the symbols represent simulations at finite dimension $d$.The insets show the training accuracy as a function of time.

value at $t = 0$. In the left panel of Figure 2.2.6, we show that while this produces a small discrepancy at intermediate times that diminishes with growing size, at long times the DMFT tracks perfectly the evolution of the algorithm. The right panel of Figure 2.2.6 summarises the effect of the characteristic time $\tau$ in the p-SGD, related to the typical persistence time of each pattern in the training mini-batch. When $\tau$ decreases, the p-SGD algorithm is observed to be getting a better early-stopping generalisation error and the dynamics gets closer to the usual SGD dynamics. As expected, the $\tau \to \mathrm{d}t/\mathtt{b}$ limit of the p-SGD converges to SGD. The SGD-inspired discretisation of the DMFT equations shows a perfect agreement with the numerics.

Figure 2.2.7 presents the influence of the weight norm at initialisation $R$ on the dynamics, for the two-cluster (left) and three-cluster (right) model. For the two-cluster case, the gradient descent algorithm with all-zeros initialisation "jumps" on the Bayes-optimal (BO) error at the first iteration as derived in Article 1, and in this particular setting the generalisation error is monotonically increasing in time. As $R$ increases the early stopping error gets worse. At large times all the initialisations converge to the same value of the error, as they must, since this is a full-batch gradient descent without regularisation that at large times converges to the max-margin estimator according to Rosset et al. (2004). For the three-cluster model we observe a qualitatively similar behaviour.

## 2.3 - Characterising the algorithmic noise of stochastic gradient descent

In this chapter, we show how to apply DMFT to characterise the late-time dynamics of stochastic gradient descent (SGD) and quantify its algorithmic noise. In order to decouple the effect of the SGD noise on the optimisation process from the effects of the architecture and the data structure, we focus on a single-layer network and a simple loss landscape.

In particular, we consider the prototypical supervised-classification problem, namely the binary classification of two balanced Gaussian clusters, introduced in Chapter 1.2 and further studied in the previous Chapter 2.2 as the *two-cluster setting*. We remind that the ANN is presented with a set of $n$ training examples in dimension $d$, $\boldsymbol{X} = (\boldsymbol{x}_1, .., \boldsymbol{x}_n)^\top \in \mathbb{R}^{n\times d}$, drawn i.i.d. from $\boldsymbol{x}_\mu \sim \mathcal{N}(y_\mu \boldsymbol{w}^*/\sqrt{d}, \Delta\, \boldsymbol{I}_d)$, $\boldsymbol{w}^* = (1, \dots, 1)^\top \in \mathbb{R}^d$, and $y_\mu = \pm 1$ with equal probability. We consider the thermodynamic limit, where $n, d \to \infty$ at fixed sample complexity $\alpha = n/d \sim \mathcal{O}_d(1)$. We consider a single-layer neural network that estimates the labels according to the linear rule $\hat{y}_\mu(\boldsymbol{w}) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x}_\mu/\sqrt{d})$. The weight vector $\boldsymbol{w} \in \mathbb{R}^d$ is learned via ERM of the loss

$$\mathcal{H}(\boldsymbol{w}) = \sum_{\mu=1}^{n} \ell\left(\frac{y_\mu}{\sqrt{d}} \boldsymbol{x}_\mu^\top \boldsymbol{w}\right) + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2. \tag{2.3.1}$$

In what follows, we will always consider the *squared hinge* cost function $\ell(h) = (h-\kappa)^2\, \Theta(\kappa - h)/2$, with $\kappa > 0$ and $\Theta(\cdot)$ indicating the Heaviside step function. We note that this particular loss is zero as soon as all the samples are correctly classified with a robustness ensured by the threshold $\kappa$, i.e., $y_\mu \boldsymbol{x}_\mu^\top \boldsymbol{w}/\sqrt{d} > \kappa$ for all the samples $\mu \in \{1, \dots n\}$. This choice is meant to reflect the fact that in real implementations the dynamics is usually stopped after the training error goes to zero. As customary in practical applications, we have added a ridge regularisation of strength $\lambda \geq 0$ that will stay fixed during training.

We study the stochastic dynamics of the SGD and p-SGD algorithms introduced in Section 2.2.2. We integrate the DMFT equations in this setting up to times when the dynamics has either reached a stationary state or stopped. We highlight the difference between these two possible scenarios. Indeed we remind that, following the analogy with constraint-satisfaction problems (CSPs) introduced in Chapter 1.1, the parameters space can be split into two regions:

- the under-parametrised or *unsatisfiable* (UNSAT) phase, where the network cannot achieve zero training error and the dynamics goes to a stationary state;
- the over-parametrised or *satisfiable* (SAT) phase, where the dynamics stops at one solution with zero training error due to Heaviside function in the squared hinge loss. Note that the SAT phase is realised only at $\lambda = 0$.

The SAT-UNSAT transition value $\alpha^*$ for this problem has been computed in Chapter 1.2.

In the UNSAT phase, computing the correlation and response functions of the network weights, we characterise the stationary state by defining an effective temperature $T_{\text{eff}}$ from the fluctuation-dissipation relation. From this relation, we then extrapolate numerically the value of $T_{\text{eff}}$, which relates correlation and response at stationarity and quantifies the magnitude of SGD noise as a function of the problem hyperparameters.

In the SAT phase, the extrapolated temperature approaches zero at large times. This result aligns with the intuitive picture that SGD implements a self-annealing procedure while navigating the loss landscape (Feng & Tu, 2021). In order to assess the noise magnitude in the SAT phase, we introduce an alternative measure that we can access both analytically – via DMFT – and from numerical simulations. We consider the dynamics of two copies of the system, starting from the same initial condition but subjected to two different realisations of the stochastic noise, i.e., two different histories of mini-batch sampling. We then track the average distance $d(t)$ between these two trajectories at time $t$ as they evolve in the weight space and when they finally land on a border of the zero-training-error region. We use this distance to quantify the noise of the SGD algorithm as a function of the problem hyperparameters. Remarkably, we show that a higher noise is associated to a smaller fraction of support vectors at the end of the training and therefore to a more robust solution (Xu et al., 2009).

We investigate the role of the various hyperparameters in the SAT and UNSAT phase, which could provide theoretically-informed guidance for practical implementation. We find a qualitative agreement in the behaviour of the two different measures of noise magnitude, $T_{\text{eff}}$ and $d(t)$, as a function of the hyperparameters.

## 2.3.1 . Noise characterisation in the UNSAT phase

We first discuss our results for the UNSAT phase, where the landscape has a unique minimum in which the SGD noise induces a non-equilibrium steady state. We use the DMFT equations derived in Chapter 2.2 to track the dynamics of the SGD, p-SGD, and Langevin algorithm introduced in Section 2.2.2. We integrate the DMFT equations via the numerical iterative method described in Chapter 2.2. The main quantities of interest for our analysis are the dynamical correlation function

$$C(t,t') = \frac{1}{N}\boldsymbol{w}(t)^\top \boldsymbol{w}(t'), \tag{2.3.2}$$

and the linear response function

$$R(t,t') = \lim_{\{H_j \to 0\}} \frac{1}{N}\sum_{j=1}^{N} \frac{\delta w_j(t)}{\delta H_j(t')}, \tag{2.3.3}$$

that have been introduced in the previous chapter. In the high-dimensional limit, these two-point functions concentrate to a deterministic value. In generic equilibrium stochastic processes, correlation and response are related by the fluctuation-dissipation theorem (FDT) (Cugliandolo, 2011):

$$R(t,t') = -\frac{1}{T}\partial_t C(t,t')\,\Theta(t-t'), \tag{2.3.4}$$

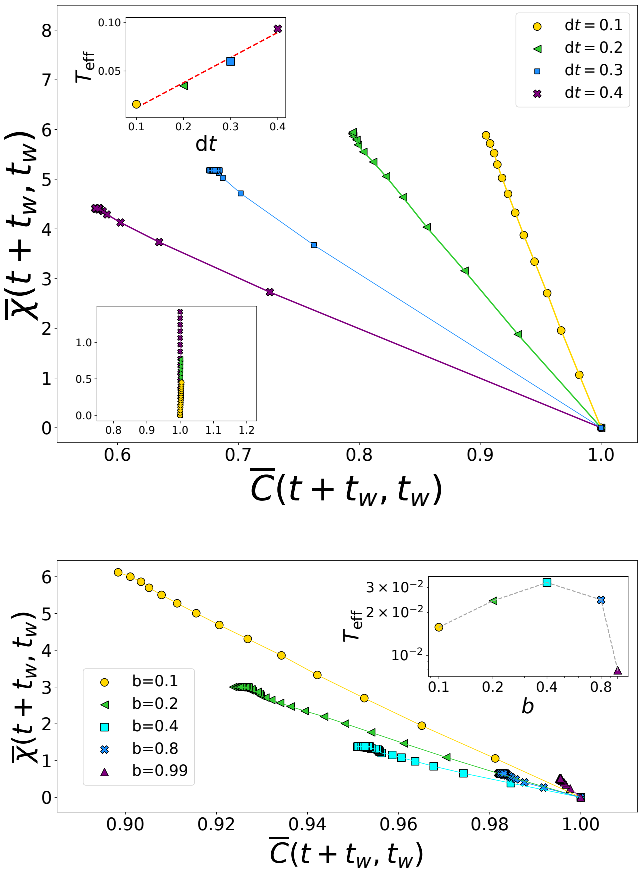


Figure 2.3.1 – **FDT plot for vanilla-SGD.** *Top panel:* the different curves represent different choices of learning rate $\mathrm{d}t = 0.1, 0.2, 0.3, 0.4$. The main plot is obtained in the UNSAT phase ($\mathtt{b} = 0.1, \alpha = 6, \Delta = 1, \lambda = 1, t_w =$), while the lower inset depicts the SAT phase ($\mathtt{b} = 0.5, \alpha = 2, \Delta = 0.5, \lambda = 0$). In the upper inset, we plot the behaviour of the effective temperature as extracted from the main plot. *Bottom panel:* the same analysis at fixed learning rate $\mathrm{d}t = 0.1$ and different batch sizes $\mathtt{b} = 0.1, 0.2, 0.4, 0.8, 0.99$.

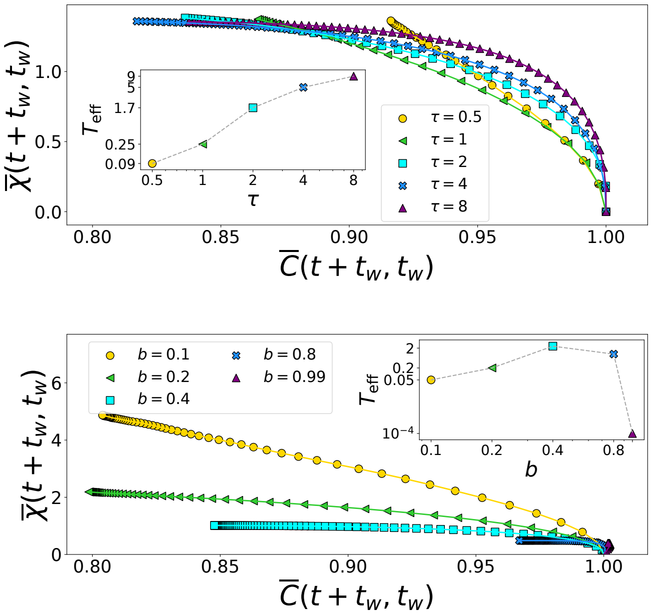


Figure 2.3.2 – **FDT plot for p-SGD.** We consider $\alpha = 8, \Delta = 1, \lambda = 1$, where the classification problem is UNSAT and we use $\mathrm{d}t = 0.05$. We consider different persistence times $\tau = 0.5, 1, 2, 4, 8$ and fixed batch size $\mathtt{b} = 0.3$ (*top panel*), and different $\mathtt{b} = 0.1, 0.2, 0.4, 0.8, 0.99$ and fixed $\tau = 2$ (*bottom panel*). The insets display the effective temperature, numerically estimated, as a function of the persistence time (*top panel*) and batch size (*bottom panel*).

where $T$ indicates the equilibrium temperature that enters in the stationary Gibbs measure. However, for a generic stochastic process that may be out of equilibrium, FDT does not hold and the stationary state is not given by the Gibbs measure. However one can define an effective temperature via FDT, which in general will be a function of $t$ and $t'$. This concept has proven useful across a variety of systems, ranging from glasses Cugliandolo (2011) to active matter Loi et al. (2008); Berthier & Kurchan (2013) [1].

It is convenient to work with the integrated response:

$$\chi(t,t') = \int_{t'}^{t} \mathrm{d}s\, R(s,t'), \tag{2.3.5}$$

that can be computed from the DMFT equations. By integrating both sides of Eq. (2.3.4), we obtain, for $t \geq t'$,

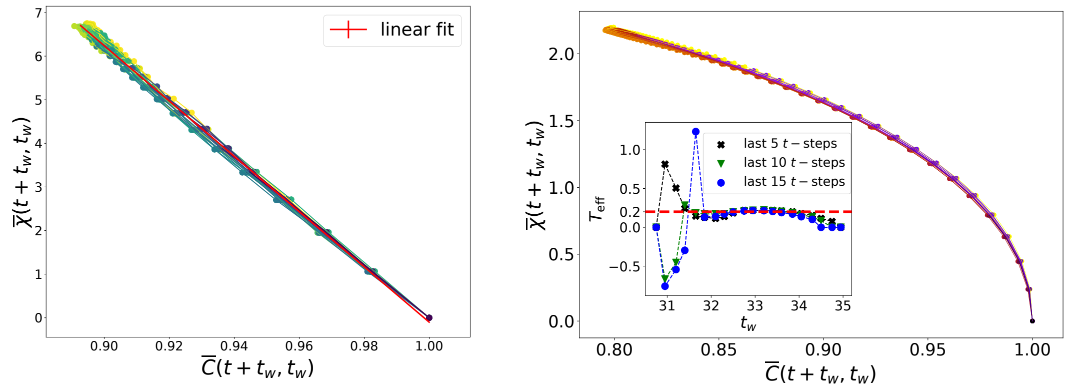
$$\bar{\chi}(t,t') = \frac{1}{T}\left(1 - \bar{C}(t,t')\right), \tag{2.3.6}$$

where we have defined $\bar{\chi}(t,t') = \chi(t,t')/C(t',t')$ and $\bar{C}(t,t') = C(t,t')/C(t',t')$. At equilibrium, (2.3.6) implies that the parametric plot of $\bar{\chi}$ versus (vs) $\bar{C}$ gives direct access to the equilibrium temperature. Therefore, we compute the integrated response $\chi(t+t_w,t_w)$ and the correlation function $C(t+t_w,t_w)$. We let the system evolve until a *waiting* time $t_w$ such that the stationary state has been reached. Then, at fixed $t_w$, we display the FDT plot $\bar{\chi}(t+t_w,t_w)$ vs $\bar{C}(t+t_w,t_w)$, parametrised by the time shift $t$. Figure 2.3.1 summarises our findings regarding the effective FDT for the vanilla-SGD algorithm. For large enough $t_w$, the relation between integrated response and correlation becomes linear and we can extrapolate numerically the effective temperature via (2.3.6).

**Extrapolation of the effective temperature —** We *define* the effective temperature $T_{\text{eff}}$ by plotting parametrically the rescaled integrated response $\bar{\chi}$ vs the rescaled correlation $\bar{C}$ and measuring the slope of this function in the stationary state, meaning for large $t'$ and $t$ and large time difference $t-t'$. We dub the corresponding plot as the FDT plot (Cugliandolo, 2011). The procedure that we have used to estimate the effective temperature is displayed in Figure 2.3.3. We treat the case of SGD and p-SGD separately since they are characterised by some important differences. Although SGD is a discrete-time algorithm, we use the definition of Eq. (2.3.6) to identify the effective temperature.

The integration of the response function is performed numerically from the DMFT equations, therefore in practice a time-discretisation is always needed. For SGD the variables $s_\mu(t)$ encoding the sampling process are i.i.d. at all times $t$. Therefore, in the stationary state, the FDT plot is a straight line and the effective temperature $T_{\text{eff}}$ is a constant at all time differences. The most efficient

[1] We note that the actual meaning of $T_{\text{eff}}$ extracted from the FDT theorem as a thermodynamic temperature is not granted and this is an open question in generic out-of-equilibrium systems, see (Cugliandolo, 2011) and (Loi et al., 2008) for more details. We do not address this issue here and use the definition of the effective temperature from FDT as a way to measure the magnitude of the noise of SGD in the stationary state.



(a) Vanilla-SGD. The red line represents the linear fit from which we compute the effective temperarue $T_{\rm eff}$. We have fixed $\alpha = 6, \mathtt{b} = 0.1, \mathrm{d}t = 0.1, \lambda = 1, \Delta = 1$ and we obtain $T_{\rm eff} = 0.015 \pm 0.00005$.

(b) p-SGD. The inset shows the estimate of $T_{\rm eff}$. For each value of $t_w$, we extrapolate the slope of the last $p$ steps in time $t$, for different values of $p = 5, 10, 15$. At large $t_w$, then $t = t_{\rm final} - t_w < \mathtt{b}\tau$ and the slope tends to zero as $t_w$ goes to $t_{\rm final}$. We extrapolate the slope at smaller $t_w$, in the regime where $T_{\rm eff}$ is constant, as the value at which the curves for different $p$ converge. We have fixed $\alpha = 8, \mathtt{b} = 0.2, \tau = 2, \mathrm{d}t = 0.05, \Delta = 1, \lambda = 1$ and we obtain $T_{\rm eff} \approx 0.2$.

Figure 2.3.3 – FDT plot with different curves representing different values of the waiting time $t_w$. Later waiting times are depicted with darker colors. Since the values of $t_w$ are such that the system is in the stationary state, all the curves are almost overlapping. At fixed $t_w$, each curve is a parametric plot with respect to $t \in [t_w, t_{\rm final} - t_w]$.

