Decision making with reciprocal chains and binary neural network models

George Stamatescu

March 19, 2020

Thesis submitted for the degree of
Doctor of Philosophy
in
Electrical and Electronic Engineering
at The University of Adelaide
Faculty of Engineering, Computer and Mathematical Sciences
School of Electrical and Electronic Engineering

THE UNIVERSITY
of ADELAIDE

Contents

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works. I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an University of Adelaide Divisional Scholarship, and through the Data to Decisions CRC Scholarship.

George Stamatescu

November 2019

Acknowledgements

First and foremost, I wish to express my gratitude and appreciation to my supervisors. To Lang, for not wavering in the encouragement of my scientific exploration since I was an undergraduate, provided there was enough mathematical rigour and clarity. The work presented here was developed by following these principles as closely as I could. To Ian, for encouraging and guiding my reading into diverse fields, and the connections between them, while still anchoring me toward fundamentals. The many hours of discussion have helped shape my approach to research. In many ways, I could not have had a better pair of people to work with, whose skills complement one another, and whose values are shared. This was my source of confidence as I travelled overseas in search of ideas.

I would like to thank Bert Kappen for hosting me at Radboud University in Nijmegen, and for donating his time to a student from abroad. My study took a decidedly non-linear turn at that point, but also swung the door toward machine learning and statistical physics wide open. The last year, after being introduced to the group in Torino, would not have been possible without the time spent together in Nijmegen.

I would like to give a special thanks to the people I have met through the course of my study. To Riley, for the many debates. To Duong, for his company and cheerful disposition. To Dominik and Hans for the many discussions about path integral control theory. To Giel for the sarcastic wit and introducing me to Balki's physics courses. To all the staff and students in Nijmegen for making me feel welcome. To Adrian for guiding me as I cobbled together code. To the Maths Learning Centre staff, in particular David and Nick for their support and instruction. Finally, to Carlo, Federica and Luca, for each of them putting their arm around me and helping me in the last year. Visiting Torino and Milan I saw how a research group can be a family, and a productive one at that.

As part of my degree, I was fortunate enough to receive funding to travel to different institutes and workshops. The Data to Decisions CRC top-up scholarships, and the University of Adelaide Research Abroad scholarship enabled me to visit Radboud University in the Netherlands for several months over two years. The School of Electrical Engineering also supported my travel to various conferences and workshops, as did the Australian Centre for Visual Technology. In addition, Lang deserves another round of thanks for his contribution to my travel in the last year.

I am deeply indebted by the many friendships I have outside of the lab. Wai Keen, for the long discussions and critical thinking offered, without nearly as much in return. My housemates the last two years, Rory and Elli, for making a home. Gem, Margot, Jeds and Alice for providing some routine. Sebastian for putting me up. To the friends at the Hectorville Football Club for building a community, and a place where I could switch off everything else. I would also like to thank Sara, Dennis, Jonno, Marlon and Kyron, amongst the many others.

Finally, I am deeply grateful to my parents, Adelina and Laurentiu, my brother Victor, and

Julia, Alex, Karina and Rowan. Your moral and material support made this possible.

Abstract

Automated decision making systems are relied on in increasingly diverse and critical settings. Human users expect such systems to improve or augment their own decision making in complex scenarios, in real time, often across distributed networks of devices. This thesis studies binary decision making systems of two forms. The first system is built from a reciprocal chain, a statistical model able to capture the intentional behaviour of targets moving through a statespace, such as moving towards a destination state. The first part of the thesis questions the utility of this higher level information in a tracking problem where the system must decide whether a target exists or not. The contributions of this study characterise the benefits to be expected from reciprocal chains for tracking, using statistical tools and a novel simulation environment that provides relevant numerical experiments. Real world decision making systems often combine statistical models, such as the reciprocal chain, with the second type of system studied in this thesis, a neural network. In the tracking context, a neural network typically forms the object detection system. However, the power consumption and memory usage of state of the art neural networks makes their use on small devices infeasible. This motivates the study of binary neural networks in the second part of the thesis. Such networks use less memory and are efficient to run, compared to standard full precision networks. However, their optimisation is difficult, due to the non-differentiable functions involved. Several algorithms elect to optimise surrogate networks that are differentiable and correspond in some way to the original binary network. Unfortunately, the many choices involved in the algorithm design are poorly understood. The second part of the thesis questions the role of parameter initialisation in the optimisation of binary neural networks. Borrowing analytic tools from statistical physics, it is possible to characterise the typical behaviour of a range of algorithms at initialisation precisely, by studying how input signals propagate through these networks on average. This theoretical development also yields practical outcomes, providing scales that limit network depth and suggesting new initialisation methods for binary neural networks.

Glossary

Abbreviations

AUC Area Under the Curve

CLT Central Limit Theorem

CNN Convolutional Neural Network

CSP Constraint Satisfaction Problem

EOC Edge of Chaos

ERM Empirical Risk Minimisation

HMC Hidden Markov Chain

HRC Hidden Reciprocal Chain

HSC Hidden Schrödinger Chain

KL Kullback-Leibler (divergence)

LRT Likelihood Ratio Test

MAP Maximum *a posteriori* Probability

MB Markov Bridge

MC Markov Chain

MDP Markov Decision Process

MFT Mean Field Theory

RC Reciprocal Chain

ROC Receiver Operator Characteristic

SB Schrödinger Bridge

SGD Stochastic Gradient Descent

SK Sherrington-Kirkpatrick

TE Track Extraction

Chapter 1

General introduction

1.1 Motivation

This thesis is concerned with the study of automated decision making systems, and is divided into two parts. The first part, consisting of chapters 2 and 3, considers a statistical model, known as a reciprocal chain, used in a problem of target tracking. The specific problem considered is the case of targets whose dynamics depend not only on local constraints, such as physical obstacles, but also on higher levels of behaviour, such as an intention to proceed to a destination. Reciprocal chains are proposed as a model able to capture this sort of behaviour, and this claim is investigated. The second part of the thesis, consisting of chapters 4 through 8, studies large multi-layer neural networks used for generic classification tasks in a supervised learning context, in the difficult case that the parameters are constrained to have low precision. A theoretical study of new algorithms is presented, with experimental results on real data testing the theoretical predictions. As with any body of work, this thesis can be placed both within a domain of application, or a domain of knowledge, such as an established discipline. The two parts of the thesis are motivated by considering both perspectives.

In terms of a domain of application, modern target tracking systems are a prime example of large scale autonomous decision making systems which motivate both parts of the thesis. Tracking problems are widespread through many applications including automatic surveillance, vehicle navigation, video labelling, human-computer interaction and activity recognition. Furthermore, human users expect such systems to improve or augment their decision making in real time and for increasingly complex scenarios. The key concept motivating both reciprocal chain models and low-precision neural networks is the idea of computing "at the edge". This term refers to the notion that the data generated by a network of devices will be "stored, processed, analysed, and acted upon locally", close to or at the edge of the network of devices [1]. The need for such systems will increase with higher throughput of data, necessitating the move away from the paradigm of cloud computing [2], where data is relayed for central processing by a remote data centre, towards processing at the edge.

This thesis argues that target trackers based on models like reciprocal chains, and low-precision neural networks for object detection, are examples of new automated decision making systems that enable increasingly advanced computation and decision making at the edge. A target tracker based on reciprocal chains is an example of what could be called a "meta-level tracker", processing higher level information about targets. It is therefore envisaged that reciprocal chains could be suited for local, autonomous decision making, and outline examples under

a more detailed overview of modern tracking systems, in Section 1.2. Neural networks have proven to be highly successful at a range of tasks, but most prominently in the classification of objects in natural images [3]. Thus they have found widespread use within tracking systems. Most commonly, neural networks form the basis of object detectors in visual tracking, however they are often costly in terms of run-time or power consumption. It is therefore desirable to devise low precision variants of neural networks, in particular binary networks. The thesis also discusses the role of neural networks and hardware considerations within tracking systems in more detail in Section 1.2.

A second line of motivation for this thesis comes from taking an academic perspective on the work, which places the thesis within a discipline. The emerging field of "machine learning", which could be described as algorithmic decision making under uncertainty, is currently seeing the interaction of several established disciplines, including for example statistics, optimisation, control theory and computer vision, to name but a few. Both parts of the thesis can be comfortably placed in this new discipline. In the remainder of this opening motivational statement, both parts of the thesis are described in this context, touching on statistics and statistical learning theory, stochastic control theory, dynamical systems and optimisation.

From a statistical decision making point of view, the first part considers a problem of binary decision making assuming knowledge of the data generation model, in particular the reciprocal chain model for target dynamics. The second part of the thesis removes the assumption of this knowledge, and instead studies an algorithmic "learning" approach to decision making based on neural network models, in a supervised learning context. The two approaches can be distinguished as using, respectively, generative and discriminative models, each of which has advantages and disadvantages. Generative models typically have the advantage of being "more understandable" to the human user of such an automated decision making system, although they are seen as less flexible and more tedious to design. On the other hand, discriminative models, in particular neural networks, are flexible and relatively easy to deploy, but can appear as "black boxes" and as such may not leave a user confident in their decision making. Modern decision making systems will require elements of both modelling approaches. Indeed, as will be seen in the next section's overview of tracking systems, using both approaches in combination is common practice. The specific problems this thesis studies are illustrative of the differences in the modelling approaches, and the contributions made both improve the capabilities and the understanding of systems built from both models.

Decision making based on generative models, while purportedly more understandable, carries its own risks. In particular, if the assumed model does not accurately represent the phenomena of interest. This can be partially addressed through a higher level model selection inference steps, although this is generally computationally difficult. For reciprocal chain models, one can consider their validity by comparing them to alternative models of higher order target dynamics. Arguably the most common idea is to considering an "agent" that exists in a statespace, equipped with certain goals or objectives to achieve over some time frame. A mathematical formulation of this idea is known as a Markov decision process [4], which is generally considered by cognitive scientists as a useful model of human agency [5]. In such a model, the dynamics of the state evolution depends not only on the environment but also the actions or control the agent exerts, with the actions selected according to some rule, or policy. This thesis argues that reciprocal chains can capture qualitatively similar dynamics to those based on Markov decision process models of agency, without the additional framework of decision making that leads to an agent taking actions. It is also possible to formally relate reciprocal chains, in special cases, to a class

of Markov decision processes studied under the title of path integral or Kullback-Liebler control theory.

Discriminative models, such as neural networks, are optimised to make decisions without knowledge of the data generation model. For an image classification task, the basic process of "training" refers to the adaptation of the network parameters so as to correctly classify a finite set of appropriately labelled images, known as the training set. Eventually the network "learns" to make correct decisions on unseen data according to some rule. Formally, this rule corresponds to a decision boundary in the high dimensional feature space of images. Practically, this means that the network may generalise well from its training set to unseen data, assuming that the network is presented with data generated in the same manner, or distribution, both at training and test time.

Since the parameters of a neural network are not related to any causal model of the data generation process, as in a generative model, some users are concerned with a lack of interpretable processing when using networks with millions of parameters. However, this may not be crucial if neural networks are well understood, in terms of their training processes and generalisation error, as function of the data structure and the algorithm design. The second part of the thesis contributes toward building an understanding of the training process of neural networks. In particular, the role of initialisation in algorithms used to optimise networks with binary parameters is investigated. The theoretical analysis enables a concrete understanding of how to initialise the various algorithms, and suggests new algorithm designs.

The principal utility of a binary neural network, as mentioned, is the efficiency at run time, in terms of power consumption and computation speed, allowing for computing "at the edge", on low power devices. The downside to such potential benefits is the increased difficulty of the optimisation problem, since the standard continuous optimisation techniques developed for neural networks are not directly applicable. A broad class of algorithms for optimising binary neural networks take the route of appropriately defining each binary variable as a stochastic variable. Based on various approximations and leveraging the stochasticity, it is possible to derive continuous surrogate networks that are open to continuous optimisation methods. However, the various approximations are not well understood, in terms of specific implementation or comparison.

The second part of the thesis therefore develops and studies both new and existing algorithms for the optimisation of several continuous surrogates. The main theoretical contribution is the analytic treatment of the typical behaviour of such algorithms at initialisation. This is achieved based on a so called dynamic mean field theory, developed within the field of statistical physics. The application to neural networks amounts to studying how signals propagate through such networks on average, and is equivalent to controlling properties of the input-output Jacobian matrix of the network. The results of this analysis are a set of recursive equations governing the signal propagation, from which depth scales are derived that limit the maximum trainable depth of the networks. This thesis sets out to use this theory to study the hyper-parameter initialisation of these surrogates, and to categorise the networks based on the initialisation properties.

The idea of studying typical behaviour, rather than, for example, worst case behaviour, is borrowed from the discipline of statistical physics. Given that many of the theoretical and conceptual advances in neural network theory, including for the binary counterparts, have been advanced using tools from statistical physics, a significant proportion of the thesis discusses some of the fundamental ideas of this discipline. Indeed, the overall thrust of the second part of the thesis aims for a unified discussion of the various disciplines that come into play when

considering the design of neural network type decision making systems.

Several of the chapters of this thesis have led to publishable works, which are as follows.

Journal publications:

"Track extraction with hidden reciprocal chains" G. Stamatescu, L. B. White, R. Bruce-Doust, *IEEE Transactions on Automatic Control* 63 (4), 1097-1104, 2017.

Conference publications:

"Critical initialisation in continuous approximations of binary neural networks" G. Stamatescu, F. Gerace, C. Lucibello, I. Fuss, L. White. *International Conference on Learning Representations 2020*

"Multi-camera tracking of intelligent targets with Hidden Reciprocal Chains", G. Stamatescu, A. Dick, L. B. White, *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA).*

1.2 Modern tracking systems

This section provides a detailed description of modern tracking systems, the domain of application which motivates the study of both reciprocal chains and binary neural networks. This overview considers three elements that largely determine the design and performance of a tracking system. The first is the sensor information available in the particular tracking setting. This refers to both the sensor type, for example video camera tracking with colour information, as well as the average size of the targets, potentially measured in number of pixels. The second element is the target tracker design itself, meaning the various methods of data processing that take raw inputs and eventually produce information displayed to a human analyst. The third element includes the fundamental challenges faced by any tracking system, that are largely insensitive to the tracking setting or methodology. Given this overview, the role that could be played by reciprocal chains and low-precision neural networks is discussed, motivated by the advent of computing "at the edge" [1].

Sensor information is a principal determinant of the data processing and analysis methods that a tracking system will employ. Although many types of sensors exist, including radar and sonar, for the purpose of the subsequent discussion, only the case of visual tracking is considered. Within visual tracking, a range of different measurement quality can be produced. In particular, one can consider the gradual transition from object tracking to point target tracking, where under different conditions this may correspond to moving from many pixels per target to only a few pixels. Depending on where along this spectrum the sensor measurements sit, different methodologies will be more suitable for tracking.

In general, the primary components of a visual tracking system will include an object detector, appearance detector, motion model and state estimation algorithm, and finally data

association algorithm. Many successful object and appearance detectors are formed from neural networks, with a variety of different designs possible. Given a target's state, referring to a combination of its position and velocity, the motion models are used by the state estimation algorithm to predict the next state, for the data association step. Note that state estimation algorithms are typically of the Kalman filter type, with a standard Newtonian motion model being assumed. The data association algorithm attempts to correctly associate the set of simultaneous detections (within a frame) with a number of existing tracks (eg. from the previous frame). Of course, it is possible for new targets to appear and existing targets to disappear during the course of a video stream.

Thus a central task for a tracking system is to correctly identify targets of interest from background noise, or "clutter", meaning spurious measurements not originating from targets. In addition to this, a system may be required to track "online", meaning that it uses measurements only up until the current time to produce estimates. Otherwise, it may be "offline", processing a batch of data over some time interval. The subsequent discussion considers only the online tracking problem. This is appropriate the discussion is quite general and online tracking places an emphasis on the speed of the overall decision making system, a theme in line with processing "at the edge".

As a starting point, assume it is possible to produce object detections of high quality from the input data. In this context, a popular approach to visual tracking is known as "track-by-detect". This paradigm essentially relies on appearance models to successfully identify targets consistently from frame to frame, or camera to camera. If this is possible, the challenges of data association are largely bypassed. A recent example of a system built to operate under these assumptions is the SORT algorithm [6], which combines a neural network based object detector with a simple Kalman filter as its state estimation algorithm, and performs the data association only from frame to frame using a greedy search based on the Hungarian algorithm [7].

A neural network based object detector can be built in a variety of ways, and many "meta-architectures" exist [8], but a key ingredient is the so called "backbone" neural network. The backbone is typically a large neural network specialised for image classification, and a prominent example are so-called convolutional neural networks (CNNs) [3]. The backbone network is trained on a large training set of images not related directly to the camera scenes the tracker is applied to. The various meta-architectures then apply this backbone to classify different parts of a frame from the camera, with differences arising in the approaches toward scanning the frame. Due to the numerous convolutions performed by deep CNNs, for the problem of online or real-time tracking, the run-time of the object detectors can become an issue. Generally, smaller neural networks and less complex meta-architectures will have shorter run-time (as well as power consumption), at the expense of detection performance.

As the sensor information reduces, for example if the number of pixels per target decreases, in general the tracking algorithm will increasingly rely on spatio-temporal information to perform data association. This information is derived from the target motion model. In the visual tracking context, recent trackers have revisited earlier data association algorithms [9], including for example joint probabilistic data association [10] or the multiple hypothesis tracking algorithm [11], the latter of which is based on a Bayesian approach to data association. In the case that a data association algorithm relies purely on spatio-temporal information for the estimation of correct tracks, this method is referred to as "track-before-detect".

Several fundamental challenges within the problem of data association are have been alluded to throughout the discussion so far. A central issue is the combinatorial nature of reasoning

about multiple targets over some length of time, under uncertainty. Problems of temporary target occlusion, or emergence and disappearance of targets, that is, an time varying number of targets, compound these issues further. For a comprehensive discussion see [9].

Reciprocal chains were introduced in the previous section as statistical models that may be able to capture higher level target dynamics, such as moving through a statespace towards a destination. Within a tracking system, one can envisage reciprocal chains playing the role of a "meta-level" tracker. In this thesis, meta-level tracking refers to the estimation of quantities more complex than just position and velocity, relevant to targets whose dynamics that are not simply those described by, for example, Newtonian mechanics. Human motion is of course a prime example of generally non-Newtonian motion; people change direction at will, and an outside observer has no reason to believe a person will continue in a direction simply based on their current position and speed. It is argued throughout the thesis that meta-level trackers may serve two complementary roles; improving track estimates, and identifying behaviours of interest.

The first part of the thesis shows that reciprocal chains can improve track estimates and target confirmation, via simulation studies in novel environment models that include clutter. The second potential role of meta-level trackers, the identification of behaviours of interest is a direct extension to the contributions of the first part of the thesis. In the context of computing at the edge this idea is particularly attractive. Consider a surveillance task over a network of cameras, and suppose a meta-level tracker is able to determine that a target is behaving for instance in anomalous way. One can envisage a protocol where the meta-level tracker will decide locally (for example, at the camera site) to flag an analyst and direct other cameras towards the scene of interest. Furthermore, this local decision making may decide to direct higher quality video stream from the local camera to the cloud for central processing. This scenario may be beneficial in systems where bandwidth is limited, or the local data processing is constrained but central processing is not.

As discussed, modern visual tracking systems often employ neural network based image classifiers as "backbones" for the basis of object detectors with some meta-architecture, often also another neural network [8]. In the context of edge processing, the neural network designs that achieve the best results for image recognition, as an example, are both memory and power hungry, as well as often being slow to run. The general rule appears to be, the larger the network, the more accurate and the slower it will be, as evidenced by open, online benchmarking initiatives, eg. [12].

There is of course always some trade-off between an object detector's speed and accuracy, and much engineering effort is being devoted to designing specialised hardware to alleviate this issues for machine learning algorithms, and deep learning in particular [13]. Concurrently, however, there is interest in designing new algorithms that produce neural networks with better run-time and power consumption, without sacrificing accuracy. Low-precision networks of course reduce memory constraints, but networks with almost exclusively binary parameters also produce significant speed gains. A recent algorithm, XNOR-Networks [14], found that for a particular image classification neural network the evaluation time was decreased by a factor of 58, meaning it is feasible to run neural networks on CPU, as well as GPU hardware. In future, customised hardware will improve this run-time, as well as power consumption [13].

1.3 Introduction to decision making with statistical models

Automated decision making systems are becoming relied upon across increasingly diverse applications in society. Research fields such as "machine learning" and "artificial intelligence" aim to develop algorithms that are able to handle large, complex data sets and produce decisions that replace, or at least augment, human decision making. Successful algorithms that have garnered the most publicity, and arguably attention from researchers, rely on what are known as discriminative models. Since these models do not attempt to describe the data generation process with statistical model, they have been considered less "interpretable" by human users. In some contexts modelling the data generation process may not make sense. For example, modelling the generation process for natural images, as captured by a physical camera, may not give the interpretability apparently sought by practitioners, or users, from automated decision making systems.

In other cases however, modelling the data generation process is a quite a natural idea. For instance, if presented with a sequence of observations of an individual in some environment, then a human tasked with making predictions about future behaviour may attempt to consider, amongst other things, the beliefs and intentions of the individual. In cognitive science, the idea that part of an individual's social intelligence is to consider another individual's state of mind, is referred to as having a theory of mind [15].

A prominent mathematical model of agency is known as a Markov decision process (MDP) [4]. In this dynamic model, an agent exists within some environment which evolves according to some possibly random dynamics, which depend on the current state and the actions of the agent at each time. The agent is typically modelled as having some cost associated with their state and actions, and the objective is to find a policy to act under which would minimise the cost, over some time horizon, in some appropriate way. This could mean an average cost for the time horizon, where the averaging might consider "all possible" random events.

In observing an agent in some environment, the task of inferring the policy or goals of an agent from observations is known as inverse planning. In the last ten years, cognitive scientists have found experimental evidence in support of the idea that humans, at least approximately, perform inverse planning via a rational inference method codified as statistical Bayesian inference [5]. In the paper "action understanding as inverse planning" [5], human subjects were asked to predict the destinations of computer simulated targets in a 2D grid-world. The predictions of the subjects matched those of the Bayesian inference method, assuming an MDP agent model, across a range of differently encoded behaviours.

Optimal Bayesian inference requires the typically intractable computation of a posterior distribution, over either latent variables or parameters. In a tracking scenario, if one assumes knowledge of a simple model of target dynamics, then a standard inference task over latent variables includes, for example, state estimation. If, however, a target's dynamics were modelled as evolving according to an unknown MDP, the inference task also involves an inverse planning problem. Specifically, the determination of the policy the target is behaving according to, based on noisy observations. Thus in a Bayesian inference setting, the posterior distribution is defined over a much larger space, corresponding to a product of states, time and the space of policies. In this part of the thesis, a simpler model of intention is considered for the problem of tracking a target in an environment, with some intention to proceed toward a destination.

The specific class of models studied in this thesis, reciprocal models, are able to embody an idea of "source-destination awareness" (e.g. [16], [17],[18]). This is achieved via an imposed statistical relationship between the target's initial and terminal states, which can in turn place

higher weight on proceeding through a statespace in specific manners. In subsequent chapters it is outlined how reciprocal models relate directly to conditional processes such as Markov or Brownian bridges, and in turn certain classes of MDPs.

1.3.1 Key questions and contributions

The key questions driving chapters 2 and 3 are, respectively,

Key questions:

- How suitable are reciprocal chain models are suitable for the modelling and tracking of targets whose dynamics correspond to what one would recognise as exhibiting intent?
- What tracking benefit is obtained from using such "higher level" information?

In chapter 2, first the background theory for reciprocal chains is reviewed. After this a discussion of issues around the application of reciprocal chain models is presented, which is relevant to their suitability for modelling and tracking of targets which exhibit some kind of intent. A particular issue of focus is the fixed time interval nature of the model. In order to specify a joint distribution over start and end points of a target trajectory, one must know the time over which trajectories occur, or at least a number of "events" which define the time indices corresponding to target measurement. Following this discussion new Markov models are derived that can be compared qualitatively to Markov decision process models.

In chapter 3 the problem of track extraction is considered. Explicitly, the problem is to confirm target existence, or otherwise, in a set of observations of uncertain origin. Typically target models assume the Markov property, which is appropriate for kinematic motion on short time scales, however may not be adequate when considering a target's behaviour on longer time scales. In applications where targets are tracked on longer time scales through road networks, or camera networks [19], a two-scale approach may be used. On the first scale, a Markovian model may specify, in a statistical sense, "fast time" target behaviour according to the nature of the specific road the target traverses. A second scale "slow time" model may subsequently be used to characterise the global behaviour of the target as it traverses the network. Markov random walk models are generally not suitable in this case (e.g. [20], [21] [16]). It is argued that a reciprocal model, operating at this higher level of abstraction may be more appropriate, but less complex than a MDP model of agency. Simulation examples are presented which show that the additional model information contained in a reciprocal chain, measured in terms of Kullback-Leibler divergence, improves detection performance when compared to Markov models similar to those used in Kalman filtering.

The contributions made in chapters 2 and 3 can be summarised as follows,

Key contributions:

- The development of a track extraction algorithm by constructing a likelihood ratio test, with likelihoods obtained from recently developed normalised hidden reciprocal chain filters
- The systematic investigation of the extent to which the joint endpoints distribution of reciprocal chains affect tracking performance
- The construction of a novel simulation environment for numerical tests of the algorithms and claims described

1.4 Introduction to decision making with neural models

The field of machine learning has seen a great deal of success and focus in the last decade, and much of this owed to the impressive performance of neural networks [3]. Recast as 'deep learning', neural networks have achieved excellent results on traditional image recognition tasks, such as prediction of class labels, as well as seeing application to areas such as speech recognition and translation, and reinforcement learning problems, where they are used for function approximation as part of a model free reinforcement learning framework [22]. As argued in the previously, the use of generative models for certain information processing tasks is not always efficient nor necessary. Discriminative models, on the other hand, offer flexible and powerful alternatives, albeit with less "interpretability", though this point is debated by practitioners within the neural network field [22]. Neural networks are currently one of the most celebrated example of a discriminative model.

A neural network is composed of multiple processing layers, with each layer consisting of N "neurons" and an $N \times N$ matrix of "weights", with the number N varying from layer to layer, in general. Each neuron is some continuous non-linear function, taking as input a linear combination of the neurons in the previously layer, weighted by a column of the weight matrices. The weights are altered by a gradient descent algorithm, implemented efficiently as "backpropagation" using specialised hardware [3].

The uptake of neural networks in more applications and domains is in large part limited by the memory and computational power needs of these systems. As an example, one of the early image recognition neural networks which is credited with sparking the most recent interest in these models, has 61M parameters, totalling 249MB of memory, and and performs 1.5×10^9 high precision operations to classify one image [23]. Therefore, while neural networks perform well on expensive, GPU-based machines [3], there is clearly a practical need to reduce these memory and computational requirements.

The focus of this part of the thesis is the optimisation of low precision neural networks, specifically networks with binary neurons and weights. The problem of constraining the optimisation or "learning" process itself to consist of only low precision operations is not considered. However, this is a problem receiving considerable interest [24], [25], in no small part due to the promise of learning on-chip. With this said, restricting attention, within this thesis, to algorithms for training binary neural networks is still a difficult and worthwhile task, and yet still in its infancy, from a theoretical perspective.

Interestingly, the problem of learning the parameters of a neural network when its weights and neurons are constrained to be discrete is a unique problem in the sense that it has begun to see the two fields of machine learning and statistical physics meet. The machine learning community, motivated largely by the potential applications, has an interest in developing new algorithms for training large low precision neural networks that maintain the performance close to that of the continuous network counterparts. Statistical physicists, on the other hand, have studied the problem for the single-layer perceptron or logistic regression for decades. Applying sophisticated mathematical techniques they have produced fascinating phenomenological descriptions of algorithmic and learning processes, with applications emerging for both machine learning and neuroscience. Both communities bring important tools and insights to bear on the problem, and in considering both and attempting to place the various approaches in a unified frame will bear fruit, in terms of new algorithms and theory.

An important example of the two communities meeting in the study of neural networks, and of direct relevance to the work presented in this thesis, is the theory of neural network

initialisation. The current, impressive performance of standard continuous neural networks was for many years limited by poor initialisation of the continuous network weights [26], [27]. The theory behind popular initialisation schemes [27] was developed only recently in a series of papers [26], [28], [29]. This work revealed that poor initialisation limits the trainable depth of the network, and thus their expressive power [28], and also the speed with which they could be trained [30]. The contributions of these authors, both analytic and experimental, have been guided by work from the statistical physics community, relying on what is known as dynamic mean field theory [31]. The results have been extended and refined, and are considered a cornerstone of the theory for gradient-based neural network optimisation algorithms.

In the case of neural networks with binary variables, it is not obvious how to apply a gradient based optimisation algorithm to minimise a cost function involving discontinuous neurons and weights. A Bayesian approach, on the other hand, has no difficulty in handling discrete variables, generally speaking. However, to date, even approximate Bayesian "message passing" algorithms do not scale to large neural networks and datasets [32].

The most successful approaches for deep binary neural networks have opted to train discrete variable networks directly via backpropagation on a differentiable surrogate network, attaining excellent performance [33], [23]. A key to this approach is in defining an appropriate surrogate network as an approximation to the discrete model, and various algorithms have been proposed. The algorithms under focus in chapters 4 through 8 are those which consider *stochastic* binary neural networks, leveraging the stochastic nature of the neurons and weights to "smooth out" the discontinuities, in a more principled manner than more heuristic alternatives [23].

