Generative modeling : statistical physics of Restricted Boltzmann Machines, learning with missing information and scalable training of Linear Flows

Giancarlo Fissore

▶ **To cite this version:**

Giancarlo Fissore. Generative modeling : statistical physics of Restricted Boltzmann Machines, learning with missing information and scalable training of Linear Flows. Disordered Systems and Neural Networks [cond-mat.dis-nn]. Université Paris-Saclay, 2022. English. NNT : 2022UPASG028 . tel-03710286

HAL Id: tel-03710286
https://theses.hal.science/tel-03710286v1

Submitted on 30 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

université
PARIS-SACLAY

Generative modeling: statistical physics of Restricted Boltzmann Machines, learning with missing information and scalable training of Linear Flows

Modélisation générative : physique statistique des Machines de Boltzmann Restreintes, apprentissage avec informations manquantes et apprentissage scalable des flux linéaires

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580 : sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Mathématiques et Informatique
Graduate School : Informatique et science du numérique
Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche
Laboratoire interdisciplinaire des sciences du numérique
(Université Paris-Saclay, CNRS),
sous la direction de **Cyril FURTLEHNER**, Chargé de recherche, et le co-encadrement d'**Aurélien DECELLE**, Chargé de recherche

Thèse soutenue à Paris-Saclay, le 9 Mars 2022, par

Giancarlo FISSORE

Composition du jury

Martin WEIGT Professeur des universités, Sorbonne Université	Président
Alexandre ALLAUZEN Professeur des universités, ESPCI	Rapporteur & Examinateur
Carlo BALDASSI Professeur associé, Bocconi University	Rapporteur & Examinateur
Andrew SAXE Directeur de recherche, Gatsby Unit & Sainsbury Wellcome Centre, UCL	Rapporteur & Examinateur
Muneki YASUDA Professeur des universités, Yamagata University	Examinateur
Cyril FURTLEHNER Chargé de recherche, Inria Saclay	Directeur de thèse

THESE DE DOCTORAT
NNT : 2022UPASG028

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Modélisation générative : physique statistique des Machines de Boltzmann Restreintes, apprentissage avec informations manquantes et apprentissage scalable des flux linéaires

Mots clés : Machines de Boltzmann Restreintes, Normalizing Flows, Modèles génératifs, Champ moyen, Informations manquantes, Réseaux de neurones

Résumé : Les modèles de réseaux neuronaux capables d'approximer et d'échantillonner des distributions de probabilité à haute dimension sont connus sous le nom de *modèles génératifs*. Ces dernières années, cette classe de modèles a fait l'objet d'une attention particulière en raison de son potentiel à apprendre automatiquement des représentations significatives de la grande quantité de données que nous produisons et consommons quotidiennement. Cette thèse présente des résultats théoriques et algorithmiques relatifs aux modèles génératifs et elle est divisée en deux parties.

Dans la première partie, nous concentrons notre attention sur la Machine de Boltzmann Restreinte (RBM) et sa formulation en physique statistique. Historiquement, la physique statistique a joué un rôle central dans l'étude des fondements théoriques et dans le développement de modèles de réseaux neuronaux. La première implémentation neuronale d'une mémoire associative (Hopfield, 1982) est un travail séminal dans ce contexte. La RBM peut être considérée comme un développement du modèle de Hopfield, et elle est particulièrement intéressante en raison de son rôle à l'avant-garde de la révolution de l'apprentissage profond (Hinton et al. 2006). En exploitant sa formulation de physique statistique, nous dérivons une théorie de champ moyen de la RBM qui nous permet de caractériser à la fois son fonctionnement en tant que modèle génératif et la dynamique de sa procédure d'apprentissage. Cette analyse s'avère utile pour dériver une stratégie d'imputation robuste de type champ moyen qui permet d'utiliser la RBM pour apprendre des distributions empiriques dans le cas difficile où l'ensemble de données à modéliser n'est que partiellement observé et présente des pourcentages élevés d'informations manquantes.

Dans la deuxième partie, nous considérons une classe de modèles génératifs connus sous le nom de Normalizing Flows (NF), dont la caractéristique distinctive est la capacité de modéliser des distributions complexes à haute dimension en employant des transformations inversibles d'une distribution simple et traitable. L'inversibilité de la transformation permet d'exprimer la densité de probabilité par un changement de variables dont l'optimisation par Maximum de Vraisemblance (ML) est assez simple mais coûteuse en calcul. La pratique courante est d'imposer des contraintes architecturales sur la classe de transformations utilisées pour les NF, afin de rendre l'optimisation par ML efficace. En partant de considérations géométriques, nous proposons un algorithme d'optimisation stochastique par descente de gradient qui exploite la structure matricielle des réseaux de neurones entièrement connectés sans imposer de contraintes sur leur structure autre que la dimensionnalité fixe requise par l'inversibilité. Cet algorithme est efficace en termes de calcul et peut s'adapter à des ensembles de données de très haute dimension. Nous démontrons son efficacité dans l'apprentissage d'une architecture non linéaire multicouche utilisant des couches entièrement connectées.

Title : Generative modeling : statistical physics of Restricted Boltzmann Machines, learning with missing information and scalable training of Linear Flows

Keywords : Restricted Boltzmann Machines, Normalizing Flows, Generative models, Mean field, Missing information, Neural networks

Abstract : Neural network models able to approximate and sample high dimensional probability distributions are known as *generative models*. In recent years this class of models has received tremendous attention due to their potential in automatically learning meaningful representations of the vast amount of data that we produce and consume daily. This thesis presents theoretical and algorithmic results pertaining to generative models and it is divided in two parts.

In the first part, we focus our attention on the Restricted Boltzmann Machine (RBM) and its statistical physics formulation. Historically, statistical physics has played a central role in studying the theoretical foundations and providing inspiration for neural network models. The first neural implementation of an associative memory (Hopfield, 1982) is a seminal work in this context. The RBM can be regarded to as a development of the Hopfield model, and it is of particular interest due to its role at the forefront of the deep learning revolution (Hinton et al. 2006). Exploiting its statistical physics formulation, we derive a mean-field theory of the RBM that allows us to characterize both its functioning as a generative model and the dynamics of its training procedure. This analysis proves useful in deriving a robust mean-field imputation strategy that makes it possible to use the RBM to learn empirical distributions in the challenging case in which the dataset to model is only partially observed and presents high percentages of missing information.

In the second part we consider a class of generative models known as Normalizing Flows (NF), whose distinguishing feature is the ability to model complex high-dimensional distributions by employing invertible transformations of a simple tractable distribution. The invertibility of the transformation allows expressing the probability density through a change of variables, whose optimization by Maximum Likelihood (ML) is rather straightforward but computationally expensive. The common practice is to impose architectural constraints on the class of transformations used for NF, in order to make the ML optimization efficient. Proceeding from geometrical considerations, we propose a stochastic gradient descent optimization algorithm that exploits the matrix structure of fully connected neural networks without imposing any constraints on their structure other than the fixed dimensionality required by invertibility. This algorithm is computationally efficient and can scale to very high dimensional datasets. We demonstrate its effectiveness in training a multilayer nonlinear architecture employing fully connected layers.

Résumé en français

Une brève histoire de l'intelligence artificielle Le domaine de l'*intelligence artificielle* (IA) a considérablement changé et évolué au cours des dernières décennies. Le même terme "intelligence artificielle" a pris différentes significations suivant les progrès de l'époque, étant le plus souvent défini de façon très vague. Nous identifions la genèse du domaine avec la première tentative de modélisation mathématique des neurones biologiques, proposée par McCulloch et Pitts dans les années 1940. Leur approche s'inspirait du cerveau biologique et s'inscrivait dans un ensemble d'efforts visant à construire un cerveau artificiel capable d'une intelligence "humaine", alimentés par les nouvelles avancées et l'émergence de disciplines scientifiques modernes telles que la théorie du calcul, les neurosciences et la cybernétique (Wiener, 1948) qui ont caractérisé le paysage scientifique de ces années-là.

Le premier écart par rapport à ce paradigme apparaît dès les années 1950, lorsqu'une nouvelle approche est proposée pour "procéder sur la base de la conjecture selon laquelle chaque aspect de l'apprentissage ou de toute autre caractéristique de l'intelligence peut en principe être décrit si précisément qu'une machine peut être faite pour le simuler" (McCarthy, 1955). Ce changement de paradigme est une conséquence directe de l'avènement de l'ordinateur moderne et il s'agit d'une approche plus abstraite de l'IA, car elle repose sur l'hypothèse que l'intelligence peut être construite et simulée sur la base de la logique et des systèmes formels. Les premières mises en œuvre réussies de systèmes d'IA sont dues à cette approche, qui est devenue connue sous le nom d'IA symbolique et a représenté le paradigme dominant pendant des décennies. Cependant, malgré une série de succès initiaux, l'approche de l'IA symbolique a eu du mal à faire face à certains problèmes centraux de la cognition humaine, tels que la perception, l'apprentissage et la reconnaissance des formes. Ces problèmes fondamentaux semblaient être plus facilement abordés par les réseaux neuronaux artificiels, systèmes d'unités de calcul du type du neurone de McCulloch et Pitts, qui ont suscité un renouvellement de l'intérêt pour ce type de modèles dans les années 1980.

C'est à ce moment que l'intérêt pour les réseaux neuronaux a commencé à s'étendre à d'autres domaines, notamment en ce qui concerne la science nouvellement établie de la *complexité*. Dans ce contexte, des contributions substantielles aux fondements théoriques et au développement des réseaux neuronaux sont venues du domaine de la physique statistique. Parmi celles-ci, Hopfield (1982) a introduit le premier modèle de réseau neuronal fonctionnel pour une mémoire associative. Peu à peu, l'objectif initial de réaliser un cerveau artificiel s'est estompé, et toute une classe de modèles utilisant des réseaux neuronaux artificiels complétés par des outils mathématiques et statistiques est apparue comme un domaine riche en soi. Désigné sous le nom de *apprentissage automatique statistique*, ce domaine se concentre désormais sur l'ingénierie de systèmes capables d'exploiter les données et les algorithmes de calcul pour résoudre des problèmes complexes.

Dans les années 2010, les progrès technologiques ont déterminé la révolution de l'*apprentissage profond* : la disponibilité de ressources et d'infrastructures de réseau plus puissants pour le calcul, ainsi que l'accès à la quantité massive de données constamment générées par des myriades de dispositifs technologiques interconnectés, ont permis de construire des modèles de réseaux neuronaux de plus en plus grands et puissants exploitant des quantités plus importantes de données pour résoudre les problèmes les plus disparates avec un succès croissant. Le lien avec l'IA se trouve dans le fait que les modèles d'apprentissage profond ont été employés pour résoudre automatiquement des problèmes qui, traditionnellement, nécessitaient l'emploi de l'intelligence humaine, comme la reconnaissance d'objets (Krizhevsky, 2012), la synthèse vocale (VanDenOord, 2018) et la traduction en langage naturel (Devlin, 2019).

Organisation de la thèse Cette thèse est structurée en deux parties principales. Dans la première partie, nous adoptons le point de vue de la physique statistique sur l'analyse des réseaux de neurones. Cette approche est principalement basée sur la théorie des *systèmes désordonnés* (Mezard, 1987), un type prototypique de système complexe particulièrement adapté à la description des réseaux de neurones ; trouvant son origine dans les années 1980, cette ligne de recherche est très active de nos jours. Dans ce contexte, l'étude ici présentée concerne la Machine de Boltzmann Restreinte (RBM),

un réseau neuronal simple qui a été parmi les protagonistes de la phase initiale de la révolution de l'apprentissage profond (Hinton, 2006). Ce réseau est particulièrement intéressant du point de vue de la physique statistique car il peut être interprété comme une généralisation du modèle de Hopfield (Barra, 2012). Dans le Chapitre 1, nous présentons le modèle de Hopfield et discutons des principes de base permettant de dériver une théorie de champ moyen pour les réseaux neuronaux. Dans le Chapitre 2, nous présentons la RBM. Nous commençons à discuter de nos travaux originaux dans le Chapitre 3, avec la présentation d'une description empirique des propriétés dynamiques du RBM. Cela pose la base de l'analyse théorique de champ moyen développée dans le Chapitre 4, en discutant des propriétés d'équilibre et dynamiques. Enfin, dans le Chapitre 5, nous montrons comment adapter la RBM au scénario difficile de l'apprentissage avec des informations manquantes, en tirant parti de l'image théorique présentée dans les sections précédentes.

Dans la deuxième partie, nous nous intéressons aux Flux Normalisateurs (NF) (Papamakarios, 2021), un type de modèle plus récent qui trouve des applications dans des domaines similaires à ceux de la RBM. Nous nous éloignons ici de la physique statistique : dans le Chapitre 6, nous proposons un modèle NF qui utilise des réseaux neuronaux entièrement connectés et nous montrons comment, à partir de simples considérations de géométrie riemannienne, nous pouvons dériver un algorithme efficace pour entraîner le modèle proposé.

Generative modeling: statistical physics of Restricted Boltzmann Machines, learning with missing information and scalable training of Linear Flows

Giancarlo Fissore

June 29, 2022

Abstract

Neural network models able to approximate and sample high dimensional probability distributions are known as *generative models*. In recent years this class of models has received tremendous attention due to their potential in automatically learning meaningful representations of the vast amount of data that we produce and consume daily. This thesis presents theoretical and algorithmic results pertaining to generative models and it is divided in two parts.

In the first part, we focus our attention on the Restricted Boltzmann Machine (RBM) and its statistical physics formulation. Historically, statistical physics has played a central role in studying the theoretical foundations and providing inspiration for neural network models. The first neural implementation of an associative memory (Hopfield, 1982) is a seminal work in this context. The RBM can be regarded to as a development of the Hopfield model, and it is of particular interest due to its role at the forefront of the deep learning revolution (Hinton et al. 2006). Exploiting its statistical physics formulation, we derive a mean-field theory of the RBM that allows us to characterize both its functioning as a generative model and the dynamics of its training procedure. This analysis proves useful in deriving a robust mean-field imputation strategy that makes it possible to use the RBM to learn empirical distributions in the challenging case in which the dataset to model is only partially observed and presents high percentages of missing information.

In the second part we consider a class of generative models known as Normalizing Flows (NF), whose distinguishing feature is the ability to model complex high-dimensional distributions by employing invertible transformations of a simple tractable distribution. The invertibility of the transformation allows expressing the probability density through a change of variables, whose optimization by Maximum Likelihood (ML) is rather straightforward but computationally expensive. The common practice is to impose architectural constraints on the class of transformations used for NF, in order to make the ML optimization efficient.

Proceeding from geometrical considerations, we propose a stochastic gradient descent optimization algorithm that exploits the matrix structure of fully connected neural networks without imposing any constraints on their structure other than the fixed dimensionality required by invertibility. This algorithm is computationally efficient and can scale to very high dimensional datasets. We demonstrate its effectiveness in training a multilayer nonlinear architecture employing fully connected layers.

Contents

Contents **iii**

Introduction **v**
A brief history of Artificial Intelligence v
Thesis organization vi
Contributions vii

I RBM **1**

1 Background **3**
1.1 The mathematical neuron 3
1.2 The Hopfield model 4
1.3 Statistical description of a neural network 7
1.4 Free energy 9
1.5 Order parameters and phase transitions 10
1.6 Effective free energy 11
1.7 Mean-field theory of the Hopfield model 12
1.8 Beyond memory: Boltzmann Machines 16
1.9 Statistical Physics approach to neural networks 18

2 Restricted Boltzmann Machines (RBM) **19**
2.1 Definition 19
2.2 Contrastive divergence training 21
2.3 Mean-field training 22
2.4 Generalized RBM 24

3 Spectral Learning Dynamics of the RBM **27**
3.1 Principal Component Analysis (PCA) 28
3.2 Singular Value Decomposition (SVD) 28
3.3 Linearized mean-field equations for a RBM 29
3.4 Distribution of the singular values 31
3.5 Dynamics of the singular vectors 32
3.6 Characterization of the modes 34

4 Mean-field theory **37**
4.1 Statistical ensemble 37
4.2 Order parameters and effective free energy 38
4.3 Phase diagram 42
4.4 Learning phase 42
4.5 Learning equations 45
4.6 Linear regime 46
4.7 Nonlinear regime 48
4.8 Empirical dynamics 49

5 Missing information **55**
5.1 Multi-output learning with incomplete data 55
5.2 Lossy-CDk 56
5.3 Mean-field imputation 58
5.4 Multi-output classification with missing information 58

II Normalizing Flows **61**

6 Relative gradient optimization **63**
6.1 Invertible transformations of probability densities and Normalizing Flows 63
6.2 Linear flows as Fully Connected neural network layers 65
6.3 Relative gradients 67
6.4 Relative backpropagation 69

Conclusions **75**

Reprints **77**

A Spectral dynamics of learning in restricted Boltzmann machines **79**

B Thermodynamics of Restricted Boltzmann Machines and Related Learning Dynamics **87**

C Robust Multi-Output Learning with Highly Incomplete Data via Restricted Boltzmann Machines **123**

D Relative gradient optimization of the Jacobian term in unsupervised deep learning **133**

Bibliography **147**

Introduction

A brief history of Artificial Intelligence

The field of *artificial intelligence* (AI) has changed and evolved considerably over the last decades. The same term "artificial intelligence" has taken on various different meanings following the advancements of the epoch, being more often then not only loosely defined. We identify the genesis of the field with the first attempt to model biological neurons mathematically, proposed by McCulloch and Pitts [60] in the 1940s. Their approach took inspiration from the biological brain and it is framed in a set of efforts to build an artificial brain capable of "human" intelligence, fueled by novel advances and the emergence of modern scientific disciplines such as the theory of computation, neuroscience and cybernetics [80, 82] that have characterized the scientific landscape in those years. The first departure from this paradigm comes as early as in the 1950s, when a new approach is proposed to "proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it" ([59]). This paradigm shift is a direct consequence of the advent of the modern computer and it is a more abstract approach to AI, as it is based on the assumption that intelligence can be constructed and simulated on the basis of logic and formal systems. The first successful implementation of AI systems are due to this approach [64], which became known as Symbolic AI and represented the dominant paradigm for decades. However, in spite of a series of initial successes, the Symbolic AI approach struggled to cope with certain problems that are central to human cognition, such as perception, learning and pattern recognition. These fundamental problems appeared to be more easily approached by *artificial neural networks* [69], systems of computational units of the kind of the McCulloch and Pitts neuron, which sparked a renewed interest in this kind of models in the 1980s. It is at this point that interest in neural networks started to spread to other fields, especially with regard to the newly established science of *complexity* [79]. In this context, substantial contributions to the theoretical foundations and the development of neural networks came from the field of statistical physics. Among these, [42] introduced the first functioning neural network model for an associative memory, subsequently analyzed in exquisite detail in [5, 7, 6], and [30, 29] produced seminal work on the storage capacity of neural networks. Gradually, the original goal of realizing an artificial brain has

faded away, and a whole class of models employing artificial neural networks complemented with mathematical and statistical tools has emerged as a rich field in its own right. Referred to as *statistical machine learning*, its focus has become that of engineering systems capable of leveraging data and computations to solve complex problems. In the 2010s, the technological advancement has determined the *deep learning* revolution: the availability of more powerful hardware and network infrastructures for computation, as well as access to the massive amount of data constantly generated by myriads of interconnected technological devices, have made it possible to build increasingly larger and more powerful neural network models leveraging larger amounts of data to solve the most disparate problems with increasing success. The relation to AI is to be found in the fact that deep learning models have been employed to automatically solve problems that traditionally needed the employment of human intelligence to be solved, like object recognition [48], speech synthesis [67] and natural language translation [23].

Thesis organization

This thesis is structured in two main parts. In the first part, we adopt the statistical physics point of view on the analysis of neural networks. This approach is mainly based on the theory of *disordered systems* [61], a prototypical kind of complex system that is particularly suited to describe neural networks; finding its origin in the 1980s, this line of research is very active nowadays [17]. In this context, the investigation here presented concerns the Restricted Boltzmann Machine (RBM), a simple neural network that has been among the protagonists in the initial phase of the deep learning revolution [39]. This is particularly interesting from the statistical physics point of view as it can be interpreted as a generalization of the Hopfield model [9]. In Chapter 1 we introduce the Hopfield model and discuss the basic principles to derive a mean-field theory for neural networks. In Chapter 2 we introduce the RBM. We start discussing original work in Section 3, with the presentation of an empirical description of the dynamical properties of the RBM. This poses the basis for the theoretical mean-field analysis developed in Chapter 4, discussing equilibrium and dynamical properties. Finally, in Chapter 5 we show how to adapt the RBM to the challenging scenario of learning with missing information, taking advantage of the theoretical picture discussed in previous Sections.

In the second part, we turn our attention to Normalizing Flows (NF) [68], a more recent kind of models that finds applications in similar domains as the RBM. Here we depart from the statistical physics focus: in Section 6.2 we propose a NF model that employs fully connected neural networks and in Sections 6.3 and 6.4 we show how proceeding from simple Riemannian geometry considerations we can derive an efficient algorithm to train the proposed model.

Contributions

A reading guide Let's start with a stylistic note. Throughout the thesis, I use "we" as the subject. This is to signal that all the work discussed has been done with the help and guidance of my supervisors and collaborators. This is the only Section in which I use "I", and this is to point out in a precise way my personal contributions.

In Part I, Chapter 1 serves as an introduction to the tools and models that we use in the following Chapters. While the topics here discussed are well-known, the presentation is rather original. The whole treatment is meant to be self-contained, ultimately introducing the RBM in a principled way. Chapter 3 reports the empirical study of the RBM that I performed in the initial phase of my PhD. In the development of the mean-field theory of Chapter 4 I took on a more marginal role. The main theoretical analysis has been developed by my supervisors while I contributed in general discussions, in performing numerical analysis and writing the paper. This Chapter is supposed to be reasonably self-contained, and it represents my attempt to more gently introduce the related paper (Reprint B). For the work in Chapter 5 and Part II, I played a more central role both in the design of the proposed algorithms and their development. These Chapters are not self-contained; they represent a brief introduction to the associated papers (Reprints C and D), that I suggest reading in full.

Publications Parts I and II of this thesis are meant to introduce and complement the papers below, whose content is reported in full in the **Reprints** section.

- **Reprint A**
  A. Decelle, G. Fissore, and C. Furtlehner. "Spectral dynamics of learning in restricted Boltzmann machines". In: *EPL* 119.6 (2017), p. 60001

- **Reprint B**
  A. Decelle, G. Fissore, and C. Furtlehner. "Thermodynamics of Restricted Boltzmann Machines and Related Learning Dynamics". In: *Journal of Statistical Physics* 172.6 (July 2018), pp. 1576–1608. ISSN: 1572-9613. DOI: 10.1007/s10955-018-2105-y. URL: http://dx.doi.org/10.1007/s10955-018-2105-y

- **Reprint C**
  Giancarlo Fissore, Aurélien Decelle, Cyril Furtlehner, and Yufei Han. "Robust Multi-Output Learning with Highly Incomplete Data via Restricted Boltzmann Machines". In: *Proceedings of the 9th European Starting AI Researchers' Symposium 2020 co-located with 24th European Conference on Artificial Intelligence (ECAI 2020)* (2020)

- **Reprint D**
  Luigi Gresele, Giancarlo Fissore, Adrián Javaloy, Bernhard Schölkopf, and Aapo Hyvarinen. "Relative gradient optimization of the Jacobian

term in unsupervised deep learning". In: *Advances in Neural Information Processing Systems* 33 (2020)

Part I

RBM

Chapter 1

Background

1.1 The mathematical neuron

To investigate the mechanisms underlying intelligent behavior, we need a model of the brain that is simple enough to be studied yet rich enough to be functional. More precisely, we want to come up with models and algorithms that are able to realize complex tasks like pattern formation, object recognition and associative memories. A fundamental assumption in this context is that solutions to these complex tasks emerge from the interaction dynamics of a large number of simple elementary constituents, so we can abstract from the biological implementation details and build models that are useful and that we can analyze thoroughly.

The first successful attempt to model neurons mathematically has been proposed by McCulloch and Pitts [60]. Their model of the brain consists of a network of "on/off" binary nodes n_i with associated thresholds μ_i and connected by weights w_{ij} (connecting nodes i and j). The behavior of the network is determined by a simple time-dependent update rule:

$$n_i(t+1) = \Theta\left(\sum_j w_{ij} n_j(t) - \mu_i\right). \tag{1.1}$$

This update rule models the firing mechanisms observed in networks of real neurons. Here the nodes are modeled as individual processing units that activate when the inputs from the other units is above a certain threshold; the weights w_{ij} mimic the role of synaptic connections, with positive and negative values respectively modeling excitatory and inhibitory synapses, and the threshold values μ_i account for the need of an activation potential. Many biological details of real neurons are not directly included in this model, and we can assume that their role is not fundamental in determining the emergence of complex behavior in the network. Nonetheless, some features of the model are not biologically plausible, most notably the implicit need for a global synchronization mechanism (note that activations at time $t+1$ depend on the state of the full network at time t) which has not been observed in the animal brain.

We will see in Section 1.2 how in the Hopfield model we relax this constraint to obtain a functioning associative memory; here let us stress the fact that ultimately our objective is not to build a model of the brain which is as close to reality as possible but to study the abstract mechanisms that give rise to complex behaviors, and it is in this sense that the McCulloch-Pitts network of neurons is a success.

In the following Section we will introduce the Hopfield network, a simple generalization of the McCulloch-Pitts model that gives rise to an associative memory. Subsequently we will introduce the tools that let us analyze the behavior and the limits of such networks, and we will see how to apply them to the Hopfield model. Finally, we will introduce the main subject of interest of this thesis, the Restricted Boltzmann Machine (RBM).

1.2 The Hopfield model

One of the simplest tasks that we can realize with a neural network is that of implementing an associative memory, i.e. a memory system that is content-addressable: given a partial observation of a pattern as input, the memory retrieves a previously stored pattern that is similar to the input. A popular neural network system to implement an associative memory is the Hopfield model [42], consisting in a network of N binary nodes $s_i = \pm 1$ with associated thresholds θ_i and connected by weights w_{ij}.

Storage of patterns. A pattern is represented by a specific configuration of the nodes s_i and storage of patterns in the network is achieved through Hebbian learning [36], which originates from the observation that synaptic connections are stronger among neurons that activate simultaneously. The Hebbian rule is commonly stated as "neurons that fire together, wire together" and in the context of the Hopfield model with p binary patterns $\boldsymbol{x}^\mu$ to store it takes the form

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} x_i^\mu x_j^\mu, \quad i \neq j. \tag{1.2}$$

We note that the model can be described by a symmetric weight matrix $\boldsymbol{W}$; furthermore, connections of a node with itself are not allowed so that we have $w_{ii} = 0$.

Patterns retrieval. In contrast to a network of McCulloch-Pitts neurons, the Hopfield network is updated asynchronously: at each time step we select a node s_i at random and we update its value following the rule

$$s_i = \text{sign}\left(\sum_j w_{ij} s_j + \theta_i \right), \quad \text{sign}(x) = \begin{cases} -1 & x \leq 0 \\ +1 & x > 0 \end{cases} \tag{1.3}$$

We assume without loss of generality that $\theta_i = 0$. Further assuming that the number of nodes in the network is large enough, stored patterns are stable

under the application of the above update rule. To show this, let's call h_i^ν the input to a node i when the network is presented with a stored pattern $\boldsymbol{x}^\nu$

$$h_i^\nu = \sum_j w_{ij} x_j^\nu = \frac{1}{N} \sum_j \sum_\mu x_i^\mu x_j^\mu x_j^\nu.$$

If the number of stored patterns is small w.r.t. the number of nodes and recalling that the patterns are binary, we have

$$\begin{aligned} h_i^\nu &= x_i^\nu + \frac{1}{N} \sum_j \sum_{\mu \neq \nu} x_i^\mu x_j^\mu x_j^\nu \\ &\simeq x_i^\nu \end{aligned}$$

and thus the following stability condition holds

$$\text{sign}(h_i^\nu) = x_i^\nu. \tag{1.4}$$

This let us propose a possible strategy for retrieval of stored patterns. Starting from an initial configuration, we let the network evolve under the update rule (1.3); if we start from a configuration that is close to a stored pattern $\boldsymbol{x}^\nu$, we might expect the network to stabilize on the configuration corresponding to $\boldsymbol{x}^\nu$, thus realizing the objective of a content-addressable memory. We now go on by presenting a simple analysis of the convergence properties of the proposed strategy, to show that it is indeed a sensible alternative to realize an associative memory.

Convergence. A simple argument for the convergence of a Hopfield network to a stable state is provided in [13]. We start by representing the network as a bipartite graph; one partition includes the nodes whose value is $+1$ and the other partition includes the nodes whose value is -1. Each node will have "internal" connections linking it to the nodes in the same partition and "external" connections linking it to the nodes in the other partition. At each step we select a node and we compute the sum of internal and external connections; if the sum of the external connections is higher, we flip the node's value and we move it to the opposite partition. This process is equivalent to adopting the Hopfield update rule (1.3) but from this point of view the convergence of the network amounts to finding a minimum cut of the graph representing the network. Given that at every step the number of connections linking the two partitions either decreases or stays the same and that the number of possible states of the network is finite (so we would eventually explore all nodes with high probability) we are guaranteed to end up in a stable state.

Unfortunately, what is not guaranteed in the Hopfield model is that a stable state corresponds to a stored pattern. It is easy to see, for instance, that the reversed patterns $-\boldsymbol{x}^\nu$ are also stable. As another example of a stable state, we consider the special case in which the patterns are random and we take the mixed state $\boldsymbol{x}^m$ defined as

$$x_i^m = \text{sign}(x_i^{\mu_1} + x_i^{\mu_2} + x_i^{\mu_2}).$$

As the patterns are random, 3 times out of 4 we expect to observe $x_i^m = x_i^{\mu_1}$ (and the same is valid for $x_i^{\mu_2}$ and $x_i^{\mu_3}$), so that for N large the following identity holds

$$\sum_j x_j^{\mu_1} x_j^m = \frac{3}{4}N - \frac{N}{4} = \frac{N}{2}.$$

Assuming again that the number of patterns is small (w.r.t. the number of nodes) we get

$$\begin{aligned} h_i^m &= \sum_j w_{ij} x_j^m \\ &= \frac{1}{2}x_i^{\mu_1} + \frac{1}{2}x_i^{\mu_2} + \frac{1}{2}x_i^{\mu_3} + \frac{1}{N} \sum_{\mu_k \in (\mu_1,\mu_2,\mu_3)} \sum_j \sum_{\mu \neq \mu_k} x_i^\mu x_j^\mu x_i^m \\ &\simeq \frac{1}{2}x_i^{\mu_1} + \frac{1}{2}x_i^{\mu_2} + \frac{1}{2}x_i^{\mu_3} \end{aligned}$$

which is equivalent to the stability condition (1.4) and thus $\boldsymbol{x}^m$ is a stable configuration of the network even though it is not a stored pattern. In general, the argument can be extended to show that linear combinations of an odd number of patterns are stable, and these are called *mixture states*.

As a final remark, let us note that patterns stability has been assessed under the assumption that the number of patterns is small compared to the number of nodes in the network. In principle this is not very problematic as we can always assume the network to be large enough, but from a practical perspective it would be useful to know what is the capacity of the model, i.e. how many patterns we can store as a function of the size N of the network. We will deal with this matter and other details about the conditions to guarantee the functioning of the Hopfield model as an associative network in Section 1.7.

The energy function. To each configuration $\boldsymbol{s}$ of a Hopfield network we can associate a scalar function E defined as follows

$$E(\boldsymbol{s}) = -\frac{1}{2}\sum_{i=1}^N \sum_{j=1}^N w_{ij} s_i s_j + \sum_{i=1}^N \theta_i s_i. \tag{1.5}$$

This is called the *energy function* of the model, for reasons that will be clarified in Section 1.3. For now we are interested in showing that the dynamics induced by the update rule (1.3) minimizes this quantity. Starting from a configuration $\boldsymbol{s}$, at each step we select a node k and either we leave its value unchanged or we transition to a new configuration $\boldsymbol{s}'$ with $s'_k = -s_k$. In the first case the energy doesn't change, while in the second case we have

$$\begin{aligned} \Delta E &= E(\boldsymbol{s}') - E(\boldsymbol{s}) \\ &= (s_k - s'_k)\left(\sum_j w_{kj} s_j - \theta_k\right) \end{aligned}$$

which is necessarily negative to be consistent with the update rule (1.3). Similarly, when the network attains a stable configuration we have $\Delta E > 0$ if we flip the value of any node.

We can now paint an interesting picture of the Hopfield model, that will be the starting point for the more advanced analysis presented later on. The configurations of the network live in an energy landscape with many minima, represented by stable configurations of the system. The network evolves under rule (1.3) to minimize the energy of the system, attaining the stable configurations. As the energy decreases at each step of the dynamical evolution, we can think of the stable states as *attractors* of the system; initializing the system close to a stable configuration, the network will be attracted to it. In this sense it seems reasonable that under the right conditions the system can function as an associative memory; Section 1.7 is devoted to delineating these conditions.

1.3 Statistical description of a neural network

We turn to a stochastic description of a network of computational neurons, which amounts to introduce some uncertainty in the model by associating probabilities to the possible states of the system. This let us reason directly on the expected probabilistic behavior of the model and not on specific instances of it, allowing us to reach general conclusions. Moreover, by analogy to neuronal networks it seems reasonable to introduce some noise as a way to take into account minor effects due to the complexities of the actual biological systems.

As a neural network of choice, we take the Hopfield model introduced in Section 1.2 and define a probability distribution $P(\boldsymbol{s})$ over the nodes of the network. On this basis, we derive the exact form of this probability distribution and we introduce the necessary tools to analyze the expected behavior of the system in great details. We remark that the Hopfield model as described in Section 1.2 can already be considered weakly stochastic; indeed, its dynamics depend on the random choice of nodes, but the update rule (1.3) is deterministic.

Defining a probability distribution over the nodes $\boldsymbol{s}$ we allow the equilibrium configurations to fluctuate around their expected value. In this context the description of the model in terms of the energy function is useful, as allowing the configurations to fluctuate is equivalent to allowing the energy to fluctuate. But what is a good functional form for the probability distribution over the nodes? Our strategy is to look for a distribution that is as general as possible, i.e. we want to avoid making unnecessary assumptions. This can be formulated as an instance of the Occam's razor, a principle stating that the most economical theories are to be preferred. The logical and quantitative reasons motivating the success of this principle lie in the fact that from a Bayesian standpoint the most economical explanation consistent with the observations at hand is more probable ([55], Chapter 28). Intuitively, in our case the only observation that the appropriate probability distribution should encode is that the average value of the energy is fixed and constant; as for the rest, avoiding assumptions

means that we are left in a state of maximum uncertainty about the fluctuations of the system around the average. The information-theoretic expression that quantifies uncertainty is the entropy of the distribution [71, 18]

$$H[P] = -\sum_{\boldsymbol{s}} P(\boldsymbol{s}) \log(P(\boldsymbol{s})).$$

A sensible strategy to derive the functional form of the probability $P(\boldsymbol{s})$ is thus to apply the principle of maximum entropy [45]: we look for the normalized distribution whose entropy is maximal under the only constraint of constant average energy

$$\begin{aligned} &\text{maximize} \quad H[P] \\ &\textit{subject to} \quad \sum_{\boldsymbol{s}} P(\boldsymbol{s}) = 1, \sum_{\boldsymbol{s}} P(\boldsymbol{s}) E(\boldsymbol{s}) = \langle E \rangle. \end{aligned}$$

We can solve this constrained optimization problem by using Lagrange multipliers. Introducing the multipliers λ and β we can write the Lagrangian

$$L = H(\boldsymbol{s}) - \lambda \left(\sum_{\boldsymbol{s}} P(\boldsymbol{s}) - 1 \right) - \beta \left(\sum_{\boldsymbol{s}} P(\boldsymbol{s}) E(\boldsymbol{s}) - \langle E \rangle \right)$$

that we extremize (eliminating λ) to get

$$P(\boldsymbol{s}) = \frac{e^{-\beta E(\boldsymbol{s})}}{Z}, \quad Z = \sum_{\boldsymbol{s}} e^{-\beta E(\boldsymbol{s})}. \tag{1.6}$$

Here β is a parameter that regulates the noise. To see this, we discuss the stochastic dynamics of the network. An equilibrium configuration $\boldsymbol{s}$ of the network is determined by a certain energy $E(\boldsymbol{s})$ with associated probability $P(\boldsymbol{s})$. Being the system at equilibrium, the total probability of the network to transition to a different state $\boldsymbol{s}'$ must be counterbalanced by the probabilities of transitioning to state $\boldsymbol{s}$ when found in a state $\boldsymbol{s}' \neq \boldsymbol{s}$. Calling $T_{\boldsymbol{s} \to \boldsymbol{s}'}$ the transition probability from state $\boldsymbol{s}$ to state $\boldsymbol{s}'$, a sufficient condition to guarantee equilibrium is given by the *detailed balance* condition

$$T_{\boldsymbol{s} \to \boldsymbol{s}'} P(\boldsymbol{s}) = T_{\boldsymbol{s}' \to \boldsymbol{s}} P(\boldsymbol{s}')$$

which we can write as

$$\frac{T_{\boldsymbol{s} \to \boldsymbol{s}'}}{T_{\boldsymbol{s}' \to \boldsymbol{s}}} = \frac{P(\boldsymbol{s}')}{P(\boldsymbol{s})} = e^{-\beta(E(\boldsymbol{s}) - E(\boldsymbol{s}'))} = e^{-\beta \Delta E}. \tag{1.7}$$

A stochastic update rule for the Hopfield network must then obey condition (1.7). A particularly fitting choice is the update rule defined by Glauber dynamics [32]

$$s_i = \begin{cases} +1 & \text{with probability } f_\beta(h_i) \\ -1 & \text{with probability } 1 - f_\beta(h_i) \end{cases} \tag{1.8}$$

where

$$f_\beta(h_i) = \frac{1}{1 + e^{-2\beta h_i}}$$
$$h_i = \sum_j w_{ij} s_j + \theta_i.$$

Given rule (1.8) we see that the evolution of the system is completely random for $\beta \to 0$ (maximum noise) while it falls back to the deterministic rule (1.3) for $\beta \to +\infty$ (no noise).

Equation (1.6) defines the Boltzmann-Gibbs distribution, with Z being called the partition function. Such a distribution is used to model a wide variety of physical systems at thermal equilibrium, where the energy function effectively represents the energy of the system. In this context, the temperature is a proxy for the noise and it is related to β as

$$\beta = \frac{1}{T}.$$

We will refer to β as the *inverse temperature* and to T as the *temperature* of our model. While it might seem strange to define a temperature for a neural network, we note that it is in truth pretty natural. In physical systems the temperature is generally interpreted as a quantity that specifies the level of noise, abstracting from the actual microscopic mechanisms generating the noise; here, we introduced the inverse temperature in the exact same way.

1.4 Free energy

We have seen in Section 1.2 that in the deterministic case the energy of the Hopfield network is minimized; the model evolves towards the states of minimum energy and such states are stable. In the stochastic case, due to the noise, the system is allowed to move away from the states of lowest energy. The actual value of the energy fluctuates, while its average value is constant. The quantity that is now extremized is the entropy, which is required to be maximal. Rewriting the entropy to explicitate the average energy

$$H[P] = -\sum_{\boldsymbol{s}} P(\boldsymbol{s}) \log P(\boldsymbol{s}) = \beta \langle E \rangle + \log Z$$

we see that the quantity that gets minimized is now $\log Z$, which is defined to be the *free energy*

$$F \stackrel{\text{def}}{=} -T \log Z = \langle E(\boldsymbol{s}) \rangle - TH[P]. \tag{1.9}$$

This let us refine our picture of the Hopfield model. The system still evolves towards states of minimal energy, but this tendency is counteracted by the entropic term that allows other energy configurations at equilibrium. The entropic term is weighted by the temperature: when $T \to 0$ we get back to the

deterministic case and the energy of the system is minimized, while for $T \to +\infty$ the entropic term dominates the dynamics and all energy configurations are allowed.

The free energy fully determines the behavior of the system. From its expression we can derive a variety of quantities of interest to describe the system, like the average values of the nodes and their correlations

$$\langle s_i \rangle = -\frac{\partial F}{\partial \theta_i}$$
$$\langle s_i s_j \rangle = -\frac{\partial F}{\partial w_{ij}}.$$

A tractable form of the free energy is thus a very powerful tool to analyze the system in depth. Unfortunately, this is not easy to compute as the partition function Z involves a sum over an exponential number of terms. A standard procedure is to express the free energy in a more general form and compute its value in an approximated way, as we'll see in Section 1.6.

1.5 Order parameters and phase transitions

The Hopfield network can operate in different regimes, depending on the aggregated characteristic of the actual configurations of the nodes. These regimes are called phases, and the terminology for the regimes comes from the theory of the Ising model. The Ising model is an abstract model of a ferromagnet, in which the nodes are called spins and the spins model the magnetic moments of atoms in magnetic material. The specificity of the Ising model is that the interactions among spins (the weights, in the Hopfield model) are all positive and constant. At high temperature the entropic term dominates and the spins are randomly oriented; the average value of the spins is thus zero, and this phase is called paramagnetic in analogy to the paramagnetic material. When the temperature is low, the interaction among the spins drive the system to the minimum of the energy, which is identified by the configuration in which all the spins are aligned. The system is thus found in a different phase in which the spins are ordered, called the ferromagnetic phase; the magnetization is now different from 0 and the transition from a null value to a finite value signals the transition to a different phase of the system. The magnetization is called *order* parameter as it signals the phase transition. For the Hopfield model, at high temperature we can observe the same behavior and we have again a paramagnetic phase. The behavior in the ferromagnetic phase is more subtle; given that the weights can have both positive and negative values, the system can be found in situations in which there are no ways to coordinate all the nodes to attain orientations that minimize the energy. These are frustrated configurations that evidence how the magnetization is not a good order parameter to describe the Hopfield model, and more refined alternatives need to be used. Edwards-Anderson order parameters are appropriate to describe the separation between paramagnetic and spin glass phase.

1.6 Effective free energy

In this section we approach the problem of computing a good approximation to the free energy by exploiting the saddle point method, a general methodology that is applicable to a wide variety of models. We recall that the exact form of the free energy (1.9) contains an exponential number of terms, making the computation of its value a computationally hard problem. Adopting a statistical description of the Hopfield model, we know from Section 1.4 that the system evolves towards equilibrium states that minimize the free energy. Furthermore, the equilibrium configurations are found in specific phases of the system, characterized by the appropriate order parameters. Exploiting these observations, the strategy consists in defining an *effective free energy* as a function of the order parameters. By construction, the minimum of the effective free energy will be equivalent to the true free energy of the system, reducing the problem of computing the free energy to a minimization problem. Further assuming that the system is "large enough" we can compute the minimization with the saddle point approximation. To exemplify this methodology, we will consider the Ising ferromagnet and the associated magnetization. This choice is made to simplify the presentation of the method, as the Hopfield model necessitates of more elaborate order parameters (Section 1.5) and some more involved mathematical manipulations. The general approach is nonetheless the same; details about the Hopfield model are discussed in Section 1.7 and a more elaborate method is presented in Section 4.2 with the introduction of the Replica Method.

For the Ising ferromagnet, the magnetization m is the only order parameter that we need to characterize the system

$$m = \frac{1}{N} \sum_i \langle s_i \rangle. \tag{1.10}$$

At equilibrium, the system will have a specific magnetization m_{eq} for which the free energy is defined. To extend the notion of free energy, we want to consider all possible values of magnetization. We start by rewriting the partition function in terms of the free energy

$$Z = \sum_{\boldsymbol{s}} e^{-\beta E(\boldsymbol{s})} = e^{-\beta F}.$$

Now we can fix the value of the magnetization and consider all configurations $\boldsymbol{s}_m$ of the system that are consistent with such value, i.e. for which we have $\frac{1}{N}\sum_i s_i = m$. Summing over all possible values of the magnetization, we can rewrite the partition function as

$$Z = \sum_m \sum_{\boldsymbol{s}_m} e^{-\beta E(\boldsymbol{s}_m)}$$

from which a natural generalization of the free energy for a specific magneti-

zation is defined

$$F(m) \stackrel{\text{def}}{=} -T \log \left(\sum_{S_m} e^{-\beta E(S_m)} \right), \qquad Z = \sum_m e^{-\beta F(m)}. \tag{1.11}$$

We call this magnetization-dependent quantity the *effective free energy* and we will also be interested in its per-node value

$$f(m) = \frac{F(m)}{N}.$$

For a large system ($N \to +\infty$) the magnetization can take an almost-continuous set of values (flipping one node determines a change of $\frac{2}{N}$ in the magnetization) so we can rewrite the partition function as an integral

$$Z = \frac{N}{2} \int_{-1}^{+1} dm \; e^{-\beta N f(m)}.$$

As it turns out, for large N the value of the above integral is dominated by the minimum value of f. We can thus substitute the integral to obtain

$$Z \simeq e^{-\beta N f(m_{\min})}.$$

This is called the *saddle point* approximation and it lets us link the effective free energy to the equilibrium free energy

$$F \simeq F(m_{\min}), \quad m_{\min} \equiv m_{\text{eq}}.$$

The problem of computing the partition function can be solved by deriving a tractable form for $f(m)$ and determining m_{eq}.

1.7 Mean-field theory of the Hopfield model

A complete mean-field theory of the Hopfield model has been derived in [5, 7] under the assumption that the variables describing the patterns are independent. In this section we summarize the derivation and clarify the role of the patterns in defining a good order parameter for the Hopfield model. To begin with, we will follow the strategy presented in Section 1.6 and derive a tractable effective free energy. Subsequently, we discuss the need to average over the patterns to describe a general statistical behavior of the system. We will only present the results of this averaging; a more detailed account can be found in [37, 66].

Effective free energy. To better understand why the partition function is hard to compute, we rewrite the energy function of the Hopfield model as

$$E(\boldsymbol{s}) = -\frac{1}{2N} \sum_{\mu=1}^{p} \left(\sum_i s_i \xi_i^\mu \right)^2 + \text{const.}$$

where we made explicit the dependence of the weights w_{ij} on the patterns and we neglected the thresholds θ_i. The constant term comes from the diagonal weights w_{ii} and we can drop it as it doesn't impact the energy minimization. We observe that the energy function can be interpreted as a sum of contributions from each stored pattern, where each contribution is measured by the square of the overlap of the system configuration with the considered pattern. The partition function takes the form

$$Z = \sum_S e^{\beta E(S)} = \sum_S \exp\left(-\frac{\beta}{2N}\sum_{\mu=1}^{p}\left(\sum_i s_i \xi_i^\mu\right)^2\right)$$

which is hard to compute because the sum in the exponential includes quadratic terms that cannot be factored. A simple yet powerful strategy to deal with this problem is to introduce a Gaussian integral to verify the identity

$$\exp\left[-\frac{\beta}{2N}\left(\sum_i s_i \xi_i^\mu\right)^2\right] = \left(\sqrt{\frac{\beta N}{2\pi}}\right)^p \int_{-\infty}^{+\infty} dm^\mu \times \exp\left[-\frac{\beta N}{2}\left(m^\mu\right)^2 + \left(\sum_i s_i \xi_i^\mu\right) m^\mu\right]$$

where we have introduced an auxiliary variable m^μ for each pattern. This is a Hubbard-Stratonovich transformation, useful to introduce the order parameters $\boldsymbol{m} = \{m^\mu, \mu = 1, \dots, p\}$. We will give the interpretation for these order parameters in the next paragraph; for now we proceed to rewrite the partition function by exploiting the above identity

$$Z = \sum_{\boldsymbol{s}} \left(\sqrt{\frac{\beta N}{2\pi}}\right)^p \int_{-\infty}^{+\infty} \prod_\mu dm^\mu \exp\left[-\frac{\beta N}{2}\left(m^\mu\right)^2 + \beta\left(\sum_i s_i \xi_i^\mu\right) m^\mu\right]. \tag{1.12}$$

The overlap terms in the exponential are now linear, so we can introduce a tractable effective free energy $f(\beta, \boldsymbol{m})$

$$Z = \left(\sqrt{\frac{\beta N}{2\pi}}\right)^p \int_{-\infty}^{+\infty} d\boldsymbol{m}\, e^{-\beta N f(\beta, \boldsymbol{m})}$$

$$f(\beta, \boldsymbol{m}) = \frac{1}{2}\boldsymbol{m}^2 - \frac{1}{\beta N}\sum_i \log\left[2\cosh\left(\beta \sum_\mu m^\mu \xi_i^\mu\right)\right]. \tag{1.13}$$

Employing the saddle point approximation we calculate the equilibrium free energy

$$F = N f(\beta, \boldsymbol{m}_{\text{eq}})$$

and we can derive an implicit expression for m^μ_{eq}

$$\left.\frac{\partial f}{\partial m^\mu}\right|_{m^\mu_{\text{eq}}} = 0 \quad \Longrightarrow \quad m^\mu_{\text{eq}} = \frac{1}{N}\sum_i \xi_i^\mu \tanh\left(\beta \sum_\nu m^\nu_{\text{eq}}\, \xi_i^\nu\right).$$

Order parameters. In the previous paragraph we have omitted the thresholds θ_i to simplify the notation. Reintroducing them in a Hebbian fashion, we get a set of external fields $\theta_i = \sum_\mu h^\mu \xi_i^\mu$ correlated to the patterns, where h^μ is a scalar value determining the strength of the threshold. The partition function and the effective free energy (1.13) become

$$\log Z = \log \sum_S \exp\left(\frac{\beta}{2N}\sum_{\mu=1}^{p}\left(\sum_i s_i \xi_i^\mu\right)^2 + h_\mu \sum_i \xi_i^\mu s_i\right)$$
$$f(\beta, \boldsymbol{m}) = \frac{1}{2}\boldsymbol{m}^2 - \frac{1}{\beta N}\sum_i \log\left[2\cosh\left(\beta \sum_\mu \left(m^\mu + h^\mu\right)\xi_i^\mu\right)\right].$$

Deriving both expressions by the thresholds' strenghts h^μ we get

$$\frac{\partial F}{\partial h^\mu} = -\frac{1}{\beta}\frac{\partial \log Z}{\partial h^\mu} = -\sum_i \langle s_i \rangle \xi_i^\mu$$
$$\frac{\partial F}{\partial h^\mu} = N\frac{\partial f}{\partial h^\mu} = -\sum_i \xi_i^\mu \tanh\left(\beta \sum_\nu \left(m^\nu + h^\nu\right)\xi_i^\nu\right)$$

and similarly to the previous paragraph we can get the equilibrium value of the magnetization

$$m_{\text{eq}}^\mu = \frac{1}{N}\sum_i \xi_i^\mu \tanh\left(\beta \sum_\nu \left(m_{\text{eq}}^\nu + h^\nu\right)\xi_i^\nu\right).$$

Putting together all of the above equations we finally get the equivalence

$$m^\mu = \frac{1}{N}\sum_i \langle s_i \rangle \xi_i^\mu \tag{1.14}$$

where we see that m^μ represents the overlap between the network's configuration and pattern $\boldsymbol{\xi}^\mu$. Its value depends on the correlation of the nodes s_i to the pattern: we have $m^\mu = 1$ when the configuration of the network reproduces pattern $\boldsymbol{\xi}^\mu$ exactly and $m^\mu = 0$ when there is no correlation. The order parameters m^μ can thus be thought of as the magnetizations of the network correlated to the μth pattern, providing a concrete physical interpretation.

Patterns averaging and phase diagram. The expression that we have derived for the effective free energy depends explicitly on the specific patterns to store. To describe our model in full generality, we want to get rid of the patterns by averaging out their effect on the free energy. This averaging process requires fixing the statistical properties of the patterns that we can store, to come up with a description of a statistical ensemble rather than a specific model instance. The way this ensemble is chosen is fundamental to describe functional systems, and the assumptions made in this context define the range

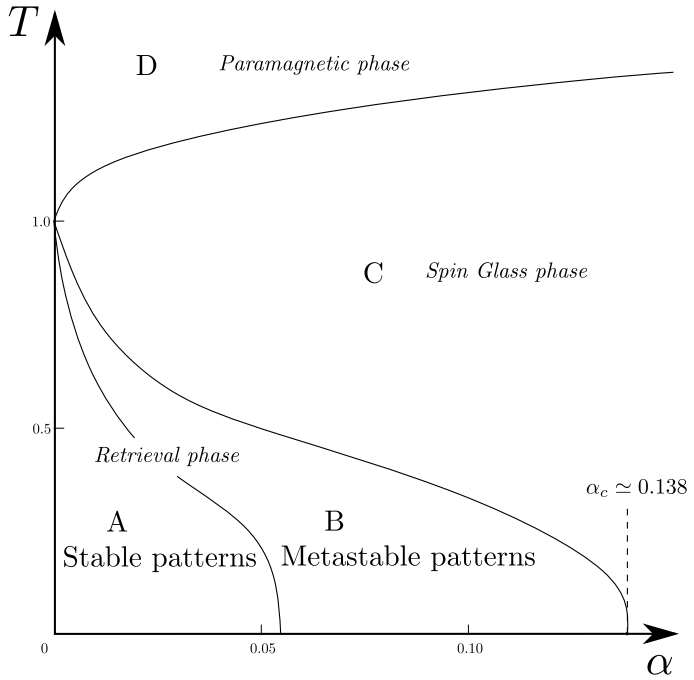


Figure 1.1: Phase diagram of the Hopfield model.

of systems that are well described by the mean-field picture. We stress this point here as it is fundamental in our treatment of a realistic model of RBM. For the Hopfield model, the assumption is that the variables representing the patterns are i.i.d. with $\xi_i^\mu = \pm 1$ at random. Here, the difficulty lies in the fact that it is not clear how to perform the average over the patterns in Equation (1.13). In Section 4.2 we will introduce the Replica Method to address this problem; for now, we skip the details on the averaging to list the results for the Hopfield model.