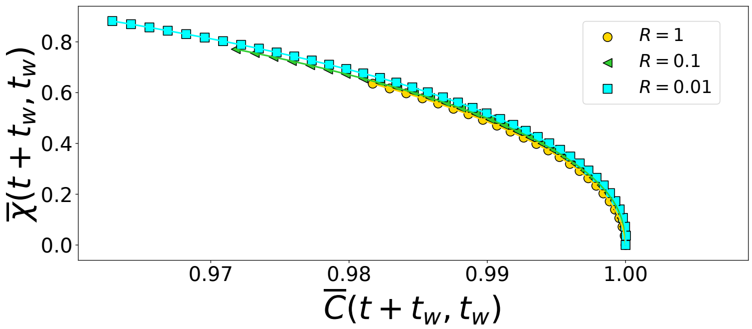
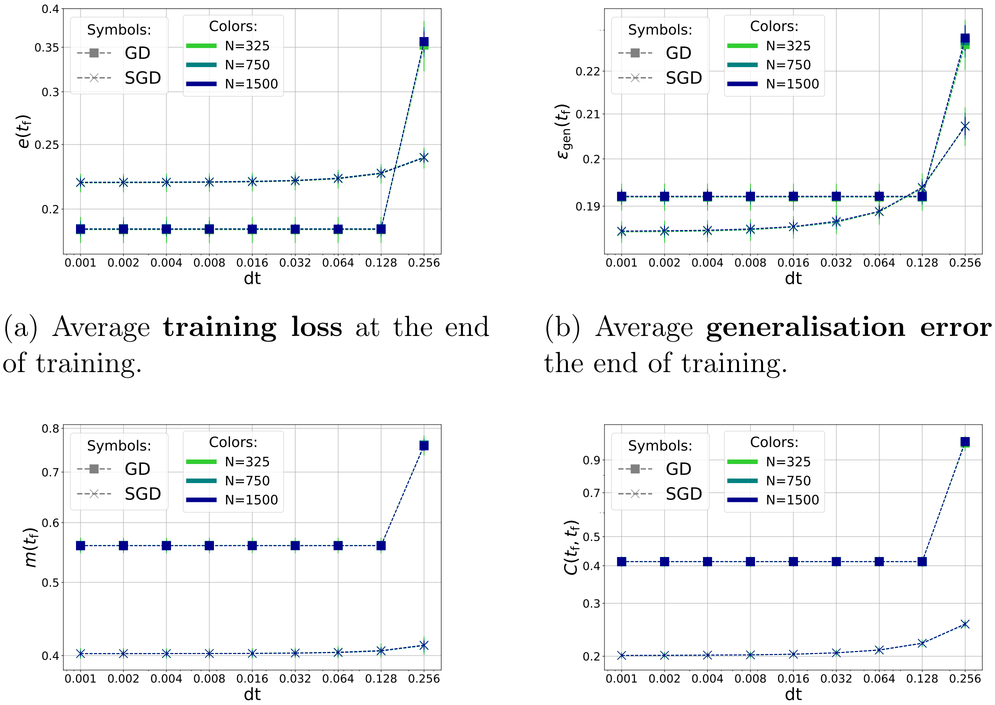


Figure 2.3.4 – FDT plot for p-SGD. We consider $\alpha = 8, \Delta = 1, \lambda = 1$, where the classification problem is UNSAT and we use $\mathrm{d}t = 0.01$ and look at different initialisation variances $R = 1, 0.1, 0.01$ and fixed $\mathtt{b} = 0.3$ and $\tau = 2$. In all cases we obtain an estimate of the effective temperature $T_{\rm eff} \sim 0.05$.



(a) Average **training loss** at the end of training.

(b) Average **generalisation error** at the end of training.

(c) Average **magnetisation** at the end of training.

(d) Average **squared norm** at the end of training.

Figure 2.3.5 – Numerical simulations on the behaviour of GD and SGD at the end of training, as a function of the learning rate d$t$. We fix the parameter space such that the problem lies in the UNSAT phase ($\alpha = 6$, $\lambda = 1$, $\Delta = 1$). The squares mark the behaviour of GD while the crosses represent SGD at batch size $\mathtt{b} = 0.3$. Different colors mark different system sizes: $N = 325$ (green), $N = 750$ (teal), $N = 1500$ (blue). The simulations are averaged over 150 realisations of the input data, the initialisation and the sampling noise.

way to extrapolate numerically the value of $T_{\text{eff}}$ is hence to fit a line to all points $\{(\bar{C}(t+t_w, t+w), \bar{\chi}(t+t_w, t+w))\}_{t\in[t_w, t_{\text{final}}-t_w]}$ for a collection of large enough waiting times $t_w$ and time differences $t$ ranging between $t_w$ and the final time $t_{\text{final}}$. This estimate is quite precise as can be seen from the example displayed in the left panel of Figure 2.3.3. We find that this slope is essentially constant for SGD, meaning that in the stationary state the algorithm is characterised by an effective FDT with a well-defined effective temperature that we can compute.

In the case of p-SGD, the autocorrelation between a sampling variable at different times decays exponentially with the time difference $t > 0$ at a rate that is given by the sum of the activation and deactivation rates $1/\mathtt{b}\tau$, i.e., $\langle s(t_w)s(t+t_w)\rangle - \mathtt{b}^2 = \mathtt{b}(1-\mathtt{b})\exp(-t/\mathtt{b}\tau)$. This behaviour is reflected by the fact that the effective temperature $T_{\text{eff}}$ is not constant with the time difference. Instead, it is lower at small $t$ given the higher correlation between samples in the gradient, while it goes to a constant at $t$ larger than the typical decay time $\mathtt{b}\tau$. This behaviour is clearly seen in the right panel of Figure 2.3.3 and in Figure 2.3.2. Therefore, in the case of p-SGD one should actually define the effective temperature depending on the time difference $T_{\text{eff}}(t)$ even in the stationary state. However, for simplicity we will refer to the constant $T_{\text{eff}}(t > \mathtt{b}\tau)$ as the effective temperature. This is motivated by the fact that $\mathtt{b}\tau$ is usually small compared to the observation time and we are always able to observe this regime.

The procedure through which we estimate $T_{\text{eff}}$ for p-SGD is detailed in the right panel of Figure 2.3.3. Finally, Figure 2.3.4 shows that the effective temperature does not depend on the initialisation variance $R$. Indeed, in the UNSAT phase the solution of the classification problem is unique and therefore the algorithm ends up rattling in the unique minimum with a noise strength given by $T_{\text{eff}}$.

The FDT plot depends both on the batch size and the learning rate. The top panel of Figure 2.3.1 shows that for vanishing learning rate the effective temperature of SGD approaches zero. However, for this particular problem we observe that the vanishing-learning-rate limit of SGD does not approach GD flow, as further illustrated by numerical simulations in Figure 2.3.5. Interestingly, the discrepancy between the vanishing$-\mathrm{d}t$ limit that we find numerically between GD and SGD in the classification problem under consideration will disappear in the regression problem studied in the next Chapter 2.4. This observation suggests that the behaviour of SGD in approximately-continuous time may depend non trivially on the problem landscape and is worth further investigation. Increasing the learning rate results in a noisier dynamics and a higher effective temperature. The behaviour of the effective temperature with batch size is more intriguing. Indeed, when we fix the learning rate and vary the batch size we observe a non-monotonic curve. For a batch size close to one, the dynamics tends to GD flow and the noise shrinks to zero. If the batch size is small – which here corresponds to the limit of sub-extensive mini batches – we again observe a decrease in the algorithmic noise. Quite surprisingly, the highest noise is attained at intermediate extensive batch sizes.

Next, we turn our focus to p-SGD. In Figure 2.3.2, we study the FDT plot for p-SGD changing the persistence time and the batch size. We observe that $T_{\text{eff}}$ is monotonically increasing with the persistence time. The physical interpretation of this property is rather clear: if the persistence time is far from the SGD limit, i.e., $\tau \gg \mathrm{d}t/\mathtt{b}$, the system ends up on a local minimum of a *partial* loss, namely the loss function evaluated only on the samples belonging to the current mini batch. Therefore, the system is fitting well such subset of samples. Conversely when $t - t' \gg \tau\mathtt{b}/(1-\mathtt{b})$ the dynamics has seen many mini-batches beyond the one which was there at $t'$. When a mini-batch is renewed the system finds itself in a very atypical, random-like configuration with a large stochastic gradient. This effect produces a high effective temperature that stays constant as the FDT plot becomes linear at large $t - t'$. Note that, in the SGD limit recovered at $\tau \approx \mathrm{d}t/\mathtt{b}$, the system never has the possibility to equilibrate in the partial loss but this comes with the effect that the stochastic gradient does not have big jumps as with p-SGD with finite $\tau$. This idea is similar to what has been done in active systems (Mandal & Sollich, 2021), which strengthens the connection between SGD and other types of out-of-equilibrium systems. At fixed persistence time, we observe a non-monotonic effective temperature as a function of the batch size $\mathtt{b}$, consistently with our results for vanilla-SGD.

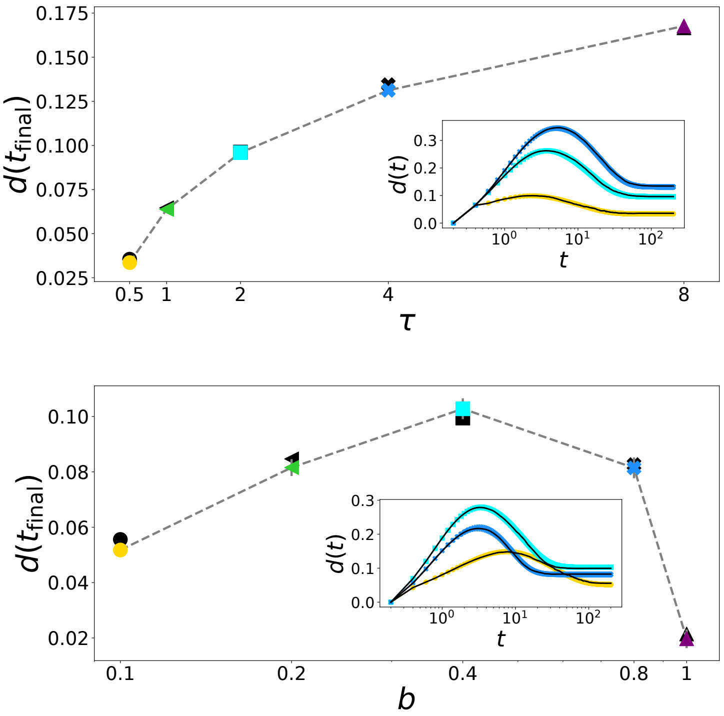


Figure 2.3.6 – Average final distance $d(t_{\text{final}})$ between two copies of the p-SGD dynamics. *Top panel:* We plot $d(t_{\text{final}})$ as a function of the persistence time $\tau$ at fixed batch size $\mathtt{b} = 0.3$. In the inset we show the time evolution of the distance $d(t)$ from numerical simulations at $\tau = 0.5$ (yellow), $\tau = 2$ (cyan), $\tau = 4$ (blue). *Bottom panel:* We plot $d(t_{\text{final}})$ as a function of the batch size at fixed persistence time $\tau = 2$. In the inset we show the time evolution of the distance $d(t)$ from numerical simulations at $\mathtt{b} = 0.1$ (yellow), $\mathtt{b} = 0.4$ (cyan), $\mathtt{b} = 0.4$ (blue). In both panels, the black symbols in the main plots and the black lines in the insets mark the theoretical prediction from DMFT. The learning rate is $dt = 0.2$ in both DMFT and simulations. The other parameters are fixed such that the classification problem lies in the SAT phase: $\alpha = 0.5, \Delta = 0.5, \lambda = 0, \kappa = 1$.

## 2.3.2 . Noise characterisation in the SAT phase

We now consider the characterisation of the effective noise in the SAT phase, which is more interesting for practical applications since artificial feed-forward neural networks typically lie in this regime. In order to quantify the noise magnitude in the SAT phase, we consider the rescaled distance (root mean square displacement) $d(t) = \|\boldsymbol{w}^1(t) - \boldsymbol{w}^2(t)\|_2/\sqrt{d}$ between two realisations ($\boldsymbol{w}^1$ and $\boldsymbol{w}^2$) of the stochastic dynamics, starting at the same initialisation point $\boldsymbol{w}^1(t = 0) = \boldsymbol{w}^2(t = 0)$ and subjected to two different noise realisations $\boldsymbol{s}^1(t) \neq \boldsymbol{s}^2(t)$. At the end of training, the distance $d(t = t_{\text{final}})$ quantifies the spread of the solutions found by different runs of the algorithm. This quantity can be computed analytically again via DMFT (see in particular Sompolinsky et al. (1988); Crisanti & Sompolinsky (2018); Krishnamurthy et al. (2020), where this procedure gives access to the Lyapunov exponent of the

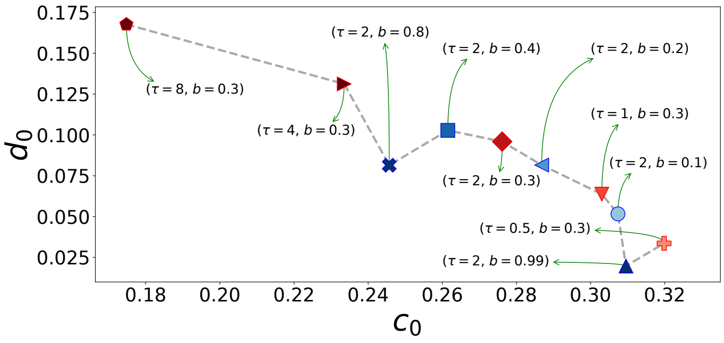
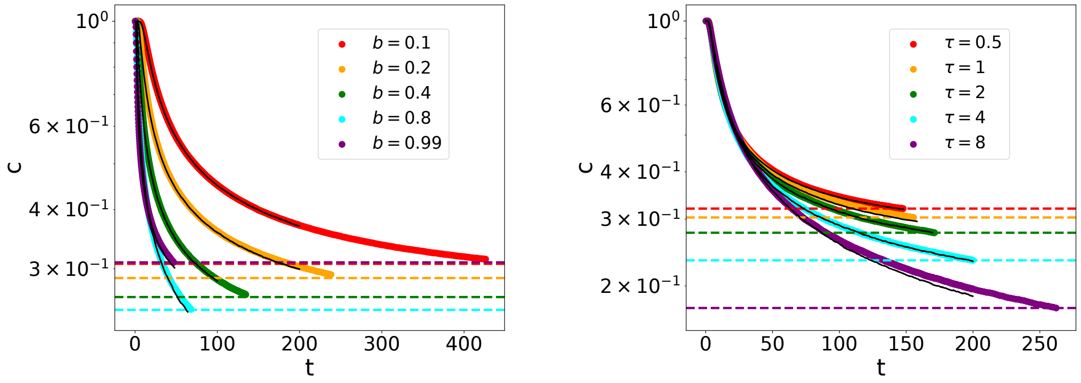


Figure 2.3.7 – Average distance $d_0$ between two replicas as a function of the average fraction of support vectors $c_0$. Each symbol represents a different choice of the persistence time $\tau$ and the batch size $b$ in the p-SGD algorithm. Darker colors correspond to higher values of $b$ and $\tau$. The learning rate is fixed to $dt = 0.2$. The other parameters are fixed such that the problem lies in the SAT phase: $\alpha = 0.5, \Delta = 0.5, \lambda = 0, \kappa = 1$.



(a) Different values of the batch size at fixed persistence time $\tau = 2$.

(b) Different values of the persistence time at fixed batch size $\mathsf{b} = 0.4$.

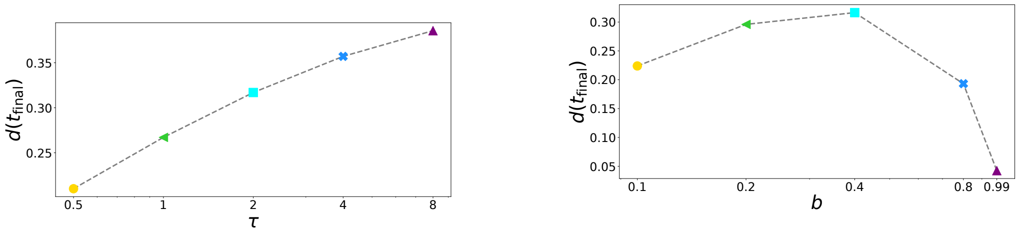
Figure 2.3.8 – Fraction of support vectors $c$ as a function of time $t$. The horizontal dashed lines mark the stopping times at which we have computed the value of $c_0$. The average energy has reached $10^{-9} - 10^{-10}$ for all parameter settings. The full lines correspond to the DMFT solution.

underlying chaotic dynamics in recurrent neural networks).

The starting point of the analysis is the (coupled) dynamical partition function:

$$Z_{\text{dyn}} = \mathbb{E}_{\boldsymbol{w}^{(0)}} \int_{\boldsymbol{w}^{1,2}(0)=\boldsymbol{w}^{(0)}} \mathcal{D}\boldsymbol{w}^1(t)\mathcal{D}\boldsymbol{w}^2(t) \prod_{j=1}^{N} \prod_{a=1,2} \delta\left(\dot{w}_j^a(t) + \tilde{\partial}_{w_j^a}^{\boldsymbol{s}^a(t)} \mathcal{H}(\boldsymbol{w}(t))\right), \quad (2.3.7)$$

that allows to compute the correlation and response functions of the coupled system of replicas, both initialised at $\boldsymbol{w}^{(0)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d R)$, $R > 0$. We have denoted by $\tilde{\partial}_{w_j^a}^{\boldsymbol{s}^a(t)}$ the approximate derivative of the empirical risk with respect to $w_j^a$, computed



(a) Changing persistence time $\tau = 0.5, 1, 2, 4, 8$ at fixed batch size $\mathtt{b} = 0.3$.

(b) Changing batch size $\mathtt{b} = 0.1, 0.2, 0.4, 0.8, 0.99$ at fixed persistence time $\tau = 2$.

Figure 2.3.9 – Average final distance $d(t_{\mathrm{final}})$ between two copies of the p-SGD dynamics. The results are obtained by numerical simulations at size $N = 750$, averaged over 100 seeds. The other parameters are the same as in Figure 2.3.2, such that the problem lies in the UNSAT phase: $\alpha = 8, \Delta = 1, \lambda = 1$. The learning rate is $\mathrm{d}t = 0.05$.

only on the samples selected by $\boldsymbol{s}^a(t)$. The details of this computation do not differ significantly to the on performed in Chapter 2.2 and can be found in Article 4. The bottom inset of Figure 2.3.1 shows that in the SAT phase the dynamics implements an automatic self-annealing procedure, and we obtain a zero effective temperature in the zero-loss region, reached at the end of training. This observation is obvious in problems where a “lake”, i.e., a large connected set, of solutions is found at late times so that the dynamics stops. However, the way in which the self annealing is produced drastically affects the learning trajectory in more complex problems. Indeed, in problems like phase retrieval (Fienup, 1982) where there is no such lake of solutions, but just one global minimum (modulo some symmetry) and a proliferation of local minima, the self-annealing property appears to be crucial to achieve good generalisation, as we will further discuss in the next Chapter 2.4.

In Figure 2.3.6 we plot the final distance $d(t = t_{\mathrm{final}})$ between two replicas of p-SGD as a function of the batch size and learning rate. We use $d(t_{\mathrm{final}})$ to probe the algorithmic noise. Indeed, we expect that noisier regimes are associated to a higher degree of landscape exploration, resulting in a greater divergence between two replicated trajectories. We find that the behaviour of $d(t_{\mathrm{final}})$ in the SAT phase mirrors the one of the effective temperature $T_{\mathrm{eff}}$ in the UNSAT phase: the effective noise increases with the persistence time and is non-monotonic with the batch size. This is confirmed also by Figure 2.3.9, where the same quantity is computed in the UNSAT phase. To improve the characterisation of the endpoints of the dynamics in the SAT phase, we also compute the size of the *support vectors set* (Boser et al., 1992) in the following way.

In the case of GD-flow dynamics, defined in continuous time with $\mathrm{d}t \to 0^+$, the endpoint of training lies on the border of the lake of solutions. We consider the number of unsatisfied constraints, i.e., misclassified samples, as it evolves with time. The long time limit of this quantity is finite, since all solutions $\boldsymbol{w}$ lying on this border marginally classify some samples, i.e., $y_\mu \boldsymbol{w}^\top \boldsymbol{x}_\mu = \kappa$ for some $\mu \in \{1, \dots, n\}$. These marginally classified samples are called *support vectors*. We indicate the (rescaled) size of the set of support vectors as $c = |\{\boldsymbol{x}_\mu,\ \mu = 1, \dots, n,\ \text{such that}\ y_\mu \boldsymbol{w}^\top \boldsymbol{x}_\mu =$

$\kappa\}|/n$.