1.4.1 Key questions and contributions

The contributions of chapters 4 through 8, concerning the application of gradient based algorithms to optimising binary neural networks, are motivated by the following key questions,

Key questions:

- What are the relationships between the various binary neural network algorithms in the literature?
- What are the relationships between Bayesian approaches to the binary neural network problem and the non-Bayesian approaches to learning *stochastic* binary neural networks?
- Is initialisation an important aspect for these binary neural network algorithms (more specifically, the surrogate networks), given that this is a crucial element for training standard continuous networks? If so, how should one initialise these algorithms?
- What are the relationships between binary and continuous neural networks, optimised for a given learning problem?

The questions above arise from what is arguably a lack of coherence in the literature, coincident with little theoretical study of the algorithms currently being deployed to train binary neural networks. Chapters 4 through 8 make important contributions to the literature on binary neural networks, by beginning to frame the learning problem in a way that unifies much of the current, disparate literature. From this more coherent framework, new algorithms are introduced, and a theoretical analysis of several algorithms, at initialisation, is presented.

The work of unifying the framework for the binary neural network learning problem begins in chapters 4 and 5, by outlining the various choices involved in the construction of the algorithms. This includes for example the choices of binary neural network model (eg. choosing binary neurons or weights, or both), the method used to adapt its parameters (such as choosing an optimisation objective function), the approximations that render the objective function differentiable, and by extension define the continuous surrogate network. In the process, a Markov chain representation of stochastic binary neural networks is developed. From this basis it is possible to re-derive both existing surrogate networks more cleanly, and introduce new surrogates. The various heuristics borrowed from standard continuous neural networks are also discussed, since these may improve the performance of the continuous surrogates.

Chapter 6 makes another contribution toward a more unified framework for learning binary neural networks by arguing, as many have previously, that statistical physics can assist in studying algorithms, whether explicitly Bayesian or gradient based. Over the last two centuries, statistical physics has been an exploratory scientific discipline with a history of interacting experimental and theoretical progress. From this rich history, one can find suggestions for the relevant questions to ask (eg. which quantities are of interest), and how one might go about answering them (eg. how to calculate the various quantities). Furthermore, from this point it is possible to run experiments that measure the suggested quantities of interest and thus empirically test the corresponding theory developed.

The ideas borrowed from statistical physics are compelling, but are also of course an arbitrary choice from which to "unify" the various algorithms and methods. Other researchers may of course prefer points of view from different disciplines. For this reason, where possible connections are highlighted that have been made between the ideas in play here and those in the fields of statistical estimation, continuous optimisation theory and differential geometry.

Chapters 7 and 8, elucidate the important role of parameter initialisation in the surrogate networks, by extending the dynamic mean field formalism [31] to several of the binary neural network algorithms presented. The principal analytic results are the derivation of sets of coupled scalar equations describing how input signals propagate through a given surrogate network. From these equations depth scales are derived that limit the maximum trainable depth of the networks. For some models, these depth scales diverge under so called critical initialisation, while for other models it is proven that there is no divergence. Moreover, it is predicted theoretically and confirmed numerically, whether a parameter initialising scheme will successfully attain good training performance. This includes the schemes used in standard continuous networks and simply applied to the mean values of the stochastic binary weights.

These contributions are important as they begin to fill two gaps in the literature on neural networks. The first directly addresses the currently limited understanding of binary network algorithms, in terms of the conditions under which they should be expected to perform well, and in comparison with one another. To date, the initialisation of binary neural network algorithms have not been studied. The work here is important for understanding and developing successful binary neural network algorithms. The second gap that the contributions here begin to address is the lack of understanding of the relationships between binary and continuous weight neural networks, as well as their stochastic counterparts.

In conclusion, the key contributions of chapters 4 through 8 can be summarised as follows,

Key contributions:

- The presentation of a unified framework for learning binary neural networks, including optimisation objective functions and principled approximations therein
- The development of new Markov chain representations for stochastic binary networks
- The development of new gradient based algorithms
- The theoretical study of the developed algorithms at initialisation, based on the Gaussian central limit theorem

1.5 Outline of chapters

Chapter 2 begins with a review and general discussion of Markov and reciprocal chains for meta-level tracking. This includes both the relevant background theory for reciprocal chains, and new contributions dealing with the fixed interval nature of these models, and proposing alternatives. Specifically, new infinite horizon models are derived, and compared with the Markov decision process models of agency.

Chapter 3 presents the main contributions of the first part of the thesis, as published in [34], relating to the problem of track extraction. The final concluding remarks and a discussion of future prospects, are deferred until chapter 9.

Chapter 4 first describes the problem of statistical learning, which is the objective for any learning system. This is followed by a summary of the essential elements to standard neural networks used in the classification setting. Specific mention is made of the particular issues that motivate second order optimisation methods and careful parameter initialisation.

Chapter 5 presents the problem of learning multi-layer *stochastic* binary neural networks, outlining the optimisation problem which includes the gradient approximation and estimation that effectively yields the various continuous surrogate models. Existing and new surrogate models are presented, and discuss some qualitative similarities and differences, thus providing a more unified view of the literature on direct optimisation of binary neural networks.

Chapter 6 introduces the basic concepts of statistical physics, many of which underpin the technical contributions. The examples discussed in this chapter also serve to review the recent theoretical literature concerning the binary perceptron.

Chapter 7 begins with a derivation of the coupled scalar equations for a class of deterministic surrogate network presented in chapter 5. Numerical simulations and experimental results are presented which confirm the predictions of the mean field theory. The focus is first on surrogates for networks with stochastic binary weights and neurons. Also investigated is the role of the underlying stochasticity of the neuron slope, and the form of the non-linearity, by finding numerically the 'edges of chaos' where in the hyper-parameter space. This relates directly to the choice of binary neuron noise model (in its latent variable interpretation). This chapter also presents the results for cases of continuous weights.

Chapter 8 presents the corresponding analysis for the Monte Carlo based surrogate, which is referred to as the 'perturbed' surrogate. This chapter also presents the analogous dynamic mean field theory for deterministic and stochastic binary networks, results that may provide insight into the relationship between the surrogate and binary networks.

The thesis is concluded in chapter 9. The conclusion presents a summary of results of the thesis, as well as a discussion of avenues for future research.

Chapter 2

Stochastic processes for meta-level tracking

This chapter presents a review and discussion of stochastic processes that, when interpreted as models of target dynamics, correspond to targets which exhibit "destination awareness". This notion, which one can consider to be the progress of a target toward a terminal state, from some initial state, is an example of the behaviour a meta-level tracker may incorporate into its target model.

Conditional Markov processes are first discussed, followed by a description of the construction of reciprocal chains, both of which are stochastic processes defined on a fixed time interval. Following this new infinite-horizon, time-invariant Markov chains are presented, motivated by the idea of a model of destination awareness without the fixed time interval constraint. This contribution, which is exploratory in nature, points to interesting connections to the stochastic optimal control literature, specifically of a class of problems studied under the title of path integral control or Kullback-Leibler control theory.

2.1 Destination awareness and conditional processes

There have been a range of approaches to incorporating available *a priori* information about terminal states within the class of Markov models, both at the estimation stage and in the target dynamics model itself. In both approaches, the methods amount to "back propagating" the influence of a future state value or distribution by appropriate conditioning.

For example in [35], [36], *a priori* destination information is incorporated via a corrective term into the state update of continuous-time Gauss-Markov processes. The well known Markov bridge [37] incorporates information about a single fixed destination (such as a scheduled stop). In [37] and [38] this was generalised to specifying an *a priori distribution* for a future time of the state process (such as the notion that vehicles enter and leave the field of view at its borders), creating a new Markovian process referred to as a Schrödinger bridge. From a modelling point of view, source-destination awareness is taken to mean that the initial and final target states may have an arbitrary *joint* probability distribution and that target dynamics are therefore anticipative, reflecting the intention of the target to move towards its destination. The specified probabilistic dependence between future and past states corresponds to a class of models known as *reciprocal processes*.

Reciprocal processes were studied in detail by [37] in a general setting, and subsequently by

[39] who considered the realisation and state estimation problem for the Gaussian discrete index parameter case. There are many other related works which are summarised in [40]. Reciprocal chains (RCs) are finite state RP, and hidden reciprocal chains (HRCs) are stochastic processes generated from an RC via some noisy and/or incomplete observation mechanism, analogous to hidden Markov chains (HMCs). These observations may have continuous or finite states, also like HMCs. Finite state models are preferred for the usual reasons such as the ability to easily incorporate state space constraints (obviously important in the ground target tracking example) and abstract state attributes. Naturally, one needs to be aware of the potential for significant computational complexity likely to arise in estimation algorithms derived from these models.

A brief recount the development of inference of HRCs is as follows; un-normalised optimal filters/smoothers for HRC were derived using a Bayesian approach in [40]. Normalised filters and smoothers were developed in [41], [42], which also considered incorporating "waypoints" on the target trajectory. Maximum likelihood estimation of state sequences for HRC was presented in [43]. In [17] destination aware tracking based on HRC was first proposed, but the RC models tested, being pinned to a single destination, remained in the Markov class.

2.2 Reciprocal Models

This section provides definitions for Markov and Schrödinger bridges, reciprocal chains, as well as providing a concise summary of the Markov bridge construction of a RC as outlined in [40]. The cases when a reciprocal chain satisfies the Markov are described, and the concept of a reciprocal class is introduced in a natural way, by considering the Kullback-Leibler divergence between processes on an interval.

2.2.1 Conditioning a Markov Chain on the Future

A Markov bridge (MB) is formed by taking a reference Markov process Z_t, $t = 0, \ldots, T$, for some fixed $T \geq 2$, and conditioning it on taking a fixed value for Z_T. A rigorous description of this notion, in continuous time, is provided in [44]. For clarity of exposition, assume the reference process has time homogeneous transitions. Specifically, let A denote the time homogeneous transition probability matrix for the reference Markov chain (MC) Z_t, taking values on a finite state space $\mathcal{S} = \{1, \ldots, N\}$, where $N \geq 2$, and let the entries of the matrix be $A_{i,j} = \mathbf{P}(Z_{t+1} = j | Z_t = i)$, which are all assumed to be strictly positive. This assumption may be relaxed by considering questions of state reachability for MB, an issue which is beyond the scope of this thesis. The MB transition probabilities are constructed by applying Bayes rule as shown in [40],

$$\begin{aligned} B^k_{i,j}(t) &= \mathbf{P}(Z_{t+1} = j | Z_t = i, Z_T = k) \\ &= \frac{A_{i,j} \left(A^{T-(t+1)}\right)_{j,k}}{\left(A^{T-t}\right)_{i,k}} , \end{aligned} \tag{2.1}$$

for $t = 0, \ldots, T-2$. The Schrödinger bridge (SB) is the time-inhomogeneous Markov process that attains a specified marginal distribution at its initial and final times , and which has transition probabilities, i.e. 'dynamics', closest, in some sense, to the specified *a priori* dynamics of the reference process (see [45] for a finite-state proof). The "measure" of closeness used is the Kullback-Leibler (KL) divergence [46], defined for discrete distributions $P(X = x)$ and

$Q(X = x)$, with the same (finite) sample space $x \in \mathcal{S}$ is defined as,

$$KL[P(X)\|Q(X)] = \sum_{x\in\mathcal{S}} P(X = x) \log \frac{P(X = x)}{Q(X = x)} \tag{2.2}$$

The KL divergence is not of course a distance in the technical sense, since it is symmetric. Note that KL divergence is finite under certain conditions. For example, one condition is that the support of $P(X)$ is contained within the support of $Q(X)$, see [46] for details. Other conditions pertain to the behaviour of the moments of the distributions, relating to heavy tail phenomena of the densities. This is not an issue in the discrete spaces considered here, where the condition on the support of $P(X)$ is sufficient to guarantee finite divergence.

The following construction of the SB originates from Schrödinger's idea of minimising the KL divergence, although not set out explicitly in these terms, since his work pre-dates Kullback and Liebler's paper by 20 years, as well as being contemporary to Kolmogorov's work on probability (see [45] for a relevant discussion). The original idea was formalised by [37], and recently specialised for the discrete-state setting in [45]. With A as before, let X_t be the Schrödinger bridge of Z_t with marginal distributions (row vectors) π_0 and π_T on X_0 and X_T respectively. Let ψ_0, ψ_T be the N dimensional positive row vectors that are solutions of the following coupled equations

$$\pi_T = \psi_T \circ \psi_0 \, A^T, \qquad \pi_0 = \psi_0 \circ \psi_T \, \left(A'\right)^T \tag{2.3}$$

where $\circ$ is the element-wise product and A' denotes the transpose of A. Existence and uniqueness of solutions is proven in [45]. Define the positive row vectors $\psi_t = \psi_T \, (A')^{(T-t)}$, then the transition probabilities of the SB are given by

$$S_{i,j}(t) = \mathbf{P}(X_{t+1} = j | X_t = i) = A_{i,j} \frac{\psi_{t+1}(j)}{\psi_t(i)} \ . \tag{2.4}$$

2.2.2 Reciprocal Chains

A reciprocal process is a generalisation of the SB allowing any source-destination relationship, as described in [37]. It can also be derived from a reference Markov process by fixing the start and end points of the reference process and allowing them to vary according to an arbitrary joint distribution. The new process generated by this method is generally not Markov, however all Markov processes are reciprocal [37]. Formally, for a random process $\{X_t\}$ indexed by $t \in \{0, 1, \ldots, T\}$ for some fixed integer $T \geq 2$ the process $\{X_t\}$ is said to be *reciprocal* [37], if

$$\mathbf{P}(X_t | X_s, \forall\, s \neq t) = \mathbf{P}(X_t | X_{t-1}, X_{t+1}) \ , \tag{2.5}$$

for each $t = 1, \ldots, T-1$. Thus X_t is conditionally independent of $X_0, \ldots, X_{t-2}, X_{t+2}, \ldots, X_T$ given its neighbours X_{t-1} and X_{t+1}. The reciprocal model is specified by the set of three-point transition functions (2.5) together with the given joint distribution on the end points $\mathbf{P}(X_0, X_T)$. Denote the three-point transition functions in (2.5) by

$$Q_{i,j,k}(t) = \mathbf{P}(X_t = j | X_{t-1} = i, X_{t+1} = k), \tag{2.6}$$

for $i, j, k \in \mathcal{S}$, $t = 1, \ldots, T-1$. As in the Markov bridge case (2.1), the three-point transition functions of a RC derived from a reference Markov chain are given via Bayes rule,

$$Q_{i,j,k}(t) = \frac{A_{i,j} \, A_{j,k}}{\sum_{\ell=1}^{N} A_{i,\ell} \, A_{\ell,k}} \ ,$$

and the end-points distribution is denoted by

$$\Pi_{i,j} = \mathbf{P}(X_0 = i, X_T = j), \quad i, j \in \mathcal{S} \ . \tag{2.7}$$

Pinning the end point of a RC generates a Markov bridge, a property that allows for the causal representation and thus processing of RC [40]. Considering the joint distribution of the states of a RC and using direct Bayes' conditioning and (2.5), the relevance of Markov bridges becomes apparent,

$$\mathbf{P}(X_0, ..., X_T) = \mathbf{P}(X_0, X_T) \prod_{t=1}^{T-1} \mathbf{P}(X_t | X_{t-1}, X_T). \tag{2.8}$$

The terms contained within the product in (2.8) are the state transitions for a MB pinned at X_T. Thus any RC may be viewed as a mixture over a set of N MBs[1]. More precisely, any RC is uniquely specified by the finite set of MBs with probability transition matrices given by (2.1) and initial distributions π_i^k for each final state k, given by the conditional distribution

$$\pi_i^k(0) = \mathbf{P}(X_0 = i | X_T = k) \ = \ \frac{\Pi_{i,k}}{\sum_{j=1}^N \Pi_{j,k}} \ . \tag{2.9}$$

The process for generating a sample path of this RC is to draw the initial and final points X_0 and X_T from Π, which specifies the MB transitions corresponding to $X_T = k$. The sample path is then constructed in the standard way for a MC, starting from X_0 using the transitions (2.1).

An alternative formulation for a RC can be posed in terms of a finite set of N Schrödinger bridges by pinning the initial state rather than the final state. The Schrödinger bridge corresponding to a particular state $(X_0 = i)$ can be constructed by specifiying the final distribution to be attained,

$$\pi_k^i(T) = \mathbf{P}(X_T = k | X_0 = i) \ = \ \frac{\Pi_{i,k}}{\sum_{j=1}^N \Pi_{i,j}} \ . \tag{2.10}$$

Sample paths can also be generated using this formulation, identically as for MBs [2]. This alternative formulation of a RC as a mixture of Schrödinger bridges highlights that a reciprocal target's dynamics depend not only on its current state, but also on its final and initial states. This idea is discussed at length in [48], where the author considers the "forgetting" properties of reciprocal chains as a function of the time between the initial and terminal states, using appropriately defined mathematical tools.

A RC on a finite interval remains Markov when its joint endpoints distribution (2.7) factorises according to,

$$\Pi = \mathrm{diag}(\lambda_T) \ A^T \ \mathrm{diag}(\lambda_0). \tag{2.11}$$

This result was established in full generality in [37] and includes the Markov bridge as a special case. An example of a non-Markov RC model is that of a target that returns to its origin by

[1] It is also possible to consider the compound process $\{X_t, X_T\}$ of a RC, which is Markov [47].

[2] However it requires solving the system of non-linear equations (2.3) in order to determine the transition probabilities, thus only the MB approach is used

time T, which is referred to in this thesis as a *loitering* RC. This can be modelled via the joint end-points distribution

$$\Pi_{LRC} = \mathbf{P}(X_0 = i, X_T = j) = \begin{cases} p_r(i) & i = j \\ 0 & i \neq j. \end{cases} \tag{2.12}$$

where $p_r(i)$ is the probability of starting and returning to a particular state, and $\sum_{i=1}^{N} p_r(i) = 1$.

It is well known in the literature on Schrödinger bridges [45], [38] that the KL divergence between any two processes with the same three-point dynamics reduces to the KL divergence between the joint endpoints distributions. Explicitly, for two distributions $\mathbf{P}_1$ and $\mathbf{P}_2$ defined over the set of all possible trajectories through the statespace, $\mathcal{X} = \{X_0, \ldots, X_T\}$,

$$KL[\mathbf{P}_1(\mathcal{X}) \| \mathbf{P}_2(\mathcal{X})] = KL[\mathbf{P}_1(X_0, X_T) \| \mathbf{P}_2(X_0, X_T)] \tag{2.13}$$

This result can be established using (2.8), and holds for non-Markov processes. Since the endpoints distribution encodes source-destination awareness, one can evaluate the difference between models for target dynamics by evaluating the KL divergence between their endpoint distributions (2.13). This will be used in Section 3.4.

2.3 Models on infinite horizons

A reasonable criticism of reciprocal chain models for tracking is that they require a pre-determined fixed time interval, or fixed number of events such as a target sighting, in order to model a target's behaviour. In modelling an intention to proceed to a destination, this fixed time appears unrealistic, since different individuals with different "velocities" (or local dynamics) may still have the same destination, despite arriving at different times.

A Markov decision process (MDP) is commonly defined on an infinite horizon. Furthermore, a terminal or destination state can be encoded for the agent whose dynamics the MDP describes. This can be achieved if an agent's cost function penalises the agent for every time step spent away from the terminal state. The resulting dynamics, formed by an agent behaving according to an "optimal policy" [4], will typically mean that, for suitable state spaces and dynamics, most agents will reach their destination at an average time, assuming some dispersion if the environment dynamics are stochastic.

This sort of model is more appealing from the point of view of modelling human behaviour since it appears more realistic for the aforementioned reasons. However, in order to avoid the "full blown" MDP model of agency, there is an alternative for the infinite horizon case. It is possible to derive, from a Markov bridge or reciprocal chain, time-homogeneous Markov approximations. The following section presents derivations of these approximations, and following this briefly present some simulations to compare the models.

2.3.1 Markov Approximations to RC Models

This section presents the exact analytical expressions for the Markov chains which best approximate a non-Markov RC, in terms of minimising the Kullback-Leibler divergence between the joint probability distributions of the processes. Henceforth when referring to the 'closest' Markov approximation, it is meant in this sense.

The first result is an expression for the closest time-inhomogeneous MC approximation to an RC, meaning the transition probabilities are time dependent. While this does not change

the fixed time interval nature of the process, this derivation can be used to find the closest time-homogeneous MC approximation to an RC. Specifically, this homogeneous process can be seen as the time average of the closest in homogeneous MC.

Time-inhomogeneous MC Approximation

Now consider the KL divergence between an RC with joint distribution $p(x_0, \dots x_T)$ and a stationary MC with joint distribution $q(x_0, \dots x_T)$.

$$\begin{aligned}\mathcal{D}(p\|q) &= \sum_{\mathcal{X}} p(\mathcal{X}) \log \frac{p(\mathcal{X})}{q(\mathcal{X})} \\ &= \sum_{\mathcal{X}} p(\mathcal{X}) \log p(\mathcal{X}) - \sum_{\mathcal{X}} p(\mathcal{X}) \log q(\mathcal{X})\end{aligned}$$

The first term is simply the entropy of the RC as before, so all that remains is to calculate the cross entropy term. While the MB characterisation can also be used for a MC, one can simply use the standard chain rule for Markov chains. Thus one obtains, for the cross entropy term,

$$\begin{aligned}H_{\mathcal{X}}^{p,q} &= -\sum_{\mathcal{X}} p(\mathcal{X}) \log q(\mathcal{X}) \\ &= -\sum_{x_0, x_T} p(x_0, x_T) \log q(x_0) \\ &+ \sum_{x_0, x_T} p(x_0, x_T) \Bigg(\sum_{x_1} p(x_1|x_0, x_T) \log q(x_1|x_0) \\ &+ \sum_{t=1}^{T-2} \sum_{x_t} p(x_t|x_0, x_T) \sum_{x_{t+1}} p(x_{t+1}|x_t, x_T) \log q(x_{t+1}|x_t) \\ &+ \sum_{x_{T-1}} p(x_{T-1}|x_0, x_T) \log q(x_T|x_{T-1}) \Bigg)\end{aligned} \tag{2.14}$$

The next step is to formulate the Lagrangian to minimise this divergence with respect to the transitions of $q(x_0, \dots, x_t)$, which are denoted as $M_{ij}(t)$. The constraint is simply that the $M_{ij}(t) = q(x_{t+1} = j|x_t = i)$ be valid Markov transitions, that is, that the probabilities are non-negative and sum to one.

$$\mathcal{L} = \mathcal{D}(p\|q) + \sum_{m,t} \lambda_{m,t} (\sum_{n=1} M_{m,n}(t) - 1).$$

Then

$$\frac{\partial \mathcal{L}}{\partial M_{ij}(t)} = \frac{\partial H_{\mathcal{X}}^{p,q}}{\partial M_{ij}(t)} + \lambda_{i,t}$$

From (2.14)

$$\frac{\partial \log q(x_{t+1} = j|x_t = i)}{\partial M_{ij}(t)} = \frac{\partial \log M_{ij}(t)}{\partial M_{ij}(t)} = \frac{1}{M_{ij}(t)}$$

Then, for $0 < t < T-1$, the cross entropy term in the minimisation step becomes

$$\begin{aligned}\frac{\partial H_{\mathcal{X}}^{p,q}}{\partial M_{ij}(t)} =& -\sum_{x_0,x_T} p(x_0,x_T)p(x_t^i|x_0,x_T)p(x_{t+1}^j|x_t^i,x_T)\frac{1}{M_{ij}(t)} \\ =& -\frac{1}{M_{ij}(t)}\sum_{x_T} p(x_{t+1}^j|x_t^i,x_T)\sum_{x_0} p(x_t^i,x_0,x_T) \\ =& -\frac{1}{M_{ij}(t)}\sum_{x_T} p(x_{t+1}^j|x_t^i,x_T)p(x_t^i,x_T) \\ =& -\frac{1}{M_{ij}(t)}p(x_{t+1}^j|x_t^i)p(x_t^i)\end{aligned}$$

The notation x_t^i is used to denote $x_t = i$ for shorthand. The same expressions for $M_{ij}(t)$ is obtained at $t = 0$ and $t = T-1$. Setting the partial derivatives of the Lagrangian to zero,

$$M_{ij}(t) = \frac{p(x_{t+1} = j|x_t = i)p(x_t = i)}{\lambda_{i,t}}$$

and by enforcing the constraint,

$$\lambda_{i,t} = p(x_t = i)$$

Therefore it can be shown that,

$$M_{ij}(t) = p(x_{t+1} = j|x_t = i), \tag{2.15}$$

which are the two point transitions of the RC, a result which makes intuitive sense. In the special case of a RC constructed with an endpoint distribution resulting in the process being Markov, then the derived expression will reduce to the underlying Markov transitions, meaning the divergence will be equal to zero. A relevant discussion of the relationship between reciprocal and Markov processes, in terms of a comparisons of the influence of endpoints distribution on the dynamics is presented in [48].

Time-homogeneous Markov Chain Approximation

Consider now the case of finding a homogeneous approximation to any RC. It is possible to extend the analysis from above as follows. In this case $M_{ij}(t) = A_{ij}$, $\forall\, t$, and with corresponding constraints on A_{ij}, the Lagrangian is now

$$\mathcal{L} = \mathcal{D}(p\|q) + \sum_m \lambda_m(\sum_{n=1} A_{m,n} - 1)$$

Taking again partial derivatives, and noting that once more the minimisation is only over the cross entropy terms,

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial A_{ij}} &= \frac{\partial \mathcal{D}(p\|q)}{\partial A_{ij}} + \frac{\partial \sum_m \lambda_m (\sum_{n=1} A_{m,n} - 1)}{\partial A_{ij}} \\
&= -\sum_{x_0, x_T} p(x_0, x_T) \sum_{t=1}^{T-1} p(x_t^i | x_0, x_T) p(x_{t+1}^j | x_t^i, x_T) \frac{1}{A_{ij}} \\
&\quad + \sum_{x_T} p(x_0^i, x_T) p(x_1^j | x_0^i, x_T) \frac{1}{A_{ij}} + \lambda_i \\
&= -\frac{1}{A_{ij}} \sum_{t=1}^{T-1} p(x_{t+1}^j | x_t^i) p(x_t^i) + p(x_1^j | x_0^i) p(x_0^i) + \lambda_i \\
&= -\frac{1}{A_{ij}} \sum_{t=0}^{T-1} p(x_{t+1}^j | x_t^i) p(x_t^i) + \lambda_i
\end{aligned}$$

Again setting the partial derivatives of the Lagrangian to zero, it follows that

$$A_{ij} = \frac{\sum_{t=0}^{T-1} p(x_{t+1} = j | x_t = i) p(x_t = i)}{\lambda_i},$$

and by evaluating the constraint,

$$\lambda_i = \sum_{t=0}^{T-1} p(x_t = i),$$

It is clear that the Lagrange multipliers λ_i are a time-average of the marginals of state i. Explicitly,

$$A_{ij} = \sum_{t=0}^{T-1} p(x_{t+1} = j | x_t = i) \frac{p(x_t = i)}{\sum_{s=0}^{T-1} p(x_s = i)} \tag{2.16}$$

This result builds nicely on the previous result, since the homogeneous transitions probabilities are clearly a time average of the inhomogeneous transition probabilities $p(x_t + 1 = j | x_t = i)$, weighted by a time average of the probability of being in state $x_t = i$.

2.4 Numerical examples

This section presents a short numerical investigation whose purpose is to advance the qualitative discussion of infinite-horizon alternatives to reciprocal chain models, which nonetheless retain an intention proceed to a destination. In order to first get a better sense of the dynamics of the Markov bridge as compared to the homogeneous Markov chain, Figure 2.1 present several sample paths in a one dimensional statespace. for a state space with $N = 10$ states, the Markov bridge is defined to terminate in state $S = 10$ in time $T = 20$ steps, assuming some base dynamics. The time-homogeneous Markov chain, with dynamics as derived above, observes some dispersion in its 'hitting times' of state $S = 10$, though clearly the dynamics have a 'drift' toward this state.

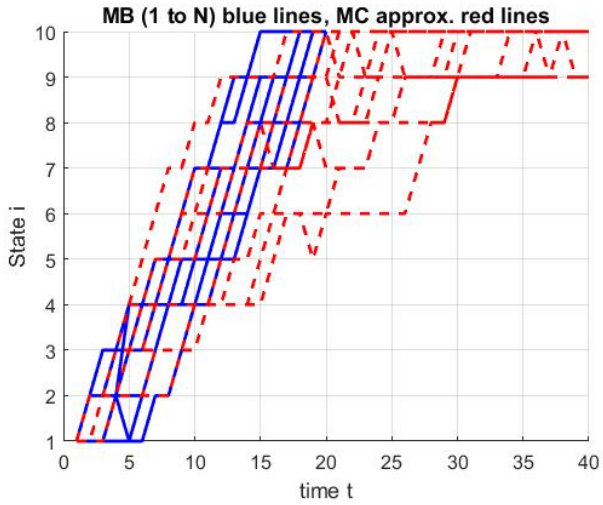


Figure 2.1: *Sample paths*: Markov bridge model (blue lines). Time-homogeneous Markov chain.

It is interesting to compare a time-homogeneous Markov chain constructed in this way to the resulting process from a Markov decision process acting under an optimal policy [4]. In Figure 2.2 the optimal policy for a two dimensional "grid-world" environment is shown, where the cost is the number of steps taken before reaching the goal state. The illustrated policy is optimal in a deterministic (noiseless) environment, or in a stochastic environment with isotropic noise in the dynamics between states. In the case of a the Markov transition matrix for a one dimensional state space (ie. a walk along the natural numbers), both the stochastic transition matrices for the MDP and time-homogeneous Markov chain are presented in Figure 2.3.

A sensible question is to ask what kind of tracking performance such a Markov model would achieve for a non-Markovian target, such as a target modelled by a reciprocal model. An in depth numerical study is warranted, guided once again by considering the KL divergence between oath distributions, taking into account the lack of a fixed time interval in the time-homogeneous Markov chain.