In Section 1.2 we have seen how the stored patterns are stable states of the network. We have also seen how certain kinds of mixture states are stable, and we couldn't roll out the possibility of stable *spurious states*, i.e. states that are not correlated to the patterns. Moreover, we worked with the assumption that the number of stored patterns is small and we didn't provide any indications about the capacity of the network. By analyzing the stability of the minima of the averaged mean-field effective free energy, we can address all of the above problems. A detailed discussion is provided in [37], and the results are summarized in the phase diagram reported in Figure 1.1. We define the capacity α of the network as the number of patterns per unit node: $\alpha = \frac{p}{N}$. The patterns are stable only in region A and B of the phase diagram; in both regions we also find stable mixture and spurious states, with the mixture states always presenting higher free energy than the stored patterns and thus the mixture

states are local minima in regions A and B. In region A the patterns represent global minima and the spurious states are local minima, while in region B the situation is reversed. Patterns are thus stable in the A region and metastable in the B region. We call the AB region the *retrieval phase*. In region C only the spurious states are stable, while in region D the noise dominates and only states with $m = 0$ are possible. We note that there is a critical value α_c for the capacity above which the patterns are not stable, and that determines a bound on the capacity of a functional Hopfield network.

The picture that emerges from the mean-field analysis is rather complete and detailed. We pass from a not very well-defined energy landscape to a more refined free energy landscape in which the functioning of the network is described in detail. Finally, for $T \to 0$ we recover the behavior of the deterministic Hopfield model.

As a final note, let us remind that all of the above results are obtained for random patterns. To describe a Hopfield model able to memorize real-world data, we need more realistic assumptions. We will deal with this fundamental problem in the chapters about the RBM.

1.8 Beyond memory: Boltzmann Machines

In the previous section we have seen how a stochastic description of the Hopfield network is useful to analyze the model in great detail. Nonetheless, we have seen that the deterministic rule (1.3) is sufficient to robustly retrieve stored patterns, successfully realizing an associative memory. In this section we will show how the introduction of stochastic units is not only useful for the purpose of analyzing the model, but it can serve to approach the task of pattern formation and data generation. We will then discuss the need to introduce hidden units to obtain more expressive networks, and define the Boltzmann machine model in general terms.

Stochastic units. Let's consider binary nodes $v_i = 0, 1$ as opposed to $s_i = \pm 1$; we will keep this notation consistent throughout the chapters. The units are stochastic, such that similarly to rule (1.8) we have

$$v_i = \begin{cases} 1 & \text{with probability } p(\boldsymbol{x}) \\ 0 & \text{with probability } 1 - p(\boldsymbol{x}) \end{cases}$$

with

$$\begin{aligned} p(\boldsymbol{x}) &= \frac{1}{1 + \exp\left[-\beta \left(\sum_j w_{ij} x_j + \theta_i\right)\right]} \\ &= \text{sigm}\left(\beta \sum_j w_{ij} x_j + \theta_i\right) \end{aligned}$$

where $\boldsymbol{x}$ represents the data (i.e. the patterns) and the sigmoid function $\text{sigm}(x) = 1/(1+\exp(-x))$ appears naturally due to the use of $v_i = 0, 1$ binary units. Similarly to (1.6) the probability over the nodes is given by

$$P(\boldsymbol{v}) = \frac{e^{-\beta E(\boldsymbol{v})}}{Z}, \quad Z = \sum_{\boldsymbol{v}} e^{-\beta E(\boldsymbol{v})}. \tag{1.15}$$

The above expression defines a parametrized probability distribution over the data, with parameters $\boldsymbol{W}, \boldsymbol{\theta}$. Assuming that the dataset is described by a ground truth distribution, our goal is to determine the set of parameters that best approximates such a distribution. Adding the requirement that we must be able to efficiently sample from the learned distribution, we obtain a *generative model* of the data.

Hidden units. The probabilistic model that we introduced above, in the current formulation, presents a major limitation. The energy function only contains first- and second-order terms in the nodes, meaning that the network can enforce only up to pairwise correlations; higher-order correlations are not contemplated. As an example, consider the configurations of even parity for a network with 3 nodes: (000), (011), (101), (110). There doesn't exist a set of values for the weights of the network such that all of those configurations can be represented by equilibrium states of the network. While in this simple case the problem could be solved by adding an extra node with the appropriate value for each configuration, in practice the extra information is not present in the dataset to model. We need to add an extra set of *hidden nodes* to the network, which are not used to represent the data but to introduce higher-order dependencies among the original nodes. The original nodes are called *visible* by virtue of the fact that they are used to directly represent the data.

The kind of stochastic network with visible and hidden nodes that we described above is known as *Boltzmann machine*; its energy function includes three different weight matrices $\boldsymbol{L}, \boldsymbol{J}, \boldsymbol{W}$ to model visible-to-visible, hidden-to-hidden and visible-to-hidden correlations. Denoting by v_i and h_j respectively the visible and hidden nodes, the energy function reads

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\frac{1}{2}\sum_{i,j} v_i l_{ij} v_j - \frac{1}{2}\sum_{ij} h_i j_{ij} h_j - \sum_{ij} v_i w_{ij} h_j - \sum_i \theta_i v_i - \sum_j \eta_j h_j. \tag{1.16}$$

One final problem that we need to address to use the Boltzmann Machine to model the empirical distribution of a dataset is to define a procedure to learn the parameters of the model. A training algorithm for the Boltzmann Machine has been proposed in [1]; in next chapters, we will see how dropping some parameters let us define the Restricted Boltzmann Machine, for which a more efficient training algorithm is readily obtained.

1.9 Statistical Physics approach to neural networks

This Chapter serves as an example of how statistical physics can be useful to study neural network models in great detail. Scientific work adopting this approach has been flourishing in the last decade [84, 17], tackling the challenge of studying the theoretical foundations of deep learning. The work on the RBM that is the object of next Chapters finds its spot in this context. In Section 1.8 we have discussed the similarity between the RBM and the Hopfield model; a detailed analysis of this connection is found in [9, 8, 62]. What makes the RBM a reference model from the statistical physics point of view is its similarity to the Sherrington-Kirkpatrick (SK) model of a spin glass, that has been studied in detail in [72, 74, 3]. Indeed, the RBM can be seen as a bipartite variant of the SK model, and its theoretical analysis is based on this observation.

Chapter 2

Restricted Boltzmann Machines (RBM)

2.1 Definition

In section 1.8 we have seen how Boltzmann Machines (BM) provide a model to go beyond an associative memory (Hopfield model) to build a generative model of the data. While this is a success from the theoretical point of view, BMs are not very practical as their training is very expensive for high dimensional data. To improve on this situation, the RBM forbids connections among visible nodes and among hidden nodes, making it possible to efficiently train the model.

The actual RBM model then consists in a bipartite graph with a layer of hidden units h_j and a layer of visible units v_i. The units in one layer are not connected among them but are connected to all the units in the other layer.We restrict our treatment to the case of binary units $h_i, v_i = 0, 1$. The energy function of the BM is readily adapted to the bipartite case

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i,j} v_i w_{ij} h_j - \sum_i \theta_i v_i - \sum_j \eta_j h_j \tag{2.1}$$

where θ_i and η_j are *external fields*, or *biases*, acting respectively on the visible and hidden units. The probability of a configuration (1.15) becomes

$$P(\boldsymbol{v}, \boldsymbol{h}) = \frac{e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}}{Z}, \quad Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}. \tag{2.2}$$

To simplify the equations, we set $\beta = 1$ without loss of generality. We recall that in Section 1.3 we discussed how β is a parameter that regulates the noise in the network; for the RBM, a proxy for β is given by the variance of the weights $\boldsymbol{W}$, as we'll see in Section 2.3.

The crucial simplification in the RBM lies in the fact that units in the same layers are not connected: they are thus conditionally independent given the nodes in the other layer, and we can compute the individual probabilities in

closed form

$$P(v_i = 1|\boldsymbol{h}) = \frac{1}{1 + e^{-\theta_i - \sum_j w_{ij} h_j}} = sigm\left(\theta_i + \sum_j w_{ij} h_j\right) \tag{2.3}$$

$$P(h_j = 1|\boldsymbol{v}) = \frac{1}{1 + e^{-\eta_j - \sum_i w_{ij} v_i}} = sigm\left(\eta_j + \sum_i w_{ij} v_i\right). \tag{2.4}$$

What we are interested in is the probability for the visible units, which is the layer we use to represent data. It can be easily defined as

$$P(\boldsymbol{v}) = \sum_{\boldsymbol{h}} P(\boldsymbol{v}, \boldsymbol{h}) = \frac{e^{-F_c(\boldsymbol{v})}}{Z}, \quad Z = \sum_{\boldsymbol{v}} e^{-F_c(\boldsymbol{v})} \tag{2.5}$$

where we have introduced the *clamped free energy*

$$F_c(\boldsymbol{v}) = -\log \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{h}, \boldsymbol{v})} = -\sum_i a_i v_i - \sum_j \log\left(1 + e^{\left(b_j + \sum_i w_{ij} v_i\right)}\right). \tag{2.6}$$

To use the RBM as a generative model, we want to maximize $P(\boldsymbol{v})$ for the samples belonging to the training set. This is done by performing gradient ascent over the log-likelihood $\log P(\boldsymbol{v})$, whose derivative with respect to the weights can be computed to be

$$\frac{\partial \log P(\boldsymbol{v})}{\partial w_{ij}} = \langle v_i P(h_j = 1|\boldsymbol{v})\rangle_{data} - \langle v_i h_j\rangle_{model} \tag{2.7}$$

where $\langle\cdot\rangle_{data}$ denotes an average over the empirical distribution of the training set ($P_{data}(\boldsymbol{v}) = \frac{1}{N}\sum_{\boldsymbol{v}_n \in data} \delta(\boldsymbol{v} - \boldsymbol{v}_n)$) and $\langle\cdot\rangle_{model}$ denotes the average over the distribution (2.5). Introducing the *learning rate* α (as a parameter for gradient ascent) we obtain an update rule for the weight matrix

$$\Delta \boldsymbol{W} = \alpha\left(\langle \boldsymbol{v}\boldsymbol{h}^T\rangle_{data} - \langle \boldsymbol{v}\boldsymbol{h}^T\rangle_{model}\right). \tag{2.8}$$

In the same way we can get the update rules for the external fields

$$\Delta \boldsymbol{\theta} = \alpha\left(\boldsymbol{v}_{data} - \boldsymbol{v}_{model}\right) \tag{2.9}$$

$$\Delta\boldsymbol{\eta} = \alpha\left(\boldsymbol{h}_{data} - \boldsymbol{h}_{model}\right). \tag{2.10}$$

Given the update rules, the training consists in actually performing the gradient ascent. Once this is done, it is possible to sample the equilibrium configurations of the RBM to obtain samples which are generated according to the approximated probability distribution of the training data. Unfortunately, the average over the model distribution $\langle\cdot\rangle_{model}$ is intractable as such term is exponential in the number of visible units. Approximations are then necessary to train and sample from an RBM; in particular, Monte Carlo based algorithms are generally employed, such as *k-steps contrastive divergence* (CDk) [38] and *persistent contrastive divergence* (PCD) [76]. Approximate algorithms based on mean-field methods have also been employed, originally in [81] and more recently in [28].

2.2 Contrastive divergence training

The standard strategy to compute an approximation of the gradient (2.8) is to use a Montecarlo sampling procedure. The first term $\langle\boldsymbol{v}\boldsymbol{h}^T\rangle_{data}$, also called the *positive term*, is easily computed: here $\boldsymbol{v}$ is represented by the actual data while $\boldsymbol{h}$ is computed with (2.4). The term $\langle\boldsymbol{v}\boldsymbol{h}^T\rangle_{model}$, also called the *negative term*, is more problematic: to average over the actual parameters of the model, we can initialize $\boldsymbol{v}$ with the data and perform Gibbs sampling (Algorithm 1) for a high number of steps k, i.e. until convergence is reached.

Algorithm 1 Gibbs sampling

1: **Init:** take a random configuration $\boldsymbol{v}$
2: **for** $i = 0$ to k **do**
3: $\quad\boldsymbol{h} \sim P(\boldsymbol{h}|\boldsymbol{v})$
4: $\quad\boldsymbol{v} \sim P(\boldsymbol{v}|\boldsymbol{h})$
5: **end for**
6: Set $v_i = 1$ with probability $P(v_i = 1|\boldsymbol{h})$

This would give a good unbiased estimate of the negative term, at the cost of a lengthy iterative procedure. In practice, instead of running the Montecarlo procedure until equilibrium is reached, the negative term is usually estimated by performing only a small number of Gibbs steps k, and this training procedure is called k-steps Contrastive Divergence (CDk) [38]. CDk uses a rather crude approximation to the gradient; a slightly refined training strategy consists in using the data to initialize the Gibbs sampling procedure only at the first gradient update, and keep iterating over the same chain for subsequent updates. The rationale is that every gradient update perturbs the weight matrix only slightly, meaning that after many minibatches and epochs of training we can expect the sampled configurations to better approximate the equilibrium state of the model. The sampled configurations are called the *persistent chain* and the procedure k-steps Persistent Contrastive Divergence (PCDk) [76]. Both

CDk and PCDk have been analyzed in detail in [22], which shows that the RBM can operate in two different regimes depending on the interplay between the number of steps k and the mixing time of the Gibbs sampling procedure. For small values of k (relative to the mixing time) the RBM operates in an out-of-equilibrium regime in which it is possible to produce good qualitative samples but whose data generation abilities don't properly match the training dataset. To obtain a properly equilibrated model, high values of k are necessary and in this case the RBM is able to produce good qualitative samples that also respect the expected statistics of the training set. In Algorithm 2 we detail the CDk training strategy in vectorized (matrix) form. Here, we choose to randomly initialize the negative term's sampling chain as proposed in [21] while traditionally the negative chain is initialized with the dataset samples as for the positive term [40, 26]. Moreover, we note that in principle the number of samples in the negative chain can be freely chosen and we expect to better approximate the negative term by using a high number of samples. However, the number of samples in the negative chain is generally chosen to be equal to the batch size, which can be motivated by the need to keep the statistical errors in between positive and negative terms comparable.

Algorithm 2 k-steps Contrastive Divergence (CDk)

Data: a training set of N data vectors $\boldsymbol{v}_i$
Randomly initialize the weight matrix $\mathbf{W}$
for t = 0 to T (# of epochs) **do**
    Divide the training set in m minibatches of n data vectors $\boldsymbol{v}_i$
    for all minibatches m **do**
     Positive term:
        $\boldsymbol{V}_m = [\boldsymbol{v}_1\ \boldsymbol{v}_2\ \dots\ \boldsymbol{v}_n]$
        sample $\boldsymbol{H}_p \sim P(\boldsymbol{H}_p \mid \boldsymbol{V}_m)$ (Equation (2.4))
        sample $\boldsymbol{V}_p \sim P(\boldsymbol{V}_p \mid \boldsymbol{H}_p)$ (Equation (2.3))
        compute the positive term $\langle \boldsymbol{v}\boldsymbol{h}^T \rangle_{data} = \frac{1}{n}\boldsymbol{V}_p\boldsymbol{H}_p^T$
     Negative term:
        initialize $\boldsymbol{V}_m$ randomly
        sample $\boldsymbol{H}_n, \boldsymbol{V}_n$ using Algorithm 1 for k steps
        compute the negative term $\langle \boldsymbol{v}\mathbf{h}^T \rangle_{model} = \frac{1}{n}\boldsymbol{V}_n\boldsymbol{H}_n^T$
     Full update:
        update $\boldsymbol{W}$ with Equation (2.8)
    end for
end for

2.3 Mean-field training

We have seen in Section 1.9 how mean-field theories are a powerful tool to study the kind of neural network models we are dealing with. Here we follow [28] to introduce a mean-field strategy to compute the gradient of the log-

likelihood of the RBM. The advantage of this method is that we can substitute the Montecarlo sampling with deterministic iterative mean-field equations.

To derive an effective free energy we explicitly reintroduce the inverse temperature β

$$P(\boldsymbol{v}, \boldsymbol{h}) = \frac{e^{-\beta E(\boldsymbol{v},\boldsymbol{h})}}{Z}. \tag{2.11}$$

Exploiting the similarity of the RBM to the SK model discussed in Section 1.9, we follow the derivation in [31] and adapt it to the bipartite case. This consists in expanding the free energy at high temperature by setting $\beta \to 0$ to obtain the Thouless-Anderson-Palmer (TAP) expression for the effective free energy [75] that, truncated at second order, is given by

$$\begin{aligned} F_{TAP}(\boldsymbol{m}^v, \boldsymbol{m}^h) = &+ S(\boldsymbol{m}^v) + S(\boldsymbol{m}^h) \\ &- \sum_i a_i m_i^v - \sum_j b_j m_j^h - \sum_{i,j} w_{ij} m_i^v m_j^h \\ &+ \sum_{i,j} \frac{w_{ij}^2}{2} \left(m_i^v - m_i^{v\,2}\right) \left(m_j^h - m_j^{h\,2}\right) \end{aligned} \tag{2.12}$$

with $S(\boldsymbol{m}) = -\sum_i \left[m_i \log m_i + (1 - m_i) \log(1 - m_i)\right]$ and with $\boldsymbol{m}^v, \boldsymbol{m}^h$ being the equilibrium magnetizations of visible and hidden nodes. The minimization of the above Equation (2.12) gives a valid approximation to the free energy F

$$F \simeq F_{TAP}(\tilde{\boldsymbol{m}}^v, \tilde{\boldsymbol{m}}^h), \qquad \left.\frac{dF_{TAP}}{d\boldsymbol{m}}\right|_{\tilde{\boldsymbol{m}}^v, \tilde{\boldsymbol{m}}^h} = 0. \tag{2.13}$$

To obtain $\tilde{\boldsymbol{m}}^v, \tilde{\boldsymbol{m}}^h$ it is then necessary to extremize (2.12) to obtain the following coupled equations

$$m_i^v \simeq sigm\left\{\theta_i + \sum_j \left[w_{ij} m_j^h - w_{ij}\left(m_i^v - \frac{1}{2}\right)\left(m_j^h - m_j^{h\,2}\right)\right]\right\} \tag{2.14}$$

$$m_j^h \simeq sigm\left\{\eta_j + \sum_i \left[w_{ij} m_i^v - w_{ij}\left(m_j^h - \frac{1}{2}\right)\left(m_i^v - m_i^{v\,2}\right)\right]\right\} \tag{2.15}$$

that can be solved by iteration [11]. Finally, given the approximation to the free energy (2.13), the optimization problem over the log-likelihood (2.7) is greatly simplified: the average over the training samples $\langle\cdot\rangle_{data}$ is unchanged while the intractable average over the model distribution $\langle\cdot\rangle_{model}$ is substituted by the maximization of (2.13), giving

$$\Delta \boldsymbol{W} = \alpha \left(\langle \boldsymbol{v}\boldsymbol{h}^T \rangle_{data} - \frac{\partial F_{TAP}(\tilde{\boldsymbol{m}}^v, \tilde{\boldsymbol{m}}^h)}{\partial w_{ij}}\right) \tag{2.16}$$

with

$$\frac{\partial F_{TAP}}{\partial w_{ij}} = -\sum_{i,j} m_i^v m_j^h + w_{ij}(m_i^v - m_i^{v\,2})(m_i^h - m_i^{h\,2}).$$

Summarizing, the training procedure based on TAP approximation is reported in Algorithm 3. We note that similarly to CD/PCD, instead of running the TAP equations (2.14)(2.15) to convergence we choose a fixed number of iterations k to approximate the equilibrium results.

Algorithm 3 k-steps Extended mean-field training (EMFk)

Data: a training set of N data vectors $\boldsymbol{v}_i$
Randomly initialize the weight matrix $\mathbf{W}$
for t = 0 to T (# of epochs) **do**
    Divide the training set in m minibatches of n data vectors $\boldsymbol{v}_i$
    for all minibatches m **do**
     Positive term:
        $\boldsymbol{V}_m = [\boldsymbol{v}_1\ \boldsymbol{v}_2\ \ldots\ \boldsymbol{v}_n]$
        sample $\boldsymbol{H}_p \sim P(\boldsymbol{H}_p \mid \boldsymbol{V}_m)$ (Equation (2.4))
        sample $\boldsymbol{V}_p \sim P(\boldsymbol{V}_p \mid \boldsymbol{H}_p)$ (Equation (2.3))
        compute the positive term $\langle \boldsymbol{v}\boldsymbol{h}^T \rangle_{data} = \frac{1}{n}\boldsymbol{V}_p\boldsymbol{H}_p^T$
     Negative term:
        initialize the batched magnetizations $\boldsymbol{M}_v = \boldsymbol{V}_m$, $\boldsymbol{M}_h = P(\boldsymbol{H} \mid \boldsymbol{V}_m)$
        approximate $\tilde{\boldsymbol{M}}_v$, $\tilde{\boldsymbol{M}}_h$ by iterating Eqs. (2.14),(2.15) for k steps
        negative term: average Equation (2.3) over the samples $\tilde{\boldsymbol{M}}_v$, $\tilde{\boldsymbol{M}}_h$
     Full update:
        update $\boldsymbol{W}$ with Equation (2.8)
    end for
end for

We conclude by noting how the reintroduction of the inverse temperature β is just a formal passage; in the context of the RBM the high-temperature expansion $\beta \to 0$ is substituted by a weak-couplings expansion, under the assumption that the weights w_{ij} are small enough. The variance of the weight matrix can then serve as an *effective inverse temperature* [63]

$$T_{eff} = \frac{1}{Var(\boldsymbol{W})}. \tag{2.17}$$

2.4 Generalized RBM

In Section 2.1 we introduced the RBM using binary units for both the visible and the hidden layers. While this is the original and the most commonly used formulation of the model, we can generalize the definition to use different kinds of units. This is done by introducing a prior probability distribution over visible and hidden variables, that we denote respectively by q_v and q_h:

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} q_v(\boldsymbol{v}) q_h(\boldsymbol{h}) e^{-E(\boldsymbol{v},\boldsymbol{h})}. \tag{2.18}$$

The conditional probability for visible and hidden layers read (Bayes formula)

$$p(\boldsymbol{h}|\boldsymbol{v}) = \frac{p(\boldsymbol{v},\boldsymbol{h})}{\sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h})} = \frac{q_h(\boldsymbol{h})e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{h}} q_h(\boldsymbol{h})e^{-E(\boldsymbol{v},\boldsymbol{h})}} \tag{2.19}$$

$$p(\boldsymbol{v}|\boldsymbol{h}) = \frac{p(\boldsymbol{v},\boldsymbol{h})}{\sum_{\boldsymbol{v}} p(\boldsymbol{v},\boldsymbol{h})} = \frac{q_v(\boldsymbol{v})e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{v}} q_v(\boldsymbol{v})e^{-E(\boldsymbol{v},\boldsymbol{h})}}. \tag{2.20}$$

For the standard binary-binary RBM, a Bernoulli prior is used for both visible and hidden units

$$q_v(v_i) = \frac{1}{2}(\delta_{v_i,0} + \delta_{v_i,1}) \tag{2.21}$$

$$q_h(h_j) = \frac{1}{2}(\delta_{h_j,0} + \delta_{h_j,1}) \tag{2.22}$$

which leads to Equations (2.3),(2.4) for the conditional activations.

When dealing with data whose features are better represented by real numbers, we can substitute the binary variables in the visible layer with Gaussian units. The prior distribution in this case is

$$q_v(v_i) = \frac{1}{\sqrt{2\pi\sigma_v^2}} e^{-\frac{v_i^2}{2\sigma_v^2}} \tag{2.23}$$

and the conditional activation of the visible layer becomes

$$p(v_i,\boldsymbol{h}) = \mathcal{N}\left(v_i; \sum_j w_{ij}h_j + \theta_i, \sigma_v^2\right) \tag{2.24}$$

where $\mathcal{N}(x;\mu,\sigma^2)$ is the Gaussian distribution of variable x with mean μ and variance σ^2. In practice, when using Gaussian visible units, the data are centered and normalized as a preprocessing step and the variance σ_v^2 is set to 1 and kept fixed.

Other choices of priors for the units of the RBM are possible, as well as different combinations of visible and hidden units types; a detailed account can be found in [21]. In this document we are only concerned with the Bernoulli-Bernoulli and Gaussian-Bernoulli models described above.

Chapter 3

Spectral Learning Dynamics of the RBM

Concerning the learning procedure of neural networks, many recent statistical physics based analyses have been proposed, most of them within teacher-student setting [84]. This imposes a rather strong assumption on the data: it is assumed that these are generated from a model belonging to the parametric family of interest, hiding as a consequence the role played by the data themselves in the procedure. From the analysis of related linear models [77, 12], it is already a well established fact that a selection of the most important modes of the Singular Values Decomposition (SVD) of the data is performed in the linear case. In fact in the simpler context of linear feed-forward models the learning dynamics can be fully characterized by means of the SVD of the data matrix [70], showing in particular the emergence of each mode by order of importance with respect to the corresponding singular values.

In this section we present some empirical results to qualify the role of the SVD of the weight matrix of the RBM during learning. We show how the learning procedure follows the dynamics of the strongest SVD modes, in a data-driven fashion. This will let us single out the information content of the RBM, leading us to formulate the assumption that the SVD spectrum is split in a continuous bulk of singular vectors corresponding to noise and a set of outliers that represent the information content. This assumption forms the basis for the mean-field analysis presented in Chapter 4.

The results presented here have been originally detailed in [19] (Reprint A). We will briefly introduce the SVD technique and its relation to Principal Components Analysis. Subsequently, we show how the linearized mean-field equations for the RBM naturally suggest the use of the SVD. We then go on to observe the SVD dynamics in a real-world scenario, characterizing the SVD modes and the statistical assumptions over the weight matrix.

3.1 Principal Component Analysis (PCA)

The PCA technique can be introduced by considering the covariance matrix of a dataset. Given a data matrix $\boldsymbol{X}$ of dimension $n \times d$ with $n > d$, where n is the number of samples and d is the dimension of each sample, and further assuming that samples are centered (i.e. column means have been subtracted, as data are arranged by rows), we can define an unbiased estimator for the related covariance matrix (square and symmetric)

$$\boldsymbol{C} = \frac{\boldsymbol{X}^T \boldsymbol{X}}{n-1} \tag{3.1}$$

that can be diagonalized

$$\boldsymbol{C} = \boldsymbol{V}\boldsymbol{L}\boldsymbol{V}^T \tag{3.2}$$

where the columns of $\boldsymbol{V}$ are eigenvectors of $\boldsymbol{C}$ and $\boldsymbol{L}$ is the diagonal matrix of the eigenvalues λ_α. Projecting the samples over the eigenvectors of the covariance matrix (also called *principal directions* in the context of PCA) we obtain the *principal components*: new, independent variables that account for the maximum possible variability in the data. More precisely, the first *principal component* maximizes the variance of the projections of the data (i.e. it has the highest possible variance) and the succeeding components maximize the variance while satisfying the constraints of being orthogonal to the preceding components. A rigorous demonstration of the properties of *principal components* is given in [10], Section 12.1.

3.2 Singular Value Decomposition (SVD)

The SVD is the generalization of eigenmodes decomposition to rectangular matrices, and it is given by

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{3.3}$$

where $\boldsymbol{U}$ is an orthogonal $n \times d$ matrix whose columns are the *left singular vectors* $\boldsymbol{u}^\alpha$, $\boldsymbol{V}$ is an orthogonal $d \times d$ matrix whose columns are the *right singular vectors* $\boldsymbol{v}^\alpha$ and $\boldsymbol{\Sigma}$ is a diagonal $d \times d$ matrix whose elements are the singular values w_α. The separation into left and right singular vectors is due to the rectangular nature of the decomposed matrix, and the similarity with eigenmodes decomposition is revealed by the following SVD equations

$$\boldsymbol{X}\boldsymbol{v}^\alpha = w_\alpha \boldsymbol{u}^\alpha \tag{3.4}$$

$$\boldsymbol{X}^T\boldsymbol{u}^\alpha = w_\alpha \boldsymbol{v}^\alpha. \tag{3.5}$$

Plugging the SVD of $\boldsymbol{X}$ into the definition of covariance matrix

$$\begin{aligned} \boldsymbol{C} = \frac{\boldsymbol{X}^T\boldsymbol{X}}{n-1} &= \frac{\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T}{n-1} \\ &= \boldsymbol{V}\frac{\boldsymbol{\Sigma}^2}{n-1}\boldsymbol{V}^T \end{aligned} \tag{3.6}$$

we see how the *right singular vectors* can be identified as the *principal directions* and a relation between the singular values w_α and the eigenvalues of the covariance matrix λ_α is easily found

$$\lambda_\alpha = \frac{w_\alpha^2}{n-1} \tag{3.7}$$

Finally, the *principal components* are given by $\boldsymbol{U\Sigma}$ ($\boldsymbol{XV} = \boldsymbol{U\Sigma V^T V} = \boldsymbol{U\Sigma}$).

3.3 Linearized mean-field equations for a RBM

To see how the SVD enters the picture, we linearize the mean-field equations [66] of our model. As there are no connections among variables in the same layer, we can initialize the magnetizations m_i^v, m_j^h with Equations (2.3),(2.4) and write the iterative mean-field equations

$$m_i^v = \text{sigm}\left(\theta_i + \sum_j w_{ij} m_j^h - \sum_j w_{ij}\right) \tag{3.8}$$

$$m_j^h = \text{sigm}\left(\eta_j + \sum_i w_{ij} m_i^v - \sum_i w_{ij}\right). \tag{3.9}$$

At initialization the weights w_{ij} are small, and we can get rid of the external visible field by centering the training data. The external hidden field is instead initialized to zero and it varies slowly, so it doesn't have any effects at the beginning of the training. Thus, neglecting both the external fields we can linearize the mean-field equations to obtain (defining $\tilde{m}_i^v = m_i^v - 1/2, \tilde{m}_j^h = m_j^h - 1/2$ for convenience)

$$\tilde{m}_i^v \simeq \frac{1}{4}\sum_j w_{ij}\tilde{m}_j^h \tag{3.10}$$

$$\tilde{m}_j^h \simeq \frac{1}{4}\sum_i w_{ij}\tilde{m}_i^v. \tag{3.11}$$

We can now express the weights w_{ij} in terms of the SVD as ($u_{i,\alpha}$ identifies the i_{th} component of the α_{th} columns of $\boldsymbol{U}$, and analogous notation is used for $\boldsymbol{V}$)

$$w_{ij} = \sum_\alpha w_\alpha u_{i,\alpha} v_{j,\alpha} \tag{3.12}$$

and expand the magnetizations over the singular vectors

$$\tilde{m}_\alpha^v = \sum_i u_{i,\alpha}\tilde{m}_i^v \tag{3.13}$$

$$\tilde{m}_\alpha^h = \sum_j v_{j,\alpha}\tilde{m}_j^h. \tag{3.14}$$

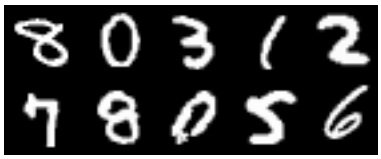


Figure 3.1: Samples of handwritten digits from the MNIST dataset.

Combining Equations (3.10),(3.12),(3.13) and recalling that the columns of $\boldsymbol{U}$ form an orthonormal basis, we get

$$\begin{aligned}\tilde{m}_\alpha^v &= \frac{1}{4}\sum_{\alpha'} w_\alpha \delta_{\alpha,\alpha'} \tilde{m}_\alpha^h \\ &= \frac{1}{4} w_\alpha \tilde{m}_\alpha^h.\end{aligned}$$

We can proceed in an analogous way with $\tilde{m}_\alpha^h$ to finally obtain the coupled iterative equations

$$\tilde{m}_\alpha^v = \frac{1}{4} w_\alpha \tilde{m}_\alpha^h \tag{3.15}$$

$$\tilde{m}_\alpha^h = \frac{1}{4} w_\alpha \tilde{m}_\alpha^v. \tag{3.16}$$

These equations show that at the linear level the magnetizations aligned to the singular vectors with a strong w_α are amplified, while magnetizations related to small w_α are penalized. We then expect the samples generated by a trained RBM to be affine to the strongest singular vectors, and we can try to understand how. To this end, we can better specify the role of the SVD matrices in the context of a RBM:

- $\boldsymbol{U}$ encodes the singular vectors related to the visible layer; these can be visualized in the pixel space and basically consist in the principal components of $\boldsymbol{W}$.

- $\boldsymbol{V}$ is related to the hidden layer; it is a square orthogonal matrix that can be interpreted as a rotation and its columns are the principal directions of $\boldsymbol{W}$.

- The singular values w_j contained in $\boldsymbol{\Sigma}$ can be thought of as scaling factors whose action is to weigh the singular vectors composing $\boldsymbol{W}$.

Given the above characteristics we focused our attention on $\boldsymbol{\Sigma}$ and $\boldsymbol{U}$, tracking the distribution of the singular values and looking at the corresponding left singular vectors during the training.

3.4 Distribution of the singular values

We now turn to the analysis of a training run: we use PCD to train the RBM over the MNIST dataset [50] and we monitor the evolution of the SVD components. The MNIST dataset is composed by 70000 images of handwritten digits (60000 for training and validation, 10000 for testing) of size 28×28 pixels. Some samples from the dataset are shown in Figure 3.1. The weight matrix $\boldsymbol{W}$ is initialized as a Gaussian random matrix with variance σ_w (and zero mean). The eigenvalues distribution of the corresponding symmetric square matrix $\boldsymbol{W}^T\boldsymbol{W}$ is known to be given by the Marchenko-Pastur law [56] in its canonical form. The singular values w_α are related to the eigenvalues λ_j by (3.7), and by defining the parameter $r = N_h/N_v$, with N_h the size of the hidden layer and N_v the size of the visible layer (we recall that $\boldsymbol{W}$ is a $N_v \times N_h$ matrix), the expression of the Marchenko-Pastur law is given by (in the limit $N_h, N_v \to \infty$ with r finite)

$$\rho(\lambda) = \frac{1}{2\pi\sigma_w^2} \frac{\sqrt{(\lambda - r_-)(r_+ - \lambda)}}{r\lambda} \tag{3.17}$$

where the higher and lower bounds $r_\pm$ are

$$r_\pm = \sigma_w^2 \left(1 \pm \sqrt{r}\right)^2 . \tag{3.18}$$

Figure 3.2a shows the agreement between the empirical distribution and the theoretical distribution. In particular, we note how all w_α have values below the threshold set by the Marchenko-Pastur law, forming a *bulk* of singular values. Starting with the training, we see that many singular values increase in magnitude and overcome the threshold for a Gaussian random matrix; these are *outliers* leaving the bulk, shown in Figure 3.2b-3.2d. During the first epochs of training this process is very fast and many w_α values are easily extracted from the bulk, growing by many orders of magnitude. The bulk is instead shrunk to low values, meaning that the w_α values that do not overcome the threshold decrease in magnitude. Going on with the training this process slows down, but it does not stop: outliers keep growing slowly, and the bulk keeps shrinking to approach a spike around zero. It is important to note that a kind of hierarchy is maintained in the process: the first outliers are never overcome by the newly extracted w_α, and this is made clear by looking at the corresponding left singular vectors (see next Section). In the final epochs of training, the singular values w_α are separated into two categories: a concentrated set of almost-null singular values and a set of outliers spread above the threshold, as shown in Figure 3.2e.

The evolution of the w_α distribution described above suggests that the training process is able to discern between the *most important* singular vectors, that are brought above threshold first and heavily strengthened, and a bulk of *less important* singular vectors, that end up above threshold but whose w_α reach values order of magnitude smaller than those of the strongest singular vectors. Moreover, the below-threshold singular vectors are practically eliminated by cutting down the corresponding w_α. These observations give a good

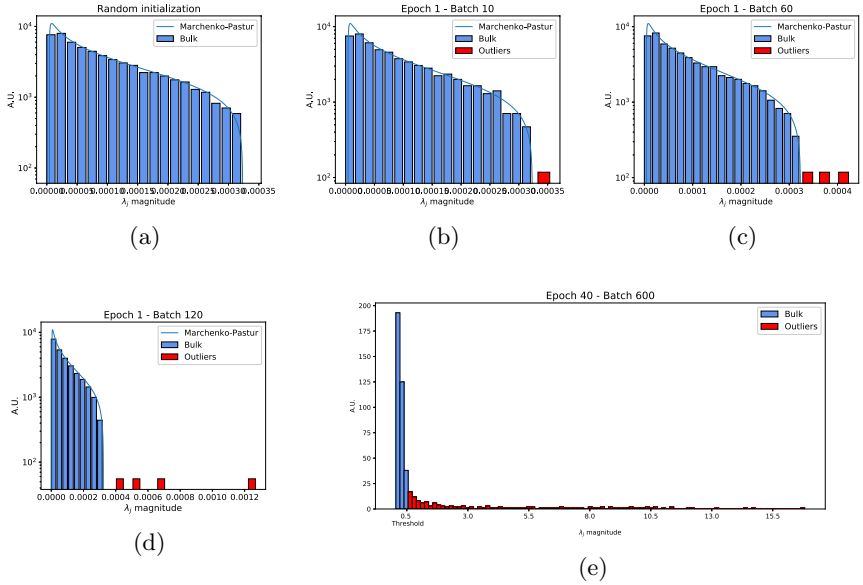


Figure 3.2: **(a)** Singular values distribution of the initial random matrix compared to Marchenko-Pastur law. **(b)-(d)** With the training we can see some singular values strengthening and overcoming the threshold set by the Marchenko-Pastur law. **(e)** Distribution of the singular values after a long training: we can see many outliers spread above threshold and a spike of below-threshold singular values near zero.

indication about what are the dynamics of the learning process, but a couple of matters need to be addressed: (i) it is not clear what singular vectors actually represent, (ii) the meaning of *more* and *less* important singular vectors has to be specified. We will deal with these problems in the next Section.

3.5 Dynamics of the singular vectors

To understand the role of the left singular vectors of an RBM we must keep in mind the interpretation for the SVD decomposition of $\boldsymbol{W}$ given previously. We have seen how the matrix of singular values $\boldsymbol{\Sigma}$ is shaped during learning, and we recall that $\boldsymbol{V}$ is interpreted as a rotation in the space of the hidden units. We then expect to recover the structure of the training data into the $\boldsymbol{U}$ matrix; using the MNIST dataset proves useful in this context as we can visualize the $\boldsymbol{u}^{\alpha}$ vectors as images in the pixel space.

Before focusing on the left singular vectors, we note that also the external visible field can be visualized in the pixel space. Following [40] we can initialize the field to

$$\theta_i = \log[p_i/(1-p_i)] \tag{3.19}$$

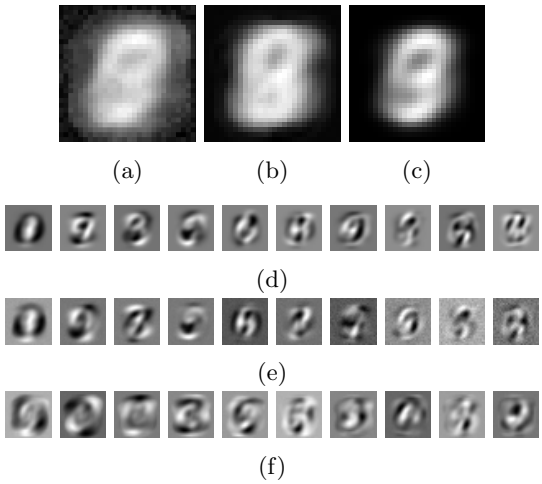


Figure 3.3: **(a)** First mode learnt by the RBM with the external visible field initialized as a null vector. **(b)** External visible field initialized with rule (3.19). **(c)** First principal components extracted from the training set. **(d)** Principal components extracted from the training set (starting from the second). **(e)** The first 10 modes of a RBM trained for 1 epoch, using initialization rule (3.19). **(f)** Same as (e) but after a 10 epochs training.

where p_i is the proportion of training samples in which unit i is active. Using the above rule (3.19) we are able to encode into the field the mean activations of the visible layer, which is clearly shown in Figure 3.3b in the pixel space. If we instead initialize the visible field with a null vector, the mean activation pattern is learned very effectively as the strongest left singular vector $\boldsymbol{u}^\alpha$. The striking resemblance between the mean activation pattern computed from the training data and the one learned by the RBM is shown in Figure 3.3a-3.3b and it serves as a first example of what the $\boldsymbol{u}^\alpha$ vectors represent. It seems then equivalent to either encode the mean activation pattern into the visible field since the beginning or letting the RBM learn such a pattern as a left singular vector. In practice the second case is not desirable as the RBM associates to the mean activation pattern a very strong singular value, many orders of magnitude higher than the strongest outliers. This results in a bias in the sampling from the trained machine, such that the samples whose activation pattern is nearest to the mean are sampled with a higher frequency (in the worst case, those are the only configurations sampled at equilibrium).

The first 10 left singular vectors of a trained RBM are shown in Figure 3.3e-3.3f. They are all composed by a homogeneous background on the borders and a set of alternating dark and light traits in the center, highlighting the fact that each singular vector acts globally on the visible layer. Even if the pictures seen in Figures 3.3e-3.3f are quite different one from another, an interesting trend is found: a higher number of alternating traits is present in the successive vectors.

These observations suggest that the RBM is able to learn the modes that compose the activation patterns of the data, starting with the low frequency modes and proceeding with the high frequency ones. Moreover, with reference to the w_α distribution (Figure 3.2e), we note that the low frequency modes are given a higher weight. Recalling the connection between SVD and PCA (Section 3.2) these modes are understood as the principal modes of variation of the data. In Figures 3.3d-3.3e we compare the SVD modes extracted from the data and those of the $\boldsymbol{W}$ matrix, which prove to be very similar.

The dynamics described here present some similarities to the learning dynamics of deep linear neural networks [70]. Going on with the training we expect non-linear effects to kick-in and this is seen in Figure 3.3f where the SVD of the data is not comparable to the SVD of $\boldsymbol{W}$ anymore.

Analyzing the learning dynamics more in detail, we observed that the modes take shape one by one as the corresponding singular value w_α is brought above threshold. The subsequent strengthening of the w_α values corresponds to refinements and rotations, with little effects on the characterization of the modes as high or low frequency modes. For what concerns the modes below threshold, they present a dark border and a random configuration in the center; in this case the only effect of the training is to discern what are the units which are never activated and no information about the actual structure of the data is found.

Summarizing, some insights on the behavior of an RBM in the linear regime were given by looking at the SVD-like equations (3.15)-(3.16), where we have seen how the magnetizations aligned to the strongest SVD modes are amplified. These magnetizations are thus unstable and they drive the formation of new mean-field fixed points during learning, that correspond to the magnetizations affine to the samples in the training set. These observations highlight a connection between the SVD of the data in the training set and the SVD of the weight matrix $\boldsymbol{W}$, at least in the linear regime.

3.6 Characterization of the modes

By looking at the singular values distribution of $\boldsymbol{W}$ we have seen that there seem to be *more* and *less important* singular vectors. In the previous Section, we have then refined this observation by highlighting how the lowest-frequency modes are given the highest weights. We can then identify the more (less) important modes as the low (high) frequency ones. To gain some intuition about the meaning of this separation we can think about the Fourier decomposition of a square wave; in such a case the superposition of the low frequency harmonics is sufficient to build a good approximation of a square wave, while the role of the high frequency harmonics is that of sharpening the waveform at the discontinuity points. In the context of a trained RBM, we then expect that good approximations to the training data are obtained by exploiting only the low frequency modes, while the high frequency modes should represent minor corrections. To discern between high and low frequency modes, we look at the w_α

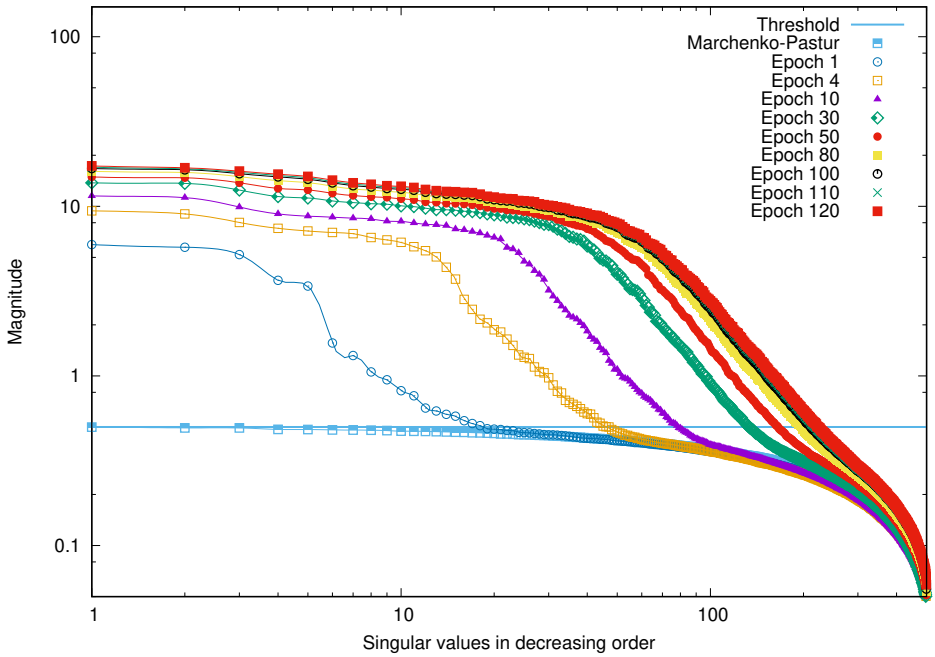
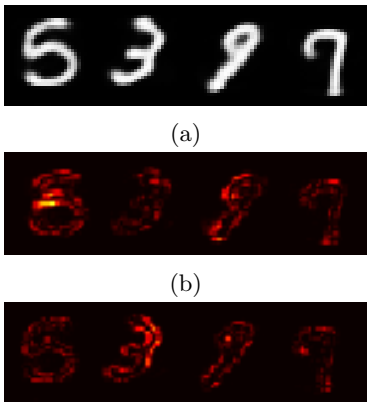


Figure 3.4: Log-log plot of the singular values represented as discrete abscissas (in decreasing order) with their magnitude reported on the ordinates. A cutoff is highlighted by the onset of the linear behavior.

values in decreasing order on a log-log plot (Figure 3.4): the strongest w_α values are located far above threshold and are of comparable magnitude, followed by a tail of exponentially damped values. This picture is consistent across the training, the only difference being the damping cutoff that is increasing with the epochs. After a relatively long training, however, the increase in the cutoff is very slow and this could serve as a signal to stop the learning, as such situation amounts to slowly strengthening singular values which are exponentially less important than the already learned modes. We are then driven to define the more important low frequency modes as the modes before cutoff, and the exponentially less important high frequency modes as those after cutoff. A consistency check is shown in Figure 3.5, where just the 100 strongest modes are retained to construct the samples and the remaining modes are shown to encode boundary corrections. Choosing the first 100 modes is arbitrary; in Figure 3.4 we can see how the cutoff is well below 100, so with this choice we are sure that we included in the reconstruction of the samples all the strong modes before cutoff plus a small number of modes giving boundary corrections.

As a conclusion, the above observations suggest that the weight matrix of a trained RBM is composed by two classes of modes; recalling the expansion



(a)

(b)

(c)

Figure 3.5: **(a)** The image shows some samples obtained with the trained RBM (after a 40 epochs training) and then "filtered" by eliminating the 400 weakest modes (just the 100 strongest modes are retained). **(b)** The images are composed by eliminating the 100 strongest modes to see what the weakest modes actually encode (20 epochs training). **(c)** As in **(b)** but after a 40 epochs training.

(3.12) we can express the components of $\boldsymbol{W}$ as

$$w_{ij} = \sum_{\alpha \in bulk} w_\alpha u_i^\alpha v_j^\alpha + \sum_{\alpha \in outliers} w_\alpha u_i^\alpha v_j^\alpha \tag{3.20}$$

where the modes of the bulk are those that remain random after the training (below threshold or after cutoff) while the outliers correspond to the modes that actually encode the structure of training data. This separation sets the basis for the mean-field analysis that we develop in the next section.

Chapter 4

Mean-field theory

In Section 1.9 we have discussed how statistical physics tools have been fruitfully applied to model neural network models. In particular, various works [2, 44, 43, 8, 63] have studied the equilibrium properties of the RBM in order to understand collective phenomena in the latent representation. The common assumption is that the components of the weight matrix are i.i.d., and some insights about the equilibrium phase of a trained RBM are given. However, this approximation is problematic since the learning procedure introduces correlations among the weights making the assumption of independent weights unsuitable to describe a realistic model of the RBM.

In this Chapter we introduce a statistical ensemble for the weight matrix $\boldsymbol{W}$ in which we drop the assumption of independent components and we apply the Replica Method to derive a mean-field theory of the RBM and characterize the equilibrium states of the model. These results have been presented originally in [20] (Reprint B).

4.1 Statistical ensemble

In Chapter 3 we have seen how the evolution of the singular values of the trained model departs strongly from a Marchenko-Pastur distribution, which would be expected if the weights were i.i.d. Gaussian. In particular, we have seen how the information content of the trained RBM is encoded into a finite number of spectral modes, while a residual part of the spectrum constitutes a noise source. [1] We can then assume the weight matrix $\boldsymbol{W}$ to be composed by K spectral modes and a portion of random noise n

$$\boldsymbol{W}_{ij} = \sum_{\alpha=1}^{K} w_\alpha u_i^\alpha v_j^\alpha + r_{ij}. \tag{4.1}$$

[1] The separation of the weight matrix spectrum into a structured part and a random part has been observed over different real-world datasets and for different configurations of the trained RBM model. While this separation might not always hold, it is a relevant case in practice.

The vectors $\boldsymbol{u}^{\boldsymbol{\alpha}}, \boldsymbol{v}^{\boldsymbol{\alpha}}$ represent the K strongest components of the SVD of the weight matrix, while the components r_{ij} represent Gaussian noise. The above formula describes a class of models that we can analyze statistically, i.e. the statistical ensemble that we choose to model the RBM. In the following, we derive the static and dynamical properties of this ensemble.

4.2 Order parameters and effective free energy

To obtain an effective free energy (Section 1.6) for the RBM we apply the Replica method [61], which is based on the following identity

$$E_{u,v,r}\left[\log Z\right] = \lim_{p\to 0} \frac{d}{dp} E_{u,,v,r}\left[Z^p\right] \tag{4.2}$$

where u, v are the singular vectors components and r the Gaussian noise defined in Section 4.1. The $\boldsymbol{v}^\alpha$ and $\boldsymbol{u}^\alpha$ vectors are orthonormal, giving $v \sim O\left(1/\sqrt{N_h}\right)$ and $u \sim O\left(1/\sqrt{N_v}\right)$, and the noise terms are i.i.d. $r_{ij} = N(0, \sigma^2/L)$ where N_v, N_h are the numbers of visible and hidden units and $L = \sqrt{N_v N_h}$. We see from this equation that our objective is to average the free energy over the noise and the components of the singular vectors; the second average is the crucial one, as it represents the average over the structured part of the weight matrix, i.e. the part that encodes the correlations among the weights. Nonetheless, the components of the vectors $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ are assumed to be i.i.d., and we will see that the choice of their distribution determines the properties of the equilibrium states.

An important assumption in the replica method is that the exponent p assumes integer values, hence the replicated partition function Z^p can be thought of as the partition function of a system comprising p copies, i.e. the replicas, of the original system. We will see that exploring the correlations among the different replicas of the system will be crucial to fully characterize the equilibrium behavior of the system. For now the only concern is that Equation (4.2) requires taking the limit $p \to 0$ while here we are assuming p to be integer; we will discuss this incongruence at the end of the Section.

We can start by writing down the replicated partition function Z^p

$$\begin{aligned} Z^p = \prod_{a=1}^{p} \sum_{\boldsymbol{s}^a, \boldsymbol{\sigma}^a} \exp\left\{ -\sum_{i,j,\alpha} s_i^a u_i^\alpha w_\alpha v_j^\alpha \sigma_j^a - \sum_{i,j} s_i^a \sigma_j^a r_{ij} \right\} \\ \times \exp\left\{ -\sum_i \eta_i s_i^a - \sum_j \theta_j \sigma_j^a \right\}. \end{aligned} \tag{4.3}$$

By isolating the part of Z^p that contains the noise terms r_{ij} we can compute the noise average to obtain

$$\exp\left[\frac{\sigma^2}{2L}\left(\sum_a s_i^a \sigma_j^a\right)^2\right] = \exp\left[\frac{\sigma^2}{2L}\left(p + \sum_{a\neq b} s_i^a s_i^b \sigma_j^a \sigma_j^b\right)\right].$$

Considering now the expansion of the units over the singular modes

$$s_\alpha^a = \frac{1}{\sqrt{L}} \sum_i s_i^a u_i^\alpha, \qquad \sigma_\alpha^a = \frac{1}{\sqrt{L}} \sum_j \sigma_j^a v_j^\alpha \tag{4.4}$$

and neglecting the fields η_i, θ_j for a moment we can rewrite the replicated partition function as

$$Z^p \propto \prod_{a=1}^{p} \sum_{\boldsymbol{s}^a, \boldsymbol{\sigma}^a} \exp\left\{ -L \sum_\alpha w_\alpha s_\alpha \sigma_\alpha + \frac{\sigma^2}{2L} \sum_{i,j,a\neq b} s_i^a s_i^b \sigma_j^a \sigma_j^b \right\}. \tag{4.5}$$

We observe that both sums in the exponential include higher order terms that we cannot factorize to simply compute the equilibrium averages over u and v. Following the strategy presented in Section 1.7 for the Hopfield model, we can introduce an appropriate set of order parameters to linearize the exponents. For the first term we can consider the magnetization of the system $m_\alpha^a, \bar{m}_\alpha^a$ correlated with the spectral modes, expressed as

$$m_\alpha^a \sim E_{u,v,r}\left(\langle \sigma_\alpha^a \rangle\right), \qquad \bar{m}_\alpha^a \sim E_{u,v,r}\left(\langle s_\alpha^a \rangle\right) \tag{4.6}$$

and introduced with the following integral identity (Hubbard-Stratonovich transformation):

$$\exp\left(L \sum_\alpha w_\alpha s_\alpha^a \sigma_\alpha^a \right) \propto \int \prod_\alpha \frac{dm_\alpha^a d\bar{m}_\alpha^a}{2\pi} \times \exp\left(-L \sum_\alpha w_\alpha \left(m_\alpha^a \bar{m}_\alpha^a - m_\alpha^a s_\alpha^a - \bar{m}_\alpha^a \sigma_\alpha^a \right) \right).$$

We note that the magnetizations $m_\alpha^a, \bar{m}_\alpha^a$ play the same role as the magnetization over the stored patterns in the Hopfield model; for the RBM we effectively substitute the patterns with the spectral modes. The second sum in (4.5) stems from the average over the noise r_{ij}, whose effect has been to couple the different replicas of the system. To treat the now interacting replicas, we take into consideration their correlations by introducing the following order parameters:

$$Q_{ab} \sim E_{u,v,r}\left(\langle \sigma_i^a \sigma_i^b \rangle \right), \qquad \bar{Q}_{ab} \sim E_{u,v,r}\left(\langle s_j^a s_j^b \rangle \right). \tag{4.7}$$

These are the Edwards-Anderson order parameters and are fundamental to paint a detailed description of the equilibrium states of the system. They are introduced with the following Hubbard-Stratonovich transformation

$$\exp\left[\frac{\sigma^2}{2L} \left(\sum_{i,j,a\neq b} s_i^a s_i^b \sigma_j^a \sigma_j^b \right) \right] = \int \prod_{a\neq b} \frac{dQ_{ab} d\bar{Q}_{ab}}{2\pi} \times \exp\left[-\frac{L\sigma^2}{2} \sum_{a\neq b} \left(Q_{ab}\bar{Q}_{ab} - \frac{Q_{ab}}{N_v} \sum_i s_i^a s_i^b - \frac{\bar{Q}_{ab}}{N_h} \sum_j \sigma_j^a \sigma_j^b \right) \right]$$

whose utility is again to simplify the expression by linearizing the sum in the exponent w.r.t. the nodes of single replicas.