In the case of p-SGD, the flow limit remains well defined, since the algorithm admits a continuous-time description, as also shown in the next Chapter 2.4. We can therefore apply the same definition as for GD. However, while for the flow limit there are no ambiguities, in practice, integrating the dynamics requires a finite learning rate. As done in Hwang & Ikeda (2020), for full-batch GD we fix a threshold on the stochastic gradient, rescaled by the batch size: $\| \tilde{\nabla}^B_{\boldsymbol{w}} \mathcal{H} \|^2_2 /bn \leq 10^{-10}$. We compute the values of $c(t)$ and $d(t)$ as soon as this threshold is reached and we take them as the proxy for their limiting values $c_0 = \lim_{t\to\infty} c(t)$, $d_0 = \lim_{t\to\infty} d(t)$. This procedure is depicted in Figure 2.3.8 and further explained in Article 4.

The physical meaning of the size of the support vectors is related to the description of the local density of solutions at the endpoint of the dynamics. If $c(\boldsymbol{w})$ is close to one, the solution $\boldsymbol{w}$ lies in a narrow corner of the solutions space. Conversely, a low value of $c$ is indicative of a wide region of the solutions space. In other words, the dynamics has landed on a "shore" with a wide lake of solutions just in front of it. In the latter case, one may expect the solution to be more robust to perturbations.

In Figure 2.3.7, we illustrate the behaviour of $d_0$ as a function of $c_0$. We observe that a larger algorithmic noise leads to a smaller $c_0$. Therefore, SGD brings the system to wider (or flatter) regions of the lake of solutions. Note that this notion of "flatness" differs from the one proposed in the recent literature on wide minima, see for instance Pittorino et al. (2021). Indeed, in the present case the lake of solutions is unique and the width is encoded by the number of support vectors. The smaller is $c$, the larger the density of solutions close to the endpoint of the dynamics.

**Insights on more complicated architectures and data structures —** In more complicated settings, it is reasonable to expect that the under-parametrised regime is glassy with many local minima. This is well known in non-convex continuous CSPs (Franz et al., 2017, 2019b), where the high dimensional limit is characterised by replica symmetry breaking (Mézard et al., 1987). In this case, one naturally expects that pure GD dynamics goes to a stationary state where the system ages and drifts on a landscape of marginally stable minima (Cugliandolo & Kurchan, 1993b) [2]. The aging dynamics is controlled by an effective temperature that encodes for the roughness of the underlying landscape.

However, it is well known that driving systems governed by glassy relaxation stops aging dynamics (Kurchan, 1997; Berthier & Kurchan, 2013). Indeed, aging is essentially due to the progressive annealing in the landscape. More annealed systems surf on stationary points that are more and more stable and as a consequence their dynamics slows down. However, if the system is driven, the dynamics is renewed and aging stops. Based on the above considerations we may argue that both in the stationary state of the under-parametrised regime and in the early-time over-parametrised regime, the noise of SGD is a mixture of the noisy dynamics induced by the roughness of the underlining loss-landscape and the stochasticity induced by the algorithm itself. In this setting, it is useful to compare the dynamics to the one of complex driven systems such as low-temperature amorphous solids under deformation: the noise induces activated jumps between local minima, which can

[2] This has been found in numerical simulations in Baity-Jesi et al. (2018).

get further destabilised resulting in an avalanche dynamics (Nicolas et al., 2018). This may lead to power-law distributed jumps and connect with recent literature on Lévy flights (Simsekli et al., 2019). Further investigation on more complex models is needed to asses this phenomenology.

## 2.4 - The interplay of algorithmic noise and landscape roughness in the sign-retrieval problem

As already discussed in previous chapters, algorithms based on gradient descent (GD) are the workhorses of many machine learning applications involving the optimisation of a high-dimensional non-convex loss function. In particular, stochastic gradient descent (SGD) has proved to be extremely efficient in navigating complex loss landscapes. However, despite its practical success, the theoretical understanding of the reasons behind the good generalisation properties of the algorithm remains sparse. Empirical evidence suggests that the interplay between the optimisation algorithm and the landscape is crucial to achieve good performances. While the practical success of SGD compared to GD is rather generally accepted, it is still far from clear what is really the key factor responsible for this. Cases where the superiority of SGD with respect to GD was shown theoretically are sparse (Abbe & Sandon, 2020; HaoChen et al., 2020).

Learning theory and computer science usually proceed in a manner that makes minimalist assumptions on the data distribution. Statistical physics usually takes a complementary way of understanding well prototypical settings that capture the essence of the question (see also the discussion in the *Motivation and background* chapter). This is the path we take in this chapter: we compare the behaviour of GD-based algorithms on a prototypical choice of data and learning model leading to a high-dimensional and non-convex landscape.

Specifically, we consider the problem of phase retrieval where the task consists in recovering an unknown signal from a set of observations – the absolute value of the signal's projections onto measurement vectors. This problem appears in a series of applications, including optics Walther (1963); Millane (1990), acoustics Balan et al. (2006), and quantum mechanics Corbett (2006). We will consider the real version of the problem – therefore more appropriately named *sign* retrieval – where the measurements are i.i.d. Gaussian vectors, and the usual linear sample complexity regime in the thermodynamic limit, with $n$ measurements in dimension $d$ and $\alpha = n/d \sim \mathcal{O}_d(1)$, $n, d \to \infty$.

We view the sign retrieval as a prototypical example of a simple single-layer ANN where the measurement vectors correspond to the input samples, and the signal corresponds to the teacher-network weights. The measurements then represent the output labels. We stress that it is not the goal of this work to provide a competitive algorithm for the sign retrieval. In the setting considered in this chapter (i.e., i.i.d. Gaussian inputs and teacher-produced labels) it was conjectured that the AMP algorithm cannot be beaten in the large size limit Barbier et al. (2019). Instead, the main goal of this chapter is to study the performance of gradient-based algorithms and the loss landscape of the sign retrieval problem serves us as a high-dimensional intrinsically non-convex prototype having multiple spurious minima and only one solution (with a $\mathbb{Z}_2$ symmetry) leading to perfect generalisation error.

We note that the landscape of the sign retrieval problem is somewhat different than the one of deep neural networks, that are highly overparametrised and present entire regions of solutions with zero training error and a good generalisation. Consequences of this difference and thus relevance of the present study for learning with state-of-the-art DNNs is left for future work. Instead this chapter investigates the performance of gradient-based algorithms in an archetypal non-convex high-dimensional setting providing a benchmark to assess the role played by stochasticity in non-convex optimisation problems in general.

**Further related works —** The loss landscape complexity of this problem was studied using the Kac-Rice method in Maillard et al. (2020). However, bringing this analysis to concrete results seemed to be technically challenging. Signal recovery in this problem was studied from the information-theoretic point of view and using AMP algorithms that are considered optimal among all polynomial algorithms for this case (Barbier et al., 2019; Mondelli & Montanari, 2018; Ma et al., 2019). In particular it is known that while information-theoretically zero generalisation error can be reached for $\alpha > 1$, the AMP algorithm is able to do so for $\alpha > 1.13$.

The performance of gradient descent for phase retrieval is worse than the one of AMP in terms of sample complexity and also harder to analyse. In practice, one often uses GD initialized spectrally (Dong et al., 2019), i.e., in the eigenvector corresponding to the leading eigenvalue of a suitable matrix constructed from the labels and the measurement vectors (Luo et al., 2019). Such spectral initialisation is also motivating our use of warm start that is mimicking it. Concerning randomly initialized gradient descent, Chen et al. (2019) showed that gradient descent needs a training set of size $\sim \mathcal{O}(d\,\text{poly}(\log d))$ to retrieve the hidden signal. Several other works in computer science consider gradient descent-type algorithms for phase retrieval requiring $\mathcal{O}(d\text{poly}(\log d))$ samples (Ma et al., 2018).

The analysis carried out in Mannelli et al. (2020b) suggests that the randomly-initialized algorithm can achieve perfect generalisation with much lower linear sample complexity. Authors of Cai et al. (2021) then show that linear (with unspecified large constant) sample complexity is achievable with randomly initialized gradient descent for a suitably chosen loss function. Finally Mannelli et al. (2020c) have shown that over-parametrisation can bring the sample complexity of randomly initialized gradient descent down to $\alpha = 2$.

While in the present work we will not be considering overparametrisation, we are interested in performance of gradient-based algorithms for similarly small sample complexity $\alpha$. We will be investigating several gradient-based algorithms and judge their performance by the number of samples they require for recovery of the signal. The fewer samples the better. This is why we focus on the regime of $\alpha = \mathcal{O}_d(1)$.

The online SGD for phase retrieval has been studied, e.g., in Tan & Vershynin (2019). A theoretical understanding of the performance of (multi-pass) SGD at small sample complexity requires taking into account the full trajectory of the algorithm which is challenging and done in this chapter. The interested reader is referred to Dong et al. (2022) for a recent comprehensive review on the phase retrieval problem.

## 2.4.1 . Introduction to the task

We study the supervised learning problem of recovering a $d-$dimensional real-valued vector $\boldsymbol{w}^* = \{w_1^*, \ldots, w_N^*\}$ from a set of $n = \alpha d$ real-valued noiseless measurements $\boldsymbol{x}_\mu = \{x_{\mu 1}, \ldots, x_{\mu d}\}$ of dimension $d$. We consider the signal $\boldsymbol{w}^*$ to be extracted with the uniform measure on the $d$-dimensional hyper sphere $\|\boldsymbol{w}^*\|_2^2 = N$. We take the components of the vectors $\boldsymbol{x}_\mu$ to be i.i.d. Gaussian random variables with zero mean and unit variance. The non-linear measures of the signal vector $\boldsymbol{w}^*$ are encoded in the labels

$$y_\mu = \left| \frac{1}{\sqrt{d}} \boldsymbol{x}_\mu^\top \boldsymbol{w}^* \right|, \qquad \forall \mu = 1, \ldots, n. \tag{2.4.1}$$

We note that in applications the complex-valued phase retrieval is more relevant, yet for the purpose of our study, which is studying the performance of the gradient-based algorithms, the real-valued version is sufficiently rich.

We consider learning with a single-layer neural network by the minimisation of the empirical risk

$$\mathcal{H}\left(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}\right) = \sum_{\mu=1}^{n} \ell\left(h_\mu, h_\mu^*\right), \tag{2.4.2}$$

where $\ell$ is a cost function having a global minimum at $h_\mu = h_\mu^*$ and we have defined

$$h_\mu = \frac{1}{\sqrt{d}} \boldsymbol{x}_\mu^\top \boldsymbol{w}, \qquad h_\mu^* = \frac{1}{\sqrt{d}} \boldsymbol{x}_\mu^\top \boldsymbol{w}^*. \tag{2.4.3}$$

In what follows we consider a loss of the form:

$$\ell(h, h_0) = \frac{1}{4}(h^2 - h^{*2})^2. \tag{2.4.4}$$

Note that the empirical risk depends on the labels $y_\mu$ only through $h_\mu^*$. We consider a particular regularisation of the weights where the training dynamics of $\boldsymbol{w}(t)$ is constrained on the hyper-sphere. In Article 5, we show that our results hold in a qualitative same manner for the more standard ridge regularisation.

We analyse the dynamics of the training algorithms already introduced in Chapter 2.2, namely GD, SGD, p-SGD, and the Langevin algorithm. In order to explore the energy landscape more thoroughly we consider here, next to the usual random initialisation, informed/warm initialisations. We initialize the weight vector as follows:

$$\boldsymbol{w}(t=0) = m_0\, \boldsymbol{w}^{(0)} + c\, \boldsymbol{z} \in \mathbb{R}^d, \tag{2.4.5}$$

where $m_0 > 0$ is (on average) the initial projection of the weight vector onto the signal, i.e., the average *magnetisation*

$$m(t) = \frac{1}{d} \boldsymbol{w}(t) \cdot \boldsymbol{w}^* \tag{2.4.6}$$

at time $t = 0$. The components of $\boldsymbol{z}$ are i.i.d. standard Gaussian variables and the coefficient $c$ is such that $|\boldsymbol{w}(t=0)|^2 = d$. Note that the warm initialisation breaks the $\mathbb{Z}_2$ symmetry of the problem. Therefore, in the following $m(t) \in (0, 1]$, $\forall t$.

We stress here that while in learning we are usually concerned with the test error/performance, in the setting considered here (under the spherical constraint) the test error is monotonic in the magnetisation (see Article 5 for a simple argument). Thus, in the following we directly use the magnetisation as a measure of accuracy. This warm initialisation can be thought of as a proxy for algorithms where GD (or its variants) is run after the weights have been spectrally initialised, i.e. , using the principal eigenvalue of a given pre-processing matrix as initial guess for the weights. Spectrally initialised GD is used in a range of applications, see, e.g., Dong et al. (2019), as well as studied theoretically, see, e.g., Mondelli et al. (2020).

## 2.4.2 . Discussion on the training dynamics

We apply DMFT to obtain a closed-form characterisation of the flow dynamics of the training algorithms presented in Section 2.2.2 for the sign retrieval problem in the high-dimensional limit. The derivation follows the one of Chapter 2.2, with the only difference that in this case we want to enforce the spherical constraint $\|\boldsymbol{w}\|^2 = d$ at all times of the dynamics, instead of the ridge regularisation. This is equivalent to a projection on the sphere at each iteration, which is how we implement the numerical simulations and can be modeled analytically via a Lagrange multiplier that plays the exact same role of the ridge strength $\lambda$ but is now time dependent. The training dynamics in the sign retrieval therefore is given by

$$\dot{w}_j(t) = -\left[\sum_{\mu=1}^{n} s_\mu(t)\ell'\left(h_\mu, h_{\mu^*}\right)\frac{x_{\mu,j}}{\sqrt{d}} + \lambda(t)w_j(t)\right] + \varsigma_j(t), \qquad \forall j = 1, ...d., \tag{2.4.7}$$

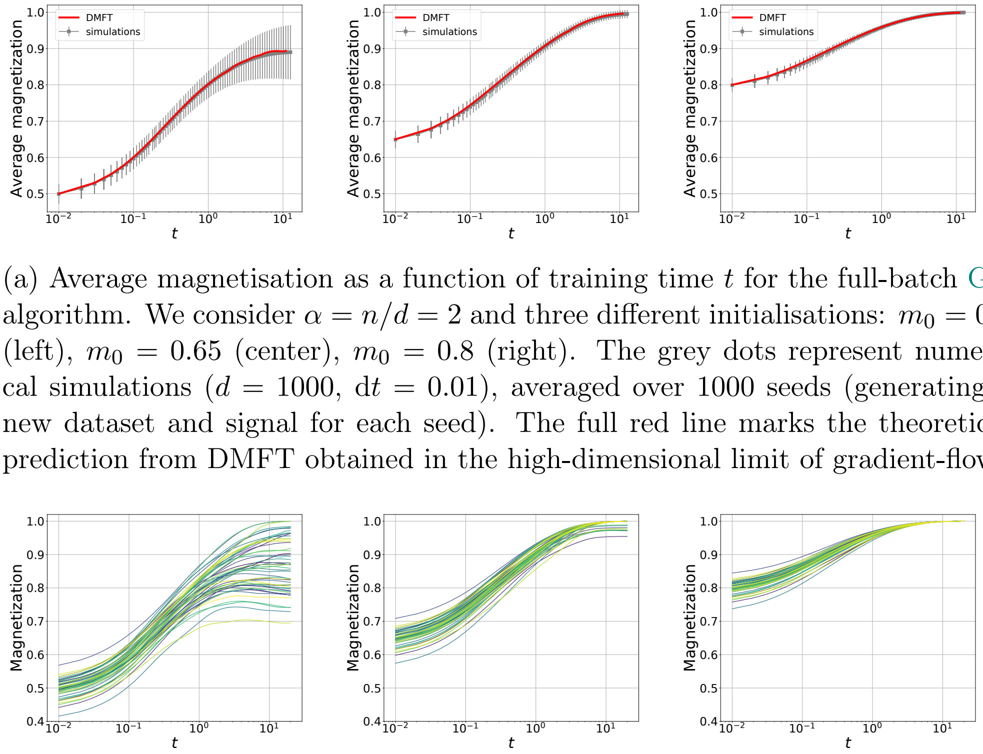
and we remind that $s_\mu(t)$ encodes the mini-batch sampling protocol at fixed batch size $\mathtt{b}$ and $\boldsymbol{\zeta}(t)$ denotes the Langevin noise at temperature $T$. Gradient flow is recovered by setting $\mathtt{b} = 1$ and $T = 0$. The detailed definition of the stochastic algorithms is provided in Section 2.2.2. We need an additional equation in the DMFT to describe the evolution of the Lagrange multiplier, which reads

$$\lambda(t) = -\alpha\langle s(t)r(t)\ell'(r(t), h^*)\rangle + T, \tag{2.4.8}$$

obtained enforcing the spherical constraint $\mathrm{d}\left(\sum_{j=1}^{d} w_j^2\right)/\mathrm{d}t = 0$ and by applying Itô's formula to Eq. (2.4.7). We do not report again the derivation here, we instead refer to Article 5 for the full computation.

In this section, we discuss our findings on the dynamics of the gradient-based algorithms under consideration. We compare the results from simulations to the DMFT theoretical prediction. This analysis sheds light on how stochasticity helps to navigate the loss landscape and on the impact of the different hyperparameters, notably the batch size $\mathtt{b}$, temperature $T$, and persistence time $\tau$, on the test performance.

**The trapping landscape —** Figure 2.4.1 illustrates the performance of gradient descent starting from three increasing initialisations: $m_0 = 0.5$ (left), $m_0 = 0.65$ (center), and $m_0 = 0.8$ (right) at $\alpha = 2$, i.e. , number of samples twice the dimension.

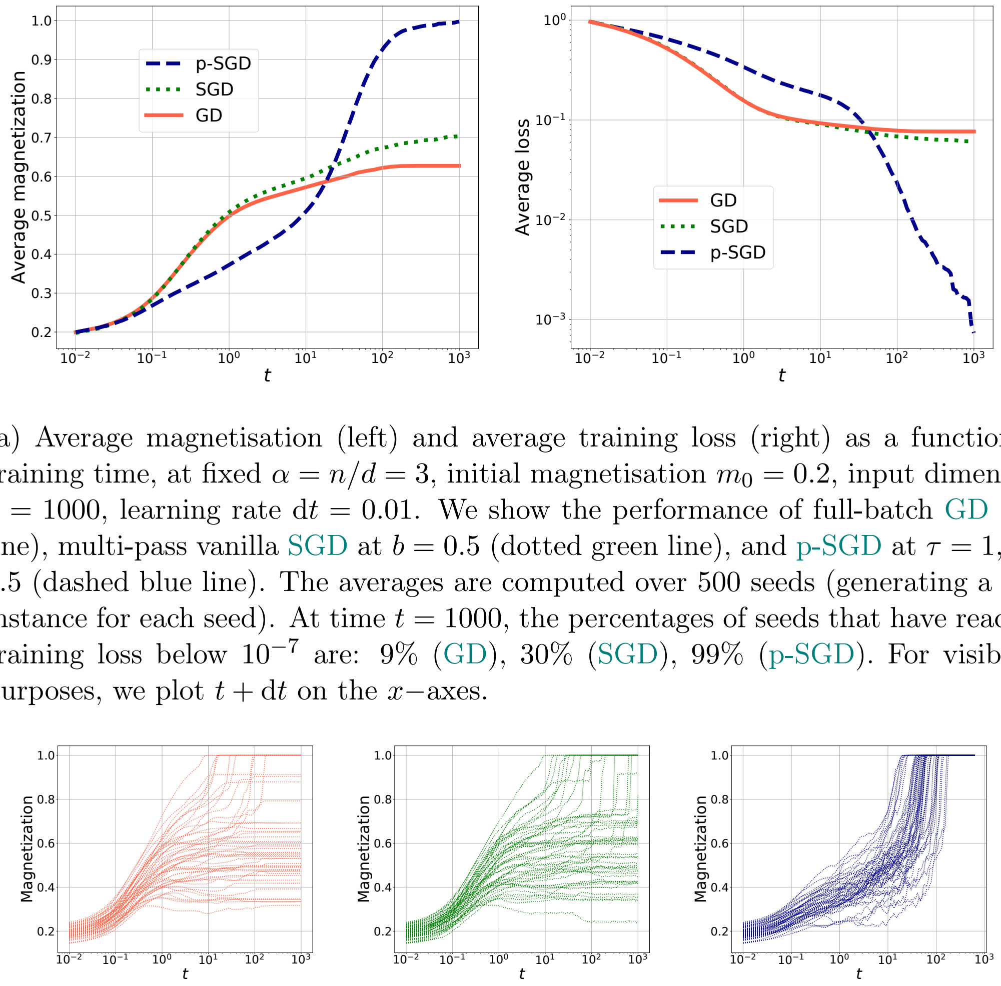


(a) Average magnetisation as a function of training time $t$ for the full-batch GD algorithm. We consider $\alpha = n/d = 2$ and three different initialisations: $m_0 = 0.5$ (left), $m_0 = 0.65$ (center), $m_0 = 0.8$ (right). The grey dots represent numerical simulations ($d = 1000$, $\mathrm{d}t = 0.01$), averaged over 1000 seeds (generating a new dataset and signal for each seed). The full red line marks the theoretical prediction from DMFT obtained in the high-dimensional limit of gradient-flow.