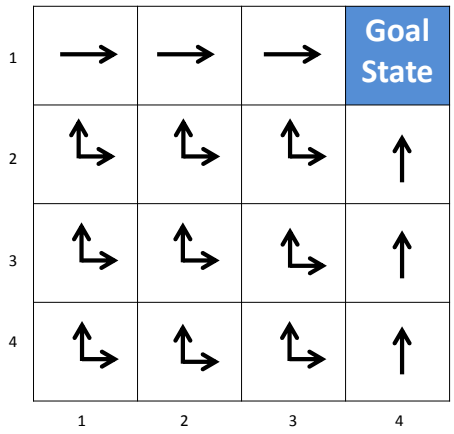


Figure 2.2: *Gridworld solution*: An example of an optimal policy in the "grid world" toy model.

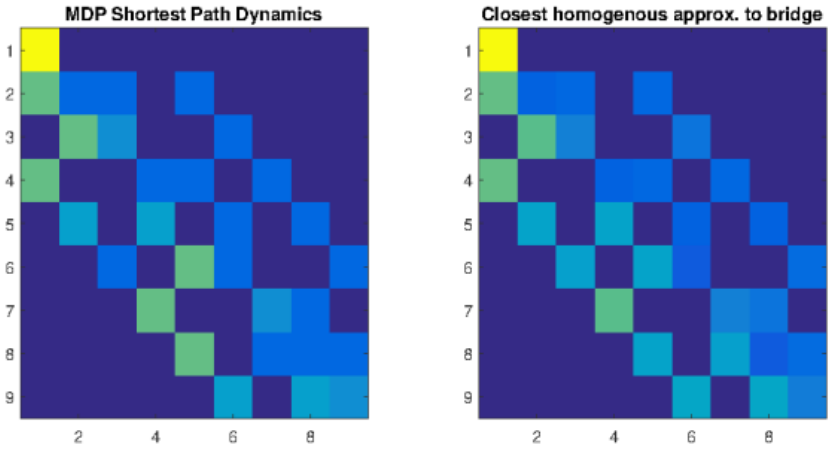


Figure 2.3: *Stochastic transition matrices*: MDP (left), time-homogeneous Markov chain (right). The transition probabilities are qualitatively very similar.

2.5 Chapter conclusion

This chapter considered stochastic processes, both Markov and reciprocal, that correspond to targets which exhibit "destination awareness". The discussion began with the Markov and Schrödinger bridge models, from which reciprocal chains may be constructed. Following this new infinite-horizon (or time-invariant) Markov chains were presented, motivated by the idea of a model of destination awareness without the fixed time interval constraint. The developments in this second section point to similarities, on a qualitative level, between reciprocal processes generally and Markov decision processes. In certain special cases, the relationships can be made exact.

The connection between Schrödinger bridges and stochastic optimal control problems is well established [49]. The original Schrödinger problem, in continuous time and state, ie. a diffussion process, was formally related to a stochastic optimal control problem with quadratic control costs and a final time constraint. The optimal control $u^*(x,t)$ for this problem can be related to the positive function defined earlier, $\psi(x,t)$ adapted to the continuous time and state case, as shown in [49]. Equivalence of this problem to the minimum Kullback-Liebler divergence over distributions over paths are also illuminated. This point formally links the Schrödinger problem to wider classes of control problems, known variously as path integral control or Kullback-Liebler control problems [50],[51], [52].

Chapter 3

Track Extraction

This chapter develops track extraction algorithms for tracking with reciprocal chains. Track extraction (TE) [53], [54] is a statistical hypothesis test used to determine whether some subset of a given set of sensor measurements originate from a moving target, as opposed to arising from false-alarm (clutter) returns. TE is a useful procedure when individual clutter and target returns are otherwise indistinct. More specifically, whereas standard multiple hypothesis tracking algorithms confirm or delete individual candidate tracks based on their individual likelihood [55], TE algorithms use the entire set of observations to decide whether a target exists by considering, in a Bayesian sense, all possible trajectories consistent with a given statistical model of the target dynamics. If a target is detected, state estimates (e.g. location) can then be formed, using the *a posteriori* probabilities associated with the Bayesian filter (or smoother).

TE is therefore also a detection process, with its own quantities such as detection and false alarm probabilities, but it works on a higher level of abstraction than a detector applied to each sensor measurement, because it considers the likelihoods of target *trajectories* rather than individual point detections. Thus TE is an example of a meta-level tracker function as discussed in the introduction, in particular the overview of tracking systems in subsection 1.2.

The contributions made in this chapter are motivated in particular by answering the questions of whether and to what extent modelling the source-destination awareness of a target can improve track extraction, as a function of the parameters of a HRC model. Particular use is made of the Kullback-Leibler (KL) divergence between stochastic processes which are defined to belong to the same *reciprocal class*, that is, Markov and non-Markov processes with the same reciprocal (acausal) dynamics [38]. The KL divergence of two such models reduces to the divergence of the models' endpoints distributions. This result guides the numerical simulations and the descriptions of the potential benefits of HRC for tracking. As with most finite data, non-linear inference problems, it is infeasible to obtain *a priori* performance metrics, so instead numerical simulations are used to study the performance of the new algorithms proposed here.

Since the aim is to capture more complex target behaviour than simple Newtonian motion models, it makes sense to formulate a simulation setting corresponding to domains where target intent is relevant. Thus a contribution of this chapter is to introduce an observation model for the discrete space that includes "clutter", meaning observations of uncertain origin. This framework is significant since it is a logical step before considering multi-target tracking. In summary, this chapter makes the following contributions:

- Develops a track extraction algorithm by constructing a likelihood ratio test, with likelihoods obtained from recently developed normalised hidden reciprocal chain filters

- Investigates systematically the extent to which the joint endpoints distribution of reciprocal chains affect tracking performance
- Constructs a novel simulation environment for numerical tests of the algorithms and claims described

3.1 Track Extraction for Reciprocal Chains

The following two examples further motivate the idea of considering entire track trajectories. The first is of tracking a target through a camera network, and the second is of tracking a vehicle through a road network. Consider a target moving through a camera network, assuming it has been identified by a low level object detectors as being in a particular camera's view. For the task of tracking this target through the network, certain schemes attempt to build a 3D model of the world and then tracking the object with a 3D Kalman filter [56]. Other methods simply use a 2D motion model in the image plane and based on camera topology assist the object detector in matching targets. These methods tend to work best in camera networks with overlapping cameras. In networks with only some or no overlapping cameras, solving inter-camera tracking has seen focus from probabilistic or statistical methods. These methods are underscored by the principle that by accumulating evidence of movement patterns one can learn and use the camera network topology in some way [57], [58], [59]. In [58] the *acitivity topology* is estimated, which essentially considers the chance of moving between any two cameras. In [57] the movement of targets within and between cameras is modelled with a Markov chain, and a stochastic transition matrix is trained from a series of observations. The hidden reciprocal chain (HRC) model can be thought of as an extension to this Markov chain motion model, altering the dynamics to incorporate initial and destination states for various trajectories.

As a second example, consider a ground vehicle that is being tracked through a road network by a single airborne radar, which, after preprocessing, passes point estimates of uncertain origin to a tracker [60]. In reality, the target vehicle's motion is highly constrained by the nature of the road network, while still proceeding through the network with a typically pre-determined destination. In such an application, of tracking a vehicle on a road, a two-scale approach may be used, where a Markovian model specifies (in a statistical sense) "fast time" local vehicle dynamics, according to the nature of the specific road the target traverses. A second "slow time" model may be used to characterise the global behaviour of the target as it traverses the road network. Markov random walk models are generally not suitable in this case (e.g. [20], [21] [16]), and a model with a higher level of abstraction may be more appropriate. Note that for the camera network model, a two-scale approach may also be appropriate, especially in the case of a network with non-overlapping cameras.

Therefore the choice of a target dynamic model, or class of such models, is an important part of track extraction algorithm design, which necessarily operates at a higher level of abstraction, since it considers entire trajectories. The remainder of this chapter develops TE algorithms based on reciprocal models. As discussed, these models are generally non-Markovian and embody the idea of "source-destination awareness" (e.g. [16], [17], [18]) where an imposed statistical relationship between the target's original and final states can place higher weight on proceeding through the network in a specific manner.

The track extraction framework employed is similar to that in [53], with three key differences. Firstly, optimal Bayesian estimation is employed over a finite statespace, rather than a continuous space; secondly the process considered exists on a fixed interval; and thirdly, the likelihood

ratio test here has a single threshold, though it could be formulated as the sequential probability ratio test of [53] which has two thresholds. Track extraction deals with two types of uncertainty, false alarms - detections that originate from "clutter", and sensor noise which degrades position estimates. This could include the effect of quantisation of measurements associated with the finite states. In section V of [53], signal-strength information is used and improves the track extraction performance, but here any explicit notion of SNR and signal amplitude is removed, in order to emphasise the role of the model in differentiating clutter from target detections. Instead, the association between observation and the source of the detection is treated probabilistically, via an observation likelihood function and *a priori* probabilities over false alarms. An observation model representative of the track extraction regime is developed and tested. There is at most one target during the tracking interval. Multiple detections are allowed at each time, with at most one due to the target. A dynamic model of the target is constructed which also includes the global source-destination attributes. The likelihood associated with a given set of detections is evaluated using the class of normalised hidden reciprocal chain (HRC) filters. This likelihood is used to construct the track extraction algorithm based on a likelihood ratio test (LRT).

3.2 Estimation for Markov Process Models

This section reviews hidden Markov models (HMMs) and the associated optimal Bayesian estimation algorithms for filtering. This review is worthwhile since the hidden reciprocal chain filters are built in a special way, from a set of N distinct HMM filters. Only filtering and not offline smoothing is considered since the detectors derived for the track extraction algorithms require only filtering algorithms. Details on the smoothing problem can be located in [61], for example.

HMM State Estimation

The dynamics of a Markov Chain (MC), denoted X_t, are specified by its initial and transition probabilities,

$$\begin{aligned} \pi_i(0) &= \mathbf{P}\{X_0 = i\} \\ A_{i,j}(t) &= \mathbf{P}\{X_{t+1} = j | X_t = i\} \ , \end{aligned} \tag{3.1}$$

for $i, j = 1, \ldots, N$ and $t = 0, \ldots, T-1$. See [44] for a complete presentation of the details. The observation likelihoods satisfy the conditional independence property

$$\mathbf{P}\{Y_0, \ldots, Y_T | X_0, \ldots, X_T\} = \prod_{t=0}^{T} \mathbf{P}\{Y_t | X_t\} \ . \tag{3.2}$$

The evaluated (conditional) observation densities are denoted by $C_i(t) = \mathbf{P}\{Y_t | X_t = i\}$ for $t = 0, \ldots, T$ and $i = 1, \ldots, N$. These terms are also sometimes referred to as observation likelihoods.

Two fundamental problems are state estimation, where one seeks estimates of the state at a time t of the system of interest, based on the measurements; and parameter estimation, where one seeks to estimate the parameters which define the system. In this thesis knowledge of the parameters is assumed, whether the model is Markov or reciprocal. Parameter estimation is

a difficult but studied problem in the HMC literature, but as yet unexplored in the reciprocal literature.

State Estimation for HMCs - filtering

The optimal filtering problem involves determination of the *a posteriori* state probabilities,

$$\alpha_{t|t}(i) \;=\; \mathbf{P}\left\{X_t = i | Y_0, \ldots, Y_t\right\} \tag{3.3}$$

for each $t = 0, \ldots T$. Assuming one can compute the vector $\underline{\alpha}_{t|t}$ from the observations $Y_0, \ldots, Y_t$, then the information available at time t is $Y_0, \ldots, Y_t$, and $\underline{\alpha}_{0|0}, \ldots, \underline{\alpha}_{t|t}$.

Upon receiving the subsequent measurement, Y_{t+1}, the aim is to compute $\underline{\alpha}_{t+1|t+1}$ from the available information. The Markovian property of the state process makes it possible to write $\underline{\alpha}_{t+1|t+1}$ as a function of the previous estimate $\underline{\alpha}_{t|t}$ and the new measurement Y_{t+1} as follows. Consider (using Bayes' rule and the HMM properties),

$$\begin{aligned}
\alpha_{t+1|t+1}(i) &= \mathbf{P}\left\{X_{t+1} = i | Y_0, \ldots, Y_{t+1}\right\} \\
&= \frac{\mathbf{P}\left\{X_{t+1} = i, Y_0, \ldots, Y_{t+1}\right\}}{\mathbf{P}\left\{Y_0, \ldots, Y_{t+1}\right\}} \\
&= \frac{\mathbf{P}\left\{Y_{t+1} | X_{t+1} = i\right\}\ \mathbf{P}\left\{X_{t+1} = i, Y_0, \ldots, Y_t\right\}}{\mathbf{P}\left\{Y_0, \ldots, Y_{t+1}\right\}} \\
&= \frac{\mathbf{P}\left\{Y_{t+1} | X_{t+1} = i\right\}}{\mathbf{P}\left\{Y_0, \ldots, Y_{t+1}\right\}} \sum_{j=1}^{N} \mathbf{P}\left\{X_t = j, X_{t+1} = i, Y_0, \ldots, Y_t\right\} \\
&= \frac{C_i(t+1)}{\mathbf{P}\left\{Y_0, \ldots, Y_{t+1}\right\}} \sum_{j=1}^{N} \mathbf{P}\left\{X_{t+1} = i | X_t = j\right\}\ \mathbf{P}\left\{X_t = j, Y_0, \ldots, Y_t\right\} \\
&= \frac{C_i(t+1)}{\mathbf{P}\left\{Y_{t+1} | Y_0, \ldots, Y_t\right\}} \sum_{j=1}^{N} \mathbf{P}\left\{X_{t+1} = i | X_t = j\right\}\ \mathbf{P}\left\{X_t = j | Y_0, \ldots, Y_t\right\} \\
&= \frac{C_i(t+1)}{\mathbf{P}\left\{Y_{t+1} | Y_0, \ldots, Y_t\right\}} \sum_{j=1}^{N} A_{j,i}(t)\ \alpha_{t|t}(j)\ .
\end{aligned} \tag{3.4}$$

Let $g_t = \mathbf{P}\left\{Y_{t+1} | Y_0, \ldots, Y_t\right\}$, then (3.4) becomes

$$\alpha_{t+1|t+1}(i) \;=\; \frac{C_i(t+1)}{g_t} \sum_{j=1}^{N} A_{j,i}(t)\ \alpha_{t|t}(j)\ , \tag{3.5}$$

where the normalisation quantity

$$g_t \;=\; \sum_{i=1}^{N} C_i(t+1) \sum_{j=1}^{N} A_{j,i}(t)\ \alpha_{t|t}(j)\ , \tag{3.6}$$

can be used for the formation of a detector, as is shown in later sections. The filters are initialised according to

$$\begin{aligned}
\alpha_{0|0}(i) &= \mathbf{P}\left\{X_0 = i | Y_0\right\} \;=\; \frac{\mathbf{P}\left\{Y_0 | X_0 = i\right\}\ \mathbf{P}\left\{X_0 = i\right\}}{g_0} \\
&= \frac{C_0(i)\ \pi_0(i)}{g_0}\ ,
\end{aligned}$$

where the normalisation constant is given by

$$g_0 \;=\; \mathbf{P}\{Y_0\} \;=\; \sum_{i=1}^{N} C_0(i)\ \pi_0(i)\ .$$

In order to compute all the filtered quantities, $O(N^2T)$ calculations are required [62]. Typically, normalisation is not needed for estimation problems. A prime example is maximum *a posteriori* probability (MAP) estimation, where the most likely state is selected as the state estimate at a given time. However, as will be shown, for a detection problem such as track extraction, the normalisation constant provides the likelihood of a sequence of observations, forming the basis of the detector.

3.2.1 Optimal Filtering for Hidden Reciprocal Chains (HRCs)

Hidden reciprocal chains (HRC) can be thought of as being the analogue to hidden Markov chains. This idea is reviewed in this section, after which the optimal filters for the estimation of the state sequence of a HRC and evaluation of the likelihood of a HRC observation sequence are presented. The algorithms for filtering and smoothing for (partially observed) discrete state reciprocal processes were first proposed in [40], and these methods form the basis for the algorithms to be described in subsequent chapters. The normalised filters presented here were appeared first in [41], as well as [19], but not in the context of a detection problem such as track extraction.

Hidden Reciprocal Chains

Suppose that the RC $\mathcal{X} = \{X_0, , \ldots, X_T\}$ is observed via the observation process $\mathcal{Y} = \{Y_0, \ldots, Y_T\}$. Assume that the observation at time t given the state X_t is conditionally independent of X_τ and Y_τ, $\tau \neq t$. This conditional independence implies that

$$\mathbf{P}(Y_0, \ldots, Y_T | X_0, \ldots, X_T) = \prod_{t=0}^{T} \mathbf{P}(Y_t | X_t). \tag{3.7}$$

The process $\mathcal{Y}$ is called a hidden reciprocal chain (HRC) because the property (3.7) is analogous to the usual assumption made for hidden Markov chains. In (3.7) the terms $\mathbf{P}(Y_t | X_t = i)$ are the conditional observation densities of the HRC. The observations may be either discrete or continuous random variables defined on an appropriate probability space, where one would define a parametrised probability density function or probability mass function, respectively.

Optimal HRC Filters

Consider a HRC $\mathcal{Y}$ with state $\mathcal{X}$, known MB transition probability matrices $B^k_{i,j}(t)$, $t = 0, \ldots, T-2$. Given a sequence of observations, $\{Y_0, \ldots, Y_T\}$, the optimal filter, in the Bayesian sense, computes the *a posteriori* probabilities (APP)

$$q_i(t) =: \mathbf{P}(X_t = i | Y_0, \ldots, Y_t)\ , \tag{3.8}$$

for each $t = 0, \ldots, T$, and each $i = 1, \ldots, N$. These probabilities can be calculated for $t = 0, \ldots, T-1$ via

$$q_i(t) = \sum_{k=1}^{N} \mathbf{P}(X_t = i, X_T = k | Y_0, \ldots, Y_t) = \sum_{k=1}^{N} q_i^k(t) \ ,$$

applying the law of total probability. It has been shown that the joint process (X_t, X_T) is Markov [41], therefore analogously to the hidden Markov model filter, it is easily shown via Bayes' rule that $q_i^k(t)$ can be evaluated recursively, for $t = 1, \ldots, T-1$ by

$$q_i^k(t) = \frac{C_i(t) \sum_{j=1}^{N} B_{j,i}^k(t-1)\, q_j^k(t-1)}{h(t)} \ , \tag{3.9}$$

where the normalising term is

$$\begin{aligned} h(t) &= \mathbf{P}(Y_t | Y_0, \ldots, Y_{t-1}) \\ &= \sum_{i,k=1}^{N} C_i(t) \sum_{j=1}^{N} B_{j,i}^k(t-1)\, q_j^k(t-1) \ . \end{aligned}$$

and the terms $C_i(t), t = 0, \ldots, T, i = 1, \ldots, N$ are the evaluated conditional obervation densities in (3.7). Initialisation at $t = 0$ is via

$$q_i^k(0) = \frac{C_i(0)\, \Pi_{i,k}}{h(0)},$$

where $h(0) = \mathbf{P}(Y_0) = \sum_{i,k=1}^{N} C_i(Y_0)\, \Pi_{i,k}$. The APP for the final point follows also from Bayes' rule

$$q_i(T) = \frac{C_i(T) \sum_{k=1}^{N} q_k^i(T-1)}{h(T)}$$

where $h(T) = \sum_{i,k=1}^{N} C_i(T)\, q_k^i(T-1)$. The overall computational cost of the above filtering recursions is $O(N^3 T)$ compared to $O(N^2 T)$ for a HMC filter.

For the detection task, it is common practice to take the logarithm of the evaluated observation density, which can be obtained from the normalisation terms as follows,

$$\log \mathbf{P}(Y_0, \ldots, Y_T) = \sum_{t=0}^{T} \log h(t) \tag{3.10}$$

This approach avoids any potential numerical underflow issues which might arise if the unnormalised MB filters as defined in [40] were used. A track extraction algorithm can be obtained by comparing the log sequence likelihood to a threshold, as shown in sub-section 3.4.

3.3 Observation Model

In this section the novel observation model is defined. This model allows the incorporation of multiple observations in each time epoch t. Such a model may reflect the use of multiple sensors

or multiple sensor modes (e.g. hyper-spectral sensors, multi-mode radars), and also allows the multiple sensor measurements at each time to be recorded asynchronously.Also included are clutter (background noise and interference) in the observation model as well as sensor noise and errors. Such a model also applies to the ground vehicle tracking problem mentioned in the introduction, which is used as an illustrative example. It is assumed here that at most one target is either present for the whole processing interval $t = 0, \ldots, T$. This might be a natural assumption for the application considered, or in multi-target problems, "gating" may be used to separate multiple targets as a pre-processing step. This thesis doesn't consider the problem of multiple targets, although it is a natural extension of the current work, albeit one with considerable challenges from a computational point of view.

At each time t, a vector $Y_t = \left(y_t^1, \ldots y_t^M\right)$ is produced by the sensor system, where $M \geq 1$ is a given integer (for instance the number of sensors), and is assumed constant (although this can be generalised to make M time-varying) and each component y_t^m is a noisy version of a possible target or clutter (false return) state. More complicated mappings from states to observations are also possible as for the general HMC model. It is assumed each y_t^m is a real valued vector quantity (corresponding to a two dimensional position). The ordering of the components of Y_t may be arbitrary under the sensor model considered (e.g. enabling the modelling of asynchronous measurements or transmission of measurements over a network). It is convenient to define a sequence of random variables $\{a_0, \ldots, a_T\}$, called *association variables*, each taking values in the set $\{0, 1, \ldots, M\}$, where for $m = 1, \ldots, M$, $a_t = m$ if and only if the measurement y_t^m corresponds to a true target detection, the remainder being clutter returns. The outcome $a_t = 0$ means that no target is detected by any sensor at time t thus all M components of Y_t correspond to clutter detections. In the absence of additional *a priori* information, the sequence a_t is assumed to be independent and identically distributed (i.i.d.), and is also independent of the state sequence X_t. This assumption is typically an approximation, e.g. if no target is present for all t, then clearly $a_t = 0$ for all t with probability one, and thus a_t is then a dependent sequence. The *a priori* probability distribution for a_t are denoted by

$$\mathbf{P}(a_t = i) = \lambda_i, \quad i \in \{0, ..., M\} \ . \tag{3.11}$$

It is assumed that these quantities can be determined from the sensor and target models. The clutter, as seen by each sensor mode, is modelled by the random process $\mathcal{U} = \{U_t^m : t = 0, \ldots, T, m = 1, \ldots, M\}$, taking values in $\mathcal{S}$, which are assumed i.i.d. across both t and m. Additionally, all U_t^i are assumed independent of the target state when it is present. Let $U_t = \left(U_t^1, \ldots, U_t^M\right)$ denote the vector of clutter random variables at time t. The sensor signal model is assumed to have the conditional independence property

Γ)

U= $\prod_{t=0}^{T} \prod_{m=0}^{M} \mathbf{P}\left(y_t^m | X_t, a_t, U_t\right)$,

which generalises (3.7). Let

$$\begin{aligned} d^m(t) &= \mathbf{P}\left(y_t^m | a_t = 0\right) = \mathbf{E}_U\left\{\mathbf{P}\left(y_t^m | U_t, a_t = 0\right)\right\}, \\ \tilde{c}_i^m(t) &= \mathbf{P}\left(y_t^m | X_t = i, a_t = m\right) \\ &= \mathbf{E}_U\left\{\mathbf{P}\left(y_t^m | U_t, X_t = i, a_t = m\right)\right\}, \\ c_{i,\ell}^m(t) &= \mathbf{P}\left(y_t^m | X_t = i, a_t = \ell\right) \\ &= \mathbf{E}_U\left\{\mathbf{P}\left(y_t^m | U_t, X_t = i, a_t = \ell\right)\right\}, \end{aligned} \tag{3.12}$$

where $\mathbf{E}_U$ denotes expectation with respect to the common prior distribution on the clutter process, which is assumed known. Note that when $a_t = 0$, there's no dependence on a target state X_t because no target is present in this case.

The observation vector likelihood $C_i(t) = \mathbf{P}(Y_t | X_t = i)$ can thus be obtained by applying total probability and then conditioning over the a_t yielding

$$\mathbf{P}\left(y_t^m | X_t = i\right) = \sum_{\ell=1, \ell \neq m}^{M} \lambda_\ell \, c_{i,\ell}(t) + \lambda_m \, \tilde{c}_i^m(t) + \lambda_0 \, d^m(t).$$

Then $C_i(t) = \prod_{m=1}^{M} \mathbf{P}\left(y_t^m | X_t = i\right)$ using the conditional independence of the y_t^m. Numerical examples of the multiple observation model above will be used in sec. 3.5.

To make this sensor modelling concept more concrete, it is useful to return to the ground target tracking example. Consider a scenario where the radar system supplies the TE detector with M measurements, y_t^m at each time t, being locations in two dimensions where target and/or clutter returns originate. The two-dimensional field of view of the surveillance system is divided into a grid of N cells which is determined by the road network (e.g. [18]). These are the target/clutter states. It is assumed that if a target is present then it is represented by one and only one of the measurements terms (i.e. $y_t^{a_t}$). Given models of the sensor processing, clutter, target return (when present) and sensor noise (including state quantisation noise), the quantities in Eqn. (3.12) and thus the $C_i(t)$ terms can, in principle, be determined. Note that the complexity of the HRC filter for the general case ($M > 1$) is $O(N^3T) + O(M^2N^2T)$, where the latter term is the burden of computing the observation densities for the list of M observations at each time t. If $M << N$, the complexity of the filter is dominated by the term $O(N^3T)$.

3.4 Track Extraction Detectors

The observation model along with the associated filter enables the investigation of potential track extraction (TE) benefits from incorporating source-destination awareness into a detector's target model. A track extraction detector is defined as a test with two competing hypotheses. Based on the filter derived in sec. 3.2.1, the likelihood ratio tests (LRT) are formulated, which express how many times more likely the data are under one model than the other. Since an analytic form for the probability distribution of the sequence log likelihood under the null or alternative hypotheses cannot be found analytically in general, numerical approaches are required in order to sensibly set the detection threshold. The simulations implement and assume uniform priors over the hypotheses. In practice these priors could be set by the analyst based on empirical observations from data or prior knowledge.

The null hypothesis is that the observations are all clutter generated ($\lambda_0 = 1$). The alternative hypothesis presumes there is a target, and over many realisations one can expect to receive

target generated observations at a rate $(1-\lambda_0)$:

$$\begin{aligned} H_0: &\quad \lambda_0 = 1, \quad \text{(no target present)} \\ H_1: &\quad \lambda_0 < 1, \quad \text{(a target is present).} \end{aligned}$$

If the target dynamics correspond to a reciprocal target, this detector is called a *reciprocal detector*.It is sensible to also form a detector where the alternative hypothesis instead has sequence log likelihood obtained from a standard HMC filter. The Markovian target dynamics considered are either those of the reference MC, or of the Schrödinger bridge with dynamics given by equation (2.4). This alternative detector is called a *Markov detector* if the reference Markov process is used to model dynamics, or, alternatively, a *Schrödinger detector* if a SB is used. Since the observations Y_t are independent under H_0, the observation log likelihood for the null hypothesis is given by

$$\log \mathbf{P}\left(\mathcal{Y}|H_0\right) = \sum_{t=0}^{T} \log \mathbf{P}\left(Y_t|H_0\right) = \sum_{t=0}^{T} \sum_{m=1}^{M} \log d^m(t).$$

To formulate the LRT for the reciprocal TE detector, the observation log likelihood under H_R is computed using the HRC filter as described in sec. 3.2.1. The LRT for Markov and SB TE detectors under the relevant alternative hypothesis is determined using the appropriate HMC filters.

The performance of all of the detectors and filters considered depends on the ability of the underlying target model to accurately describe the dynamics of the target model in the generated data. It would be expected that the HRC based detector should perform best since it is matched to the data. However, the error of unmatched models relative to the matched RC model may be small. In general this will depend of course on the difference between the underlying target models, which is measured via the KL divergence between the distribution over trajectories which reduces to those of the endpoints as described in sec. 2.2.2. Therefore the numerical studies have focused on the parameters of the endpoints distribution, and suggest an empirical relationship of the form

$$\text{Difference in Detection Error} = f(KL(\Pi_{Data}||\Pi_{Tracker}))$$

where $f(\cdot)$ is some monotonically increasing function, and the tracker's joint distribution may not be matched to that of the data. The difference in detection error is taken to be given by the difference in the "area under the curve" (AUC) of the receiver operator characteristic (ROC) curve for each detector, which can be thought of as a measure of quality of a detector [63].

3.5 Numerical Examples

The simulations presented in this section compare the HRC tracker (filter and detector) to trackers based on a Hidden Markov chain (HMC) and based on a Schrödinger bridge, which is henceforth called a Hidden Schrödinger Chain (HSC) tracker. In all simulations the target trajectory data is generated according to a RC target model, constructed from a reference Markov process, equipped with a distribution Π. All the trackers use target models with dynamics derived from the reference process. The trackers' models therefore differ only in terms of the statistical characterisation of the end-points (and thus their dynamics). The HMC has only

initial state distribution π_0 equal to the marginal distribution of Π at the initial time. The HSC, which attempts to incorporate source-destination information in a model that remains Markov, is supplied with both initial and final marginals of Π, that is π_0 and π_T.

Two HRC trackers are considered, one with joint distribution matched to the RC generating the data, and another unmatched. This latter prompts the introduction of a notion of an "uninformative" joint distribution for a non-Markov RC, one whose marginals match the known marginals π_0 and π_T, but places uniform weighting over destinations for a given source. This is realised by simply taking the outer product of the true marginals. Therefore, the mismatched model has a joint endpoints distribution that factorises into the product of its marginals, but not in such a way that is Markov, see equation (2.11).