Putting together the above manipulations and including the fields, we rewrite the full average as

$$E_{u,v,r}\left[Z^p\right] = \int \prod_{a,\alpha} \frac{dm_\alpha^a d\bar{m}_\alpha^a}{2\pi} \prod_{a\neq b} \frac{dQ_{ab} d\bar{Q}_{ab}}{2\pi}$$
$$\times \exp\left[-L\left(\sum_{a,\alpha} w_\alpha m_\alpha^a \bar{m}_\alpha^a + \frac{\sigma^2}{2}\sum_{a\neq b} Q_{ab}\bar{Q}_{ab} - \frac{1}{\sqrt{\kappa}} A(m,Q) - \sqrt{\kappa} B(\bar{m},\bar{Q})\right)\right] \tag{4.8}$$

with $\kappa = \frac{N_h}{N_v}$ and

$$A(m,Q) \stackrel{\text{def}}{=} \log\left(\sum_{s^a\in -1,1} \mathbb{E}_u\left[\exp\left(\frac{\sqrt{\kappa}\sigma^2}{2}\sum_{a\neq b} Q_{ab}s^a s^b + \kappa^{\frac{1}{4}}\sum_{a,\alpha}(w_\alpha m_\alpha^a - \eta_\alpha)u^\alpha s^a\right)\right]\right)$$
$$B(\bar{m},\bar{Q}) \stackrel{\text{def}}{=} \log\left(\sum_{\sigma^a\in -1,1} \mathbb{E}_v\left[\exp\left(\frac{\sqrt{\kappa}\sigma^2}{2}\sum_{a\neq b} \bar{Q}_{ab}\sigma^a\sigma^b + \kappa^{\frac{1}{4}}\sum_{a,\alpha}(w_\alpha m_\alpha^a - \theta_\alpha)v^\alpha\sigma^a\right)\right]\right).$$

where the fields are also expanded over the spectral modes and the projections are assumed to be $O(1)$:

$$\eta_\alpha \stackrel{\text{def}}{=} \frac{1}{\sqrt{L}}\sum_i \eta_i u_i^\alpha = O(1) \tag{4.9}$$

$$\theta_\alpha \stackrel{\text{def}}{=} \frac{1}{\sqrt{L}}\sum_j \theta_j v_j^\alpha = O(1). \tag{4.10}$$

The expression in Equation (4.8) makes it possible to compute the average over u, v with the saddle point method by letting $L \to \infty$ (thermodynamic limit), which amounts to considering the average over a large system (we recall that $L = \sqrt{N_v N_h}$). In order to proceed, we need to address a couple of issues:

i) we need to specify an explicit dependence of the order parameters on the replica indices a, b;

ii) given the solution to the saddle point equations, we need to consider the analytic continuation $p \to 0$ as in (4.2) to obtain the actual average free energy. At this point, it is unclear how to perform this limit.

As for point i) the simplest assumption is that the order parameters should not depend on the replica index; this seems to make sense given that replicas have been introduced artificially in Equation (4.2) for mathematical convenience and therefore we expect that they do not have any effects on the physics of the system. Point ii) is more subtle, and a possible approach is to consider a

specific structure for the order parameters such that the saddle point equations are analytic in p [61].

The above assumptions are realized by the *replica symmetric ansatz*

$$Q_{ab} = \delta_{ab} + (1 - \delta_{ab})q \tag{4.11}$$
$$\bar{Q}_{ab} = \delta_{ab} + (1 - \delta_{ab})\bar{q} \tag{4.12}$$

in which the dependence on the replica index is dropped and the set of Edwards-Anderson order parameters is reduced to the couple of $q, \bar{q}$ parameters. The resulting Q matrices are symmetric, filled with unity on the diagonal and with $q, \bar{q}$ off-diagonal, and they constitute a simple example of an ultrametric matrix. This simple structure makes it easy to plug (4.11), (4.12) into (4.8) to transform the sums involving the $Q_{ab}, \bar{Q}_{ab}$ terms into analytic expressions involving p and $q, \bar{q}$.

We can now take the limit $p \to 0$ to obtain the effective free energy

$$\begin{aligned} f(m, \bar{m}, q, \bar{q}) = &\sum_{\alpha} w_\alpha m_\alpha \bar{m}_\alpha - \frac{\sigma^2}{2} q\bar{q} + \frac{\sigma^2}{2}(q + \bar{q}) \\ &- \frac{1}{\sqrt{\kappa}} \mathbb{E}_{u,x} \left[\log 2\cosh(h(x,u))\right] - \sqrt{\kappa}\, \mathbb{E}_{v,x} \left[\log 2\cosh\big(\bar{h}(x,v)\big)\right]. \end{aligned} \tag{4.13}$$

where $x \sim \mathcal{N}(x; 0, 1)$.

Finally, in the limit $L \to \infty$ we extremize the effective free energy to obtain the saddle point equations

$$m_\alpha = \kappa^{\frac{1}{4}}\, \mathbb{E}_{v,x}[v^\alpha \tanh\big(\bar{h}(x,v)\big)] \qquad q = \mathbb{E}_{v,x}[\tanh^2(\bar{h}(x,v))] \tag{4.14}$$
$$\bar{m}_\alpha = \kappa^{-\frac{1}{4}}\, \mathbb{E}_{u,x}[u^\alpha \tanh(h(x,u))] \qquad \bar{q} = \mathbb{E}_{u,x}[\tanh^2(h(x,u))] \tag{4.15}$$

with

$$h(x,u) \overset{\text{def}}{=} \kappa^{\frac{1}{4}} \left(\sigma\sqrt{q}x + \sum_\gamma (w_\gamma m_\gamma - \eta_\gamma) u^\gamma \right)$$
$$\bar{h}(x,v) \overset{\text{def}}{=} \kappa^{-\frac{1}{4}} \left(\sigma\sqrt{\bar{q}}x + \sum_\gamma (w_\gamma \bar{m}_\gamma - \theta_\gamma) v^\gamma \right).$$

The numerical solution of (4.14), (4.15) characterizes the equilibrium states of the system and in the next Section we will see what are the different phases of operation of the RBM and what is the impact of the choice of distribution for the u, v components.

To conclude this Section, let us remark that to obtain Equation (4.13) we need to take the limit $p \to 0$ coming from Equation (4.2) and the thermodynamic limit $L \to \infty$ by carefully applying the Replica method's recipe. Indeed, there is no mathematical justification for taking the analytic continuation $p \to 0$ after having treated p as an integer; moreover, we freely exchanged

the order of the $p \to 0$ and $L \to \infty$ limits. While lacking a rigorous mathematical justification, the Replica method has been shown to provide the correct equilibrium description for the SK model and other similar models to which it has been applied [61]; for the RBM, our experimental results (Section 4.8) are compatible with the theoretical description.

4.3 Phase diagram

The state of the system described by the solutions to Equation (4.14)(4.15) is directly characterized by the value of the order parameters. The stable solutions are determined by looking at the values of the order parameters for which the Hessian of the effective free energy is positive definite. In turn, this let us identify the unstable modes that define the lines of separation among the different phases. In the basic case with no biases we identify three different phases:

- a paramagnetic phase ($q = \bar{q} = m_\alpha = \bar{m}_\alpha = 0$) (P),
- a ferromagnetic phase ($q, \bar{q}, m_\alpha, \bar{m}_\alpha \neq 0$) (F),
- a spin glass phase ($q, \bar{q} \neq 0; m_\alpha = \bar{m}_\alpha = 0$) (SG).

The full phase diagram is found in Figure 4.1.

4.4 Learning phase

For an RBM learned on real-world data we expect that at equilibrium the nodes condense over the spectral modes, determining a certain magnetization of the system. We are thus interested in fully characterizing this learning phase, which corresponds to the ferromagnetic phase introduced in Section 4.3 with $m_\alpha, \bar{m}_\alpha \neq 0$. Our objective is describing the magnetizations of the system in terms of a combination of spectral modes, in order to obtain compositional states that are suited to represent realistic data. This is in analogy to [63] in which the equilibrium states are described in terms of a composition of hidden nodes.

To our end, we are going to show how the distribution of the singular vector components u, v determines the structure of the learning phase and how the choice of an appropriate distribution is fundamental to describe a realistic system. As a starting point, we consider Gaussian i.i.d. components. Once again we discard the biases, so that the Gaussian averaging of the magnetization part of the saddle point equations (4.14)(4.15) reads

$$m_\alpha = w_\alpha \bar{m}_\alpha (1 - q) \tag{4.16}$$

$$\bar{m}_\alpha = w_\alpha m_\alpha (1 - \bar{q}) \tag{4.17}$$

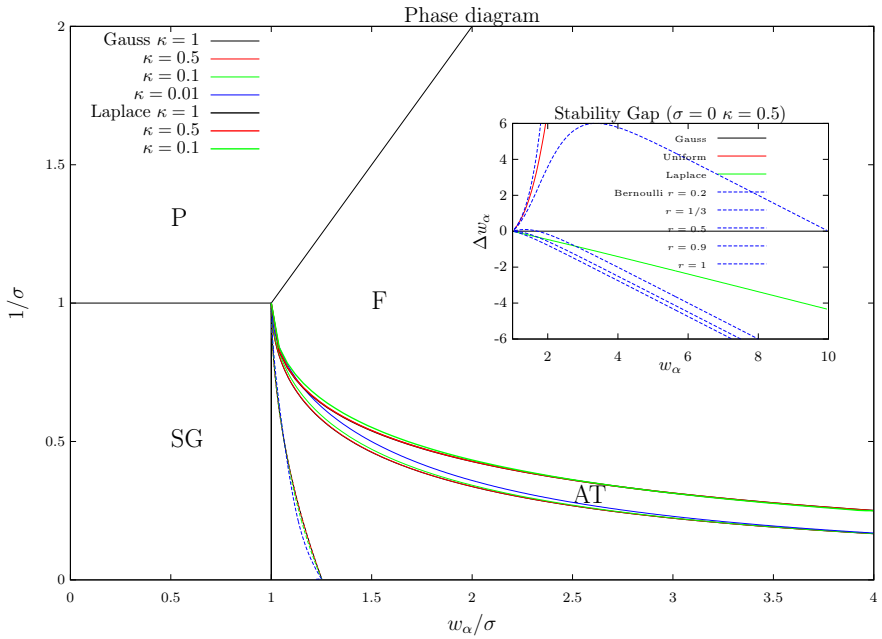


Figure 4.1: Phase diagram in absence of bias and with a finite number of modes, with Gaussian and Laplace distributions for u and v. Inset: high temperature ($\sigma = 0$) stability gap Δw_α corresponding to a fixed point associated to a mode β, expressed as a function of w_α and considering various distributions. Reproduced from [20].

from which we see that the singular value of a mode contributing a non-zero magnetization respects (singular values being non-negative)

$$w_\alpha = \frac{1}{\sqrt{(1-q)(1-\bar{q})}}. \tag{4.18}$$

With non-degenerate singular values the above eq. (4.18) can be satisfied by a single spectral mode, revealing how the i.i.d. Gaussian averaging falls short of realizing the compositional phase we are looking for.

To see why the Gaussian averaging fails and how to fix it, we can approach the averaging in a more general fashion. Assuming the distribution of u and v to be even and introducing the auxiliary distributions

$$p^*(u) \overset{\text{def}}{=} -\int_{-\infty}^{u} xp(x)dx = \int_{|u|}^{\infty} xp(x)dx, \qquad p(x) = p(-x)$$
$$p_\alpha(\boldsymbol{u}) \overset{\text{def}}{=} p^*(u^\alpha) \prod_{\beta \neq \alpha} p(u^\beta)$$

we can define the new variables $q_\alpha, \bar{q}_\alpha$

$$q_\alpha = \int dx \frac{e^{-x^2/2}}{\sqrt{2\pi}} d\boldsymbol{v} p_\alpha(\boldsymbol{v}) \tanh^2 \left(\kappa^{-1/4} \left(\sigma \sqrt{\bar{q}} x + \sum_\gamma w_\gamma \bar{m}_\gamma v^\gamma \right) \right) \tag{4.19}$$

$$\bar{q}_\alpha = \int dx \frac{e^{-x^2/2}}{\sqrt{2\pi}} d\boldsymbol{u} p_\alpha(\boldsymbol{u}) \tanh^2 \left(\kappa^{1/4} \left(\sigma \sqrt{q} x + \sum_\gamma w_\gamma m_\gamma u^\gamma \right) \right) \tag{4.20}$$

and rewrite the averaging in the saddle point equations (4.14)(4.15) as

$$m_\alpha = w_\alpha \bar{m}_\alpha (1 - q_\alpha) \tag{4.21}$$

$$\bar{m}_\alpha = w_\alpha m_\alpha (1 - \bar{q}_\alpha). \tag{4.22}$$

from which we get a generalized version of Equation (4.18) for the singular value of a mode contributing to the magnetization

$$w_\alpha = \frac{1}{\sqrt{(1 - q_\alpha)(1 - \bar{q}_\alpha)}} \stackrel{\text{def}}{=} w(q_\alpha, \bar{q}_\alpha). \tag{4.23}$$

To explicitate the conditions under which multiple modes are able to condense and contribute to the magnetization, we can assume that a single mode α has condensed and consider the onset of a mode $\beta (\beta \neq \alpha)$. The stability condition for mode β coexisting with mode α is given by setting $w_\gamma = 0, \forall \gamma \neq \alpha, \beta$ in the expression of the effective free energy (4.13) and looking for the positive definiteness of its Hessian. This gives a stability gap

$$\Delta w_\alpha \stackrel{\text{def}}{=} w(q, \bar{q}) - w(q_\alpha, \bar{q}_\alpha) \tag{4.24}$$

for which an arbitrary β mode is unstable if

$$w_\beta < w_\alpha + \Delta w_\alpha. \tag{4.25}$$

In the Gaussian case the distribution p^* that we introduced is still Gaussian ($p^*(u) = p(u)$), meaning that the $q_\alpha, \bar{q}_\alpha$ parameters don't really depend on α and resulting in a null gap $\Delta w_\alpha = 0$. This discards the possibility of a compositional phase as all modes β with $w_\beta < w_\alpha$ are unstable and thus only the strongest spectral mode contributes to the magnetization of the system. Considering other distributions for the u, v components results in obtaining a set of different $q_\alpha, \bar{q}_\alpha$ parameters for each α mode, impacting the value of the stability gap. With a negative gap, the possibility of multiple spectral modes being stable is allowed and thus a compositional phase is possible. In the inset of Figure 4.1 the stability gap is traced for various simple distributions. Of particular interest is the Laplace distribution for the spectral components, as it presents a negative gap, making it a good candidate for a realistic theoretical description of the RBM. We will exploit this fact in the following.

4.5 Learning equations

Building on the insights given in Chapter 3 and the above Sections, we will exploit the decomposition of the learning equations (2.8), (2.9), (2.10) over the SVD components. Introducing a time variable t we rewrite (4.1) as

$$w_{ij}(t) = \sum_{\alpha} w_{\alpha}(t) u_{i,\alpha}(t) v_{j,\alpha}(t) \tag{4.26}$$

where we have discarded the noise term r_{ij} in (4.1) as at the onset of the learning the weight matrix $\boldsymbol{W}$ is random and thus we don't need to consider a separate set of noisy modes. Moreover, in order to define deterministic dynamics, we also discard the noise due to the stochastic ascent optimization by taking the continuous limit of eqs. (2.8), (2.9), (2.10) to obtain

$$\begin{aligned}
\frac{dw_{ij}}{dt} &= \langle s_i \sigma_j \rangle_{data} - \langle s_i \sigma_j \rangle_{model} \\
\frac{d\eta_i}{dt} &= \langle s_i \rangle_{data} - \langle s_i \rangle_{model} \\
\frac{d\theta_j}{dt} &= \langle \sigma_j \rangle_{data} - \langle \sigma_j \rangle_{model}.
\end{aligned}$$

Projecting the above equations on the SVD basis we obtain

$$\frac{1}{L}\left(\frac{d\mathbf{W}}{dt}\right)_{\alpha\beta} = \langle s_\alpha \sigma_\beta \rangle_{data} - \langle s_\alpha \sigma_\beta \rangle_{model} \tag{4.27}$$

$$\frac{1}{\sqrt{L}}\left(\frac{d\eta}{dt}\right)_{\alpha} = \langle s_\alpha \rangle_{data} - \langle s_\alpha \rangle_{model} \tag{4.28}$$

$$\frac{1}{\sqrt{L}}\left(\frac{d\theta}{dt}\right)_{\alpha} = \langle \sigma_\alpha \rangle_{data} - \langle \sigma_\alpha \rangle_{model} \tag{4.29}$$

with

$$s_\alpha = \frac{1}{\sqrt{L}} \sum_i s_i u_{i,\alpha}\,, \qquad \sigma_\alpha = \frac{1}{\sqrt{L}} \sum_j \sigma_j v_{j,\alpha}. \tag{4.30}$$

The left-hand side of (4.27),(4.28),(4.29) can be expressed in a simple form by expanding it over the basis defined by the SVD

$$\begin{aligned}\left(\frac{d\mathbf{W}}{dt}\right)_{\alpha\beta} &= \sum_{ij} u_{i,\alpha}\frac{dw_{ij}}{dt}v_{j,\beta} \\ &= \delta_{\alpha,\beta}\frac{dw_\alpha}{dt} + (1-\delta_{\alpha\beta})\left(w_\alpha\frac{d\boldsymbol{v}^{\alpha,T}}{dt}\boldsymbol{v}^\beta + w_\beta\boldsymbol{u}^{\alpha,T}\frac{d\boldsymbol{u}\beta}{dt}\right) \\ &= \delta_{\alpha,\beta}\frac{dw_\alpha}{dt} + (1-\delta_{\alpha\beta})\left(w_\alpha\Omega^u_{\alpha\beta} + w_\beta\Omega^v_{\beta\alpha}\right) \end{aligned} \tag{4.31}$$

$$\frac{1}{\sqrt{L}}\left(\frac{d\eta}{dt}\right)_\alpha = \frac{d\eta}{dt} - \sum_\beta \Omega^u_{\alpha\beta}\eta_\beta \tag{4.32}$$

$$\frac{1}{\sqrt{L}}\left(\frac{d\theta}{dt}\right)_\alpha = \frac{d\theta}{dt} - \sum_\beta \Omega^v_{\alpha\beta}\eta_\beta \tag{4.33}$$

where we have defined the generators of rotations in both $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ bases

$$\Omega^u_{\alpha\beta}(t) = \frac{d\boldsymbol{u}^{\alpha,T}}{dt}\boldsymbol{u}^\beta \tag{4.34}$$

$$\Omega^v_{\alpha\beta}(t) = \frac{d\boldsymbol{v}^{\alpha,T}}{dt}\boldsymbol{v}^\beta. \tag{4.35}$$

Finally, we can express the rotation generators in terms of the other quantities

$$\Omega^u_{\alpha\beta}(t) = -\frac{1}{w_\alpha + w_\beta}\left(\frac{dW}{dt}\right)^A_{\alpha\beta} + \frac{1}{w_\alpha - w_\beta}\left(\frac{dW}{dt}\right)^S_{\alpha\beta} \tag{4.36}$$

$$\Omega^v_{\alpha\beta}(t) = \frac{1}{w_\alpha + w_\beta}\left(\frac{dW}{dt}\right)^A_{\alpha\beta} + \frac{1}{w_\alpha - w_\beta}\left(\frac{dW}{dt}\right)^S_{\alpha\beta} \tag{4.37}$$

with

$$\left(\frac{dW}{dt}\right)^{A,S}_{\alpha\beta} \stackrel{\text{def}}{=} \frac{1}{2}\left(\langle s_\alpha\sigma_\beta\rangle_{Data} \pm \langle s_\beta\sigma_\alpha\rangle_{Data} \mp \langle s_\beta\sigma_\alpha\rangle_{RBM} - \langle s_\alpha\sigma_\beta\rangle_{RBM}\right). \tag{4.38}$$

4.6 Linear regime

The random initialization of $\boldsymbol{W}$ generally entails that the magnetization of the system is small in the beginning, so that we can describe the onset of learning by considering an expansion of the mean-field free energy up to second order in the magnetizations. For the RBM, we can adapt the expression of the effective free energy (1.11) to contemplate a bipartite structure and non-constant w_{ij}

couplings to obtain

$$F(\boldsymbol{m}^v, \boldsymbol{m}^h) \simeq \frac{1}{2}\sum_{i=1}^{N}(1+m_i^v)\log(1+m_i^v) + (1-m_i^v)\log(1-m_i^v) \tag{4.39}$$

$$+ \frac{1}{2}\sum_{j=1}^{M}(1+m_j^h)\log\big(1+m_j^h\big) + (1-m_j^h)\log\big(1-m_j^h\big) \tag{4.40}$$

$$- \sum_{i,j} w_{ij} m_i^v m_j^h + \sum_{i=1}^{N} a_i m_i^v + \sum_{j=1}^{M} b_j m_j^h \tag{4.41}$$

$$\simeq \frac{1}{2}\sum_{i=1}^{N}\left(m_i^v\right)^2 + \frac{1}{2}\sum_{j=1}^{M}\left(m_j^h\right)^2 - \sum_{ij} w_{ij} m_i^v m_j^h + \sum_{i=1}^{N} a_i m_i^v + \sum_{j=1}^{M} b_j m_j^h. \tag{4.42}$$

The corresponding probability measure at fixed magnetization is Gaussian (1.11), for which we can write the covariance matrix in the following form (σ_v, σ_h being the variances of visible and hidden magnetizations)

$$\mathrm{cov}(\boldsymbol{m}^v, \boldsymbol{m}^h) = \begin{pmatrix} \frac{\sigma_h^{-2}}{\sigma_v^{-2}\sigma_h^{-2} - \mathbf{W}\mathbf{W}^\mathbf{T}} & \mathbf{W}\frac{1}{\sigma_v^{-2}\sigma_h^{-2} - \mathbf{W}^\mathbf{T}\mathbf{W}} \\ \mathbf{W}^\mathbf{T}\frac{1}{\sigma_v^{-2}\sigma_h^{-2} - \mathbf{W}\mathbf{W}^\mathbf{T}} & \frac{\sigma_h^{-2}}{\sigma_v^{-2}\sigma_h^{-2} - \mathbf{W}\mathbf{W}^\mathbf{T}} \end{pmatrix}. \tag{4.43}$$

In this mean-field picture, the values of visible and hidden nodes are identified with the magnetizations. This consists in treating a RBM with Gaussian variables, and further assuming the external fields and the mean of the data to be null (normalization and rescaling of the training set can determine such conditions) we can rewrite the empirical expectation in (4.27) as

$$\langle s_\alpha \sigma_\beta \rangle_{data} = \sigma_h^2 w_\beta \langle s_\alpha \sigma_\beta \rangle_{data} = \sigma_h^2 w_\beta \,\mathrm{cov}(s_\alpha, s_\beta) \tag{4.44}$$

where we see that the covariance matrix of the data comes out.

The deterministic learning equation of the weight matrix (4.27) can thus be written explicitly: the empirical average is given by Equation (4.44), while the model average is given by the covariance matrix (4.43). The diagonal part of the equation then reads

$$\frac{dw_\alpha}{dt} = \sigma_h^2 w_\alpha \left(\langle s_\alpha^2 \rangle_{data} - \frac{\sigma_v^2}{1 - \sigma_v^2 \sigma_h^2 w_\alpha^2} \right). \tag{4.45}$$

while for the off-diagonal part we retain the rotation generators $\Omega_{\alpha\beta}^{u,v}$ rewriting them as

$$\Omega_{\alpha\beta}^{u,v} = (1-\delta_{\alpha\beta})\sigma_h^2 \left(\frac{w_\beta - w_\alpha}{w_\alpha + w_\beta} \mp \frac{w_\beta + w_\alpha}{w_\alpha - w_\beta} \right) \langle s_\alpha s_\beta \rangle_{Data}. \tag{4.46}$$

In the equation above we see that the rotations are null only when $\langle s_\alpha s_\beta \rangle$ is diagonal, and this happens when the spectral modes are aligned to the principal

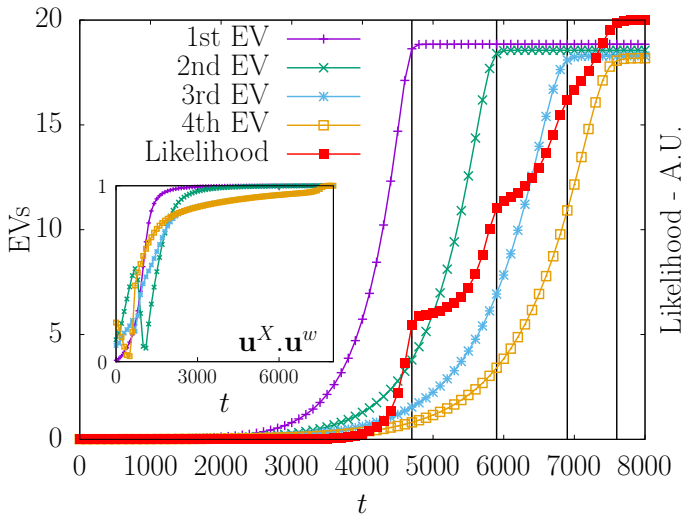


Figure 4.2: Linear RBM. Reproduced from Reprint A.

components of the data. In turn, the diagonal evolution expressed in Equation (4.45) involves only the singular values. The stable solutions of this diagonal dynamics are

$$w_\alpha^2 = \begin{cases} \frac{\langle s_\alpha^2 \rangle_{data} - \sigma_v^2}{\sigma_v^2 \sigma_h^2 \langle s_\alpha^2 \rangle_{data}} & \langle s_\alpha^2 \rangle_{data} > \sigma_v^2 \\ 0 & \langle s_\alpha^2 \rangle_{data} < \sigma_v^2 \end{cases} \tag{4.47}$$

where we see how the evolution of the singular values is driven by the SVD modes of the training data. The strongest modes, those above the threshold σ_v^2, are selected and learned while the modes below threshold are damped.

Equations (4.45)(4.46)(4.47) let us simply summarize the behavior of the RBM in the linear regime: first, the learning equations drive the spectral modes of the weight matrix to align to the principal components of the data; subsequently, the singular values of the aligned modes are selected and amplified to match the dataset. Figure 4.2 shows the behavior of a linear RBM, in agreement with our theoretical analysis.

4.7 Nonlinear regime

To derive the dynamics in the non-linear regime, we will make use of the mean-field results of Sections 4.2 and 4.4 to compute the $\langle \dots \rangle_{Data}$ (empirical) and $\langle \dots \rangle_{RBM}$ (model) averages in Equations (4.27)-(4.29). In the empirical average the value of the nodes of the network is driven by the data, so that we can directly substitute the nodes values in Equations (4.21)(4.22) to obtain

$$\langle \sigma_\alpha \rangle_{Data} = \langle w_\alpha s_\alpha (1 - q_\alpha[\boldsymbol{s}]) \rangle_{Data} \tag{4.48}$$

$$\langle s_\alpha \sigma_\alpha \rangle_{Data} = \langle s_\alpha w_\beta s_\beta (1 - q_\beta[\boldsymbol{s}]) \rangle_{Data} \tag{4.49}$$

where $q[\boldsymbol{s}]$ is the empirical counterpart of the q_α variable (4.19), here defined as

$$q[\boldsymbol{s}] \stackrel{\text{def}}{=} \int dx \frac{e^{\frac{x^2}{2}}}{\sqrt{2\pi}} d\boldsymbol{v} p_\alpha(\boldsymbol{v}) \tanh^2\left(\kappa^{-1/4}\left(\sigma x + \sum_\gamma w_\gamma s_\gamma v^\gamma\right)\right). \tag{4.50}$$

The above averages are relatively easy to compute for a given dataset, but it is important to follow the prescriptions of Section 4.4 and select for the $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ components an appropriate distribution that leads to a compositional phase. The model average $\langle \dots \rangle_{RBM}$ is instead computed by averaging the stable mean-field solutions weighted by the corresponding free energies. The mean-field partition function is thus defined as

$$Z_{Therm} \stackrel{\text{def}}{=} \sum_\omega e^{-Lf(m^\omega, \bar{m}^\omega, q^\omega, \bar{q}^\omega)} \tag{4.51}$$

where ω indexes all the solutions of the saddle point equations (4.14),(4.15). The model averages then read

$$\langle s_\alpha \rangle_{RBM} = \frac{1}{Z_{Therm}} \sum_\omega e^{-Lf(m^\omega, \bar{m}^\omega, q^\omega, \bar{q}^\omega)} \bar{m}_\alpha^\omega \stackrel{\text{def}}{=} \langle \bar{m}_\alpha \rangle_{Therm} \tag{4.52}$$

$$\langle s_\alpha \sigma_\beta \rangle_{RBM} = \frac{1}{Z_{Therm}} \sum_\omega e^{-Lf(m^\omega, \bar{m}^\omega, q^\omega, \bar{q}^\omega)} \bar{m}_\alpha^\omega m_\beta^\omega \stackrel{\text{def}}{=} \langle \bar{m}_\alpha m_\beta \rangle_{Therm} \tag{4.53}$$

and we can put together all of the above to write a set of nonlinear dynamical equations that we are able to solve numerically

$$\frac{1}{L}\frac{dw_\alpha}{dt} = \langle s_\alpha w_\alpha s_\alpha (1 - q_\alpha[\boldsymbol{s}]) \rangle_{Data} - \langle \bar{m}_\alpha w_\alpha \bar{m}_\alpha (1 - q_\alpha) \rangle_{RBM} \tag{4.54}$$

$$\frac{d\eta}{dt} = \langle \bar{m}_\alpha \rangle_{Therm} - \langle s_\alpha \rangle_{Data} + \sum_\beta \Omega^u_{\alpha\beta} \eta_\beta \tag{4.55}$$

$$\frac{d\theta_\alpha}{dt} = \langle w_\alpha \bar{m}_\alpha (1 - q_\alpha) \rangle_{Therm} - \langle w_\alpha s_\alpha (1 - q_\alpha[\boldsymbol{s}]) \rangle_{Data} + \sum_\beta \Omega^v_{\alpha\beta} \theta_\beta. \tag{4.56}$$

4.8 Empirical dynamics

To experimentally analyze the mean-field dynamics of the RBM given by Equations (4.54)-(4.56) we introduce a simple synthetic dataset, composed by C clusters of data with low dimensionality d embedded in a high dimensional space represented by N binary variables $s_i = \pm 1$, with $N \gg d$. To generate the data, we select a set of d orthonormal random vectors $\boldsymbol{b}^\alpha$ as a basis and we fix the magnetizations

$$m_i^c = \sum_{\alpha=1}^{d} m_\alpha^c b_i^\alpha \tag{4.57}$$

where the m_α^c factors are drawn at random between $[-1, 1]$ and normalized. The multimodal distribution associated to the dataset is

$$P(\boldsymbol{s}) = \sum_{c=1}^{C} p_c \prod_{i=1}^{N} \frac{e^{h_i^c s_i}}{2\cosh(h_i^c)} \tag{4.58}$$

where p_c is a probability weighting cluster c and h_i^c is determined from the magnetizations $m_i^c = \tanh h_i^c$. Samples are generated by choosing a cluster according to p_c and setting the visible variables to ± 1 according to

$$p(s_i = 1) = \frac{1}{1 + e^{-2h_i^c}}.$$

The generated dataset is used to compute the empirical averages and the Equations (4.54)-(4.56) are integrated numerically, choosing the Laplace distribution for $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ components to compute the $q, \bar{q}, q_\alpha$ terms. The resulting data-driven analytical dynamics are shown in Figure 4.3. For the singular values we

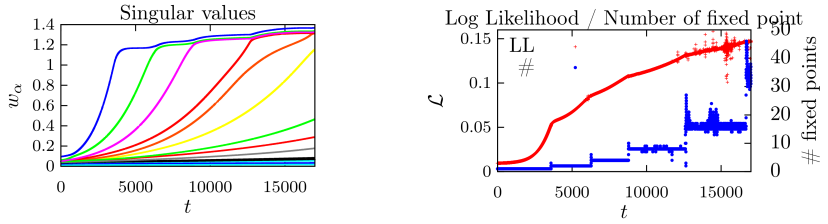


Figure 4.3: Predicted mean evolution of an RBM for a synthetic dataset, with $(N_v, N_h) = (1000, 500)$ and intrinsic dimension $d = 15$. The singular values evolution is obtained by integration of Equation (4.54).

observe how the evolution is analogous to the linear case in which the modes emerge from the bulk one by one, with the difference that while in the linear regime the modes were evolving independently, here we notice how the expressed modes are interacting, with lower modes exerting a repulsive pressure on the singular values above threshold. The number of fixed point solutions used to compute the model averages increases in steps, roughly doubling each time a new mode is expressed, and the learning trajectory on the phase diagram (Figure 4.4) clearly shows how the RBM is found in the paramagnetic phase at initialization and the learning dynamics drive the model towards the learning phase, supporting our expectations that the model works in such a phase at equilibrium.

Using the same synthetic dataset, we also performed a standard training of the RBM (using CDk, see Section 2.2) and observed the spectral dynamics shown in Figure 4.5. We see how the evolution of the singular values and the emergence of new fixed point solutions follow closely the analytical evolution described above, with the difference that the RBM training conserves a lower

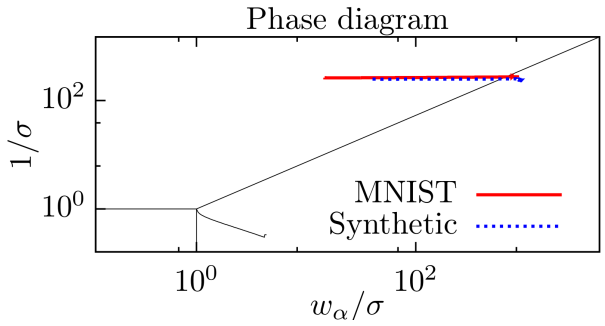


Figure 4.4: Empirical trajectories of the RBM weight matrix in the phase diagram.

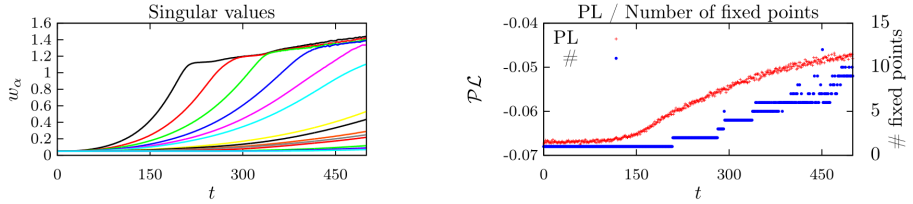


Figure 4.5: Empirical evolution of the RBM for a synthetic dataset with $(N_v, N_h) = (1000, 500)$ and intrinsic dimension $d = 15$. The behavior is comparable to the predicted evolution shown in Figure 4.3.

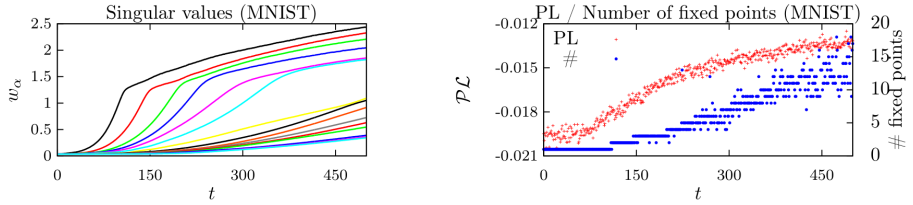


Figure 4.6: Empirical evolution of the RBM for the MNIST dataset.

number of fixed point solutions. This is probably due to the fact that the biases are expanded over an incomplete basis in Equations (4.9)(4.10), meaning that we are neglecting a residual part perpendicular to the K-modes spectral basis; this represents a limitation of our theoretical analysis but the overall behavior is well described, meaning that our mean-field model is sound.

Finally, in Figure 4.6 we show the empirical dynamics over the MNIST dataset. We see that the qualitative behavior follows the analytical prescriptions, showing that the mean-field model is well adapted to real-world scenarios. The picture of the RBM model that emerges is now rather complete. We have seen how the learning equations drive the selection of a certain number of

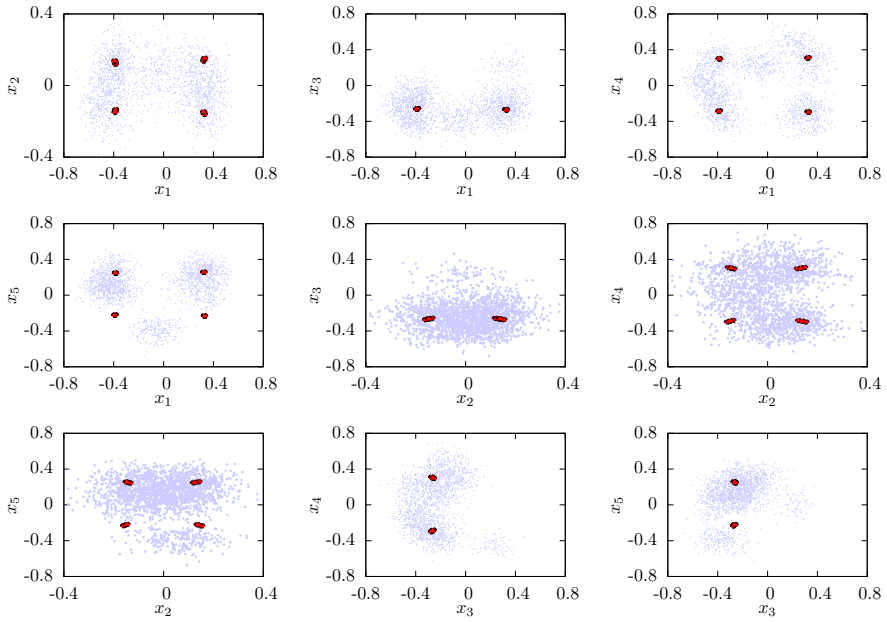


Figure 4.7: Projections of the mean-field magnetizations (in red) and samples from a synthetic dataset over the planes defined by the strongest spectral directions $x_1, x_2, \ldots x_5$.

modes, which in turn determine the generation of new fixed point solutions for the equilibrium phase. In Equation (4.54) we see how the learning converges once the model average over the fixed point solutions matches the empirical average, meaning that the fixed point magnetizations are able to accurately represent the full dataset. This is seen by plotting the fixed point magnetizations together with the data along the spectral directions. In Figure 4.8 we see that the learned RBM is able to accurately cover the data distribution with the appropriate fixed points, while the extra fixed points in the mean-field model shown in Figure 4.7 are not representative of the data. Nonetheless, the data are clustered in the spectral space and the fixed point solutions in Figure 4.8 are coherent with the convergence expectations of Equation (4.54).

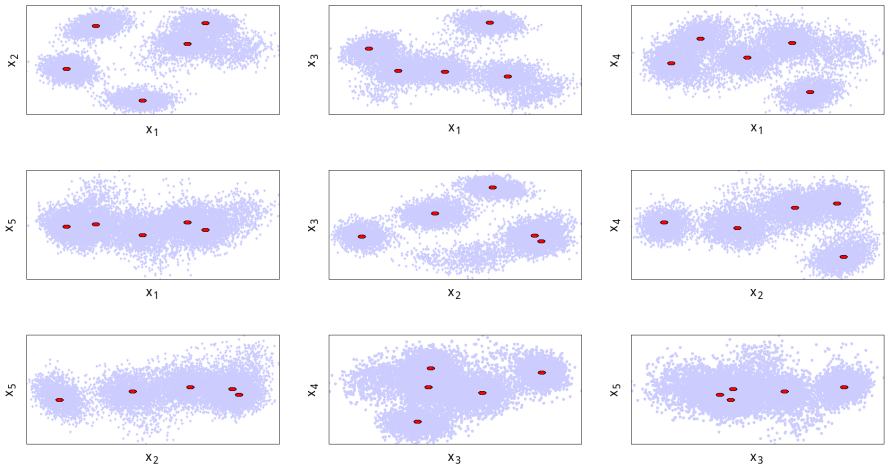


Figure 4.8: Empirical projections of the TAP fixed points of a trained RBM (in red) and samples from a synthetic dataset over the planes defined by the strongest spectral directions $x_1, x_2, \dots x_5$.

Chapter 5

Missing information

A key advantage of the RBM is that it makes it possible to efficiently compute the conditional probability of a set of variables given the other variables. This property is fundamental to define its learning strategy and it adapts naturally to the case of missing information, i.e. datasets in which the visible variables are not fully observed. In this context, of particular interest is the case of learning with missing information and missing labels. We will show that the RBM can easily be adapted to model the joint probability of features and labels of a dataset, and this modeling results beneficial in imputing missing values for both features and labels collaboratively.

In the next Section, we will introduce the problem more in detail and underline its relevance. We will then introduce our approach and discuss our results.

5.1 Multi-output learning with incomplete data

Modern machine learning techniques usually require large sets of fully observed and well labelled data for training, which are seldom available in real-world applications. Sometimes a random subset of features is absent (e.g. failed sensors of a monitoring system), sometimes the data are insufficiently annotated (limitation due to the difficulties of human annotation) meaning that a lot of labels are missing. In fact the most common situation is that we have to train a model with both missing features and labels. Machine learning models that are able to deal with missing observations and missing labels are therefore highly desired.

In our work, we will consider multi-class and multi-label learning; the former associates an input instance to one class of a finitely defined class set, while the latter allows one instance to be associated to multiple labels simultaneously. Though both the problems of data imputation and multi-output learning with semi-supervised labels have been discussed in previous works [73, 14, 52], the proposed models are generally not designed to handle both challenges concurrently. In our work we shall consider that both features and labels are missing

completely at random, meaning that the mask associated to a sample data is assumed to be stochastically independent of the data.

Recently, methods based on Deep Latent Variables Models (DLVM) have been proposed to deal with missing data. In [58], the Variational Autoencoder [46] has been adapted to be trained with missing data and a sampling algorithm for data imputation is proposed. Other approaches based on Generative Adversarial Networks (GAN) [33] are proposed in [83] and [51]. Impressive results on image datasets are displayed for these models, at the price of a rather high model complexity and the need for a large training set. In addition these works are focused on features reconstruction, and additional specifications and fine-tuning would be necessary to be able to take partially observed labels into account. The models specifications are quite involved, and any new specificity of the dataset may increase both the cost and the difficulty in training.

We chose to address the missing data problem in a more economical and robust manner. We consider the simple architecture of the Restricted Boltzmann Machine and adapt it to the multi-output learning context (RBM-MO) with missing data. The RBM-MO method serves as a generative model which collaboratively learns the marginal distribution of features and label assignments of input data instances, despite the incomplete observations. Building on the ideas expressed in [65] we adapt the approach to the PCD training procedure (Section 2.2) and we propose a mean-field imputation method. Convincing results are shown on various real-world datasets. The advantage of the RBM-MO model is that of providing a robust and flexible method to deal with missing data, with little additional complexity with respect to the classic RBM. Moreover, it works seamlessly with multi-class and multi-label tasks, providing a unified framework for multi-output learning.

5.2 Lossy-CDk

In this section, we consider an RBM model with a layer of binary hidden nodes $h_j = 0, 1$. The visible nodes are indicated by $\boldsymbol{v}$ and they will be either binary ($v_i = 0, 1$) or Gaussian, according to a prior distribution p_{prior} as described in Section 2.4, determining the following probability distribution

$$P(\boldsymbol{v}, \boldsymbol{h}) = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{Z} p_{prior}(\boldsymbol{v}). \tag{5.1}$$

To model the presence of missing values in the dataset, we separate the visible nodes in a set $\mathcal{O}$ of observed values and a set $\mathcal{M}$ of missing values. The visible layer then separates into the observed nodes $\boldsymbol{v}_o = \{v_i, i \in \mathcal{O}\}$ and the missing nodes $\boldsymbol{v}_m = \{v_i, i \in \mathcal{M}\}$. The probability distribution over the observed variables can be written as

$$P(\boldsymbol{v}_o) = \frac{Z_{\mathcal{O}}}{Z} \tag{5.2}$$

where $Z_{\mathcal{O}}$ is given by a marginalization over the missing variables

$$Z_{\mathcal{O}} = \int \prod_{i \in \mathcal{M}} p_{prior}(v_i) dv_i \\ \times e^{\sum_{k \in \mathcal{V}} a_k v_k} \prod_j \left(1 + \exp\left(\sum_{k \in \mathcal{V}} w_{kj} v_k + b_j\right)\right), \quad \mathcal{V} = \mathcal{O} \cup \mathcal{M}. \tag{5.3}$$

The log-likelihood gradient w.r.t. the weights takes the form

$$\frac{\partial \mathcal{L}(\boldsymbol{v})}{\partial w_{ij}} = \langle I_o(i) v_i \sum_{h_j} h_j p(h_j | \boldsymbol{v}_{\mathcal{O}}) \rangle_{data} \tag{5.4}$$

$$+ \langle (1 - I_o(i)) \sum_{h_j} \int dv_i v_i h_j p(h_j | \boldsymbol{v}_o) \rangle_{data} \tag{5.5}$$

$$- \langle v_i h_j \rangle_{RBM} \tag{5.6}$$

where I_o is the indicator function of the set of observed nodes $\mathcal{O}$. We note that w.r.t. the standard case the empirical average is now split in two terms, and we need to integrate over the missing variables $v_i, i \in \mathcal{M}$ to compute the conditional probability $p(h_j|\boldsymbol{v}_o)$ that we need to perform the Gibbs sampling. This integration is akin to the computation of the negative term, meaning that we can estimate the full positive term with a similar strategy: we pin the observed visible nodes to their value and we iterate the Gibbs sampling equations (Algorithm 1) for a prescribed number of steps k. The full training algorithm becomes:

Algorithm 4 Lossy-CDk (RBM training with incomplete data)

Data: a training set of N data vectors
Randomly initialize the weight matrix **W**
for t = 0 to T (# of epochs) **do**
    Divide the training set in m minibatches
    for all minibatches m **do**
     Positive term:
        pin variables $v_i, i \in \mathcal{O}$ to their observed value
        initialize $v_i, i \in \mathcal{M}$ randomly
        sample $\boldsymbol{h}, \boldsymbol{v}$ using $p(\boldsymbol{v} \mid \boldsymbol{h})$ and $p(\boldsymbol{h} \mid \boldsymbol{v})$ for k steps
        compute the positive terms in (5.4) and (5.5)
     Negative term:
        initialize $\boldsymbol{v}$ randomly
        sample $\boldsymbol{h}, \boldsymbol{v}$ using $p(\boldsymbol{v} \mid \boldsymbol{h})$ and $p(\boldsymbol{h} \mid \boldsymbol{v})$ for k steps
        compute $\langle \boldsymbol{v}\boldsymbol{h}^T \rangle_{model}$
     Full update:
        update **W** with equations (2.8) and (5.4)-(5.6)
    end for
end for

5.3 Mean-field imputation

Given a trained RBM modeling an empirical data distribution, this can be used to impute missing values for samples belonging to the same data distribution. Similarly to the Lossy-CDk training, the idea is to fix the values of the observed features and sample the missing features; fixing the known variables has the effect of biasing the sampling procedure towards the correct equilibrium configuration and helps in speeding up convergence, given that the fraction of missing features is small enough. For high percentages of missing features, we might expect this biased sampling to provide degraded solutions, as the low information given by the observed features can be supplanted by the sampling noise. To mitigate this effect, we can average multiple imputations weighted by the learned distribution. Recalling the mean-field description of the RBM in Chapter 4, we expect the equilibrium configurations to be represented by a limited number of fixed point magnetizations. Averaging values imputed from the fixed point magnetizations correlated with the observed variables provides imputations that are weighted by the model distribution (assuming the free energy differences of the fixed point solutions to be negligible), giving more reliable solutions. A set of self-consistent equations to compute the mean-field imputations is obtained by writing the mean-field equations at lowest order for the marginal probabilities of visible variables m_i and the marginal probabilities of hidden variables q_j

$$m_i = \left(\sum_j w_{ij} q_j + a_i \right) \sigma_v^2, \qquad i \notin \mathcal{O} \tag{5.7}$$

$$q_j = \sigma \left(\sum_i w_{ij} m_i + b_j \right) \tag{5.8}$$

that we can iterate to convergence while keeping fixed the observed values $v_i, i \in \mathcal{O}$. To obtain the weighted imputations, we simply run the above equations N_f times (take $N_f \sim 10$) starting from random initial conditions and average the results

$$\bar{m}_i = \frac{1}{N_f} \sum_{n=1}^{N_f} m_i^{(n)}. \tag{5.9}$$

5.4 Multi-output classification with missing information

A rather flexible approach to introduce labels in the RBM model is to simply model the joint probability distribution of both features and labels of an annotated dataset [49]. This boils down to separating the set of visible nodes $\mathcal{V}$ into a set of features nodes $\mathcal{V}_f$ and a set of label nodes $\mathcal{V}_l$ and let the model learn the correct label structure and the correlations between labels and features. The flexibility of this approach lies in the fact that multi-class and multi-label scenarios can be handled seamlessly with the same model, testing the capacity

of the RBM to properly learn the constraints on labels structure in the different cases.

The learning algorithm is thus unmodified, while the mean-field imputation includes the p_i magnetizations for label nodes giving the following set of iterative equations

$$m_i = \left(\sum_j w_{ij} q_j + a_i \right) \sigma_v^2, \qquad i \notin \mathcal{O} \tag{5.10}$$

$$p_i = \sigma \left(\sum_j w_{ij} q_j + a_i \right), \qquad i \notin \mathcal{O} \tag{5.11}$$

$$q_j = \sigma \left(\sum_{i \in \mathcal{V}_f} w_{ij} m_i + \sum_{i \in \mathcal{V}_l} w_{ij} p_i + b_j \right). \tag{5.12}$$

Qualitative results of the RBM learning with missing information are shown in Figure 5.1 for the MNIST dataset; quantitative results for classification with missing labels and missing features in the multiclass setting are reported in Table 5.1, again for the MNIST dataset. The accuracy score in the base case in which only 30% of both features and labels are missing comes close to the literature result for RBM classification with fully observed samples [49], providing a good sanity check for our proposed mean-field imputation strategy. In the extreme case in which 80% of features and labels are missing the model still provides meaningful results, testifying the robustness of the approach. Other datasets and experimental settings (the multilabel learning case in particular) are discussed in [27] (Reprint C).

Table 5.1: Classification results for the MNIST dataset with missing information. q_{ml}% is the percentage of missing labels, and q_{fea}% the percentage of missing features. The AUC score is the Area Under the ROC curve.

	Averaged AUC			Accuracy		
q_{fea}% \ q_{mc}%	30%	50%	80%	30%	50%	80%
30%	0.981	0.977	0.946	0.957	0.927	0.824
50%	0.975	0.971	0.933	0.954	0.915	0.830
80%	0.946	0.941	0.914	0.922	0.860	0.747

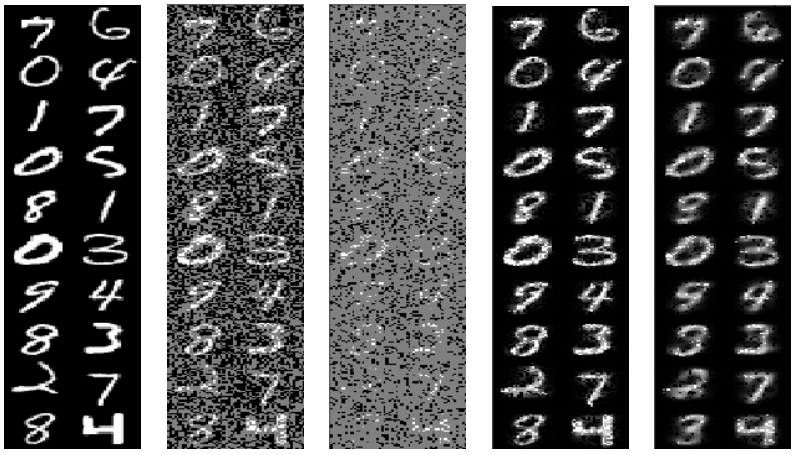


Figure 5.1: Features reconstruction by RBM-MO trained over an incomplete dataset with 50% missing-at-random features, whose classification accuracy has been measured to be around 91%. The first block shows some complete testing instances. The second and third block show the same testing instances after hiding respectively 50% and 80% of the pixels. The last two columns show the results of the mean-field imputations over the incomplete testing instances.