(b) We fix the landscape by fixing the dataset and we show 50 instances of the magnetisation as a function of time, for different realisations of the noise vector $\boldsymbol{z}$, at $m_0 = 0.5$ (left), $m_0 = 0.65$ (center), $m_0 = 0.8$ (right). For visibility purposes, we plot $t + \mathrm{d}t$ on the $x-$axes.

Figure 2.4.1 – Full-batch GD gets trapped by the roughness of the landscape: theory VS simulations.

In the lower panels (Figure 2.4.1b), we plot the magnetisation for different seeds – corresponding to different realisations of the noise vector $\boldsymbol{z}$ defined in Eq. (2.4.5) – with a dataset drawn at random and fixed. The evolution of different instances from simulations is thus probing the very same loss-landscape, the figure then highlights the complexity of the landscape. First, we observe that a warm start is not enough to reach perfect recovery. This suggests that the landscape is very rough, with multiple local minima at all heights. Indeed, we see that gradient descent can get stuck even very close to the global minimum at $m = 1$. From the right panel of the figure, we see that at time $t \sim 10$ all seeds initialised with magnetisation $m_0 = 0.8$ have achieved perfect recovery $m = 1$. However, the left and center panels show that some seeds starting at $m_0 < 0.8$ and reaching $m = 0.8$ only at $t > 0$ can get stuck for long times. Hence we deduce that the topological complexity of the landscape is such that some regions of the weights space can trap the dynamics even if they are closer to the signal than other regions that do not trap the dynamics. We observe that a more informed initialisation does not guarantee a better generalisation. This can be further seen comparing the left panel to the central one. Indeed, we find that some seeds initialised at $m_0 > 0.6$ are stuck at $m < 1$ at time $t \sim 10$, while some seeds starting at $m_0 < 0.6$ have already reached perfect generalisation. Consequently, in



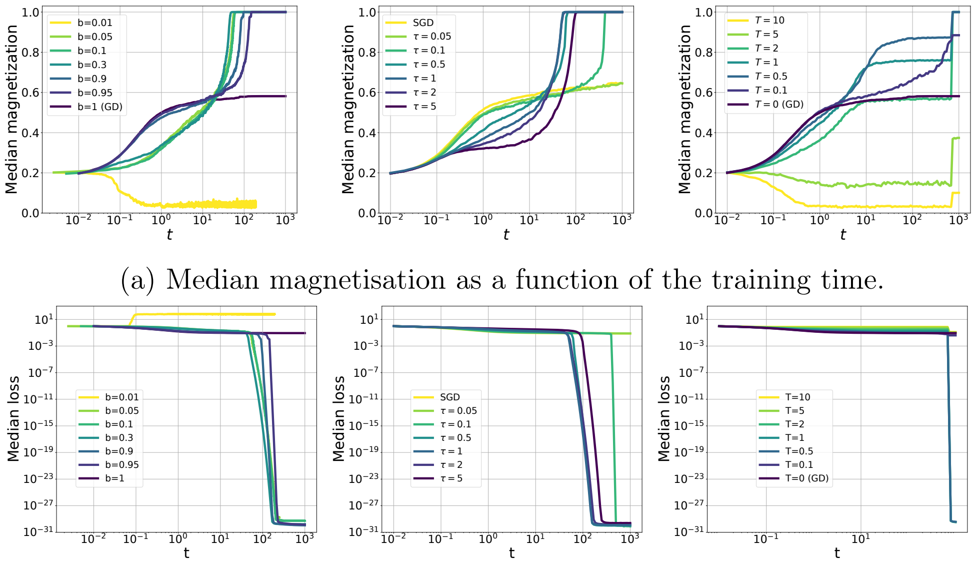
(a) Average magnetisation (left) and average training loss (right) as a function of training time, at fixed $\alpha = n/d = 3$, initial magnetisation $m_0 = 0.2$, input dimension $d = 1000$, learning rate $\mathrm{d}t = 0.01$. We show the performance of full-batch GD (red line), multi-pass vanilla SGD at $b = 0.5$ (dotted green line), and p-SGD at $\tau = 1, b = 0.5$ (dashed blue line). The averages are computed over 500 seeds (generating a new instance for each seed). At time $t = 1000$, the percentages of seeds that have reached training loss below $10^{-7}$ are: 9% (GD), 30% (SGD), 99% (p-SGD). For visibility purposes, we plot $t + \mathrm{d}t$ on the $x-$axes.

(b) We show 50 instances of the magnetisation as a function of training time for the algorithmic settings considered in the above panel. For each seed, a new instance is generated. For visibility purposes, we plot $t + \mathrm{d}t$ on the $x-$axes.

Figure 2.4.2 – Multi-pass SGD has a built-in self-annealing protocol allowing to outperform GD. The disappearance of the plateaus is a feature of finite persistence time.

this regime of parameters, the full trajectory of the algorithm is crucial to achieve perfect recovery.

In the upper panels (Figure 2.4.1a), we compare the average magnetisation from numerical simulations at finite system size and finite learning rate (grey dots) to the theoretical prediction (red line) obtained by integrating the DMFT equations derived in the high-dimensional continuous limit. In this case, we generate a new dataset for each simulations in order to remove sample-to-sample fluctuations. We find a very good agreement between asymptotic theory and the average from simulations already for the used system sizes and learning rates, indicating that the observed behavior is not a feature of finite size or finite learning rate effects.

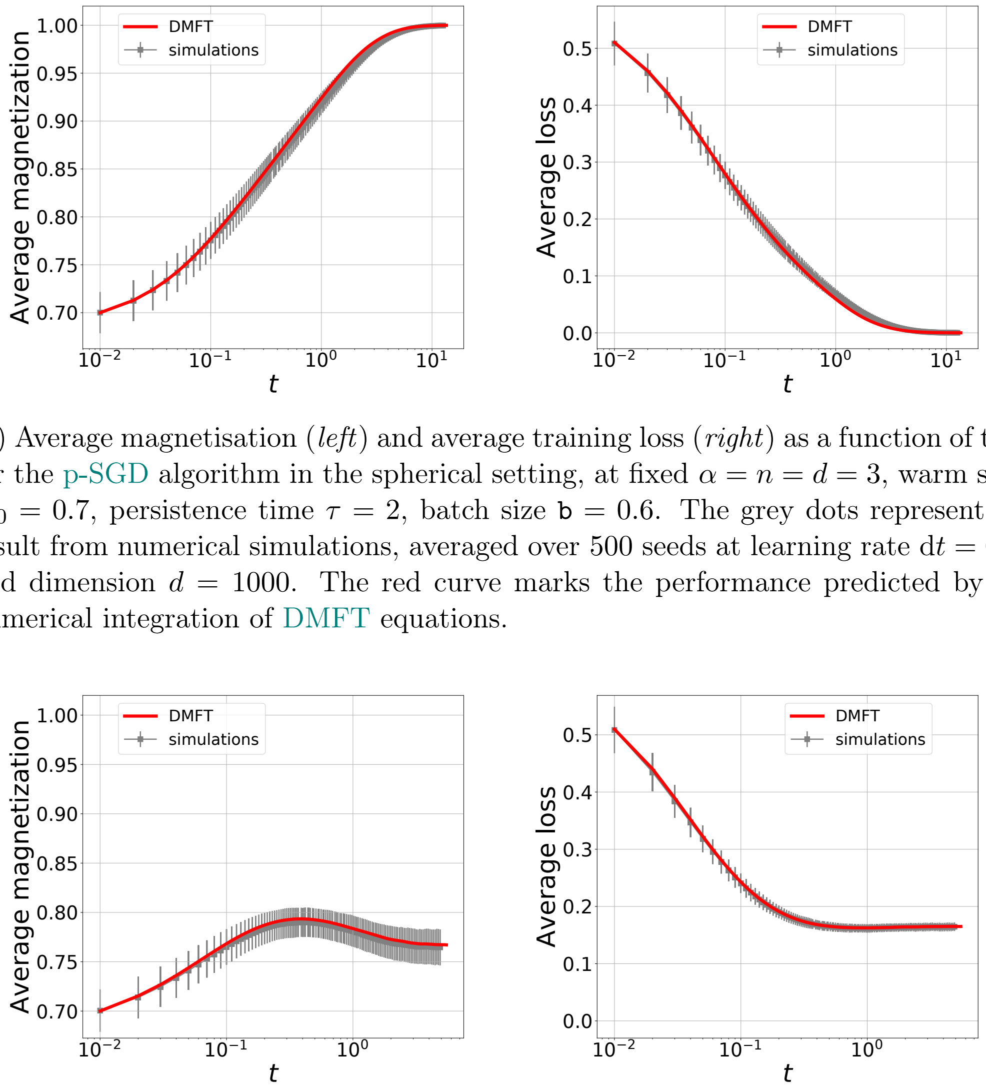


(a) Median magnetisation as a function of the training time.

(b) Median loss as a function of the training time.

Figure 2.4.3 – We fix the parameters $\alpha = n/d = 3$, initialisation $m_0 = 0.2$, dimension $d = 1000$, learning rate $\mathrm{d}t = 0.01$. The median is computed over 250 seeds, drawing a new dataset, signal and initialisation for each seed. For visibility purposes, we plot $t + \mathrm{d}t$ on the $x-$axes. *Left.* We show p-SGD for increasing values of batch size $\mathtt{b} = 0.01, 0.05, 0.1, 0.3, 0.9, 0.95, 1$ and fixed persistence time $\tau = 1$. In the case $\mathtt{b} = 0.1$, the learning rate has been reduced to $\mathrm{d}t = 0.005$, for $\mathtt{b} = 0.01, 0.05$ we have used $\mathrm{d}t = 0.0025$.

**Multi-pass SGD outperforms GD —** Figure 2.4.2 shows the average magnetisation and the average training loss as a function of time for full-batch GD, multi-pass SGD and its persistent version p-SGD. In the case of multi-pass SGD, we sample (with replacement) mini batches of size $bn$ at each time step. In Figure 2.4.2b, we depict different instances of the dynamics, corresponding to different realisations of the dataset and the noise vector $\boldsymbol{z}$ (Eq. (2.4.5)). We find that SGD and p-SGD with $\tau = 1$ outperform GD in recovering the hidden signal. Indeed, at time scales at which p-SGD has already reached magnetisation one and zero loss, gradient descent is stuck in regions of poorer generalisation. The average magnetisation of SGD lies between the two. Therefore, a finite batch size is beneficial for the performance. Furthermore, the behavior of the curves for different seeds unveils an important role played by the persistence time. Indeed, while the evolution of the magnetisation for GD is characterised by long plateaus alternated by sudden jumps, p-SGD is not stuck in the same region for long times. Again, the behavior of SGD is intermediate between the two: we see from Figure 2.4.2b that the disappearance of the plateaus is a feature of a finite persistence time. These findings suggest that the interplay of the finite batch size and the persistence time is crucial to achieve the optimal performance.



(a) Average magnetisation (*left*) and average training loss (*right*) as a function of time for the p-SGD algorithm in the spherical setting, at fixed $\alpha = n = d = 3$, warm start $m_0 = 0.7$, persistence time $\tau = 2$, batch size $\mathtt{b} = 0.6$. The grey dots represent the result from numerical simulations, averaged over 500 seeds at learning rate $\mathrm{d}t = 0.01$ and dimension $d = 1000$. The red curve marks the performance predicted by the numerical integration of DMFT equations.

(b) Average magnetisation (left) and average training loss (right) as a function of time for the Langevin algorithm in the spherical setting, at fixed $\alpha = M/N = 3$, warm start $m_0 = 0.7$, temperature $T = 1$. The grey dots represent the result from numerical simulations, averaged over 1000 seeds at learning rate $\mathrm{d}t = 0.01$ and dimension $N = 1000$. The red curve marks the performance predicted by the numerical integration of DMFT equations.

Figure 2.4.4 – DMFT VS simulations for stochastic training algorithms.

**The role of the noise —** Figure 2.4.3 illustrates the effect of different sources of stochasticity on the generalisation performance. In particular, we compare the role played by the white noise at temperature $T$ in the Langevin algorithm to the double source of noise in the SGD algorithm: the finite batch size $\mathtt{b}$ and the persistence time $\tau$. In the left panel, we depict the dependence of the SGD algorithm on the batch size, at fixed persistence time. We find that the generalisation performance is non-monotonic in the batch size and the optimal value is attained at intermediate

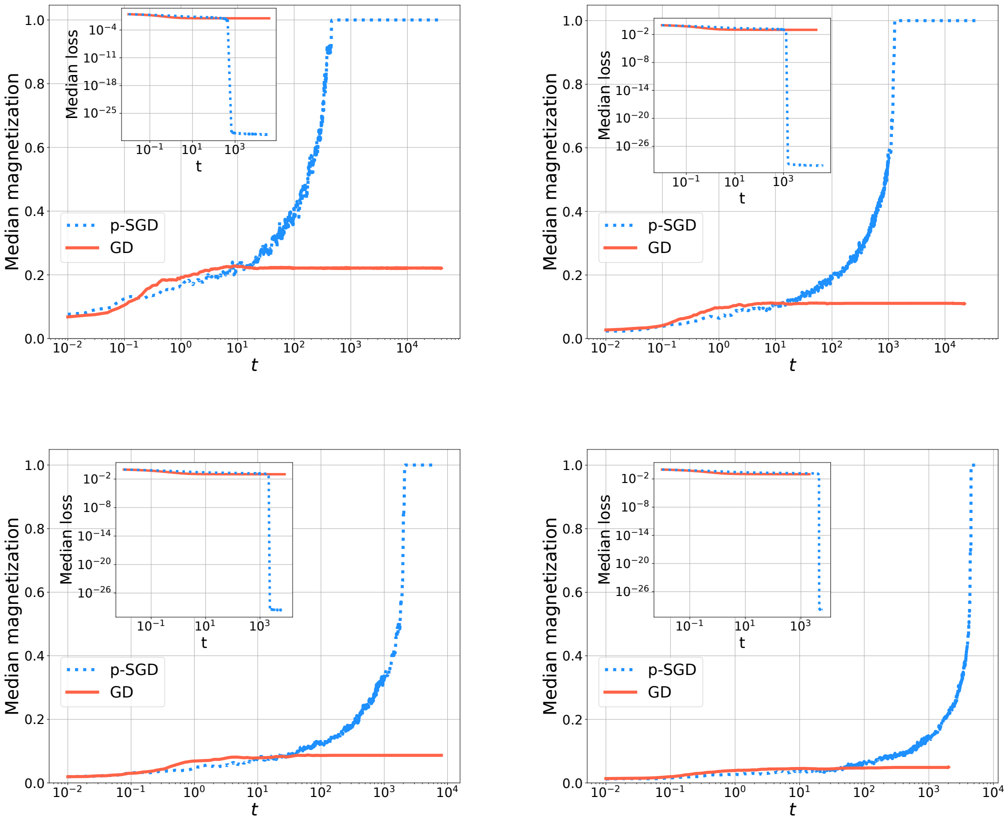


Figure 2.4.5 – Median magnetisation (main plot) and median loss (inset) as a function of time from numerical simulation for the spherical setting at fixed $\alpha = n/d = 2.5$, learning rate $dt = 0.01$, 100 seeds, and increasing dimension $d = 100, 500, 1000, 2500$ from upper left to lower right. We consider random initialisation $m_0 = 0$, so the finite initial overlap with the signal is only due to finite size effects. The full red line marks the performance of full-batch gradient descent, while the dotted blue line represents the persistent-SGD algorithm at batch size $\mathsf{b} = 0.5$ and persistence time $\tau = 2$.

$\mathsf{b}$. Therefore, at variance with what observed in deep neural networks trained on real datasets (Jastrzebski et al., 2017; Keskar et al., 2017), in our case we obtain that the optimal batch size is an extensive fraction of the total number of samples.

The central panel displays the (median) performance of SGD for different values of the persistence time $\tau$, at fixed batch size. For times $t \leq \tau$, the samples used to compute the gradient (on average) do not change, and thus the dynamics presents plateaus. However, as soon as $t > \tau$, the mini batch is refreshed. This results in a sudden increase in performance at times $t \sim \tau$, that becomes more visible the larger $\tau$. Moreover, we observe a non-monotonic behavior of the performance as a function of $\tau$. On the one hand, increasing $\tau$ shifts the final plateau at larger times, delaying the recovery of the signal. On the other hand, if the persistence time is too small, the dynamics gets trapped close to the signal, displaying plateaus followed by sudden jumps similarly as for GD (see Figure 2.4.2b). There is therefore

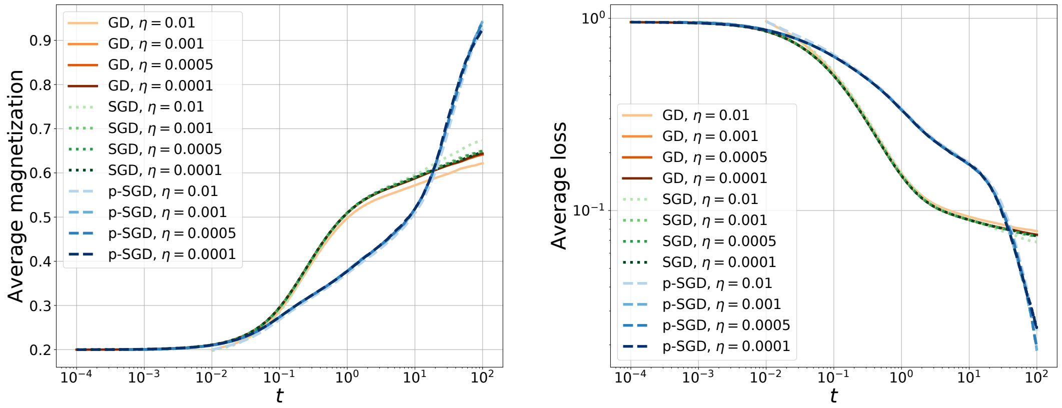


Figure 2.4.6 – Average magnetisation (right) and average loss (less) as a function of training time for the three algorithms: GD (full red lines), SGD (dotted green lines) and p-SGD (dashed blue lines). The numerical simulations are run at fixed $\alpha = n/d = 3$, warm start $m_0 = 0.2$ and input dimension $N = 1000$, over 250 seeds. The stochastic algorithms are run at fixed batch size $\mathtt{b} = 0.5$. We consider decreasing values of learning rate $\mathrm{d}t = 0.01, 0.001, 0.0005, 0.0001$, depicted with increasing color intensity. For visibility purposes, we plot $t + \mathrm{d}t$ on the x-axes.

an intermediate range of persistence times $\tau$ for which the performance is the best (better than vanilla SGD).