3.6 Simulation Design

To represent a typical road network, states take values on a regular two-dimensional 8×8 cellular "gridworld". The reference process is a one-step Markovian random walk model, parametrised by the probability of remaining in a cell p_R and equal probabilities of moving to the neighbouring cells (states). Note that neighbouring cells include those on the diagonal, meaning the random walk is '8-connected', rather than 4-connected. Jumps outside the gridworld are not permitted (although this can be included e.g. [19]). Figure 3.1 shows two sample trajectories. Both targets have the same source-destination pairs (encoded via Π) and dynamics built from the same Markov reference process.

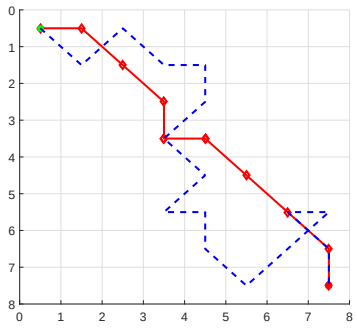


Figure 3.1: *Source destination aware trajectories*: The paths of 2 RC targets which cross the 8×8 lattice from $(1, 1)$ to $(8, 8)$. The green marker (top left) denotes the origin state of any realisation. Each trajectory corresponds to that of a Markov bridge with a fixed start, the simplest type of destination awareness. The red target has $T = 12$ steps to reach state $(8, 8)$, while the blue has $T = 32$.

In order to generate an observation sequence, the independent clutter process U_t^1 uniformly distributed on the state space is also realised (in the case $M = 1$, and similarly for $M > 1$). A sensor detection in the observation sequence is obtained by adding zero-mean Gaussian noise to the centre coordinates of the cell that the target or clutter is in at time t. This illustrative choice can easily be generalised. The added Gaussian noise has equal variance σ^2 in the x and y directions with the x and y components of the noise being statistically independent.

The specific distributions Π for the RC used to generate data are mixtures of the form

$$\Pi_{RC} = \alpha\Pi_{CRC} + (1-\alpha)\Pi_{LRC} \ , \tag{3.13}$$

where $\alpha \in [0,1]$, LRC is a loitering RC as defined in sec. 2.2.2, and CRC is a crossing RC, one which a target crosses the gridworld from any corner to its opposite corner, as in Figure 3.1. Therefore as simulations are conducted with different α, it is possible to test the hypothesis that increasing KL divergence results in an increasing drop in performance from the HMC tracker with respect to the matched model.

3.6.1 Results

Detection results are presented using receiver operator characteristic (ROC) curves to begin with in Figure 3.2, which plot the estimated probability of detection P_D against the estimated probability of false alarm P_F as the LRT detection threshold is varied. Simulations include realisations of both hypotheses in equal number, RC target present and no target present, reflecting the uniform priors chosen, $P(H_0) = P(H_1) = 0.5$. The detectors perform the LRT using the uniform priors, equal penalties for incorrect decisions and no penalties for correct decisions, that is, a minimum error probability test.

To begin with, the single observation scenario is considered, with a clutter rate $\lambda_0 = 50\%$, while α is increased from zero to one. One should expect to see the error of the unmatched models increase with α, which is indeed observed in Figure 3.2. The general multi-observation case produces similar results, except that the performance of all the trackers degraded, but with the reciprocal trackers degrading less than other trackers. The filtering example of Figure 3.3, corresponds to $\alpha = 1$, and a consideration of only CRC targets in the multiple observation scenario with $M = 5$. In this case $1 - \lambda_0$ represents the probability the target is indirectly observed, which is set to 75%. Thus in the multi-observation case, M hits are observed at each time, but occasionally none of the hits will correspond to a target. The performance of the filters is measured by the RMSE associated with the conditional mean estimates of target location.

The phenomenon of increasing performance of HRC over alternative (more poorly matched) models as tracking conditions worsen was observed for increasing sensor detector noise variance σ^2, clutter rate λ_0 and number of sensor hits M. In Figure **??** the effect of increasing M on the trackers' filtering performance is shown, since the effect on the ROC curves is negligible. The x-axis plots different sequence lengths, and the y-axis plots the difference in average-per-sample RMSE (over the interval). Note that the benefit of HRC over HMC decreases with increasing sequence length T, as reported earlier [40].

3.6.2 Discussion

The results show that the HRC tracker, which is matched to the model generating realisations, performs best when compared to the Markov trackers, as expected. Furthermore the monotonic relationship between difference in detection error and KL divergence is strongly supported, as shown in Figure 3.4. The HMC and HSC results reveal that although the SB model incorporates more future information than the Markov model, it can perform no better. A reason for this is that the SB construction introduces source destination pairs having non-zero probability despite these pairs having zero probability for the HRC. A second reason is that the clutter observation model allows the HSC tracker to confidently assign measurements to clutter. If for example the RC is a loitering target that loiters uniformly across the statespace, i.e. $p_r(i) = 1/N$ in (2.12),

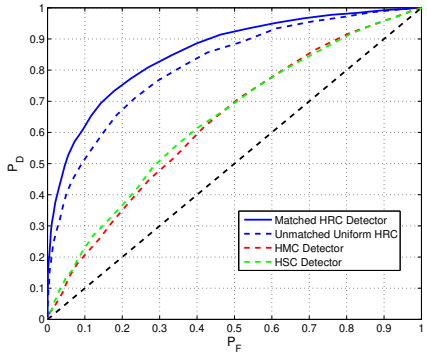
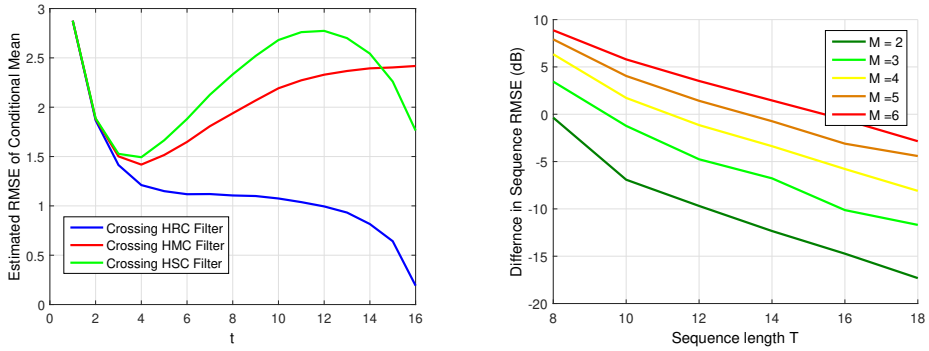


Figure 3.2: *Single Observation Detection Results*: ROC curve for a single observation system with clutter rate $\lambda_0 = 0.5$ and sensor detection noise $\sigma^2 = 1$. The RC has interval length $T = 16$, and $\alpha = 1$.



(a) (b)

Figure 3.3: *Multiple Observation Filtering*: (a) Estimated RMSE of the Conditional Mean for each of the trackers with the target observed for the fraction $1 - \lambda_0 = 0.5$, sensor detector noise of $\sigma^2 = 1$, $T = 16$, and $\alpha = 1$, meaning one can expect the HRC trackers to perform best. (b) Difference in the Estimated Sequence RMSE (in dB) of the Conditional Mean between the matched HRC and the reference HMC trackers. The target was observed for the fraction $\lambda_0 = 0.5$, and sensor detector noise of $\sigma^2 = 1$. The interval lengths increasing from $T = 8$ to 18, and $\alpha = 1$, meaning one can expect the HRC trackers to perform best.

the marginals passed to the SB will be uniform, and the HSC tracker will 'expect' trajectories between any two states. Thus for highly constrained target motion, which is often the case in practice, RC models appear to outperform SB models even though both models have the same marginal distributions on the end-points. In particular, the strong performance of the unmatched HRC in the detection task is noteworthy, despite also having access to only initial and final distributions.

The poorer HSC result can be further understood by considering the illustrative example in Figure 3.5 which shows a crossing target's path $\mathcal{X}$, a noiseless path of the "sensor" detections (target and clutter origins), the observations $\mathcal{Y}$ and the HSC filter's maximum *a posteriori* probability (MAP) estimates. The majority of the HSC estimates are focused around the starting

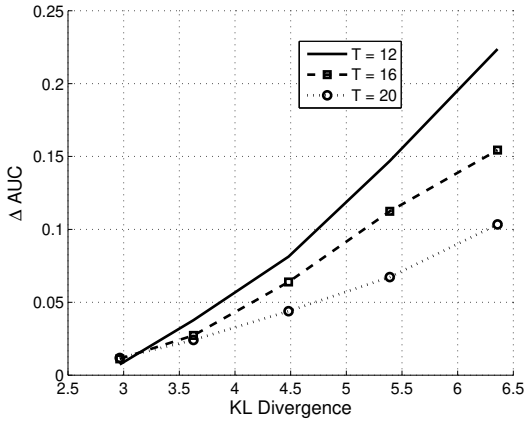


Figure 3.4: ΔAUC *vs. KL Divergence*: Error of unmatched HMC with reference process as target model, taken to be area between ROC curves of matched HRC and ROC curves of HMC (ΔAUC). Estimates for in a single observation scenario, with $\lambda_0 = 0.5$, $\sigma^2 = 1$, $T = 16$ as α increases.

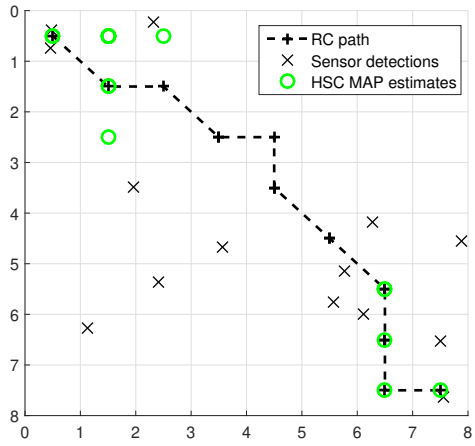


Figure 3.5: *HSC predicting incorrect trajectory*: A target's path from $(1, 1)$ to $(8, 8)$ (dotted black line), with sensor detections (black points) and the HSC filter MAP estimates (green), for a crossing RC ($\alpha = 1$) with $T = 16$. At time $t = 9$ the filter determines that the measurement sequence has higher likelihood with a trajectory of $(1, 1)$ to $(8, 8)$.

vertex $(1, 1)$, since under its model it is possible for a target to start and end at the same location, until at time $t = 9$ the filter determines that the measurement sequence has a higher likelihood with a trajectory of $(1, 1)$ to $(8, 8)$. This effect corresponds to Figure 3.3, where the RMSE in the HSC conditional mean estimates decreases close to T.

3.7 Chapter conclusion

This part of the thesis has considered a non-Markovian target dynamic model called a reciprocal chain (RC), which is able to incorporate a simple notion of intent. The potential of this model to improve the tracking performance over alternative Markov models was studied. A reciprocal chain can be constructed from a Markov chain (MC) reference process together with a joint endpoints distribution which models the source-destination awareness of a target. A track extraction scenario was proposed, with an observation model in which multiple observations could be generated by a single target or clutter. Normalised hidden RC (HRC) filters were reviewed and observation models and corresponding likelihood based detectors were constructed for the track extraction context.

Numerical simulations of targets on a cellular gridworld, with states as cells, and dynamics restricted to one-step walks over cells were developed. HRC filters and detectors were constructed, one with a target model matching that which generated the data, and another unmatched HRC with endpoints distribution replaced by the product of the marginals of the data generating model. These two HRC models were compared via numerical simulations to both a HMC tracker with the reference process as its target model, and a tracker based on the Schrödinger bridge, which models some destination information.

An important insight drawn from these numerical simulations was that the error of the unmatched models, constructed from the same reference process, can be related to the Kullback-Leibler divergence of the joint endpoints distributions between the unmatched model and that which is generating the data. An important secondary insight is that the Schrödinger bridge tracker, despite having correct initial and final marginals pre-specified, can in fact perform worse than a Markov tracker using only the reference process. This supports the main result, which claims that HRC better model source-destination awareness than HMC or HSC. In particular, the results reveal that for reasonable groundtruth trajectories, and access limited to only the initial and final marginals, the reciprocal model appears to be more robust to modelling errors than the Schrödinger bridge tracker.

A natural extension to the single target tracking problem is of course to consider multi-target tracking problem. For Bayes optimal inference, the computational complexity is a significant obstacle, due to the combinatorial explosion with the number of targets M (under all possible orderings of at least M measurements, at each time over a finite set of observations T). This issue is compounded further in the case of reciprocal models where the dependence on the state space size N is cubic. Approximate inference techniques, such as particle filtering (or smoothing), are then required, possibly in combination with various "pruning" heuristics to reduce the number of possible targets and tracks. The difficulties in applying Bayes optimal inference, in combination with a model that attempts to model a more complex data generation process, motivates many researchers to consider discriminative models for tracking. A popular discriminative model, a neural network, is studied in the remaining chapters of the thesis.

Chapter 4

Statistical learning with neural networks

This chapter presents the background material for statistical learning algorithms for binary neural networks. The aim of the chapter is that a reader with some familiarity with statistics, optimisation or machine learning will be able to read this section and appreciate the various ingredients that go into the design of the current state of the art algorithms. The chapter begins with a review of statistical learning problems, quickly specialising to the problem of binary parameters. In the process the relationships to traditional statistical estimation are discussed. Following this, continuous optimisation techniques are reviewed for large multi-layer neural networks. This helps to inform a reader of the problems one can expect to encounter in the binary case.

The contribution of this chapter is the presentation of diverse background material that intersects on the problem of statistical learning with binary neural networks. The material presented here is known, but to the best available knowledge has not been presented together in this way. As such, this is a useful contribution, since the literature relating to the binary neural network learning problem is disconnected. In summary, this chapter achieves the following,

- Introduces the background for statistical learning problems that one may attempt to solve using neural networks (continuous or binary), focusing on generic classification problems in a supervised setting

- Argues that the standard problem that neural networks solve is empirical risk minimisation (ERM) with a convex surrogate loss, and subsequently identifies agreement between ERM with Bayesian approaches, and the special cases of maximum likelihood estimation, for the logistic regression problem

- Defines standard continuous neural network models and the stochastic gradient methods used for optimisation of the convex surrogate ERM objective. Presents some basic intuition on the impact of second order properties of the loss surface on the optimality of the gradient based optimisation process

4.1 Statistical learning

This chapter begins with a general introduction to what will be referred to as "statistical learning". This is considered to be a set of problems that share many similarities to those in statistics and optimisation, to name just two fields. However, there are important differences as well. In particular, the dimensionality of the data is large, and the number of parameters are typically of the same order as the number of data points. This differs from traditional statistics in particular, where methods rely on the assumption that the number of data points grows asymptotically, while the number of parameters of the model remains fixed. In order to tether the discussion, examples will be kept simple and relevant to neural networks, in particular binary neural networks where possible.

Consider the logistic regression conditional probability model in a supervised setting [64]. Assume the dataset is a finite set of independently and identically distributed (i.i.d.) samples from some unknown distribution, which is denoted as $D = \{x^m, y^m\}_{m=1}^M$ with $y \in \{+1, -1\}$ and $x^m \in \mathbb{R}^N$ vector valued in general. In a supervised setting, the task amounts to finding a conditional model of the data, which can be written as a conditional probability distribution $p(\mathbf{y}|\mathbf{x})$. The standard logistic regression model has parameters which are referred to as weights w_i parameterising the conditional model of the data,

$$p(\mathbf{y} = 1|\mathbf{x} = x; w) = \frac{1}{1 + \exp(-\sum_i w_i x_i)} \tag{4.1}$$

where random variables are denoted with bold font. The aim of "learning" is to determine these weights by some automatic method. Consider maximum likelihood and Bayesian estimation [65], two approaches usually pitted against one another in traditional statistics. The maximum likelihood objective to be maximised is the log likelihood of the data,

$$\log p(D; w) = \sum_{m=1}^{M} \log p(\mathbf{y}^m|\mathbf{x}^m, w) \tag{4.2}$$

where the likelihood $L(w; D) = p(D; w)$ is a function from the parameter space to the real numbers, relating the parameters to the data, in particular, expressing which parameter values are more plausible. This objective is differentiable in the case where the w_i are continuous, it is plain that this is not the case when restricted to binary values.

Bayesian estimation, on the other hand, aims to find the posterior distribution over the weights w given the data given a prior $p(\mathbf{w})$ which can encode the binary constraints,

$$p(\mathbf{w}|D) = \frac{\prod_{m=1}^M p(\mathbf{y}^m|\mathbf{x}^m, \mathbf{w})p(\mathbf{w})}{p(D)} \tag{4.3}$$

Unfortunately, computing the partition function, or normalisation constant $Z = P(D)$ is intractable. This is because it is defined by the multi-dimensional integral

$$p(D) = \int d\mathbf{w} \prod_{m=1}^{M} p(\mathbf{y}^m|\mathbf{x}^m, \mathbf{w})p(\mathbf{w}), \tag{4.4}$$

which either has no closed form solution, or is computational intractable to evaluate. Therefore, approximations are needed. This difficulty has motivated the development of approximate message passing algorithms which attempt to approximate the posterior that is otherwise intractable

to compute. Message passing algorithms have been derived for the binary weight case [32],[66] but these algorithms fail to scale to large neural networks, where the number of parameters and data points are large. The first reason for this failure to scale is the sheer size of the probabilistic graphical model which corresponds to the variables under the posterior distribution to be calculated [67]. Despite message passing algorithms like belief propagation scaling linearly with the number of edges, which is otherwise efficient, for large networks and datasets this number is easily in the billions [68]. The second reason behind the lack of scalability is that the graphical model does not have the form of a tree, meaning that it contains many loops and therefore requiring many iterations for an algorithm such as loopy belief propagation converge.

The Bayesian picture can be extended, as is well known in statistics, to form a hierarchical Bayesian model, where a parameterised prior distribution has a (hyper-) prior over its own parameters [69]. This is the statistical setting that most popular gradient algorithms for binary weight neural networks could be couched. Henceforth, the binary weights will be denoted[1] as $\mathbf{s}_i$. Then, for instance, if some distribution for the binary weights $p(\mathbf{s};\theta)$ is assumed, with some parameters θ, then one objective that could be optimised is the log marginal likelihood,

$$\log p(D;\theta) = \sum_m \log \mathbb{E}_{p(\mathbf{s};\theta)} p(y^m|x^m,\mathbf{s}) \tag{4.5}$$

where the notation $\mathbb{E}_{p(\mathbf{s};\theta)}[f(\mathbf{s})] = \sum_{\mathbf{s}} f(\mathbf{s})p(\mathbf{s};\theta)$ defines the expectation of a function $f(\mathbf{s})$ of the weights. This objective can be viewed as the formal starting point for several of the current most popular approximations from which gradient descent algorithms can be derived, by taking the gradient with respect to each parameter θ.

The steps leading to the current state of the art gradient algorithms will be outlined more carefully in subsequent chapters, but essentially the next key step is to model the binary random variables as independent, and then approximate them in some way with continuous random variables. The most common approach is to apply the central limit theorem to the linear combination input to the logistic regression or "local field" $\mathbf{h} = \sum_i \mathbf{s}_i x_i \sim \mathcal{N}(\mu, \sigma^2)$. The mean and variance of this Gaussian are then functions of each individual θ_i that controls the distribution over each weight $\mathbf{s}_i$. Following this Gaussian assumption, an estimate of the gradient at a particular value of θ_i can be derived.

In the statistics literature the use of the objective (4.5), to set the hyperparameters θ, is known as type II maximum likelihood, or an empirical Bayesian method [69] [70] . In machine learning, the objective is called the evidence [71]. In this thesis it is argued that this is just one frame from which to view these methods. A possibly more compelling view is around empirical and expected risk minimisation. This alternative does not assume any particular model of the data. Instead the average performance of a classifier under a given loss function, under the empirical and "true" data distribution is studied.

One might be inclined to ask what the Bayesian, and non-Bayesian approaches of estimation have to do with each other or indeed the problem of learning a classifier from a finite dataset? This is a good question. A satisfactory answer is to consider the supervised learning classification problem from the perspective of decision theory. This helps to separate the definition and modelling of the problem, from the statistical estimation and subsequent approximations that lead to algorithms for learning the binary weights. In taking a step back to discuss decision theory, this will lead quite sensibly to the ideas of expected and empirical risk minimisation mentioned,

[1] The choice of the letter 's' for the binary weight $\mathbf{s}_i$ follows a convention in physics, where one usually considers binary *spins*

which are arguably better suited to describing the large scale gradient learning methods [72]. Following this high level discussion the thesis will move on to the specific approximations to derive gradient based binary neural network algorithms.

4.1.1 Binary classification by learning from data

A classification problem is defined on an input space $\mathcal{X}$, corresponding to input data, and an output space $\mathcal{Y}$ corresponding to labels. In the binary classification case, $\mathcal{Y} = \{+1, -1\}$. It is assumed that there exists some joint distribution $p(\mathbf{xy})$ over $\mathcal{X} \times \mathcal{Y}$ that generates a finite dataset, $D = \{x^m, y^m\}_{m=1}^M$. A classifier is a function $\phi : \mathcal{X} \to \mathcal{Y}$, that takes an input and predicts the label of the output. The task of supervised learning, in the classification context, is to find a good classifier given only the dataset D, assumed to be sampled i.i.d. from the joint $p(\mathbf{x}, \mathbf{y})$.

In order to measure the quality of any given classifier, it makes sense to consider some kind of cost or loss function associated with any decision. A sensible loss function therefore measures how different a predicted label $\hat{y} = \phi(x)$ is from the true label y, given some input. A common or "natural" loss is the the so called $0-1$ loss,

$$L(\phi(x), y) = I\{y \neq \phi(x)\} \tag{4.6}$$

where I is the indicator function. The statistical risk of a given classifier $\hat{y} = \phi(x)$ with this particular loss function, is defined as:

$$R(\phi) = \mathbb{E}_{p(\mathbf{x},\mathbf{y})} L(\phi(\mathbf{x}), \mathbf{y}) \tag{4.7}$$

The performance of the best possible classifier is known as the (optimal) Bayes' risk. For the $0-1$ loss function, such a classifier will minimise the probability of error on new data. A particular classifier that attains Bayes' risk is known as the Bayes classifier

$$\phi^*(x) = 2I\{\eta(x) > 0.5\} - 1 \tag{4.8}$$

where $\eta(x) = P_{\mathbf{y}|\mathbf{x}}(\mathbf{y} = 1|\mathbf{x} = x)$ is the evaluated conditional probability distribution for $\mathbf{y}$ given $\mathbf{x} = x$. More precisely, $\phi^*(x)$ is a (possibly non-unique) minimum of the statistical risk (4.7).

Now that the classification problem is defined, it makes sense to discuss the aspect of modelling the data. In Bayesian decision theory, as one might guess, the objective is to obtain the Bayes classifier via Bayesian methods. If one assumes knowledge of the form of the joint distribution, one can estimate the posterior over the parameters given the finite dataset $D = \{x^m, y^m\}_{m=1}^M$. For a new input data point x_{M+1}, one can apply Bayes' rule and obtain a Bayesian classifier,

$$p(\mathbf{y}|\mathbf{x} = x^{M+1}) = \frac{p(\mathbf{x} = x^{M+1}, \mathbf{y})}{p(\mathbf{x} = x^{M+1})} = \frac{\int p(\mathbf{x} = x^{M+1}, \mathbf{y}|\theta) p(\theta|D)}{p(\mathbf{x} = x^{M+1})} \tag{4.9}$$

where it is assumed that the joint probability distribution over the data is parameterised by some θ, and $p(\theta|D)$ is the posterior over the parameters[2]. Modelling concerns the specification of $p(\mathbf{x}, \mathbf{y}|\theta)$, where one can distinguish between generative and discriminative modelling approaches.

[2]Note that θ is used for parameters within more general discussion, and w is used for the weights when discussing neural networks.

A generative model aims to model the generation of the data $\mathbf{x}$, via $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y}; \theta)p(\mathbf{y}; \theta)$, where the prior on class membership probabilities is given by $p(\mathbf{y}; \theta)$ (with dependence on the model parameters θ in general). In the discriminative approach[3], the practitioner chooses the form of the conditional probability distribution $p(\mathbf{y} = 1|\mathbf{x} = x; \theta)$, and ignores any estimation of $p(\mathbf{x})$. A common approach is to model this conditional distribution via a linear combination of the data $\mathbf{x}$, passed through a logistic function, as in (4.1). In this case the classifier corresponding to this conditional distribution is

$$\phi(x) = \arg\max_y p(\mathbf{y} = y|\mathbf{x} = x; w) = \arg\max_y \frac{1}{1 + \exp(-y \sum_i w_i x_i)} \tag{4.10}$$

where the parameters are considered to be the weights, $\theta = \{w_i\}_{i=1}^N$. Note that the linear combination in the above expression is known as a linear discriminant in this context, since the vector of weights $\{w_i\}_{i=1}^N$ defines a linear decision boundary [65]. Determining the parameters of a classifier when the w_i are continuous can be cast as the logistic regression problem, common to statistics [65].

There is a special case for generative classifiers under the assumption that $p(\mathbf{x}|\mathbf{y}; \theta)$ is in an exponential family. This assumption results in the conditional likelihood $p(\mathbf{y}|\mathbf{x} = x)$ taking the form of a logistic function of a linear combination of the input's elements. That is, one can make a connection to logistic regression. This connection is discussed extensively in [64]. Of course, with either generative or discriminative models, the decision theoretic approach demands statistical estimation of the unknown parameters. The advantage of Bayesian approaches over non-Bayesian ones, assuming the joint model correctly represents the underlying data distribution, is extensively discussed in decision theory texts [69], [70].

4.1.2 Bayesian and non-Bayesian estimation

In the context of the supervised binary classification problem above, it makes sense to discuss the relationship between Bayesian and non-Bayesian estimation methods, with an interest in binary weight constraints kept in mind. The non-Bayesian methods considered in this thesis are maximum likelihood and what is commonly referred to as empirical Bayes methods. Note that choosing to use a particular estimation approach is independent of whether or not there a hierarchical model, with hyperparameters, has been assumed.

It is well known that calculating the posterior distribution is typically intractable, so approximate Bayesian methods are used in practice. At an extreme end of the scale, one can obtain point estimates for parameters in a Bayesian frame, and this is known as maximum *a posteriori* probability (MAP) estimation, defined as

$$\theta_{MAP} = \arg\max_\theta \log p(\theta|D) = \arg\max_\theta \ \log \frac{\prod_{i=1}^M p(y^m|x^m, \theta)p(\theta)}{p(D)} \tag{4.11}$$

$$\propto \sum_{m=1}^M \log p(y^m|x^m, \theta) + \log p(\theta) \tag{4.12}$$

The above expression relates to a common relationship reported in machine learning material that essentially argues that "MAP estimation is equivalent to regularised maximum likelihood" [71]. The term $R(\theta) = \log p(\theta)$ corresponds in this context to what is commonly called a

[3]Non-probabilistic approaches such as SVMs and other margin based techniques are ignored in this discussion.

regulariser. Generally, regularisation can mean either the addition information in order to solve an ill-posed problem (such as a least squares problem involving singular matrices), or the addition of information to prevent overfitting [73]. In the present context, the regulariser corresponds to the latter case. It is clear that a maximum likelihood objective including a regulariser $R(\theta)$ can be equivalent to MAP estimation. For example a ℓ_2 penalty corresponds to placing a zero mean Gaussian prior on the parameters, via the term $\log p(\theta)$.

A "more Bayesian" approach aims to approximate the posterior, and the two most popular approaches are Monte Carlo simulation or variational approximations. This chapter ignores the former, and focuses on the latter for several reasons. First, it is quite popular in the neural network literature, second it is fundamentally an optimisation problem, and finally it has been applied to the binary weight perceptron problem with success [32], [66]. It is worth mentioning there has been justifiable confusion in the machine learning binary neural network literature over which algorithms are Bayesian or not, since certain variational algorithms closely resemble the empirical Bayesian approaches [33], [74].

4.1.3 Variational approximations to Bayesian estimation

A variational approach to Bayesian estimation attempts to find a tractable proxy $q(\theta)$ to the true posterior $p(\theta|D)$, selecting a good approximation by minimising a distance measure between the two distributions. The most common distance measure is the Kullback-Liebler divergence,

$$KL(q||p) = \int q(\theta) \log \frac{q(\theta)}{p(\theta|D)} d\theta \tag{4.13}$$

In machine learning the above divergence, with q as the first argument, is known as "variational Bayes" KL divergence, and is used because the expectation should be tractable given a tractable choice of q. As argued in [66] this choice cannot be used in the case of binary weights, since this divergence may not be defined, for example if $q(\theta) \neq 0$ and $p(\theta) = 0$ for some θ. An alternative choice for the binary weight problem is the "reverse", $KL(p||q)$. Of course, the expectation over p is intractable to compute, and so the alternative is to minimise this divergence *locally*, leading to message passing algorithms such as expectation propagation. See [66] for a comprehensive review.

The most common choice for the approximating distribution $q(\theta)$ is what is known as the mean field approximation, a fully factorised model:

$$q(\theta) = \prod_i q_i(\theta_i) \tag{4.14}$$

In the case of binary weights s_i, each factor is a Bernoulli distribution, with some parameter θ_i that controls the probability that the weight is $+1$ or -1. This is the choice used in the best performing message passing algorithms for the binary perceptron, including belief propagation [32] and expectation propagation [66]. This mean field approximate posterior is the same choice as in the hierarchical *models* used in several of the most common gradient algorithms, which is identified here as optimising empirical Bayes objectives.