Part II

Normalizing Flows

Chapter 6

Relative gradient optimization

In Part I about the RBM we have seen how a generative model of the data can be expressed as an energy-based model. Here we proceed from the observation that fundamentally we introduced generative models as probability distributions that we can learn from a dataset and subsequently sample from (Section 1.8). In this Part we present a simple invertible neural network architecture to model empirical distributions, together with an efficient training algorithm. This work has been originally presented in [35] (Reprint D).

6.1 Invertible transformations of probability densities and Normalizing Flows

We proceed from the assumption that there exists a "ground truth" probability distribution that assigns a normalized density to each sample $\boldsymbol{x}$ of the dataset of interest. The samples are considered to be high dimensional, with dimension $d \gg 1$, and our goal is to learn a parametrized and tractable approximation of the distribution $P(\boldsymbol{x}) = P(\boldsymbol{x}; \boldsymbol{\theta})$. The strategy we adopt here is to consider a transformation T that maps a tractable *base distribution* $P_s(\boldsymbol{s})$ into the data distribution $P(\boldsymbol{x})$. That is, T transforms a set of samples $\boldsymbol{s}$ drawn according to P_s into the dataset samples $\boldsymbol{x}$

$$\boldsymbol{x} = T(\boldsymbol{s}), \qquad \boldsymbol{s} \sim P_s(\boldsymbol{s}).$$

Restricting the transformation T to be invertible and differentiable (with a differentiable inverse too), we can rewrite the data distribution in terms of the $\boldsymbol{s}$ samples by introducing a change of variable ([10], Section 1.2.1). This introduces the Jacobian matrix of the inverse transformation, that we denote $\boldsymbol{J}_{T^{-1}}$, and let us write the data log-likelihood as

$$\log P(\boldsymbol{x}) = \log P_s(\boldsymbol{s}) + \log |\det \boldsymbol{J}_{T^{-1}}(\boldsymbol{x})|, \qquad \boldsymbol{s} = T^{-1}(\boldsymbol{x}). \tag{6.1}$$

which we can optimize by Maximum Likelihood

$$P(\boldsymbol{x}) = P(\boldsymbol{x}; \boldsymbol{\theta}^*)$$
$$\theta^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x} \in Data} \left[\log P(\boldsymbol{x}; \boldsymbol{\theta}) \right].$$

Choosing a tractable distribution P_s, the complexity of computing and optimizing the data distribution $P(\boldsymbol{x})$ is encapsulated into the determinant of the Jacobian matrix, to which we will refer as the Jacobian term in the following. We remark that restricting the transformation T to be invertible and differentiable not only let us employ the change of variable (6.1), but it also gives us the freedom to choose a tractable distribution $P_s(\boldsymbol{s})$ without restricting the class of distributions $P(\boldsymbol{x})$ that we can model. Indeed, under mild conditions over P_s and P, Equation (6.1) can represent any distribution $P(\boldsymbol{x})$ if T is a diffeomorphism, i.e. it is continuous and both T and T^{-1} are differentiable [68]. On the other hand, requiring T to be a diffeomorphism entails that the transformation must preserve the topological properties of the input space, i.e. the spaces in which $\boldsymbol{x}$ and $\boldsymbol{s}$ are embedded will have the same topology. Concretely, this comes with two major constraints:

i) $\boldsymbol{x}$ and $\boldsymbol{s}$ have the same dimension d;

ii) the base distribution P_s and the data distribution $P(\boldsymbol{x})$ model the same number of disconnected modes.

Constraint ii) tells us that, for instance, to choose an appropriate base distribution P_s we need to know the number of separate clusters that the dataset is separated into, which is generally unknown and it is not clear how to determine it. This problem might seam daunting, but in practice we can ignore it and let the data distribution $P(\boldsymbol{x})$ connect the separate clusters with a negligible amount of density mass; in practice, the employment of a unimodal base distribution has been shown to be able to approximate as well as possible some complex input topologies ([68, 25], Figure 6.3). Constraint i) poses instead some strong limitations on the approximation capacity of the model; for instance, it is known that for transformations realized through neural networks, the ability to arbitrarily choose the model's width is necessary to achieve universal approximation capacity for arbitrary functions [53]. Notwithstanding this limitation, we recall that models employing a diffeomorphic transformation T are universal approximators for distributions, which seams to suggest that such models are powerful enough.

Another crucial property of diffeomorphic transformations T is that they are naturally composable and they present a simple expression for the Jacobian term in Equation (6.1)

$$T = T_1 \circ T_2 \implies \det \boldsymbol{J}_{T_1^{-1} \circ T_2^{-1}} = \det \boldsymbol{J}_{T_1^{-1}} \cdot \det \boldsymbol{J}_{T_2^{-1}}.$$

This let us build complex transformations of the base samples $\boldsymbol{s}$ by chaining together simpler transformations in successive steps, as exemplified in Figure

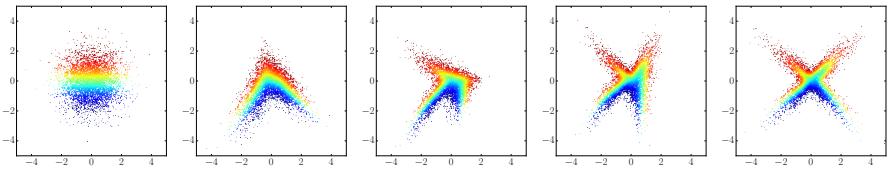


Figure 6.1: Example of a 4-step flow transforming samples from a standard-normal base density to a cross-shaped target density. Reproduced from [68].

6.1. This idea is at the base of the class of models known as *Normalizing Flows* (NF), where the name comes from the fact that we can consider the inverse transformation taking the data samples and "normalizing" them into samples of the base distribution through a "flow" of transformations.

A comprehensive account of NF models is found in [68]. The unifying aspect is that all NF models propose some kind of neural network architecture to parametrize the transformation T and optimize the log-likelihood (6.1) through backpropagation. The differences lie in the strategy that each model proposes to deal with the Jacobian term, which is generally expensive to compute. Indeed, being the Jacobian a $d \times d$ matrix, its determinant can be computed in $O(d^3)$ time and the optimization with backpropagation also takes $O(d^3)$ times. For high-dimensional datasets, it becomes quickly unfeasible to directly optimize the Jacobian term; NF models solve this problem by imposing constraints on the structure of the Jacobian matrix or by computing the Jacobian term in an approximate way, to make the optimization of the Jacobian term feasible.

6.2 Linear flows as Fully Connected neural network layers

One of the simplest transformations that we can consider in the context of Equation (6.1) is an invertible linear transformation

$$\boldsymbol{s} = \boldsymbol{W}\boldsymbol{x} \tag{6.2}$$

where $\boldsymbol{W}$ is an invertible $d \times d$ weight matrix parametrizing the inverse transformation T^{-1}. In the literature, the above transformation is known as a *linear flow* [68] and its Jacobian matrix is simply the weight matrix $\boldsymbol{W}$ itself, implying that the Jacobian term is given by the determinant of $\boldsymbol{W}$. Computation of the determinant is an $O(d^3)$ operation, as well as the computation of its derivative

$$\frac{\partial \log |\det \boldsymbol{W}|}{\partial \boldsymbol{W}} = \left(\boldsymbol{W}^T\right)^{-1} \tag{6.3}$$

where the $O(d^3)$ time complexity comes from the need of inverting the matrix $\boldsymbol{W}^T$. Maximum likelihood optimization of a linear flow with backpropagation

and Stochastic Gradient Descent (SGD) is thus a $O(d^3)$ process, as backpropagation essentially computes the gradient (6.3) with the chain rule.

A more general nonlinear transformation can be obtained by identifying a linear flow as a Fully Connected (FC) layer of a multilayer neural network. Denoting by σ the activation function of the network, the k-th layer $\boldsymbol{g}_k$ is expressed as

$$\boldsymbol{g}_k(\boldsymbol{x}) = \sigma\left(\boldsymbol{W}_k \boldsymbol{x}\right). \tag{6.4}$$

Choosing σ to be invertible and differentiable, we obtain the diffeomorphic transformation $T_k^{-1} = \boldsymbol{g}_k$. The composition of multiple FC blocks let us define a multilayer FC neural network $\boldsymbol{g} = \boldsymbol{g}_1 \circ \boldsymbol{g}_2 \circ \cdots \circ \boldsymbol{g}_L$ with

$$\boldsymbol{z}_k = \boldsymbol{g}_k(\boldsymbol{z}_{k-1}), \qquad k = 1, \dots, L \tag{6.5}$$

where L is the number of layers. The input to the network will be the data samples $\boldsymbol{x}$ and the output represents the base samples $\boldsymbol{s}$, so that we have $\boldsymbol{z}_0 = \boldsymbol{x}$ and $\boldsymbol{z}_L = \boldsymbol{s}$. Finally, for the base distribution we choose a standard multivariate normal with identity covariance, meaning that P_s factorizes over the components of $\boldsymbol{s}$ and we can write the loglikelihood of the multilayer transformation as

$$\begin{aligned} \log P(\boldsymbol{x}) &= \sum_i \log \mathcal{N}([\boldsymbol{g}(\boldsymbol{x})]_i; 0, 1) + \log \left|\det \boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x})\right| \\ &= \sum_i \log \mathcal{N}([\boldsymbol{g}(\boldsymbol{x})]_i; 0, 1) + \sum_{k=1}^{L} \log \left|\det \boldsymbol{J}_{\boldsymbol{g}_k}(\boldsymbol{z}_{k-1})\right| \end{aligned} \tag{6.6}$$

where we stress again that the computational complexity in estimating the probability density of the data lies in the computation of the Jacobian term; indeed, a forward pass through the network $\boldsymbol{g}$ has $O(d^2)$ complexity, which becomes negligible w.r.t. the $O(d^3)$ complexity of computing the Jacobian for large d. We note however that the computation of the Jacobian term has to be performed only once, and its result can be reused to compute the sample densities; in practice this mitigates the problem, as we can perform a single $O(d^3)$ operation in advance and cache it to subsequently compute sample densities in $O(d^2)$ time. Unfortunately this strategy doesn't work in the Maximum Likelihood optimization phase, in which we have to compute the gradients (6.3) at every layer for each step of the SGD update.

To obtain a better scaling behavior and make the training of linear flows feasible, we can parametrize the invertible matrix $\boldsymbol{W}$ in specific ways. [24, 47] proposed to compute the $\boldsymbol{PLU}$ decomposition of $\boldsymbol{W}$, where $\boldsymbol{P}$ is a permutation matrix (with unitary determinant) and $\boldsymbol{L}$, $\boldsymbol{U}$ are lower- and upper-triangular matrices, whose determinant is the product of the diagonal elements. Computation of the Jacobian term is thus $O(d)$, and the overall optimization is $O(d^2)$. While this parametrization is in principle able to represent any invertible matrix $\boldsymbol{W}$ [68], in practice we cannot optimize the permutation matrix $\boldsymbol{P}$, thus limiting the class of matrices that we can represent. A more flexible

alternative is to consider the $\boldsymbol{QR}$ decomposition of $\boldsymbol{W}$, where $\boldsymbol{Q}$ is an orthogonal matrix and $\boldsymbol{R}$ is upper triangular. Computing $\boldsymbol{Q}$ in full generality requires $O(d^3)$ operations, but [78] showed that we can apply the $\boldsymbol{Q}$ transformation as a sequence of at most d symmetry transformations each taking linear time. This makes it possible to compute and optimize the $\boldsymbol{QR}$ decomposition of $\boldsymbol{W}$ in $O(d^2)$ time; note however that the sequential nature of the computation makes the method unsuitable for parallel optimization, making it rather slow in practice. An experimental comparison of the performance of the $\boldsymbol{PLU}$ and $\boldsymbol{QR}$ decompositions against the direct optimization of $\boldsymbol{W}$ is found in [41].

6.3 Relative gradients

The gradient of a function is generally introduced with the implicit assumption that the objects to optimize are defined w.r.t. a Euclidean space, i.e. a n-dimensional vector space in which the notions of distance and "orientation" (angle) are well defined. Computations are thus performed in the usual Cartesian coordinate system over the real space[1] $\mathbb{R}^n$. In this context, invertible $d \times d$ matrices can be represented in the $\mathbb{R}^{d^2}$ vector space, i.e. a real space of dimension d^2. This is possible due to the isomorphism between the space of invertible matrices $\mathbb{R}^{d\times d} = \mathbb{R}^d \otimes \mathbb{R}^d$, where $\otimes$ is the Kronecker product, and the real space $\mathbb{R}^{d^2}$. However, while the isomorphism preserves the algebraic structure thus letting us perform valid computations, it doesn't guarantee that the geometric notions (such as distance) are preserved in the transformation. In the general case, invertible matrices live on a space with its own specific metric, i.e. a specification of the geometric notions of distance, angle, curvature and others. The Euclidean space has a flat metric; in dimension 3, this corresponds to the intuitive and evident properties of the physical world that we perceive, i.e. "naturally flat" surfaces and the interior angles of a triangle summing up to 180°. The space of invertible matrices is a Riemannian manifold, a smooth space that is locally equivalent to a Euclidean space but with a different metric. Riemannian manifolds are generally thought of as curved spaces; the prototypical example is the surface of a sphere, which is naturally a 2-dimensional curved space. It is well known that it is not possible to project this surface on a 2-dimensional plane without distorting distances, which motivated the use of different maps and projections to represent the Earth surface over the centuries.

In the context of optimization, the use of the *ordinary* gradient defined in the Euclidean space becomes problematic when it is used with objects defined on a Riemannian manifold, as it doesn't respect the intrinsic curvature properties of such spaces. For invertible matrices, [4] showed how the steepest direction is not represented by the ordinary gradient but by the Natural Gradient (NG), which is introduced as the gradient computed directly on the

[1] Here we refer to the n-dimensional Euclidean space and the real space $\mathbb{R}^n$ interchangeably; it is understood that we work in the real space $\mathbb{R}^n$ equipped with Euclidean structure and a Cartesian coordinate system.

manifold and not on its projection in Euclidean space. In the context of blind source separation, [16] derived a Relative Gradient (RG) for invertible matrices that stems from the properties of the multiplicative form of linear transformations (of the same kind as Equation (6.2)) and enjoys desirable convergence properties. As it turns out, the NG is identical to the RG as far as invertible matrices are concerned, with the NG being applicable in more general terms to any smooth manifold [15, 57].

For the layers of the FC network defined in Section 6.2, instead, the RG is not equivalent to the NG; consequently, in this context the RG doesn't represent the steepest direction and it doesn't enjoy the theoretical guarantees of learning efficiency that the NG does [4]. Nevertheless, the same properties that motivated the introduction of the RG are satisfied for the layers of the FC network we introduced, making it an attractive alternative to the ordinary gradient. We follow [16, 35] to introduce the RG in a simple way and show that it provides a valid descent direction for a linear transformation.

A general way to compute the gradient of a function f is to perturb its input and consider the first-order term of the resulting Taylor expansion. When the input to the function is a matrix $\boldsymbol{W}$, the perturbation will be represented by a matrix $\boldsymbol{\mathcal{E}}$ with infinitesimal entries. The ordinary gradient $\nabla f(\boldsymbol{W})$ is computed by adding the perturbation to the input

$$\boldsymbol{W} \to \boldsymbol{W} + \boldsymbol{\mathcal{E}} \tag{6.7}$$

and computing the Taylor expansion

$$f(\boldsymbol{W} + \boldsymbol{\mathcal{E}}) - f(\boldsymbol{W}) = \langle \nabla f(\boldsymbol{W}), \boldsymbol{\mathcal{E}} \rangle + o(\boldsymbol{\mathcal{E}}) \tag{6.8}$$

where $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{ij} a_{ij} b_{ij}$ is the Euclidean scalar product of matrices. The corresponding SGD update rule is obtained by aligning the update opposite to the gradient, i.e. setting $\boldsymbol{\mathcal{E}} = -\lambda \nabla f(\boldsymbol{W})$ to get

$$\boldsymbol{W} \to \boldsymbol{W} - \lambda \nabla f(\boldsymbol{W}) \tag{6.9}$$

where λ is the step size (corresponding to the learning rate in the context of neural networks training). The relative gradient $\nabla f_{rel}(\boldsymbol{W})$ is obtained by considering a multiplicative perturbation instead

$$\boldsymbol{W} \to \boldsymbol{W} + \boldsymbol{\mathcal{E}}\boldsymbol{W} = (\boldsymbol{I} + \boldsymbol{\mathcal{E}})\, \boldsymbol{W} \tag{6.10}$$

where $\boldsymbol{I}$ is the identity matrix and the Taylor expansion reads

$$f((\boldsymbol{I} + \boldsymbol{\mathcal{E}})\boldsymbol{W}) - f(\boldsymbol{W}) = \langle \nabla f(\boldsymbol{W}), \boldsymbol{\mathcal{E}}\boldsymbol{W} \rangle + o(\boldsymbol{\mathcal{E}}) = \langle \nabla f(\boldsymbol{W})\boldsymbol{W}^T, \boldsymbol{\mathcal{E}} \rangle + o(\boldsymbol{\mathcal{E}}) \tag{6.11}$$

from which we obtain the equation defining the relative gradient[2]

$$\nabla f_{rel}(\boldsymbol{W}) = \nabla f(\boldsymbol{W})\boldsymbol{W}^T. \tag{6.12}$$

[2]([35], Reprint D) calls the update term in (6.13) the relative gradient. Here we align to the original nomenclature by [16].

Considering again a perturbation opposite to the gradient $\boldsymbol{\mathcal{E}} = -\lambda \nabla f_{rel}(\boldsymbol{W})$ and applying it in a multiplicative fashion, we obtain the *relative* update rule

$$\boldsymbol{W} \to \boldsymbol{W} - \lambda \nabla f(\boldsymbol{W}) \boldsymbol{W}^T \boldsymbol{W}. \tag{6.13}$$

This update rule is valid, in the sense that it determines a decrease in the value of f. This is easily seen by taking λ small and computing the value of f for the updated value of $\boldsymbol{W}$

$$\begin{aligned} f((\boldsymbol{I} - \lambda \nabla f_{rel}(\boldsymbol{W}))\boldsymbol{W}) &= f(\boldsymbol{W} - \lambda \nabla f_{rel}(\boldsymbol{W})\boldsymbol{W}) \\ &= f(\boldsymbol{W}) - \lambda \langle \nabla f(\boldsymbol{W}), \nabla f_{rel}(\boldsymbol{W})\boldsymbol{W} \rangle + o(\lambda) \\ &= f(\boldsymbol{W}) - \lambda \langle \nabla f(\boldsymbol{W}), \nabla f(\boldsymbol{W})\boldsymbol{W}^T\boldsymbol{W} \rangle + o(\lambda) \\ &\simeq f(\boldsymbol{W}) - \lambda \langle \nabla f(\boldsymbol{W})\boldsymbol{W}^T, \nabla f(\boldsymbol{W})\boldsymbol{W}^T \rangle \\ &= f(\boldsymbol{W}) - \lambda \|\nabla f_{rel}(\boldsymbol{W})\|^2 < f(\boldsymbol{W}) \end{aligned}$$

where $\|\cdot\|^2$ is the Frobenius norm.

6.4 Relative backpropagation

The crucial advantage of the RG is that it let us simplify the expression of the derivative of the Jacobian term for linear flows. We will show how this makes it possible to optimize linear flows efficiently and in full generality, i.e. without imposing constraints on the structure of the weight matrix $\boldsymbol{W}$ as it is usually done in NF literature. Subsequently, we will consider a more complex non-linear model (the multilayer FC network introduced in Section 6.2) and show how the RG can play nicely with backpropagation, letting us introduce a Relative Backpropagation algorithm that makes it possible to efficiently optimize models employing unconstrained linear flows, with a cost that is equivalent to ordinary backpropagation. Finally, we will show some experimental results to empirically test the computational and statistical efficiency of the proposed algorithm.

Optimizing Linear Flows. Taking a linear flow $f(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x}$, we recall from Section 6.2 that the Jacobian term $\boldsymbol{D}_f$ is given by

$$\boldsymbol{D}_f(\boldsymbol{x}) = \log |\det \boldsymbol{J}_f(\boldsymbol{x})| = \log |\det \boldsymbol{W}|$$

and its gradient is

$$\nabla \boldsymbol{D}_f(\boldsymbol{W}) = \left(\boldsymbol{W}^T\right)^{-1}. \tag{6.14}$$

Exploiting Equation (6.12) we see that the RG is given by the identity matrix

$$\nabla \boldsymbol{D}_{f_{rel}} = \nabla \boldsymbol{D}_f \boldsymbol{W}^T = \boldsymbol{I} \tag{6.15}$$

which is a remarkable result as it let us avoid the computation of the gradient altogether and directly use the very simple update rule

$$\boldsymbol{W} \to (\boldsymbol{I} - \lambda)\, \boldsymbol{W} \tag{6.16}$$

where λ is the step size. This makes the optimization of the Jacobian term of linear flows computationally efficient, as we avoid the matrix inversion. Moreover, we stress the fact that this optimization strategy can be applied in full generality without imposing any constraints on the structure of the weight matrix $\boldsymbol{W}$, in stark contrast to the NF literature.

Backpropagating relative gradients. We consider the model $\boldsymbol{g}$ introduced in Section 6.2 and defined by Equations (6.4)(6.5). Defining the two objectives

$$\begin{aligned}\mathcal{L}_p(\boldsymbol{x}) &= \sum_i \log \mathcal{N}([\boldsymbol{g}(\boldsymbol{x})]_i; 0, 1)\\ \mathcal{L}_J(\boldsymbol{x}) &= \log|\det \boldsymbol{J}(\boldsymbol{x})|\end{aligned}$$

where $\mathcal{L}_J$ is the Jacobian term of the full network, the loglikelihood (6.6) can be written as

$$\mathcal{L}(\boldsymbol{x}) = \mathcal{L}_p(\boldsymbol{x}) + \mathcal{L}_J(\boldsymbol{x}). \tag{6.17}$$

We further split the Jacobian term over the layers and separate the linear transformations from the nonlinearities to get

$$\begin{aligned}\mathcal{L}_J(\boldsymbol{x}) &= \sum_{k=1}^{L} \mathcal{L}_k(\boldsymbol{z}_k)\\ \mathcal{L}_k(\boldsymbol{z}_k) &= \mathcal{L}_{k,1}(\boldsymbol{z}_k) + \mathcal{L}_{k,2}(\boldsymbol{z}_k)\\ \mathcal{L}_{k,1}(\boldsymbol{z}_k) &= \sum_{i=1}^{d} \log|[\boldsymbol{\sigma}(\boldsymbol{y}_k)]_i|\\ \mathcal{L}_{k,2}(\boldsymbol{z}_k) &= \log|\det \boldsymbol{W}_k|\end{aligned}$$

where $\boldsymbol{y}_k = \boldsymbol{W}_k \boldsymbol{z}_{k-1}$ and $\mathcal{L}_{k,2}$ is the Jacobian determinant of a linear flow.

Optimizing the full objective $\mathcal{L}(\boldsymbol{x})$ in an efficient manner is not straightforward. Indeed, with backpropagation we can calculate the ordinary gradients of each term of the loglikelihood, with the problem that the $\mathcal{L}_{k,2}$ term would be expensive to compute (Section 6.2). Substituting the ordinary gradient with the relative gradient using Equation (6.12) we can optimize $\mathcal{L}_{k,2}$ efficiently with the update rule (6.16), but the update rules for the other terms are

$$\boldsymbol{W}_k \to (\boldsymbol{I} - \lambda \nabla f(\boldsymbol{W}_k)\boldsymbol{W}_k^T)\boldsymbol{W}_k \sim O(d^3), \quad f = \{\mathcal{L}_p, \mathcal{L}_{k,1}\}$$

which are in turn expensive to compute due to the $O(d^3)$ matrix multiplications, as both $\nabla f(\boldsymbol{W}_k)$ and $\boldsymbol{W}_k$ are $d \times d$ matrices. However, this problematic is only apparent and using some care it is possible to efficiently compute the relative gradient updates for all the terms of the loglikelihood. To see this, we use the chain rule to compute the gradient of the terms $f = \{\mathcal{L}_p, \mathcal{L}_{k,1}\}$ w.r.t. the parameters of layer k, i.e. the weight matrix $\boldsymbol{W}_k$:

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{W}_k} = \frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{W}_k}\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{y}_k} = \boldsymbol{z}_{k-1}\boldsymbol{\delta}_k^T \qquad \boldsymbol{\delta}_k = \left(\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{y}_k}\right)^T$$

where $\boldsymbol{\delta}_k$ is the *error vector* [3] that is computed with automatic differentiation in the backpropagation algorithm. The relative gradient update rule then reads

$$\boldsymbol{W}_k \to \boldsymbol{W}_k - \lambda \boldsymbol{z}_{k-1}((\boldsymbol{\delta}_k^T \boldsymbol{W}_k^T)\boldsymbol{W}_k) \sim O(d^2) \tag{6.18}$$

which can be computed in $O(d^2)$ time by carefully avoiding the matrix-matix multiplications (parentheses are important in Equation (6.18)). The full objective $\mathcal{L}(\boldsymbol{x})$ can thus be efficiently optimized in $O(d^2)$ time with a hybrid approach: the $\mathcal{L}_{k,2}$ term is directly optimized with the update rule (6.16), while for $\mathcal{L}_p$ and $\mathcal{L}_{k,1}$ we first compute the backpropagated errors $\boldsymbol{\delta}_k$ with automatic differentiation and then apply the update rule (6.18).

Experimental results. To test the computational efficiency and the empirical capacity of the model proposed in Section 6.2, we performed a couple of experiments:

i) a benchmark comparing optimization through the ordinary and relative gradients;

ii) probability density estimation over some simple 2-dimensional toy datasets.

In both cases, we used a model $\boldsymbol{g} = \boldsymbol{g}_1 \circ \boldsymbol{g}_2 \circ \cdots \circ \boldsymbol{g}_L$ with L layers comprising the composition of a linear flow (6.2) and an invertible nonlinearity $\boldsymbol{\sigma}_k$ defined as a smooth alternative to the Leaky ReLU function [54] commonly used in neural network architectures

$$\sigma_k(x) = \alpha x + (1-\alpha)\log(1+e^x)$$

where α is a leakage coefficient that we treat as a hyperparameter. We note that the above expression is not invertible in closed form; in practice, this doesn't represent a problem as we can invert it numerically with the Newton method in a fixed number of iterations. For the last network layer we set $\boldsymbol{\sigma}_L(\boldsymbol{x}) = \boldsymbol{x}$, in an attempt to help the network to more easily map the outputs to a centered Normal distribution (the motivation comes from noticing that using the smooth Leaky ReLU defined above, we have $\sigma_k(0) > 0$).

The results of experiment i) are summarized in Figure 6.2. Even though the theoretical asymptotic difference between $O(d^3)$ complexity for the ordinary gradient and $O(d^2)$ complexity of the relative gradient is not clearly identified, the relative gradient optimization shows to be two order of magnitudes faster at both low and high dimensionality. We report benchmarking results for dimensionality up to $d \sim 10^4$; for higher dimensionality, the real bottleneck becomes the number of parameters required by a FC layer, and the associated memory requirements.

Experiments ii) consists in training our model over data sampled from some simple 2-dimensional distributions and visualizing the learned probability densities as a heatmap. In Figure 6.3 we can see that the model achieves very

[3] The derivative to compute $\boldsymbol{\delta}_k$ is written in numerator layout notation; the derivatives w.r.t. the matrix $\boldsymbol{W}_k$ are naturally arranged in the column vector $\boldsymbol{z}_{k-1}$.

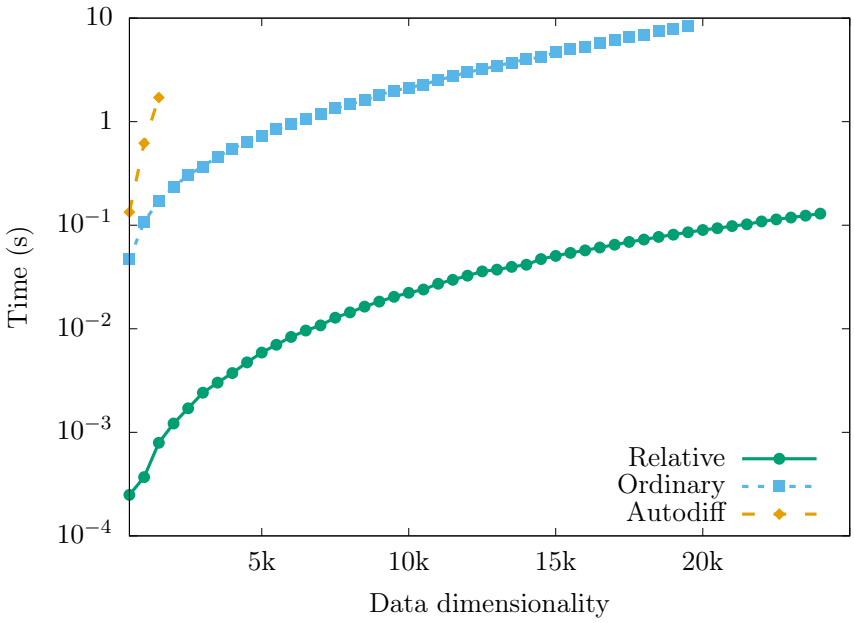


Figure 6.2: Comparison of the average computation times of a single evaluation of the gradient of the log-likelihood over a batch of 100 samples; the standard error of the mean is not reported as it is orders of magnitude smaller than the scale of the plot. We set the number of layers to $L = 2$ and performed the experiment using a Tesla P100 Nvidia GPU. Reproduced from [35].

good results in this simple setting. We remark that the original distributions to model are composed by disconnected clusters, and we know from Section 6.2 that these cannot be mapped to a unimodal Normal distribution by smooth invertible transformations. In spite of this, we see how the model is able to seemingly connect the cluster with a negligible amount of density mass.

Further experimental results over real-world datasets are reported in the original work ([35], Reprint D).

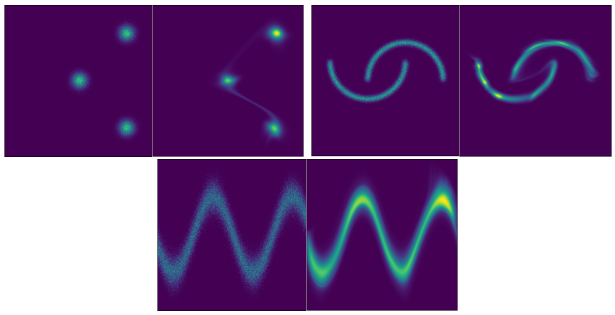


Figure 6.3: Illustrative examples of 2-dimensional density estimation. Samples from the true distribution and predicted densities are shown, in this order, side by side.

Conclusions

In Part I we have presented an extensive empirical analysis of the RBM during training, with particular focus on the dynamics of the SVD components of the weight matrix. We observed how the learning procedure selects a finite number of spectral modes to amplify, and a linear analysis of this process elucidates the role of the data in driving this selection. These observations motivated the choice of a statistical ensemble for the weight matrix of the RBM composed of a structured part and a random part. Mean-field analysis of the chosen ensemble let us derive a set of deterministic equations to describe both the dynamical evolution of the RBM during training and its equilibrium properties. The main outcome of this mean-field analysis is a clustering interpretation of the learning process, showing how the RBM is able to cluster the data around the solutions of its mean-field equations. While the mean-field equations showed some shortcomings in properly reproducing this clustering, a trained RBM fit this interpretation effectively.

When dealing with missing information, the simple formulation of the RBM naturally adapts to imputation problems. The clustering interpretation of the learning procedure motivated an imputation strategy in which multiple mean-field fixed-point solutions are averaged to obtain more robust results. This imputation algorithm is economical and successful in practice. A distinctive feature is its applicability to a wide range of different datasets including moderately low-dimensional data with only few hundred samples in the training set, as exemplified by an Internet-of-Things dataset discussed in [27] (Reprint C). Data-hungry deep learning models would most probably struggle in such a scenario. These considerations suggest that the proposed RBM-based imputation algorithm is a convenient alternative in real-world scenarios.

A couple of remarks, which are not discussed in this thesis, provide interesting avenues for further research. First, we note that a characterization of the free energy landscape of the RBM could in principle provide more refined imputations. The rationale is that the free energy of the mean-field fixed-point solutions can be used to weight the imputations over which we average; for simplicity, we disregarded this aspect in our work. Second, splitting the data features in observed and missing variables sets is akin to introducing a second layer of hidden units in the RBM architecture. From this point of view, with fully observed data and a RBM with 2 layers of hidden units the mean field imputation strategy can be analyzed as a training algorithm in its own right.

In Part II we have introduced the relative gradient and shown how it can be used to train Linear Flows without trading off expressive capacity for computational efficiency. We further proposed the use of Linear Flows as building blocks of a nonlinear multilayer neural network, and shown how the relative gradient plays nicely with standard backpropagation to provide an efficient training algorithm for the full architecture. The novelty of the proposed model is that it can scale to very high dimensionality without the need to impose structural constraints in the architecture or employing approximations during training, providing a new flexible model for density estimation. This feature makes it interesting from the theoretical point of view as it suggests a broader field of applications. In particular, the common "Gaussianization" operated by Normalizing Flows is conceptually close to Independent Component Analysis (ICA). In our work we took inspiration from the ICA literature in which the relative gradient is widely applied; inverting point of view, an interesting research proposition is to study how and if our model can be applied to perform ICA.

To conclude we note how the main subjects of this thesis have been combined before [34], suggesting that the investigation of the interplay between Restricted Boltzmann Machines and Normalizing Flows can be a fruitful one.

Reprints

Reprint A

Spectral dynamics of learning in restricted Boltzmann machines

Spectral Dynamics of Learning Restricted Boltzmann Machines

A. Decelle[1] [(a)], G. Fissore[1,2] and C. Furtlehner[2]

[1] *LRI, AO team, Bât 660 Université Paris Sud, Orsay Cedex 91405*
[2] *Inria Saclay - Tau team, Bât 660 Université Paris Sud, Orsay Cedex 91405*

arXiv:1708.02917v2 [cond-mat.dis-nn] 30 Nov 2017

PACS 02.70.Hm – Spectral methods
PACS 02.30.Zz – Inverse problems
PACS 89.75.-k – Complex systems

Abstract – The Restricted Boltzmann Machine (RBM), an important tool used in machine learning in particular for unsupervized learning tasks, is investigated from the perspective of its spectral properties. Starting from empirical observations, we propose a generic statistical ensemble for the weight matrix of the RBM and characterize its mean evolution. This let us show how in the linear regime, in which the RBM is found to operate at the beginning of the training, the statistical properties of the data drive the selection of the unstable modes of the weight matrix. A set of equations characterizing the non-linear regime is then derived, unveiling in some way how the selected modes interact in later stages of the learning procedure and defining a deterministic learning curve for the RBM.

Introduction. – A Restricted Boltzmann machine (RBM) [1] constitutes nowadays a common tool on the shelf of machine learning practitioners. It is a generative model, in the sense that it defines a probability distribution, which can be learned to approximate any distribution of data points living in some N-dimensional space, with N potentially large. It also often constitutes a building block of more complex neural network models [2, 3]. The standard learning procedure called contrastive divergence [4] is well documented [5] although being still a not so well understood fine empirical art, with many hyperparameters to tune without much guidelines. At the same time an RBM can be regarded as a statistical physics model, being defined as a Boltzmann distribution with pairwise interactions on a bipartite graph. Similar models have been already the subject of many studies in the 80's [6–9] which mainly concentrated on the learning capacity, i.e. the number of independent patterns that could be stored in such a model. The second life of neural networks has renewed the interest of statistical physicists for such models. Recent works actually propose to exploit its statistical physics formulation to define mean-field based learning methods using TAP equations [10–12]. Meanwhile some analysis of its static properties, assuming a given learned weight matrix W, have been proposed [13, 14] in order to understand collective phenomena in the latent representation [15], i.e. the way latent variables organize themselves to represent actual data. One common assumption made in these works is that the weights of W are i.i.d. which as we shall see is unrealistic. Concerning the learning procedure of neural networks, many recent statistical physics based analysis have been proposed, most of them within teacher-student setting [16] which imposes a strong assumption on the data, namely that these are generated from a model belonging to the parametric family of interest, hiding as a consequence the role played by the data themselves in the procedure. From the analysis of related models [17, 18], it is already a well established fact that a selection of the most important modes of the singular value decomposition (SVD) of the data is performed in the linear case. In fact in the simpler context of linear feed-forward models the learning dynamics can be fully characterized by means of the SVD of the data matrix [19], showing in particular the emergence of each mode by order of importance regarding singular values.

In this work we follow this guideline in the context of a general RBM. We propose to characterize both the learned RBM and the learning process itself by the SVD spectrum of the weight matrix in order to isolate the information content of an RBM. This allows us then to write a deterministic learning equation leaving aside the fluctuations. This equation is subsequently analyzed first in the linear regime to identify the unstable deformation modes of W; secondly at equilibrium assuming the learning is converg-

[(a)] E-mail: aurelien.decelle@lri.fr

ing, in order to understand the nature of the non-linear interactions between these modes and how these are determined from the input data. In the first section we recall the RBM model and associated learning algorithm. In the second section we show how this algorithm can be described by a generic learning equation. Then we first analyze the linear regime and thereafter we describe what happens with the binary RBM. A set of dynamical parameters is shown to emerge naturally from the SVD decomposition of the weight matrix. The convergence toward equilibrium is analyzed and illustrated later with actual tests on the MNIST dataset.

The RBM and associated learning procedure. – An RBM is a Markov random field with pairwise interactions defined on a bipartite graph formed by two layers of non-interacting variables: the visible nodes and the hidden nodes representing respectively data configurations and latent representations. The former noted $\boldsymbol{s} = \{s_i, i = 1 \dots N_v\}$ correspond to explicit representations of the data while the latter noted $\boldsymbol{\sigma} = \{\sigma_j, j = 1 \dots N_h\}$ are there to build arbitrary dependencies among the visible units. They play the role of an interacting field among visible nodes. Usually the nodes are binary-valued (of boolean type or Bernoulli distributed) but gaussian distributions or more broadly arbitrary distributions on real-valued bounded support are also used [20], ultimately making RBMs adapt for more heterogeneous data sets. Here to simplify we assume that visible and hidden nodes will be taken as binary variables $s_i, \sigma_j \in \{-1, 1\}$ (using ± 1 values has the advantage of symmetrizing the equations hence avoiding to deal with "hidden" biases on the variables when considering binary $\{0, 1\}$ variables). Like the Hopfield model [6] which can actually be cast into an RBM [21] an energy function is defined for a configuration of nodes

$$E(\boldsymbol{s}, \boldsymbol{\sigma}) = -\sum_{i,j} s_i w_{ij} \sigma_j - \sum_{i=1}^{N_v} \eta_i s_i - \sum_{j=1}^{N_h} \theta_j \sigma_j \qquad (1)$$

and this is exploited to define a joint distribution between visible and hidden units, namely the Boltzmann distribution

$$p(\boldsymbol{s}, \boldsymbol{\sigma}) = \frac{e^{-E(\boldsymbol{s},\boldsymbol{\sigma})}}{Z} \qquad (2)$$

where W is the weight matrix and $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ are biases, or external fields on the variables. $Z = \sum_{\boldsymbol{s},\boldsymbol{\sigma}} e^{-E(\boldsymbol{s},\boldsymbol{\sigma})}$ is the partition function of the system. The joint distribution between visible variables is then obtained by summing over hidden ones. In this context, learning the parameters of the RBM means that, given a dataset of M samples composed of N_v variables, we ought to infer values to W, $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ such that new generated data obtained by sampling this distribution should be similar to the input data. The general method to infer the parameters is to maximize the likelihood of the model, where the pdf (2) has first been summed over the hidden variables

$$\mathcal{L} = \sum_j \log(2\cosh(\sum_i w_{ij} s_i + \theta_j)) - \log(Z). \qquad (3)$$

Different methods of learning have been set up and proven to work efficiently, in particular the contrastive divergence (CD) algorithm from Hinton [4] and more recently TAP based learning [10]. They all correspond to expressing the gradient ascent on the likelihood as

$$\Delta w_{ij} = \gamma \left(\langle s_i \sigma_j p(\sigma_j | \boldsymbol{s}) \rangle_{\text{Data}} - \langle s_i \sigma_j \rangle_{p_{\text{RBM}}} \right) \qquad (4)$$

where γ is the learning rate. Similar equations can be derived for the biases. The main problem is the second term on the rhs of (4) which is not tractable, and various methods basically differ in their way of estimating this term (Monte-Carlo chains, mean field, TAP ...). For an efficient learning the first term also has to be approximated by making use of random mini batches of data at each step.

Deterministic dynamics of the learning. – In order to understand the dynamics of the learning we first project the CD equation (4) onto the basis defined by the SVD of W. As a generalization of eigenmodes decomposition to rectangular matrices, the SVD for a RBM is given by

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \qquad (5)$$

where $\mathbf{U}$ is an orthogonal $N_v \times N_h$ matrix whose columns are the left singular vectors $\mathbf{u}^\alpha$, $\mathbf{V}$ is an orthogonal $N_h \times N_h$ matrix whose columns are the right singular vectors $\mathbf{v}^\alpha$ and $\mathbf{\Sigma}$ is a diagonal matrix whose elements are the singular values w_α. The separation into left and right singular vectors is due to the rectangular nature of the decomposed matrix, and the similarity with eigenmodes decomposition is revealed by the following SVD equations

$$\mathbf{W}\mathbf{v}^\alpha = w_\alpha \mathbf{u}^\alpha$$
$$\mathbf{W}^T\mathbf{u}^\alpha = w_\alpha \mathbf{v}^\alpha$$

We consider the usual situation where $N_h < N_v$, which means that the rank of W is at most N_h. $W(t)$ represents the learned weight matrix at time t. Let $\{w_\alpha(t) \in [0, +\infty[\}$, $\{u_\alpha(t) \in \mathbb{R}^{N_v}\}$ and $\{v_\alpha(t) \in \mathbb{R}^{N_h}\}$ such that the following decomposition $w_{ij}(t) = \sum_\alpha u_i^\alpha(t) w_\alpha(t) v_j^\alpha(t)$ holds. Discarding stochastic fluctuations usually inherent to the learning procedure and letting the learning rate $\gamma \to 0$, the continuous version of (4) can be recast as follows:

A. Spectral dynamics of learning in restricted Boltzmann machines

$$\left(\frac{dw}{dt}\right)_{\alpha\beta} = \delta_{\alpha,\beta}\frac{dw_\alpha}{dt}(t) + (1-\delta_{\alpha,\beta})\Big(w_\beta(t)\Omega^v_{\beta\alpha}(t) + w_\alpha(t)\Omega^h_{\alpha\beta}\Big) = \langle s_\alpha\sigma_\beta\rangle_{\rm Data} - \langle s_\alpha\sigma_\beta\rangle_{\rm RBM} \tag{6}$$

$$\Omega^v_{\alpha\beta}(t) = -\Omega^v_{\beta\alpha} \stackrel{\rm def}{=} \frac{d\boldsymbol{u}^{\alpha,T}}{dt}\boldsymbol{u}^\beta = \frac{-1}{w_\alpha + w_\beta}\left(\frac{dw}{dt}\right)^{\rm A}_{\alpha\beta} + \frac{1}{w_\alpha - w_\beta}\left(\frac{dw}{dt}\right)^{\rm S}_{\alpha\beta} \tag{7}$$

$$\Omega^h_{\alpha\beta}(t) = -\Omega^h_{\beta\alpha} \stackrel{\rm def}{=} \frac{d\boldsymbol{v}^{\alpha,T}}{dt}\boldsymbol{v}^\beta = \frac{1}{w_\alpha + w_\beta}\left(\frac{dw}{dt}\right)^{\rm A}_{\alpha\beta} + \frac{1}{w_\alpha - w_\beta}\left(\frac{dw}{dt}\right)^{\rm S}_{\alpha\beta} \tag{8}$$

Here everything is expressed in the reference frame defined by singular vectors of W. $s_\alpha = \sum_i u^\alpha_i s_i$ and $\sigma_\alpha = \sum_j v^\alpha_j \sigma_j$ represent spin configurations in this frame. Note that one has to keep track of the original reference frame to be able to evaluate the data and RBM average in particular when the basic variables are discrete. We have introduced the skew-symmetric rotation generators $\Omega^{v,h}_{\alpha\beta}(t)$ of the basis vectors induced by the dynamics. These tell us how the data rotate relatively to this frame. The superscript S,A indicate the symmetric (resp. antisymmetric) part of the matrix. Note that these equations become singular when some degeneracy occurs in W because then the SVD is not uniquely defined. This is not really a problem since we are interested in rotations among non-degenerate modes, the rest corresponding to gauge degrees of freedom. Similar equations can be derived for the fields $\eta_\alpha(t) \stackrel{\rm def}{=} \sum_i \eta_i(t) u^\alpha_i(t)$ and $\theta_\alpha(t) \stackrel{\rm def}{=} \sum_j v^\alpha_j(t)\theta_j(t)$ projected onto the SVD modes. At this point we make the assumption that the learning dynamics is represented by a trajectory of $(\{w_\alpha(t), \eta_\alpha(t), \theta_\alpha(t), \Omega^{v,h}_{\alpha\beta}(t)\}$, while the specific realization of the u^α_i and v^α_j is considered to be irrelevant, and can be averaged out with respect to some simple distributions, as long as this average is correlated with the data. This means that the decomposition $\hat{s}_\alpha = \sum_i u^\alpha_i \hat{s}_i$ of any given sample configuration is assumed also to be kept fixed while averaging. What matters mainly is the strength given by $w_\alpha(t)$ and the rotation given by $\Omega^{v,h}_{\alpha\beta}(t)$ of these SVD modes. Assuming for example i.i.d centered normal distribution with respective variance $1/N_v$ and $1/N_h$ for u^α_i and v^α_j, the empirical term takes the simple form:

$$\langle s_\alpha\sigma_\beta\rangle_{\rm Data} = \frac{1}{N_h}\Big\langle s_\alpha(s_\beta w_\beta - \theta_\beta)V\Big(\frac{1}{N_h}\sum_\gamma (w_\gamma s_\gamma - \theta_\gamma)^2\Big)\Big\rangle_{\rm Data} \text{ where } V(x) = \int dy \frac{e^{-y^2/2}}{\sqrt{2\pi}}\,\mathrm{sech}^2(\sqrt{x}y), \tag{9}$$

which actually depends on the activation function (an hyperbolic tangent in this case). The main point here is that the empirical term defines an operator whose decomposition onto the SVD modes of W functionally depends solely on w_α, θ_α and on the projection of the data on the SVD modes of W. This term is precisely driving the dynamics. The adaptation of the RBM to this driving force is given by the second term which can be as well estimated in the thermodynamic limit, as a function of w_α, θ_α and η_α alone.

Linear instabilities. – First let us consider the linear regime which can be analyzed thoroughly. It can be obtained by rescaling all the weights and fields by a common "inverse temperature" β factor and let this go to zero in equations (6). This limit can be understood by keeping up to quadratic terms in the mean field free energy and should correspond to the first stages of the learning. In this limit, magnetizations (μ_v, μ_h) of visible and hidden variables have Gaussian fluctuations with covariance matrix

$$C(\mu_v,\mu_h) \stackrel{\rm def}{=} \begin{bmatrix} \sigma_v^{-2} & -W \\ -W^T & \sigma_h^{-2}\end{bmatrix}^{-1}$$

with $\sigma_v^2 = \sigma_h^2 = 1$ introduced for sake of generality when considering general linear RBM. To simplify the exposition, we discard the biases of the data and related fields $(\theta_\alpha, \eta_\alpha)$ of the RBM. In that case the empirical term in (6) involves directly the covariance matrix of the data expressed in the frame defined by the SVD modes of W

$$\langle s_\alpha\sigma_\beta\rangle_{\rm Data} = \sigma_h^2 w_\beta \langle s_\alpha s_\beta\rangle_{\rm Data}.$$

From $C(\mu_v,\mu_h)$ we get the other terms yielding the following equations:

$$\frac{dw_\alpha}{dt} = w_\alpha\sigma_h^2\Big(\langle s^2_\alpha\rangle_{\rm Data} - \frac{\sigma_v^2}{1-\sigma_v^2\sigma_h^2 w_\alpha^2}\Big)$$

$$\Omega^{v,h}_{\alpha\beta} = (1-\delta_{\alpha\beta})\sigma_h^2\Big(\frac{w_\beta - w_\alpha}{w_\alpha + w_\beta} \mp \frac{w_\beta + w_\alpha}{w_\alpha - w_\beta}\Big)\langle s_\alpha s_\beta\rangle_{\rm Data}$$

Note that these equations are exact for a linear RBM, since they can be derived without any reference to the coordinates of u_α and v_α over which we average in the non-linear regime. These equations tell us that, during the learning the vectors $\boldsymbol{u}^\alpha$ (and also $\boldsymbol{v}^\alpha$) will rotate until being aligned to the the principal components of the data, i.e. until $\langle s_\alpha s_\beta\rangle_{\rm Data}$ becomes diagonal. Then calling $\hat{w}^2_\alpha$ the corresponding empirical variance given by the data,

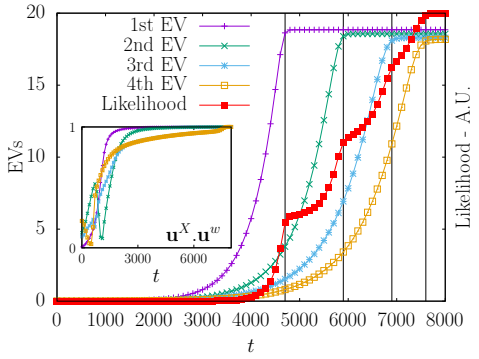


Fig. 1: Time evolution of the eigenvalues in the linear model and of the likelihood. We observe very clearly how the different modes emerge from the bulk and how the likelihood increases at each eigenvalue learned. In the inset, the scalar product of the vectors $\boldsymbol{u}$ obtained from the SVD of the data and of $\boldsymbol{w}$. The $\boldsymbol{u}$s of $\boldsymbol{w}$ are aligned with the SVD of the data at the end of the learning.

the system reach the following equilibrium values:

$$w_\alpha^2 = \begin{cases} \dfrac{\hat{w}_\alpha^2 - \sigma_v^2}{\sigma_v^2 \sigma_h^2 \hat{w}_\alpha^2} & \text{if} \quad \hat{w}_\alpha^2 > \sigma_v^2, \\ 0 & \text{if} \quad \hat{w}_\alpha^2 \leq \sigma_v^2. \end{cases}$$

From this we see that the RBM selects the strongest SVD modes in the data. The linear instabilities correspond to directions for which the variance of the data is above the threshold σ_v^2. This determines the deformations of the weight matrix which can develop during the learning and will eventually interact, following the usual mechanism of non-linear pattern formation like e.g. in reaction-diffusion processes [22]. Other possible deformations are damped to zero. The linear RBM will therefore learn all (up to N_h) principal components that passed the threshold but it is important to remember that the resulting distribution will still be unimodal. Note that this selection mechanism is already known to occur for linear auto-encoders [18] or some other similar linear Boltzmann machines [17]. On Fig. 1 we can see the eigenvalues being learned one by one in a linear RBM. For non-linear RBM when the system escapes the linear regime, a well suited mean-field theory is required to understand the dynamics and the steady-state regime.