Since the literature often compares the SGD noise to the Langevin noise (Cheng et al., 2020; Li et al., 2017; Jastrzebski et al., 2017; Hu et al., 2019; Zhu et al., 2018) we compare here to the performance achieved by the Langevin algorithm at fixed temperature. The right panel of Figure 2.4.3 depicts the performance of the Langevin algorithm for different values of temperature $T$. At large times ($t = 700$ in the figure) the temperature is switched to zero. We find that the best performance is again reached for intermediate values of the temperature $T$.

We underline the qualitative difference between the effective noise introduced by multi-pass SGD and the white noise of Langevin algorithm. The variance of the noise in Langevin is fixed by the temperature, therefore – in order to reach a minimum – an annealing protocol must be implemented and optimised. In contrast, the noise introduced by SGD is automatically reduced during training and it is zero at the global minimum. Therefore, multi-pass SGD has a built-in self annealing protocol, that can be optimised by tuning only two parameters ($\mathtt{b}$ and $\tau$) instead of the whole trajectory of the temperature over time.

**More on the analytic characterisation** — Figure 2.4.4a shows the comparison between the average performance of p-SGD obtained from numerical simulations (grey symbols) with the prediction derived by integrating the DMFT equations (red line). The left panel depicts the average magnetisation, while the right panel displays the average training loss as a function of time. Figure 2.4.4b displays the same comparison for the Langevin algorithm. In both cases, we find a very good agreement between theory and simulations.

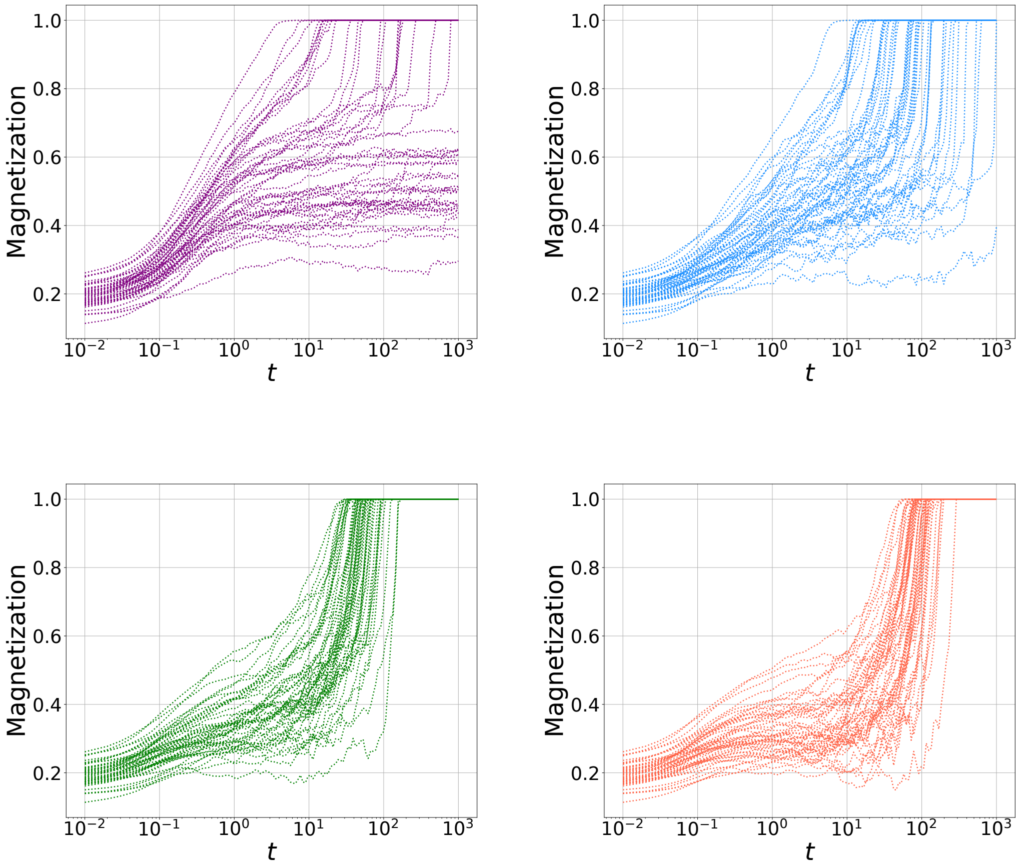


Figure 2.4.7 – Instances of the magnetisation as a function of time from numerical simulations for the persistent SGD algorithm at fixed $\alpha = n/d = 3$, batch size $\mathtt{b} = 0.5$ and warm initialisation $m_0 = 0.2$. We consider four different values of the persistence time: $\tau = 0.05$ (upper left), $\tau = 0.5$ (upper right), $\tau = 2$ (lower left), $\tau = 5$ (lower right). For each panel, we show 50 different seeds, corresponding to different realisations of the landscape and initial weights. The simulations are run at dimension $d = 1000$ and learning rate $\mathrm{d}t = 0.01$.

**Random initialisation —** Figure 2.4.5 investigates the behavior of full-batch GD (full red lines) and p-SGD (dashed blue lines) starting from random initialisation at fixed $\alpha = 2.5$. p-SGD is run at fixed $\mathtt{b} = 0.5$, $\tau = 2$. We show the median magnetisation (main plots) and the median loss (insets) as a function of time for increasing values of the dimension: $d = 100$ (above-left panel), $d = 500$ (above-right panel), $d = 1000$ (below-left panel). and $d = 2500$ (below-right panel). In this case $m_0 = 0$ and the warm start in the four panels is only given by finite size effects. We clearly see that, at time scales shown here, gradient descent is stuck at a plateau of height decreasing as the dimension $d$ increases.

As studied in Mannelli et al. (2020b), the recovery transition of gradient descent starting from random initialisation for comparable system sizes happens at $\alpha \approx 6$, which is few times larger than the value $\alpha = 2.5$ considered here. However, we observe that already at $\alpha = 2.5$ the persistent-SGD algorithm can reach perfect recovery for the system sizes under consideration. The time to reach the solution from random initialisation is, as expected, compatible with logarithmic increase in the system size. These observations suggest that the recovery transition for stochastic gradient descent starting from random initialisation is shifted to lower values of $\alpha$ when compared to gradient descent. This is an interesting direction for future investigations.

Figure 2.4.6 compares the average magnetisation (left panel) and loss (right panel) as a function of training time for gradient-descent, SGD and p-SGD for decreasing values of the learning rate. We observe that, in the limit of small learning rate, the learning curves of SGD collapse to the ones of gradient descent. On the contrary, the p-SGD algorithm has a well-defined continuous time limit that is different than the one of full batch gradient descent.

Figure 2.4.7 summarises the effect of increasing the persistence time on the performance of the p-SGD algorihm. We show the instances of the magnetisation as a function of time – corresponding to 50 different realisations of the problem landscape and initialisations of the weight vector. We consider increasing values of the parameter $\tau = 0.05$ (upper left panel), $\tau = 0.5$ (upper right panel), $\tau = 2$ (lower left panel), and $\tau = 5$ (lower right panel), at a fixed ratio $\alpha = 3$ of training samples over input dimensions, batch size $\mathtt{b} = 0.5$ and warm initialisation $m_0 = 0.2$. On the one hand, we observe that increasing the persistence time gradually diminishes the number of seeds that get stuck at intermediate plateau, resulting in an improved generalisation performance. On the other hand, until time $t \sim \tau$ the samples in the mini batch have not been reshuffled yet (on average). Therefore, for large values of $\tau$ the plateaus disappear but the magnetisation is stuck at the beginning of the training and only at training time $t > \tau$ it has a sudden increase.

# Article 3

## Dynamical mean-field theory for stochastic gradient descent in Gaussian mixture classification

Francesca Mignacco, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborová.

Advances in Neural Information Processing Systems, 2020, vol. 33.
Published in the “Machine Learning 2021” Special Issue, J. Stat. Mech. (2021) 124008

☞ Advances in Neural Information Processing Systems 33, p. 9540–9550
☞ Journal of Statistical Mechanics: Theory and Experiment, Volume 2021, December 2021
☞ ArXiv preprint: *arXiv:2006.06098*

### Abstract

We analyze in a closed form the learning dynamics of stochastic gradient descent (SGD) for a single layer neural network classifying a high-dimensional Gaussian mixture where each cluster is assigned one of two labels. This problem provides a prototype of a non-convex loss landscape with interpolating regimes and a large generalization gap. We define a particular stochastic process for which SGD can be extended to a continuous-time limit that we call stochastic gradient flow. In the full-batch limit we recover the standard gradient flow. We apply dynamical mean-field theory from statistical physics to track the dynamics of the algorithm in the high-dimensional limit via a self-consistent stochastic process. We explore the performance of the algorithm as a function of control parameters shedding light on how it navigates the loss landscape.

# Article 4

## The effective noise of Stochastic Gradient Descent

Francesca Mignacco, and Pierfrancesco Urbani.

Journal of Statistical Mechanics: Theory and Experiment, Volume 2022, August 2022.

☞ *J. Stat. Mech.* (2022) 083405
☞ ArXiv preprint: *arXiv:2112.10852*

**Abstract**

Stochastic Gradient Descent (SGD) is the workhorse algorithm of deep learning technology. At each step of the training phase, a mini batch of samples is drawn from the training dataset and the weights of the neural network are adjusted according to the performance on this specific subset of examples. The mini-batch sampling procedure introduces a stochastic dynamics to the gradient descent, with a non-trivial state-dependent noise. We characterize the stochasticity of SGD and a recently-introduced variant, persistent SGD, in a prototypical neural network model. In the under-parametrized regime, where the final training error is positive, the SGD dynamics reaches a stationary state and we define an effective temperature from the fluctuation-dissipation theorem, computed from dynamical mean-field theory. We use the effective temperature to quantify the magnitude of the SGD noise as a function of the problem parameters. In the over-parametrized regime, where the training error vanishes, we measure the noise magnitude of SGD by computing the average distance between two replicas of the system with the same initialization and two different realizations of SGD noise. We find that the two noise measures behave similarly as a function of the problem parameters. Moreover, we observe that noisier algorithms lead to wider decision boundaries of the corresponding constraint satisfaction problem.

# Article 5

## Stochasticity helps to navigate rough landscapes: comparing gradient-descent-based algorithms in the phase retrieval problem

Francesca Mignacco, Pierfrancesco Urbani, Lenka Zdeborová.

Machine Learning: Science and Technology, 2021, 2(3), p.035029.

☞ Machine Learning: Science and Technology, Volume 2, Number 3
☞ ArXiv preprint: *arXiv:2103.04902*

### Abstract

Stochastic Gradient Descent (SGD) is the workhorse algorithm of deep learning technology. At each step of the training phase, a mini batch of samples is drawn from the training dataset and the weights of the neural network are adjusted according to the performance on this specific subset of examples. The mini-batch sampling procedure introduces a stochastic dynamics to the gradient descent, with a non-trivial state-dependent noise. We characterize the stochasticity of SGD and a recently-introduced variant, persistent SGD, in a prototypical neural network model. In the under-parametrized regime, where the final training error is positive, the SGD dynamics reaches a stationary state and we define an effective temperature from the fluctuation-dissipation theorem, computed from dynamical mean-field theory. We use the effective temperature to quantify the magnitude of the SGD noise as a function of the problem parameters. In the over-parametrized regime, where the training error vanishes, we measure the noise magnitude of SGD by computing the average distance between two replicas of the system with the same initialization and two different realizations of SGD noise. We find that the two noise measures behave similarly as a function of the problem parameters. Moreover, we observe that noisier algorithms lead to wider decision boundaries of the corresponding constraint satisfaction problem.

# 3 - Conclusions and perspectives

## 3.1 - Final remarks on this thesis

In this conclusive chapter, we summarise the contributions of this thesis and their relation with subsequent research developments, and we discuss some possible extensions.

**Binary Gaussian Mixture Model —** The binary Gaussian mixture model (GMM) introduced in Chapter 1.2 has served us as a prototype classification task to discuss in a unified fashion different phenomena of interest in ML theory.

In Article 1, we have studied the performance of regularised convex classifiers at separating the mixture of two Gaussian clusters in the noisy regime where even an oracle knowing the centers of the clusters would make a finite fraction of mistakes. We have derived rigorous closed-form formulas for the generalisation and training errors in the limit where the number of samples and dimensions go to infinity, while their ratio is a fixed controlled parameter. We have then applied our theoretical findings to shed light on the role of the different model parameters on the generalisation performance. We have considered the setup with a generic bias $\kappa$, two clusters with generic sizes tuned by $\rho \in (0,1)$, showing that the case $\kappa = 0$, $\rho = 0.5$ is singular, and the generic case, $\rho \neq 0.5$, has qualitatively different behaviour when regularisation is added. Finally, we have obtained that the linear separability transition explicitly depends on the cluster size and the noise variance.

Given the full understanding of the static properties of the problem, in Article 3 we have turned our focus to the characterisation of the multi-pass stochastic gradient descent (SGD) dynamics, where the training dataset is fixed and the samples are reused multiple times. In particular, we have analysed an SGD algorithm in which, at each iteration, the mini batch of samples used to approximate the gradient of the loss is drawn at random, and we have defined a *persistent* variant of this stochastic process for which multi-pass SGD can be extended to a continuous-time limit that we call stochastic gradient flow. We have managed to describe the high-dimensional limit of the randomly initialised SGD using dynamical mean-field theory (DMFT) from disordered systems, that leads to a description of the dynamics in terms of a self-consistent stochastic process. We have integrated numerically the self-consistent DMFT equations, that notably hold also for non-convex variants of the problem, and we have found excellent agreement with experiments at finite dimension and finite time step.

In Article 4, we have analysed the nature of the stochastic noise in SGD-type algorithms in the setting of binary classification of Gaussian mixtures. We have shown that this noise can be described by an effective temperature defined through the fluctuation-dissipation theorem. In the underparametrised regime, where the loss landscape displays a unique minimum, both vanilla-SGD and p-SGD converge to a steady state which is driven by the algorithmic noise. We have shown that the stationary state of vanilla-SGD is characterised by an effective temperature that tends to zero for vanishing learning rate. For p-SGD, we have shown that the effective temperature increases with the persistence time, while it is non-monotonic with the batch size. In the over-parametrised regime, we have presented an alternative characterisation of the magnitude of algorithmic noise. In particular, we have found that the noisier the algorithm, the smaller the fraction of support vectors at the end of the dynamics. These results have been derived for a simple yet paradigmatic supervised learning task. In the UNSAT phase, this setting provides the advantage that the noise captured by our analysis comes entirely from the algorithm itself since there is no other source of randomness at fixed dataset and initialisation.

**Teacher-student multi-class classification** — In Article 2, we have extended the characterisation of the learning curves of the teacher-student perceptron for supervised classification to the case where the labels come from more than two classes. We have derived rigorous asymptotic expressions for the performance of the Bayes-optimal (BO) estimator as well as for regularised empricial risk minimisation (ERM). This model provides a theoretical playground where important practical questions can be quantitatively explored, e.g., the role of regularisation and the optimal tuning of hyperparameters. Indeed, we have observed that the cross-entropy with optimally-tuned ridge regularisation can achieve close-to-optimal performance in the case of Gaussian teacher prior. For binary teacher prior, we have found instead that a first order transition arises in the BO error. It would be interesting to investigate how these observations modify in the limit of very large number of classes and to incorporate a more realistic data structure in the model.

**The sign-retrieval problem** — In Article 5, we have considered the real-valued phase retrieval problem as a paradigmatic highly non-convex optimisation problem to test the generalisation performance of full-batch GD and some of its stochastic variants: multi-pass SGD, its persistent version p-SGD, and the Langevin algorithm. We have shown that stochasticity is crucial to achieve perfect recovery of the hidden signal at low sample complexity so that SGD outperforms GD in this task. We have observed intriguing features of the loss profile and illustrated how various sources of noise allow the dynamics to circumvent the traps in the landscape. We have provided an analytic description of the learning curves in the infinite-dimensional limit via DMFT, showing that the observed behaviour is not due to finite size effects or to a finite learning rate.

Article 5 leads to interesting extensions both on the analytic and numerical sides. On the one hand, the characterisation of the dynamical evolution of the algorithms via DMFT can be extended to include smart initialisations (e.g., spectral initialisation) by means of the replica trick (Houghton et al., 1983), and regularisation strategies (e.g., trimming) that are commonly applied in practical applications in this context. On the other hand, it would be interesting to test the persistent variant of multi-pass SGD and investigate the role of the persistence time on real datasets and architectures, which we leave for future work.

**Related works on GMMs** — The binary GMM studied here has been further employed in subsequent works to understand interesting phenomena from real applications, for instance regularisation inheritance via knowledge distillation (Saglietti &

Zdeborová, 2022) and the bias-inheritance mechanism (Mannelli et al., 2022).

GMMs are well-studied in statistical learning theory and their supervised version has recently triggered a surge of interest in ML theory. Indeed, Seddik et al. (2020) have shown that realistic data generated by a GAN behave as Gaussian mixtures, while Papyan et al. (2020) have observed that the cross-entropy loss learns features that *collapse* into a mixture of clusters that can be separated by the readout layer.

In Loureiro et al. (2021), the authors have extended our derivation of closed-form error formulas to the case of multiple clusters with generic covariances and means. Refinetti et al. (2021b) have studied the online dynamics of Gaussian mixture classification to investigate the gap between the feature and lazy learning regimes.

**Related works on the learning dynamics** — The characterisation of the dynamical trajectory of learning algorithms has attracted increasing attention in the last few years. An interesting development concerns the rigorous proof of the DMFT equations where the integro-differential system involves an effective self-consistent stochastic process. This has been achieved in the recent work by Celentano et al. (2021) in the case of full-batch GD on i.i.d. Gaussian inputs and one-hidden-layer networks in the limit of infinite dimensional data and samples at finite hidden-layer size. The proof is based on a mapping between the GD updates and the iterates of an AMP algorithm.

Bodin & Macris (2021a,b) have leveraged on recent advances in random matrix theory to derive explicit formulas for the gradient-flow dynamics in rank-one matrix estimation and random feature regression in the high-dimensional asymptotic limit.

In Bordelon & Pehlevan (2021), the authors have considered linear regression on random features with arbitrary covariance structure. They have derived exact formulas for the dynamics of SGD both in the online and batch cases at fixed inputs. Bordelon & Pehlevan (2022) have also analysed feature learning in infinite-width shallow neural networks at fixed inputs and have derived self-consistent equations for the learning curves via a path integral formulation of gradient flow dynamics.