4.1.4 Empirical Bayes optimisation

In the case of binary weights $\mathbf{s}_i$, the hierarchical model can be written as follows

$$p(\mathbf{y}|\mathbf{x};\theta) = \sum_{\mathbf{s}} p(\mathbf{y}|\mathbf{x},\mathbf{s}) p(\mathbf{s};\theta) = \mathbb{E}_{p(\mathbf{s};\theta)} p(\mathbf{y}|\mathbf{x},\mathbf{s}) \tag{4.15}$$

where the sum is over all possible values $\mathbf{s} = \{\mathbf{s}_i\}_{i=1}^N$. As stated, the most common model choice for the binary weights is the mean field assumption, $p(\mathbf{s}|\theta) = \prod_i p(\mathbf{s}_i|\theta_i)$. The most common objective used to optimise the binary weight models is the log marginal likelihood,

$$\log p(\mathbf{y}|\mathbf{x}, \theta) = \log \sum_{\mathbf{s}} p(\mathbf{y}|\mathbf{x}, \mathbf{s}) \prod_i p(\mathbf{s}_i; \theta_i) = \log \mathbb{E}_{p(\mathbf{s}_i|\theta_i)} p(\mathbf{y}|\mathbf{x}, \mathbf{s}) \tag{4.16}$$

The marginal likelihood is also known as the model evidence, following [71]. The mean field model of the binary weights opens the door to simple approximations that then yield a differentiable objective function when applied within the marginal likelihood. These approximations are detailed in the next section. Of course, the marginal likelihood is just one objective that could be used to choose the hyperparameters w in the hierarchical model. The following subsections present a framework that handles this question in a particularly pleasing way.

4.1.5 Expected and Empirical Risk minimisation

The "statistical learning theory" proposed by Vapnik [75] and others, which is advocated in machine learning by [72], agrees with decision theory in that the optimal risk is the Bayes risk and the Bayes' classifier is one particular classifier that attains this performance. Where this school of thought deviates is that it makes no assumptions on the form of the joint distribution $P_{\mathbf{xy}}$. Instead, this theory considers the role of a classifier's "function class" $\mathcal{C}$ to be of prime importance. More generally, the theory views the process of determining a classifier $\phi(x)$ from the data as a stochastic process itself.

Under statistical learning theory, the only access to the joint distribution is the finite data set D. In general, no assumptions on the form or type of distribution are made, though it is assumed the data is sampled independently and identically. The function class $\mathcal{C}$ is typically assumed to be restricted in the sense that it may or may not be sufficiently "rich" to attain the optimal Bayes risk. The idea that the process of determining the classifier is random sits naturally across the algorithmic approaches to learning, especially optimisation algorithms such as stochastic gradient descent (SGD) as in [72]. To illustrate, the statistical risk from this view is defined as a conditional expectation:

$$R(\phi) = \mathbb{E}_{p(\mathbf{x},\mathbf{y})}[L(\phi(\mathbf{x}), \mathbf{y})|\phi] \tag{4.17}$$

where the conditional statement is to ensure the definition is sensible even if $\phi(\cdot)$ is itself random. Thus expected risk is the expectation over the possibly random learning process that determines ϕ,

$$\text{Expected Risk} = \mathbb{E}\big[\mathbb{E}_{p(\mathbf{x},\mathbf{y})}[L(\phi(\mathbf{x}), \mathbf{y})|\phi]\big] = \mathbb{E}[L(\phi(\mathbf{x}), \mathbf{y})] \tag{4.18}$$

The empirical risk is of course the empirical estimate of the expected risk[4], over the data D. The purpose of the statistical learning theory of Vapnik, and related theories, is to relate the minimisation of the empirical risk to the expected risk, given D, a class of functions $\mathcal{C}$ and a process of determining a classifier.

[4]Somewhat confusingly, this is the same as an empirical estimate of the statistical risk defined in (4.7)

4.1.6 Convex surrogates

As stated at the beginning of this section, the common choice for the loss function for binary classification is the $0-1$ loss (4.6). In an equally sensible approach, one can place gradient based learning under the umbrella of empirical risk minimisation by replacing the $0-1$ loss with a convex, possibly differentiable, surrogate loss. This will obviously have statistical consequences, and surrogate losses that are upper bounds for the $0-1$ loss have been studied thoroughly in recent years [76],[77].

Of particular interest is the logistic loss surrogate for a classifier $\phi(x) = \text{sign}(f(x))$.

$$\sum_m I\{y^m \neq \phi(x^m)\} \approx \sum_m \log(1 + \exp(-y^m f(x^m))) \tag{4.19}$$

Consider the case that the discriminant $f(x)$ is linear, that is $f(x) = \sum_i w_i x_i$. If the aim is to minimise the empirical risk using this differentiable, convex loss one obtains the negative log-likelihood objective that is minimised for logistic regression, corresponding of course to a maximum likelihood approach.

Equally, one can consider empirical risk minimisation under the constraint of binary weights as well. It is convenient to start by defining the classifier in terms of the following expectation over some stochastic binary weights,

$$\phi(x) = \arg\max_y p(y|\mathbf{x} = x; \theta) = \arg\max_y \mathbb{E}_{p(s;\theta)}[p(y|\mathbf{x} = x, s)] \tag{4.20}$$

If this classifier is evaluated via its empirical risk under a logistic loss, this corresponds to minimising a negative log marginal likelihood of a hierarchical model,

$$-\sum_m \log p(y|\mathbf{x} = x; \theta) = -\sum_m \log \mathbb{E}_{p(\mathbf{s};\theta)}[p(y^m|\mathbf{x} = x^m, \mathbf{s})] \tag{4.21}$$

$$= -\sum_m \log \mathbb{E}_{p(\mathbf{s};\theta)}[\frac{1}{1 + \exp(-y^m \sum_i s_i x_i^m)}] \tag{4.22}$$

Alternatively one might define the empirical risk in no connection to the marginal likelihood, but still have a classifier with stochastic weights. A empirical risk objective that works in accordance with this might instead take the average over weights outside the log

$$\sum_m \mathbb{E}_{p(\mathbf{s};\theta)}[\log\left(1 + \exp(-y^m \sum_i \mathbf{s}_i x_i^m)\right)] \tag{4.23}$$

It is possible to connect these two expressions in one framework via the theory of risk sensitive optimisation.

4.1.7 Risk sensitive optimisation

Consider the problem of minimising the expectation over variable s, $\mathbb{E}_{p(\mathbf{s};\theta)}[L(s)]$, with respect to some parameters θ, for a given loss function $L(s)$. The risk sensitive generalisation is given by,

$$F_\beta(\theta) = -\frac{1}{\beta} \log \int p(s;\theta) e^{-\beta L(s)} ds \tag{4.24}$$

where β controls whether the objective is risk averse or risk seeking. See [78] for a simple, intuitive introduction to these notions. One can illustrate what is meant by risk averse or seeking by expanding F_β for small β,

$$F_\beta(\theta) \approx \mathbb{E}[L(s)] - \beta\mathbb{V}[L(s)] \tag{4.25}$$

where $\mathbb{V}[L(s)]$ is the variance of the loss. So if $\beta < 0$ the objective F_β will be risk averse, since it tries to minimise the variance in the loss as well as the first moment. If $\beta > 0$ the objective is risk taking.

For the case of the stochastic binary weight logistic regression model, $L(s) = \log p(y|x, s)$, it is possible to identify $\beta = 1$ with the marginal log likelihood, and $\beta = 0$ with the risk neutral cost. Note that the risk sensitivity is with respect to the stochastic binary weights, and not the random data.

What is interesting is the clear similarity between the quenched and annealed averages in statistical mechanics, which is reviewed in Chapter 6. It is beyond the scope of this thesis to describe with confidence the physical correspondence of the two expressions for each value of β. Speculatively, for $\beta = 0$ the objective F_β may correspond to a physical system where the stochastic binary weights s_i fluctuate on a similar time scale to the "couplings" between weights, induced by the data points (x^m, y^m). For $\beta = 1$ this might correspond to the weights fluctuating at a higher rate than the couplings.

4.2 Continuous optimisation of neural networks

Neural networks have risen to prominence in the past decade, with impressive performance in image recognition and new domains [3]. The literature on this topic is vast, therefore the scope is limited to produce a "shortest path" to understanding the optimisation of binary weight neural networks.

The chapter begins by defining a neural network and its optimisation objective, in light of the previous section on statistical learning. The workhorse of neural networks or "deep learning" is arguably stochastic gradient descent algorithm (SGD), which optimises the objective by adapting the weights or parameters of the network. Although much can be said for the methods collected under the name "backpropagation" which enables the efficient application to very large models and datasets [3], these are not fundamental to explaining the success of deep neural network, such as their generalisation performance. Since the algorithms presented in this thesis are also based on SGD to optimise binary weight networks, via continuous surrogate models, it is sensible to consider some of the fundamental problems that can be encountered when attempting to optimise standard neural networks. This will inform readers of the problems to expect in the binary case.

The view presented of the optimisation of neural networks, the basic elements of SGD and its obstacles, largely follows [72]. This comprehensive review concerns optimisation methods for large datasets, providing a treatment both rigorous and intuitive.

4.2.1 Neural networks

A neural network is typically defined as a deterministic non-linear function. The most common model is the multi-layer feedforward type, and the most basic "architecture" of this type is the fully-connected model. The network is composed of $N^\ell \times N^{\ell-1}$ weight matrices W^ℓ and

bias vectors b^ℓ in each layer $\ell \in \{0, \dots, L\}$, with elements $W^\ell_{ij} \in \mathbb{R}$ and $b^\ell_i \in \mathbb{R}$. Given an $M-$dimensional input vector $x^0 \in \mathbb{R}^M$, the network is defined in terms of the following vector equations,

$$x^\ell = \phi^\ell(h^\ell_{\text{cts}}), \qquad h^\ell_{\text{cts}} = W^\ell x^{\ell-1} + b^\ell \tag{4.26}$$

where the pointwise non-linearity is, for example, $\phi^\ell(\cdot) = \tanh(\cdot)$, and the (vector of) neurons of the network are defined as the outputs of the non-linearities, $x^\ell = \phi^\ell(W^\ell x^{\ell-1} + b^\ell)$, as is standard. The term h^ℓ_{cts} is referred to as the pre-activation field. The overall neural network function can be written in a way that makes its concatenated form more clear:

$$f(x_0; W, b)) = W^L \phi^L(W^{L-1}\phi^{L-2}(\cdots \phi^1(W^1 x^0 + b^1)) + b^{L-1}) + b^L \tag{4.27}$$

The network training is the process of adapting of the weights and biases so as to minimise a loss function. In the supervised learning setting this is the sum of the individual losses of the network to the target y_m:

$$\mathcal{L}_{\mathcal{D}}(f; W, b) = \sum_{\mu \in \mathcal{D}} \ell(y_\mu, f(x_\mu; W, b)) = \sum_{\mu \in \mathcal{D}} \log p(y_\mu = f(x_\mu; W, b)) \tag{4.28}$$

As discussed, it is possible to interpret the loss function $\ell(\cdot)$ as a log-likelihood of a data point, and the weights and biases as the parameters which one might try to estimate via a maximum likelihood method. However in this thesis $\mathcal{L}_{\mathcal{D}}$ is isntead recognised as the Empirical Risk, following [72]. Modern neural networks are trained by stochastic gradient descent, that is, some variant of the original Robbins-Monro "stochastic approximation" algorithm [72].

4.2.2 Optimisation objective

Consider again the supervised learning problem, defined on an input space $\mathcal{X}$ and output space $\mathcal{Y}$ of labels, with some joint distribution $p_{\mathbf{xy}}$ over $\mathcal{X} \times \mathcal{Y}$. Given a finite dataset $D = \{x_i, y_i\}_{i=1}^M$, which is assumed to be sampled i.i.d. from the joint $P_{\mathbf{xy}}$, and a neural network of a given architecture and set of weights $\{W\}$, the optimisation objective minimised by neural networks is the empirical risk

$$\text{Empirical Risk: } R_M(W) = \frac{1}{M} \sum_{m=1}^{M} \ell(f(x_m; W), y_m) \tag{4.29}$$

where $\ell(\cdot, \cdot)$ is a continuous, differentiable loss function. For convenience the loss per example is defined via the notation

$$\ell_m(W) := \ell(f(x_m; W), y_m) \tag{4.30}$$

Again, when interested in classification, this would likely a be a surrogate to the $0-1$ loss. In minimising the empirical risk, ideally one would minimise the expected risk.

The choices of surrogate losses that are common have been reported to not have a large impact on the statistical properties of the resulting classifiers, in the sense that a classifier which minimises the expected risk with the surrogate loss will also minimise the $0-1$ loss [76]. However the choice will affect the optimisation process, and therefore the typical minima attained in the

first place. In the context of neural networks, the combination of the loss, the neural network function and the stochastic gradient descent method have led to remarkable performance in terms of the expected risk. Explanations to this phenomena leads one to attempt to bound the expected risk in the sense of [75]. This is the subject of current research, where the first non-vacuous bounds were only established recently [79].

4.2.3 Stochastic gradient methods

The most basic stochastic gradient method selects one example pair at each iteration k and updates the weights W_k, all written as a single vector, according to

$$W_{k+1} = W_k - \alpha \nabla_{W_k} \ell_{m_k}(W_k) \tag{4.31}$$

where m_k corresponds to the seed selecting the pair (x_k, y_k), and α is the step size or learning rate. This is referred to as an online update. The full batch method calculates the gradient over the dataset D, that is $\nabla_{W_k} R_M(W_k)$.

Implementations of SGD for large neural networks utilise "mini-batches" which, as the name suggests, selects a random subset of the data $\mathcal{S}_k \subset \{1, \dots, M\}$ and performs the update

$$W_{k+1} = W_k - \alpha \sum_{i \in \mathcal{S}_k} \nabla \ell_i(W_k) \tag{4.32}$$

One reason practitioners have historically opted for mini-batch SGD is due to computational constraints. Usually the datasets are so large that computing the gradient over the full batch of data is costly, and also unnecessary if many of the data points share similarities.

In the non-convex optimisation setting, it is known that algorithms with some level of noise can assist in finding "better minima" or regions of the parameter space corresponding to lower loss values, in a similar vein to simulated annealing. In recent years, in the search for explanations of the success of deep learning, more sophisticated arguments have been made as to how the non-isotropic noise of mini-batch SGD contributes to the low expected risk obtained by these models [80], [79].

For more detailed motivation of using mini-batches, both intuitive, practical and theoretical, pleaserefer to [72].

4.2.4 Second order properties

There are two issues with the SGD algorithm commonly identified [72]. The first is the noisy estimates of the true full batch gradient, which essentially prevent the algorithm from converging to a final solution. This chapter closes by focusing on another common issue, arising from non-linearities of the function being optimised, and the subsequent ill-conditioning of the gradient process.

The motivation for considering second order properties of gradient descent can be understood by investigating what happens in the full batch (ie. noiseless) case, when the loss surface is approximately a quadratic well. Suppose the weights are given by W_k, and the step direction and size is given by αg where α is the learning rate (step size) and $g = \frac{\partial \ell(W)}{\partial W}|_{W=W_k}$ the gradient at W_k. The update will be of the form $W_{k+1} = W_k - \alpha g_k$.

Consider the Taylor expansion of the loss function to second order, about the new point W_k:

$$\ell(W) \approx \ell(W_k) + (W - W_k)^T g + \frac{1}{2}(W - W_k)^T H (W - W_k) \tag{4.33}$$

where H is the matrix of second order partial derivatives, known as the Hessian matrix. Substituting now $W = W_k - \alpha g$ into this expression,

$$\ell(W_k - \alpha g) \approx \ell(W_k) - \alpha g^T g + \frac{1}{2}\alpha^2 g^T H g \tag{4.34}$$

The three terms in this expression are the original function value, the expected improvement due to the slope of the function and the correction for curvature of the function. Notice that when α is too large, the value of the loss function can actually increase.

The optimal learning rate can likewise be obtained by first differentiating (4.33) with respect to W_k,

$$\frac{\partial \ell(W)}{\partial W_k} = \frac{\partial \ell(W_k)}{\partial W_k} + (W - W_k)^T H \tag{4.35}$$

with the second order term disappearing since it is constant for a quadratic loss. Letting $W = W_{min}$ and setting this expression to zero, one finds[5]

$$W_{min} = W_k - H^{-1}\frac{\partial \ell(W_k)}{\partial W_k} \tag{4.36}$$

Thus the optimal step size for each parameter depends on the curvature of the loss.

This approach is the basis of a Newton-type optimisation scheme. In general, the loss functions are of course not quadratic in the weights, but the idea is that the functions may be well approximated by quadratics at least locally. Therefore, the conditioning of the problem, meaning the ratio of largest to smallest eigenvalues of the Hessian, does indeed have an impact on the optimisation process. This has been confirmed for standard neural networks [81], and this thesis makes some inroads to filling out this picture for binary neural network algorithms.

4.2.5 Decomposition of the Hessian

The Hessian of a neural network admits a specific decomposition for the case of a $K - class$ classification problem. To reveal this decomposition, consider a "final layer" to a neural network, known as the the softmax output, interpreted as providing a probabilistic classifier. So, the loss function is given by

$$\ell(f(x_m, W), y_m) = -\log p(y|f(x_m, W)) \tag{4.37}$$

$$\text{where } \; p(y = k|f(x_m, W)) = \frac{\exp\big(h_k(x_m, W)\big)}{\sum_{k'=1}^{K} \exp\big(h_{k'}(x_m, W)\big)} \tag{4.38}$$

By the chain rule, the Jacobian matrix (of first order partial derivatives) can be written as $J = J_{\ell,h} J_{h,w}$. In this notation $J_{\ell,h}$ represents the Jacobian of the loss $\ell(\cdot)$ with respect to the vector of output fields $h_k(x_m, W)$, and $J_{h,w}$ the Jacobian of the fields with respect to all weights of the network.

In turn, the Hessian of the network can be written out by applying the product rule to this Jacobian decomposition,

$$H = \frac{\partial}{\partial \vec{w}}(J_{\ell,h} J_{h,w}) = J'_{h,w} H_{\ell,h} J_{h,w} + \sum_{k'=1}^{K} (J_{\ell,h})_{k'} H_{h_{k'},w} \tag{4.39}$$

[5]Note that $\frac{\partial \ell(W)}{\partial W_k}_{W_{min}} = 0$ by definition.

The Hessian matrices $H_{L,h}$ and $H_{h_k,w}$ can be defined similarly, in correspondence with the Jacobian.

The reason for writing down this decomposition is that the technical contributions of this thesis concern the Jacobian of the network at initialisation, and the effect this has on the trainability of a deep binary neural network. From the second order optimisation perspective, it can be seen immediately that the surface at the initial point will depend on the conditioning of the Jacobian, via this decomposition of the Hessian. In controlling the Jacobian, it is possible to control the Hessian at initialisation.

4.3 Chapter conclusion

This chapter has reviewed the commonly presented approaches to statistical leaning, with particular focus on the problem of learning neural networks with binary parameter constraints. While other approaches may exist, especially in the optimisation literature, the material presented is encountered most commonly in the machine learning literature. In the next chapter the problem of both binary weights and neurons is considered.

The overall contribution of this chapter, as discussed in the introduction, is the clarification of the various approaches to selecting an optimisation objective for a neural network with binary parameters. This is a significant contribution, since the machine learning literature often presents presents an objective without discussion or comparison. On occasion, this has led to confusion, with some algorithms being described as Bayesian, due to the problem considering *stochastic* binary parameters, or weights. The next chapter presents some examples of this.

Chapter 5

Optimising neural networks with binary weights and neurons

This chapter outlines the specific approximations that yield differentiable surrogate neural networks that, when optimised, yield binary neural networks that "solve" the original statistical learning problem. More specifically, this chapter shows how to apply various principled and heuristic approximations to the empirical risk objective function for stochastic binary neural networks, which involves an expectation over the stochastic binary variables to produce various surrogates.

The chapter begins by clearly defining deterministic and stochastic binary networks. This includes the case of binary weights and continuous neurons, and the case that both weights and neurons are binary. The basic ideas of gradient approximation and estimation are then summarised. A heuristic from the machine learning community known as the "Straight-Through estimator" is also presented. This heuristic has been proposed with little justification, yet appears to work well in practice.

Following this, several approximations used for multi-layer stochastic binary neural networks are presented. The approximations yield what will be referred to in this thesis as surrogate networks, that are differentiable. The chapter focuses on two approaches based on Gaussian approximations, deriving the surrogates with the help of a novel Markov chain representation of stochastic neural networks.

The first approximation yields a deterministic surrogate, and the other remains stochastic, which is referred to as the "perturbed" surrogate. In the latter case an algorithm for the case of both stochastic binary weights and neurons, which is new. Also presented is a new surrogate based on what is known as the "concrete" gradient estimator.

The contributions of this chapter are both the presentation of new algorithms, as well as taking a step toward unifying the presentation of algorithms for optimising binary neural networks, similar to the previous chapter. In summary, this chapter makes the following contributions,

- Presents several gradient approximations and estimators for functions involving discrete, stochastic variables, based on either the Gaussian central limit theorem (CLT), or heuristics
- Presents derivations for surrogate networks by starting with a novel Markov chain representation of a stochastic neural network. This starting point encompasses all stochastic binary neural network surrogates

- Presents a clear derivation of a deterministic surrogate network based on Gaussian central limit assumptions and analytic integration. This surrogate can be developed from a range of neuron noise models, including logistic or Gaussian (probit) models, as well as simple deterministic sign(·) neuron
- Presents the derivation of a so called "perturbed" surrogate, based on Gaussian central limit assumptions and a single Monte Carlo sampling step. This includes a formulation with stochastic neurons, an unexplored algorithm and thus a new contribution in itself
- Defines clean notation the recursive forward propagation equations for the Gaussian based surrogate networks that reveal some similarities to standard neural networks
- Proposes a new heuristic approximation based on the concrete estimator for discrete stochastic variables. This approximation is related qualitatively to the perturbed surrogate.

5.1 Binary neural networks

5.1.1 Deterministic binary neural networks

First recall the conventions adopted in this thesis. A neural network model is composed of $N^\ell \times N^{\ell-1}$ weight matrices W^ℓ and bias vectors b^ℓ in each layer $\ell \in \{1, \dots, L\}$, with elements $W^\ell_{ij} \in \mathbb{R}$ and $b^\ell_i \in \mathbb{R}$. Given an input vector $x^0 \in \mathbb{R}^{N_0}$, the network is defined in terms of the following vector equations,

$$x^\ell = \phi^\ell(h^\ell_{\text{cts}}), \qquad h^\ell_{\text{cts}} = W^\ell x^{\ell-1} + b^\ell \tag{5.1}$$

where the pointwise non-linearity is for example $\phi^\ell(\cdot) = \tanh(\cdot)$.

A deterministic binary neural network simply has weights $W^\ell_{ij} \in \{\pm 1\}$ and $\phi^\ell(\cdot) = \text{sign}(\cdot)$, and otherwise the same propagation equations. Of course, this is not differentiable, thus it is common practice to define stochastic binary variables in order to smooth out the non-differentiable network. The product of training a surrogate of a stochastic binary network is ideally a deterministic binary network that is able to generalise from its training set. It is possible to also use the stochastic binary network, but this is not as computationally advantageous in standard hardware.

5.1.2 Stochastic binary neural networks

In stochastic binary neural networks, the weight matrices are denoted as $\mathbf{S}^\ell$ with all weights[1] $\mathbf{S}^\ell_{ij} \in \{\pm 1\}$ being independently sampled binary variables with probability is controlled by the mean $M^\ell_{ij} = \mathbb{E}\mathbf{S}^\ell_{ij}$. Neuron activation in this model are also binary random variables, due to pre-activation stochasticity and to inherent noise. In this chapter parameterised neurons are considered, such that the mean activation conditioned on the pre-activation is given by some function taking values in $[-1, 1]$, i.e. $\mathbb{E}[\mathbf{x}^\ell_i \,|\, \mathbf{h}^\ell_i] = \phi(\mathbf{h}^\ell_i)$, for example $\phi(\cdot) = \tanh(\cdot)$. The propagation rules for the stochastic network are written as follows:

$$\mathbf{S}^\ell \sim p(\,\bullet\,; M^\ell); \qquad \mathbf{h}^\ell = \frac{1}{\sqrt{N^{\ell-1}}}\,\mathbf{S}^\ell \mathbf{x}^{\ell-1} + b^\ell; \qquad \mathbf{x}^\ell \sim p(\,\bullet\,; \phi(\mathbf{h}^\ell)) \tag{5.2}$$

[1] Random variables are denoted with bold font.

Notice that the distribution of $\mathbf{x}^\ell$ factorizes when conditioning on $\mathbf{x}^{\ell-1}$, as $p(\mathbf{x}^{\ell+1}|\mathbf{x}^\ell) = \prod_i p(\mathbf{x}_i^{\ell+1}|\mathbf{x}^\ell, \mathbf{S}_i^\ell)$. The form of the neuron's mean function $\phi(\cdot)$ depends on the underlying noise model. A binary random variable $\mathbf{x} \in \{\pm 1\}$ with $\mathbf{x} \sim p(\mathbf{x}; \theta)$ can be expressed via its latent variable formulation $\mathbf{x} = \text{sign}(\theta + \alpha\mathbf{L})$. In this form θ is referred to as a "natural" parameter, and the term $\mathbf{L}$ is a latent random noise, whose cumulative distribution function $\sigma(\cdot)$ determines the form of the non-linearity since $\phi(\cdot) = 2\sigma(\cdot) - 1$. In general the form of $\phi(\cdot)$ will impact on the surrogates' performance, including within and beyond the mean field description presented here. However, a result from the analysis in Chapter 7 is that choosing a deterministic binary neuron, ie. the sign$(\cdot)$ function, or a stochastic binary neuron, produces the same signal propagation equations, up to a scaling constant.

5.2 Gradient estimators and approximations

Recall that in the stochastic binary learning problem defined here, the following expected loss function is considered,

$$\mathcal{L}(M, b) = -\frac{1}{P}\sum_{\mu=1}^{P} \log \mathbb{E}_{\mathbf{S},\mathbf{x}}\, p(y_\mu \,|\, x_\mu, \mathbf{S}, \mathbf{x}, b), \tag{5.3}$$

For simplicity, consider the term in the logarithm for some arbitrary cost $C(s)$, $\mathbb{E}_{p(s;\theta)}[C(s,\theta)]$ that is to be minimised via a gradient descent procedure. $C(s,\theta)$ is a cost function with two types of dependence on the parameter θ. The explicit dependence of the cost on parameters θ is known as a pathwise dependence. In contrast, the cost depends implicitly on θ through the random variable s. This is said to be a dependence in distribution or in measure. Writing down the gradient of this expression with respect to θ,

$$\frac{\partial \mathbb{E}_{p(S;\theta)}[C(S,\theta)]}{\partial\theta} = \mathbb{E}_{p(S;\theta)}\frac{\partial C(S,\theta)}{\partial\theta} + \int C(S,\theta)\frac{\partial p(S;\theta)}{\partial\theta}dS \tag{5.4}$$

The first term is a pathwise gradient estimator, which can be estimated via Monte Carlo simulation. The second term can be problematic, but one possibility is to form the following estimator by application of the reverse chain rule to the logarithm function,

$$\int C(S,\theta)\frac{\partial p(S;\theta)}{\partial\theta}d\theta = \int C(S,\theta)\frac{\partial \log p(S;\theta)}{\partial\theta}p(S;\theta)dS \tag{5.5}$$

This is known variously as the score function estimator, the likelihood ratio method [82] or REINFORCE estimator [83]. From this point, it is possible to again use Monte Carlo estimates. This is an unbiased estimator of the gradient, but unfortunately it suffers from impractically high variance, so in general a pathwise estimator is preferred. Obtaining a pathwise estimator is possible if the random variable is "reparameterisable", meaning that it is possible to rewrite the random variable to change a cost function's dependence on a parameter to a pathwise dependence.

A prime example are random variables with location-scale distributions, such as the Gaussian. Explicitly, it is possible to write a one dimensional Gaussian variable $h \sim \mathcal{N}(\mu, \sigma^2)$ as $h = \mu + \sigma\epsilon$ where $\epsilon \sim \mathcal{N}(0,1)$. Thus a distributional dependence can be turned into a pathwise dependence

$$\mathbb{E}_{p(h;\mu,\sigma^2)}C(h) = \mathbb{E}_{p(\epsilon)}C(\mu + \sigma\epsilon) \tag{5.6}$$

This is the key "trick" used to obtain practical low variance estimators. Of course, this reparameterisation trick does not work for discrete variables directly, and so some approximations are needed to move from discrete to continuous variables.

5.2.1 A heuristic: "straight through estimator"

A popular heuristic proposed by the machine learning community [84] and used most widely [23], is the so called "straight-through estimator". The approximation defines the derivative for the sign function $f(z) = \text{sign}(z)$as follows,

$$g(z) = \begin{cases} 1 & \text{if } |z| < 1 \\ 0 & \text{else} \end{cases} \tag{5.7}$$

Of course, the derivative of $f(z) = \text{sign}(z)$ would have an impulse at the origin, and so this "approximation" is something In the application to neural networks, every weight of the network W_{ij}^{ℓ} is then passed through the $\text{sign}(\cdot)$ function. Likewise, each neuron has as activation function the $\text{sign}(\cdot)$ function as well. Thus, on the "forward pass" the network is entirely binary during training, but has derivatives defined "synthetically".

It is not clear why this heuristic is an "estimator", since the original presentations did not discuss what is being estimated. Theoretical explanations for this approach are lacking to date, despite the heuristic's apparent success [23]. In Chapter 8 the signal propagation properties of binary networks are studied, which is relevant for the forward pass defined in this approach.