Non-linear regime. – During the linear regime some specific modes are selected and at some point these modes start to interact in a non-trivial manner. The empirical term in (6) involves higher order statistics of the data as exemplified by (9) and the Gaussian estimation with $\sigma_v^2 = \sigma_h^2 = 1$ of the RBM response term $\langle s_\alpha \sigma_\beta \rangle_{\text{RBM}}$ is no longer valid. In order to estimate this term in the thermodynamic limit, some assumptions on the form of the weight matrix are needed. A common assumption consists in considering i.i.d. random variables for the weights w_{ij} and this, like for example in [13–15], generally leads to a Marchenko-Pastur distribution of the singular values of W, which as we shall see in the next section is unrealistic. Instead, based on our experiments such distribution corresponds to the noise of the weight matrix, while its information content is better expressed by the presence of SVD modes outside of the bulk. This leads us to write the weight matrix as

$$w_{ij} = \sum_{\alpha=1}^{K} w_\alpha u_i^\alpha v_j^\alpha + r_{ij} \tag{10}$$

where the $w_\alpha = O(1)$ are isolated singular values (describing a rank K matrix), the $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ are the eigenvectors of the SVD decomposition and the $r_{ij} = \mathcal{N}(0, \sigma^2/L)$ where $L = \sqrt{N_h N_v}$ are i.i.d. corresponding to noise. To be consistent with the linear analysis, these modes are assumed to span the (left) subspace corresponding to the part of the empirical SVD above threshold while r spans the complementary space of empirical modes below threshold. We limit the analysis here to the case where K is finite. This then allows us to assume simple distributions p_u and p_v for the components of $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ considered i.i.d. for instance. This altogether defines our statistical ensemble of RBM to which we restrict ourselves to study the learning procedure. For K extensive we should instead average over the orthogonal group which would lead to a slightly different mean-field theory [23, 24]. In the present form our model of RBM is similar to the Hopfield model and recent generalizations [25], the patterns being represented by the SVD modes outside the bulk. The main difference, in addition to the bipartite structure of the graph, is the non-degeneracy of the singular values w_α. Still the analysis in the thermodynamic limit follows classical treatments like [7,26] for the Hopfield model or [14] for bipartite models. The starting point is to express the average over u, v and weights r_{ij} of the log partition function Z in (2) with the help of the replica trick:

$$\mathsf{E}_{u,v,r}[\log(Z)] = \lim_{p \to 0} \frac{d}{dp} \mathsf{E}_{u,v,r}[Z^p].$$

After averaging over the iid weights, 4 sets of order parameters $\{(m_\alpha^a, \bar{m}_\alpha^a), a = 1, \dots p, \alpha = 1, \dots K\}$ and $\{(Q_{ab}, \bar{Q}_{ab}), a, b = 1, \dots p, a \neq b\}$ are introduced with help of two distinct Hubbard-Stratonovich transformations. These variables represent the following quantities:

$$m_\alpha^a \sim \frac{1}{\sqrt{L}} E_{u,v,r}\big(\langle \sigma_\alpha^a \rangle\big) \qquad \bar{m}_\alpha^a \sim \frac{1}{\sqrt{L}} E_{u,v,r}\big(\langle s_\alpha^a \rangle\big)$$

$$Q_{ab} \sim E_{u,v,r}\big(\langle \sigma_i^a \sigma_i^b \rangle\big) \qquad \bar{Q}_{ab} \sim E_{u,v,r}\big(\langle s_j^a s_j^b \rangle\big),$$

namely the correlations of the hidden [resp. visible] states with the left [resp. right] singular vectors and the Edward-Anderson order parameters measuring the correlation between replicas of hidden or visible states. E_u and E_v denote an average wrt to the rescaled components $u \simeq \sqrt{N_v} u_i^\alpha$

A. Spectral dynamics of learning in restricted Boltzmann machines

and $v \simeq \sqrt{N_h} v_j^\alpha$ of the SVD modes. The transformations involve pairs of complex integration variables because of the asymmetry introduced by the two-layers structure by contrast to fully connected models. They lead to the following representation:

$$\mathsf{E}_{u,v,r}[Z^p] = \int \prod_{a,\alpha} \frac{dm_\alpha^a d\bar{m}_\alpha^a}{2\pi} \prod_{a\neq b} \frac{dQ_{ab} d\bar{Q}_{ab}}{2\pi} \exp\Big\{ -L\Big(\sum_{a,\alpha} w_\alpha m_\alpha \bar{m}_\alpha + \frac{\sigma^2}{2} \sum_{a\neq b} Q_{ab}\bar{Q}_{ab} - \frac{1}{\sqrt{\kappa}} A[m,Q] - \sqrt{\kappa} B[\bar{m},\bar{Q}] \Big) \Big\}$$

$$\text{with} \qquad A[m,Q] \stackrel{\text{def}}{=} \log\Big[\sum_{S^a\in\{-1,1\}} \mathsf{E}_u \Big(e^{\frac{\sqrt{\kappa}\sigma^2}{2} \sum_{a\neq b} Q_{ab} S^a S^b + \kappa^{\frac{1}{4}} \sum_{a,\alpha} (m_\alpha^a w_\alpha - \eta_\alpha) u^\alpha S^a} \Big) \Big],$$

$\kappa = N_h/N_v$ and $B[\bar{m},\bar{Q}]$ obtained from $A[m,Q]$ by replacing u by v, η by θ and κ by $1/\kappa$. The thermodynamic properties are obtained by first letting $L \to \infty$ allowing for a saddle point approximation and then the limit $p \to 0$ is taken. We restrict here the discussion to replica symmetric (RS) saddle points [27]. The breakdown of RS can actually be determined by computing the so-called AT line [28] and will be detailed somewhere else [29]. In the RS case the set $\{(Q_{ab}, \bar{Q}_{ab}\}$ reduces to a pair $(q,\bar{q})$ of spin glass parameters, while quenched magnetization towards the SVD directions are now represented by $\{(m_\alpha, \bar{m}_\alpha), \alpha = 1, \dots K\}$. Letting $x = \mathcal{N}(0,1)$ and skipping some details, the saddle-point equations are given by

$$(m_\alpha, \bar{m}_\alpha) = \mathsf{E}\Big(\kappa^{\frac{1}{4}} v^\alpha \tanh\big(\bar{h}(x,v)\big), \kappa^{-\frac{1}{4}} u^\alpha \tanh\big(h(x,u)\big) \Big) \tag{11}$$

$$(q,\bar{q}) = \mathsf{E}\Big(\tanh^2\big(\bar{h}(x,v)\big), \tanh^2\big(h(x,u)\big) \Big), \tag{12}$$

with E denoting the average over (u,v,x) and

$$h(x,u) \stackrel{\text{def}}{=} \kappa^{\frac{1}{4}} \big(\sigma\sqrt{q}x + \sum_\gamma (w_\gamma m_\gamma - \eta_\gamma) u^\gamma \big)$$

$$\bar{h}(x,v) \stackrel{\text{def}}{=} \kappa^{-\frac{1}{4}} \big(\sigma\sqrt{\bar{q}}x + \sum_\gamma (w_\gamma \bar{m}_\gamma - \theta_\gamma) v^\gamma \big).$$

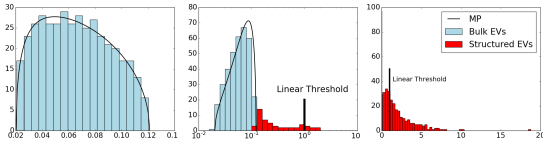
These fixed point equations can be solved numerically to tell us how the variables condensate on the SVD modes within each equilibrium state of the distribution and whether a spin glass phase is present or not. The important point here is that with K finite and a non-degenerate spectrum the mode with highest singular value dominates the ferromagnetic phase. The phase diagram looks in fact similar to the one of the SK model with ferromagnetic coupling, when $1/\sigma$ is interpreted as a temperature and w_{max}/σ the ferromagnetic coupling. Some subtleties arise when considering various ways of averaging over singular vectors components [29]. In [15, 30] it is underlined the importance of the capability of networks to produce compositional states structured by combination of hidden variables. In our representation, we don't have direct access to this property, but to the dual one in some sense, namely states corresponding to combination of modes. Their presence and their structure, are rather sensitive to the way the average over u and v is performed. In this respect the case where $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ are Gaussian i.i.d distributed is very special: all other fixed points associated to lower modes can be shown to be unstable as well as fixed points associated to combinations of modes. Instead, for other distributions with smaller kurtosis, like uniform or Bernoulli, stable fixed points associated to many different single modes or combinations of modes can exist and contribute to the thermodynamics.

Coming back to the learning dynamics, the first thing which is expected, already from the linear analysis, is that the noise term in (10) vanishes by condensing into a delta function of zero modes. Then the term corresponding to the response of the RBM in (6) is estimated (in absence of bias) in the thermodynamic by means of the order parameters defined previously:

$$\langle s_\alpha \sigma_\beta \rangle_{\text{RBM}} = \frac{L}{Z_{\text{MF}}} \sum_{q=1}^{C} e^{-F_q} \bar{m}_\alpha^{(q)} m_\beta^{(q)}, \quad Z_{MF} \stackrel{\text{def}}{=} \sum_q e^{-F_q}$$

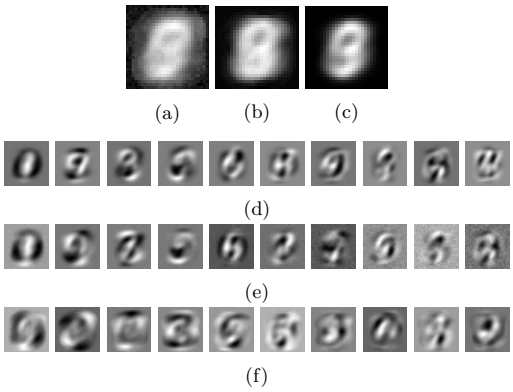
where the index q run over all stable fixed point solutions of (11,12) weighted accordingly to their free energy. These are the dominant contributions as long as free energy differences are $O(1)$, internal fluctuations given by each fixed point are comparatively of order $O(1/L)$. Note that this is the reason why the RBM needs to reach a ferromagnetic phase with many states to be able to match the empirical term in (6) in order to converge. For instance, in the case of a multimodal data distribution with many well separated clusters, the SVD modes of W which will develop are the one pointing in the direction of the magnetizations defined by these clusters. In this simple case the RBM will evolve as in the linear case to a state such that the empirical term becomes diagonal, while the singular values adjust themselves until matching the proper magnetization in each fixed point. More precise statements about the phase diagram of the RBM and the behaviour of our dynamical equations including the dynamics of the external fields η_α and θ_α will be given in [29].

Tests on the MNIST dataset. – We illustrate our results on the MNIST dataset. The MNIST dataset is composed of 60000 images of handwritten digits of 28×28



(a) (b) (c)

Fig. 2: **(a)** Singular values distribution of the initial random matrix compared to Marchenko-Pastur law. **(b)** With the training we can see some singular values strengthening and overcoming the threshold set by the Marchenko-Pastur law. **(c)** Distribution of the singular values after a long training: we can see many outliers spread above threshold and a spike of below-threshold singular values near zero.



(a) (b) (c) (d) (e) (f)

Fig. 3: **(a)** First mode learnt by the RBM with the external visible field initialized as a null vector. **(b)** External visible field initialized on the empirical mean. **(c)** First principal components extracted from the training set. **(d)** Principal components extracted from the training set (starting from the second). **(e)** The first 10 modes of a RBM trained for 1 epoch. **(f)** Same as (e) but after a 10 epochs training.

pixels. It is known that RBMs perform reasonably well on this dataset and therefore we can now interprete in the light of the preceding sections how the learning goes. For the training of the MNIST dataset we use the following parameters. The weights of the matrix W were initiated randomly from a centered Gaussian distribution with a variance of 0.01 such that the MP bulk do not pass the threshold. The visible fields are initialized to reproduce the empirical mean of the data for each visible variable. The hidden field is put to zero. The learning rate is chosen to be ≈ 0.01. With these parameters we verified that our machine was able to sample digits in a satisfactory way after 20 epochs. Now we can investigate the value of some observables introduced previously. First, we look at the SVD modes of the matrix w during the learning on Fig. 2. We see that, after seeing only few updates the system has already learned many SVD modes from the data.

On Figure 2a-2c, we observe what is expected from the linear regime. Some modes escape from the Marchenko-Pastur bulk of the eigenvalues while other condense down to zero. In particular, we can see that the modes at the beginning of the learning correspond exactly to the SVD modes of the data, see Fig. 3. On this figure, we notice that the modes of the W matrix are the same as the ones of the data at the beginning of the learning as predicted by the linear theory.

After many epochs, we observe on Fig. 3-f that non-linear effects have deformed the SVD modes of W with respect to the beginning of the learning. We can also look at the evolution of the eigenvalues of W. On Fig. 4 we observe their evolution and when they start to be amplified (or dumped). On the inset, we see how the strongest mode get out of the bulk and increase while the lowest ones are dumped after many epochs. We also observe that the top part of the spectrum of W appear flattened as compared to empirical SVD spectrum. This presumably favors the expression of many states of similar free energy related to various digit configurations, able to contribute to RBM response term in (6).

Discussion. – The equations obtained for the dynamics and the MF theory that allows us to compute them constitute a phenomenological description of the learning of an RBM. This is assumed to represent a typical learning trajectory in the limit of infinite batch size. These equations have been obtained by averaging over the components of left and right SVD vectors of the weight matrix, keeping fixed a certain number of quantities considered to be the relevant ones, fully characterizing a typical RBM during the learning process. This averaging corresponds actually to a standard self-averaging assumption in a RS phase. The singular values spectrum $\{w_\alpha\}$ is playing the main role. The projections $(\eta_\alpha, \theta_\alpha)$ of the bias onto the eigenmodes of W are also considered as intrinsic quantities. Finally the rotation vectors $\{\Omega^{v,h}_{\alpha,\beta}\}$ give us the relative motion of the data w.r.t the time dependent frame given by the singular vectors of W. In our phenomenological description the learning dynamics is represented by a trajectory of $\{w_\alpha(t), \eta_\alpha(t), \theta_\alpha(t), \Omega^{v,h}_{\alpha\beta}(t)\}$ which is uniquely determined by our equations once an initial condition specified by the decomposition of the data on the singular vectors of W is given. By contrast to usual approaches which rely on the teacher-student scenario, we may obtain generic learning curves of non-linear neural networks, which are driven by intrinsic properties of the data. The point is to give insights into the relationship between model and data. This allows us to give some elements of understanding on which properties of the data drive the learning and how they are represented in the model. Eventually this will lead us to identify and cure

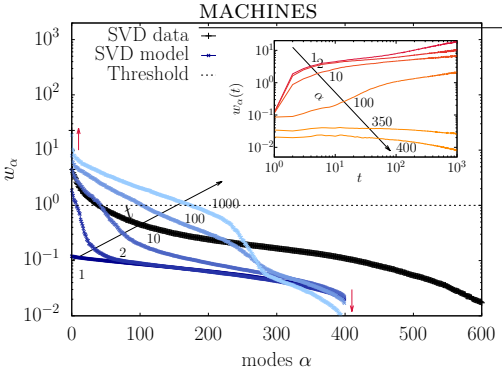


Fig. 4: Log-log plot of the singular values represented as discrete abscissas (in decreasing order) with their magnitude reported on the ordinates. The RBM contained 400 hidden variables. A cutoff is highlighted by the onset of the linear behaviour and the SVD modes of the data in black. We qualitatively observe that beyond some α_{tresh} the modes are dumped while before they are amplified. In the inset, the time evolution of the modes 1, 2, 10, 100, 350, 400 during the learning as a function of the number of epochs, we see that for large value of α, the modes are decreasing. We observe that the linear cutoff (around $\alpha \approx 50$ seems different from the one observed when going deep into the non-linear regime ($\alpha \approx 250$).

some flaws of present learning methods.

* * *

We thank B. Seoane helping us improving the quality of the letter.

REFERENCES

[1] Smolensky P., *In Parallel Distributed Processing: Volume 1 by D. Rumelhart and J. McLelland* 194-281 (MIT Press) 1986 Ch. 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory.

[2] Hinton G. E. and Salakhutdinov R. R., *Science*, **313** (2006) 504.

[3] Salakhutdinov R. and Hinton G., *Deep Boltzmann machines* in proc. of *Artificial Intelligence and Statistics* 2009 pp. 448–455.

[4] Hinton G. E., *Neural computation*, **14** (2002) 1771.

[5] Hinton G. E., *A Practical Guide to Training Restricted Boltzmann Machines* (Springer Berlin Heidelberg, Berlin, Heidelberg) 2012 pp. 599–619.

[6] Hopfield J. J., *Proceedings of the National Academy of Sciences of the United States of America*, **79** (1982) 2554.

[7] Amit D. J., Gutfreund H. and Sompolinsky H., *Annals of Physics*, **173** (1987) 30.

[8] Gardner E., *EPL (Europhysics Letters)*, **4** (1987) 481.

[9] Gardner E. and Derrida B., *Journal of Physics A: Mathematical and General*, **21** (1988) 271.

[10] Gabrié M., Tramel E. W. and Krzakala F., *Training restricted Boltzmann machines via the Thouless-Anderson-Palmer free energy* in proc. of *Proceedings of the 28th International Conference on Neural Information Processing Systems* NIPS'15 2015 pp. 640–648.

[11] Huang H. and Toyoizumi T., *Physical Review E*, **91** (2015) 050101.

[12] Takahashi C. and Yasuda M., *Journal of the Physical Society of Japan*, **85** (2016) 034001.

[13] Huang H., *Journal of Statistical Mechanics: Theory and Experiment*, **2017** (2017) 053302.

[14] Barra A., Genovese G., Sollich P. and Tantari D., *Phase diagram of restricted Boltzmann machines and generalized Hopfield networks with arbitrary priors* arXiv:1702.05882 (2017).

[15] Monasson R. and Tubiana J., *Phys. Rev. Let.*, **118** (2017) 138301.

[16] Zdeborová L. and Krzakala F., *Advances in Physics*, **65** (2016) 453.

[17] Tipping M. E. and Bishop C. M., *Neural Comput.*, **11** (1999) 443.

[18] Bourlard H. and Kamp Y., *Biological Cybernetics*, **59** (1988) 291.

[19] Saxe A. M., McClelland J. L. and Ganguli S., *Exact solutions to the nonlinear dynamics of learning in deep linear neural networks* arXiv:1312.6120 (2014).

[20] Tramel E. W., Gabrié M., Manoel A., Caltagirone F. and Krzakala F., *A Deterministic and Generalized Framework for Unsupervised Learning with Restricted Boltzmann Machines* (2017).

[21] Barra A., Bernacchia A., Santucci E. and Contucci P., *Neural Networks*, **34** (2012) 1.

[22] Hohenberg P. C. and Cross M. C., *An introduction to pattern formation in nonequilibrium systems* (Springer Berlin Heidelberg, Berlin, Heidelberg) 1987 pp. 55–92.

[23] Parisi G. and Potters M., *Journal of Physics A: Mathematical and General*, **28** (1995) 5267.

[24] Opper M. and Winther O., *Physical Review E*, **64** (2001) 056131.

[25] Mézard M., *Phys. Rev. E*, **95** (2017) 022117.

[26] Amit D. J., Gutfreund H. and Sompolinsky H., *Phys. Rev. A*, **32** (1985) 1007.

[27] Mézard M., Parisi G. and Virasoro M. A., *Spin Glass Theory and Beyond* (World Scientific, Singapore) 1987.

[28] Almeida J. R. L. and Thouless D. J., *J.Phy. A:Math. Gen*, **11** (1978) 983.

[29] Decelle A., Fissore G. and Furtlehner C., in preparation.

[30] Agliari E., Barra A., Galluzzi A., Guerra F. and Moauro F., *Phys. Rev. Lett.*, **109** (2012) 268101.

Reprint B

Thermodynamics of Restricted Boltzmann Machines and Related Learning Dynamics

arXiv:1803.01960v2 [cond-mat.dis-nn] 17 Aug 2018

Thermodynamics of Restricted Boltzmann Machines and Related Learning Dynamics

A. Decelle *G. Fissore* *C. Furtlehner*

Abstract

We investigate the thermodynamic properties of a Restricted Boltzmann Machine (RBM), a simple energy-based generative model used in the context of unsupervised learning. Assuming the information content of this model to be mainly reflected by the spectral properties of its weight matrix W, we try to make a realistic analysis by averaging over an appropriate statistical ensemble of RBMs.

First, a phase diagram is derived. Otherwise similar to that of the Sherrington-Kirkpatrick (SK) model with ferromagnetic couplings, the RBM's phase diagram presents a ferromagnetic phase which may or may not be of compositional type depending on the kurtosis of the distribution of the components of the singular vectors of W.

Subsequently, the learning dynamics of the RBM is studied in the thermodynamic limit. A "typical" learning trajectory is shown to solve an effective dynamical equation, based on the aforementioned ensemble average and explicitly involving order parameters obtained from the thermodynamic analysis. In particular, this let us show how the evolution of the dominant singular values of W, and thus of the unstable modes, is driven by the input data. At the beginning of the training, in which the RBM is found to operate in the linear regime, the unstable modes reflect the dominant covariance modes of the data. In the non-linear regime, instead, the selected modes interact and eventually impose a matching of the order parameters to their empirical counterparts estimated from the data.

Finally, we illustrate our considerations by performing experiments on both artificial and real data, showing in particular how the RBM operates in the ferromagnetic compositional phase.

1 Introduction

The Restricted Boltzmann Machine (RBM) [1] is an important machine learning tool used in many applications, by virtue of its ability to model complex probability distributions. It is a neural network which serves as a generative model, in the sense that it is able to approximate the probability distribution corresponding to the empirical distribution of any set of high-dimensional data points living in a discrete or real space of dimension $N \gg 1$. From the theoretical point of view, the RBM is of high interest as it is one of the simplest neural network generative models and the probability distribution that it defines presents a simple analytic form. Moreover, there are clear

connections between RBMs and well known disordered systems in statistical physics. As an example, when data are composed by vectors with binary components the discrete RBM takes the form of an heterogeneous Ising model composed of one layer of visible units (the observable variables) connected to one layer of hidden units (the latent or hidden variables building up the dependencies between the visible ones), in which couplings and fields are obtained from the training data through a learning procedure. In order to build more powerful models, RBMs can be stacked to form "deep" architectures. In such a case, they can form a multi-layer generative model known as a Deep Boltzmann Machine (DBM) [2] or they can be stacked and trained layerwise as a pre-training procedure for neural networks [3]. The standard learning algorithms in use are the contrastive divergence [4] (CD) and the refined Persistence CD [5] (PCD), which are based on a quick Monte Carlo estimation of the response function of the RBM and are efficient and well documented [6]. Nevertheless, despite some interesting interpretations of CD in terms of non-equilibrium statistical physics [7], the learning of RBMs remains a set of obscure recipes from the statistical physics point of view: hyperparameters (like the size of the hidden layer) are supposed to be set empirically without any theoretical guidelines.

Historically, statistical physics played a central role in studying the theoretical foundations of neural networks. In particular, during the 1980s many works on the Hopfield model [8, 9, 10, 11] managed to define its learning capacity and to compute the number of independent patterns that it could store. It is worth noticing that, as RBMs are ultimately defined as a Boltzmann distribution with pairwise interactions on a bipartite graph, they can be studied in a way similar to that used for the Hopfield model. The analogy is even stronger since connections between the Hopfield model and RBMs have been made explicit when using Gaussian hidden variables [12], here the number of patterns of the Hopfield model corresponding to the number of hidden units. Motivated by a renewed excitement for neural networks, recent works actually propose to exploit the statistical physics formulation of the RBM to understand what is its learning capacity and how mean-field methods can be exploited to improve the model. In [13, 14, 15], mean-field based learning methods using TAP equations are developed. TAP solutions are usually expected to define a decomposition of the measure in terms of pure thermodynamical states and are useful both as an algorithm to compute the marginals of the variables of the model and to identify the pure states when they are yet unknown. For instance, in a sparse explicit Boltzmann machine (i.e. without latent variables) this implicit clustering can be done by means of belief propagation fixed points [1] with simple empirical learning rules [16]. In [17, 18], an analysis of the static properties of RBMs is done assuming a given weight matrix W, in order to understand collective phenomena in the latent representation, i.e. the way latent variables organize themselves in a compositional phase [19, 20] to represent actual data. These analysis make use of the replica trick (or equivalent) making the common assumption that the components of the weight matrix W are i.i.d.; despite the fact that this approach may give some insights into the retrieval phase, this approximation is problematic since, as far as a realistic RBM is concerned (an RBM learned on data), the learning mechanism introduces correlations within the weights of W and then it seems rather crude to continue to assume the independence and hope to understand the realistic statistical properties of the model.

Concerning the learning procedure of neural networks, many recent statistical physics based analyses have been proposed, most of them within teacher-student set-

[1] a somewhat different form of the TAP equations

ting [21]. This imposes a rather strong assumption on the data in the sense that it is assumed that these are generated from a model belonging to the parametric family of interest, hiding as a consequence the role played by the data themselves in the procedure. From the analysis of related linear models [22, 23], it is already a well established fact that a selection of the most important modes of the singular values decomposition (SVD) of the data is performed in the linear case. In fact in the simpler context of linear feed-forward models the learning dynamics can be fully characterized by means of the SVD of the data matrix [24], showing in particular the emergence of each mode by order of importance with respect to the corresponding singular values.

First steps to follow this guideline have been done in [25], in the context of a general RBM and to address the shortcomings of previous analyses, in particular concerning the assumptions over the weights distribution. To this end it has been proposed to characterize both the learned RBM and the learning process itself by means of the SVD spectrum of the weight matrix in order to single out the information content of the RBM. It is assumed that the SVD spectrum is split in a continuous bulk of singular vectors corresponding to noise and a set of outliers that represent the information content. By doing this it is possible to go beyond the usual unrealistic assumption of i.i.d. weights made for analyzing RBMs. Proceeding along this direction, in the present work we first present a thermodynamic analysis of RBMs under the more realistic assumptions over the weight matrix that we propose. Then, on the same basis, the learning dynamics of RBMs is studied by direct analysis of the dynamics of the SVD modes, both in the linear and non-linear regimes.

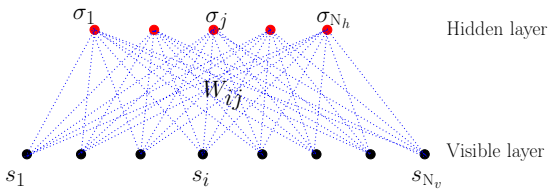


Fig. 1: bipartite structure of the RBM.

The paper is organized as follows: in Section 2 we introduce the RBM model and its associated learning procedures. Section 3 presents the static thermodynamical properties of the RBM with realistic hypothesis on its weights: a statistical ensemble of weight matrices is discussed in Section 3.1; mean-field equations in the replica-symmetric (RS) framework are given in Section 3.2 and the corresponding phase diagram is studied in Section 3.3 with a proper delimitation of the RS domain where the learning procedure is supposed to take place. The ferromagnetic phase is studied in great details in 3.4 by looking in particular at the conditions leading to a compositional phase. Section 4 is devoted to the learning dynamics. In Section 4.1, a deterministic learning equation is derived in the thermodynamic limit and a set of dynamical parameters is shown to emerge naturally from the SVD of the weight matrix. This equation is analyzed for linear RBMs in Section 4.2 in order to identify the unstable deformation modes of W that result in the first emerging patterns at the beginning of the learning process; the non-linear regime is described in Section 4.3, on the basis of the thermodynamic analysis, by numerically solving the effective learning equations in simple cases. Our analysis is finally illustrated and validated in Section 5 by actual tests on the MNIST

dataset.

2 The RBM and its associated learning procedure

An RBM is a Markov random field with pairwise interactions defined on a bipartite graph formed by two layers of non-interacting variables: the visible nodes and the hidden nodes representing respectively data configurations and latent representations (see Figure 1). The former noted $\boldsymbol{s} = \{s_i, i = 1 \dots N_v\}$ correspond to explicit representations of the data while the latter noted $\boldsymbol{\sigma} = \{\sigma_j, j = 1 \dots N_h\}$ are there to build arbitrary dependencies among the visible units. They play the role of an interacting field among visible nodes. Usually the nodes are binary-valued (of Boolean type or Bernoulli distributed) but Gaussian distributions or more broadly arbitrary distributions on real-valued bounded support are also used [26], ultimately making RBMs adapted to more heterogeneous data sets. Here to simplify we assume that visible and hidden nodes will be taken as binary variables $s_i, \sigma_j \in \{-1, 1\}$ (using ± 1 values gives the advantage of working with symmetric equations hence avoiding to deal with the "hidden" biases on the variables that appear when considering binary $\{0, 1\}$ variables). Like in the Hopfield model [8], which can actually be cast into an RBM [12], an energy function is defined for a configuration of nodes

$$E(\boldsymbol{s}, \boldsymbol{\sigma}) = -\sum_{i,j} s_i W_{ij} \sigma_j + \sum_{i=1}^{N_v} \eta_i s_i + \sum_{j=1}^{N_h} \theta_j \sigma_j \tag{1}$$

and this is exploited to define a joint distribution between visible and hidden units, namely the Boltzmann distribution

$$p(\boldsymbol{s}, \boldsymbol{\sigma}) = \frac{e^{-E(\boldsymbol{s}, \boldsymbol{\sigma})}}{Z} \tag{2}$$

where W is the weight matrix and $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ are biases, or external fields on the variables. $Z = \sum_{\boldsymbol{s}, \boldsymbol{\sigma}} e^{-E(\boldsymbol{s}, \boldsymbol{\sigma})}$ is the partition function of the system. The joint distribution between visible variables is then obtained by summing over hidden ones. In this context, learning the parameters of the RBM means that, given a dataset of M samples composed of N_v variables, we ought to infer values to W, $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ such that new generated data obtained by sampling this distribution should be similar to the input data. The general method to infer the parameters is to maximize the log likelihood of the model, where the pdf (2) has first been summed over the hidden variables

$$\mathcal{L} = \sum_j \langle \log(2 \cosh(\sum_i W_{ij} s_i - \theta_j)) \rangle_{\text{Data}} - \sum_i \eta_i \langle s_i \rangle_{\text{Data}} - \log(Z). \tag{3}$$

Different learning methods have been set up and proven to work efficiently, in particular the contrastive divergence (CD) algorithm from Hinton [4] and more recently TAP based learning [13]. They all correspond to expressing the gradient ascent on the likelihood as

$$\Delta W_{ij} = \gamma \left(\langle s_i \sigma_j p(\sigma_j | \boldsymbol{s}) \rangle_{\text{Data}} - \langle s_i \sigma_j \rangle_{p_{\text{RBM}}} \right) \tag{4}$$

$$\Delta \eta_i = \gamma \left(\langle s_i \rangle_{p_{\text{RBM}}} - \langle s_i \rangle_{\text{Data}} \right) \tag{5}$$

$$\Delta \theta_j = \gamma \left(\langle \sigma_j \rangle_{p_{\text{RBM}}} - \langle \sigma_j p(\sigma_j | \boldsymbol{s}) \rangle_{\text{Data}} \right) \tag{6}$$

where γ is the learning rate. The main problem are the $\langle\cdots\rangle_{p_{\rm RBM}}$ terms on the right hand side of (4-6). These are not tractable and the various methods basically differ in their way of estimating those terms (Monte-Carlo Markov chains, naive mean-field, TAP...). For an efficient learning the $\langle\cdots\rangle_{\rm Data}$ terms must also be approximated by making use of random mini-batches of data at each step.

3 Static thermodynamical properties of an RBM

3.1 Statistical ensemble of RBMs

When analyzing the thermodynamical properties of RBMs, it is common to assume that the weights W_{ij} are i.i.d. random variables, like for example in [20, 17, 18]. This generally leads to a Marchenko-Pastur (MP) distribution [27] of the singular values of W, which is unrealistic.

In order to clarify our notation, let us recall the definition of the singular value decomposition (SVD). As a generalization of eigenmodes decomposition to rectangular matrices, the SVD for a RBM is given by

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{7}$$

where $\mathbf{U}$ is an orthogonal $N_v \times N_h$ matrix whose columns are the left singular vectors $\mathbf{u}^\alpha$, $\mathbf{V}$ is an orthogonal $N_h \times N_h$ matrix whose columns are the right singular vectors $\mathbf{v}^\alpha$ and $\mathbf{\Sigma}$ is a diagonal matrix whose elements are the singular values w_α. The separation into left and right singular vectors is due to the rectangular nature of the decomposed matrix, and the similarity with eigenmodes decomposition is revealed by the following SVD equations

$$\mathbf{W}\mathbf{v}^\alpha = w_\alpha \mathbf{u}^\alpha$$
$$\mathbf{W}^T\mathbf{u}^\alpha = w_\alpha \mathbf{v}^\alpha$$

In [25] it is argued that the MP distribution of SVD modes actually corresponds to the noise of the weight matrix, while the information content of the RBM is better expressed by the presence of SVD modes outside of this bulk. This leads us to write the weight matrix as

$$W_{ij} = \sum_{\alpha=1}^{K} w_\alpha u_i^\alpha v_j^\alpha + r_{ij} \tag{8}$$

where the $w_\alpha = O(1)$ are isolated singular values (describing a rank K matrix), the $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ are the dominant eigenvectors of the SVD decomposition and the $r_{ij} = \mathcal{N}(0, \sigma^2/L)$ are i.i.d. terms corresponding to noise, with $L = \sqrt{N_h N_v}$. The $\{u^\alpha\}$ and $\{v^\alpha\}$ are two sets of respectively N_v and N_h-dimensional orthonormal vectors, which means that their components are respectively $O(1/\sqrt{N_v})$ and $O(1/\sqrt{N_h})$, and $K \leq N_v, N_h$. We assume $N_h < N_v$ to be the rank of W and $w_\alpha > 0$ and $O(1)$ for all α. Note that in the limit $N_v \to \infty$ and $N_h \to \infty$ with $\kappa \stackrel{\rm def}{=} N_h/N_v$ fixed and $K/L \to 0$, WW^T has a spectrum density $\rho(\lambda)$ composed of a Marchenko-Pastur bulk of eigenvalues and of set of discrete modes:

$$\rho(\lambda) = \frac{L}{2\pi\sigma^2}\frac{\sqrt{(\lambda^+ - \lambda)(\lambda - \lambda^-)}}{\kappa\lambda}\mathbb{1}_{\{\lambda\in[\lambda^-,\lambda^+]\}} + \sum_{\alpha=1}^{K}\delta(\lambda - w_\alpha^2),$$

with

$$\lambda^{\pm} \stackrel{\text{def}}{=} \sigma^2\big(\kappa^{\frac{1}{4}} \pm \kappa^{-\frac{1}{4}}\big)^2.$$

The interpretation for the noise term r_{ij} is given by the presence of an extensive number of modes at the bottom of the spectrum, along which the variables won't be able to condense but that still contribute to the fluctuations. In the present form our model of RBM is similar to the Hopfield model and recent generalizations [28], the patterns being represented by the SVD modes outside of the bulk. The main difference, in addition to the bipartite structure of the graph, is the non-degeneracy of the singular values w_α. The choice made here is to consider K finite, giving $W_{ij} = O(1/N)$ which means that the thresholds θ_j (having the meaning of feature detectors) should be $O(1)$ because feature j is detected when an extensive number of spins S_i is aligned with W_{ij}. In addition, this allows us to assume simple distributions for the components of $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ (for instance, considering them i.i.d.). Altogether, this defines the statistical ensemble of RBM to which we restrict our analysis of the learning procedure.

Another approach would be to consider $K = N_h$ extensive, thereby assuming that all modes can potentially condense even though they are associated to dominated singular values. In that case, the separation between the condensed modes and the rest should be made when order parameters are introduced and the noise would then correspond to uncondensed modes. If the number of condensed modes is assumed to be extensive, then we should instead consider an average over the orthogonal group which would lead to a slightly different mean-field theory [29, 30].

3.2 Replica symmetric Mean-field equation

Our analysis in the thermodynamic limit follows classical treatments using replicas, like [31, 9] for the Hopfield model or [17] for bipartite models. The starting point is to express the average over u, v and r_{ij} of the log partition function Z in (2) with the help of the replica trick:

$$\mathsf{E}_{u,v,r}[\log(Z)] = \lim_{p\to 0} \frac{d}{dp}\mathsf{E}_{u,v,r}[Z^p].$$

First the average over r_{ij} yields

$$\exp\Big[\frac{\sigma^2}{2L}\Big(\sum_a s_i^a\sigma_j^a\Big)^2\Big] = \exp\Big[\frac{\sigma^2}{2L}\Big(p + \sum_{a\neq b} s_i^a s_i^b \sigma_j^a\sigma_j^b\Big)\Big].$$

After this averaging, 4 sets of order parameters $\{(m_\alpha^a, \bar{m}_\alpha^a), a = 1,\dots p, \alpha = 1,\dots K\}$ and $\{(Q_{ab}, \bar{Q}_{ab}), a, b = 1,\dots p, a \neq b\}$ are introduced with the help of two distinct Hubbard-Stratonovich transformations. The first one corresponds to

$$\exp\Big[\frac{\sigma^2}{2L}\Big(\sum_{i,j,a\neq b} s_i^a s_i^b \sigma_j^a\sigma_j^b\Big)\Big] = \int \prod_{a\neq b} \frac{dQ_{ab}d\bar{Q}_{ab}}{2\pi}$$
$$\times \exp\Big[-\frac{L\sigma^2}{2}\sum_{a\neq b}\Big(Q_{ab}\bar{Q}_{ab} - \frac{Q_{ab}}{N_v}\sum_i s_i^a s_i^b - \frac{\bar{Q}_{ab}}{N_h}\sum_j \sigma_j^a\sigma_j^b\Big)\Big].$$

The second one is aimed at extracting magnetization's contributions correlated with the modes:

$$\exp\Big(L\sum_{\alpha} w_\alpha s_\alpha^a \sigma_\alpha^a\Big) \propto \int \prod_\alpha \frac{dm_\alpha^a d\bar{m}_\alpha^a}{2\pi} \times \exp\Big(-L\sum_\alpha w_\alpha\big(m_\alpha^a \bar{m}_\alpha^a - m_\alpha^a s_\alpha^a - \bar{m}_\alpha^a \sigma_\alpha^a\big)\Big),$$

with

$$s_\alpha^a \stackrel{\text{def}}{=} \frac{1}{\sqrt{L}}\sum_i s_i u_i^\alpha \qquad \text{and} \qquad \sigma_\alpha^a \stackrel{\text{def}}{=} \frac{1}{\sqrt{L}}\sum_j \sigma_j^a v_j^\alpha, \tag{9}$$

These variables represent the following quantities:

$$m_\alpha^a \sim E_{u,v,r}\big(\langle \sigma_\alpha^a\rangle\big) \qquad \bar{m}_\alpha^a \sim E_{u,v,r}\big(\langle s_\alpha^a\rangle\big)$$
$$Q_{ab} \sim E_{u,v,r}\big(\langle \sigma_i^a \sigma_i^b\rangle\big) \qquad \bar{Q}_{ab} \sim E_{u,v,r}\big(\langle s_j^a s_j^b\rangle\big),$$

namely the correlations of the hidden [resp. visible] states with the left [resp. right] singular vectors and the Edward-Anderson (EA) order parameters measuring the correlation between replicas of hidden or visible states. E_u and E_v denote an average w.r.t. the rescaled components $u \simeq \sqrt{N_v}u_i^\alpha$ and $v \simeq \sqrt{N_h}v_j^\alpha$ of the SVD modes. The transformations involve pairs of complex integration variables because of the asymmetry introduced by the two-layers structure in contrast to fully connected models.

We obtain the following representation:

$$\mathsf{E}_{u,v,r}[Z^p] = \int \prod_{a,\alpha} \frac{dm_\alpha^a d\bar{m}_\alpha^a}{2\pi} \prod_{a\neq b} \frac{dQ_{ab}d\bar{Q}_{ab}}{2\pi} \times \exp\Big\{-L\Big(\sum_{a,\alpha} w_\alpha m_\alpha \bar{m}_\alpha + \frac{\sigma^2}{2}\sum_{a\neq b} Q_{ab}\bar{Q}_{ab} - \frac{1}{\sqrt{\kappa}}A[m,Q] - \sqrt{\kappa}B[\bar{m},\bar{Q}]\Big)\Big\}$$

with $\kappa = N_h/N_v$ and

$$A[m,Q] \stackrel{\text{def}}{=} \log\Big[\sum_{S^a\in\{-1,1\}} \mathsf{E}_u\Big(e^{\frac{\sqrt{\kappa}\sigma^2}{2}\sum_{a\neq b} Q_{ab}S^aS^b + \kappa^{\frac{1}{4}}\sum_{a,\alpha}(w_\alpha m_\alpha^a - \eta_\alpha)u^\alpha S^a}\Big)\Big], \tag{10}$$

$$B[\bar{m},\bar{Q}] \stackrel{\text{def}}{=} \log\Big[\sum_{S^a\in\{-1,1\}} \mathsf{E}_v\Big(e^{\frac{\sqrt{\kappa}\sigma^2}{2}\sum_{a\neq b} \bar{Q}_{ab}\sigma^a\sigma^b + \kappa^{-\frac{1}{4}}\sum_{a,\alpha}(w_\alpha \bar{m}_\alpha^a - \theta_\alpha)v^\alpha\sigma^a}\Big)\Big], \tag{11}$$

$$\tag{12}$$

with

$$\theta_\alpha \stackrel{\text{def}}{=} \frac{1}{\sqrt{L}}\sum_j \theta_j v_j^\alpha = O(1).$$

Since $\{v^\alpha\}$ is an incomplete basis we also need to take care of the potential residual

transverse parts $\eta^\perp$ and $\theta^\perp$, such that the following decompositions hold:

$$\eta_i = \eta_i^\perp + \sqrt{L}\sum_\alpha \eta_\alpha u_i^\alpha, \tag{13}$$

$$\theta_j = \theta_j^\perp + \sqrt{L}\sum_\alpha \theta_\alpha v_j^\alpha. \tag{14}$$

To keep things tractable, both $\eta^\perp$ and $\theta^\perp$ will be considered negligible in the sequel. Taking into account these components would lead to the addition of a random field to the effective RS field of the variables and eventually to a richer set of saddle point solutions. Note that the order of magnitude of η_α and θ_α is at this stage an assumption. If η_i and u_i^α (or θ_j and v_j^α) were uncorrelated they would scale as $1/\sqrt{L}$. Moreover, regarding the ensemble average, we will consider η_α and θ_α fixed in the sequel.

The thermodynamic properties are obtained by first making a saddle point approximation possible by letting first $L \to \infty$ and taking the limit $p \to 0$ afterwards. We restrict here the discussion to RS saddle points [32]. The breakdown of RS can actually be determined by computing the so-called AT line [33] (see Appendix A). At this point we assume a non-broken replica symmetry. The set $\{Q_{ab}, \bar{Q}_{ab}\}$ reduces then to a pair $(q, \bar{q})$ of spin glass parameters, i.e. $Q_{ab} = q$ and $\bar{Q}_{ab} = \bar{q}$ for all $a \neq b$, while quenched magnetizations on the SVD directions are now represented by $\{(m_\alpha, \bar{m}_\alpha), \alpha = 1, \dots K\}$.

Taking the limit $p \to 0$ yields the following limit for the free energy:

$$f[m, \bar{m}, q, \bar{q}] = \sum_\alpha w_\alpha m_\alpha \bar{m}_\alpha - \frac{\sigma^2}{2} q\bar{q} + \frac{\sigma^2}{2}(q + \bar{q})$$

$$- \frac{1}{\sqrt{\kappa}} \mathsf{E}_{u,x}\Big[\log 2\cosh\big(h(x,u)\big)\Big] - \sqrt{\kappa}\,\mathsf{E}_{v,x}\Big[\log 2\cosh\big(\bar{h}(x,v)\big)\Big]. \tag{15}$$

Assuming a replica-symmetric phase, the saddle-point equations are given by

$$m_\alpha = \kappa^{\frac{1}{4}} \mathsf{E}_{v,x}\Big[v^\alpha \tanh\big(\bar{h}(x,v)\big)\Big], \qquad q = \mathsf{E}_{v,x}\Big[\tanh^2\big(\bar{h}(x,v)\big)\Big] \tag{16}$$

$$\bar{m}_\alpha = \kappa^{-\frac{1}{4}} \mathsf{E}_{u,x}\Big[u^\alpha \tanh\big(h(x,u)\big)\Big], \qquad \bar{q} = \mathsf{E}_{u,x}\Big[\tanh^2\big(h(x,u)\big)\Big] \tag{17}$$

where

$$h(x,u) \stackrel{\text{def}}{=} \kappa^{\frac{1}{4}}\big(\sigma\sqrt{q}x + \sum_\gamma (w_\gamma m_\gamma - \eta_\gamma)u^\gamma\big)$$

$$\bar{h}(x,v) \stackrel{\text{def}}{=} \kappa^{-\frac{1}{4}}\big(\sigma\sqrt{\bar{q}}x + \sum_\gamma (w_\gamma \bar{m}_\gamma - \theta_\gamma)v^\gamma\big),$$

and $\kappa = N_h/N_v$, with $\mathsf{E}_{u,x}$ and $\mathsf{E}_{v,x}$ denoting an average over the Gaussian variable $x = \mathcal{N}(0,1)$ and the rescaled components $u \sim \sqrt{N_v}u_i^\alpha$ and $v \sim \sqrt{N_h}v_j^\alpha$ of the SVD modes. We note that the equations are symmetric under the exchange $\kappa \to \kappa^{-1}$, simultaneously with $m \leftrightarrow \bar{m}$, $q \leftrightarrow \bar{q}$ and $\eta \leftrightarrow \theta$, given that u and v have the same distribution. In addition, for independently distributed u_i^α and v_j^α and vanishing fields ($\eta = \theta = 0$), solutions corresponding to non-degenerate magnetizations have symmetric counterparts: each pair of non-vanishing magnetizations can be negated independently as $(m_\alpha, \bar{m}_\alpha) \to (-m_\alpha, -\bar{m}_\alpha)$, generating new solutions. So to one solution presenting n condensed modes, there correspond 2^n distinct solutions.

3.3 Phase Diagram

The fixed point equations (16, 17) can be solved numerically to tell us how the variables condensate on the SVD modes within each equilibrium state of the distribution and whether a spin-glass or a ferromagnetic phase is present. The important point here is that with K finite and a non-degenerate spectrum the mode with highest singular value dominates the ferromagnetic phase.

In absence of bias ($\eta = \theta = 0$) and once $1/\sigma$ is interpreted as temperature and w_α/σ as ferromagnetic couplings, we get a phase diagram similar to that of the Sherrington-Kirkpatrick (SK) model with three distinct phases (see Figure 2)

- a paramagnetic phase ($q = \bar{q} = m_\alpha = \bar{m}_\alpha = 0$) (P),
- a ferromagnetic phase ($q, \bar{q}, m_\alpha, \bar{m}_\alpha \neq 0$) (F),
- a spin glass phase ($q, \bar{q} \neq 0$; $m_\alpha = \bar{m}_\alpha = 0$) (SG).

In general, the lines separating the different phases correspond to second order phase transitions and can be obtained by a stability analysis of the Hessian of the free energy. They are related to unstable modes of the linearized mean-field equations and correspond to an eigenvalue of the Hessian becoming negative.

The (SG-P) line is obtained by looking at the Hessian in the $(q, \bar{q})$ sector:

$$H_{q\bar{q}} \underset{\substack{m=0\\ q=0}}{=} -\frac{1}{2}\begin{bmatrix} \sigma^2 & \frac{\sigma^4}{\sqrt{\kappa}} \\ \sqrt{\kappa}\sigma^4 & \sigma^2 \end{bmatrix}$$

from what results that the spin glass phase develops when $\sigma \geq 1$[2]. This transition line is understood tacking directly into account the spectral properties of the weight matrix. Classically, this is done with the help of the linearized TAP equations and exploiting the Marchenko-Pastur distribution [32]. In our context, the linearized TAP equations read

$$\begin{bmatrix} \mu \\ \nu \end{bmatrix} = \begin{bmatrix} -\sqrt{\kappa}\sigma^2 & W^T \\ W & -\frac{\sigma^2}{\sqrt{\kappa}} \end{bmatrix} \begin{bmatrix} \mu \\ \nu \end{bmatrix}$$

given the variance σ^2/L of the weights in absence of dominant modes. Then we can show that the paramagnetic phase becomes unstable when the highest eigenvalue of the matrix on the rhs is equal to 1: if λ is a singular value of W, the corresponding eigenvalues $\Lambda^\pm$ verify the relation

$$\big(\frac{\Lambda^\pm}{\sqrt{\kappa}} \pm \sigma^2\big)\big(\sqrt{\kappa}\Lambda^\pm \pm \sigma^2\big) = \lambda^2.$$

from which it is clear that the largest eigenvalue Λ_{max} corresponds to the largest singular value λ_{max}. Owing to the Marchenko-Pastur distribution $\lambda_{max} = \sigma^2(\sqrt{\kappa} + 1)(1 + 1/\sqrt{\kappa})$ so Λ_{max} verifies

$$\big(\frac{\Lambda_{max}}{\sqrt{\kappa}} + \sigma^2\big)\big(\sqrt{\kappa}\Lambda_{max} + \sigma^2\big) = \sigma^2(\sqrt{\kappa} + 1)\big(\frac{1}{\sqrt{\kappa}} + 1\big).$$

$\Lambda_{max} = 1$ is readily obtained for $\sigma^2 = 1$.

[2] Note that in [17] a dependence $\sqrt{\kappa(1-\kappa)}$ $\Big(\sqrt{\alpha(1-\alpha)}$in their notation$\Big)$ is found. This dependence is hidden in our definition of σ^2 giving $L = \sqrt{N_v N_h}$ times the variance of r_{ij} instead of $N_v + N_h$ as in their case.

For the (F-SG) frontier we can look at the sector $(m_\alpha, \bar{m}_\alpha)$ corresponding to the emergence of a single mode α (written in the spin-glass phase):

$$H_{\alpha\alpha} = \begin{bmatrix} w_\alpha & w_\alpha^2 \mathsf{E}_{v,x}\Big[(v^\alpha)^2 \operatorname{sech}^2\big(\bar{h}(x,v)\big)\Big] \\ w_\alpha^2 \mathsf{E}_{u,x}\Big[(u^\alpha)^2 \operatorname{sech}^2\big(h(x,u)\big)\Big] & w_\alpha \end{bmatrix}$$

$$\underset{m_\alpha=0}{=} \begin{bmatrix} w_\alpha & w_\alpha^2(1-q) \\ w_\alpha^2(1-\bar{q}) & w_\alpha \end{bmatrix}$$

From this it is clear that the first mode to become unstable is the mode α with highest singular value w_α and this occurs when q and $\bar{q}$, solutions of (16,17), verify

$$(1-q)(1-\bar{q})w_\alpha^2 = 1.$$

As for the SK model, this line appears to be well below the de Almeida-Thouless (AT) line, which is the line above which the RS solution is stable (see Figure 2, and Appendix A for the computation of the AT line). This means that in principle a replica symmetry breaking treatment would be necessary to properly separate the two phases. However, we will leave aside this point as we are mainly interested in the practical aspects, namely the ability of the RBM to learn arbitrary data, and so we are mostly concerned with the ferromagnetic phase above the AT line.

For the (P-F) line we consider the same sector of the Hessian but now written in the paramagnetic phase, i.e. setting $q = 0$ in the above equation, and this simply yields the emergence of the single mode α for $w_\alpha = 1$.

Note that all of this is independent on how the statistical average over u and v is performed. Instead, as we shall see later on, the way of averaging influences the nature of the ferromagnetic phase.

Regarding the stability of the RS solution, the computation of the AT line reported in Appendix A is similar to the classical one made for the SK model, though slightly more involved. In fact we were not able to fully characterize, in replica space, all the possible instabilities of the Hessian which would potentially lead to a breakdown of the replica symmetry. At least the one responsible for the ordinary SK model RS breakdown has a counterpart in the bipartite case that gives a necessary condition for the stability of the RS solution:

$$\frac{1}{\sigma^2} > \sqrt{\mathsf{E}_{x,u}\Big(\operatorname{sech}^4\big(h(x,u)\big)\Big)\mathsf{E}_{x,v}\Big(\operatorname{sech}^4\big(\bar{h}(x,v)\big)\Big)},$$

For $\kappa = 1$ the terms below the radical become identical and the condition reduces to the one of the SK model, except for the u averages which are not present in the SK model. In Figure 2, is shown the influence on the phase diagram of the value of κ and of the type of average made on u and v.

3.4 Nature of the Ferromagnetic phase

Some subtleties arise when considering various ways of averaging over the components of the singular vectors. In [19, 20] is emphasized the importance for networks to be able to reproduce compositional states structured by combination of hidden variables. In our representation, we don't have direct access to this property but, in some sense, to the dual one, which is given by states corresponding to combinations of modes. Their presence and their structure are rather sensitive to the way the average over u

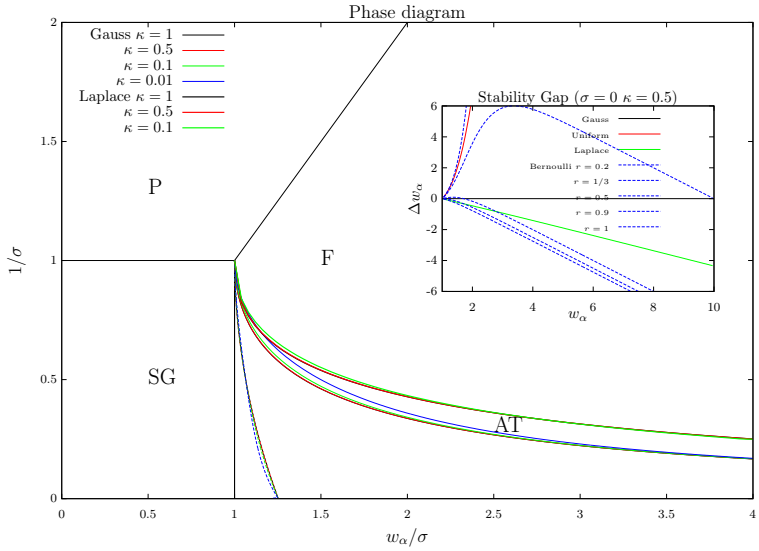


Fig. 2: Phase diagram in absence of bias and with a finite number of modes, with Gaussian and Laplace distributions for u and v. The dotted line separates the spin glass phase from the ferromagnetic phase under the RS hypothesis. The RS phase is unstable below the AT line. The influence of κ on the AT and SG-F lines is shown. In all cases, the hypothetical SG-F line lies well inside the broken RS phase. Inset: high temperature ($\sigma = 0$) stability gap Δw_α corresponding to a fixed point associated to a mode β, expressed as a function of w_α and considering various distributions.

and v is performed. In this respect the case in which $\boldsymbol{u}^\alpha$ and $\boldsymbol{v}^\alpha$ have i.i.d. Gaussian components is very special: all fixed points associated to dominated modes can be shown to be unstable and fixed points associated to combinations of modes are not allowed. To see this, first notice that in such a case the magnetization's part of the saddle point equations (16,17) read

$$m_\alpha = (w_\alpha \bar{m}_\alpha - \theta_\alpha)(1 - q) \tag{18}$$

$$\bar{m}_\alpha = (w_\alpha m_\alpha - \eta_\alpha)(1 - \bar{q}). \tag{19}$$

Since the role of the bias is mainly to introduce some asymmetry between otherwise degenerated fixed points obtained by sign reversal of at least one pair $(m_\alpha, \bar{m}_\alpha)$, let us analyze the situation without fields, i.e. by setting $\eta = \theta = 0$. We immediately see that as long as the singular values are non degenerate, only one single mode may condense at a time. Indeed if mode α condenses we necessarily have

$$w_\alpha^2 (1 - q)(1 - \bar{q}) = 1,$$

and this can be verified only by one mode at a time. Looking at the stability of the fixed points, we see that only the fixed point associated to the largest singular value is actually stable (details reported after the introduction of lemma 3.1).