Çakmak et al. (2022) have applied DMFT to characterise the dynamics of a sequential message-passing algorithm for approximate inference in a teacher-student Gaussian-latent-variable model. Remarkably, at variance with our DMFT equations for SGD derived in Chapter 2.2, in Çakmak et al. (2022) there is no memory term in the effective stochastic process. This allows to obtain a simple recursion formula for the correlation functions.

In this thesis we have only considered training algorithms with constant learning rate. The optimal tuning of the learning rate schedule for the Langevin algorithm has been studied by d'Ascoli et al. (2022) in two planted $p-$spin models ($p = 2$ and the spiked matrix-tensor model introduced in Chapter 2.1).

# 3.2 - Future directions

In this chapter, we extend the discussion on some possible future directions that could be inspired by the results of this thesis.

Unveiling the connection between the dynamical evolution of a high-dimensional system and the underlying energy landscape is a fundamental question in the physics of glasses, that has been fully understood only in some special cases, as discussed in Chapter 2.1. This open puzzle actually unifies a wide variety of interdisciplinary applications in information theory and computer science. The implicit bias of SGD in supervised learning problems belongs to this category of problems, and is therefore well-suited to be addressed with the tools developed to study the physics of glassy systems.

**Generalisation and the geometry of the solution space —** A crucial step forward elucidating the missing link between the statics and the dynamics properties of learning would be a full understanding of the connection between the geometry of the solution space and the generalisation abilities of the solutions. This perspective inscribes in the literature investigating the landscape "flatness" properties, that we have previously discussed in Chapter 2.1.

Our analysis of the SAT phase in Article 4 suggests that the magnitude of the SGD noise could be connected to the width of the decision boundary of a given solution of the classification. Interestingly, Baldassi et al. (2021) consider ANNs with binary weights and show that high-margin – i.e., robust – minima tend to concentrate in particular regions that are also dense of lower-margin solutions.It would be relevant to investigate the geometrical properties of the boundary of the "lake" of solutions in supervised learning problems in the SAT phase with *continuous* degrees of freedom and link them to the generalisation abilities. Indeed, for all the losses with a cutoff and, in practice, whenever the dynamics is stopped as soon as the training error reaches zero, both GD and SGD (without momentum) stop at the boundary of the solution space. Therefore, in practice, the zero-error solutions accessible to gradient-based algorithms lie on a boundary, whose width determines the solution robustness, as pictorially shown in Figure 3.2.1.

**Measuring the effective temperature of SGD in real applications —** The definition of effective temperature $T_{\text{eff}}$ that we have introduced for the binary GMM in Article 4 as a measure of the magnitude of SGD noise is fully general and independent of the specifics of the problem. The effective temperature can be computed analytically for all the synthetic tasks that can be treated via DMFT, including non-convex losses where an increase in $T_{\text{eff}}$ can be also triggered by the roughness of the underlying landscape, as already discussed in Chapter 2.3.

An interesting future direction of investigation would be to test the violation of the FDT by measuring correlation and integrated response functions for the dynamics of SGD in the overparametrised settings of practical ML applications via

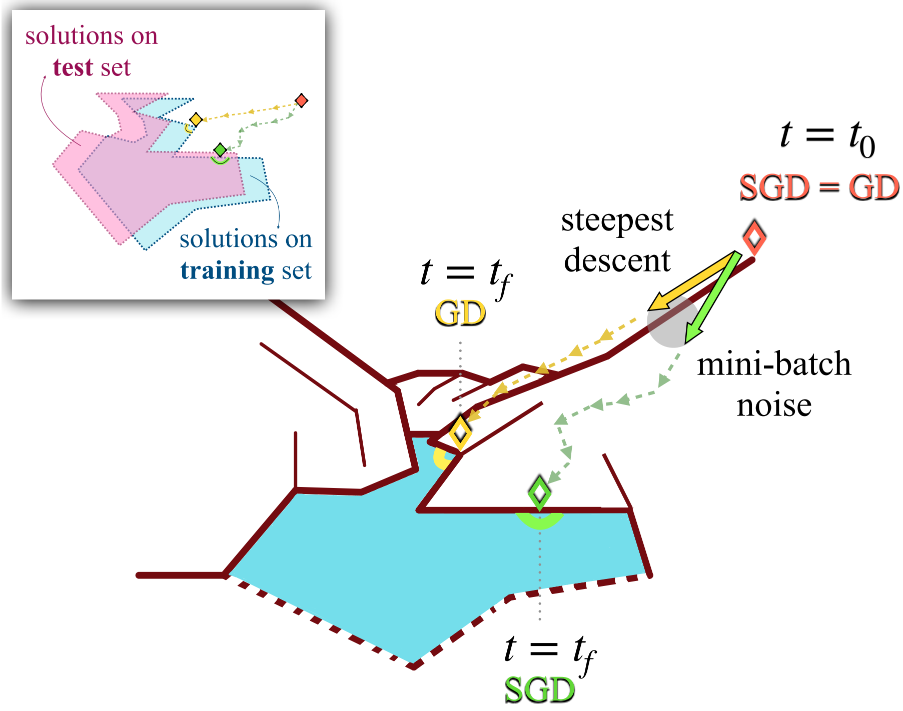


Figure 3.2.1 – Schematic representation of the different borders of the solution space that can be reached according to the optimisation procedure. We sketch two realisations of the GD and SGD dynamics starting from the same initial condition and navigating a non-convex loss landscape with a wide basin, or "lake", of solutions. The algorithmic noise can lead the dynamics of SGD to a different endpoint with respect to the solution found by GD. The geometrical properties of the border of the solution space, or the classification boundary of a solution in dual representation, are crucial for the performance and deserve further investigation.

the numerical procedure that is normally used for glassy systems (see, e.g., Ricci-Tersenghi (2003) and references therein) in a regime where the dynamics reaches a stationary state. A number of questions could be investigated in this framework, for instance the impact of SGD noise on the slowing-down in the algorithmic convergence and the crossover between the feature learning and the lazy learning regimes.

**The theory of biologically-plausible training algorithms** — Biologically-inspired alternatives to the backpropagation algorithm used to train multi-layer networks have recently attracted a surge of interest, first from a practical implementation perspective and then from a theoretical point of view. Well-known examples are the feedback alignment algorithm (Lillicrap et al., 2016) and direct feedback alignment (DFA) (Nøkland, 2016). Launay et al. (2020) have shown that DFA-based algorithms can be successfully trained on state-of-the-art models, but still perform poorly on convolutional layers. Clark et al. (2021) have overcome this issue by proposing an alternative learning rule that operates on a specific class of ANNs, while the backward pass has been completely removed and approximated via two forward passes in Dellaferrera & Kreiman (2022).

These works pave the way for the promising research direction of closing the gap between artificial and biological learning. However, theoretical understanding of the success and limitations of these implementations is still sparse. In Refinetti et al. (2021a), the authors have analysed the online dynamics of DFA unveiling the presence of two learning phases: the alignment of the approximate gradient with the true one, followed by a data-fitting phase, biasing the dynamics towards the solution which maximises gradient alignment. It would be interesting to explore the interplay of these alternative learning schemes and the multi-pass SGD algorithm, and to investigate the role of persistence in this framework.

**More on the algorithm and the architecture** — Another possible research direction would be to extend the DMFT equations to incorporate more realistic architectures and data structures, as well as different learning protocols. For instance, Sarao Mannelli & Urbani (2021) have studied the dynamics of GD with momentum in the spiked matrix-tensor inference model introduced in Chapter 2.1 and one could extend their analysis to the case of multi-pass SGD with momentum in supervised learning models.

Since DMFT equations allow to incorporate time-dependence in the labels and data, different training protocols with SGD can be further investigated. Some examples are adversarial initialisation when training with random labels (Liu et al., 2019) and the mini-batch dynamics of curriculum learning (see, e.g., Saglietti et al. (2021) for the online case). A number of crucial phenomena take place during the early phase of training, as summarised by the set of experiments carried out in (Frankle et al., 2019). This is precisely the time window that is accessible by the DMFT formalism. It would be thus interesting to explore whether DMFT can shed light on some of this early-training observations, e.g., emergence of learning sub-phases, (lack of) robustness to perturbations, fast correlation of the weigths.

Regarding the architecture, a crucial open theoretical question is how to model depth in an efficient way, beyond the one-hidden-layer cases already discussed in

Chapter 2.1. Indeed, when multiple layers are added, the correlations introduced by the training procedure drastically complicate the analytic methods presented in this thesis and theoretical results have been obtained for deep networks only in very specific cases (see, e.g., Saxe et al. (2013); Li & Sompolinsky (2021)). Finding appropriate regimes where depth can be efficiently analysed is a fundamental direction that must be addressed.

Finally, in this thesis we have only considered fully-connected networks, however it would be relevant to extend the present analysis to convolutional architectures, where the hidden neurons are connected to only a subset of variables in the previous layer (a so-called *receptive field*). A first step in this direction would be to consider an intermediate regime between a fully connected and a tree-like committee machine (Franz et al., 2019b), with partially overlapping receptive fields.

**Beyond supervised learning —** A final extension of the results of this thesis could be in the direction of devising simple models beyond supervised learning, for instance exploring self-supervised learning protocols (see, e.g., Grill et al. (2020); Chen & He (2021) and references therein). This is a much less explored domain where the tools from disordered systems physics could shed light on new interesting phenomenology.

# Bibliography

Abbe, E. and Sandon, C. Poly-time universality and limitations of deep learning, 2020.

Advani, M. and Ganguli, S. An equivalence between high dimensional bayes optimal inference and m-estimation. *Advances in Neural Information Processing Systems*, 29, 2016.

Advani, M. S., Saxe, A. M., and Sompolinsky, H. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

Agoritsas, E., Biroli, G., Urbani, P., and Zamponi, F. Out-of-equilibrium dynamical mean-field equations for the perceptron model. *Journal of Physics A: Mathematical and Theoretical*, 51(8):085002, 2018.

Amit, D. J., Gutfreund, H., and Sompolinsky, H. Spin-glass models of neural networks. *Phys. Rev. A*, 32:1007–1018, Aug 1985a. doi: 10.1103/PhysRevA.32.1007. URL https://link.aps.org/doi/10.1103/PhysRevA.32.1007.

Amit, D. J., Gutfreund, H., and Sompolinsky, H. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.*, 55:1530–1533, Sep 1985b. doi: 10.1103/PhysRevLett.55.1530. URL https://link.aps.org/doi/10.1103/PhysRevLett.55.1530.

Anderson, P. W. Spin glass iii: Theory raises its head. *Physics Today*, 41:9–11, 1988a.

Anderson, P. W. Spin glass i: A scaling law rescued. *Physics Today*, 41(1):9–11, 1988b.

Anderson, P. W. Spin glass vii: Spin glass as a paradigm. *Physics Today*, 43(3): 9–11, 1990.

Arous, G. B., Guionnet, A., et al. Symmetric langevin spin glass dynamics. *The Annals of Probability*, 25(3):1367–1422, 1997.

Aubin, B. *Mean-field methods and algorithmic perspectives for high-dimensional machine learning*. PhD thesis, Université Paris-Saclay, 2020. Thèse de doctorat dirigée par Zdeborová, Lenka Physique université Paris-Saclay 2020.

Aubin, B., Maillard, A., Barbier, J., Krzakala, F., Macris, N., and Zdeborová, L. The committee machine: Computational to statistical gaps in learning a two-layers neural network. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124023, 2019.

Aubin, B., Krzakala, F., Lu, Y., and Zdeborová, L. Generalization error in high-dimensional perceptrons: Approaching bayes error with convex optimization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12199–12210. Curran Associates, Inc., 2020.

Bahri, Y., Kadmon, J., Pennington, J., Schoenholz, S. S., Sohl-Dickstein, J., and Ganguli, S. Statistical mechanics of deep learning. *Annual Review of Condensed Matter Physics*, 11:501–528, 2020.

Baity-Jesi, M., Sagun, L., Geiger, M., Spigler, S., Arous, G. B., Cammarota, C., LeCun, Y., Wyart, M., and Biroli, G. Comparing dynamics: Deep neural networks versus glassy systems. In *International Conference on Machine Learning*, pp. 314–323. PMLR, 2018.

Balan, R., Casazza, P., and Edidin, D. On signal reconstruction without phase. *Applied and Computational Harmonic Analysis*, 20:345–356, 05 2006.

Baldassi, C., Borgs, C., Chayes, J. T., Ingrosso, A., Lucibello, C., Saglietti, L., and Zecchina, R. 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. doi: 10.1073/pnas.1608103113. URL https://www.pnas.org/doi/abs/10.1073/pnas.1608103113.

Baldassi, C., Malatesta, E. M., and Zecchina, R. 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. URL https://link.aps.org/doi/10.1103/PhysRevLett.123.170602.

Baldassi, C., Pittorino, F., and Zecchina, R. Shaping the learning landscape in neural networks around wide flat minima. *Proceedings of the National Academy of Sciences*, 117(1):161–170, 2020.

Baldassi, C., Lauditi, C., Malatesta, E. M., Perugini, G., and Zecchina, R. Unveiling the structure of wide flat minima in neural networks. *Physical Review Letters*, 127 (27):278301, 2021.

Baldi, P. and Chauvin, Y. Temporal evolution of generalization during learning in linear networks. *Neural Computation*, 3(4):589–603, 1991.

Baldi, P., Chauvin, Y., and Hornik, K. Supervised and unsupervised learning in linear networks. In *International neural network conference*, pp. 825–828. Springer, 1990.

Baldi, P. F. and Hornik, K. Learning in linear neural networks: A survey. *IEEE Transactions on neural networks*, 6(4):837–858, 1995.

Barbier, J. Overlap matrix concentration in optimal bayesian inference. *Information and Inference: A Journal of the IMA*, 10(2):597–623, 2021.

Barbier, J., Krzakala, F., Macris, N., Miolane, L., and Zdeborová, L. Optimal errors and phase transitions in high-dimensional generalized linear models. *Proceedings of the National Academy of Sciences*, 116(12):5451–5460, 2019.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov): 463–482, 2002.

Bauschke, H. H., Combettes, P. L., et al. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

Bayes, T. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, (53):370–418, 1763.

Ben Arous, G., Dembo, A., and Guionnet, A. Cugliandolo-kurchan equations for dynamics of spin-glasses. *Probability theory and related fields*, 136(4):619–660, 2006.

Berezin, F. A. *Introduction to superanalysis*, volume 9. Springer, Reidel, Dordrecht, 1987.

Berthier, L. and Kurchan, J. Non-equilibrium glass transitions in driven and active matter. *Nature Physics*, 9(5):310–314, 2013.

Biehl, M. and Schwarze, H. Learning by on-line gradient descent. *Journal of Physics A: Mathematical and general*, 28(3):643, 1995.

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Blum, A. and Rivest, R. Training a 3-node neural network is np-complete. *Advances in neural information processing systems*, 1, 1988.

Bodin, A. and Macris, N. Model, sample, and epoch-wise descents: exact solution of gradient flow in the random feature model. *Advances in Neural Information Processing Systems*, 34:21605–21617, 2021a.

Bodin, A. and Macris, N. Rank-one matrix estimation: analytic time evolution of gradient descent dynamics. In *Conference on Learning Theory*, pp. 635–678. PMLR, 2021b.

Bordelon, B. and Pehlevan, C. Learning curves for sgd on structured features. In *International Conference on Learning Representations*, 2021.

Bordelon, B. and Pehlevan, C. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *arXiv preprint arXiv:2205.09653*, 2022.

Bös, S. and Opper, M. Dynamics of training. In *Advances in Neural Information Processing Systems*, pp. 141–147, 1997.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pp. 144–152, New York, NY, USA, 1992. Association for Computing Machinery. ISBN 089791497X. doi: 10.1145/130385.130401. URL https://doi.org/10.1145/130385.130401.

Bottou, L. On-line learning and stochastic approximations. 1999.

Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Bouchaud, J.-P., Cugliandolo, L., Kurchan, J., and Mézard, M. Mode-coupling approximations, glass theory and disordered systems. *Physica A: Statistical Mechanics and its Applications*, 226(3):243–273, 1996. ISSN 0378-4371. doi: https://doi.org/10.1016/0378-4371(95)00423-8. URL https://www.sciencedirect.com/science/article/pii/0378437195004238.

Bouchaud, J.-P., Cugliandolo, L. F., Kurchan, J., and Mézard, M. Out of equilibrium dynamics in spin-glasses and other glassy systems. *Spin glasses and random fields*, 12:161, 1998.

Bouten, M. and Derrida, B. Replica symmetry instability in perceptron models. *Journal of Physics A: Mathematical and General*, 27(17):6021–6025, sep 1994. doi: 10.1088/0305-4470/27/17/033. URL https://doi.org/10.1088/0305-4470/27/17/033.

Breiman, L. Reflections after refereeing papers for nips. In *The Mathematics of Generalization*, pp. 11–15. CRC Press, 2018.

Bös, S. and Opper, M. Dynamics of batch training in a perceptron. *Journal of Physics A: Mathematical and General*, 31(21):4835–4850, may 1998. doi: 10.1088/0305-4470/31/21/004. URL https://doi.org/10.1088/0305-4470/31/21/004.

Cai, J., Huang, M., Li, D., and Wang, Y. Solving phase retrieval with random initial guess is nearly as good as by spectral initialization. *arXiv preprint arXiv:2101.03540*, 2021.

Çakmak, B., Lu, Y. M., and Opper, M. Analysis of random sequential message passing algorithms for approximate inference. *arXiv preprint arXiv:2202.08198*, 2022.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., and Zdeborová, L. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.

Celentano, M., Montanari, A., and Wu, Y. The estimation error of general first order methods. In *Conference on Learning Theory*, pp. 1078–1141. PMLR, 2020.

Celentano, M., Cheng, C., and Montanari, A. The high-dimensional asymptotics of first order methods with random data. *arXiv preprint arXiv:2112.07572*, 2021.

Charbonneau, P. History of rsb interview. *Transcript of an oral interview conducted by Patrick Charbonneau and Francesco Zamponi in 2021,* History of RSB Project*, CAPHES École normale supérieure, Paris*, 2021.

Chaudhari, P. and Soatto, S. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–10. IEEE, 2018.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019 (12):124018, 2019.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Chen, Y., Chi, Y., Fan, J., and Ma, C. Gradient descent with random initialization: fast global convergence for nonconvex phase retrieval. *Mathematical Programming*, 176(1-2):5–37, Feb 2019.

Cheng, X., Yin, D., Bartlett, P., and Jordan, M. Stochastic gradient and langevin processes. In *International Conference on Machine Learning*, pp. 1810–1819. PMLR, 2020.

Chizat, L. and Bach, F. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pp. 3036–3046, 2018.

Chung, S., Lee, D. D., and Sompolinsky, H. Linear readout of object manifolds. *Physical Review E*, 93(6):060301, 2016.

Chung, S., Lee, D. D., and Sompolinsky, H. Classification and geometry of general perceptual manifolds. *Physical Review X*, 8(3):031003, 2018.

Clark, D., Abbott, L., and Chung, S. Credit assignment through broadcasting a global error vector. *Advances in Neural Information Processing Systems*, 34: 10053–10066, 2021.

Cohen, U., Chung, S., Lee, D. D., and Sompolinsky, H. Separability and geometry of object manifolds in deep neural networks. *Nature communications*, 11(1):1–13, 2020.

Coolen, A. C., Kühn, R., and Sollich, P. *Theory of neural information processing systems.* OUP Oxford, 2005.