5.3 Differentiable surrogate networks

The idea behind several recent papers [33], [85], [86], [74] is to adapt the mean of the binary stochastic weights, with the stochastic model essentially used to "smooth out" the discrete variables and arrive at a differentiable function, open to the application of continuous optimisation techniques. In this chapter now derive both the deterministic surrogate and local reparamterization trick based surrogates, in a common framework. Once again, a supervised classification task is considered, with training set $\mathcal{D} = \{x_\mu, y_\mu\}_{\mu=1}^{P}$, with y_μ the label. A loss function is defined in order to train the surrogate model as follows,

$$\mathcal{L}(M, b) = -\frac{1}{P}\sum_{\mu=1}^{P} \log \mathbb{E}_{\mathbf{S},\mathbf{x}}\, p(y_\mu \,|\, x_\mu, \mathbf{S}, \mathbf{x}, b), \tag{5.8}$$

For a given input x_μ and a realization of weights, neuron activations and biases in all layers, denoted by $(\mathbf{S}, \mathbf{x}, b)$, the stochastic neural network produces a probability distribution over the classes. Expectations over weights and activations are given by the mean values, $\mathbb{E}\mathbf{S}^{\ell} = M^{\ell}$ and $\mathbb{E}[\mathbf{x}^{\ell}|\mathbf{h}^{\ell}] = \phi(\mathbf{h}^{\ell})$.

The starting point for the derivations comes from rewriting the expectation (5.8) as the marginalization of a Markov chain, with layers indexes corresponding to time indices $\ell \in \{1, \ldots, L\}$.

Markov chain representation of stochastic neural network:

$$\mathbb{E}_{\mathbf{S},\mathbf{x}}\, p(y_\mu \,|\, x_\mu, \mathbf{S}, b, \mathbf{x}) = \sum_{\mathbf{S},\mathbf{x}\,:\,\mathbf{x}^0=x_\mu} p(y_\mu \,|\, \mathbf{x}^L) \prod_{\ell=1}^{L} p(\mathbf{x}^\ell \,|\, \mathbf{x}^{\ell-1}, \mathbf{S}^\ell)\, p(\mathbf{S}^\ell; M^\ell)$$
$$= \sum_{\mathbf{S}^L,\mathbf{x}^{L-1}} p(y_\mu|\mathbf{S}^L, \mathbf{x}^{L-1}) p(\mathbf{S}^L) \sum_{\mathbf{S}^{L-1},\mathbf{x}^{L-2}} p(\mathbf{x}^{L-1}|\mathbf{x}^{L-2}, \mathbf{S}^{L-1}) p(\mathbf{S}^{L-1}) \cdots \sum_{\mathbf{S}^1} p(\mathbf{x}^1|x_\mu, \mathbf{S}^1) p(\mathbf{S}^1) \tag{5.9}$$

where in the second line the dependence on M^ℓ is dropped from the notation for $p(\mathbf{S}^\ell; M^\ell)$, for brevity. Therefore, for a stochastic network the forward pass consists in the propagation of the joint distribution of layer activations, $p(\mathbf{x}^\ell\,|x_\mu)$, according to the Markov chain. The explicit dependence on the initial input x_μ is also dropped from now on.

In what follows $\phi(\mathbf{h}^\ell)$ will denote the average value of $\mathbf{x}^\ell$ according to $p(\mathbf{x}^\ell)$. The first step to obtaining a differentiable surrogate is to introduce continuous random variables. By taking the limit of large layer width and appealling to the central limit theorem, the field $\mathbf{h}^\ell$ can be modelled as Gaussian, with mean $\bar{h}^\ell$ and covariance matrix Σ^ℓ.

Assumption 1: (CLT for stochastic binary networks) *In the large N limit, under the Lyapunov central limit theorem, the field $\mathbf{h}^\ell = \frac{1}{\sqrt{N^{\ell-1}}}\mathbf{S}^\ell\mathbf{x}^{\ell-1} + b^\ell$ converges to a Gaussian random variable with mean $\bar{h}_i^\ell = \frac{1}{\sqrt{N^{\ell-1}}}\sum_j M_{ij}^\ell \phi(\mathbf{h}_j^{\ell-1}) + b_i^\ell$ and covariance matrix Σ^ℓ with diagonal $\Sigma_{ii}^\ell = \frac{1}{N^{\ell-1}}\sum_j 1 - (M_{ij}^\ell\phi(\mathbf{h}_j^{\ell-1}))^2$.*

While this assumption holds true for large enough networks, due to $\mathbf{S}^\ell$ and $\mathbf{x}^{\ell-1}$ independency, the Assumption 2 below, is stronger and tipically holds only at initialization.

Assumption 2: (correlations are zero)
The independence of the pre-activation field $\mathbf{h}^\ell$ between any two dimensions is assumed. Specifically the covariance $\Sigma = Cov(\mathbf{h}^\ell, \mathbf{h}^\ell)$ is assumed to be well approximated by $\Sigma_{MF}^\ell(\phi(\mathbf{h}^{\ell-1}))$, with MF denoting the mean field (factorized) assumption, where

$$\big(\Sigma_{MF}^\ell(x)\big)_{ii'} = \delta_{ii'}\,\frac{1}{N^{\ell-1}}\sum_j 1 - (M_{ij}^\ell\phi(\mathbf{h}_j^{\ell-1}))^2 \tag{5.10}$$

This assumption approximately holds assuming the neurons in each layer are not strongly correlated. In the first layer this is certainly true, since the input neurons are not random variables[2]. In subsequent layers, since the fields $\mathbf{h}_i^\ell$ and $\mathbf{h}_j^\ell$ share stochastic neurons from the previous layer, this cannot be assumed to be true. It is expected that this correlation will not play a significant role, since the weights act to decorrelate the fields, and the neurons are independently sampled. However, the choice of surrogate influences the level of dependence. The sampling procedure used within the local reparametrization trick reduces correlations since variables are sampled, while the deterministic surrogate entirely discards them.

The surrogate network, studied in the remainder of this chapter, is obtained by successively approximating the marginal distributions, $p(\mathbf{x}^\ell) = \int d\mathbf{h}^\ell\; p(\mathbf{x}^\ell|\mathbf{h}^\ell) \approx \hat{p}(\mathbf{x}^\ell)$, starting from the first layer. Such an approximation can be achieved by either (i) marginalising over the Gaussian

[2]In this case the variance is actually $\frac{1}{N^{\ell-1}}\sum_j \big(1 - (M_{ij}^1)^2\big)(x_{\mu,j})^2$.

field using analytic integration, or (ii) sampling from the Gaussian (or some other approximating distribution). After this, the approximation $\hat{p}(\mathbf{x}_i^\ell)$ is used to form the Gaussian approximation for the next layer, and so on.

5.3.1 Deterministic surrogate

The analytic integration can be performed based on the analytic form of $p(\mathbf{x}_i^{\ell+1}|\mathbf{h}^\ell) = \sigma(\mathbf{x}_i^\ell \mathbf{h}_i^\ell)$, with $\sigma(\cdot)$ a sigmoidal function. In the case that $\sigma(\cdot)$ is the Gaussian CDF, one obtains $\hat{p}(\mathbf{x}_i^\ell)$ exactly by the Gaussian integral of the Gaussian cumulative distribution function,

$$\hat{p}(\mathbf{x}_i^\ell) = \int dh\ \sigma(\mathbf{x}_i^\ell h)\, \mathcal{N}(h\,; \bar{h}_i^\ell, \Sigma_{MF,ii}^\ell) = \Phi(\frac{\bar{h}_i^\ell}{(1+\Sigma_{MF}^\ell)_{ii}^{1/2}} \mathbf{x}_i^\ell) \tag{5.11}$$

Since the approximation starts from the first layer, all random variables are marginalised out, and thus $\bar{h}_i^\ell$ has no dependence on random $\mathbf{h}_j^{\ell-1}$ via the neuron means $\phi(\mathbf{h}^\ell)$ as in Assumption 1. Instead, there is a dependence on means $\bar{x}^\ell = \mathbb{E}_{\mathbf{h}^\ell}\mathbb{E}\left[\mathbf{x}^\ell \,|\, \mathbf{h}^\ell\right] = \mathbb{E}_{\mathbf{h}^\ell}\, \phi(\mathbf{h}^\ell)$. Thus it is convenient to define the mean under $\hat{p}(\mathbf{x}_i^\ell)$ as $\varphi^\ell(\bar{h}, \sigma^2) = \int dh\ \phi^\ell(h)\mathcal{N}(h\,; \bar{h}, \sigma^2)$. In the case that $\sigma(\cdot)$ is the Gaussian CDF, then $\varphi^\ell(\cdot)$ is the error function.

Finally, the forward pass can be expressed as

$$\bar{x}^\ell = \varphi^\ell(h^\ell) \qquad h^\ell = (1+\Sigma_{MF}^\ell)^{-\frac{1}{2}} \bar{h}^\ell \qquad \bar{h}^\ell = \frac{1}{\sqrt{N^{\ell-1}}}\, M^\ell \bar{x}^{\ell-1} + b^\ell, \tag{5.12}$$

This is a more general formulation than that in [33], which considered sign activations, which is obtained in the appendices as a special case. Furthermore, in all implementations the algorithms "backpropagate" through the variance terms $\Sigma_{MF}^{-\frac{1}{2}}$, which were ignored in the previous work of [33]. Note that the derivation here is simpler as well, not requiring complicated Bayesian message passing arguments, and approximations therein.

Integrating over stochastic or deterministic binary neurons

This section carefully steps through the deterministic surrogate approximation in greater detail. The form of each neuron's probability distribution depends on the underlying noise model. As described in the previous section for the "concrete" approximation, one can express a Bernoulli random variable $\mathbf{S} \in \{\pm 1\}$ with $\mathbf{S} \sim p(\mathbf{S}; \theta)$ as,

$$\mathbf{S} = \text{sign}(\theta + \alpha \mathbf{L}) \tag{5.13}$$

In this form θ is referred to as a "natural" parameter, from the statistics literature on exponential families [87]. The term $\mathbf{L}$ is a latent random noise, which determines the form of the probability distribution. It makes sense to also introduce a scaling α to control the variance of the noise, so that as $\alpha \to 0$ the neuron becomes a deterministic sign function.

Letting $\alpha = 1$ for simplicity, one can see that the probability of the Bernoulli variable taking a positive value is

$$p(\mathbf{S} = +1) = \int_{-\infty}^{-\theta} p(\mathbf{L}) d\mathbf{L} \tag{5.14}$$

where $p(\mathbf{L})$ is the known probability density function for the noise $\mathbf{L}$. The two common choices of noise models are Gaussian or logistic noise. The Gaussian of course has shifted and scaled $\text{erf}(\cdot)$

function as its cumulative distribution. The logistic random variable has the classic "sigmoid" or logistic function as its CDF, $\sigma(z) = \frac{1}{1+e^{-z}}$.

Thus, the probability of a the variable being positive is a function of the CDF. In the Gaussian case, this is $\Phi(\theta)$. By symmetry, the probability of $p(\mathbf{S} = -1) = \Phi(-\theta)$. Thus, the probability distribution for the Bernoulli random variable in general is the CDF of the noise $\mathbf{L}$, and one can write $p(\mathbf{S}) = \Phi(\mathbf{S}\theta)$. In the logistic noise case $p(\mathbf{S}) = \sigma(\mathbf{S}\theta)$.

For the stochastic neurons, the natural parameter is the incoming field $\mathbf{h}_i^\ell = \sum_j \mathbf{S}_{i,j}^\ell \mathbf{x}_j^{\ell-1} + b_i^\ell$. Assuming this is approximately Gaussian in the large layer width limit, one can successively marginalise over the stochastic inputs to each neuron, calculating an approximation of each neuron's probability distribution, $\hat{p}(\mathbf{x}_i^\ell)$. This approximation is then used in the central limit theorem for the next layer, and so on.

For the case of neurons with latent Gaussian noise as part of the Bernoulli model, the integration over the pre-activation field (assumed to be Gaussian) is exact. Explicitly,

$$\begin{aligned} p(\mathbf{x}_i^\ell) &= \sum_{\mathbf{x}^{\ell-1}} \sum_{\mathbf{S}^\ell} p(\mathbf{x}_i^\ell | \mathbf{x}^{\ell-1}, \mathbf{S}^\ell) p(\mathbf{S}^{\ell-1}) \hat{p}(\mathbf{x}^\ell) \\ &\approx \int \Phi(\mathbf{x}_i^\ell \mathbf{h}_i^\ell) \mathcal{N}(\mathbf{h}_i^\ell | \bar{h}_i^\ell, (\Sigma_{MF}^\ell)_{ii}) \\ &= \Phi\left(\frac{\bar{h}_i^\ell}{\sqrt{1 + 2(\Sigma_{MF}^\ell)_{ii}}} \mathbf{x}_i^\ell \right) := \hat{p}(\mathbf{x}_i^\ell) \end{aligned} \tag{5.15}$$

where $\Phi(\cdot)$ is the CDF of the Gaussian distribution. Again Σ_{MF} denotes the mean field approximation to the covariance between the stochastic binary pre-activations. The Gaussian expectation of the Gaussian CDF is a known result, which is stated in more generality in the next section, where neurons with logistic noise are also considered

This new approximate probability distribution $\hat{p}(\mathbf{x}_i^\ell)$ can then used as part of the Gaussian CLT applied at the next layer, since it determines the means of the neurons in the next layer,

$$\bar{x}^\ell = \mathbb{E}_{\mathbf{h}^\ell} \phi(\mathbf{h}^\ell) = 2\Phi\left(\frac{\bar{h}_i^\ell}{\sqrt{1 + (\Sigma_{MF}^\ell)_{ii}}} \mathbf{x}_i^\ell \right) - 1 := \varphi(\mathbf{h}^\ell) \tag{5.16}$$

Following these steps from layer to layer, it is clear approximate means for the neurons are being propagated forwards, combined non-linearly with the means of the weights. Given the approximately analytically integrated loss function, it is possible to perform gradient descent with respect to the means and biases, M_{ij}^ℓ and b_i^ℓ.

In the case of deterministic sign() neurons a particularly simple expression is obtained. In this case the "probability" of a neuron taking, for instance, positive is just Heaviside step function of the incoming field. Denoting the Heaviside with $\Theta(\cdot)$, one obtains

$$\begin{aligned} p(\mathbf{x}_i^\ell) &= \sum_{\mathbf{x}^{\ell-1}} \sum_{\mathbf{S}^\ell} p(\mathbf{x}_i^\ell | \mathbf{x}^{\ell-1}, \mathbf{S}^\ell) p(\mathbf{S}^{\ell-1}) \hat{p}(\mathbf{x}^{\ell-1}) \\ &\approx \int \Theta(\mathbf{x}_i^\ell \mathbf{h}_i^\ell) \mathcal{N}(\mathbf{h}_i^\ell | \bar{h}_i^\ell, (\Sigma_{MF}^\ell)_{ii}) \\ &\approx \Phi\left(\frac{\bar{h}_i^\ell}{(\Sigma_{MF}^\ell)_{ii}^{-\frac{1}{2}}} \mathbf{x}_i^\ell \right) := \hat{p}(\mathbf{x}_i^\ell) \end{aligned} \tag{5.17}$$

The network forward equations can be written out for the case of deterministic binary neurons, since it is a particularly elegant result. In general,

$$\bar{x}_i^\ell = \varphi(\eta\, h^\ell), \quad h^\ell = \Sigma_{MF}^{-\frac{1}{2}} \bar{h}^\ell, \quad \bar{h}^\ell = M^\ell x^{\ell-1} + b^\ell \tag{5.18}$$

where $\varphi(\cdot) = \text{erf}(\cdot)$ is the mean of the next layer of neurons, being a scaled and shifted version of the neuron's noise model CDF. The constant is $\eta = \frac{1}{\sqrt{2}}$, standard for the Gaussian CDF to error functin conversion.

Approximate Gaussian integration of sigmoidal functions

This section presents the approximate integration of stochastic neurons with logistic noise as part of their latent variable models. The logistic case is an approximation built on the Gaussian case, motivated by approximating the logistic CDF with the Gaussian CDF. The reason to use logistic CDFs, rather than just considering latent Gaussian noise models which integrate exactly, is not justified in any rigorous or experimental way. Any such analysis would likely consider the effect of the tails of the logistic versus the Gaussian distributions, where the logistic tails are much heavier than those of the Gaussian. One historic reason for considering the logistic function is the prevalence of logistic-type functions (such as $\tanh(\cdot)$) in the neural network literature. The computational cost of evaluating either logistic or error functions is similar, so there is no motivation from the efficiency side. Instead it seems a historic preference to have logistic type functions used with neural networks. The effects of each function are investigated more closely in Chapter 7.

As seen in the previous subsection, the integration over the analytic probability distribution for each neuron gave a function which enables the calculation neuron means in the next layer. It makes sense then to directly calculate the expression for the means.

The Gaussian integral of the Gaussian CDF was used in the previous section to derive the exact probability distribution for the Bernoulli neuron in the next layer. The result is well known, and can be stated in generality as follows,

$$\int_{-\infty}^{\infty} \Phi(ay) \frac{e^{-\frac{(y-x)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} dy = \Phi(\frac{x}{\sqrt{1+a^2\sigma^2}}) \tag{5.19}$$

A logistic noise Bernoulli neuron can be calculated using this result as well. The idea is to approximate the logistic noise with a suitably scaled Gaussian noise. However, since the overall network approximation results in propagating means from layer to layer, one can equivalently approximate the means, replacing $\tanh(\cdot)$ with the erf function. Specifically, given $f(x;\alpha) = \tanh(\frac{x}{\alpha})$, an approximation is $g(x;\alpha) = \text{erf}(\frac{\sqrt{\pi}}{2\alpha}x)$, by requiring equality of derivatives at the origin. In order to establish this, consider

$$f'(0;\alpha) = (1 - \tanh^2(0/\alpha)\frac{1}{\alpha} = \frac{1}{\alpha} \tag{5.20}$$

and

$$\frac{d\,\text{erf}(x;\sigma)}{dx}|_{x=0} = \frac{2}{\sqrt{\pi\sigma^2}} e^{-x^2/\sigma^2}|_{x=0} = \frac{2}{\sqrt{\pi\sigma^2}} \tag{5.21}$$

Equating these, gives $\sigma^2 = \frac{4\alpha^2}{\pi}$, thus $\sigma = \frac{2\alpha}{\sqrt{\pi}}$.

The approximate integral over the Bernoulli neuron mean is then

$$\int_{-\infty}^{\infty} f(y;\alpha)\frac{e^{-\frac{(y-x)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}dy \approx \int_{-\infty}^{\infty} \text{erf}(\frac{\sqrt{\pi}}{2\alpha}y)\frac{e^{-\frac{(y-x)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}dy \tag{5.22}$$

$$= \text{erf}(\frac{\sqrt{\pi}}{2\alpha\gamma}x) \tag{5.23}$$

$$\text{with } \gamma = \sqrt{1+\frac{\pi\sigma^2}{2\alpha^2}} \tag{5.24}$$

If desired, this can be approximated again with a $\tanh(\cdot)$ using the $\tanh(\cdot)$ to $\text{erf}(\cdot)$ approximation in reverse. The scale parameter of this $\tanh(\cdot)$ will be $\alpha_2 = \frac{\pi}{4\alpha\gamma}$. If $\alpha = 1$ as is standard, then

$$\text{erf}(\frac{\sqrt{\pi}}{2\gamma}x) \approx \tanh(\frac{\pi x}{4\gamma}) \tag{5.25}$$

5.3.2 Perturbed surrogate

This section presents a Monte Carlo based approximation which produces what is referred to in this thesis the perturbed surrogate model. The approximation is known as a pathwise estimator for a gradient, as reviewed earlier.

A pathwise estimator can be formed by rewriting the incoming Gaussian field $\mathbf{h} \sim \mathcal{N}(\mu, \Sigma)$ as $\mathbf{h} = \mu + \sqrt{\Sigma}\,\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. Thus expectations over $\mathbf{h}$ can be written as expectations over $\boldsymbol{\epsilon}$ and approximated by sampling. The resulting network is thus differentiable, albeit not deterministic. The forward propagation equations for this surrogate are

$$\bar{\mathbf{x}}^\ell = \phi^\ell(\mathbf{h}^\ell), \qquad \mathbf{h}^\ell = \frac{1}{\sqrt{N^{\ell-1}}}\,M^\ell\bar{\mathbf{x}}^{\ell-1} + b^\ell + \sqrt{\Sigma^\ell_{MF}(\bar{\mathbf{x}}^{\ell-1})}\,\boldsymbol{\epsilon}^\ell \tag{5.26}$$

This approximation, known in the machine learning as the local reparameterisation trick [88] has been previously used to obtain differentiable surrogates for binary networks. The authors of [86] considered only the case of stochastic binary weights, since they did not write the network as a Markov chain. [74] considered stochastic binary weights and neurons, but relied on other approximations to deal with the neurons, having not used the Markov chain representation.

The basic idea of this approximation, of reparameterising the Gaussian field and taking a Monte Carlo sample, was used in both [86], [74]. However, it is important to note that in [86] the neurons were deterministic and continuous, not binary. Also, in [74] the neurons were sampled using the concrete approximation detailed in the next section. Note that by sampling neurons, the subsequent fields have no correlations amongst neurons. This was not remarked upon in these papers.

Perturbed surrogate for stochastic binary weights and continuous neurons

In the case that the stochastic binary network has continuous neurons and stochastic binary weights, it is possible to follow a similar derivation using the Markov chain representation in conjunction with the reparameterisation trick. The Markov chain over the fields can be written

directly in this case, and in the second line make use of the CLT directly to write integrals

$$\mathbb{E}_{\mathbf{S},\mathbf{x}}\, p(y_\mu \,|\, x_\mu, \mathbf{S}, b) = \sum_{\mathbf{S},\, \mathbf{x}^0 = x_\mu} p(y_\mu \,|\, \mathbf{h}^L; M^L) \prod_{\ell=1}^{L} p(\mathbf{h}^\ell \,|\, \mathbf{h}^{\ell-1}; M^\ell)$$
$$\approx \int_{d\mathbf{h}^L} p(y_\mu|\mathbf{h}^L) \int_{d\mathbf{h}^{L-1}} p(\mathbf{h}^L|\mathbf{h}^{L-1}) \cdots \int_{d\mathbf{h}^1} p(\mathbf{h}^2|\mathbf{h}^1) p(\mathbf{h}^1; x_\mu) \tag{5.27}$$

where again the dependence on M^ℓ has been dropped from the notation in the second layer, for brevity. Applying the reparameterisation trick as before one obtains the same forward propagation equations but with a different variance along the diagonal of the covariance matrix,

$$\bar{\mathbf{x}}^\ell = \phi^\ell(\mathbf{h}^\ell), \qquad \mathbf{h}^\ell = \frac{1}{\sqrt{N^{\ell-1}}}\, M^\ell \bar{\mathbf{x}}^{\ell-1} + b^\ell + \sqrt{\Sigma^\ell_{MF}(\bar{\mathbf{x}}^{\ell-1})}\, \boldsymbol{\epsilon}^\ell, \tag{5.28}$$

where

$$\big(\Sigma^\ell_{MF}(x)\big)_{ii'} = \delta_{ii'}\, \frac{1}{N^{\ell-1}} \sum_j \big(1 - (M^\ell_{ij})^2\big) \phi^2(\mathbf{h}^{\ell-1}_j)) \tag{5.29}$$

The reason for a different variance term is that, once sampled, $\mathbf{h}^\ell$ is no longer a random variable, thus the neurons are not either.

5.3.3 Concrete surrogates

An alternative, heuristic approximation for stochastic binary neural networks also exists, that is not based on the central limit theorem, but still yielding a stochastic networl similar to the perturbed surrogate. The idea was originally developed simultaneously by [89] and [90], and is referred to as a "concrete" (continuous-discrete) approximation, or the "Gumbel-softmax" approximation. In the binary case, the approximation starts by rewriting a Bernoulli random variable $\mathbf{S}_i \in \{\pm 1\}$ with $\mathbf{S}_i \sim p(\mathbf{S}_i; \theta_i)$ as,

$$\mathbf{S}_i = \text{sign}(\theta_i + \mathbf{L}_i) \tag{5.30}$$

where $\mathbf{L}_i$ is typically a logistic random variable. This is a well known formulation in statistics, referred to as the latent variable model for logistic regression. If $\mathbf{L}_i$ is a Guassian random variable, this would correspond to a latent variable model for probit regression.

In order to obtain a continuous approximation is to then approximate the $\text{sign}(\cdot)$ function with a smooth sigmoidal-type function, such as $\tanh(\cdot)$. In the machine learning literature replacing the discontinuous function by the smoother alternative is sometimes referred to as a "relaxation".

Both [89] and [90] introduce a *temperature* parameter λ to control the relaxation,

$$\mathbf{s}_i \approx \tanh(\frac{1}{\lambda}(\theta_i + \mathbf{L}_i)) \tag{5.31}$$

so that when $\lambda \to 0$, the approximation becomes exact.

An extension introduced in [91] is based on the realisation that as $\lambda \to \infty$ the approximation tends to 0. Instead, one might want the approximation to tend to the mean of $\mathbf{s}_i$, that is $M_i = \tanh(\theta_i)$. One possible parameterisation given by [91] is

$$\mathbf{S}_i \approx \tanh(\frac{1}{\lambda} \frac{\lambda^2 + \lambda + 1}{\lambda + 1} \theta_i + \frac{1}{\lambda} \mathbf{L}_i) \tag{5.32}$$

by inspection this satisfies the limit that as $\lambda \to \infty$ the approximation approaches M_i. This second parameterisation (5.32), is termed *mean concrete*, whereas the previous parameterisation (5.31) is referred to as *naive concrete*.

The application of the concrete approximation to multilayer networks is straightforward. The forward propagation equations can be written for each neuron, in the naive concrete case, as

$$\mathbf{x}_i^\ell = \phi^\ell(\mathbf{h}_i^\ell), \quad \mathbf{h}_i^\ell = \sum_j \tanh(\theta_{ij}^\ell + \frac{1}{\lambda}\mathbf{L}_{ij}^\ell)\phi(h^{\ell-1} + \frac{1}{\lambda}\mathbf{L}_j^\ell) + b_i^\ell \tag{5.33}$$

Perturbed means as a simplifying picture

One might be interested in asking if there is any relationship between the Gaussian and the new concrete approximations, since both take discrete random variables into a continuous space. To this end, let's consider the mean concrete approximation in a high temperature region $\lambda >> 1$ and expand $\tanh(\cdot)$ to first order in small $\frac{1}{\lambda}$,

$$\tanh(\frac{1}{\lambda}\frac{\lambda^2+\lambda+1}{\lambda+1}\theta_i + \frac{1}{\lambda}\mathbf{L}_i) \approx \tanh(\theta_i + \frac{1}{\lambda}\mathbf{L}_i) \approx \tanh(\theta_i) + (1-\tanh^2(\theta_i))\frac{1}{\lambda}\mathbf{L}_i \tag{5.34}$$

This reveals that for high temperature, the mean concrete approximation replaces each weight with its mean plus some perturbation $\frac{1}{\lambda}\mathbf{L}_i$ that is scaled by the variance of the weight, since $\mathbb{V}(s_i) = 1 - \tanh^2(\theta_i)$.

It is clarifying to write down side by side the Gaussian and the high-temperature mean concrete approximations to the input fields $h = \sum_i \mathbf{s}_i x_i$,

$$\text{(Gaussian)} \quad \mathbf{h} \approx \sum_i M_i \mathbf{x}_i + \boldsymbol{\epsilon}\sqrt{\textstyle\sum_i (1-m_i^2)\mathbf{x}_i^2} \tag{5.35}$$

$$\text{(Mean Concrete)} \quad \mathbf{h} \approx \sum_i M_i \mathbf{x}_i + \sum_i (1-m_i^2)\mathbf{x}_i \frac{1}{\lambda}\mathbf{L}_i \tag{5.36}$$

So it can be clearly seen that the mean-concrete approximation, at high temperature, can be written as a mean field $\bar{h} = \sum_i M_i x_i$ perturbed by some random quantity related to the variance of the underlying weights, similar to the Gaussian CLT based approximation.

Following this line of thinking, that these continuous approximations to stochastic binary weights essentially perturb a mean input field, can lead one to the work around "noisy" training of neural networks. Some of the early work considered the effect of noise added to the weights [92] or inputs x_i [93] and proceeded to expand the cost function in the perturbation.

A deeper investigation of the relationship between the two approximations is left for future work, however some comments can be made here on some differences. Clearly, in the Gaussian case there is no explicit temperature λ. One could say its temperature is "baked in", since it is deterministically given by the variance of the underlying Bernoulli weights. As a consequence, the perturbations to each weight in the Gaussian case are all perfectly correlated, whereas in the concrete case, the perturbations are independent. A second difference between the approximations is that the concrete algorithms have perturbations in the natural parameter space, so that θ is perturbed. The Gaussian case has perturbations of the means M_i instead.

5.4 Chapter conclusion

This chapter has presented approximations to derive continuous surrogate neural networks from the principled expected cost objective function, with expectation in terms of stochastic weights and neurons. Based on this expectation, and the layerwise processing of neural networks, it is possible to write the expectation as a series of nested conditional expectations, similar to a Markov chain, with layers corresponding to time indices.

From the Markov chain representation, it is possible to derive any of the more principled surrogates existing in the literature. It was also possible to derive new surrogates, in particular the Gaussian based "perturbed" surrogate. This involved Monte Carlo sampling of a Gaussian distribution assumed under the central limit theorem. A surrogate based on a non-Gaussian approximation known as the "concrete" estimator was also derived.

Apart from the new surrogates presented, the contributions of this chapter are significant since they result in the organisation of existing approximations from the literature. A discussion of qualitative similarities and differences between the approximations was offered in the final sections of the chapter as well. This provides a more coherent view of the algorithms currently being deployed, which should inform both future theoretical and practical work.

Chapter 6

A statistical physics description of machine learning

The theoretical contributions of Chapters 7 and 8, concerning multi-layer binary networks and their optimisation algorithms, have been inspired by recent theoretical work on continuous networks rooted in the statistical physics literature. This chapter aims to introduce someone familiar with statistics or optimisation to the basic concepts of statistical physics that may be relevant to the analysis of algorithms and learning systems. The utility of this Chapter is that it enables and encourages communication of ideas between different disciplines.