For other distributions like uniform Bernoulli or Laplace, instead, stable fixed points associated to many different single modes or combinations of modes can exist and contribute to the thermodynamics. In order to analyze this question in more general terms we first rewrite the mean-field equations in a convenient way which require some preliminary remarks. We restrict the discussion to i.i.d. variables so that we can consider single variable distributions. Joint distributions will be distinguished from single variable distributions by the use of bold: $\mathbf{u} = \{u^\alpha, \alpha = 1, \dots, K\}$, K being the (finite) number of modes susceptible of condensing.

Given the distribution p and assuming it to be even, we define a related distribution $p^\star$ attached to mode α:

$$p^\star(u) \stackrel{\text{def}}{=} -\int_{-\infty}^{u} xp(x)dx = \int_{|u|}^{\infty} xp(x)dx, \tag{20}$$

This distribution has some useful properties.

Lemma 3.1. *Given that p is centered with unit variance and kurtosis κ_u, $p^\star$ is a centered probability distribution with variance*

$$\int_{-\infty}^{\infty} u^2 p^\star(u)du = \frac{\kappa_u}{3}.$$

Proof. Consider the moments of $p^\star$. For n odd they vanish while for n even they read:

$$\begin{aligned}
\int_{-\infty}^{+\infty} u^n p^\star(u)du &= 2\int_0^{\infty} u^n p^\star(u)du \\
&= 2\int_0^{\infty} duu^n \int_u^{\infty} xp(x)dx \\
&= 2\int_0^{\infty} xp(x)dx \int_0^{x} u^n du \\
&= \frac{1}{n+1}\int_{-\infty}^{\infty} x^{n+2} p(x)dx,
\end{aligned}$$

i.e. the n_{th} even moments of $p^\star$ relate to moments of order $n+2$ of p. The lemma then follows from the fact that p has unit variance. ∎

In this respect, the Gaussian averaging is special because we have $\kappa_u = 3$ and $p^\star = p$. Then the mean-field equations (16,17) corresponding to the magnetizations can be rewritten in a form similar to (18,19) by introducing the variables q_α and $\bar{q}_\alpha$:

$$m_\alpha = (w_\alpha \bar{m}_\alpha - \theta_\alpha)(1 - q_\alpha), \tag{21}$$

$$\bar{m}_\alpha = (w_\alpha m_\alpha - \eta_\alpha)(1 - \bar{q}_\alpha), \tag{22}$$

with

$$q_\alpha = \int dx \frac{e^{-x^2/2}}{\sqrt{2\pi}} d\mathbf{v} p_\alpha(\mathbf{v}) \tanh^2\Big(\kappa^{-\frac{1}{4}}\big(\sigma\sqrt{\bar{q}}x + \sum_\gamma (w_\gamma \bar{m}_\gamma - \theta_\gamma)v^\gamma\big)\Big), \tag{23}$$

$$\bar{q}_\alpha = \int dx \frac{e^{-x^2/2}}{\sqrt{2\pi}} d\mathbf{u} p_\alpha(\mathbf{u}) \tanh^2\Big(\kappa^{\frac{1}{4}}\big(\sigma\sqrt{q}x + \sum_\gamma (w_\gamma m_\gamma - \eta_\gamma)u^\gamma\big)\Big), \tag{24}$$

where

$$p_\alpha(\mathbf{u}) \stackrel{\text{def}}{=} p^\star(u^\alpha) \prod_{\beta\neq\alpha} p(u^\beta).$$

This rewriting will prove very useful also in the next section when analyzing the learning dynamics.

Let us now assume, in absence of bias, a non-degenerate fixed point associated to some given mode β with finite $(m_\beta, \bar{m}_\beta)$ and $m_\alpha = \bar{m}_\alpha = 0, \forall \alpha \neq \beta$. The fixed point equation imposes the relation

$$w_\beta = \frac{1}{\sqrt{(1-q_\beta)(1-\bar{q}_\beta)}} \stackrel{\text{def}}{=} w(q_\beta, \bar{q}_\beta). \tag{25}$$

The stability of such a fixed point with respect to any other mode α is related to the positive definiteness of the following block of the Hessian

$$H_{\alpha\alpha} = \begin{bmatrix} w_\alpha & w_\alpha^2 \mathsf{E}_{v,x}\Big[(v^\alpha)^2 \operatorname{sech}^2\big(\bar{h}(x,v)\big)\Big] \\ w_\alpha^2 \mathsf{E}_{u,x}\Big[(u^\alpha)^2 \operatorname{sech}^2\big(h(x,u)\big)\Big] & w_\alpha \end{bmatrix}$$

with, in the present case

$$h(x,u) = \kappa^{\frac{1}{4}}\big(\sigma\sqrt{q}x + w_\beta \bar{m}_\beta u^\beta\big) \qquad \text{and} \qquad \bar{h}(x,v) = \kappa^{-\frac{1}{4}}\big(\sigma\sqrt{\bar{q}}x + w_\beta \bar{m}_\beta v^\beta\big),$$

This reduces to

$$H_{\alpha\alpha} = \begin{bmatrix} w_\alpha & w_\alpha^2(1-q) \\ w_\alpha^2(1-\bar{q}) & w_\alpha \end{bmatrix}.$$

Therefore for the Gaussian averaging case, since $q_\beta = q$, $\bar{q}_\beta = \bar{q}$ and given (25), we necessarily have

$$1 - (1-q)(1-\bar{q})w_\alpha^2 = 1 - \frac{w_\alpha^2}{w_\beta^2} < 0 \qquad \text{for} \qquad w_\alpha > w_\beta,$$

i.e. the Hessian has negative eigenvalues. This means that if the mode β is dominated by another mode α, the magnetization $(m_\alpha, \bar{m}_\alpha)$ will develop until $(1-q)(1-\bar{q})w_\alpha^2 = 1$, while m_β will vanish.

For the general case of i.i.d. variables, assuming u^α and v^α obey the same distribution p, let F and F_α be the cumulative distributions associated respectively to p and p_α

$$F(u) \stackrel{\text{def}}{=} \int_{-\infty}^{u} p(x)dx$$

$$F_\alpha(u) \stackrel{\text{def}}{=} \int d\mathbf{u}\ \theta(u-u^\alpha) p_\alpha(\mathbf{u})dx = -\int_\infty^u du^\alpha \int_{-\infty}^{u^\alpha} xp(x)dx.$$

Given the values of $(q, \bar{q})$ obtained from the fixed point associated to mode β, we have the following property:

Proposition 3.2. *If*

$$(i)\ F_\beta(u) < F(u), \qquad \forall u \in \mathbb{R}^+ \qquad \textit{then} \qquad q_\beta > q \qquad \textit{and} \qquad \bar{q}_\beta > \bar{q},$$

$$(ii)\ F_\beta(u) > F(u), \qquad \forall u \in \mathbb{R}^+ \qquad \textit{then} \qquad q_\beta < q \qquad \textit{and} \qquad \bar{q}_\beta < \bar{q},$$

which in turn implies

$$w(q, \bar{q}) < w_\beta\ (i) \qquad \textit{and} \qquad w(q, \bar{q}) > w_\beta\ (ii)$$

with

$$w(q, \bar{q}) \stackrel{\text{def}}{=} \frac{1}{\sqrt{(1-q)(1-\bar{q})}}.$$

Proof. This is obtained by straightforward by parts integration respectively over u and v in equations (16,17), relative to magnetizations. ∎

In other words if F_β dominates F on $\mathbb{R}^+$ then there is a positive stability gap defined as

$$\Delta w_\beta \stackrel{\text{def}}{=} w(q, \bar{q}) - w_\beta \tag{26}$$

such that there is a non-empty range for higher values of $w_\alpha \in [w_\beta, w(q, \bar{q})[$ for which the fixed point associated to mode β corresponds to a local minimum of the free energy. Note that property (i) [resp. (ii)] is analogous (in the sense that it implies it) to p_β having a larger [resp. smaller] variance than p, i.e. $\kappa_u > 3$ [resp. $\kappa_u < 3$]. Therefore distributions p with negative relative kurtosis $(\kappa_u - 3)$ will tend to favor the presence of metastable states, while the situation will tend to be more complex for probabilities with positive relative kurtosis. Indeed, in the latter case the fixed point associated to the highest mode α_{max} might not correspond to a stable state if lower modes in the range $[w(q, \bar{q}), w_{\alpha_{max}}[$ are present, and fixed points associated to combinations of modes have to be considered. Note that in contrary with the Gaussian case, this can happen because q_α is different for each mode and therefore more flexibility is offered by equations (21,22) than from equations (18,19).

Let us give some examples. Denote by $\gamma_u \stackrel{\text{def}}{=} \kappa_u - 3$ the relative kurtosis. As already said the Gaussian distribution is a special case with $\gamma_u = 0$. In addition, for instance for p corresponding to Bernoulli, Uniform or Laplace, we have the following properties illustrated in the inset of Figure 2:

- Bernoulli ($\gamma_u = -2$):

$$p(u) = \frac{1}{2}\big(\delta(u+1) + \delta(u-1)\big), \qquad F(u) = \frac{1}{2}\big(\theta(u+1) + \theta(u-1)\big)$$
$$p_\alpha(u) = \frac{1}{2}\theta(1-u^2), \qquad F_\alpha(u) = \frac{1}{2}\theta(1-u^2)(u+1) + \theta(u-1)$$

  then $F_\alpha(u) > F(u)$ for $u > 0$, yielding a positive stability gap.

- Uniform ($\gamma_u = -6/5$):

$$p(u) = \frac{1}{2\sqrt{3}}\theta(3-u^2), \qquad F(u) = \frac{1}{2\sqrt{3}}\theta(3-u^2)(u+\sqrt{3}) + \theta(u-\sqrt{3})$$
$$p_\alpha(u) = \frac{1}{4\sqrt{3}}\theta(3-u^2)(3-u^2), \qquad F_\alpha(u) = \frac{1}{4\sqrt{3}}\theta(3-u^2)(3u - \frac{u^3}{3} + 2\sqrt{3}) + \theta(u-\sqrt{3}).$$

It can be verified that $F_\alpha(u) > F(u)$ for $u > 0$, yielding again a positive stability gap.

- Laplace ($\gamma_u = 3$):

$$p(u) = \frac{1}{\sqrt{2}} e^{-\sqrt{2}|u|}, \qquad F(u) = \frac{1}{2} + \frac{u}{2|u|}\big(1 - e^{-\sqrt{2}|u|}\big)$$
$$p_\alpha(u) = \frac{1}{2}\big(|u| + \frac{1}{\sqrt{2}}\big) e^{-\sqrt{2}|u|}, \qquad F_\alpha(u) = F(u) - \frac{u}{2\sqrt{2}} e^{-\sqrt{2}|u|}.$$

Here we have $F_\alpha(u) < F(u)$ for $u > 0$, yielding a negative stability gap.

These three examples fall either in condition (i) or (ii), with a stability gap Δw_β that is either always positive or always negative, independently of w_β. We can also provide examples for which the stability condition may vary with w_β. Consider for instance a sparse Bernoulli distribution, with $r \in [0, 1]$ a sparsity parameter:

$$p(u) = \frac{r}{2}\big(\delta(u + \frac{1}{\sqrt{r}}) + \delta(u - \frac{1}{\sqrt{r}})\big) + (1 - r)\delta(u).$$

The relative kurtosis is in this case

$$\gamma_u(r) = \frac{1}{r} - 3.$$

Looking at $F(u)$ and $F_\alpha(u)$ it is seen that both conditions (i) and (ii) are not fulfilled, except for $r = 1$ which corresponds to the plain Bernoulli case. As we see in the inset of Figure 2, for $r < 1/3$ the stability gap is always negative, meaning that a unimodal ferromagnetic phase is not stable, and it is replaced by a compositional ferromagnetic phase at all temperatures. Instead, for $r > 1/3$ and at sufficiently high temperature (low w_α) the single mode fixed point dominate the ferromagnetic phase.

Laplace distribution: let us look at the properties of the phase diagram in the case of singular vectors' components being Laplace i.i.d., case in which a negative stability gap is expected and it may lead to a compositional phase. For this we need the expression for a sum of Laplace variables to compute the averages involved in (16,17). For this purpose, we define the following distributions:

$$f(s) = \int \prod_\gamma du^\gamma \frac{\lambda_\gamma}{2} e^{-\lambda_\gamma |u^\gamma|}\ \delta(s - \sum_\gamma u^\gamma),$$

$$g_\alpha(s) = \int du^\alpha \frac{\lambda_\alpha}{4}(\lambda_\alpha |u^\alpha| + 1) e^{-\lambda_\alpha |u^\alpha|} \prod_{\gamma \neq \alpha} du^\gamma \frac{\lambda_\gamma}{2} e^{-\lambda_\gamma |u^\gamma|}\ \delta(s - \sum_\gamma u^\gamma).$$

Their Laplace transform upon decomposing into partial fractions reads:

$$\tilde{f}(\omega) = \prod_\gamma \frac{\lambda_\gamma^2}{\lambda_\gamma^2 - \omega^2} = \sum_\gamma C_\gamma \frac{\lambda_\gamma^2}{\lambda_\gamma^2 - \omega^2}$$

and

$$\tilde{g}_\alpha(\omega) = \frac{\lambda_\alpha^2}{\lambda_\alpha^2 - \omega^2} \prod_\gamma \frac{\lambda_\gamma^2}{\lambda_\gamma^2 - \omega^2}$$
$$= C_\alpha \frac{\lambda_\alpha^4}{(\lambda_\alpha^2 - \omega^2)^2} + \sum_{\gamma \neq \alpha} C_\gamma \frac{\lambda_\gamma^2 \lambda_\alpha^2}{\lambda_\alpha^2 - \lambda_\gamma^2}\Big(\frac{1}{\lambda_\gamma^2 - \omega^2} - \frac{1}{\lambda_\alpha^2 - \omega^2}\Big).$$

where

$$C_\gamma \stackrel{\text{def}}{=} \prod_{\delta\neq\gamma} \frac{\lambda_\delta^2}{\lambda_\delta^2-\lambda_\gamma^2}.$$

From these decompositions we immediately identify

$$f(s) = \frac{1}{2}\sum_\gamma C_\gamma \lambda_\gamma e^{-\lambda_\gamma|s|},$$

$$g_\alpha(s) = \frac{\lambda_\alpha C_\alpha}{4}(\lambda_\alpha|s|+1)e^{-\lambda_\alpha|s|} + \frac{1}{2}\sum_{\gamma\neq\alpha} C_\gamma \frac{\lambda_\gamma\lambda_\alpha}{\lambda_\alpha^2-\lambda_\gamma^2}\big(\lambda_\alpha e^{-\lambda_\gamma|s|} - \lambda_\gamma e^{-\lambda_\alpha|s|}\big).$$

This results in the following decomposition of the EA parameters:

$$q = \int dxds \frac{e^{-\sqrt{2}|s|-x^2/2}}{2\sqrt{\pi}} \sum_\gamma C_\gamma[\bar{m}] \tanh^2\big(\bar{h}_\gamma(x,s)\big) \tag{27}$$

$$q_\alpha = \int dxds \frac{e^{-\sqrt{2}|s|-x^2/2}}{2\sqrt{\pi}} \Big[\frac{1}{\sqrt{2}}(|s|+\frac{1}{\sqrt{2}})C_\alpha[\bar{m}]\tanh^2\big(\bar{h}_\alpha(x,s)\big) \tag{28}$$

$$+\sum_{\gamma\neq\alpha} C_\gamma[\bar{m}] \frac{(w_\gamma\bar{m}_\gamma-\theta_\gamma)^2\tanh^2\big(\bar{h}_\gamma(x,s)\big) - (w_\alpha\bar{m}_\alpha-\theta_\alpha)^2\tanh^2\big(\bar{h}_\alpha(x,s)\big)}{(w_\gamma\bar{m}_\gamma-\theta_\gamma)^2-(w_\alpha\bar{m}_\alpha-\theta_\alpha)^2}\Big] \tag{29}$$

with

$$\bar{h}_\gamma(x,s) \stackrel{\text{def}}{=} \kappa^{-\frac{1}{4}}\big(\sigma\sqrt{\bar{q}}x + (w_\gamma\bar{m}_\gamma-\theta_\gamma)s\big)$$

and

$$C_\gamma[\bar{m}] \stackrel{\text{def}}{=} \prod_{\delta\neq\gamma} \frac{(w_\gamma\bar{m}_\gamma-\theta_\gamma)^2}{(w_\gamma\bar{m}_\gamma-\theta_\gamma)^2-(w_\delta\bar{m}_\delta-\theta_\delta)^2}.$$

This allows for an efficient resolution of the mean-field equations (16,17,21,22), which let us observe the appearance of a purely compositional phase in the ferromagnetic domain when the modes at the top of the spectrum get close enough. In order to characterize this phase, we consider the stability gap $\Delta^{(n)}(w_\alpha)$ for which the range $[w_a-\Delta^{(n)}(w_\alpha), w_a]$ lies below the highest mode w_a, such that the ferromagnetic states correspond to the condensation of n distinct modes present in this interval, including the highest.

In addition, this will prove useful when analyzing the learning dynamics described in the next section.

4 Learning dynamics of the RBM

4.1 Learning dynamics in the thermodynamic limit

A mean field analysis of the learning dynamics has been proposed in [25], in the form of phenomenological equations obtained after averaging over some parameters of the RBM, i.e. by choosing a well defined statistical ensemble of RBMs and using self-averaging properties in the thermodynamic limit. Here we rederive these equations, we add some details and then explore their properties in the light of the preceding section.

B. Thermodynamics of Restricted Boltzmann Machines and Related Learning Dynamics

First we project the gradient ascent equations (4-6) onto the bases $\{u_\alpha(t) \in \mathbb{R}^{N_v}\}$ and $\{v_\alpha(t) \in \mathbb{R}^{N_h}\}$ defined by the SVD of W. Discarding stochastic fluctuations usually inherent to the learning procedure and letting the learning rate $\gamma \to 0$, the continuous version of (4-6) can be recast as follows:

$$\frac{1}{L}\Big(\frac{dW}{dt}\Big)_{\alpha\beta} = \langle s_\alpha \sigma_\beta \rangle_{\text{Data}} - \langle s_\alpha \sigma_\beta \rangle_{\text{RBM}}, \tag{30}$$

$$\frac{1}{\sqrt{L}}\Big(\frac{d\eta}{dt}\Big)_{\alpha} = \langle s_\alpha \rangle_{\text{RBM}} - \langle s_\alpha \rangle_{\text{Data}}, \tag{31}$$

$$\frac{1}{\sqrt{L}}\Big(\frac{d\theta}{dt}\Big)_{\alpha} = \langle \sigma_\alpha \rangle_{\text{RBM}} - \langle \sigma_\alpha \rangle_{\text{Data}}, \tag{32}$$

with s_α and σ_α given in (9). We also have

$$\Big(\frac{dW}{dt}\Big)_{\alpha\beta} = \delta_{\alpha,\beta}\frac{dw_\alpha}{dt} + (1-\delta_{\alpha,\beta})\Big(w_\beta(t)\Omega^v_{\beta\alpha}(t) + w_\alpha(t)\Omega^h_{\alpha\beta}\Big)$$

$$\frac{1}{\sqrt{L}}\Big(\frac{d\eta}{dt}\Big)_{\alpha} = \frac{d\eta_\alpha}{dt} - \sum_\beta \Omega^v_{\alpha\beta}\eta_\beta$$

$$\frac{1}{\sqrt{L}}\Big(\frac{d\theta}{dt}\Big)_{\alpha} = \frac{d\theta_\alpha}{dt} - \sum_\beta \Omega^h_{\alpha\beta}\theta_\beta$$

where

$$\Omega^v_{\alpha\beta}(t) = -\Omega^v_{\beta\alpha} \overset{\text{def}}{=} \frac{d\boldsymbol{u}^{\alpha,T}}{dt}\boldsymbol{u}^\beta$$
$$\Omega^h_{\alpha\beta}(t) = -\Omega^h_{\beta\alpha} \overset{\text{def}}{=} \frac{d\boldsymbol{v}^{\alpha,T}}{dt}\boldsymbol{v}^\beta$$

By eliminating $\left(\frac{dw}{dt}\right)_{\alpha\beta}$, $\left(\frac{d\eta}{dt}\right)_{\alpha}$ and $\left(\frac{d\theta}{dt}\right)_{\alpha}$ we get the following set of dynamical equations:

$$\frac{1}{L}\frac{dw_\alpha}{dt} = \langle s_\alpha \sigma_\alpha \rangle_{\text{Data}} - \langle s_\alpha \sigma_\alpha \rangle_{\text{RBM}} \tag{33}$$

$$\frac{d\eta_\alpha}{dt} = \langle s_\alpha \rangle_{\text{RBM}} - \langle s_\alpha \rangle_{\text{Data}} + \sum_\beta \Omega^v_{\alpha\beta}\eta_\beta \tag{34}$$

$$\frac{d\theta_\alpha}{dt} = \langle \sigma_\alpha \rangle_{\text{RBM}} - \langle \sigma_\alpha \rangle_{\text{Data}} + \sum_\beta \Omega^h_{\alpha\beta}\theta_\beta \tag{35}$$

along with the infinitesimal rotation generators of the left and right singular vectors

$$\Omega^v_{\alpha\beta}(t) = -\frac{1}{w_\alpha + w_\beta}\left(\frac{dW}{dt}\right)^{\text{A}}_{\alpha\beta} + \frac{1}{w_\alpha - w_\beta}\left(\frac{dW}{dt}\right)^{\text{S}}_{\alpha\beta} \tag{36}$$

$$\Omega^h_{\alpha\beta}(t) = \frac{1}{w_\alpha + w_\beta}\left(\frac{dW}{dt}\right)^{\text{A}}_{\alpha\beta} + \frac{1}{w_\alpha - w_\beta}\left(\frac{dW}{dt}\right)^{\text{S}}_{\alpha\beta} \tag{37}$$

where

$$\Big(\frac{dW}{dt}\Big)^{\mathrm{A,S}}_{\alpha\beta} \stackrel{\text{def}}{=} \frac{1}{2}\Big(\langle s_\alpha\sigma_\beta\rangle_{\mathrm{Data}} \pm \langle s_\beta\sigma_\alpha\rangle_{\mathrm{Data}} \mp \langle s_\beta\sigma_\alpha\rangle_{\mathrm{RBM}} - \langle s_\alpha\sigma_\beta\rangle_{\mathrm{RBM}}\Big).$$

The dynamics of learning is now expressed in the reference frame defined by the singular vectors of W. The skew-symmetric rotation generators $\Omega^{v,h}_{\alpha\beta}(t)$ of the basis vectors (induced by the dynamics) tell us how data rotate relatively to this frame. Given the initial conditions, these help us keeping track of the representation of data in this frame. Note that these equations become singular when some degeneracy occurs in W because then the SVD is not uniquely defined. Except from the numerical point of view, where some regularizations might be needed, this does not constitute an issue. In fact only rotations among non-degenerate modes are meaningful, while the rest corresponds to gauge degrees of freedom.

At this point our set of dynamical equations (33-37) is written in a general form. Our goal is to find the typical trajectory of the RBM within a certain statistical ensemble. For this reason, we make the hypothesis that the learning dynamics is represented by a trajectory in the space $\{w_\alpha(t), \eta_\alpha(t), \theta_\alpha(t), \Omega^{v,h}_{\alpha\beta}(t)\}$, while the specific realization of u^α_i, v^α_j and r_{ij} in (8) can be considered irrelevant and only the way they are distributed is important. We are then allowed to perform an average over u^α_i, v^α_j and r_{ij} with respect to some simple distributions, as long as this average is correlated with the data. By this we mean that the components s_α of any given sample are kept fixed while averaging. In the end, what really matters are the strength and the rotation of the SVD modes, respectively determined by $w_\alpha(t)$ and $\Omega^{v,h}_{\alpha\beta}(t)$. As a simplification and also by lack of understanding of what intrinsically drives their evolution, the distributions of u^α_i and v^α_j will be considered stationary in the sequel. Concerning r_{ij}, we allow its variance σ^2/L to vary with time in order to give a minimal description of how the MP bulk evolves during the learning. The detailed dynamics of σ will be derived later in Section 4.3. Using the same notation of Section 3.4 and in particular using the rescaling $v \sim \sqrt{N_h} v^\alpha_i$, the empirical terms take the form:

$$\langle\sigma_\alpha\rangle_{\mathrm{Data}} = \langle (s_\alpha w_\alpha - \theta_\alpha)\big(1 - q_\alpha[\mathbf{s}]\big)\rangle_{\mathrm{Data}} \tag{38}$$

$$\langle s_\alpha\sigma_\beta\rangle_{\mathrm{Data}} = \langle s_\alpha(s_\beta w_\beta - \theta_\beta)\big(1 - q_\beta[\mathbf{s}]\big)\rangle_{\mathrm{Data}} \tag{39}$$

where

$$q_\alpha[\mathbf{s}] \stackrel{\text{def}}{=} \int dx \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} d\mathbf{v} p_\alpha(\mathbf{v}) \tanh^2\Big(\kappa^{-\frac{1}{4}}\big(\sigma x + \sum_\gamma (w_\gamma s_\gamma - \theta_\gamma) v^\gamma\big)\Big),$$

Note that the last equation actually depends on the activation function (hyperbolic tangent in this case), and the term σx corresponds to $\sum_k r_{kj} s_k$ and is obtained by central limit theorem from the independence of the r_{kj}. $q_\alpha[\mathbf{s}]$ is the empirical counterpart of the EA parameters q and q_α already encountered in Section 3.4, and for simple i.i.d. distributions like Gaussian or Laplace it can be estimated easily. The main point here is that the empirical terms (38,39) define operators whose decomposition over the SVD modes of W functionally depends only on w_α, θ_α and on the projection of the data over the SVD modes of W. These terms are driving the dynamics in a precise way. The adaptation of the RBM to this driving force is given by the $\langle\dots\rangle_{\mathrm{RBM}}$ terms in (33,34,35), which can be estimated in the thermodynamic limit (see Section 4.3) as a function of w_α, θ_α and η_α alone, by means of the order parameters $(m_\alpha, \bar{m}_\alpha)$ given

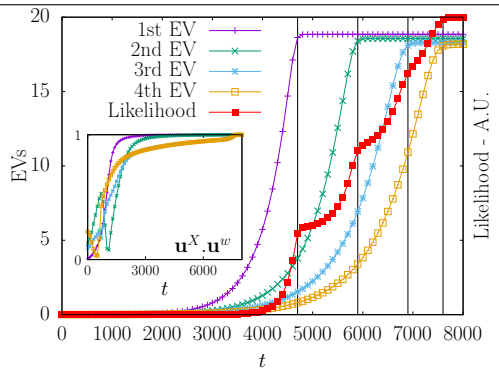


Fig. 3: Time evolution of the eigenvalues and of the likelihood in the linear model. We observe very clearly how the different modes emerge from the bulk and how the likelihood increases with each learned eigenvalue. In the inset, the scalar product of the vectors $\boldsymbol{u}$ obtained from the SVD of the data and from the weights $\boldsymbol{w}$. The $\boldsymbol{u}$s of $\boldsymbol{w}$ are aligned with the SVD of the data at the end of the learning.

in Section 3.2 and once the mean-field equations (16,17) have been solved. Of course, all of this is based on the hypothesis that the RBM stays in the RS domain during learning. Experimental evidence supports this hypothesis (see Section 5).

4.2 Linear instabilities

At the beginning of the learning, the elements of the weight matrix W are usually small; therefore, we can analyze the linear behavior of the RBM in order to understand what happens. In particular, we will see that the dynamics of a non-linear RBM at the beginning of the learning can be understood by looking at the stability analysis of the learning process. The purpose of this analysis is to identify which "deformation modes" of the weight matrix are the most unstable, and how they are related to the input data. Additionally, a good feature of the linear case is that no averaging is needed, the dynamics being actually independent on the particular realization of the components u_i^α and v_j^β. Also, always relative to the linear case, no distinction has to be made between dominant modes and other modes to be treated as the noise component of equation (8), we can simply put all of the modes on the same footing.

Let us analyze the linear regime for an RBM with binary units. The derivation is done by rescaling all the weights and fields by a common "inverse temperature" β and letting this go to zero in equation (4). In principle, the stability analysis would lead to assume both the weights and the magnetizations to be small. However, we can assume only the magnetizations to be small and consider a slightly more general case with no approximations. Such a case is analogous to a linear RBM whose magnetizations undergo Gaussian fluctuations, and it is derived by keeping up to quadratic terms of

the magnetizations in the mean field free energy:

$$
\begin{aligned}
F_{MF}(\mu,\nu) \simeq & \frac{1}{2}\sum_{i=1}^{N}(1+\mu_i)\log(1+\mu_i)+(1-\mu_i)\log(1-\mu_i) \\
& + \frac{1}{2}\sum_{j=1}^{M}(1+\nu_j)\log(1+\nu_j)+(1-\nu_j)\log(1-\nu_j) \\
& - \sum_{i,j}\big(W_{ij}\mu_i\nu_j - \frac{1}{2}W_{ij}^2(\mu_i^2+\nu_j^2)\big) + \sum_{i=1}^{N}\eta_i\mu_i + \sum_{j=1}^{M}\theta_j\nu_j \\
& = \frac{1}{2\sigma_v^2}\sum_{i=1}^{N}\mu_i^2 + \frac{1}{2\sigma_h^2}\sum_{j=1}^{M}\nu_i^2 - \sum_{ij}W_{ij}\mu_i\nu_j + \sum_{i=1}^{N}\eta_i\mu_i + \sum_{j=1}^{M}\theta_j\nu_j.
\end{aligned}
$$

where the variances (σ_v^2, σ_h^2) of respectively visible and hidden variables read ($N_h < N_v$):

$$\sigma_v^{-2} = 1 + \sum_j W_{ij}^2 \simeq 1 + \sum_\alpha w_\alpha^2 \tag{40}$$

$$\sigma_h^{-2} = 1 + \sum_i W_{ij}^2 = 1 + \sum_\alpha w_\alpha^2. \tag{41}$$

We omitted the quadratic term in W_{ij} coming from the TAP contribution to the free energy, which is optional for our stability analysis. In absence of this term the modes evolve strictly independently, while taking it into account leads to a correction to individual variances which couples the modes.

Magnetizations (μ,ν) of visible and hidden variables have now Gaussian fluctuations with covariance matrix

$$C(\mu_v,\mu_h) \stackrel{\text{def}}{=} \begin{bmatrix} \sigma_v^{-2} & -W \\ -W^T & \sigma_h^{-2} \end{bmatrix}^{-1}$$

We can discard the biases of the data and the related fields $(\theta_\alpha, \eta_\alpha)$ with a proper centering of the variables, and we consider equation (33) directly involving the covariance matrix of the data expressed in the frame defined by the SVD modes of W

$$\langle s_\alpha \sigma_\beta\rangle_{\text{Data}} = \sigma_h^2 w_\beta \langle s_\alpha s_\beta\rangle_{\text{Data}}.$$

From $C(\mu_v,\mu_h)$ we get the other terms yielding the following equations:

$$
\begin{aligned}
\frac{dw_\alpha}{dt} &= w_\alpha\sigma_h^2\Big(\langle s_\alpha^2\rangle_{\text{Data}} - \frac{\sigma_v^2}{1-\sigma_v^2\sigma_h^2 w_\alpha^2}\Big) \\
\Omega^{v,h}_{\alpha\beta} &= (1-\delta_{\alpha\beta})\sigma_h^2\Big(\frac{w_\beta - w_\alpha}{w_\alpha + w_\beta} \mp \frac{w_\beta + w_\alpha}{w_\alpha - w_\beta}\Big)\langle s_\alpha s_\beta\rangle_{\text{Data}}
\end{aligned}
$$

Note that these equations are exact for a linear RBM, since they can be derived without any reference to the coordinates of u_α and v_α over which we average in the non-linear regime. These equations tell us that the learning dynamics drives the rotation of the

vectors $\boldsymbol{u}^\alpha$ (and $\boldsymbol{v}^\alpha$) until they are aligned to the principal components of the data, i.e. until $\langle s_\alpha s_\beta \rangle_{\text{Data}}$ becomes diagonal. Calling $\hat{w}_\alpha^2$ the empirical variance of the data, the system reaches the following equilibrium values:

$$w_\alpha^2 = \begin{cases} \frac{\hat{w}_\alpha^2 - \sigma_v^2}{\sigma_v^2 \sigma_h^2 \hat{w}_\alpha^2} & \text{if} \quad \hat{w}_\alpha^2 > \sigma_v^2, \\ 0 & \text{if} \quad \hat{w}_\alpha^2 \leq \sigma_v^2. \end{cases}$$

assuming (σ_v, σ_h) fixed. From this we see that the RBM selects the strongest SVD modes of the data. The linear instabilities correspond to directions along which the variance of the data is above the threshold σ_v^2, and they determine the development of the unstable deformation modes of the weight matrix; during the learning process, these modes will eventually interact following the usual mechanism of non-linear pattern formation encountered for instance in reaction-diffusion processes [34]. Other possible deformations are damped to zero. The linear RBM will therefore learn all the principal components that passed the threshold (up to N_h). Note that this selection mechanism is already known to occur for linear auto-encoders [23] or other similar linear Boltzmann machines [22]. On Fig. 3 we can see the eigenvalues being learned one by one in a linear RBM.

If we take into account the expressions (40,41) for (σ_v, σ_h), we see that the system cannot reach a stable solution except for the case in which all the modes are below the threshold at the beginning. Otherwise the modes that are excited first will eventually grow like $\sqrt{t}$ for a large time, and the excitation threshold will tend to zero for all modes.

In any case, by the definition of a multivariate Gaussian, this simple non-linear analysis describes a unimodal distribution. In order to properly understand the dynamics and the steady-state regime of a non-linear RBM, a well suited mean-field theory is required.

4.3 Non-linear regime

In the linear regime, some specific modes are selected and at some point they start to interact in a non-trivial manner. As seen explicitly in (39), the empirical terms in (4-6) involve higher order statistics of the data and then the Gaussian estimation with $\sigma_v^2 = \sigma_h^2 = 1$ of the RBM response terms $\langle s_\alpha \rangle_{\text{RBM}}$ and $\langle s_\alpha \sigma_\beta \rangle_{\text{RBM}}$ is no longer valid when the interactions kick in. Schematically, the linear regime is valid as long as the RBM is found in the paramagnetic phase. But as soon as one mode passes the linear threshold, the system enters the ferromagnetic phase. Then the proper estimation of the response terms follows from the thermodynamic analysis performed in Section 3, and depends on the assumptions made on the statistical properties of the components of the singular vectors of the weight matrix. In the case of Gaussian i.i.d. components, given the analysis proposed in Section 3.4, we know that the mode with the highest singular value completely dominates the ferromagnetic phase: we expect one single ferromagnetic state characterized by magnetizations aligned to this mode only, while magnetizations correlated to other modes vanish. To be precise, this is the correct picture without fields ($\eta = \theta = 0$) but we don't expect this picture to drastically change in the case of non-vanishing fields. In fact, solving the mean-field equations in presence of the fields show the appearance of meta-stable states correlated with single dominated modes; however, the free energy difference with respect to the ground state, i.e. the state correlated with the mode with the highest singular value,

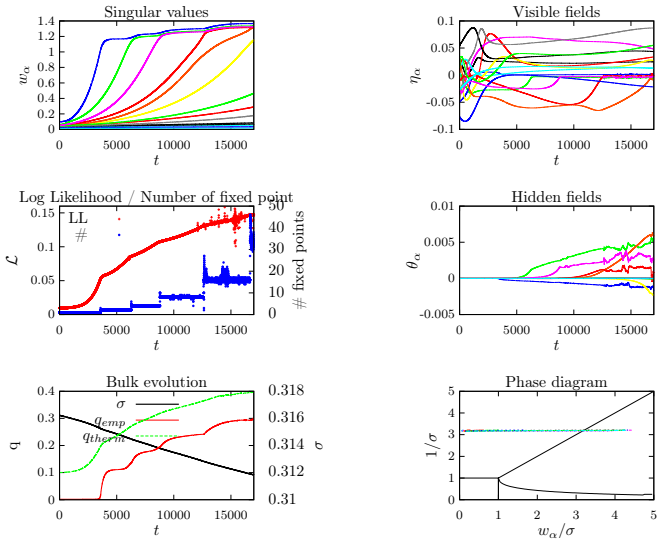


Fig. 4: Predicted mean evolution of an RBM of size $(N_v, N_h) = (1000, 500)$ learned on a synthetic dataset of 10^4 samples of size $N_v = 1000$ obtained from a multimodal distribution with 20 clusters randomly defined on a submanifold of dimension $d = 15$. The dynamics follows the projected magnetizations in this reduced space with help of 15 modes. We observe a kind of pressure on top singular values from lower ones.

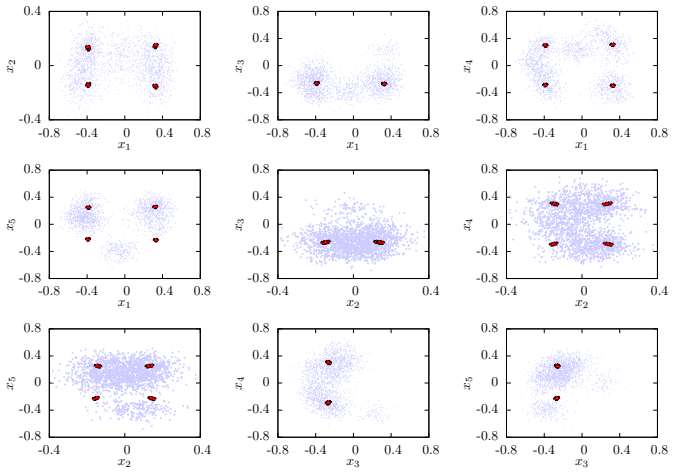


Fig. 5: Scatter plots of the mean-field magnetizations (in red) and the samples (in blue) in various plan projections defined by pairs of left eigenvectors of W. This case corresponds to an RBM of size $(N_v, N_h) = (100, 50)$ learned on a synthetic dataset of 10^4 samples of size $N_v = 100$ obtained from a multimodal distribution with 11 clusters randomly defined on a submanifold of dimension $d = 5$. The scatter plot is obtained at a point where 5 modes have already condensed and 16 saddle point solutions have been found.

is of order $O\big(L(w_\alpha - w_{max})\big)$, which means that the contribution of those meta-stable states become rapidly negligible with large system size.

To draw a realistic picture of the learning process we now consider Laplace i.i.d. components for the SVD modes that, as seen in Section 3.4, allow the ferromagnetic phase to be of compositional type. The reason for this is that the Laplace distribution leads to less interference among modes than the Gaussian distribution, so that the modes will weakly interact in the mean-field equations. Solving equations (21,22,27,29) in absence of fields yields the following picture: one fixed point solution will typically have non-vanishing magnetizations $\{m_\alpha, \bar{m}_\alpha\}$ for all α such that $w_\alpha \in [w_{max} - \Delta w, w_{max}]$, where Δw is approximately the gap $\Delta w(q, \bar{q})$ defined in (26). This solution is a degenerate ground state, all other solutions being obtained by independently reversing the signs of the condensed magnetizations $(m_\alpha, \bar{m}_\alpha)$. Hence for K condensed modes we get a degeneracy of 2^K. When the fields are included, all these fixed points are displaced in the direction of the fields, and some of them may disappear. In the end we are left with a potentially large amount of nearly degenerate states able to cover the empirical distribution of the data, at least in some simple cases.

Coming back to the learning dynamics the terms corresponding to the response of

the RBM in (4,6) are estimated in the thermodynamic limit by means of the previously defined order parameters:

$$\langle s_\alpha \rangle_{\mathrm{RBM}} = \frac{1}{Z_{\mathrm{Therm}}} \sum_\omega e^{-Lf(m^\omega, \bar{m}^\omega, q^\omega, \bar{q}^\omega)} \bar{m}_\alpha^\omega \stackrel{\mathrm{def}}{=} \langle \bar{m}_\alpha \rangle_{\mathrm{Therm}},$$

$$\langle s_\alpha \sigma_\beta \rangle_{\mathrm{RBM}} = \frac{1}{Z_{\mathrm{Therm}}} \sum_\omega e^{-Lf(m^\omega, \bar{m}^\omega, q^\omega, \bar{q}^\omega)} \bar{m}_\alpha^\omega m_\beta^\omega \stackrel{\mathrm{def}}{=} \langle \bar{m}_\alpha m_\beta \rangle_{\mathrm{Therm}}.$$

Here $\langle \dots \rangle_{\mathrm{Therm}}$ denotes the thermodynamical average and the partition function is expressed, in the thermodynamic limit, as

$$Z_{\mathrm{Therm}} \stackrel{\mathrm{def}}{=} \sum_\omega e^{-Lf(m^\omega, \bar{m}^\omega, q^\omega, \bar{q}^\omega)}$$

The index ω runs over all the stable fixed point solutions of (16,17) weighted accordingly to the free energy given by (15). These are the dominant contributions as long as free energy differences are $O(1)$, and the internal fluctuations given by each fixed point are comparatively of order $O(1/L)$. In addition, the dynamics of the bulk can be characterized by empirically defining σ^2:

$$\sigma^2 = \frac{1}{L} \sum_{ij} r_{ij}^2,$$

whose evolution is:

$$\frac{d\sigma^2}{dt} = \frac{1}{L} \sum_{ij} r_{ij} \frac{dW_{ij}}{dt},$$

$$= \frac{1}{L} \sum_{ij} r_{ij} \left[\langle s_i \tanh\Big(\sum_k r_{kj} s_k + \kappa^{-\frac{1}{4}} \sum_\alpha (w_\alpha s_\alpha - \theta_\alpha) v_j^\alpha \sqrt{L} \Big) \rangle_{\mathrm{Data}} - \langle s_i \sigma_j \rangle_{\mathrm{RBM}} \right]$$

given the independence of r_{i*} (resp. r_{*j}) and u_i^α (resp. v_i^α).

Exploiting the self-averaging properties of both the empirical and the response terms with respect to r_{ij}, u_i^α and v_j^α yields

$$\frac{1}{L^2} \sum_{ij} r_{ij} \langle s_i \sigma_j \rangle_{\mathrm{Data}} = \frac{\sigma^2}{L} \big(1 - \langle q[\mathbf{s}] \rangle_{\mathrm{Data}} \big)$$

$$\frac{1}{L^2} \sum_{ij} r_{ij} \langle s_i \sigma_j \rangle_{\mathrm{RBM}} = \frac{\sigma^2}{L} \big(1 - \langle q \rangle_{\mathrm{Therm}} \big),$$

with

$$q[\mathbf{s}] \stackrel{\mathrm{def}}{=} \int dx \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} d\mathbf{v} p(\mathbf{v}) \tanh^2 \Big(\kappa^{-\frac{1}{4}} \big(\sigma x + \sum_\gamma (w_\gamma s_\gamma - \theta_\gamma) v^\gamma \big) \Big).$$

Summarizing, our equations take the suggestive form

$$\frac{1}{L}\frac{dw_\alpha}{dt} = \langle s_\alpha(w_\alpha s_\alpha - \theta_\alpha)(1-q_\alpha[\mathbf{s}])\rangle_{\text{Data}} - \langle \bar{m}_\alpha(w_\alpha \bar{m}_\alpha - \theta_\alpha)(1-q_\alpha)\rangle_{\text{Therm}}, \tag{42}$$

$$\frac{d\eta_\alpha}{dt} = \langle \bar{m}_\alpha\rangle_{\text{Therm}} - \langle s_\alpha\rangle_{\text{Data}} + \sum_\beta \Omega^v_{\alpha\beta}\eta_\beta, \tag{43}$$

$$\frac{d\theta_\alpha}{dt} = \langle (w_\alpha \bar{m}_\alpha - \theta_\alpha)(1-q_\alpha)\rangle_{\text{Therm}} - \langle (w_\alpha s_\alpha - \theta_\alpha)(1-q_\alpha[\mathbf{s}])\rangle_{\text{Data}} + \sum_\beta \Omega^h_{\alpha\beta}\theta_\beta, \tag{44}$$

$$\frac{d\sigma^2}{dt} = \sigma^2\Big(\langle q\rangle_{\text{Therm}} - \langle q[\mathbf{s}]\rangle_{\text{Data}}\Big), \tag{45}$$

with $\Omega^{v,h}$ taking the form of a difference between a data averaging $\langle\dots\rangle_{\text{Data}}$ and a thermodynamical averaging $\langle\dots\rangle_{\text{Therm}}$ involving only order parameters. Note here that the w_α variables, with respect to the other variables, evolve on a faster time scale. This is our final and main result, which might possibly help improving current learning algorithms of RBMs. From this, it is clear what the learning of an RBM is aimed at: the equations will converge once the dataset is clustered in such a way that each cluster is represented by a solution of the mean-field equations with magnetizations $\bar{m}_\alpha$ and EA parameters q_α corresponding respectively to their empirical counterparts $\langle s_\alpha\rangle$ and $\langle q_\alpha[\mathbf{s}]\rangle$ representing cluster magnetization and variance. In particular, these clusters can somehow be regarded as the attractors in the context of feed-forward networks, defining a partition of the data. This can be seen by starting from random configurations and letting the system evolve using the TAP equations or a MCMC method. At the end the system will end up in one of those clusters (characterized by a fixed point of the mean-field equations). Note that this is the reason why the RBM needs to reach a ferromagnetic phase with many states to be able to match the empirical term in (4) and reach convergence.

Additionally, the log likelihood (3) can be estimated in the thermodynamic limit (after normalization by L).

$$\mathcal{L} = \Big\langle \sqrt{\kappa}\mathsf{E}_{x,v}\Big[\log\cosh\Big(\kappa^{-\frac{1}{4}}\Big(\sigma x + \sum_\alpha (w_\alpha s_\alpha - \theta_\alpha)v^\alpha\Big)\Big)\Big]\Big\rangle_{\text{Data}}$$
$$- \Big\langle \sum_\alpha \eta_\alpha s_\alpha\Big\rangle_{\text{Data}} - \frac{1}{L}\log\big(Z_{\text{Therm}}\big),$$

As an example, for a multimodal data distribution with a finite number of clusters embedded in a high dimensional configuration space, the SVD modes of W that will develop are the one pointing to the directions of the magnetizations defined by these clusters (which will be almost surely orthogonal, given the high dimensionality of the embedding space). In this simple case the RBM will evolve, as in the linear case, to a state in which the empirical term becomes diagonal, while the singular values will adjust to match the proper magnetization in each fixed point.

We have integrated equations (42,43,44,45,36,37) in simple cases by using the Laplace averaging of the components of the SVD modes and using for the EA parameters the expressions given in (27,29). Basically, the hidden distribution to be

modeled is defined by

$$P(s) = \sum_{c=1}^{C} p_c \prod_{i=1}^{N} \frac{e^{h_i^c s_i}}{2\cosh(h_i^c)}, \tag{46}$$

i.e. a multimodal distribution composed of C clusters of independent variables, where the magnetization of each variable i in cluster c is given by $m_i^c = \tanh(h_i^c)$. Each cluster is weighted by some probability p_c. In addition we assume these magnetization vectors m^c to be embedded in a low dimensional space of dimension $d << N$. d defines the rank of W. The initial conditions for W are such that the left singular vectors $\{u_\alpha, \alpha = 1, \dots d\}$ span this low dimensional space. An example of the typical dynamics obtained in the case at hand is shown in Figure 4. In contrast to the linear problem where singular values evolve independently, here we distinctively witness the interaction between singular values: a kind of pressure is exerted by lower modes on higher ones resulting in successive bumps in the dynamics of the top modes. The number of states is roughly multiplied by two each time a mode condenses and get close enough to the top modes. Concerning the dynamics of the fields, we don't really observe convergence towards stable directions. Some (possibly numerical) instability is observed when many modes condense, with both the fields and the number of fixed point solutions becoming very noisy. It is also interesting to see how the magnetizations related to the states are distributed with respect to the dataset. On Figure 5 we see that the fixed points tend (as expected) to settle within dense regions of sample points. However, our coarse description shows some limitations for more complex situations, the number of adjustable parameters being too limited to be able to match arbitrary distributions of clusters. It is then appropriate to think about this behaviour in a mean sense; at least, it is able to reproduce a realistic learning dynamics of the singular values of the weight matrix.

5 Numerical Experiments

Given the comprehensive theoretical analysis of the RBM model given in the previous sections, we are now able to provide a meaningful description of the learning dynamics for a RBM trained with k-steps contrastive divergence (CDk) [4]. The observations presented in this section will serve as a validation for the theoretical analysis. First, to provide a more direct comparison to section 4.3, we will look at the learning dynamics of an RBM trained on a set of simple synthetic data. Subsequently, we will test the model against real world data by training on the MNIST dataset.

5.1 Synthetic dataset

As a simple case, we trained the RBM over the same dataset defined in fig. 4, derived from the simple multimodal distribution in eq. 46 (see Appendix B for details). Thus we set $N_v = 1000$, $N_h = 500$ and we trained using 10^4 samples with an effective dimension $d = 15$ organized in 20 separate clusters. The weights are initialized from a Gaussian distribution with standard deviation $\sigma = 10^{-3}$, while the hidden bias is initialized to 0 and the visible bias is initialized with the empirical mean of the data

$$\eta_i = \frac{1}{2}\log\left(\frac{p_i}{1-p_i}\right)$$

where p_i is the empirical probability of activation for the i_{th} hidden node.

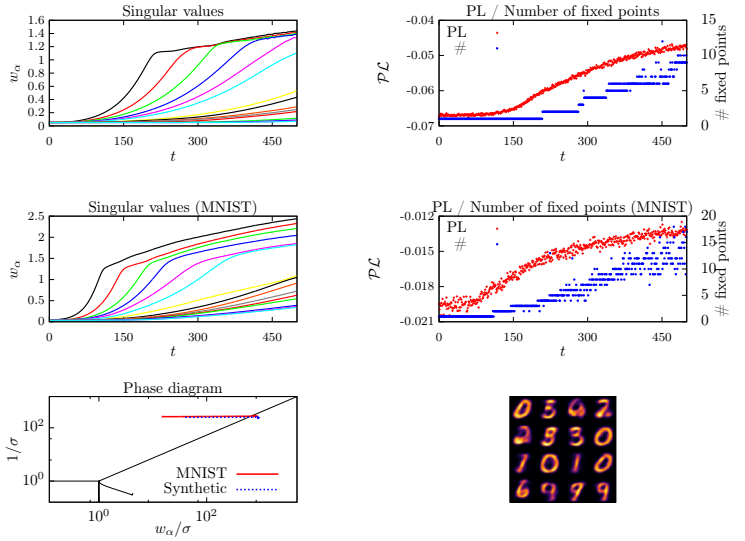


Fig. 6: Experimental evolution of an RBM during training for a synthetic dataset (top plots, to compare to Fig. 4) and for MNIST (central plots). The bottom left plot shows the learning trajectories in the phase diagram, while the bottom right image shows some examples of fixed point solutions for MNIST (we note the presence of some spurious fixed points).

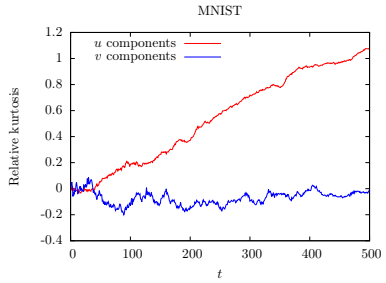


Fig. 7: Relative kurtosis of the components of the modes after training on MNIST.

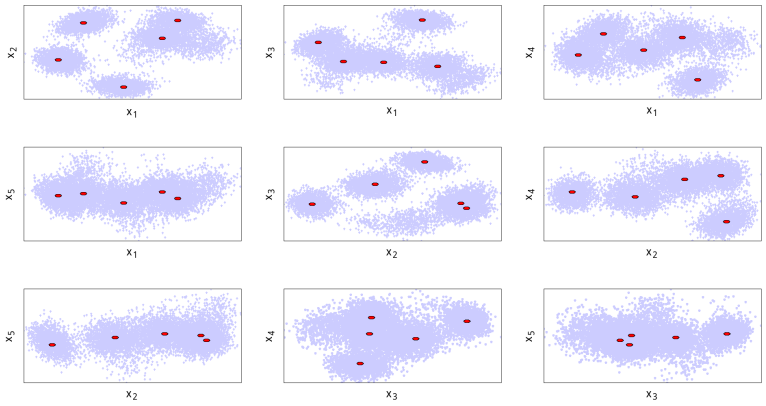


Fig. 8: Scatter plots of samples (blue) and fixed points (red) in various plan projections defined by pairs of left eigenvectors of W. The dataset is the same as in Fig. 5 and in this case 5 modes have condensed and 7 fixed point solutions have been found.

Finally, the training set is divided into batches of size 20, 5 Gibbs sampling steps are used (CD5) and the learning rate γ is kept low in order to reduce noise, $\gamma = 5 \times 10^{-8}$. The results of the analysis are shown in fig. 6. We see that the dynamics of the singular values obtained by direct integration of the mean-field equations (Fig. 4) are very well reproduced, the only difference being a slightly higher pressure on the strongest modes. The number of fixed point solutions also seems to follow the same trend but more noise is present, an indication of the fact that the RBM has a tendency to learn spurious fixed points during the training. The learning trajectory on the phase diagram is also of interest; we see that the RBM is initialized in the paramagnetic state as expected and the effect of the learning is to drive the model to the ferromagnetic phase. Once in the ferromagnetic phase, the trajectory slows down and the model is assessed near the critical line between paramagnetic and ferromagnetic states, where the estimate of the weights is most stable (according to [35]). Finally, in Fig. 8 we see how the RBM is able to generate a proper clustering of the data over the spectral modes. In particular, the TAP fixed points of the trained model are well distributed and able to cover the full data distribution, improving over the typical behaviour for Laplace distributed weights that emerged with our theoretical analysis (Fig. 5).