Copelli, M. and Caticha, N. On-line learning in the committee machine. *Journal of Physics A: Mathematical and General*, 28(6):1615, 1995.

Corbett, J. The pauli problem, state reconstruction and quantum-real numbers. *Reports on Mathematical Physics*, 57:53–68, 02 2006.

Cover, T. M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 1965.

Cramer, H. Mathematical methods of statistics, princeton univ. *Press, Princeton, NJ*, 1946.

Crisanti, A. and Sommers, H. The spherical p-spin interaction spin glass model: the statics, 1992 z. *Phys. B*, 87:341.

Crisanti, A. and Sompolinsky, H. Dynamics of spin systems with randomly asymmetric bonds: Langevin dynamics and a spherical model. *Physical Review A*, 36 (10):4922, 1987.

Crisanti, A. and Sompolinsky, H. Path integral approach to random neural networks. *Physical Review E*, 98(6):062120, 2018.

Crisanti, A., Horner, H., and Sommers, H.-J. The sphericalp-spin interaction spin-glass model. *Zeitschrift für Physik B Condensed Matter*, 92(2):257–271, 1993.

Cugliandolo, L. F. Dynamics of glassy systems. *arXiv preprint cond-mat/0210312*, 2002.

Cugliandolo, L. F. The effective temperature. *Journal of Physics A: Mathematical and Theoretical*, 44(48):483001, 2011.

Cugliandolo, L. F. and Kurchan, J. Analytical solution of the off-equilibrium dynamics of a long-range spin-glass model. *Phys. Rev. Lett.*, 71:173–176, Jul 1993a. doi: 10.1103/PhysRevLett.71.173. URL https://link.aps.org/doi/10.1103/PhysRevLett.71.173.

Cugliandolo, L. F. and Kurchan, J. Analytical solution of the off-equilibrium dynamics of a long-range spin-glass model. *Physical Review Letters*, 71(1):173, 1993b.

Cugliandolo, L. F. and Lecomte, V. Rules of calculus in the path integral representation of white noise langevin equations: the onsager–machlup approach. *Journal of Physics A: Mathematical and Theoretical*, 50(34):345001, 2017.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

d'Ascoli, S., Refinetti, M., and Biroli, G. Optimal learning rate schedules in high-dimensional non-convex optimization problems. *arXiv preprint arXiv:2202.04509*, 2022.

De Dominicis, C. Dynamics as a substitute for replicas in systems with quenched random impurities. *Phys. Rev. B*, 18:4913–4919, Nov 1978. doi: 10.1103/PhysRevB.18.4913. URL https://link.aps.org/doi/10.1103/PhysRevB.18.4913.

Del Giudice, P., Franz, S., and Virasoro, M. Perceptron beyond the limit of capacity. *Journal de Physique*, 50(2):121–134, 1989.

Dellaferrera, G. and Kreiman, G. Error-driven input modulation: Solving the credit assignment problem without a backward pass. *arXiv preprint arXiv:2201.11665*, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

Deng, Z., Kammoun, A., and Thrampoulidis, C. A model of double descent for high-dimensional binary linear classification. *Information and Inference: A Journal of the IMA*.

DeWitt, B. *Supermanifolds*. Cambridge University Press, 1992.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

Dominicis, C. d. Technics of field renormalization and dynamics of critical phenomena. In *J. Phys.(Paris), Colloq*, pp. C1–247, 1976.

Dong, J., Krzakala, F., and Gigan, S. Spectral method for multiplexed phase retrieval and application in optical imaging in complex media. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4963–4967, 2019.

Dong, J., Valzania, L., Maillard, A., Pham, T.-a., Gigan, S., and Unser, M. Phase retrieval: From computational imaging to machine learning. *arXiv preprint arXiv:2204.03554*, 2022.

Donoho, D. L., Maleki, A., and Montanari, A. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45): 18914–18919, 2009.

Dotsenko, V. *Introduction to the Replica Theory of Disordered Statistical Systems*. Collection Alea-Saclay: Monographs and Texts in Statistical Physics. Cambridge University Press, 2000. doi: 10.1017/CBO9780511524592.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12 (7), 2011.

Efetov, K. Supersymmetry and theory of disordered metals. *advances in Physics*, 32(1):53–127, 1983.

Eissfeller, H. and Opper, M. New method for studying the dynamics of disordered spin systems without finite-size effects. *Physical review letters*, 68(13):2094, 1992.

Eissfeller, H. and Opper, M. Mean-field monte carlo approach to the sherrington-kirkpatrick model with asymmetric couplings. *Physical Review E*, 50(2):709, 1994.

Ellis, R. S. Large deviations for a general class of random vectors. *The Annals of Probability*, 12(1):1–12, 1984.

Ellis, R. S. *Entropy, large deviations, and statistical mechanics*, volume 1431. Taylor & Francis, 2006.

Engel, A. and Van den Broeck, C. *Statistical Mechanics of Learning*. Cambridge University Press, 2001. doi: 10.1017/CBO9781139164542.

Feng, Y. and Tu, Y. The inverse variance–flatness relation in stochastic gradient descent is critical for finding flat minima. *Proceedings of the National Academy of Sciences*, 118(9), 2021. ISSN 0027-8424. doi: 10.1073/pnas.2015617118.

Fienup, J. R. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15): 2758–2769, 1982.

Folena, G., Franz, S., and Ricci-Tersenghi, F. Rethinking mean-field glassy dynamics and its relation with the energy landscape: The surprising case of the spherical mixed p-spin model. *Physical Review X*, 10(3):031045, 2020.

Frankle, J., Schwab, D. J., and Morcos, A. S. The early phase of neural network training. In *International Conference on Learning Representations*, 2019.

Franz, S., Amit, D. J., and Virasoro, M. A. Prosopagnosia in high capacity neural networks storing uncorrelated classes. *Journal de Physique*, 51(5):387–408, 1990.

Franz, S., Parisi, G., Sevelev, M., Urbani, P., and Zamponi, F. Universality of the sat-unsat (jamming) threshold in non-convex continuous constraint satisfaction problems. *SciPost Physics*, 2(3):019, 2017.

Franz, S., Hwang, S., and Urbani, P. Jamming in multilayer supervised learning models. *Physical review letters*, 123(16):160602, 2019a.

Franz, S., Hwang, S., and Urbani, P. Jamming in multilayer supervised learning models. *Physical review letters*, 123(16):160602, 2019b.

Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Gabrié, M. Mean-field inference methods for neural networks. *Journal of Physics A: Mathematical and Theoretical*, 53(22):223002, 2020.

Gabrié, M., Dani, V., Semerjian, G., and Zdeborová, L. Phase transitions in the q-coloring of random hypergraphs. *Journal of Physics A: Mathematical and Theoretical*, 50(50):505002, 2017.

Gardner, E. Maximum storage capacity in neural networks. *Europhysics Letters (EPL)*, 4(4):481–485, aug 1987. doi: 10.1209/0295-5075/4/4/016. URL https://doi.org/10.1209/0295-5075/4/4/016.

Gardner, E. 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. URL https://doi.org/10.1088/0305-4470/21/1/030.

Gardner, E. and Derrida, B. Optimal storage properties of neural network models. *Journal of Physics A: Mathematical and general*, 21(1):271, 1988.

Gardner, E. and Derrida, B. Three unfinished works on the optimal storage capacity of networks. *Journal of Physics A: Mathematical and General*, 22(12):1983, 1989.

Gärtner, J. On large deviations from the invariant measure. *Theory of Probability & Its Applications*, 22(1):24–39, 1977.

Geiger, M., Spigler, S., d'Ascoli, S., Sagun, L., Baity-Jesi, M., Biroli, G., and Wyart, M. Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Physical Review E*, 100(1):012115, 2019.

Georges, A., Kotliar, G., Krauth, W., and Rozenberg, M. J. Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions. *Reviews of Modern Physics*, 68(1):13, 1996.

Gerace, F., Loureiro, B., Krzakala, F., Mézard, M., and Zdeborová, L. Generalisation error in learning with random features and the hidden manifold model. In *International Conference on Machine Learning*, pp. 3452–3462. PMLR, 2020.

Gilboa, D., Chang, B., Chen, M., Yang, G., Schoenholz, S. S., Chi, E. H., and Pennington, J. Dynamical isometry and a mean field theory of lstms and grus, 2019.

Goldt, S., Advani, M., Saxe, A. M., Krzakala, F., and Zdeborová, L. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. In *Advances in Neural Information Processing Systems*, pp. 6979–6989, 2019.

Goldt, S., Mézard, M., Krzakala, F., and Zdeborová, L. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X*, 10(4):041044, 2020.

Goldt, S., Loureiro, B., Reeves, G., Krzakala, F., Mézard, M., and Zdeborová, L. The gaussian equivalence of generative models for learning with shallow neural networks. In *Mathematical and Scientific Machine Learning*, pp. 426–471. PMLR, 2022.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. 2016.

Gordon, Y. Some inequalities for Gaussian processes and applications. *Israel Journal of Mathematics*, 50(4):265–289, Dec 1985.

Grassberger, P. and Nadal, J.-P. *From statistical physics to statistical inference and back*, volume 428. Springer Science & Business Media, 2012.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018.

Gurbuzbalaban, M., Simsekli, U., and Zhu, L. The heavy-tail phenomenon in sgd. In *International Conference on Machine Learning*, pp. 3964–3975. PMLR, 2021.

Györgyi, G. First-order transition to perfect generalization in a neural network with binary synapses. *Physical Review A*, 41(12):7097, 1990.

Györgyi, G. Techniques of replica symmetry breaking and the storage problem of the mcculloch–pitts neuron. *Physics Reports*, 342(4-5):263–392, 2001.

Györgyi, G. and Reimann, P. Beyond storage capacity in a single model neuron: Continuous replica symmetry breaking. *Journal of Statistical Physics*, 101(1): 679–702, 2000.

HaoChen, J. Z., Wei, C., Lee, J. D., and Ma, T. Shape matters: Understanding the implicit bias of the noise covariance. *arXiv preprint arXiv:2006.08680*, 2020.

Hardy, G. H., Littlewood, J. E., Pólya, G., Pólya, G., et al. *Inequalities*. Cambridge university press, 1952.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2): 949–986, 2022.

He, F., Liu, T., and Tao, D. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. *Advances in Neural Information Processing Systems*, 32, 2019.

Hebb, D. O. *The Organization of Behavior. A Neuropsychological Theory.* John Wiley and Sons, Inc., New York., 1949.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hodgkinson, L. and Mahoney, M. Multiplicative noise and heavy tails in stochastic optimization. In *International Conference on Machine Learning*, pp. 4262–4274. PMLR, 2021.

Hoerl, A. E. and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. doi: 10.1073/pnas.79.8.2554. URL https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554.

Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Houghton, A., Jain, S., and Young, A. P. Role of initial conditions in the mean-field theory of spin-glass dynamics. *Phys. Rev. B*, 28:2630–2637, Sep 1983. doi: 10.1103/PhysRevB.28.2630. URL https://link.aps.org/doi/10.1103/PhysRevB.28.2630.

Hu, W., Li, C. J., Li, L., and Liu, J.-G. On the diffusion approximation of nonconvex stochastic gradient descent. *Annals of Mathematical Sciences and Applications*, 4 (1), 2019.

Hwang, S. and Ikeda, H. Force balance controls the relaxation time of the gradient descent algorithm in the satisfiable phase. *Physical Review E*, 101(5):052308, 2020.

Iba, Y. The nishimori line and bayesian statistics. *Journal of Physics A: Mathematical and General*, 32(21):3875, 1999.

J. Kurchan. Supersymmetry in spin glass dynamics. *J. Phys. I France*, 2(7):1333–1352, 1992. doi: 10.1051/jp1:1992214. URL https://doi.org/10.1051/jp1:1992214.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Janssen, H. On a lagrangean for classical field dynamics and renormalization group calculations of dynamical critical properties. *Zeitschrift für Physik B Condensed Matter and Quanta*, 23:377–380, 01 1976. doi: 10.1007/BF01316547.

Jastrzebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. Three factors influencing minima in SGD. *arXiv preprint arXiv:1711.04623*, 2017. Artificial Neural Networks and Machine Learning, ICANN.

Javanmard, A. and Montanari, A. State evolution for general approximate message passing algorithms, with applications to spatial coupling. *Information and Inference: A Journal of the IMA*, 2(2):115–144, 2013.

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kini, G. R. and Thrampoulidis, C. Analytic study of double descent in binary classification: The impact of loss. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2527–2532. IEEE, 2020.

Kini, G. R. and Thrampoulidis, C. Phase transitions for one-vs-one and one-vs-all linear separability in multiclass gaussian mixtures. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4020–4024, 2021. doi: 10.1109/ICASSP39728.2021.9414099.

Kinouchi, O. and Caticha, N. Optimal generalization in perceptions. *Journal of Physics A: mathematical and General*, 25(23):6243, 1992.

Kinzel, W. and Rujan, P. Improving a network generalization ability by selecting examples. *EPL (Europhysics Letters)*, 13(5):473, 1990.

Kirkpatrick, T. R. and Thirumalai, D. p-spin-interaction spin-glass models: Connections with the structural glass problem. *Physical Review B*, 36(10):5388, 1987.

Krauth, W. and Mézard, M. Storage capacity of memory networks with binary couplings. *Journal de Physique*, 50(20):3057–3066, 1989.

Krishnamurthy, K., Can, T., and Schwab, D. J. Theory of gating in recurrent neural networks. *arXiv preprint arXiv:2007.14823*, 2020.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 5(4):1, 2010.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Krogh, A. and Hertz, J. A. Generalization in a linear perceptron in the presence of noise. *Journal of Physics A: Mathematical and General*, 25(5):1135, 1992.

Krzakala, F. and Zdeborová, L. On melting dynamics and the glass transition. ii. glassy dynamics as a melting process. *The Journal of chemical physics*, 134(3): 034513, 2011.

Krzakała, F., Montanari, A., Ricci-Tersenghi, F., Semerjian, G., and Zdeborová, L. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proceedings of the National Academy of Sciences*, 104(25):10318–10323, 2007.

Kurchan, J. Rheology, and how to stop aging. *Jamming and Rheology: Constrained Dynamics on Microscopic and Macroscopic Scales*, 72, 1997.

Kurchan, J. Supersymmetry, replica and dynamic treatments of disordered systems: a parallel presentation. *arXiv preprint cond-mat/0209399*, 2002.

Launay, J., Poli, I., Boniface, F., and Krzakala, F. *Proceeding of the 2020 Advances in Neural Information Processing Systems*, 33(CONF):9346–9360, 2020.

Le Cun, Y., Kanter, I., and Solla, S. A. Eigenvalues of covariance matrices: Application to neural-network learning. *Physical Review Letters*, 66(18):2396, 1991.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pp. 319–345. Springer, 1999.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Lee, S., Goldt, S., and Saxe, A. Continual learning in the teacher-student setup: Impact of task similarity. In *International Conference on Machine Learning*, pp. 6109–6119. PMLR, 2021.

Lee, S., Mannelli, S. S., Clopath, C., Goldt, S., and Saxe, A. Maslow's hammer for catastrophic forgetting: Node re-use vs node activation. *arXiv preprint arXiv:2205.09029*, 2022.

Lelarge, M. and Miolane, L. Asymptotic bayes risk for gaussian mixture in a semi-supervised setting. In *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 639–643. IEEE, 2019.

Li, Q. and Sompolinsky, H. Statistical mechanics of deep linear neural networks: The backpropagating kernel renormalization. *Physical Review X*, 11(3):031059, 2021.

Li, Q., Tai, C., and Weinan, E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pp. 2101–2110. PMLR, 2017.

Li, Z., Malladi, S., and Arora, S. On the validity of modeling sgd with stochastic differential equations (sdes). *arXiv preprint arXiv:2102.12470*, 2021.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7, 2016.

Liu, S., Papailiopoulos, D., and Achlioptas, D. Bad global minima exist and sgd can reach them. *arXiv preprint arXiv:1906.02613*, 2019.

Loi, D., Mossa, S., and Cugliandolo, L. F. Effective temperature of active matter. *Physical Review E*, 77(5):051111, 2008.

Loureiro, B., Sicuro, G., Gerbelot, C., Pacco, A., Krzakala, F., and Zdeborová, L. Learning gaussian mixtures with generalized linear models: Precise asymptotics in high-dimensions. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 10144–10157. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/543e83748234f7cbab21aa0ade66565f-Paper.pdf.

Loureiro, B., Gerbelot, C., Refinetti, M., Sicuro, G., and Krzakala, F. Fluctuations, bias, variance & ensemble of learners: Exact asymptotics for convex losses in high-dimension. *arXiv preprint arXiv:2201.13383*, 2022.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

Luo, W., Alghamdi, W., and Lu, Y. M. Optimal spectral initialization for signal recovery with applications to phase retrieval. *IEEE Transactions on Signal Processing*, 67(9):2347–2356, 2019.

Lyu, K. and Li, J. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2019.

Ma, C., Wang, K., Chi, Y., and Chen, Y. Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval and matrix completion. In *International Conference on Machine Learning*, pp. 3345–3354. PMLR, 2018.

Ma, J., Xu, J., and Maleki, A. Optimization-based amp for phase retrieval: The impact of initialization and l2 regularization. *IEEE Transactions on Information Theory*, 65(6):3600–3629, 2019.

Mai, X. and Liao, Z. High dimensional classification via regularized and unregularized empirical risk minimization: Precise error and optimal loss. *arXiv preprint arXiv:1905.13742*, 2019.

Maillard, A., Arous, G. B., and Biroli, G. Landscape complexity for the empirical risk of generalized linear models. In *Mathematical and Scientific Machine Learning*, pp. 287–327. PMLR, 2020.

Maimbourg, T., Kurchan, J., and Zamponi, F. Solution of the dynamics of liquids in the large-dimensional limit. *Physical review letters*, 116(1):015902, 2016.

Majer, P., Engel, A., and Zippelius, A. Perceptrons above saturation. *Journal of Physics A: Mathematical and General*, 26(24):7405, 1993.

Mallat, S. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203, 2016. doi: 10.1098/rsta.2015.0203. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2015.0203.

Manacorda, A., Schehr, G., and Zamponi, F. Numerical solution of the dynamical mean field theory of infinite-dimensional equilibrium liquids. *The Journal of Chemical Physics*, 152(16):164506, 2020.

Mandal, R. and Sollich, P. How to study a persistent active glassy system. *Journal of Physics: Condensed Matter*, 33(18):184001, 2021.

Mandt, S., Hoffman, M. D., and Blei, D. M. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

Mannelli, S. S., Krzakala, F., Urbani, P., and Zdeborova, L. Passed & spurious: Descent algorithms and local minima in spiked matrix-tensor models. In *international conference on machine learning*, pp. 4333–4342, 2019.

Mannelli, S. S., Biroli, G., Cammarota, C., Krzakala, F., Urbani, P., and Zdeborová, L. Marvels and pitfalls of the langevin algorithm in noisy high-dimensional inference. *Physical Review X*, 10(1):011057, 2020a.

Mannelli, S. S., Biroli, G., Cammarota, C., Krzakala, F., Urbani, P., and Zdeborová, L. Complex Dynamics in Simple Neural Networks: Understanding Gradient Flow in Phase Retrieval. In *NeurIPS 2020*, 2020b.

Mannelli, S. S., Vanden-Eijnden, E., and Zdeborová, L. Optimization and generalization of shallow neural networks with quadratic activation functions. *arXiv preprint arXiv:2006.15459*, 2020c.