The long established analogy between physical and learning systems is discussed using the complementary examples of simple magnetic systems and the simplest single-layer neural network, known as the perceptron. This chapter considers perceptrons with both continuous and binary weights. The binary perceptron is of course the basic 'building block' from which a binary neural network may be constructed. As it turns out, simple as the binary perceptron may be to define, it exhibits a rich phenomenology when considered as a physical system. Therefore it motivates an introduction to the concepts of equilibrium states, phase transitions and non-equilibrium dynamics, which can provide useful tools and language for understanding the behaviour of algorithms. The progression toward these increasingly advanced concepts of statistical physics are part of an underlying trend in the study of complex learning systems.

This chapter is not crucial for the reading of the Chapters 7 and 8, but provides a reader with an introduction to their background literature and terminology. This includes the so-called dynamic mean field theory, as well as the more recent theoretical advances on the binary perceptron, both of which originally developed in the field of disordered systems. The chapter concludes with a discussion of the meeting of these two ideas in this thesis, and future prospects. In summary, this chapter achieves the following:

- Describes the established analogy between physical systems and learning systems, studied through the lens of statistical physics
- Introduces the central ideas of statistical physics, including the notion of the equilibrium state of a system, which has helped to characterise the solution space for single layer neural networks, with both continuous and binary weights
- Argues that the equilibrium description serves as a good *reference* distribution by which to describe the typical solutions found by algorithms

- Reviews recent literature that suggests successful binary network algorithms find solutions that are atypical according to the equilibrium description, suggesting the algorithm dynamics are out-of-equilibrium
- Introduces and defines language that will be used in subsequent chapters, such as typicality, self-averaging behaviour, mean field theory, dynamic and static random variables, and disordered systems
- Recounts the recent results describing the geometry of solutions of the binary perceptron, and the dynamics of multi-layer continuous neural networks

6.1 Introduction

In the past two decades, continuous multi-layer neural networks have emerged as models that are able to achieve high performance on tasks, such as image classification, that were previously considered very difficult. As discussed in the preceding chapters, the basic algorithm and workhorse of these models is stochastic gradient descent. This algorithm adapts the parameters or weights of a neural network by taking small steps in directions which locally minimise a cost function of the training data and parameters. It is not yet understood why the gradient based algorithms, coupled with a range of heuristic design choices, are so successful. The central mystery appears to be that the networks are able to overfit random data, but when applied to real datasets, they do not overfit. This is also true of the algorithms for multi-layer binary networks, where further approximations and heuristics are required, and yet performance is still maintained.

The simplest neural network is the continuous weight perceptron, closely related to logistic regression in statistics. The essential task a perceptron performs is to define a linear separating plane which successfully separates M input vectors $\{x^\mu\}_{\mu=1}^M$ of dimension N, into their respective classes $y^\mu \in \pm 1$. The perceptron and the various algorithms used to train it are well understood theoretically. The binary weight counterpart however remains a challenge for practitioners and theoreticians alike.

Arguably, the underlying difficulty for practitioners is that the binary perceptron is an NP-hard combinatorial optimisation problem. This means that in the worst case the number of elementary computational operations needed to find a solution is expected to grow exponentially with the dimension of the problem N. This corresponds to performing an extensive check over all possible assignments for the weights, starting from some random initial state. Practically, this means a Monte Carlo algorithm such as simulated annealing is ineffective for this problem.

In the last 10 years however, several heuristically modified Bayesian and non-Bayesian algorithms have been found to solve large instances of the problem in polynomial time. This suggests that the worst case picture of computational hardness is too pessimistic. Instead, physicists have attempted to consider how algorithms *typically* behave. From this perspective, the theoretical understanding of the dynamical behaviour of these heuristic algorithms has seen significant progress.

The starting point for the use of statistical physics in understanding these algorithms arises from not considering any particular realisation of the data and algorithm, but instead considering an *ensemble*. A statistical ensemble of systems is composed of many systems, that are all constructed alike. Each element of an ensemble is a replica of the system of interest that is in one of the states that is accessible to it. For a large system, the fluctuations due to

particular realisations of the data and other stochastic elements of the algorithm are 'averaged out', similarly to the central limit theorem, from which typical behaviours emerge.

6.2 What is statistical physics?

Statistical physics is a body of knowledge that deals with large systems of elements (such as particles, magnets or molecules) that interact according to simple microscopic laws. For large enough systems, from the complex microscopic dynamics there often emerges universal, deterministic macroscopic behaviour. The philosophy of statistical mechanics is guided by this observation. Any ambition to solve such systems at microscopic level is abandoned, since the complexity of the system makes it too difficult. Motivated by the complexity, a probabilistic description of the microscopic dynamics is proposed (reasonably so, if not rigorously) and from these dynamics it is possible to calculate laws describing some suitably chosen macroscopic variables.

The experience of the last 150 years serves as a guide to choosing the correct macroscopic variables and the relevant mathematical subtleties and irrelevant ones; indeed, in moving from microscopic to macroscopic laws, physicists often consider statistical mechanics as a set of clever ways to do the bookkeeping of probabilities. In a later section probability theory, in particular the theory of large deviations, will be used to define or derive the basic concepts and quantities of statistical mechanics from first principles, thereby making the bookkeeping precise.

A system has microstates $\omega = (\omega_1, \ldots, \omega_N)$ with ω_i describing the state of the ith particle, with some state dynamics assumed. Modern statistical mechanics has evolved to distinguish two types of systems, equilibrium and non-equilibrium. The two systems, and the corresponding theories, can be broadly distinguished by whether time plays a role. The equilibrium picture describes static random variables, meaning there is no dependence on time, generally because it is assumed that transient effects have played out, after waiting a long time. The non-equilibrium picture describes dynamic random variables whose properties, either macroscopic or microscopic, depend explicitly on time, though exceptions exist.

The algorithmic setting might seem to be quite far removed from real world systems of interacting particles. In the analogy for learning systems, the parameters or weights correspond to particles, the interactions are defined by practitioners (for example via an algorithm), and the macroscopic variables of interest could be the generalisation error of a class of algorithms, as compared to the average energy or pressure of a physical system.

Interactions between weights can be introduced in two ways. The first is by defining an algorithm or a rule for updating weights to 'solve' the learning problem. This corresponds to directly defining the dynamics of the particles in time. A second way to introduce interactions leaves the dynamics unspecified, instead defining a set of constraints corresponding to the learning problem, typically encoded via a cost function. This cost function is interpreted as the energy function of a system in thermodynamic equilibrium. In this case, the dynamics are not specified, since various dynamics can lead to equilibrium.

This second formulation has the advantage that it allows the learning problem to be studied with some form of generality, not having to specify an algorithm. As will be shown however, successful algorithms for some of the more difficult binary systems may not correspond to dynamics that agree with an equilibrium picture. Interestingly, this formulation introduces ideas from the field of disordered systems, where particles are 'frustrated' due to competing and contradictory interactions which are random. In the learning problem, the constraints are the interactions and the data playing the role of disorder, which will be explained in detail. A real world example

of a disordered system is a glass. The fascinating properties of glassy materials is that they are technically relaxing towards their equilibrium state but get stuck in metastable states for extremely long periods of time. For the algorithmic picture, disordered systems and all the tools and techniques developed, provides a window into the out of equilibrium phenomena of learning systems, from the viewpoint of equilibrium thermodynamics, where analytical calculation of macroscopic quantities, such as generalisation error, are possible.

For either type of system statistical mechanics can explain the existence of phases, based on the macroscopic variables chosen. For example water can be found in the common phases of matter (solid, liquid and gas). Based on an appropriate choice of macroscopic variables, precise predictions about phase transitions can be made. For example, given atmospheric pressure it is possible to calculate for water the temperature at which an abrupt transition occurs as liquid turns to gas. Or, for example, it is possible to calculate the temperature at which a continuous transition occurs for a ferromagnetic material with two states $+1$ and -1, where despite having only local interactions, the system fluctuates on all length scales between positive and negative magnetic states. The case of continuous phase transitions is referred to as the subject of critical phenomena, and holds a special place in statistical mechanics, partially because of the theoretical tools which deal with it, but also because of the universality of the behaviour: many of the properties observed are system independent. Indeed, the analogy of phase transitions can apply to many learning systems, in particular those which have a correspondence to disordered systems.

6.3 Equilibrium and non-equilibrium systems

Equilibrium statistical mechanics considers thermodynamic ensembles over the space of all possible microstates $P(\omega)$ that are interpreted as the stationary distribution of the microscopic dynamics. Associated with these ensembles are equilibrium states that are stable against small perturbations and can be described by having recourse to a few macroscopic or "coarse-grained" variables called macrostates. A macrostate is a function $M_N(\omega)$ of the microstates. If $P(\omega)$ is a valid thermodynamic ensemble, then the distribution of the macrostate $P(M_N)$ becomes highly concentrated around its most probable or typical values as N increases. These values are called the equilibrium states of M_N.

This limiting behaviour is described precisely by the theory of large deviations [94], which generalises the Central Limit Theorem and the Law of Large Numbers. A distribution is said to satisfy a large deviation principle if the limit

$$\lim_{N\to\infty} -\frac{1}{N}\log P(M_N = m) = I(a) \tag{6.1}$$

exists. The decay function defined by this limit is called the rate function, and the factor N is the speed of the large deviation principle (for some distributions one may need to divide by N^2, or some other function of N). The limit as N goes to infinity is called the thermodynamic limit. The large deviation principle essentially says that the distribution decays to zero with N, so that approximately,

$$P(M_N = m) \approx e^{NI(a)} \tag{6.2}$$

The theory of large deviations can be used to describe the concentration of measure of distributions in equilibrium systems, where it can be shown the rate function $I(a)$ is related (via a

Legendre transform) to the free energy of the system. The equilibrium picture, including the free energy, is examined in the next section.

The non-equilibrium case considers systems that can be modelled by Markov processes, as follows. Denote the state of the system at time t by ω^t. Then for a trajectory $\{\omega^{0:T}\}$ with assumed discrete time steps $\{0, 1, 2, \ldots, T\}$, the distribution over the trajectory factorises as

$$P(\omega^{0:T}) = P(\omega^T|\omega^{T-1})P(\omega^{T-1}|\omega^{T-2})...P(\omega^1|\omega^0)P(\omega^0) \tag{6.3}$$

the dependence of the next state on only the previous state is known as the Markov property. Large deviations play a central role in non-equilibrium systems, for a discussion see [95].

Equilibrium and non-equilibrium systems can be distinguished by considering trajectories of the systems in time. A necessary condition for a system to be an equilibrium system is that its dynamics are time-reversible. This means that for a given trajectory $\omega^{0:t} = \{\omega^0, \omega^1, \ldots, \omega^t\}$ then the time-reversed trajectory $\omega^{t:0}$ is equally likely under $P(\omega)$. An equivalent mathematical definition of reversibility is that the dynamics satisfy the conditions of detailed balance. What this property suggests is that there is no preferred direction between the trajectory and its time-reversed version.

A system which does not satisfy this condition is a non-equilibrium system[1]. However, to say that a system *is an equilibrium system* is different to saying a system *in a state of equilibrium*. Examples of equilibrium systems that can be out of equilibrium include the relaxation (fast or slow) toward the stationary distribution from some initial state, or perturbations about equilibrium. Glasses, which satisfy detailed balance and have equilibrium distributions, nevertheless exhibit a phenomena termed *aging*. This means that the approach to equilibrium becomes slower as time increases.

6.4 Equilibrium formulation of learning problems

As mentioned, it is possible to study learning systems without specifying an algorithm. This is useful, as one can study a system with some degree of generality, provided the learning problem can be written as an optimisation problem of some cost function. In this case, it is perfectly acceptable to interpret the cost function as the energy function, or Hamiltonian, of a system in thermodynamic equilibrium. Note that any arbitrary algorithm dynamics which minimise a cost function do not necessarily have a stationary distribution that is the same as the equilibrium distribution corresponding to that cost function[2].

Consider now the following formulation of the perceptron problem as a system in thermodynamic equilibrium. This formulation holds for binary or continuous weights. Following this walk through, a discussion will be presented, about how to calculate answers to questions about macroscopic variables, such as the generalisation error, or whether solutions exist for a random instance of the problem. As a necessity, this discussion will lead directly to the field of disordered systems.

Assume a finite data set made of pairs, sampled i.i.d, $D = \{x^m, y^m\}_{m=1}^M$ with labels $y \in \{+1, -1\}$ and inputs $x^m \in \mathbb{R}^N$ vector valued in general. The standard perceptron model has continuous parameters which are referred to as weights w_i, and computes for each data point m

[1] Non-equilibrium systems may have stationary distributions but these are generally much more complicated than those for equilibrium systems

[2] In fact, certain dynamics may not have stationary distributions at all.

the linear combination

$$h^m = \sum_i w_i x_i \tag{6.4}$$

Algorithms for the perceptron attempt to adapt the weights such that each input is labelled correctly, so that $y^m h^m > 0$ for all $m \in D$. To this end, a cost is assigned for the data set, a possible choice is a simple count of the number of errors

$$E_D(w) = \sum_{m=1}^{M} \Theta(-y^m \sum_i w_i x_i) \tag{6.5}$$

where $\Theta()$ is the Heaviside step function. This is the $0-1$ loss, a sensible choice and the first loss encountered in any decision theory course (Bayesian and non-Bayesian).

There exist several possible thermodynamic ensembles in statistical mechanics. One particular ensemble, which provides powerful analytic techniques, is known as the canonical ensemble. This is a heuristic motivated by considering a system whose energy fluctuates but its average is constrained by being in contact with a system so large that its temperature remains constant [3]. The canonical ensemble and the useful expressions for various quantities it provides, can be codified by the minimum free energy principle. In the canonical setting, this free energy is known as the Helmholtz free energy, which takes the form

$$G(p) = -\sum_w p(w)E(w) + \frac{1}{\beta}\sum_w p(w)\log p(w) \tag{6.6}$$

and which is to be minimised over the space of distributions p. One can see that the Helmholtz free energy presents a trade off between the energy of the system, and the entropy. The Boltzmann-Gibbs distribution is the unique minimum in this space of distributions and has the following form,

$$p_\beta(w) = \frac{1}{Z(\beta; \{x_i, y_i\}_i)} \exp(-\beta E_D(w)) p_0(w) \tag{6.7}$$

This distribution should be familiar to practitioners who have encountered the simulated annealing algorithm for optimisation, with lower energy regions of weight space or 'states' having greater probabilistic weight. The partition function $Z(\beta; \{x_i, y_i\}_i)$ depends on both the inverse 'temperature' β and the realisation of the training data. Note that including β controls the roughness of the distribution, allowing for interpolation between $\beta = 0$, corresponding to a uniform distribution, and $\beta = \infty$ a distribution concentrated on the regions of weight space corresponding to zero errors, or 'ground states' in statistical physics terminology.

A connection to Bayesian statistics

It is possible to make a connection here to Bayesian statistics, since there is a sense in which $\beta = 0$ corresponds to having only prior information since the Boltzmann-Gibbs distribution

[3] The approach is heuristic in that it is not derived from a more fundamental argument regarding microscopic dynamics, but its description of standard quantities agree with more fundamental ensembles in the large system limit, at least under certain conditions [94]

reduces to the prior, and for $\beta = \infty$ the regions of zero error is reminiscent of the regions of maximum likelihood. Given a model of the likelihood of the data $p(y|x, w)$, one can rewrite posterior,

$$p_\beta(w|\{y, x\}) = \frac{1}{Z} p^\beta(y|x, w) p_0(w) = \frac{1}{Z} \exp(-\beta(-\log p(y|x, w))) p_0(w) \tag{6.8}$$

The perceptron as written before in Equation 6.5 is not a posterior with any recognisable likelihood, thus the connection to Bayesian logistic regression is not clean. However, what is important is to recognise that either the posterior with known likelihood, or the Boltzmann-Gibbs distribution defined in Equation 6.7 are simply assigning probability mass over the solution space.

This might leave the reader wondering whether any assignment is arbitrary, or which might be more appropriate. The first reason that so much attention has been paid to the $0-1$ loss, or Heaviside function, is that in the low temperature limit when all mass is put on solutions, it can determine whether solutions exist at all for typical instances of certain learning problems. As discussed in chapter 4, concerning particularly Empirical Risk Minimisation, for classifiers trained to minimise the $0-1$ loss, the optimal classifier minimises the probability of error. For these reasons, the $0-1$ loss is a good choice for a *reference* Boltzmann distribution, by which one can judge the effects of other costs and other algorithmic design choices.

6.4.1 Macroscopic variables from thermodynamic potentials

Expressions for macroscopic variables of interest can be obtained from thermodynamic potentials. Thermodynamic potentials are functions of the inverse temperature β, and often other parameters that define the energy function. Properties of the Boltzmann distribution can be summarised through the potentials, usually by taking derivatives. An important potential is the canonical free energy of a system

$$F(\beta) = -\frac{1}{\beta} \log Z(\beta) \tag{6.9}$$

From this, two more thermodynamic potentials can be found, the average energy $\langle E(w) \rangle$ and canonical entropy $S(\beta)$, by taking derivatives of the $F(\beta)$,

$$\langle E(w) \rangle = \frac{\partial\big(\beta F(\beta)\big)}{\partial \beta}, \qquad S(\beta) = \beta^2 \frac{\partial^2 F(\beta)}{\partial \beta^2} \tag{6.10}$$

where the expectation with respect to the Boltzmann distribution is denoted using angular brackets, $\langle A(w) \rangle = \sum_w A(w) p(w)$.

6.5 Mean field theory

The idea that matter exists in phases has been touched on already, such as the well known phases that water can take, depending on the temperature, pressure and other external conditions. The macroscopic properties of each phase differ widely due to these conditions. In order to investigate such sharp changes in the states of materials, physicists study simple models of interacting many-body systems. The Ising model is one of the simplest of such systems, and represents magnet spins on a regular lattice. While it is not intended to explain the phase transition of water,

the arguments developed contain the standard theory to describe the essential features of phase transitions.

After defining the Ising model, it is possible to introduce mean field theory. Mean field theory is an approximation that allows for simple, analytic calculations of averages that are intractable under the Boltzmann distribution of the system. As a consequence, analytic expressions are obtained for the free energy of the system in terms of what physicists refer to as *order parameters*. These parameters represent, as the name suggest, an ordering property of the system. In the Ising magnet case, this is the average magnetisation. As discussed, one of the principles of statistical mechanics is that the free energy of the system is minimised in thermal equilibrium. This holds equally for a mean field model as well. Furthermore, when expressed in terms of order parameters, the free energy can predict whether a model undergoes a phase transition. Whether the mean field model accurately predicts the properties of the original system being approximated is an important question, which is discuss.

Learning systems, when formulated as systems in thermodynamic equilibrium, are more complex than the simple Ising model. As seen in the perceptron example, there is a source of randomness in the interactions, generated by the data. This randomness allows for the analogy between learning systems in thermal equilibrium and the mean field theory of disordered systems. The Ising model with disorder is known as a spin glass, where the couplings J_{ij} are random variables, for example they could be Gaussian distributed.

Along with the disorder of the interaction, another fundamental feature of spin glasses is that of *frustration*. This is a situation in which the structure of the couplings of a particular realisation of the system are such that it is impossible to satisfy all constraints simultaneously. A simple example is of three spins, with pairwise couplings, and the sign of the product of the couplings is negative. By careful consideration, one finds that there will be no unique ground state with all couplings satisfied. The mean field theories of disordered systems seek to explain the properties of such systems, where the free energy has many valleys, corresponding to metastable states.

6.5.1 The Ising model

The Ising model considers a set of sites indexed by the integers from 1 to N. A variable S_i is assigned to each site i, and the Ising spin corresponds to a binary value $S_i = \pm 1$. It can be helpful to consider the problem of magnetism, especially throughout the remainder of this chapter, in which S_i represents whether the microscopic magnetic moment is pointing up or down.

The interaction between two sites (ij), or the bond, is denoted as J_{ij}. In the Ising model sites interact only with their nearest neighbours on a lattice, a set denoted as $\mathcal{N}(i)$ for site i. The interaction is uniform for all pairs, $J_{ij} = J$, meaning the interaction is uniform across the lattice and symmetric between any two pairs. The energy of an interaction is simply $-JS_iS_j$. So, in the case that $S_i = S_j$ the energy is $-J$, and is J otherwise. Thus, if $J > 0$ the aligned case is more stable than the anti-aligned case, since it has lower energy. Once again, in the magnetism problem the alignment corresponds to the spins being up or down simultaneously. This positive interaction is called a ferromagnetic interaction, and can lead to macroscopic magnetism (ie. ferromagnetism).

The total energy function of the system includes all of these interaction terms, as well as self interaction terms $-hS_i$, for some external "field" h. This allows one to write down the energy

function, or Hamiltonian, as,

$$H = -J \sum_{(ij)\in B} S_i S_j - h \sum_{i=1}^{N} S_i \tag{6.11}$$

where the sum is over all bonds, denoted by the set B. The postulates of statistical mechanics suggest to write down the Boltzmann distribution for the Hamiltonian,

$$P_\beta(\mathcal{S}) = \frac{1}{Z(\beta)} e^{-\beta H} \tag{6.12}$$

where $\mathcal{S} = \{S_i\}$ denotes the set of all spin states, or configurations, and $e^{-\beta H}$ is the Boltzmann factor.

In order to get some intuition for the model, it can help to consider a model of dynamics. A model of dynamics which satisfies detailed balance and attains asymptotically the Boltzmann distribution are known as Glauber dynamics. The dynamics are Markov, for those readers familiar with random processes, and operate by sequential updates of the spins, randomly across the lattice, using conditional probabilities to flip the spins. A computer simulation of an Ising model with Glauber dynamics would proceed according to the following algorithm:

1. Initialise all spins s_i randomly. Then, repeat the following two steps:
2. Choose site j at random. Calculate the field at site j:

$$h_j = J \sum_{i\in\mathcal{N}(j)} s_j S_i - h s_j \tag{6.13}$$

3. Update site j from $s_j \to -s_j$ with probability:

$$p(s_j \to -s_j) = \frac{1}{2}(1 + s_j \tanh(\beta h_j)) \tag{6.14}$$

The form of this probability is derived from the Boltzmann factor. Implemented in this way on a computer, Glauber dynamics allow a practitioner to (asymptotically) sample from the Boltzmann distribution. This sampling procedure is known as Gibbs sampling or heat bath sampling, in computer science.

From the Boltzmann distribution it is possible to compute the expectations of any physical quantity. One such expectation is the magnetisation of the system, which characterises the macroscopic properties of the Ising model,

$$m = \frac{1}{N} \left\langle \sum_{i=1}^{N} S_i \right\rangle_{P_\beta(\mathcal{S}).} \tag{6.15}$$

The magnetisation measures the total ordering of the system (ignoring fluctuations over time, as is appropriate in a system that is in equilibrium). In this context, the magnetisation is an example of what physicists refer to as an order parameter, which are variables that indicate whether a system is in an ordered state. In the ferromagnetic case, an ordered state corresponds to the spins aligning, whereas if there are equal numbers of aligned and anti-aligned spins, there is an absence of any order.

Once again, it is usually very difficult to carry out the sum over 2^N terms appearing in the partition function,

$$Z(\beta) = \sum_{S_1=\pm 1} \sum_{S_2=\pm 1} ... \sum_{S_N=\pm 1} e^{-\beta H} = \sum_{\mathcal{S}} e^{-\beta H}. \tag{6.16}$$

Therefore, one must resort to approximations in order to calculate expectations. A simple approximation widely used is mean field theory (MFT).

6.5.2 Mean field theory of the Ising model

A mean field theory aims to replace all the interactions or the "field", felt by a body at one site, with an effective or average field. Effective field theories begin by writing the fields in terms of order parameters and their fluctuations, which in the Ising model is taken to be the magnetisation. The simplest mean field theory thus replaces the true local field with the magnetisation m, ignoring fluctuations.

A derivation of the mean field approximation to the system is now presented. This derivation is common in the physics literature, being based on a local perturbative expansion. An alternative derivation, left to the appendix, reveals the global nature of the approximation. In physicists language it is a (global) variational approach that considers a Gibbs free energy. It should be familiar, having been touched on in the variational Bayesian approximations in [Section]. Thus, readers with experience in statistical inference may feel more comfortable with this approach, since the Gibbs free energy can be identified as a Kullback-Liebler divergence between the true distribution and a mean field approximation.

The perturbative approach begins by writing the spin at a site as the magnetisation plus some local fluctuation, $S_i = m + \delta S_i$, where by definition $\delta S_i := S_i - m$. From this the effective Hamiltonian can be derived,

$$H = -J \sum_{(ij)\in B} (m + \delta S_i)(m + \delta S_j) - h \sum_{i=1}^{N} S_i \tag{6.17}$$

$$= -Jm^2 N_B - Jm \sum_{(ij)} (\delta S_i + \delta S_j) - h \sum_{i=1}^{N} S_i \tag{6.18}$$

where the magnitude of the cross terms $\delta S_i \delta S_j$ are assumed to be negligibly small (also note that any correlations between spins are ignored in this approach). The above expression can be simplified to give the mean field Hamiltonian,

$$H_{MF} = -Jm^2 N_B - Jmz \sum_i \delta S_i - h \sum_i S_i \tag{6.19}$$

$$= -Jm^2 N_B - (Jmz \sum_i + h) \sum_i S_i \tag{6.20}$$

where the sum is over sites instead of pairs, identifying z as the number of bonds emanating from a site. The mean field Hamiltonian can be used to define a different Boltzmann type distribution, but of a system of non-interacting particles. What the approximation buys, first of all, is the ability to calculate expectations more readily. The partition function is easily computed, for

example,

$$Z_{MF}(\beta) = \sum_{\mathcal{S}} e^{-\beta H_{MF}} = e^{-\beta N_B J m^2} \{2\cosh(\beta[Jmz + h])\}^N \tag{6.21}$$

Of particular interest is the local magnetisation of a site S_i under this new mean field Boltzmann distribution. Since one would like the original system and its mean field approximation to be consistent at this level, the local average of the spin is equated to the global average magnetisation,

$$m = \sum_{\mathcal{S}} S_i \frac{e^{-\beta H_{MF}}}{Z_{MF}} = \tanh(\beta[Jmz + h])\} \tag{6.22}$$

This is known as the equation of state, or the Callen equation. This equation can be solved graphically to obtain a solution for m, which depends of course on the inverse temperature β. The equations of state plotted in Figure 6.1 suggest the presence of a phase transition in the model, as the inverse temperature is varied. After some point β_c, the magnetisation moves away from zero, and the system spontaneously magnetises.

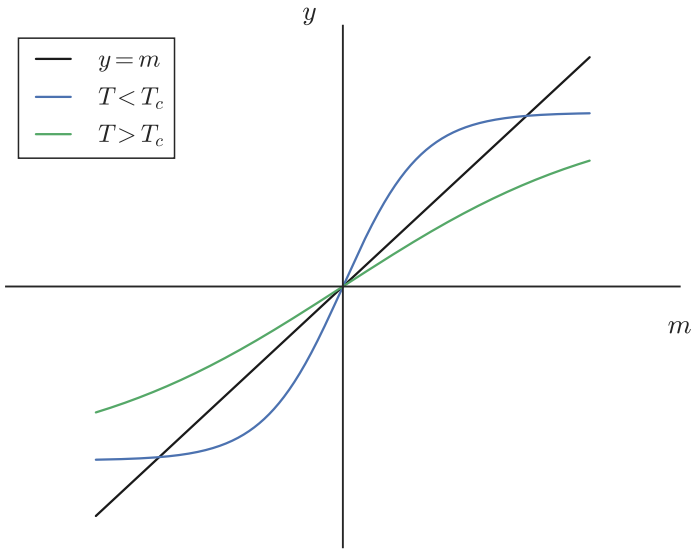


Figure 6.1: Graphical plot of the Callen equation for different temperatures around the critical temperature T_c. If $T > T_c$, the only solution is $m = 0$. If $T < T_c$, there is a non-zero solution.

For systems where the bonds are not uniform across the lattice, a set of mean field equations are obtained. In this general case the approximation will have a mean for each site m_i, and the equations are often solved by iteration until all are consistent. For this reason, mean field theory is often referred to as a self-consistent field theory.

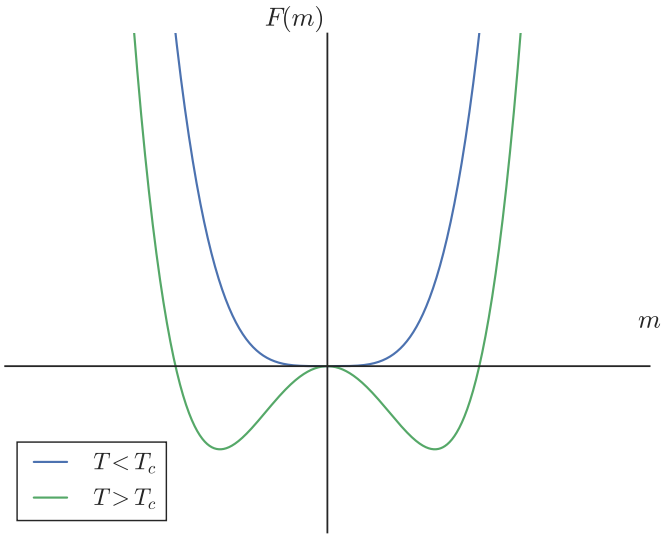


Figure 6.2: The mean field free energy of the Ising model, as a function of the order parameter, the average magnetisation m, for different temperatures T. Below the critical temperature T_c, the system will spontaneously magnetise into either a positive or negative spin state.