5.2 MNIST dataset

The MNIST dataset is composed by 70000 handwritten digits (60000 for training, 10000 for testing) of size 28×28 pixels. Being highly multimodal, we expect this dataset to push the limits of our spectral analysis. For the training, the initialization of the model is the same one used for the synthetic data, 10000 training samples are used (taken at random from the dataset) and the values of the other hyperparameters

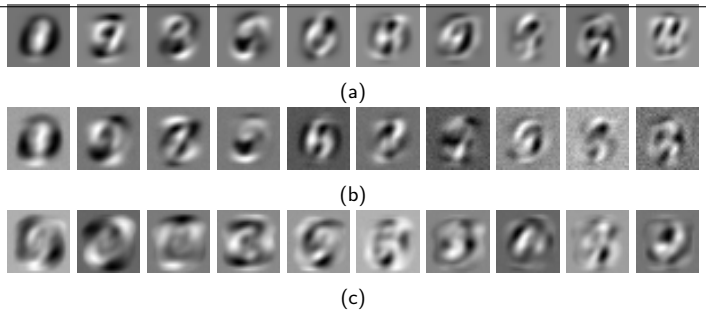


Fig. 9: **(a)** Principal components extracted from the training set (starting from the second, as the first one is encoded into the visible bias). **(b)** The first 10 modes of a RBM trained for 1 epoch (with $\gamma \simeq 0.1$). **(c)** Same as (b) but after a 10 epochs training.

are as follows: $N_v = 784$, $N_h = 100$, batch size $= 20$, $\gamma = 5 \times 10^{-7}$. With respect to the linear regime (described in section 4.2) we see in Fig. 9 how the RBM is able to learn the SVD of the dataset quite precisely at the beginning of the training, then the learning dynamics quickly enter the non-linear regime. Even in this highly multimodal scenario, our findings over simple synthetic data seem to be confirmed, as seen in Fig. 6. The high number of modes, however, determines an increase in the magnitude of the singular values of condensed modes and seems to destabilize a bit the learning, making the computation of fixed points less reliable. In fact, as a high number of modes are condensing, the model is not able to get rid of all the spurious fixed points. This problem can be mitigated by using an even smaller learning rate, at the cost of slowing down the training. Probably, using a variable learning rate could be a more practical solution (descreasing the learning rate from time to time to let the model eliminate unneeded fixed points). Concerning the (relative) kurtosis of the mode components distributions, we did not observe a very stable and systematic behavior. Either we see small fluctuations around zero, either some excursions occur and a finite value in the range $[0, 3]$ is building up either for the u or the v components, coherently to the compositional phase interpretation given previously. The latter is the case for MNIST, as shown in Fig. 7. Additionally the transverse part of the fields, meaning orthogonal to the condensed modes, is usually not completely negligible, in contrary to what we assume in (13,14). This clearly constitutes a limitation of our analysis. These transverse components offer more flexibility for generating and selecting fixed points and interfere in some non-trivial way with the kurtosis property, which possibly explains why we don't get a systematic behavior.

6 Discussion

Before drawing some perspectives, let us summarize the main outcomes of the present work:

- **(i) thermodynamic properties of realistic RBMs:** our analysis focused on a non-i.i.d. ensemble of weight matrices, whose derivation has been inspired

by empirical observations obtained by training RBMs on real data.

- **(ii) RS equations and compositional phase**: we found a way of writing the RS equations for the RBM (in particular with equations (21,22,23,24)) which leads to a simple characterization of the ferromagnetic phase where the RBM is assumed to operate. Schematically, a negative relative kurtosis for the distribution of the singular vectors' components favors the proliferation of metastable states, while a positive one tends to favor a compositional phase. In particular, we were able to precisely address a concrete case presenting the compositional phase by considering a Laplace distribution for the singular vectors' components.
- **(iii) a set of equations representing a typical learning dynamics** that defines a trajectory in $\{w_\alpha(t), \eta_\alpha(t), \theta_\alpha(t), \Omega^{v,h}_{\alpha\beta}(t), \sigma^2(t)\}$. The spectrum of the dominant singular values, represented by $\{w_\alpha(t)\}$ and expressing the information content of the RBM, is playing the main role. The bulk of dominated modes corresponding to noise sees its dynamics summarized by the evolution of $\sigma^2(t)$. Rotations of dominant singular vectors during the learning process are given by $\Omega^{v,h}$ while the projections of the biases along the main modes are given by η and θ. These equations have been obtained by averaging over the components of left and right SVD vectors of the weight matrix, while keeping fixed the quantities considered to be relevant. This averaging actually corresponds to a standard self-averaging assumption in a RS phase.
- **(iv) a clustering interpretation of the training process** is obtained through equations (42,43,44,45) where it is explicitly shown the kind of matching that the RBM is trying to perform between the order parameters obtained from the fixed point solutions and their empirical counterparts in the non-linear regime. A natural clustering of the data can actually be defined by assigning to each sample the fixed point obtained after initializing the fixed point equations with a visible configuration corresponding to that same sample.

The main picture emerging from the present analysis is that of a set of clusters corresponding to the fixed points of the RBM, which try to uniformly cover the support of the dataset. A full understanding of the mechanism by which the RBM manages to properly cover the dataset is still lacking, even though the case of Laplace distributed singular vectors' components gives some insights. By comparison, real RBMs have more flexibility than the simple "mean Laplace RBM" considered in Section 3.4 and they can produce a good covering of the data manifold. We were not yet able to precisely pinpoint the main ingredients for that mechanism, even though we suspect the transverse biases (orthogonal to the modes) of the hidden units to be the missing ingredient in our analysis.

From the theoretical point of view we would like to see how these results can be adapted to more complex models like DBM or generative models based on convolutional networks. In particular we would like to understand whether adding more layers can facilitate the covering of the dataset by fixed points. From the practical point of view these results might help to orientate the choice of the hyper-parameters used for training an RBM and to refine the criteria for assessing the quality of a learned RBM. For instance, the choice of the number of hidden variables is dictated by two considerations: the effective rank of W, i.e. the number of relevant modes to be considered, and the level of interaction between these modes. Using less hidden variables gives more compact RBMs and reduces the rank of W to its needed value, but it also leads to modes with stronger interactions, which means less flexibility for generating a good covering of fixed points.

A AT line

The stability of the RS solution to the mean-field equations is studied along the lines of [33] by looking at the Hessian of the replicated version of the free energy and identifying eigenmodes from symmetry arguments. Before taking the limit $p \to 0$ the free energy reads

$$f[m, \bar{m}, Q, \bar{Q}] = \sum_{a,\alpha} w_\alpha m_\alpha^a \bar{m}_\alpha^a + \frac{\sigma^2}{2} \sum_{a \neq b} Q_{ab} \bar{Q}_{ab} - \frac{1}{\sqrt{\kappa}} A_p[m, Q] - \sqrt{\kappa} B_p[\bar{m}, \bar{Q}],$$

with A_p and B_p given in (10,11). Assuming the small perturbations

$$m_\alpha^a = m_\alpha + \epsilon_\alpha^a \qquad \bar{m}_\alpha^a = \bar{m}_\alpha + \bar{\epsilon}_\alpha^a$$

$$Q_{ab} = q + \eta_{ab} \qquad \bar{Q}_{ab} = \bar{q} + \bar{\eta}_{ab},$$

around the saddle point $(m_\alpha, \bar{m}_\alpha, q, \bar{q})$, the perturbed free energy reads

$$\begin{aligned}
\Delta f = & \sum_{a,\alpha} w_\alpha \bar{\epsilon}_\alpha^a \epsilon_\alpha^a + \frac{\sigma^2}{2} \sum_{a \neq b} \bar{\eta}_{ab} \eta_{ab} + \sum_{a,b,\alpha,\beta} \Big[\big(\delta_{ab} \bar{A}_{\alpha\beta} + \bar{\delta}_{ab} \bar{B}_{\alpha\beta} \big) \epsilon_\alpha^a \epsilon_\beta^b + CT \Big] \\
& + \sum_{a \neq b, c, \alpha} \Big[\big((\delta_{ab} + \delta_{ac}) \bar{C}_\alpha + (1 - \delta_{ac} - \delta_{bc}) \bar{D}_\alpha \big) \epsilon_\alpha^c \eta_{ab} + CT \Big] \\
& + \sum_{a \neq b, c \neq d} \Big[\big(\delta_{(ab)(cd)} \bar{E}_0 + \mathbb{1}_{\{a \in (cd) \oplus b \in (cd)\}} \bar{E}_1 + \mathbb{1}_{\{(ab) \cap (cd) = \emptyset\}} \bar{E}_2 \big) \eta_{ab} \eta_{cd} + CT \Big],
\end{aligned}$$

where CT means "conjugate term" in the sense $\epsilon \leftrightarrow \bar{\epsilon}$, $A_{\alpha\beta} \leftrightarrow \bar{A}_{\alpha\beta} \ldots$, where $\bar{\delta}_{ab} \overset{\text{def}}{=} 1 - \delta_{ab}$ and the operators are given by

$$A_{\alpha\beta} \overset{\text{def}}{=} (\delta_{\alpha\beta} - m_\alpha m_\beta) w_\alpha w_\beta \qquad B_{\alpha\beta} \overset{\text{def}}{=} \Big(\mathsf{E}_{x,v} \big(v^\alpha v^\beta \tanh^2(\bar{h}(x,v)) \big) - m_\alpha m_\beta \Big) w_\alpha w_\beta$$

$$C_\alpha \overset{\text{def}}{=} \frac{\kappa^{1/4} \sigma^2}{2} m_\alpha (1 - q) w_\alpha \qquad D_\alpha \overset{\text{def}}{=} \frac{\kappa^{1/4} \sigma^2}{2} \Big(\mathsf{E}_{x,v} \big(v^\alpha \tanh^3(\bar{h}(x,v)) \big) - m_\alpha q \Big) w_\alpha$$

$$E_0 \overset{\text{def}}{=} \frac{\sqrt{\kappa} \sigma^4}{4} (1 - q^2) \qquad E_1 \overset{\text{def}}{=} \frac{\sqrt{\kappa} \sigma^4}{4} q (1 - q) \qquad E_2 \overset{\text{def}}{=} \frac{\sqrt{\kappa} \sigma^4}{4} \Big(\mathsf{E}_{x,v} \big(\tanh^4(\bar{h}(x,v)) \big) - q^2 \Big)$$

with

$$h(x, u) \overset{\text{def}}{=} \kappa^{1/4} \Big(\sqrt{q} \sigma x + \sum_\alpha (m_\alpha w_\alpha - \eta_\alpha) u^\alpha \Big),$$

Conjugate quantities are obtained by replacing m_α by $\bar{m}_\alpha$, q by $\bar{q}$, u^α by v^α, η_α by θ_α and κ by $1/\kappa$. As for the SK model, the $2Kp \times 2Kp$ Hessian thereby defined can be diagonalized with the help of three similar sets of eigenmodes corresponding to different permutation symmetries in replica space.

The first set corresponds to $2K + 2$ replica symmetric modes defined by $\eta_\alpha^a = \eta_\alpha$

and $\eta_{ab} = \eta$ solving the linear system

$$\Big(\frac{w_\alpha}{2} - \lambda\Big)\bar\epsilon_\alpha - \frac{1}{2}\bar A_{\alpha\alpha}\epsilon_\alpha + \sum_\beta \Big(\bar A_{\alpha\beta} + (p-1)\bar B_{\alpha\beta}\Big)\epsilon_\beta + \Big((p-1)\bar C_\alpha + \frac{(p-1)(p-2)}{2}\bar D_\alpha\Big)\eta = 0$$

$$\Big(\frac{w_\alpha}{2} - \lambda\Big)\epsilon_\alpha - \frac{1}{2}A_{\alpha\alpha}\bar\epsilon_\alpha + \sum_\beta \Big(A_{\alpha\beta} + (p-1)B_{\alpha\beta}\Big)\bar\epsilon_\beta + \Big((p-1)C_\alpha + \frac{(p-1)(p-2)}{2}D_\alpha\Big)\bar\eta = 0$$

$$\Big(\frac{\sigma^2}{2} - \lambda\Big)\bar\eta + \sum_\alpha \Big(\bar C_\alpha + \frac{p-2}{2}\bar D_\alpha\Big)\epsilon_\alpha + 2\Big(\bar E_0 + 2(p-2)\bar E_1 + \frac{(p-2)(p-3)}{2}\bar E_2\Big)\eta = 0$$

$$\Big(\frac{\sigma^2}{2} - \lambda\Big)\eta + \sum_\alpha \Big(C_\alpha + \frac{p-2}{2}D_\alpha\Big)\bar\epsilon_\alpha + 2\Big(E_0 + 2(p-2)E_1 + \frac{(p-2)(p-3)}{2}E_2\Big)\bar\eta = 0$$

with eigenvalue λ solving a polynomial equation of degree $2K+2$ corresponding to a vanishing determinant in the above system.

The second set corresponds to a broken replica symmetry where one replica a_0 is different from the others

$$(\epsilon^a_\alpha, \bar\epsilon^a_\alpha) = \begin{cases} (\epsilon_\alpha, \bar\epsilon_\alpha) & \text{for } a \neq a_0 \\ (1-p)(\epsilon_\alpha, \bar\epsilon_\alpha) & \text{for } a = a_0 \end{cases} \qquad (\eta_{ab}, \bar\eta_{ab}) = \begin{cases} (\eta, \bar\eta) & \text{for } a, b \neq a_0 \\ (1-\frac{p}{2})(\eta, \bar\eta) & \text{for } a = a_0 \text{ or } b = a_0 \end{cases}$$

This set has dimension $(2K+2)(p-1)$. Its parameterization is obtained by imposing orthogonality with the previous one. The corresponding system reads

$$\Big(\frac{w_\alpha}{2} - \lambda\Big)\bar\epsilon_\alpha - \frac{1}{2}\bar A_{\alpha\alpha}\epsilon_\alpha + \sum_\beta (\bar A_{\alpha\beta} - \bar B_{\alpha\beta})\epsilon_\beta + \frac{p-2}{2}\Big(\bar C_\alpha - \bar D_\alpha\Big)\eta = 0$$

$$\Big(\frac{w_\alpha}{2} - \lambda\Big)\epsilon_\alpha - \frac{1}{2}A_{\alpha\alpha}\bar\epsilon_\alpha + \sum_\beta (A_{\alpha\beta} - B_{\alpha\beta})\bar\epsilon_\beta + \frac{p-2}{2}\Big(C_\alpha - D_\alpha\Big)\bar\eta = 0$$

$$\Big(\frac{\sigma^2}{2} - \lambda\Big)\bar\eta + \sum_\alpha (\bar C_\alpha - \bar D_\alpha)\epsilon_\alpha + 2\Big(\bar E_0 + (p-4)\bar E_1 - (p-3)\bar E_2\Big)\eta = 0$$

$$\Big(\frac{\sigma^2}{2} - \lambda\Big)\eta + \sum_\alpha (C_\alpha - D_\alpha)\bar\epsilon_\alpha + 2\Big(E_0 + (p-4)E_1 - (p-3)E_2\Big)\bar\eta = 0$$

Finally the eigenmodes of the Hessian are made complete by considering a broken symmetry where two replicas a_0 and a_1 are different from the others, with the following parameterization dictated again by orthogonality constraints with the previous sets:

$$(\epsilon^a_\alpha, \bar\epsilon^a_\alpha) = 0, \qquad (\eta_{ab}, \bar\eta_{ab}) = \begin{cases} (\eta, \bar\eta) & \text{for } a, b \neq a_0 \\ \frac{3-p}{2}(\eta, \bar\eta) & \text{for } a \in a_0, a_1 \text{ or } b \in a_0, a_1 \\ \frac{(p-2)(p-3)}{2}(\eta, \bar\eta) & \text{for } (a, b) = (a_0, a_1). \end{cases}$$

The dimension of this set is now $p(p-3)$, and it represents eigenvectors iff the following

system of equations is satisfied

$$\big(\frac{\sigma^2}{2}-\lambda\big)\bar{\eta}+2(\bar{E}_0-2\bar{E}_1+\bar{E}_2)\eta=0$$

$$\big(\frac{\sigma^2}{2}-\lambda\big)\eta+2(E_0-2E_1+E_2)\bar{\eta}=0$$

The corresponding eigenvalues read

$$\lambda=\frac{\sigma^2}{2}\pm 2\sqrt{(\bar{E}_0-2\bar{E}_1+\bar{E}_2)(E_0-2E_1+E_2)},$$

with degeneracy $p(p-3)/2$. Finally the RS stability condition reads

$$\frac{1}{\sigma^2}>\sqrt{\mathsf{E}_{x,u}\Big(\mathrm{sech}^4\big(h(x,u)\big)\Big)\mathsf{E}_{x,v}\Big(\mathrm{sech}^4\big(\bar{h}(x,v)\big)\Big)},$$

which reduces to the same form of the AT line for the SK model when $\kappa=1$, except for the u and v averages that are specific to our model. As seen in Figure 2 the influence of κ is very limited.

B Synthetic dataset

The multimodal distribution modeling the N-dimensional synthetic data is

$$P(s)=\sum_{c=1}^{C}p_c\prod_{i=1}^{N}\frac{e^{h_i^c s_i}}{2\cosh(h_i^c)}, \tag{47}$$

where C is the number of clusters, p_c is a weight and $\boldsymbol{h}^c$ is a hidden field for cluster c. The values for p_c are taken at random and normalized, while to compute h_i^c we take into account the magnetizations $m_i^c=\tanh(h_i^c)$. Expanding over the spectral modes, we can set an effective dimension d by constraining the sum to the range $\alpha=1,\dots,d$

$$m_i^c=\sum_{\alpha=1}^{d}m_\alpha^c u_i^\alpha \tag{48}$$

Clusters' magnetizations m_α^c are drawn at random between $[-1,1]$ and normalized with the factor

$$Z=\sqrt{\frac{\sum_\alpha m_\alpha^2}{d\cdot r}},\quad r=\tanh(\eta) \tag{49}$$

where r is introduced to decrease the clusters' polarizations (in our simulations, we used $\eta=0.3$). The spectral basis u_i^α is obtained by drawing at random d N-dimensional vectors and applying the Gram-Schmidt process (which can be safely employed as N is supposedly big and thus the initial vectors are nearly orthogonal). The hidden fields are then obtained from the magnetizations

$$h_i^c=\tanh^{-1}(m_i^c) \tag{50}$$

and the samples are generated by choosing a cluster according to p_c and setting the visible variables to ±1 according to

$$p(s_i=1)=\frac{1}{1+e^{-2h_i^c}} \tag{51}$$

References

[1] P. Smolensky. *In Parallel Distributed Processing: Volume 1 by D. Rumelhart and J. McLelland*, chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory. 194-281. MIT Press, 1986.

[2] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *Artificial Intelligence and Statistics*, pages 448–455, 2009.

[3] G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[4] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14:1771–1800, 2002.

[5] T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1064–1071, New York, NY, USA, 2008. ACM.

[6] G.E. Hinton. *A Practical Guide to Training Restricted Boltzmann Machines*, pages 599–619. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[7] D.S.P. Salazar. Nonequilibrium thermodynamics of restricted Boltzmann machines. *Phys. Rev. E*, 96:022131, 2017.

[8] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, 1982.

[9] D. J. Amit, H. Gutfreund, and H. Sompolinsky. Statistical mechanics of neural networks near saturation. *Annals of Physics*, 173(1):30–67, 1987.

[10] E. Gardner. Maximum storage capacity in neural networks. *EPL (Europhysics Letters)*, 4(4):481, 1987.

[11] E. Gardner and B. Derrida. Optimal storage properties of neural network models. *Journal of Physics A: Mathematical and General*, 21(1):271, 1988.

[12] B. Barra, A. Bernacchia, E. Santucci, and P. Contucci. On the equivalence of Hopfield networks and Boltzmann machines. *Neural Networks*, 34:1–9, 2012.

[13] G. Marylou, E.W. Tramel, and F. Krzakala. Training restricted Boltzmann machines via the Thouless-Anderson-Palmer free energy. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 640–648, 2015.

[14] H. Huang and T. Toyoizumi. Advanced mean-field theory of the restricted Boltzmann machine. *Physical Review E*, 91(5):050101, 2015.

[15] C. Takahashi and M. Yasuda. Mean-field inference in gaussian restricted Boltzmann machine. *Journal of the Physical Society of Japan*, 85(3):034001, 2016.

[16] C. Furtlehner, J.-M. Lasgouttes, and A. Auger. Learning multiple belief propagation fixed points for real time inference. *Physica A: Statistical Mechanics and its Applications*, 389(1):149–163, 2010.

[17] A. Barra, G. Genovese, P. Sollich, and D. Tantari. Phase diagram of restricted Boltzmann machines and generalized Hopfield networks with arbitrary priors. arXiv:1702.05882, 2017.

[18] H. Huang. Statistical mechanics of unsupervised feature learning in a restricted Boltzmann machine with binary synapses. *Journal of Statistical Mechanics: Theory and Experiment*, 2017(5):053302, 2017.

[19] E. Agliari, A. Barra, A. Galluzzi, F. Guerra, and F. Moauro. Multitasking associative networks. *Phys. Rev. Lett.*, 109:268101, 2012.

[20] R. Monasson and J. Tubiana. Emergence of compositional representations in restricted Boltzmann machines. *Phys. Rev. Let.*, 118:138301, 2017.

[21] L. Zdeborová and F. Krzakala. Statistical physics of inference: thresholds and algorithms. *Advances in Physics*, 65(5):453–552, 2016.

[22] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Comput.*, 11(2):443–482, 1999.

[23] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4):291–294, 1988.

[24] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv:1312.6120, 2014.

[25] A. Decelle, G. Fissore, and C. Furtlehner. Spectral dynamics of learning in restricted Boltzmann machines. *EPL*, 119(6):60001, 2017.

[26] E.W. Tramel, M. Gabrié, A. Manoel, F. Caltagirone, and F. Krzakala. A Deterministic and Generalized Framework for Unsupervised Learning with Restricted Boltzmann Machines. arXiv:1702.03260, 2017.

[27] V. A. Marčenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

[28] M. Mézard. Mean-field message-passing equations in the Hopfield model and its generalizations. *Phys. Rev. E*, 95:022117, 2017.

[29] G. Parisi and M. Potters. Mean-field equations for spin models with orthogonal interaction matrices. *Journal of Physics A: Mathematical and General*, 28(18):5267, 1995.

[30] M. Opper and O. Winther. Adaptive and self-averaging Thouless-Anderson-Palmer mean field theory for probabilistic modeling. *Physical Review E*, 64:056131, 2001.

[31] D. J. Amit, H. Gutfreund, and H. Sompolinsky. Spin-glass models of neural networks. *Phys. Rev. A*, 32:1007–1018, 1985.

[32] M. Mézard, G. Parisi, and M. A. Virasoro. *Spin Glass Theory and Beyond*. World Scientific, Singapore, 1987.

[33] J. R. L. Almeida and D. J. Thouless. Stability of the Sherrington-Kirkpatrick solution of a spin glass model. *J.Phy. A:Math. Gen*, 11(5):983–990, 1978.

[34] P. C. Hohenberg and M. C. Cross. *An introduction to pattern formation in nonequilibrium systems*, pages 55–92. Springer Berlin Heidelberg, Berlin, Heidelberg, 1987.

[35] I. Mastromatteo and M. Marsili. On the criticality of inferred models. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(10):P10012, 2011.

Reprint C

Robust Multi-Output Learning with Highly Incomplete Data via Restricted Boltzmann Machines

Robust Multi-Output Learning with Highly Incomplete Data via Restricted Boltzmann Machines

Giancarlo Fissore*, Aurélien Decelle*, Cyril Furtlehner†
*Université Paris-Saclay, Laboratoire de recherche en informatique, 91405, Orsay, France.
†Inria, Inria Saclay-Île-de-France, 91120, Palaiseau, France.
Yufei Han - Nortonlifelock Research Group, yfhan.hust@gmail.com

Abstract

In this work we tackle the challenging problem of multi-output classification with partially observed features and labels. We show that a simple Restricted Boltzmann Machine can be trained with an adapted algorithm based on mean-field equations to efficiently solve problems of inductive and transductive learning in which both features and labels are missing at random. The effectiveness of the approach is demonstrated empirically on various datasets, with particular focus on a real-world Internet-of-Things security dataset.

1 Introduction

Modern machine learning models usually require large sets of fully observed data to be trained, which are often not available in real-world applications. Often a random subset of features is absent (e.g. failed sensors of a monitoring system) and the data are insufficiently annotated (limitation due to human annotation) meaning that a lot of labels are missing. Consequently, it is highly important for machine learning systems to tolerate co-occurrence of missing features and partially observed labels for robust learning in practical use.

We study both **transductive** and **inductive** multi-output classification. Multi-output classification, including **multi-class** and **multi-label** learning tasks, outputs class labels with higher dimensional representation. They thus define a more sophisticated learning scenario compared to binary classification. The former associates an input instance to one class of a finitely defined class set, while the latter allows one instance to be associated to multiple labels simultaneously. In the transductive case, the learning objective is to infer the missing features and labels based on the observed ones (no separate test set is required). In the inductive case, the model is trained over a set of **incomplete training instances** and the classification is performed by inferring the labels associated to **incomplete test instances**. In our work, we consider that both features and labels are missing completely at random, which means that the mask of missing information is assumed to be statistically independent on the data distribution.

State-of-the-art accuracies for classification with incomplete or noise-corrupted features and partially observed labels are given by **CLE** [HSSZ18], **NoisyIMC** [CHS15] and **MC-1** [ClTCB15]. In particular, **CLE** and **NoisyIMC** are able to conduct both transductive and inductive learning for multi-label classification, while **MC-1** is a transductive-only method. **NoisyIMC** and **MC-1** can work in the semi-supervised learning scenario. Notably, in [LZG15] a method similar to ours is proposed to deal with incomplete labels. However, none of the

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)
In: Rudolph S., Marreiros G. (eds.): Proceedings of STAIRS, Santiago de Compostela, Spain, august 2020, published at http://ceur-ws.org

CEUR-WS.org/Vol-2655/paper13.pdf

four methods above can handle all the challenges co-currently raised in our work. First, all of these assume the testing instances to have fully observed feature profiles. They don't consider coping with incomplete testing instances by design. Second, all of them are designed specifically for multi-label learning and adapting them to multi-class classification is not straightforward.

More recently, methods based on Deep Latent Variable Models (DLVM) have been proposed to deal with missing data. In [MF19], the Variational Autoencoder [KW14] has been adapted to be trained with missing data and a sampling algorithm for data imputation is proposed. Other approaches based on Generative Adversarial Networks (GAN) by [GPAM$^+$14] are proposed in [YJvdS18] and [LJM19]. Impressive results on image datasets are displayed for these models, at the price of a rather high model complexity and the need for a large training set. In addition these works are focused on features reconstruction, and additional specifications and fine-tuning are required to be able to take partially observed labels into account. The models specifications are quite involved and any new specificity of the dataset may increase both the cost and the difficulty in training (especially for the approaches based on GANs).

In this paper we choose to address this problem in a more economical and robust manner. We consider the old and simple architecture of the Restricted Boltzmann Machine and adapt it to the multi-output learning context (RBM-MO) with missing data. The **RBM-MO** method serves as a generative model which collaboratively learns the marginal distribution of features and label assignments of input data instances, despite the incomplete observations. Building on the ideas expressed in [NK94, GJ94] we adapt the approach to the more effective contrastive divergence training procedure [Hin02] and provide results on various real-world datasets. The advantage of the RBM-MO model is that of providing a robust and flexible method to deal with missing data, with little additional complexity with respect to the classic RBM. Indeed, the trained model can be naturally applied to both transductive and inductive scenarios, achieving superior multi-output classification performance then state-of-the-art baselines. Moreover, it works seamlessly with multi-class and multi-label tasks, providing a unified framework for multi-output learning.

2 Overview of Restricted Boltzmann Machines

An RBM is a Markov random field with pairwise interactions defined on a bipartite graph formed by two layers of non-interacting variables: the visible nodes represent instances of the input data while the hidden nodes provide a latent representation of the data instances. $\mathcal{V}$ and $\mathcal{H}$ will denote respectively the sets of visible and hidden variables. In our setting, the visible variables will further split into two subsets $\mathcal{V}_f$ and $\mathcal{V}_\ell$ corresponding respectively to features and labels, such that $\mathcal{V} = \mathcal{V}_f + \mathcal{V}_\ell$. The visible variables form an explicit representation of the data and are noted $\boldsymbol{v} = \{v_i, i \in \mathcal{V}\}$. The hidden nodes $\boldsymbol{h} = \{h_j, j \in \mathcal{H}\}$ serve to approximate the underlying dependencies among the visible units.

In this paper, we will work with binary hidden nodes $h_j \in \{0, 1\}$. The variables corresponding to the visible features will be either real with a Gaussian prior or binary, depending on the data to model, and labels variables will always be binary ($v_i \in \{0, 1\}$). The joint probability distribution over the nodes is defined through an energy function

$$P(\boldsymbol{v}, \boldsymbol{h}) = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{Z} p_{\text{prior}}(\boldsymbol{v}), \quad E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i\in\mathcal{V}, j\in\mathcal{H}} v_i w_{ij} h_j - \sum_{i\in\mathcal{V}} a_i v_i - \sum_{j\in\mathcal{H}} b_j h_j \tag{1}$$

where a_i and b_j are biases acting respectively on the visible and hidden units and w_{ij} is the weight matrix that couples visible and hidden nodes. p_{prior} is in product form and encodes the nature of each visible variable, either with a Gaussian prior $p_{\text{prior}} = \mathcal{N}(0, \sigma_v^2)$ or a binary prior $p_{\text{prior}}(v) = \delta(s^2 - s)$. $Z = \sum_{\boldsymbol{v},\boldsymbol{h}} p_{\text{prior}}(\boldsymbol{v}) e^{-E(\boldsymbol{v},\boldsymbol{h})}$ is the partition function. The classical training method consists in maximizing the marginal likelihood over the visible nodes $P(\boldsymbol{v}) = \sum_{\boldsymbol{h}} P(\boldsymbol{v}, \boldsymbol{h})$ by tuning the RBM parameters $\theta = \{w_{ij}, a_i, b_j\}$ via gradient ascent of the log likelihood $\mathcal{L}(\boldsymbol{v}; \theta)$.

The tractability of the method relies heavily on the fact that the conditional probabilities $P(\boldsymbol{v}|\boldsymbol{h})$ and $P(\boldsymbol{h}|\boldsymbol{v})$ are given in closed forms. In our case these read:

$$P(\boldsymbol{v}|\boldsymbol{h}) = \prod_{i\in\mathcal{V}_f} \frac{e^{\sum_{j\in\mathcal{H}} v_i w_{ij} h_j + a_i v_i} p_{\text{prior}}(v_i)}{\sum_{v_i} e^{\sum_{j\in\mathcal{H}} v_i w_{ij} h_j + a_i v_i} p_{\text{prior}}(v_i)} \prod_{i\in\mathcal{V}_\ell} \sigma\Big(\sum_{j\in\mathcal{H}} v_i w_{ij} h_j + a_i v_i\Big), \tag{2}$$

$$P(\boldsymbol{h}|\boldsymbol{v}) = \prod_{j\in\mathcal{H}} \sigma\Big(\sum_{i\in\mathcal{V}} v_i w_{ij} + b_j\Big), \tag{3}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the logistic function. The gradient of the likelihood w.r.t. the weights (and similarly w.r.t. the fields a_i and b_j) is given by

$$\frac{\partial \mathcal{L}(\boldsymbol{v};\theta)}{\partial w_{ij}} = \langle v_i h_j p(h_j|\boldsymbol{v})\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{RBM}} \tag{4}$$

where the brackets $\langle\rangle_{\text{data}}$ and $\langle\rangle_{\text{RBM}}$ respectively indicate the average over the data and over the distribution (1). The positive term is directly linked to the data and can be estimated exactly with (3), while the negative term is intractable. Many strategies are used to compute this last term: the *contrastive divergence* (CD) approach [Hin02] consists in estimating the term over a finite number of Gibbs sampling steps, starting from a data point and making alternate use of (2) and (3); in its *persistent* version (PCD) the chain is maintained over subsequent mini-batches; using mean-field approximation [MTK15] the term is computed by means of a low-couplings expansion.

3 Learning RBM with incomplete data

The RBM is a generative model able to learn the joint distribution of some empirical data given as input. As such, it is intrinsically able to encode the relevant statistical properties found in the training data instances that relate features and labels, and this makes the RBM particularly suitable to be used in the multi-output setting in the presence of incomplete observations. In this sense, the most natural way to deal with incomplete observations is to marginalize over the missing variables; in this section we show how the contrastive divergence algorithm can be adapted to compute such marginals.

Given a partially-observed instance $\boldsymbol{v}$, we have a new partition of the visible space $\mathcal{V} = \mathcal{O} + \mathcal{M}$, where $\mathcal{O}$ is a subset of observed values of $\boldsymbol{v}$ that can correspond both to features and labels. $\boldsymbol{v}_o = \{v_i, i \in \mathcal{O}\}$ and $\boldsymbol{v}_m = \{v_i, i \in \mathcal{M}\}$ denote respectively the observed and missing values of $\boldsymbol{v}$. The probability over the observed variables $\boldsymbol{v}_o$ is given by (θ representing the parameters of the model)

$$P(\boldsymbol{v}_o) = \frac{Z_{\mathcal{O}}[\theta]}{Z_{\emptyset}[\theta]}, \qquad Z_{\mathcal{O}}[\theta] = \int \prod_{i\in\mathcal{M}} p_{\text{prior}}(v_i) dv_i \times e^{\sum_{k\in\mathcal{V}} a_k v_k} \prod_{j\in\mathcal{H}} \left(1 + \exp\Big(\sum_{k\in\mathcal{V}} w_{kj} v_k + b_j\Big)\right)$$

Taking the log-likelihood and then computing the gradient with respect to the weight matrix element w_{ij} (also similarly for the fields a_i and b_j), we obtain two different expressions for $i \in \mathcal{O}$ and $i \in \mathcal{M}$.

$$\frac{\partial \log Z_{\mathcal{O}}[\theta]}{\partial w_{ij}} = v_i \sum_{h_j} h_j p(h_j|\boldsymbol{v}_o) \quad i \in \mathcal{O}, \qquad \frac{\partial \log Z_{\mathcal{O}}[\theta]}{\partial w_{ij}} = \sum_{h_j} \int dv_i v_i h_j p(v_i, h_j|\boldsymbol{v}_o) \quad i \in \mathcal{M} \tag{5}$$

The gradient of the LL over the weights (4) now reads

$$\frac{\partial \mathcal{L}(\boldsymbol{v};\theta)}{\partial w_{ij}} = \Big\langle I_o(i) v_i \sum_{h_j} h_j p(h_j|\boldsymbol{v}_o)\Big\rangle_{\text{data}} + \Big\langle \big(1 - I_o(i)\big) \sum_{h_j} \int dv_i v_i h_j p(h_j|\boldsymbol{v}_o)\Big\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{RBM}} \tag{6}$$

where I_o is the indicator function of the samples dependent set $\mathcal{O}$. The observed variables $v_i, i \in \mathcal{O}$ are pinned to the values given by the training samples. In terms of our model, the pinned variables play the role of an additional bias over the hidden variables of a RBM where the ensemble of visible variables is reduced to the missing ones.

With respect to the non-lossy case where $p(h_j|\boldsymbol{v})$ is given in closed form, here we need to sum over the missing variables in order to estimate $p(h_j|\boldsymbol{v}_o)$. This means that also the positive term of the gradient (6) is now intractable and we need to approximate it. For CD training, we can simply perform Gibbs sampling over the missing variables (keeping fixed the observed variables). Details are reported in Alg. 1.

We note that the extra computational burden of Lossy-CD with respect to standard CD is due only to the extra Gibbs sampling steps in the positive term. Given that the observed variables strongly bias the sampling procedure speeding up convergence, only few sampling steps are needed to compute this term. Indeed, in our experiments we observed that a single sampling step (Lossy-CD1) is enough, making the additional complexity minimal. Finally, we note that the same method can be applied to PCD and mean-field training procedures. In the first case, it is sufficient to keep track of an additional persistent chain, which requires little extra memory

and no extra computational complexity. In the second case, we only need to substitute Gibbs sampling with iterative mean-field equations.

Algorithm 1: Lossy-CDk (RBM training with Incomplete data)

Data: a training set of N data vectors
Randomly initialize the weight matrix $\mathbf{W}$
for t = 0 to T (# of epochs) **do**
  Divide the training set in m minibatches
  for all minibatches m **do**
   Positive term:
    pin variables $v_i, i \in \mathcal{O}$ to their correct value
    initialize $v_i, i \in \mathcal{M}$ randomly
    sample $\boldsymbol{h}, \boldsymbol{v}_m$ using $p(\boldsymbol{v}_m \mid \boldsymbol{h})$ and $p(\boldsymbol{h} \mid \boldsymbol{v})$ for k steps
    compute the positive terms in (5)
   Negative term:
    initialize $\boldsymbol{v}$ randomly
    iterate eq. (2), (3) (k steps) to compute $\langle v_i h_j \rangle_{model}$
   Full update:
    update $\mathbf{W}$ with equation (6)
  end for
end for

4 Mean-field based imputation with RBM

As a generative model, the trained RBM can be used to sample new data. For imputation of missing features and labels we just need to use the observed portions of our data to bias the sampling procedure in the same way as for the computation of the positive term in Alg. 1. Namely, we estimate $p(\boldsymbol{v_m}|\boldsymbol{v_o})$ by pinning the observed variables and iterating CD/PCD or mean-field to approximate the equilibrium values of the missing variables. In case of a high percentage of missing observations, however, we might expect the observed variables to be correlated to many different equilibrium configurations, such that the sampling could be biased towards the wrong sample. To overcome this problem, we simply average over multiple mean-field imputations for each incomplete data instance.

More in details, let $\{p_i, i \in \mathcal{V}_\ell\}$ and $\{q_j, j \in \mathcal{H}\}$ be the marginal probabilities respectively of visible labels and hidden variables to be activated and $\{m_i, i \in \mathcal{V}_f\}$ the marginal expectation of the visible features variables. Mean-field equations at lowest order ($\mathcal{O}(1/N)$, N being the size of the system) express self-consistent relations among these quantities

$$m_i = \Big(\sum_{j \in \mathcal{H}} w_{ij} q_j + a_i \Big) \sigma_v^2 \quad \forall i \in \mathcal{V}_f \backslash \mathcal{O} \qquad p_i = \sigma\Big(\sum_{j \in \mathcal{H}} w_{ij} q_j + a_i \Big) \quad \forall i \in \mathcal{V}_\ell \backslash \mathcal{O} \tag{7}$$

$$q_j = \sigma\Big(\sum_{i \in \mathcal{V}_f} w_{ij} m_i + \sum_{i \in \mathcal{V}_\ell} w_{ij} p_i + b_j \Big) \tag{8}$$

Higher order terms corresponding to TAP equations are discarded [M17]. These equations can be efficiently solved by iteration starting from random configurations until a fixed point is reached. Observed variables are simply introduced by pinning their corresponding probabilities (0 or 1 for label variables) or their marginal expectation (for feature variables) to the observed values. In practice we run these fixed-point equations $N_f \sim 10$ times and the imputations are obtained by simple average

$$\hat{m}_i = \frac{1}{N_f} \sum_{n=1}^{N_f} m_i^{(n)} \qquad p_i = \frac{1}{N_f} \sum_{n=1}^{N_f} p_i^{(n)}.$$

In the multi-label setting, the predictor is the indicator function $\hat{p}_i = (p_i > t)$ (t is learned, it is chosen to maximize the accuracy for known labels), while for class labels we have $\hat{p}_i = 1$ if $i = \text{argmax}_k(p_k)$

Model	RMSE			Averaged AUC			Accuracy		
q_{mc}%	30%	50%	80%	30%	50%	80%	30%	50%	80%
RBM-MO(q_{fea}% =50%)	**0.183**	**0.182**	**0.185**	**0.969**	**0.971**	**0.929**	**0.950**	**0.912**	**0.822**
CLE(q_{fea}% =50%)	0.195	0.195	0.195	0.686	0.718	0.742	0.256	0.232	0.282
NoisyIMC(q_{fea}% =50%)	0.209	0.210	0.210	0.621	0.578	0.552	0.225	0.232	0.192
MC-1(q_{fea}% =50%)	0.334	0.335	0.337	0.495	0.493	0.500	0.110	0.111	0.112
RBM-MO(q_{fea}% =80%)	**0.209**	**0.213**	**0.211**	**0.938**	**0.932**	**0.906**	**0.920**	**0.852**	**0.733**
CLE(q_{fea}% =80%)	0.206	0.208	0.206	0.673	0.678	0.625	0.230	0.215	0.220
NoisyIMC(q_{fea}% =80%)	0.212	0.211	0.213	0.652	0.577	0.537	0.230	0.217	0.210
MC-1(q_{fea}% =80%)	0.334	0.334	0.335	0.500	0.501	0.500	0.112	0.110	0.110

Table 1: Transductive test on *MNIST* multi-class data set (our method in bold, best result in red)

Model	RMSE			Micro-AUC			Hamming-Accuracy		
q_{ml}%	30%	50%	80%	30%	50%	80%	30%	50%	80%
RBM-MO(q_{fea}% =50%)	**0.131**	**0.137**	**0.123**	**0.943**	**0.934**	**0.888**	**0.919**	**0.907**	**0.873**
CLE(q_{fea}% =50%)	0.130	0.130	0.131	0.905	0.893	0.898	0.885	0.871	0.878
NoisyIMC(q_{fea}% =50%)	0.132	0.133	0.133	0.865	0.863	0.858	0.845	0.841	0.848
MC-1(q_{fea}% =50%)	0.258	0.255	0.267	0.522	0.528	0.527	0.826	0.817	0.824
RBM-MO(q_{fea}% =80%)	**0.160**	**0.158**	**0.158**	**0.875**	**0.867**	**0.826**	**0.856**	**0.858**	**0.832**
CLE(q_{fea}% =80%)	0.129	0.129	0.128	0.913	0.897	0.899	0.889	0.875	0.876
NoisyIMC(q_{fea}% =80%)	0.133	0.134	0.134	0.853	0.857	0.849	0.839	0.835	0.826

Table 2: Transductive test on *Scene* multi-label data set (our method in bold, best result in red)

5 Experimental Study

5.1 Experimental configuration

To evaluate the efficiency of RBM-MO we compare its performance against **CLE**, **NoisyIMC** and **MC-1**, which provide state-of-the-art baselines.

For the transductive experiments we randomly hide features and labels of the whole dataset to generate incomplete data for training, and we compute appropriate scores for the reconstruction of missing features and labels. In the inductive test, instead, we split the whole dataset into non-overlapping training and testing sets. Concerning the training set the same protocol is used as in the transductive test. For the test set the difference is that now all labels are hidden. Once the classifier is trained, it is applied on the test set to predict the labels. We still randomly hide the entries of test features vectors, so as to form an **incomplete testing set**. Finally, in the splitting we use 70% of the data instances for training and the remaining 30% for testing.

We denote by q_{fea}, q_{ml} and q_{mc} the percentage of masked features, labels and classes labels respectively. Note that a masked class label means that all binary variables attached to the classes of a given label are masked together. These rates of masking are kept identical in the learning and test sets.

In the transductive test, we compute the **Root Mean Squared Error (RMSE)** to measure the reconstruction accuracy with respect to the missing feature values. Furthermore, for the reconstructed labels we calculate **Micro-AUC** scores and **Hamming-accuracy** [GKG12] in the multi-label scenario, and **Averaged AUC** plus **Accuracy** [LY15] in the multi-class case. In the tables, we define **Hamming-accuracy** as **1-Hamming loss** to keep a consistent variation tendency with the AUC scores. In the inductive test we only compute the scores on the reconstructed labels, since reconstructing missing features is not the goal of inductive classification.

We run the test as described 10 times with different realizations of the missing features and labels. Average and

Model	Averaged AUC			Accuracy		
q_{mc}%	30%	50%	80%	30%	50%	80%
RBM-MO(q_{fea}% =50%)	**0.887**	**0.914**	**0.910**	**0.533**	**0.673**	**0.660**
CLE(q_{fea}% =50%)	0.785	0.791	0.791	0.297	0.256	0.268
NoisyIMC(q_{fea}% =50%)	0.780	0.771	0.781	0.302	0.272	0.265
RBM-MO(q_{fea}% =80%)	**0.891**	**0.909**	**0.889**	**0.562**	**0.682**	**0.647**
CLE(q_{fea}% =80%)	0.768	0.664	0.622	0.271	0.200	0.176
NoisyIMC(q_{fea}% =80%)	0.748	0.687	0.615	0.264	0.220	0.178

Table 3: Inductive test on *Pendigits* multi-class dataset (our method in bold, best result in red)

Model	Micro-AUC			Hamming-Accuracy		
q_{ml}%	30%	50%	80%	30%	50%	80%
RBM-MO(q_{fea}% =50%)	**0.839**	**0.826**	**0.793**	**0.970**	**0.970**	**0.965**
CLE(q_{fea}% =50%)	0.705	0.707	0.706	0.700	0.724	0.719
NoisyIMC(q_{fea}% =50%)	0.704	0.702	0.700	0.710	0.717	0.718
RBM-MO(q_{fea}% =80%)	**0.759**	**0.791**	**0.766**	**0.964**	**0.964**	**0.967**
CLE(q_{fea}% =80%)	0.693	0.688	0.694	0.718	0.706	0.718
NoisyIMC(q_{fea}% =80%)	0.689	0.688	0.685	0.705	0.704	0.704

Table 4: Inductive test on *EventCat* multi-label data set (our method in bold, best result in red)

Dataset	**No. of Instances**	**No. of Features**	**No.of Labels**	**No. of Classes**
Scene	2,407	294	6	-
Pendigits	10992	16	-	10
MNIST	70,000	784	-	10
EventCat	5,93	72	6	-

Table 5: Summary of 4 public multi-label and multi-class data sets.

standard deviation of the computed scores are recorded to compare the overall performances. In the tables, we use red fonts to denote the best reconstruction and classification performances among all the algorithms involved in the empirical study. The bold black font is used to highlight the performance of the proposed **RBM-MO** method.

For the baselines, we used grid search to choose the optimal parameter combination following the suggested ranges of parameters as in [HSSZ18].

The RBM-MO is trained following the guidelines in [Hin10]. We always use binary variables for the hidden layer, while in the visible layer we use binary variables for MNIST and Gaussian variables for the other datasets. In all the simulations, we fix the number of hidden nodes to 100. The learning rate η is fixed to 0.001 and the size of the mini-batches to 10. During training the number of Gibbs steps is set to $k = 1$ while for imputation we iterate the mean-field equations 10 times. As a stopping condition, we considered the degradation of the transductive AUC scores with a look-ahead of 500 epochs

5.2 Summary of datasets

We consider 3 publicly available datasets related to image processing. These datasets cover both multi-label and multi-class learning tasks, and they are popularly used as benchmark datasets in multi-output learning research.

In addition, we consider the challenging scenario of abnormality detection on IoT devices. The relevant dataset, that we call *EventCat*, consists in security telemetry data collected from various network appliances (e.g. smart watches, smartphones, driving assistance systems...), each reporting a features vector whose entries indicate the occurring frequency of a specific type of alert (e.g. downloading suspicious files, login failures, unfixed vulnerabilities...). Multiple labels are assigned to each device in the collected dataset, corresponding to a variety of categories of security threats.

Some details about the datasets are reported in Table.5.

5.3 Qualitative results on MNIST

A qualitative evaluation of the performance of the RBM-MO model is given by looking at features reconstruction for the MNIST dataset, as reported in Fig. 1. The model at hand has been trained over a dataset in which 50% of the features were missing. To assess the robustness of the method, we computed the reconstructions in the highly challenging case in which 80% of the features were missing. Apart from some smoothing due to the employment of mean-field imputations, the reconstructed samples look reasonably realistic. In general, from the qualitative point of you the results are comparable to those obtained with more complex and expensive DLVMs like MIWAE and MisGAN [MF19, LJM19].

5.4 Empirical results

The transductive results for MNIST (multi-class) and Scene (multi-label) datasets are reported in tables 1 and 2. Going into the details, we first observe that **RBM-MO** is by a large margin more efficient than all of the

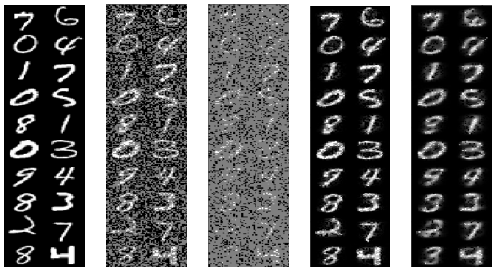


Figure 1: Features reconstruction by RBM-MO trained over an incomplete dataset with 50% missing-at-random features, whose classification accuracy has been measured to be around 91%. The first block shows some complete testing instances. The second and third block show the same testing instances after hiding respectively 50% and 80% of the pixels. The last two columns show the results of the mean-field imputations over the incomplete testing instances.

baselines for the inference of class labels (table 1), probably because it is able to encode more complex statistical properties.

On the multi-label problems, the situation is still in favour of **RBM-MO** but with less margin (table 2), in particular at a larger percentage of missing features.

Now if we look at the reconstruction error on these datasets we observe that **RBM-MO** generally achieves a higher reconstruction accuracy than the other opponents, especially on the MNIST dataset. The results verify empirically the basic motivation of using a generative model such as the RBM: **incomplete features and labels can provide complementary information to each other, so as to better recover the missing elements**. The variance of the results is omitted in the tables by lack of space. For **RBM-MO** the standard deviation of the derived RMSE, AUC and accuracy scores is not larger than 0.01 over the different datasets. Although the RMSE scores reported by the baseline methods look comparable to the RBM-MO ones, and in certain cases they are better, they also come with a slightly higher variance, such that the RBM-MO seems to be more efficient and robust for features reconstruction.

Except **MC-1**, all the baseline methods are used for inductive learning. As in the transductive test, we show only the mean of the derived metrics in the tables. Nevertheless, we have similar variance ranges for the computed scores as reported in the transductive test. Clearly **RBM-MO** is much better adapted to this setting than the baseline methods both for multi-class (table 3) and multi-label learning. The baseline inductive methods **CLE** and **NoisyIMC** are specifically designed for multi-label learning and their performance deteriorates significatively in the multi-class scenario. By comparison, **RBM-MO** can be adapted seamlessly to multi-class and multi-label learning, producing consistently good performances.

For the *EventCat* dataset, inductive results are reported in table 4. Even with highly incomplete training data, **RBM-MO** produces the best predictions over partially observed testing data instances.

6 Conclusion

Machine learning is witnessing a race to high complexity models eager for large data and computational power. In the context of multi-output classification in a challenging scenario - (i) learning with highly incomplete features and partially observed labels; ii) applying the learnt classifier with incomplete testing instances) - we advocate instead for simple probabilistic and interpretable models. After refining the learning of the RBM model, we give empirical evidences that it can be efficiently adapted to this context on a great variety of datasets. Experiments are conducted on both public databases and a real-world IoT security dataset, showing various sizes of training sets as well as features and labels vectors. Our approach consistently outperforms the state-of-the-art robust multi-class and multi-label learning approaches with imperfect training data, indicating good usability for practical applications.

References

[CHS15] Kai-Yang Chiang, Cho-Jui Hsieh, and Inderjit S.Dhillon. Matrix completion with noisy side information. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 3447–3455, Cambridge, MA, USA, 2015. MIT Press.

[ClTCB15] R. Cabral, F. D. l. Torre, J. P. Costeira, and A. Bernardino. Matrix completion for weakly-supervised multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):121–135, Jan 2015.

[GJ94] Zoubin Ghahramani and Michael I. Jordan. Supervised learning from incomplete data via an em approach. In *Advances in Neural Information Processing Systems 6*, pages 120–127. Morgan Kaufmann, 1994.

[GKG12] G.Madjarov, D. Kocev, and D. Gjorgjevikj. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, pages 3083–3104, 2012.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[Hin02] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14:1771–1800, 2002.

[Hin10] Geoffrey Hinton. A practical guide to training restricted Boltzmann machines. *Momentum*, 2010.

[HSSZ18] Yufei Han, Guolei Sun, Yun Shen, and Xiangliang Zhang. Multi-label learning with highly incomplete data via collaborative embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, pages 1494–1503, 2018.

[KW14] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, pages 4413–4423, 2014.

[LJM19] Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Learning from incomplete data with generative adversarial networks. In *International Conference on Learning Representations*, 2019.

[LY15] Yue Wu Li, Lin and Mao Ye. Experimental comparisons of multi-class classifiers. *Informatica*, 2015.

[LZG15] Xin Li, Feipeng Zhao, and Yuhong Guo. Conditional Restricted Boltzmann Machines for Multi-label Learning with Incomplete Labels. In Guy Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 635–643, San Diego, California, USA, 09–12 May 2015. PMLR.

[MÍ7] M. Mézard. Mean-field message-passing equations in the Hopfield model and its generalizations. *Phys. Rev. E*, 95:022117, 2017.

[MF19] Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4413–4423, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[MTK15] G. Marylou, E.W. Tramel, and F. Krzakala. Training restricted Boltzmann machines via the Thouless-Anderson-Palmer free energy. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 640–648, 2015.

[NK94] M. J. Nijman and Kappen. Using boltzmann machines to fill in missing values, 1994.