Mannelli, S. S., Gerace, F., Rostamzadeh, N., and Saglietti, L. Inducing bias is simpler than you think. *arXiv preprint arXiv:2205.15935*, 2022.

Marangi, C., Biehl, M., and Solla, S. A. Supervised learning from clustered input examples. *EPL (Europhysics Letters)*, 30(2):117, 1995.

Martin, C. H. and Mahoney, M. W. Traditional and heavy-tailed self regularization in neural network models. *arXiv preprint arXiv:1901.08276*, 2019.

Martin, P. C., Siggia, E. D., and Rose, H. A. Statistical dynamics of classical systems. *Phys. Rev. A*, 8:423–437, Jul 1973. doi: 10.1103/PhysRevA.8.423. URL https://link.aps.org/doi/10.1103/PhysRevA.8.423.

McCulloch, W. S. and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

Mehta, P., Bukov, M., Wang, C.-H., Day, A. G., Richardson, C., Fisher, C. K., and Schwab, D. J. A high-bias, low-variance introduction to machine learning for physicists. *Physics reports*, 810:1–124, 2019.

Mei, S. and Montanari, A. 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.

Mei, S., Montanari, A., and Nguyen, P.-M. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115 (33):E7665–E7671, 2018.

Mei, S., Misiakiewicz, T., and Montanari, A. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory*, pp. 2388–2464. PMLR, 2019.

Mézard, M. Mean-field message-passing equations in the hopfield model and its generalizations. *Physical Review E*, 95(2):022117, 2017.

Mezard, M. and Montanari, A. *Information, physics, and computation.* Oxford University Press, 2009.

Mézard, M., Parisi, G., and Virasoro, M. A. *Spin glass theory and beyond.* World Scientific, Singapore, 1987.

Millane, R. P. Phase retrieval in crystallography and optics. *J. Opt. Soc. Am. A*, 7 (3):394–411, Mar 1990.

Minsky, M. and Papert, S. An introduction to computational geometry. *Cambridge tiass., HIT*, 479:480, 1969.

Mondelli, M. and Montanari, A. Fundamental limits of weak recovery with applications to phase retrieval. In *Conference On Learning Theory*, pp. 1445–1450. PMLR, 2018.

Mondelli, M., Thrampoulidis, C., and Venkataramanan, R. Optimal combination of linear and spectral estimators for generalized linear models. *arXiv preprint arXiv:2008.03326*, 2020.

Montanari, A. and Sen, S. A short tutorial on mean-field spin glass techniques for non-physicists. *arXiv preprint arXiv:2204.02909*, 2022.

Nacson, M. S., Srebro, N., and Soudry, D. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3051–3059. PMLR, 2019.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Nesterov, Y. E. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pp. 543–547, 1983.

Neyshabur, B. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, PhD thesis., 2017.

Nicolas, A., Ferrero, E. E., Martens, K., and Barrat, J.-L. Deformation and flow of amorphous solids: Insights from elastoplastic models. *Reviews of Modern Physics*, 90(4):045006, 2018.

Nishimori, H. *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. International series of monographs on physics. Oxford University Press, 2001. ISBN 9780198509400. URL https://books.google.fr/books?id=nO0T1VzfhZcC.

Nøkland, A. Direct feedback alignment provides learning in deep neural networks. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/d490d7b4576290fa60eb31b5fc917ad1-Paper.pdf.

Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Abolafia, D. A., Pennington, J., and Sohl-dickstein, J. Bayesian deep convolutional networks with many channels are gaussian processes. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1g30j0qF7.

Opper, M. and Diederich, S. Phase transition and 1/f noise in a game dynamical model. *Physical review letters*, 69(10):1616, 1992.

Opper, M. and Haussler, D. Calculation of the learning curve of bayes optimal classification algorithm for learning a perceptron with noise. In *COLT*, volume 91, pp. 75–87, 1991.

Opper, M. and Kinzel, W. Statistical mechanics of generalization. In *Models of neural networks III*, pp. 151–209. Springer, 1996.

Opper, M. and Saad, D. *Advanced mean field methods: Theory and practice*. MIT press, 2001.

Papyan, V., Han, X., and Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Parikh, N., Boyd, S., et al. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

Parisi, G. Infinite number of order parameters for spin-glasses. *Physical Review Letters*, 43(23):1754, 1979.

Parisi, G. A sequence of approximated solutions to the sk model for spin glasses. *Journal of Physics A: Mathematical and General*, 13(4):L115, 1980.

Parisi, G. Order parameter for spin-glasses. *Physical Review Letters*, 50(24):1946, 1983.

Parisi, G., Urbani, P., and Zamponi, F. *Theory of Simple Glasses: Exact Solutions in Infinite Dimensions*. Cambridge University Press, 2020.

Pearce, M. T., Agarwala, A., and Fisher, D. S. Stabilization of extensive fine-scale diversity by ecologically driven spatiotemporal chaos. *Proceedings of the National Academy of Sciences*, 117(25):14572–14583, 2020.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Percus, A., Istrate, G., and Moore, C. *Computational complexity and statistical physics*. OUP USA, 2006.

Pesme, S., Pillaud-Vivien, L., and Flammarion, N. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. *Advances in Neural Information Processing Systems*, 34:29218–29230, 2021.

Pittorino, F., Lucibello, C., Feinauer, C., Perugini, G., Baldassi, C., Demyanenko, E., and Zecchina, R. Entropic gradient descent algorithms and wide flat minima. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124015, 2021.

Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3360–3368. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6322-exponential-expressivity-in-deep-neural-networks-through-transient-chaos.pdf.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. On the expressive power of deep neural networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2847–2854. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/raghu17a.html.

Rangan, S. Generalized approximate message passing for estimation with random linear mixing. In *2011 IEEE International Symposium on Information Theory Proceedings*, pp. 2168–2172. IEEE, 2011.

Rao, C. R. Information and the accuracy attainable in the estimation of statistical parameters. *Reson. J. Sci. Educ*, 20:78–90, 1945.

Refinetti, M. and Goldt, S. The dynamics of representation learning in shallow, non-linear autoencoders. *arXiv preprint arXiv:2201.02115*, 2022.

Refinetti, M., d'Ascoli, S., Ohana, R., and Goldt, S. Align, then memorise: the dynamics of learning with feedback alignment. In *International Conference on Machine Learning*, pp. 8925–8935. PMLR, 2021a.

Refinetti, M., Goldt, S., Krzakala, F., and Zdeborová, L. Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed. In *International Conference on Machine Learning*, pp. 8936–8947. PMLR, 2021b.

Ricci-Tersenghi, F. Measuring the fluctuation-dissipation ratio in glassy systems with no perturbing field. *Physical Review E*, 68(6):065104, 2003.

Ricci-Tersenghi, F., Semerjian, G., and Zdeborová, L. Typology of phase transitions in bayesian inference problems. *Physical Review E*, 99(4):042109, 2019.

Riegler, P. and Biehl, M. On-line backpropagation in two-layered neural networks. *Journal of Physics A: Mathematical and General*, 28(20):L507, 1995.

Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Rosset, S., Zhu, J., and Hastie, T. J. Margin maximizing loss functions. In *Advances in neural information processing systems*, pp. 1237–1244, 2004.

Rotskoff, G. M. and Vanden-Eijnden, E. Trainability and accuracy of neural networks: An interacting particle system approach. *arXiv preprint arXiv:1805.00915*, 2018.

Roy, F., Biroli, G., Bunin, G., and Cammarota, C. Numerical implementation of dynamical mean field theory for disordered systems: application to the lotka–volterra model of ecosystems. *Journal of Physics A: Mathematical and Theoretical*, 52(48):484001, 2019.

Roy, F., Barbier, M., Biroli, G., and Bunin, G. Complex interactions can create persistent fluctuations in high-diversity ecosystems. *PLoS computational biology*, 16(5):e1007827, 2020.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Saad, D. *On-line learning in neural networks*, volume 17. Cambridge University Press, 2009.

Saad, D. and Solla, S. Dynamics of on-line gradient descent learning for multilayer neural networks. *Advances in neural information processing systems*, 8, 1995a.

Saad, D. and Solla, S. A. On-line learning in soft committee machines. *Physical Review E*, 52(4):4225, 1995b.

Saad, D. and Solla, S. A. Exact solution for on-line learning in multilayer neural networks. *Physical Review Letters*, 74(21):4337, 1995c.

Safran, I. and Shamir, O. Spurious local minima are common in two-layer relu neural networks. *arXiv preprint arXiv:1712.08968*, 2017.

Saglietti, L. and Zdeborová, L. Solvable model for inheriting the regularization through knowledge distillation. In *Mathematical and Scientific Machine Learning*, pp. 809–846. PMLR, 2022.

Saglietti, L., Mannelli, S. S., and Saxe, A. An analytical theory of curriculum learning in teacher-student networks. *arXiv preprint arXiv:2106.08068*, 2021.

Sarao Mannelli, S. and Urbani, P. Analytical study of momentum-based acceleration methods in paradigmatic high-dimensional non-convex problems. *Advances in Neural Information Processing Systems*, 34:187–199, 2021.

Sarao Mannelli, S., Biroli, G., Cammarota, C., Krzakala, F., and Zdeborová, L. Who is afraid of big bad minima? analysis of gradient-flow in spiked matrix-tensor models. *Advances in Neural Information Processing Systems*, 32, 2019.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep information propagation, 2017.

Seddik, M. E. A., Louart, C., Tamaazousti, M., and Couillet, R. Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures. In *International Conference on Machine Learning*, pp. 8573–8582. PMLR, 2020.

Seung, H. S., Sompolinsky, H., and Tishby, N. Statistical mechanics of learning from examples. *Phys. Rev. A*, 45:6056–6091, Apr 1992a. doi: 10.1103/PhysRevA.45.6056. URL https://link.aps.org/doi/10.1103/PhysRevA.45.6056.

Seung, H. S., Sompolinsky, H., and Tishby, N. Statistical mechanics of learning from examples. *Physical review A*, 45(8):6056, 1992b.

Şimşekli, U., Gürbüzbalaban, M., Nguyen, T. H., Richard, G., and Sagun, L. On the heavy-tailed theory of stochastic gradient descent for deep neural networks. *arXiv preprint arXiv:1912.00018*, 2019.

Simsekli, U., Sagun, L., and Gurbuzbalaban, M. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pp. 5827–5837. PMLR, 2019.

Sirignano, J. and Spiliopoulos, K. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020.

Sompolinsky, H. and Zippelius, A. Relaxational dynamics of the edwards-anderson model and the mean-field theory of spin-glasses. *Phys. Rev. B*, 25:6860–6875, Jun 1982. doi: 10.1103/PhysRevB.25.6860. URL https://link.aps.org/doi/10.1103/PhysRevB.25.6860.

Sompolinsky, H., Crisanti, A., and Sommers, H.-J. Chaos in random neural networks. *Physical review letters*, 61(3):259, 1988.

Sompolinsky, H., Tishby, N., and Seung, H. S. Learning from examples in large neural networks. *Physical Review Letters*, 65(13):1683, 1990.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018a.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018b.

Sur, P. and Candès, E. J. A modern maximum-likelihood theory for high-dimensional logistic regression. *Proceedings of the National Academy of Sciences*, 116(29):14516–14525, 2019.

Tan, Y. S. and Vershynin, R. Online stochastic gradient descent with arbitrary initialization solves non-smooth, non-convex phase retrieval. *arXiv preprint arXiv:1910.12837*, 2019.

Thomas, V., Pedregosa, F., Merriënboer, B., Manzagol, P.-A., Bengio, Y., and Le Roux, N. On the interplay between noise and curvature and its effect on optimization and generalization. In *International Conference on Artificial Intelligence and Statistics*, pp. 3503–3513. PMLR, 2020.

Thrampoulidis, C. Theoretical insights into multiclass classification: A high-dimensional asymptotic view. *Neural Information Processing Systems (NeurIPS 2020)*, 2020.

Thrampoulidis, C., Oymak, S., and Hassibi, B. Regularized linear regression: A precise analysis of the estimation error. In *Proceedings of The 28th Conference on Learning Theory*, volume 40, pp. 1683–1709, Paris, France, 03–06 Jul 2015. PMLR.

Thrampoulidis, C., Abbasi, E., and Hassibi, B. Precise error analysis of regularized $m$-estimators in high dimensions. *IEEE Transactions on Information Theory*, 64 (8):5592–5628, 2018.

Touchette, H. The large deviation approach to statistical mechanics. *Physics Reports*, 478(1-3):1–69, 2009.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9446–9454, 2018.

Urbani, P. Statistical physics of glassy systems: tools and applications, 2018.

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27(11): 1134–1142, 1984.

Vapnik, V. Principles of risk minimization for learning theory. In Moody, J., Hanson, S., and Lippmann, R. P. (eds.), *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1992.

Vapnik, V. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999a. doi: 10.1109/72.788640.

Vapnik, V. *The nature of statistical learning theory*. Springer science & business media, 1999b.

Veiga, R., Stephan, L., Loureiro, B., Krzakala, F., and Zdeborová, L. Phase diagram of stochastic gradient descent in high-dimensional two-layer neural networks. *arXiv preprint arXiv:2202.00293*, 2022.

Vicente, R., Kinouchi, O., and Caticha, N. Statistical mechanics of online learning of drifting concepts: A variational approach. *Machine Learning*, 32(2):179–201, 1998.

Walther, A. The question of phase retrieval in optics. *Optica Acta: International Journal of Optics*, 10(1):41–49, 1963.

Wang, K., Muthukumar, V., and Thrampoulidis, C. Benign overfitting in multiclass classification: All roads lead to interpolation. *arXiv preprint arXiv:2106.10865*, 2021.

Watkin, T. L. H., Rau, A., and Biehl, M. The statistical mechanics of learning a rule. *Rev. Mod. Phys.*, 65:499–556, Apr 1993. doi: 10.1103/RevModPhys.65.499. URL https://link.aps.org/doi/10.1103/RevModPhys.65.499.

Wei, M. and Schwab, D. J. How noise affects the hessian spectrum in overparameterized neural networks. *arXiv preprint arXiv:1910.00195*, 2019.

Wong, R. Computer science and scientific computing. *Asymptotic approximations of integrals*, 1989.

Woodworth, B., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673. PMLR, 2020.

Wu, B., Chen, W., Fan, Y., Zhang, Y., Hou, J., Liu, J., and Zhang, T. Tencent ml-images: A large-scale multi-label image database for visual representation learning. *IEEE Access*, 7:172683–172693, 2019.

Xu, H., Caramanis, C., and Mannor, S. Robustness and regularization of support vector machines. *Journal of machine learning research*, 10(7), 2009.

Yaida, S. Fluctuation-dissipation relations for stochastic gradient descent. *arXiv preprint arXiv:1810.00004*, 2018.

Yang, G., Pennington, J., Rao, V., Sohl-Dickstein, J., and Schoenholz, S. S. A mean field theory of batch normalization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyMDXnCcF7.

Zdeborová, L. New tool in the box. *Nature Physics*, 13(5):420–421, 2017.

Zdeborová, L. Understanding deep learning is also a job for physicists. *Nature Physics*, 16(6):602–604, 2020.

Zdeborová, L. and Krzakala, F. Statistical physics of inference: Thresholds and algorithms. *Advances in Physics*, 65(5):453–552, 2016.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *ArXiv*, abs/1611.03530, 2017.

Zhu, Z., Wu, J., Yu, B., Wu, L., and Ma, J. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. *arXiv preprint arXiv:1803.00195*, 2018.

Zhu, Z., Wu, J., Yu, B., Wu, L., and Ma, J. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *International Conference on Machine Learning*, pp. 7654–7663. PMLR, 2019.

Zinn-Justin, J. *Quantum Field Theory and Critical Phenomena (4th edition)*, volume 113 of *International Series of Monographs on Physics*. Clarendon Press, Oxford, 2002. URL https://hal.archives-ouvertes.fr/hal-00120423. URL: http://www-spht.cea.fr/articles/t02/002.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Physique en
Île-de-France (PIF)

**Titre:** Modélisation physique statistique de la dynamique et de la généralisation dans les réseaux de neurones artificiels ..........

**Mots clés:** Réseaux de neurones artificiels, systèmes désordonnés, dynamique d'apprentissage, algorithme du gradient stochastique

**Résumé:** L'apprentissage machine est une technologie désormais omniprésente dans notre quotidien. Toutefois, ce domaine reste encore largement empirique et ses enjeux scientifiques manquent d'une compréhension théorique profonde. Cette thèse se penche vers la découverte des mécanismes sous-tendant l'apprentissage dans les réseaux de neurones artificiels à travers le prisme de la physique statistique. Dans une première partie, nous nous intéressons aux propriétés *statiques* des problèmes d'apprentissage, que nous introduisons au chapitre 1.1. Dans le chapitre 1.2, nous considérons la classification d'un mélange binaire de nuages gaussiens et nous dérivons des expressions rigoureuses pour les erreurs en dimension infinie, que nous appliquons pour éclairer le rôle des différents paramètres du problème. Dans le chapitre 1.3, nous montrons comment étendre le modèle de perceptron enseignant-étudiant pour considerer la classification multi-classes, en dérivant des expressions asymptotiques pour la performance optimale et la performance de la minimisation du risque empirique règularisè. Dans la deuxième partie, nous nous concentrons sur la *dynamique* de l'apprentissage, que nous introduisons dans le chapitre 2.1. Dans le chapitre 2.2, nous montrons comment décrire analytiquement la dynamique de l'algorithme du gradient stochastique à échantillonage mini-lots (mini-batch SGD) dans la classification binaire de mélanges gaussiens, en utilisant la théorie dynamique du champ moyen. Le chapitre 2.3 présente une analyse du bruit effectif introduit par SGD. Dans le chapitre 2.4, nous considérons le problème de la récupération des signes comme exemple d'optimisation hautement non convexe et montrons que la stochasticité est cruciale pour la généralisation. La conclusion de la thèse est présentée dans la troisième partie.

**Title:** Statistical physics insights on the dynamics and generalisation of artificial neural networks........

**Keywords:** artificial neural networks, disordered systems, dynamics of learning, stochastic gradient descent

**Abstract:** Machine learning technologies have become ubiquitous in our daily lives. However, this field still remains largely empirical and its scientific stakes lack a deep theoretical understanding. This thesis explores the mechanisms underlying learning in artificial neural networks through the prism of statistical physics. In the first part, we focus on the *static* properties of learning problems, that we introduce in Chapter 1.1. In Chapter 1.2 we consider the prototype classification of a binary mixture of Gaussian clusters and we derive rigorous closed-form expressions for the errors in the infinite-dimensional regime, that we apply to shed light on the role of different problem parameters. In Chapter 1.3, we show how to extend the teacher-student perceptron model to encompass multi-class classification deriving asymptotic expressions for the optimal performance and the performance of regularised empirical risk minimisation. In the second part, we turn our focus to the *dynamics* of learning, that we introduce in Chapter 2.1. In Chapter 2.2, we show how to track analytically the training dynamics of multi-pass stochastic gradient descent (SGD) via dynamical mean-field theory for generic non convex loss functions and Gaussian mixture data. Chapter 2.3 presents a late-time analysis of the effective noise introduced by SGD in the underparametrised and overparametrised regimes. In Chapter 2.4, we take the sign retrieval problem as a benchmark highly non-convex optimisation problem and show that stochasticity is crucial to achieve perfect generalisation. The third part of the thesis contains the conclusions and some future perspectives.