[YJvdS18] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5689–5698, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Reprint D

Relative gradient optimization of the Jacobian term in unsupervised deep learning

Relative gradient optimization of the Jacobian term in unsupervised deep learning

Luigi Gresele[*1,2] **Giancarlo Fissore**[*3,4] **Adrián Javaloy** [1]

Bernhard Schölkopf [1] **Aapo Hyvärinen** [3,5]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[3] Université Paris-Saclay, Inria, Inria Saclay-Île-de-France, 91120, Palaiseau, France
[4] Université Paris-Saclay, CNRS, Laboratoire de recherche en informatique, 91405, Orsay, France
[5] Dept of Computer Science, University of Helsinki, Finland
luigi.gresele@tuebingen.mpg.de; giancarlo.fissore@inria.fr

Abstract

Learning expressive probabilistic models correctly describing the data is a ubiquitous problem in machine learning. A popular approach for solving it is mapping the observations into a representation space with a simple joint distribution, which can typically be written as a product of its marginals — thus drawing a connection with the field of nonlinear independent component analysis. Deep density models have been widely used for this task, but their maximum likelihood based training requires estimating the log-determinant of the Jacobian and is computationally expensive, thus imposing a trade-off between computation and expressive power. In this work, we propose a new approach for exact training of such neural networks. Based on relative gradients, we exploit the matrix structure of neural network parameters to compute updates efficiently even in high-dimensional spaces; the computational cost of the training is quadratic in the input size, in contrast with the cubic scaling of naive approaches. This allows fast training with objective functions involving the log-determinant of the Jacobian, without imposing constraints on its structure, in stark contrast to autoregressive normalizing flows.

1 Introduction

Many problems of machine learning and statistics involve learning invertible transformations of complex, multimodal probability distributions into simple ones. One example is density estimation through latent variable models under a specified base distribution [51], which can also have applications in data generation [14, 33, 19] and variational inference [44]. Another example is nonlinear independent component analysis (nonlinear ICA), where we want to extract simple, disentangled features out of the observed data [27, 30].

One approach to learn such transformations, introduced in [50] in the context of density estimation, is to represent them as a composition of simple maps, the sequential application of which enables high expressivity and a large class of representable transformations. Deep neural networks parameterize functions of multivariate variables as modular sequences of linear transformations and component-wise activation functions, thus providing a natural framework for implementing that idea, as already proposed in [45].

[*]Equal contribution

Unfortunately, however, typical strategies employed in neural networks training do not scale well for objective functions like the aforementioned ones; in fact, through the change of variable formula, the logarithm of the absolute value of the determinant of the Jacobian appears in the objective. Its exact computation, let alone its optimization, quickly gets prohibitively computationally demanding as the data dimensionality grows.

A large part of the research on deep density estimation, generally referred to under the term *autoregressive normalizing flows*, has therefore been dedicated to considering a restricted class of transformations such that the computation of the Jacobian term is trivial [14, 44, 15, 34, 25, 12], thus imposing a tradeoff between computation and expressive power. While such models can approximate arbitrary probability distributions, the extracted features are strongly restricted based on the imposed triangular structure, which prevents the system from learning a properly disentangled representation. Other strategies involve the optimization of an approximation of the exact objective [5], and continuous-time analogs of normalizing flows for which the likelihood (or some approximation thereof) can be computed using relatively cheap operations [13, 19].

In this work, we provide an efficient way to optimize the exact maximum likelihood objective for deep density estimation as well as for learning disentangled representations by latent variable models. We consider a nonlinear, invertible transformation from the observed to the latent space which is parameterized through fully connected neural networks. The weight matrices are merely constrained to be invertible. The starting point is that the parameters of the linear transformations are matrices; this allows us to exploit properties of the Riemannian geometry of matrix spaces to derive parameter updates in terms of the relative gradient, which was originally introduced as the natural gradient in the context of linear ICA [11, 2], and which can be feasibly computed. We show how this can be integrated with the usual backpropagation employed to compute gradients in neural network training, yielding an overall efficient way to optimize the Jacobian term in neural networks. This is a general optimization approach which is potentially useful for any objective involving such a Jacobian term, and is likely to find many applications in diverse areas of probabilistic modelling, for example in the context of Bayesian active learning for the computation of the information gain score [48], or for fitting the reverse Kullback-Leibler divergence in variational inference [54, 7].

The computational cost of our proposed optimization procedure is quadratic in the input size— essentially the same as ordinary backpropagation— which is in stark contrast with the cubic scaling of the naive way of optimizing via automatic differentiation. The joint asymptotic scaling of forward and backward pass as a function of the input size is therefore the same that aforementioned alternative methods achieve by imposing strong restrictions on the neural network structure [44] and thus on the class of functions they can represent. In contrast, our approach allows to efficiently optimize the exact objective for neural networks with arbitrary Jacobians.

In sections 2 and 3 we review maximum likelihood estimation for latent variable models, backpropagation and the Jacobian term for neural networks, and discuss the complexity of the naive approaches for optimizing the Jacobian term. Then in section 4 we discuss the relative gradient, and show how it can be integrated with backpropagation resulting in an efficient procedure. We verify empirically the computational speedup our method provides in section 5.

2 Background

2.1 Maximum likelihood for latent variable models

Consider a generative model of the form

$$\mathbf{x} = \mathbf{f}(\mathbf{s}) \tag{1}$$

where $\mathbf{s} \in \mathbb{R}^D$ is the latent variable, $\mathbf{x} \in \mathbb{R}^D$ represents the observed variable and $\mathbf{f} : \mathbb{R}^D \to \mathbb{R}^D$ is a deterministic and invertible function, which we refer to as *forward* transformation. Under the model specified above, the log-likelihood of a single datapoint $\mathbf{x}$ can be written as

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log p_s(\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x})) + \log \left| \det \mathbf{J}\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}) \right| , \tag{2}$$

where $\mathbf{g}_{\boldsymbol{\theta}}$ is some representation with parameters $\boldsymbol{\theta}$ of the *inverse* transformation[2] of $\mathbf{f}$; $\mathbf{J}\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}) \in \mathbb{R}^{D \times D}$ its Jacobian computed at the point $\mathbf{x}$, whose elements are the partial derivatives

[2]The forward transformation could also be parameterized, but here we only explicitly parameterize its inverse.

$[\mathbf{J}\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x})]_{ij} = \partial g^i_{\boldsymbol{\theta}}(\mathbf{x})/\partial x^j$; and $p_{\boldsymbol{\theta}}$ and p_s denote, respectively, the probability density functions of $\mathbf{x}$ and of the latent variable $\mathbf{s}$ under the specified model. In many cases, it is additionally assumed that the distribution of the latent variable is sufficiently simple; for example, that it factorizes in its components,

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_i \log p_i(\mathbf{g}^i_{\boldsymbol{\theta}}(\mathbf{x})) + \log|\det \mathbf{J}\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x})|\,. \tag{3}$$

In this case, the problem can be interpreted as nonlinear independent component analysis (nonlinear ICA), and the components of $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x})$ are estimates of the original sources $\mathbf{s}$.

Another variant of this framework can be developed to solve the problem that nonlinear ICA is, in general, not identifiable without additional assumptions [29]; that means, even if the data is generated according to the assumed model, there is no guarantee that the recovered sources bear any simple relationship to the true ones. In order to obtain identifiability, it is possible to consider models [27, 28, 30, 20] in which the latent variables are not *unconditionally* independent, but rather *conditionally* independent given an additional, observed variable $\mathbf{u} \in \mathbb{R}^d$,

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{u}) = \sum_i \log p_i(\mathbf{g}^i_{\boldsymbol{\theta}}(\mathbf{x})|\mathbf{u}) + \log|\det \mathbf{J}\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x})|\,, \tag{4}$$

where d can be equal to or different from D depending on the model.

Maximum likelihood estimation for the model parameters amounts to finding, through optimization, the parameters $\boldsymbol{\theta}^*$ such that the expectation of the likelihood given by the expression in equation (3) is maximized. For all practical purposes, the expectation will be substituted with the sample average. Specifically, for optimization purposes, we will be interested in the computation of a gradient of such term on mini-batches of one or few datapoints, such that stochastic gradient descent can be employed.

2.2 Neural networks and backpropagation

Neural networks provide a flexible parametric function class for representing $\mathbf{g}_{\boldsymbol{\theta}}$ through a sequential composition of transformations, $\mathbf{g}_{\boldsymbol{\theta}} = \mathbf{g}_L \circ \ldots \circ \mathbf{g}_2 \circ \mathbf{g}_1$, where L defines the number of layers of the network. When an input pattern $\mathbf{x}$ is presented to the network, it produces a final output $\mathbf{z}_L$ and a series of intermediate outputs. By defining $\mathbf{z}_0 = \mathbf{x}$ and $\mathbf{z}_L = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x})$, we can write the forward evaluation as

$$\mathbf{z}_k = \mathbf{g}_k(\mathbf{z}_{k-1}) \;\text{ for } k = 1, \ldots, L\,. \tag{5}$$

Each module $\mathbf{g}_k$ of the network involves two transformations,

(a) a coupling layer $C_{\mathbf{W}_k}$, that couples the inputs to the layer with the parameters $\mathbf{W}_k$ to optimize;

(b) other arbitrary manipulations $\boldsymbol{\sigma}$ of inputs/outputs. Typically, these are element-wise nonlinear activation functions with fixed parameters; we can for simplicity think of them as operations of the form $\boldsymbol{\sigma}(\mathbf{x}) = (\sigma(x_1), \ldots, \sigma(x_n))$ applied to vector variables.

The resulting transformation can thus be written as $\mathbf{g}_k(\mathbf{z}_{k-1}) = \boldsymbol{\sigma}(C_{\mathbf{W}_k}(\mathbf{z}_{k-1}))$.

We will focus on fully connected modules, where the coupling $C_{\mathbf{W}}$ is simply a matrix-vector multiplication between the weights $\mathbf{W}_k$ and the input to the k-th layer; overall, the transformation operated by such a module can be expressed as $\boldsymbol{\sigma}(\mathbf{W}_k\mathbf{z}_{k-1})$. Another kind of coupling layer is given by convolutional layers, typically used in convolutional neural networks [36].

The parameters of the network are randomly initialized and then learned by gradient based optimization with an objective function $\mathcal{L}$, which is a scalar function of the final output of the network. At each learning step, updates for the weights are proportional to the partial derivative of the loss with respect to each weight.

The computation of these derivatives is typically performed by backpropagation [47], a specialized instance of automatic differentiation. Backpropagation involves a two-phase process. Firstly, during a *forward pass*, the intermediate and final outputs of the network $\mathbf{z}_1, \ldots, \mathbf{z}_L$ are evaluated and a value for the loss is returned. Then, in a second phase termed *backward pass*, derivatives of the loss with respect to each individual parameter of the network are computed by application of the chain rule. The gradients are computed one layer at a time, from the last layer to the first one; in the process,

the intermediate outputs of the forward pass are reused, employing dynamic programming to avoid redundant calculations of intermediate, repeated terms.[3]

In matrix notation, the updates for the weights of the k-th fully connected layer $\mathbf{W}_k$ can then be written as

$$\Delta \mathbf{W}_k \propto \mathbf{z}_{k-1} \boldsymbol{\delta}_k^\top \,, \tag{6}$$

where $\boldsymbol{\delta}_k$ is the cumulative result of the backward computation in the backpropagation step up to the k-th layer, also called backpropagated error. We report the full derivation in appendix A. We adopt the convention of defining $\mathbf{x}$, $\mathbf{z}_k$ and $\boldsymbol{\delta}_k$ as column vectors.

2.3 Difficulty of optimizing the Jacobian term of neural networks

In the case of the objective function specified in Eq. (3), we have $\mathcal{L}(\mathbf{x}) = \log p_{\boldsymbol{\theta}}(\mathbf{x})$. By defining

$$\mathcal{L}_p(\mathbf{x}) = \sum_i \log p_i(\mathbf{g}^i_{\boldsymbol{\theta}}(\mathbf{x})); \quad \mathcal{L}_J(\mathbf{x}) = \log \left|\det \mathbf{J}\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x})\right| \,, \tag{7}$$

the objective can be rewritten as $\mathcal{L}(\mathbf{x}) = \mathcal{L}_p(\mathbf{x}) + \mathcal{L}_J(\mathbf{x})$. The evaluation of the gradient of the first term $\mathcal{L}_p$ can be performed easily if a simple form for the latent density is chosen, as it only requires simple operations on top of a single forward pass of the neural network. Given that the loss is a scalar, as backpropagation is an instance of reverse mode differentiation [4], backpropagating the error relative to it in order to evaluate the gradients does not increase the overall complexity with respect to the forward pass alone.

In contrast, the evaluation of the gradient of the second term, $\mathcal{L}_J$, is very problematic, and our main concern in this paper. The key computational bottleneck is in fact given by the evaluation of the Jacobian during the forward pass. Since the Jacobian involves derivatives of the function $\mathbf{g}_{\boldsymbol{\theta}}$ with respect to its inputs $\mathbf{x}$, this evaluation can again be performed through automatic differentiation. Overall, it can be shown [4] that both forward and backward mode automatic differentiation for a L-layer, fully connected neural network scale as $\mathcal{O}(LD^3)$, with L the number of layers. This is prohibitive in many practical applications with a large data dimension D.

Normalizing flows with simple Jacobians An approach to alleviate the computational cost of this operation is to deploy special neural network architectures for which the evaluation of $\mathcal{L}_J$ is trivial. For example, in autoregressive normalizing flows [14, 15, 34, 25] the Jacobian of the transformation is constrained to be lower triangular. In this case, its determinant can be trivially computed with a linear cost in D. Notice however that the computational cost of the forward pass still scales quadratically in D; the overall complexity of forward plus backward pass is therefore still quadratic in the input size [44].

Most critically, such architectures imply a strong restriction on the class of transformations that can be learned. While it can be shown, based on [29], that under certain conditions this class of functions has universal approximation capacity for *densities* [25], that is less general than other notions of universal approximation [23, 24]. In fact it is obvious that functions with such triangular Jacobians cannot be universal approximators of *functions*, since, for example, the first variable can only depend on the first variable. This is a severe problem in learning features for disentanglement, for example by nonlinear ICA [27, 30], which would usually require unconstrained Jacobians. In other words, such restrictions might imply that the deployed networks are not general purpose: [5] showed that constrained designs typically used for density estimation can severely hurt discriminative performance. We further elaborate on this point in appendix E. Note that fully connected modules have elsewhere been termed *linear* flows [42], and are a strict generalization of autoregressive flows.[4]

3 Log-determinant of the Jacobian for fully connected neural networks

As a first step toward efficient optimization of the $\mathcal{L}_J$ term, we next provide the explicit form of the Jacobian for fully connected neural networks. As a starting point, notice that invertible and

[3] Note that invertible neural networks provide the possibility to not save, but rather recompute the intermediate activations during the backward pass, thus providing a memory efficient approach to backpropagation [18].

[4] Comprehensive reviews on normalizing flows can be found in [42, 35]. Other related methods are reviewed in appendix B.

differentiable transformations are *composable*; given any two such transformations, their composition is also invertible and differentiable. Furthermore, the determinant of the Jacobian of a composition of functions is given by the product of the determinants of the Jacobians of each function,

$$\det \mathbf{J}[\mathbf{g}_2 \circ \mathbf{g}_1](\mathbf{x}) = \det \mathbf{J}\mathbf{g}_2\left(\mathbf{g}_1(\mathbf{x})\right) \cdot \det \mathbf{J}\mathbf{g}_1(\mathbf{x})\,. \tag{8}$$

The log-determinant of the full Jacobian for a neural network therefore simply decomposes in a sum of the log-determinants of the Jacobians of each module, $\mathcal{L}_J(\mathbf{x}) = \sum_{k=1}^{L} \log|\det \mathbf{J}\mathbf{g}_k(\mathbf{z}_{k-1})|$. We will focus on the Jacobian term relative to a single submodule k with respect to its input $\mathbf{z}_{k-1}$; with a slight abuse of notation, we will call it $\mathcal{L}_J(\mathbf{z}_{k-1})$. As we remarked, fully connected $\mathbf{g}_k$ are themselves compositions of a linear operation and an element-wise invertible nonlinearity; applying the same reasoning, we then have

$$\mathcal{L}_J(\mathbf{z}_{k-1}) = \sum_{i=1}^{D} \log\left|\sigma'(y_k^i)\right| + \log|\det \mathbf{W}_k| \;=: \mathcal{L}_J^1(\mathbf{y}_k) + \mathcal{L}_J^2(\mathbf{z}_{k-1})\,. \tag{9}$$

where $\mathbf{y}_k = \mathbf{W}_k\mathbf{z}_{k-1}$. The first term $\mathcal{L}_J^1$ is a sum of univariate functions of single components of the output of the module, and it can be evaluated easily with few additional operations on top of intermediate outputs of a forward pass; gradients with respect to it can be simply computed via backpropagation, not unlike the $\mathcal{L}_p$ term introduced in section 2.3.

The second term $\mathcal{L}_J^2$ however involves a nonlinear function of the determinant of the weight matrix. From matrix calculus, we know that the derivative is equal to

$$\frac{\partial \log|\det \mathbf{W}_k|}{\partial \mathbf{W}_k} = \left(\mathbf{W}_k^\top\right)^{-1}\,. \tag{10}$$

Therefore, the computation of the gradient relative to such term involves a matrix inversion, with cubic scaling in the input size.[5] For a fully connected neural network of L layers, given that we have one such operation to perform for each of the layers, the gradient computation for these terms alone would have a complexity of $\mathcal{O}(LD^3)$, thus matching the one which would be obtained if the Jacobian were to be computed via automatic differentiation as discussed in section 2.

It can therefore be seen that these inverses of the weight matrices are the problematic element in the gradient computation. In the next section, we show how this problem can be solved using relative gradients.

4 Relative gradient descent for neural networks

We now derive the basic form of the relative gradient, following the approach in [11].[6] The starting point is that the parameters in a neural networks are matrices, in particular invertible in our case. Thus, we can make use of the geometric properties of invertible matrices, while they are usually completely neglected in gradient optimization in neural networks.

Relative gradient based on multiplicative perturbation In a classical gradient approach for optimization, we add a small vector $\boldsymbol{\epsilon}$ to a point $\mathbf{x}$ in a Euclidean space. However, with matrices, we are actually perturbing a matrix with another, and this can be done in different ways. In the relative gradient approach, we make a *multiplicative* perturbation of the form

$$\mathbf{W}_k \to (\mathbf{I} + \boldsymbol{\epsilon})\mathbf{W}_k \tag{11}$$

where $\boldsymbol{\epsilon}$ is an infinitesimal matrix. If we consider the effect of such a perturbation on a scalar-valued function $f(\mathbf{W}_k)$, we have

$$f((\mathbf{I} + \boldsymbol{\epsilon})\mathbf{W}_k) - f(\mathbf{W}) = \langle \nabla f(\mathbf{W}_k), \boldsymbol{\epsilon}\mathbf{W}_k\rangle + o(\mathbf{W}_k) = \langle \nabla f(\mathbf{W}_k)\mathbf{W}_k^\top, \boldsymbol{\epsilon}\rangle + o(\mathbf{W}_k) \tag{12}$$

which shows that the direction of steepest descent in this case is given by making $\boldsymbol{\epsilon} = \mu \nabla f(\mathbf{W}_k)\mathbf{W}_k^\top$ where μ is an infinitesimal step size. Furthermore, when we combine this $\boldsymbol{\epsilon}$ with the definition of a multiplicative update, we find that the best perturbation to $\mathbf{W}$ is actually given as

$$\mathbf{W}_k \to \mathbf{W}_k + \mu \nabla f(\mathbf{W}_k)\mathbf{W}_k^\top\mathbf{W}_k \tag{13}$$

[5] Though slightly more favorable exponents can in principle be obtained, see appendix C.

[6] For linear blind source separation, this approach also corresponds to the natural gradient, which can be justified with an information-geometric approach [2].

That is, the classical Euclidean gradient is replaced by $\nabla f(\mathbf{W}_k)\mathbf{W}_k^\top \mathbf{W}_k$, i.e. it is multiplied by $\mathbf{W}_k^\top \mathbf{W}_k$ from the right. This is the relative gradient.

A further alternative can be obtained by perturbing the weight matrices from the right, as $\mathbf{W}_k \to \mathbf{W}_k(\mathbf{I}+\boldsymbol{\epsilon})$. A similar derivation shows that in this case, the optimal $\boldsymbol{\epsilon}$ is given by $\mathbf{W}_k\mathbf{W}_k^\top \nabla f(\mathbf{W}_k)$; we refer to this as *transposed relative gradient*. In the context of linear ICA, the properties of the relative and transposed relative gradient were discussed in [49]. This version of the relative gradient might be useful in some cases; for example, the transposed relative gradient can be implemented more straightforwardly in neural network packages where the convention is that vectors are represented as rows.

The relative gradient belongs to the more general class of gradient descent algorithms on Riemannian manifolds [1]. Specifically, relative gradient descent is a first order optimization algorithm on the manifold of invertible $D \times D$ matrices. Almost sure convergence of the parameters to a critical point of the gradient of the cost function can be derived even for its stochastic counterpart, with decreasing step size and under suitable assumptions (see e.g. [8]).

Jacobian term optimization through the relative gradient In section 3, we showed that the difficulty in computing the gradient of the log-determinant is in the terms $\mathcal{L}_J^2$, whose gradient involves a matrix inversion. Now we show that by exploiting the relative gradient, this matrix inversion vanishes. In fact, when multiplying the right hand side of equation (10) by $\mathbf{W}_k^\top \mathbf{W}_k$ from the right we get

$$\left(\mathbf{W}_k^\top\right)^{-1} \mathbf{W}_k^\top \mathbf{W}_k = \mathbf{W}_k \,, \tag{14}$$

and similarly when multiplying by $\mathbf{W}_k\mathbf{W}_k^\top$ from the left. Most notably, we therefore have to perform *no additional operation* to get the relative gradient with respect to this term of the loss; it is, so to say, *implicitly* computed — as we know that the update for the parameters in $\mathbf{W}_k$ with respect to the error term $\mathcal{L}_J^2$ is proportional to $\mathbf{W}_k$ matrix itself.

As for the remaining terms of the loss, $\mathcal{L}_p$ and $\mathcal{L}_J^1$, simple backpropagation allows us to compute the weight updates given by the ordinary gradient in equation (6), which still need to be multiplied by $\mathbf{W}_k^\top \mathbf{W}_k$ to turn it into a relative gradient. We will next see that we can do this avoiding matrix-matrix multiplications, which would be computationally expensive. Note that backpropagation necessarily computes the $\boldsymbol{\delta}_k$ vector in equation (6) and for our model, by applying the relative gradient carefully, we can avoid matrix-matrix multiplication altogether by computing

$$(\Delta \mathbf{W}_k)\, \mathbf{W}_k^\top \mathbf{W}_k \propto \mathbf{z}_{k-1} \left(\left(\boldsymbol{\delta}_k^\top \mathbf{W}_k^\top \right) \mathbf{W}_k \right) \,. \tag{15}$$

Thus, we have a cheap method for computing the gradient of the log-determinant of the Jacobian, and of our original objective function. In appendix D we provide an explanation of how our procedure can be implemented with relative ease on top of existing deep learning packages.

While we so far only discussed update rules for the weight matrices of the neural network, our approach can be extended to include biases. Including bias terms in our multilayer network endows it with stronger approximation capacity. We detail how to do this in appendix F.

Complexity Note that the parentheses in equation (15) stress the point that the relative gradient updates only require matrix-vector or vector-vector multiplications, each of which scales as $\mathcal{O}(D^2)$, in a fixed number at each layer; that is, overall $\mathcal{O}(LD^2)$ operations. They therefore do not increase the complexity of a normal forward pass. Furthermore, the overall complexity with respect to the input size is quadratic, resulting in an overall quadratic scaling with the input size as in normalizing flow methods [44], but without imposing strong restrictions on the Jacobian of the transformation.

Extension to convolutional layers As we remarked in section 2.2, the formalism we introduced includes convolutional neural networks (CNNs) [36]. A natural question is therefore whether our approach can be extended to that case. The first natural question pertains the invertibility of convolutional neural networks; the convolution operation was shown [39] to be invertible under mild conditions (see appendix G), and the standard pooling operation can be by replaced an invertible operation [31]. We therefore believe that the general formalism can be applied to CNNs; this would require the derivation of the relative gradient for tensors. We believe that this should be possible but leave it for future work.

Invertibility and generation Given that invertible and differentiable transformations are composable, as discussed in section 3, invertibility of our learned transformation is guaranteed as long as the

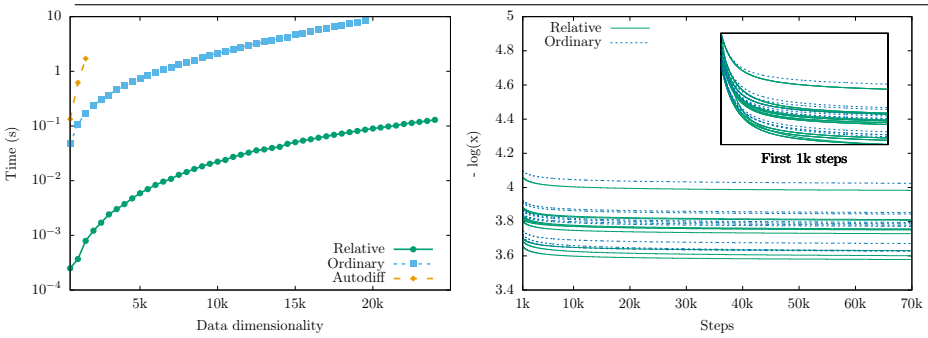


Figure 1: **Left:** Comparison of the average computation times of a single evaluation of the gradient of the log-likelihood; the standard error of the mean is not reported as it is orders of magnitude smaller then the scale of the plot. **Right:** Time-evolution of the negative log-likelihood for deterministic full-batch optimization for the two methods with the same initial points.

weight matrices and the element-wise nonlinearities are invertible. Square and randomly initialized (e.g. with uniform or normally distributed entries) weight matrices are known to be invertible with probability one; invertibility of the weight matrices throughout the training is guaranteed by the fact that the $\mathcal{L}_J^2$ terms would go to minus infinity for singular matrices (though high learning rates and numerical instabilities might compromise it in practice), as in estimation methods for linear ICA [6, 11, 26]. We additionally employ nonlinearities which are invertible by construction; we include more details about this in appendix H. If we are interested in data generation, we also need to invert the learned function. In practice, the cost of inverting each of the matrices is $\mathcal{O}(D^3)$, but the operation needs to be performed only once. As for the nonlinear transformation, the inversion is cheap since we only need to numerically invert a scalar function, for which often a closed form is available.

5 Experiments

In the following we experimentally verify the computational advantage of the relative gradient. The code used for our experiments can be found at `https://github.com/fissoreg/relative-gradient-jacobian`.

Computation of relative vs. ordinary gradient As a first step, we empirically verify that our proposed procedure using the formulas in section 4 leads to a significant speed-up in computation of the gradient of the Jacobian term. We compare the relative gradient against an explicit computation of the ordinary gradient, as described in section 3, and with a computation based on automatic differentiation, as discussed in section 2.3, where the Jacobian is computed with the JAX package [10]. While the output and asymptotic computational complexity of the ordinary gradient and automatic differentiation methods should be the same, a discrepancy is to be expected at finite dimensionality due to differences in how the computation is implemented. In the experiment, we generate 100 random normally distributed datapoints and vary the dimensionality of the data from 10 to beyond 20,000. We then define a two-layer neural network and evaluate the gradient of the Jacobian. The main comparison is run on a Tesla P100 Nvidia GPU. For the main plots, we deactivated garbage collection. Plots with CPU and further details on garbage collection can be found in appendix H.1. For each dimension we computed 10 iterations with a batch size of 100. Results are shown in figure 1, left. On the y-axis we report the average of the execution times of 100 successive gradient evaluations (forward plus backward pass in the automatic differentiation case). It can be clearly seen that *the relative gradient is much faster*, typically by two orders of magnitude. Autodiff computations could actually only be performed for the smallest dimension due to a memory problem. We report additional details on memory consumption in appendix H.1.

Optimization by relative vs. ordinary gradient Since our paper is, to the best of our knowledge, the first one proposing relative gradient optimization for neural networks (though other kinds of natural gradients have been studied [2]), we want to verify that the learning dynamics induced by the

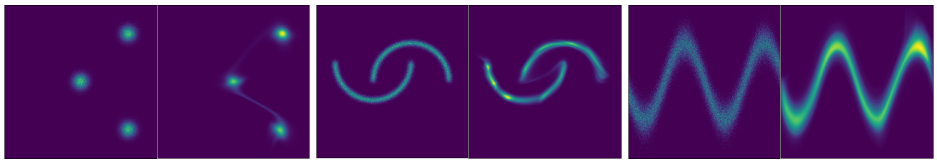


Figure 2: Illustrative examples of 2D density estimation. Samples from the true distribution and predicted densities are shown, in this order, side by side.

relative as opposed to the ordinary gradient do not bias the training procedure towards less optimal solutions or create other problems. We therefore perform a deterministic (full batch) gradient descent for both the relative and the ordinary gradient.[7] We employ 1,000 datapoints of dimensionality 2 and a two-layer neural network. We take 10 initial points and initialize both kinds of gradient descent at those same points. On the x-axis we plot the training epoch, while on the y-axis we plot the value of the loss. Figure 1, right shows the results: there is no big difference between the two gradient methods. There may actually be a slight advantage for the relative gradient, but that is immaterial since our main point here is merely to show that the *relative gradient does not need more iterations* to give the same performance.

Combining these two results, we see that the proposed relative gradient approach leads to a *much faster optimization* than the ordinary gradient. Perhaps surprisingly, the results exhibit a rather constant speed-up factor of the order of 100 although the theory says it should be changing with the dimension D; in any case, the difference is very significant in practice.

Density estimation Although our main contribution is the computational speed-up of the gradient computation demonstrated above, we further show some simple results on density estimation to highlight the potential of the relative gradient used in conjuction with the unconstrained factorial approximation in section 2.1. We use a fairly simple feedforward neural network with a smooth version of leaky-ReLU as activation function. Our empirical results show that this system, despite having quite *minimal fine-tuning* (details in appendix H.3), *achieves competitive results on all the considered datasets* compared with existing models—which are all tailored and fine-tuned for density estimation. First, we show in Figure 2 different toy examples that showcase the ability of our method to convincingly model arbitrarily complex densities. Second, in order to show the viability of our method in comparison with well-established methods we perform, as in [43], unconditional density estimation on four different UCI datasets [16] and a dataset of natural image patches (BSDS300) [41], as well as on MNIST [37]. The results are shown in Table 1. To achieve a fair comparison across models, the number of parameters was tuned so that the number of trainable parameters are as similar as possible. Note that, as we can perform every computation efficiently, all the experiments are suitable to run on usual hardware, thus avoiding the need of hardware accelerators such as GPUs. As a final remark, the reported results make no use of batch normalization, dropout, or learning-rate scheduling. Therefore, it is sensible to expect even better results by including them in future work.

Table 1: Test log-likelihoods (higher is better) on unconditional density estimation for different datasets and models (same as in Table 1 of [43]). Models use a similar number of parameters; results show mean and two standard deviations. Best performing models are in bold. More details in appendix H.3

	POWER	GAS	HEPMASS	MINIBOONE	BSDS300	MNIST
Ours	0.065 ± 0.013	6.978 ± 0.020	-21.958 ± 0.019	-13.372 ± 0.450	151.12 ± 0.28	-1375.2 ± 1.4
MADE	-3.097 ± 0.030	3.306 ± 0.039	-21.804 ± 0.020	-15.635 ± 0.498	146.37 ± 0.28	-1380.8 ± 4.8
MADE MoG	$\mathbf{0.375 \pm 0.013}$	7.803 ± 0.022	$\mathbf{-18.368 \pm 0.019}$	-12.740 ± 0.439	150.84 ± 0.27	$\mathbf{-1038.5 \pm 1.8}$
Real NVP (10)	0.182 ± 0.014	$\mathbf{8.357 \pm 0.019}$	-18.938 ± 0.021	$\mathbf{-11.795 \pm 0.453}$	$\mathbf{153.28 \pm 1.78}$	-1370.7 ± 10.1
Real NVP (5)	-0.459 ± 0.010	6.656 ± 0.020	-20.037 ± 0.020	-12.418 ± 0.456	151.76 ± 0.27	-1323.2 ± 6.6
MAF (5)	-0.458 ± 0.016	7.042 ± 0.024	-19.400 ± 0.020	-11.816 ± 0.444	149.22 ± 0.28	-1300.5 ± 1.7
MAF (10)	-0.376 ± 0.017	7.549 ± 0.020	-25.701 ± 0.025	-11.892 ± 0.459	150.46 ± 0.28	-1313.1 ± 2.0
MAF MoG (5)	0.192 ± 0.014	7.183 ± 0.020	-22.747 ± 0.017	-11.995 ± 0.462	152.58 ± 0.66	-1100.3 ± 1.6

[7] Notice that there's no need to compare to autodiff in this case because the computed gradient should be exactly the same as the ordinary gradient with the formulas in section 3.

6 Conclusions

Using relative gradients, we proposed a new method for exact optimization of objective functions involving the log-determinant of the Jacobian of a neural network, as typically found in density estimation, nonlinear ICA, and related tasks. This allows for employing models which, unlike typical alternatives in the normalizing flows literature, have no strong limitation on the structure of the Jacobian. We use modules with fully connected layers, thus strictly generalizing normalizing flows with triangular Jacobians, while still supporting efficient combination of forward and backward pass. These neural networks can represent a larger function class than autoregressive flows, which, despite being universal approximators for density functions, can only represent transformations with triangular Jacobians. Our method can therefore provide an alternative in settings where more expressiveness is needed to learn a proper inverse transformation, such as in identifiable nonlinear ICA models.

The relative gradient approach proposed here is quite simple, yet rather powerful. The importance of the optimization of the log-determinant of the Jacobian is well-known, but it has not been previously shown that there is a way around its difficulty without restricting expressivity. Now that we have shown that the optimization of this term can be done quite cheaply, a substantial fraction of the research in the field can be reformulated in stronger terms and with more generality.

Broader impact

As this paper presents novel theoretical results in unsupervised learning, the authors do not see any immediate ethical or societal concern. An important aspect of our paper is the improvement in computational efficiency with respect to naive methods. This can hopefully lead to reduced energy consumption to achieve comparable model performance.

Acknowledgments

A.H. was supported by a Fellowship from CIFAR, and by the DATAIA convergence institute as part of the "Programme d'Investissement d'Avenir", (ANR-17-CONV-0003) operated by Inria.
L.G. started working on this project while on an ELLIS exchange, hosted by the Parietal team at Inria, Saclay.
We thank Vincent Stimper, Patrick Putzky, Cyril Furtlehner and Ilyes Khemakhem for valuable comments on an earlier draft of this paper, and Isabel Valera and Guillaume Charpiat for helpful comments and tips. L.G. additionally thanks Roma Beaufret and Alexis Bozio for helpful support.

References

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

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

[3] Leemon Baird, David Smalenberger, and Shawn Ingkiriwang. One-step neural network inversion with pdf learning and emulation. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 966–971. IEEE, 2005.

[4] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18(153), 2018.

[5] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582, 2019.

[6] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[7] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[8] Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[9] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[10] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018.

[11] J-F Cardoso and Beate H Laheld. Equivariant adaptive source separation. *IEEE Transactions on signal processing*, 44(12):3017–3030, 1996.

[12] Tian Qi Chen and David K Duvenaud. Neural networks with cheap differential operators. In *Advances in Neural Information Processing Systems*, pages 9961–9971, 2019.

[13] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6572–6583, 2018.

[14] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[15] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *arXiv preprint arXiv:1605.08803*, 2016.

[16] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[17] Marc Finzi, Pavel Izmailov, Wesley Maddox, Polina Kirichenko, and Andrew Gordon Wilson. Invertible convolutional networks. 2019.

[18] Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. In *Advances in neural information processing systems*, pages 2214–2224, 2017.

[19] Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[20] Luigi Gresele, Paul K. Rubenstein, Arash Mehrjou, Francesco Locatello, and Bernhard Schölkopf. The Incomplete Rosetta Stone problem: Identifiability results for multi-view nonlinear ICA. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, page 53. AUAI Press, 2019.

[21] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*, volume 105. Siam, 2008.

[22] Emiel Hoogeboom, Rianne van den Berg, and Max Welling. Emerging convolutions for generative normalizing flows. *arXiv preprint arXiv:1901.11137*, 2019.

[23] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366, July 1989.

[24] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[25] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. *arXiv preprint arXiv:1804.00779*, 2018.

[26] Aapo Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999.

[27] Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Advances in Neural Information Processing Systems*, pages 3765–3773, 2016.

[28] Aapo Hyvärinen and Hiroshi Morioka. Nonlinear ICA of temporally dependent stationary sources. volume 54. Proceedings of Machine Learning Research, 2017.

[29] Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

[30] Aapo Hyvärinen, Hiroaki Sasaki, and Richard E Turner. Nonlinear ICA using auxiliary variables and generalized contrastive learning. *arXiv preprint arXiv:1805.08651*, 2018.

[31] Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-RevNet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*, 2018.

[32] Mahdi Karami, Dale Schuurmans, Jascha Sohl-Dickstein, Laurent Dinh, and Daniel Duckworth. Invertible convolutional flow. In *Advances in Neural Information Processing Systems*, pages 5636–5646, 2019.

[33] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[34] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

[35] Ivan Kobyzev, Simon Prince, and Marcus A Brubaker. Normalizing flows: Introduction and ideas. *arXiv preprint arXiv:1908.09257*, 2019.

[36] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[37] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The MNIST database of handwritten digits, 1998. *URL http://yann. lecun. com/exdb/mnist*, 10:34, 1998.

[38] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

[39] Fangchang Ma, Ulas Ayaz, and Sertac Karaman. Invertibility of convolutional generative networks from partial measurements. In *Advances in Neural Information Processing Systems*, pages 9628–9637, 2018.

[40] Charles C Margossian. A review of automatic differentiation and its efficient implementation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(4):e1305, 2019.

[41] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001.

[42] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

[43] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

[44] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

[45] Oren Rippel and Ryan Prescott Adams. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013.

[46] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[47] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[48] Matthias W Seeger and Hannes Nickisch. Large scale variational inference and experimental design for sparse generalized linear models. *arXiv preprint arXiv:0810.0901*, 2008.

[49] Stefano Squartini, Francesco Piazza, and Ali Shawker. New Riemannian metrics for improvement of convergence speed in ICA based learning algorithms. In *2005 IEEE International Symposium on Circuits and Systems*, pages 3603–3606. IEEE, 2005.

[50] Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

[51] Esteban G Tabak, Eric Vanden-Eijnden, et al. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

[52] Jakub M Tomczak and Max Welling. Improving variational auto-encoders using Householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

[53] Rianne Van Den Berg, Leonard Hasenclever, Jakub M Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 393–402. Association For Uncertainty in Artificial Intelligence (AUAI), 2018.

[54] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[55] Wikipedia. Computational complexity of mathematical operations — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Computational%20complexity%20of%20mathematical%20operations&oldid=958179308`, 2020. [Online; accessed 11-June-2020].

[56] Jianxin Wu. Introduction to convolutional neural networks. 2017.

Bibliography

[1] H. Ackley, E. Hinton, and J. Sejnowski. "A learning algorithm for Boltzmann machines". In: *Cognitive Science* (1985), pp. 147–169.

[2] E. Agliari et al. "Multitasking Associative Networks". In: *Phys. Rev. Lett.* 109 (26 2012), p. 268101.

[3] J R L de Almeida and D J Thouless. "Stability of the Sherrington-Kirkpatrick solution of a spin glass model". In: 11.5 (May 1978), pp. 983–990. DOI: 10.1088/0305-4470/11/5/028. URL: https://doi.org/10.1088/0305-4470/11/5/028.

[4] Shun-ichi Amari. "Natural Gradient Works Efficiently in Learning". In: *Neural Computation* 10.2 (Feb. 1998), pp. 251–276. ISSN: 0899-7667. DOI: 10.1162/089976698300017746. eprint: https://direct.mit.edu/neco/article-pdf/10/2/251/813415/089976698300017746.pdf. URL: https://doi.org/10.1162/089976698300017746.

[5] D. J. Amit, H. Gutfreund, and H. Sompolinsky. "Spin-glass models of neural networks". In: *Phys. Rev. A* 32 (1985), pp. 1007–1018.

[6] D. J. Amit, H. Gutfreund, and H. Sompolinsky. "Statistical Mechanics of Neural Networks near Saturation". In: *Annals of Physics* 173.1 (1987), pp. 30–67.

[7] D. J. Amit, H. Gutfreund, and H. Sompolinsky. "Storing Infinite numbers of patterns in a Spin-Glass Model of Neural Networks". In: *Phys. Rev. Lett.* 55.14 (1985), pp. 1530–1533.

[8] A. Barra et al. "Phase Diagram of Restricted Boltzmann Machines and Generalized Hopfield Networks with Arbitrary Priors". arXiv:1702.05882. 2017.

[9] Adriano Barra et al. "On the equivalence of Hopfield networks and Boltzmann Machines". In: *Neural Networks* 34 (2012), pp. 1–9.

[10] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[11] Erwin Bolthausen. *An iterative construction of solutions of the TAP equations for the Sherrington-Kirkpatrick model*. 2012. arXiv: 1201.2891 [math.PR].

[12] H. Bourlard and Y. Kamp. "Auto-association by multilayer perceptrons and singular value decomposition". In: *Biological Cybernetics* 59.4 (1988), pp. 291–294.

[13] J. Bruck. "On the convergence properties of the Hopfield model". In: *Proceedings of the IEEE* 78.10 (1990), pp. 1579–1585. DOI: 10.1109/5.58341.

[14] Serhat Selcuk Bucak, Rong Jin, and Anil K. Jain. "Multi-label learning with incomplete class assignments". In: *CVPR 2011*. 2011, pp. 2801–2808. DOI: 10.1109/CVPR.2011.5995734.

[15] J-F Cardoso. "Learning in manifolds: The case of source separation". In: *Ninth IEEE Signal Processing Workshop on Statistical Signal and Array Processing (Cat. No. 98TH8381)*. IEEE. 1998, pp. 136–139.

[16] Jean-François Cardoso and Beate Laheld. "Equivariant Adaptive Source Separation". In: *IEEE Trans. on Signal Processing* 44 (1996), pp. 3017–3030.

[17] Giuseppe Carleo et al. "Machine learning and the physical sciences". In: *Reviews of Modern Physics* 91.4 (Dec. 2019). ISSN: 1539-0756. DOI: 10.1103/revmodphys.91.045002. URL: http://dx.doi.org/10.1103/RevModPhys.91.045002.

[18] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006. ISBN: 0471241954.

[19] A. Decelle, G. Fissore, and C. Furtlehner. "Spectral dynamics of learning in restricted Boltzmann machines". In: *EPL* 119.6 (2017), p. 60001.

[20] A. Decelle, G. Fissore, and C. Furtlehner. "Thermodynamics of Restricted Boltzmann Machines and Related Learning Dynamics". In: *Journal of Statistical Physics* 172.6 (July 2018), pp. 1576–1608. ISSN: 1572-9613. DOI: 10.1007/s10955-018-2105-y. URL: http://dx.doi.org/10.1007/s10955-018-2105-y.

[21] Aurélien Decelle and Cyril Furtlehner. *Restricted Boltzmann Machine, recent advances and mean-field theory*. 2021. arXiv: 2011.11307 [cond-mat.dis-nn].

[22] Aurélien Decelle, Cyril Furtlehner, and Beatriz Seoane. "Equilibrium and non-Equilibrium regimes in the learning of Restricted Boltzmann Machines". In: *CoRR* abs/2105.13889 (2021). arXiv: 2105.13889. URL: https://arxiv.org/abs/2105.13889.

[23] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[24] Laurent Dinh, David Krueger, and Yoshua Bengio. "NICE: Non-linear Independent Components Estimation". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1410.8516.

[25] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. "Augmented Neural ODEs". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/21be9a4bd4f81549a9d1d241981cec3c-Paper.pdf.

[26] Asja Fischer and C. Igel. "Training restricted Boltzmann machines: An introduction". In: *Pattern Recognit.* 47 (2014), pp. 25–39.

[27] Giancarlo Fissore et al. "Robust Multi-Output Learning with Highly Incomplete Data via Restricted Boltzmann Machines". In: *Proceedings of the 9th European Starting AI Researchers' Symposium 2020 co-located with 24th European Conference on Artificial Intelligence (ECAI 2020)* (2020).

[28] Marylou Gabrié, Eric W. Tramel, and Florent Krzakala. "Training Restricted Boltzmann Machines via the Thouless-Anderson-Palmer Free Energy". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*. NIPS'15. Montreal, Canada, 2015, pp. 640–648.

[29] E Gardner and B Derrida. "Optimal storage properties of neural network models". In: *Journal of Physics A: Mathematical and General* 21.1 (1988), p. 271.

[30] E. Gardner. "Maximum Storage Capacity in Neural Networks". In: *EPL (Europhysics Letters)* 4.4 (1987), p. 481.

[31] A Georges and J S Yedidia. "How to expand around mean-field theory using high-temperature expansions". In: *Journal of Physics A: Mathematical and General* 24.9 (May 1991), pp. 2173–2192. DOI: 10.1088/0305-4470/24/9/024. URL: https://doi.org/10.1088/0305-4470/24/9/024.

[32] Roy J. Glauber. "Time Dependent Statistics of the Ising Model". In: *Journal of Mathematical Physics* 4.2 (1963), pp. 294–307.

[33] Ian J. Goodfellow et al. "Generative Adversarial Nets." In: *NIPS*. Ed. by Zoubin Ghahramani et al. 2014, pp. 2672–2680. URL: http://dblp.uni-trier.de/db/conf/nips/nips2014.html#GoodfellowPMXWOCB14.

[34] W. Grathwohl et al. "Scaling RBMs to High Dimensional Data with Invertible Neural Networks". In: *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models* (2020).

[35] Luigi Gresele et al. "Relative gradient optimization of the Jacobian term in unsupervised deep learning". In: *Advances in Neural Information Processing Systems* 33 (2020).

[36] D.O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Taylor & Francis, 2005. ISBN: 9781135631901. URL: https://books.google.fr/books?id=ddB4AgAAQBAJ.

[37] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation*. USA: Addison-Wesley Longman Publishing Co., Inc., 1991. ISBN: 0201503956.

[38] G. E. Hinton. "Training products of experts by minimizing Contrastive divergence". In: *Neural computation* 14 (2002), pp. 1771–1800.

[39] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[40] Geoffrey E. Hinton. " A Practical Guide to Training Restricted Boltzmann Machines". In: *Neural Networks: Tricks of the Trade: Second Edition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 599–619.

[41] Emiel Hoogeboom, Rianne van den Berg, and Max Welling. "Emerging Convolutions for Generative Normalizing Flows". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2771–2780. URL: http://proceedings.mlr.press/v97/hoogeboom19a.html.

[42] J. J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities". In: *Proceedings of the National Academy of Sciences of the United States of America* 79.8 (1982), pp. 2554–2558.

[43] H. Huang. "Statistical mechanics of unsupervised feature learning in a restricted Boltzmann machine with binary synapses". In: *Journal of Statistical Mechanics: Theory and Experiment* 2017.5 (2017), p. 053302.

[44] Haiping Huang and Taro Toyoizumi. "Advanced mean-field theory of the restricted Boltzmann machine". In: *Physical Review E* 91.5 (2015), p. 050101.

[45] E. T. Jaynes. "Information Theory and Statistical Mechanics". In: *Phys. Rev.* 106 (4 May 1957), pp. 620–630. DOI: 10.1103/PhysRev.106.620. URL: https://link.aps.org/doi/10.1103/PhysRev.106.620.

[46] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 2014. arXiv: http://arxiv.org/abs/1312.6114v10 [stat.ML].

[47] Durk P Kingma and Prafulla Dhariwal. "Glow: Generative Flow with Invertible 1x1 Convolutions". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[49] Hugo Larochelle et al. "Learning Algorithms for the Classification Restricted Boltzmann Machine". In: *Journal of Machine Learning Research* 13.22 (2012), pp. 643–669. URL: http://jmlr.org/papers/v13/larochelle12a.html.

[50] Yann LeCun and Corinna Cortes. "MNIST handwritten digit database". In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

[51] Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. *MisGAN: Learning from Incomplete Data with Generative Adversarial Networks*. 2019. arXiv: 1902.09599 [cs.LG].

[52] R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 2002. ISBN: 9780471183860. URL: http://books.google.com/books?id=aYPwAAAAMAAJ.

[53] Zhou Lu et al. "The Expressive Power of Neural Networks: A View from the Width". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/32cbf687880eb1674a07bf717761dd3a-Paper.pdf.

[54] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.

[55] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Copyright Cambridge University Press, 2003.

[56] V. A. Marchenko and L. A. Pastur. "Distribution of eigenvalues for some sets of random matrices". In: *Mathematics of the USSR-Sbornik* 1.4 (1967), p. 457. URL: http://stacks.iop.org/0025-5734/1/i=4/a=A01.

[57] James Martens. "New Insights and Perspectives on the Natural Gradient Method". In: *Journal of Machine Learning Research* 21.146 (2020), pp. 1–76. URL: http://jmlr.org/papers/v21/17-678.html.

[58] Pierre-Alexandre Mattei and Jes Frellsen. "MIWAE: Deep Generative Modelling and Imputation of Incomplete Data Sets". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 4413–4423. URL: https://proceedings.mlr.press/v97/mattei19a.html.

[59] J. McCarthy et al. *A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE*. http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html. 1955. URL: http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html.

[60] W. S. McCulloch and W. Pitts. "A Logical Calculus of Ideas Immanent in Nervous Activity". In: *Bulletin of Mathematical Biophysics* 5 (1943).

[61] M. Mézard, G. Parisi, and M. A. Virasoro. *Spin Glass Theory and Beyond*. World Scientific, Singapore, 1987.

[62] Marc Mézard. "Mean-field message-passing equations in the Hopfield model and its generalizations". In: *Phys. Rev. E* 95 (2 Feb. 2017), p. 022117. DOI: 10.1103/PhysRevE.95.022117. URL: https://link.aps.org/doi/10.1103/PhysRevE.95.022117.

[63] R. Monasson and J. Tubiana. "Emergence of Compositional Representations in Restricted Boltzmann Machines". In: *Phys. Rev. Let.* 118 (2017), p. 138301.

[64] A. Newell and H. Simon. "The logic theory machine–A complex information processing system". In: *IRE Transactions on Information Theory* 2.3 (1956), pp. 61–79. DOI: 10.1109/TIT.1956.1056797.

[65] M. J. Nijman and Kappen. *Using Boltzmann Machines to Fill in Missing Values*. 1994.

[66] H. Nishimori. *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. Oxford University Press, 2001.

[67] Aaron van den Oord et al. "Parallel WaveNet: Fast High-Fidelity Speech Synthesis". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 3918–3926. URL: https://proceedings.mlr.press/v80/oord18a.html.

[68] George Papamakarios et al. "Normalizing Flows for Probabilistic Modeling and Inference". In: *Journal of Machine Learning Research* 22.57 (2021), pp. 1–64. URL: http://jmlr.org/papers/v22/19-1028.html.

[69] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-propagating Errors". In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0. URL: http://www.nature.com/articles/323533a0.

[70] Andrew M. Saxe, James L. Mcclelland, and Surya Ganguli. "Exact solutions to the nonlinear dynamics of learning in deep linear neural network". In: *In International Conference on Learning Representations*. 2014.

[71] Claude E. Shannon. "A mathematical theory of communication." In: *Bell Syst. Tech. J.* 27.3 (1948), pp. 379–423. URL: http://dblp.uni-trier.de/db/journals/bstj/bstj27.html#Shannon48.

[72] David Sherrington and Scott Kirkpatrick. "Solvable Model of a Spin-Glass". In: *Phys. Rev. Lett.* 35 (26 1975), pp. 1792–1796.

[73] Yu-Yin Sun, Yin Zhang, and Zhi-Hua Zhou. "Multi-Label Learning with Weak Label". In: *AAAI*. 2010.

[74] D. J. Thouless, P. W. Anderson, and R. G. Palmer. "Solution of 'Solvable model of a spin glass'". In: *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics* 35.3 (1977), pp. 593–601. DOI: 10.1080/14786437708235992. eprint: https://doi.org/10.1080/14786437708235992. URL: https://doi.org/10.1080/14786437708235992.

[75] D. J. Thouless, P. W. Anderson, and R. G. Palmer. "Solution of 'Solvable model of a spin glass". In: *Philosophical Magazine* 35.3 (1977), pp. 593–601.

[76] T. Tieleman. "Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. 2008, pp. 1064–1071.

[77] M. E. Tipping and C. M. Bishop. "Mixtures of Probabilistic Principal Component Analyzers". In: *Neural Comput.* 11.2 (1999), pp. 443–482.

[78] Jakub M. Tomczak and Max Welling. "Improving Variational Auto-Encoders using Householder Flow". In: *CoRR* abs/1611.09630 (2016). arXiv: 1611.09630. URL: http://arxiv.org/abs/1611.09630.

[79] M. Mitchell Waldrop. *Complexity: the emerging science at the edge of order and chaos*. New York: Simon & Schuster, 1992. ISBN: 0671767895.

[80] M. Mitchell Waldrop. *The dream machine: J.C.R. Licklider and the revolution that made computing personal*. New York, N.Y., U.S.A.: Viking Penguin, 2002.

[81] Max Welling and Geoffrey E. Hinton. "A New Learning Algorithm for Mean Field Boltzmann Machines". In: *Artificial Neural Networks — ICANN 2002*. Ed. by José R. Dorronsoro. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 351–357. ISBN: 978-3-540-46084-8.

[82] Norbert Wiener. *Cybernetics: or Control and Communication in the Animal and the Machine*. 2nd ed. Cambridge, MA: MIT Press, 1948.

[83] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. "GAIN: Missing Data Imputation using Generative Adversarial Nets". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 5689–5698. URL: https://proceedings.mlr.press/v80/yoon18a.html.

[84] Lenka Zdeborová and Florent Krzakala. "Statistical physics of inference: thresholds and algorithms". In: *Advances in Physics* 65.5 (2016), pp. 453–552.