The mean field free energy about the transition temperature is illustrated in Figure 6.2. Thus, the mean field approximation provides another example of the utility of free energy functions in describing the behaviour of a system. The free energy analytically reveals the change in the order parameters as the temperature is changed, which is referred to as a control parameter. Specifically, as written by Nishimori [96]:

" *The coefficient of* m^2 *changes sign at* β_c*. The minima of the free energy are located at* $m \neq 0$ *when* $\beta > \beta_c$*, and at* $m = 0$ *if* $\beta < \beta_c$*. The statistical-mechanical average of a physical quantity obtained from the Boltzmann distribution corresponds to its value at the state that minimises the free energy (the thermal equilibrium state). Thus, the magnetisation in thermal equilibrium vanishes above the critical temperature, and is non-vanishing below it.*"

A reader may wonder how accurate the predictions from mean field theory are for the Ising model. The answer is that it correctly predicts the existence of phase transitions, but does not accurately predict the critical temperature $\beta_c = \frac{1}{T_c}$.

However, there do exist models where mean field theory is exact. An important example is known as the Curie-Weiss model, a version of the Ising model where the interactions have infinite-range. The mean field free energy agrees in this case with the analytic solution for the free energy [67]. In the more difficult analysis of disordered systems, such as spin glasses, physicists have thus turned to infinite range models with the expectation that mean field theory will also be exact for these more complex magnetic systems. Hence, the Curie-Weiss model is an important model in this regard.

6.6 Disordered systems

Systems such as the Ising model have various levels of order, and corresponding disorder, in the spins of the system. If the temperature of the system is high, all the spins flip close to independently, and the order parameter, in this case the magnetisation, is zero. The field of disordered systems [67] introduces a different notion of disorder; one of randomness in the interactions between particles.

In physical materials, disorder can occur by diluting a magnetic material such as Manganese into a noble metal such as Copper, thus spacing the magnetic atoms at random distances. The introduction of randomness via the geometry is known as structural disorder. This chapter focused on a simpler type of system known as a spin glass. These materials are more like the Ising model than the structural glasses, but have their disorder directly introduced into the couplings J_{ij}. For example, distributed according to a zero mean Gaussian.

Interesting behaviour can occur in such materials depending on how the time scale of a fluctuation of the disordered interaction, τ_{dis} compares to the timescale of the experimental observation τ_{exp}. In the case when the observation time is much longer $\tau_{exp} >> \tau_{dis}$, the disorder does not produce interesting behaviour. In the case of the Ising spins, similar behaviour to the standard ferromagnetic model is obtained. This is referred to as *annealed* disorder.

If however the observation time is much less than the typical time for a fluctuation of the random interactions, $\tau_{dis} >> \tau_{exp}$, then fascinating behaviour can be observed, such as glassy dynamics. This is referred to as *quenched* disorder. A spin glass is an example of a model with quenched disorder, as the couplings J_{ij} are random but fixed, while the spins are free to fluctuate.

The terms annealed and quenched are borrowed from the picture of a metal heated in a forge. Initially, one might have a piece of metal in equilibrium with the forge. In cooling it slowly, for example by reducing the heat of the forge, the temperature is being annealed. If instead the metal is plunged into water, it rapidly cools and is said to be quenched.

Quenched disorder creates *frustration* between the interactions: it becomes impossible to satisfy all the couplings simultaneously, as is possible for a ferromagnetic system. Frustration exists, formally, if for there exists any loop in the graph of connected spins for which the product of the couplings J_{ij} is negative. In such a loop, if one starts by fixing any one of the spins, and the proceeds to fix subsequent spins to satisfy the couplings, then it is guaranteed to return and flip the original spin. Thus, for any system that is not a tree, frustration will exist (with probability one).

Frustration, in its turn, is the source of the metastable states that riddle the energy landscape of spin glasses. For a low enough temperature, an Ising model with disorder will undergo a phase transition to a spin glass phase, similar to the way the Ising model undergoes a transition from a paramagnet to a ferromagnet.

Of relevance to the continuous and binary perceptron, and neural networks more generally, is a mean field theory developed for an infinite range spin glass known as the Sherrington-Kirkpatrick (SK) model,

$$H = \sum_{i<j} J_{ij} S_i S_j - h \sum_{i=1}^{N} S_i \tag{6.23}$$

where the notation $i < j$ denotes all spin pairs, and the couplings are drawn as $J_{ij} \sim \mathcal{N}(0, 1)$. The solution of this model, via mean field theory, is extremely difficult and there exist various

approaches. It has taken well over 30 years to prove the validity of the solutions rigorously. The results will be outlined qualitatively, particular those which are relevant to the dynamics of algorithms.

Although the infinite range Ising models (spin glass or not) are seen as first approximations to the short range systems that physicists are interested in, this is not the case for learning systems. The perceptron, when formulated as an equilibrium system, corresponds to infinite range, or fully connected models.

6.6.1 Static theory: averaging the free energy over the disorder

As usual, the target of any efforts to describe a system in thermodynamic equilibrium is the canonical free energy of the system. However, for disordered systems each instance of the system, corresponding to a realisation of the random couplings J_{ij}, will of course be different. Recall the fundamental motivation of statistical mechanics, where for large systems there is an expectation that typical behaviours can emerge. Under such settings, one tries to find the typical free energy function. It so happens that the free energy is a self-averaging quantity, meaning its typical and average values coincide,

$$F = -\frac{1}{\beta}\big\langle \log Z(\beta; J_{ij})\big\rangle_{P(J)} \tag{6.24}$$

where the average is over all the independently drawn couplings $J_{ij} \sim P(J)$ that define the Hamiltonian. The average of the logarithm, known as the quenched average, is difficult to calculate. A simple alternative is to average the partition function $Z(\beta; J_{ij})$,

$$F_a = -\frac{1}{\beta}\log\big\langle Z(\beta; J_{ij})\big\rangle_{P(J)} \tag{6.25}$$

Considering this expression, where the couplings and the spins are being averaged together (interchangeably), one realises this corresponds physically to the spins and couplings fluctuating on the same time scale. Hence this calculation is known as the *annealed average*. Unfortunately, the annealed computation does not give the typical values of interested, except as an approximation in the high temperature regime. The reason is that although for many functions of random variables the average and typical coincide, this is not the case for a product of independent random variables. A prominent example is the partition function,

$$Z(\beta; J_{ij}) = \sum_{\mathcal{S}} e^{-\beta H} \tag{6.26}$$

where the sum in the SK model's Hamiltonian Equation 6.23 can be written as a product, given that is in the exponential [67]. In the large N limit, the product in fact tends to a log-Normal distribution, which is heavy tailed. There exist more physical arguments for averaging the free energy and not the partition function [97], but a simple one is that in taking the logarithm of the product one obtains a sum of independent random variables. This of course know converges to a Gaussian, where the typical and average values agree.

Unfortunately, there are great difficulties in calculating the free energy, being an average of the *logarithm* of the partition function. Over the course of the last 40 years, physicists have attempted the calculation via a range of techniques, including the cavity method, TAP equations, dynamic generating functionals, or a generally non-rigorous procedure known as the

replica trick [98]. The general picture that has emerged from this vast program is complex to describe, and is summarised in various texts [98], [96], [99]. For the purposes of the current thesis, and in particular the discussion of simple learning systems, the most distinctive picture from these analyses is the behaviour of the free energy function in the thermodynamic limit of these disordered systems.

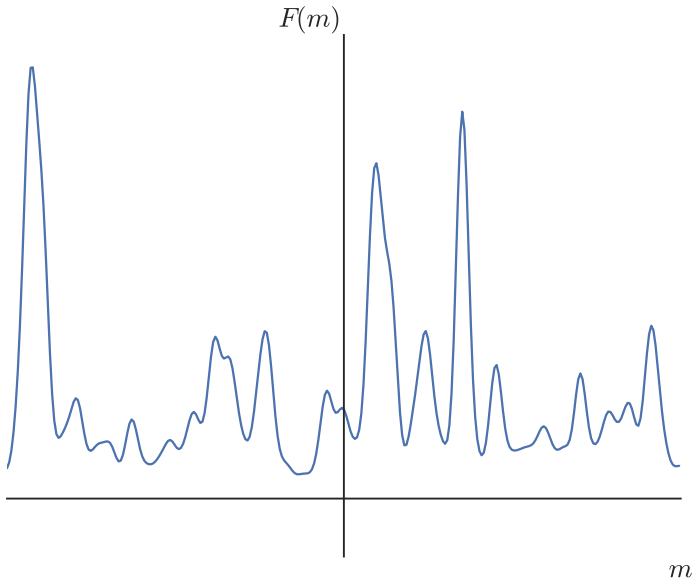


Figure 6.3: Illustration of the multi-valley structure of the free energy of the SK model, $F(M)$. In the thermodynamic limit, the energetic barriers diverge, meaning the systems will remain trapped in some subset of the configurations (sometimes termed a pure state, or ergodic component, depending on the situation).

For systems with frustration, such as the SK model, the system in the thermodynamic limit undergoes ergodicity breaking as the temperature is lowered. This corresponds to the state space of the spins breaking up into ergodic components, including both ground states and metastable states from which they cannot escape (these are collectively referred to as "pure" states [99]). The nature of the ergodicity breaking depends on the model considered. The SK model undergoes *continuous* ergodicity breaking, where each pure state continuous to break into further pure states as the temperature is lowered continuously. An illustration is provided in Figure 6.3, where a multi-valley structure of the free energy can be seen.

Other magnetic models exhibit more sudden ergodicity breaking transitions, and some none at all. The next section considers the perceptron, which can be cast as a disordered system. As described, in the continuous weight case there is no ergodicity breaking, whereas in the binary case the ergodicity breaks suddenly.

6.6.2 Equilibrium analysis of the continuous perceptron

The formulation of the perceptron problem, for both binary and continuous weights, as an equilibrium system with disorder, was first established by Gardner. A series of fundamental papers posited an equilibrium system with the weights as particles and the $0-1$ cost function defining the energy, as before. The papers considered the volume of solutions compatible with the patterns, or "version space",

$$\Omega_0(\{x^m, y^m\}_{m=1}^{\alpha N}) := \int d\mu(w) \prod_{m=1}^{\alpha N} \Theta(y^m \sum_{i=1}^{N} w_i x_i^m) \tag{6.27}$$

This is equivalent to the the partition function at zero temperature,

$$\Omega_0(\{x^m, y^m\}_{m=1}^{\alpha N}) = \lim_{\beta\to\infty} Z(\beta) = \lim_{\beta\to\infty} \int d\mu(w) e^{-\beta \sum_{m=1}^{\alpha N} \Theta(-y^m \sum_{i=1}^{N} w_i x_i^m)} \tag{6.28}$$

and is also equivalent to the canonical entropy in the zero temperature limit, $\Omega_0 = \lim_{\beta\to\infty} S(\beta)$.

The system has quenched disorder, due to the fixed data for any given instance of the problem. Thus, to study the thermodynamic picture the quenched averages must be computed. Assuming Gaussian distributed inputs x_i and random labels y_i, with the number of patterns M in constant proportion to the input dimension N, $\alpha = \frac{M}{N}$, it is possible to calculate the typical volume, or the typical free energy of the system at finite temperature. The technique used is the replica method, as for the SK model.

The results for the continuous weight perceptron are as follows. In the zero temperature limit, which studies the capacity of the problem, the critical capacity of the system is found to be $\alpha_c = 2$. This agrees with an older result by Cover using simpler geometric arguments. Furthermore, for a load $\alpha < \alpha_c$, the ground states of the free energy form a connected and convex subspace, at zero temperature and above. This means that suitable equilibrium dynamics at finite temperature, such as Langevin dynamics, will relax to the equilibrium distribution, and there is no ergodicity breaking. The finite temperature $T > 0$ case, with load greater than the capacity $\alpha > \alpha_c$, results in the 'full-RSB' effects, similar to that of the SK model. This is detailed in [100].

6.6.3 Equilibrium analysis of the binary perceptron

The thermodynamic analysis for the binary perceptron is strikingly different to that of the continuous perceptron. Following Gardner's initial work on the binary perceptron [101], which did not yield a conclusive answer for the capacity, a subsequent study [102] obtained a critical capacity of $\alpha_c = 0.833$, by the replica method. This result remains unproven, however numerical simulations support this number, as well as other non-rigorous techniques being in agreement [100].

Beyond the critical capacity, beyond which no solutions exist, the study of [102] suggested that the equilibrium system formed from the binary perceptron exhibits ergodicity breaking. More specifically, in a finite temperature analysis $T > 0$, the Boltzmann-Gibbs distribution can be written as a convex combination,

$$P(s) = \sum_{\gamma} \omega_\gamma P_\gamma(s) \tag{6.29}$$

where it is understood each $P_\gamma(s)$ is a Boltzmann-Gibbs distribution over the weights in some separated, *pure states* of the weight space denoted by γ. The statistical weight of a pure state is given by ω_γ [Spin glasses for pedestrians].

This total freezing can be contrasted to the progressive structural phase transitions that occur for random constraint satisfaction problems (CSPs). In the last 20 years, the finer details of the transitions in the solution space have emerged, with benefits to the understanding of which classes of algorithms are able to find solutions, and to the design of better algorithms. A simplified pictorial representation of the transitions is usually as given in Figure 6.4.

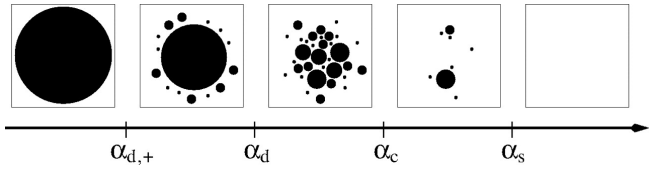


Figure 6.4: Progressive break-up of solutions for constraint satisfaction problems, with $\alpha_{dynamical} \leq \alpha_{condensation} \leq \alpha_{rigidity} \leq \alpha_{freezing} \leq \alpha_S$. The binary perceptron is exotic, in the sense that the transitions occur simultaneously, with the problem always in the frozen phase. Figure taken from [103]

Based on the typical equilibrium analysis, it has emerged that the binary perceptron, for $T = 0$, is always in a hard phase [104]. This means that $\alpha_{dynamical} = \alpha_{condensation} = \alpha_{rigidty} = \alpha_{freezing} = 0$, that is, the problem is always frozen, unlike most other CSPs. Complementary finite temperature $T > 0$ studies [105] have agreed with this picture. The analysis of [104], which studied the local geometry of solutions, furthermore showed that the solutions are separated by a Hamming distance of order N.

The product of all these analyses, and much of the experimental evidence, suggests that finding solutions reliably would be computationally very difficult, in line with the NP nature of the problem. However, the above picture only true for equilibrium algorithms, and not for those with some element of the algorithm driving them out of equilibrium.

6.7 Out-of-equilibrium algorithm dynamics

This final section describex at a high level some of the efforts of statistical physicists in building a theoretical understanding of learning processes in the binary perceptron and deep continuous neural networks. From this selective review of recent and historic progress, it is apparent that an understanding of deep binary neural network learning processes, at least using the tools of statistical physics, is in its infancy. Indeed, the contributions made in thesis provide some of the first advances toward solving this difficult problem.

6.7.1 Static picture of binary perceptron dynamics

In the last 10 years algorithms have been developed which are able to solve the binary perceptron problem effectively and efficiently. For relatively large systems, for example with input dimension $N = 2000$, heuristic algorithms have found solutions even for loads $\alpha \approx 0.80$, close to the apparent capacity of the binary perceptron [32].

The binary perceptron's equilibrium landscape of isolated solutions, separated by large Hamming distance, is at odds with these empirical results for heuristic algorithms. Therefore, the algorithms which are reliably finding solutions quickly must have have some components in their design which drives the dynamics out of equilibrium.

In order to gain some insight into the algorithm dynamics, one might ask what kind of characteristics the solutions have that these algorithms find. This is a common question posed and solved by statistical physicists [67]. For instance, one might wonder whether the solutions are typical equilibrium solutions, that is, isolated and frozen. The answer, it turns out, is no. The solutions these algorithms find instead they belong to large connected clusters of unfrozen solutions [106]. Remarkably, they are not numerically dominant, thus the equilibrium measures used in calculations under the static theory are blind to the existence of these rare but algorithmically accessible solutions.

A recent approach to describing these algorithm dynamics introduces an effective loss surface called the local entropy [106]. This surface is smoother than the original, by emphasising the statistical weight of solutions with nearby solutions. The local entropy idea allows for the comparison of various algorithm design choices in a more systematic way; it is possible to estimate a quantity that measures an algorithm's "local entropy". Naturally, those algorithms with higher local entropy are able to reach these rare, unfrozen solutions. This includes both the modified message passing algorithms [32] as well as the gradient descent based method [85], both of whose dynamics take place in an auxiliary parameter space.

6.7.2 Dynamics of deep continuous networks

Multi-layer neural networks are more complex than the single layer perceptron both in terms of the many modifications and heuristics used in practice, as well as in their analysis. As described by the authors in [107], in terms of a theoretical analysis of such learning systems,

"*A theory of deep network entails two dynamical processes. One is the dynamics of weight matrices during learning. This problem is challenging even for linear architectures and progress has been made recently on this front (see e.g. [26]). The other dynamical process is the propagation of the signal and the information it carries through the nonlinear feedforward stages*
"

Accepting this argument, then understanding the learning processes of neural networks amounts at least to an understanding of the interaction of these two dynamical processes. Several papers in recent times have made progress on this understanding.

The seminal work of [26] can largely be credited with re-opening the field of study into the second dynamical system, that is, of signal propagation through non-linear neural networks, at initialisation. However the work of [26] actually studied the interaction of *both* dynamical processes for the simpler case of deep *linear* networks. Despite the overall computation of a linear network reducing to simple linear regression, the study found complex non-linear dynamics of the weight matrices evolving under gradient descent. The reason for this unexpected complexity was due to the dynamical evolution of signals through the network, in particular the attenuation at larger depths.

Considering the system at initialisation is both practically important, but also analytically helpful since it allows for the application of what is known as a dynamic mean field theory [31]. However, [26] did not delve into this underlying theory, assuming its application to be justified. In its essence the paper [26] studied the spectral properties of the "end-to-end" Jacobian matrix of neural networks, and devised initialisation schemes for standard non-linear networks aimed at

keeping the Jacobian spectrum well conditioned (eg. having its singular values close to the unit circle). In terms of the dynamical evolution of the weight matrices under gradient descent, the impact of different design choices by the practitioner and the data distribution, amongst other things, will be captured by the spectral properties of the Jacobian.

The study of the Jacobian matrix of neural networks at initialisation has since been carried much further. One strand of research has taken to the task of obtaining analytic control over the entire spectrum using tools from random matrix theory [81], [30], whereas another strand has applied the same ideas to understand more sophisticated neural network architectures and heuristics [29], [108], [109].

Two promising contributions in the study of the Jacobian, not at initialisation, have been [110] and [80]. In [110] the authors considered again deep linear networks, but instead considered the generalisation properties of the system, revealing once more the impact of the spectral properties of the Jacobian and its interaction with the data covariance matrix. In [80] studied the evolution of the gradient process, monitoring the Jacobian, and identified that the process typically converges to limit cycles. Interestingly, similar work [111] has deep connections to the local entropy theory of [106] developed in the context of the binary perceptron. inspired rigorous work on generalisation error bounds [79].

A different thread of research, with a stronger physics flavour, was revived in its application to neural networks by [26]. This thread focuses on the assumptions and construction of the dynamic mean field theory in describing the propagation of signals at initialisation. Dynamic mean field theory itself originates from the spin glass literature, from an attempt to study the Sherrington-Kirkpatrick spin glass model in the 1980s [112]. The method is thus well understood, and it is possible to derive the approximation from a considerably more general path integral approach, as described recently [31], [113]. This more general approach allows for the calculation of different quantities to what the standard theory provides, as well as corrections accounting for finite size effects (since the network size is far from the thermodynamic limit). Note that the path integral approach is rooted in non-equilibrium statistical physics, with applications to critical phenomena [31]. Another possible approach to generalising the dynamic mean field theory was presented in [114], which is based on effective field theories and renormalisation group methods in statistical physics. Such methods find their origin in the study of critical phenomena.

It is important to note that the idea of studying random neural networks originates in work by Amari [115], who subsequently applied ideas from differential geometry to clarify some of the mathematical aspects of the mean field dynamical system [116]. Similar work that is not rooted in statistical physics, but which also analyses the dynamical system properties and its mean field assumptions includes [117], [118]. These papers also provide more advanced control over the mean field approximations, also providing, for example, corrections for finite size effects. Both papers discussed in more detail in Chapter 7.

6.8 Chapter conclusion

This chapter has presented a discussion of ideas in statistical physics that underpin much of the work on statistical learning theory of both continuous and binary neural networks. While presented at a high level by necessity, the introduction to the concepts and language is important since the analysis of algorithms using these ideas has seen recent success, the remainder of the thesis notwithstanding.

The final two chapters of this thesis concern themselves with developing a theoretical understanding of gradient based algorithms for binary neural networks. As for the continuous case, there are two dynamical processes under consideration. One process corresponds to the gradient descent dynamics in an auxiliary parameter space, which is related in some way to the binary weight space of the network. The other dynamical process is the propagation of signals through the network, or in the case of the auxiliary parameters, through the network's continuous surrogate.

Therefore the technical contribution of the remaining chapters begin with the application of dynamic mean field theory to the stochastic binary networks and continuous surrogate models, in the spirit of [28] and [29]. This is an important and necessary step toward a better understanding of the dynamical processes occurring in the optimisation processes for binary neural networks. As discussed in the concluding remarks of Chapter 9, the interaction of the binary model and its surrogate, as observed via its training and generalisation performance, leaves many mysteries to consider. It is expected that in making progress toward uncovering the answers, many of the ideas presented at length in this chapter will necessarily intersect.

Chapter 7

Signal propagation in deterministic surrogates

This Chapter studies the deterministic Gaussian surrogate model of neural networks as presented in Chapter 5, for networks with stochastic binary weights and stochastic or deterministic binary neurons. The primary contribution is to successfully apply, in the spirit of [28], a mean field theory to analyse this surrogate network. The application hinges on the use of self-averaging arguments [98]. The recursive scalar equations which govern signal propagation in randomly initialised networks are derived. This derivation reveals that regardless of whether the surrogate is derived from a network with deterministic or stochastic binary neurons, the equations are the same, up to a constant scaling.

It is demonstrated via simulation that the recursive equations accurately describe the behaviour of randomly initialised networks, confirming the self-averaging properties. From the equations the depth scales that limit the maximum trainable depth are also derived. The maximum depth increases as the networks are initialised closer to criticality, similarly to standard neural networks.

The depths scale show that, contrary to common intuition, for networks with stochastic binary neurons, the means of the stochastic binary weights should be initialised towards the upper bounds (± 1) for deeper networks to be trainable, that is, with broken symmetry. It is demonstrated experimentally that trainability is indeed delivered with this initialisation, making it possible to train deeper stochastic binary neural networks.

This chapter also discusses the alternative perspective to signal propagation, as first established in [26], that this study is equivalent to controlling the singular value distribution of the input-output Jacobian matrix of the neural network [81] [30], specifically its mean. While for standard continuous neural networks the mean squared singular value of the Jacobian is directly related to the derivative of the correlation recursion equation, in the Gaussian based approximation this is not so. It is shown that in this case the derivative calculated is only an approximation of the Jacobian mean squared singular value. However, it is also shown that the approximation error approaches zero as the layer width diverges.

Following the presentation of the basic signal propagation theory, including experimental results, the theory is refined in line with more recent literature. Similarly to [118], the edges of chaos conditions of the deterministic Gaussian surrogate are also derived. In the case an edge does exist, the final equations are solved numerically to obtain the edge in the hyper-parameter space.

In summary this chapter makes the following contributions:

- Reviews the signal propagation theory for standard continuous networks
- Derives recursive signal propagation equations, and depth scales describing trainability for a deterministic surrogate model, based on the deterministic Gaussian based approximation
- Demonstrates that in the mean-field limit the signal propagation equations for the deterministic surrogate model are invariant to the choice of stochastic or deterministic binary neurons, up to a constant scaling
- Presents experimental results confirming the accuracy of mean field description of the surrogate model and its predictions suggesting new initialisation schemes and limits on the trainable depth
- Derives analogous equations for signal propagation in deterministic surrogate networks for those with stochastic binary weights and neurons (or deterministic binary neurons)
- Determines numerically the edges of chaos for the deterministic surrogate

7.1 Background: standard continuous networks

To begin with, the basic formalism developed in [28] is reviewed. A more current discussion of the literature on signal propagation will be disucssed in the next chapter. Assume the weights of a standard continuous network are initialised with $W_{ij}^{\ell} \sim \mathcal{N}(0, \sigma_w^2)$, biases $b^{\ell} \sim \mathcal{N}(0, \sigma_b^2)$, and input signal x_a^0 has zero mean $\mathbb{E}x^0 = 0$ and variance $\mathbb{E}[x_a^0 \cdot x_a^0] = q_{aa}^0$, and with a denoting a particular input pattern. As before, the signal propagates via equation (5.1) from layer to layer.

The particular mean field approximation used here replaces each element in the pre-activation field h_i^{ℓ} by a Gaussian random variable whose moments are matched. Therefor the variance $q_{aa}^{\ell} = \frac{1}{N_\ell}\sum_i (h_{i;a}^{\ell})^2$ is computed from layer to layer, starting from a particular input x_a^0. Likewise the covariance between the pre-activations $q_{ab}^{\ell} = \frac{1}{N_\ell}\sum_i h_{i;a}^{\ell} h_{i;b}^{\ell}$ is calculated from layer to layer, given two different inputs x_a^0 and x_b^0 with known covariance q_{ab}^0. As explained in [28], assuming the independence within a layer; $\mathbb{E}h_{i;a}^{\ell} h_{j;a}^{\ell} = q_{aa}^{\ell}\delta_{ij}$ and $\mathbb{E}h_{i;a}^{\ell} h_{j;b}^{\ell} = q_{ab}^{\ell}\delta_{ij}$, it is possible to derive recurrence relations from layer to layer

$$\begin{aligned} q_{aa}^{\ell} &= \sigma_w^2 \int Dz \phi^2(\sqrt{q_{aa}^{\ell-1}}\, z) + \sigma_b^2 \\ &:= \sigma_w^2 \mathbb{E}\phi^2(h_{j,a}^{\ell-1}) + \sigma_b^2 \end{aligned} \tag{7.1}$$

with $Dz = \frac{dz}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$ the standard Gaussian measure. The recursion for the covariance is given by

$$\begin{aligned} q_{ab}^{\ell} &= \sigma_w^2 \int Dz_1 Dz_2 \phi(u_a)\phi(u_b) + \sigma_b^2 \\ &:= \sigma_w^2 \mathbb{E}\big[\phi(h_{j,a}^{\ell-1})\phi(h_{j,b}^{\ell-1})\big] + \sigma_b^2 \end{aligned} \tag{7.2}$$

where

$$u_a = \sqrt{q_{aa}^{\ell-1}}\, z_1, \;\; u_b = \sqrt{q_{bb}^{\ell-1}}\big(c_{ab}^{\ell-1} z_1 + \sqrt{1-(c_{ab}^{\ell-1})^2}\, z_2\big)$$

and c_{ab}^{ℓ} is identified as the correlation in layer ℓ. Arguably the most important quantity is the the slope of the correlation recursion equation or mapping from layer to layer, denoted as χ, which is given by:

$$\chi = \frac{\partial c_{ab}^{\ell}}{\partial c_{ab}^{\ell-1}} = \sigma_w^2 \int Dz_1 Dz_2 \phi'(u_a)\phi'(u_b) \tag{7.3}$$

At the fixed point $c^* = 1$, the slope χ is denoted with the subscript χ_1. As discussed [28], when $\chi_1 = 1$, correlations can propagate to arbitrary depth.

Definition 1: (edge of chaos) *The edge of chaos (or critical initialisations) are the points* (σ_b^2, σ_w^2) *corresponding to* $\chi_1 = 1$.

Furthermore, χ_1 is equivalent to the mean square singular value of the Jacobian matrix for a single layer $J_{ij} = \frac{\partial h_i^{\ell}}{\partial h_j^{\ell-1}}$, as explained in [28]. Therefore controlling χ_1 will prevent the gradients from either vanishing or growing exponentially with depth.

7.2 Theoretical results

7.2.1 Forward signal propagation for deterministic Gaussian-binary networks

It is assumed that at initialisation the deterministic surrogate model has its binary weight means M_{ij}^{ℓ} drawn independently and identically from a distribution $P(M)$, with mean zero and variance of the means given by σ_m^2. For instance, a valid distribution could be a clipped Gaussian[1], or another stochastic binary variable, for example $P(M) = \frac{1}{2}\delta(M + \sigma_m) + \frac{1}{2}\delta(M - \sigma_m)$, whose variance is σ_m^2. The biases at initialization are distributed as $b^{\ell} \sim \mathcal{N}(0, \sigma_b^2)$.

In the stochastic binary neuron case the field is given by

$$h_i^{\ell} = \frac{1}{\sqrt{2}} \frac{\sum_j M_{ij}^{\ell} \varphi(h_j^{\ell-1}) + \sqrt{N^{\ell-1}}\, b_i^{\ell}}{1 + 2\sqrt{\sum_j [1 - (M_{ij}^{\ell})^2 \varphi^2(h_j^{\ell-1})]}} \tag{7.4}$$

which can be read from the Eq. 5.33.

Note in the first layer the denominator expression differs since in the first level of mean field analysis the inputs are not considered random (since a supervised learning setting is considered). As in the continuous case the variance $q_{aa}^{\ell} = \frac{1}{N_{\ell}} \sum_i (h_{i;a}^{\ell})^2$ and covariance $\mathbb{E} h_{i;a}^{\ell} h_{j;b}^{\ell} = q_{ab}^{\ell} \delta_{ij}$ are computed from layer to layer via recursive formulae. The key to the derivation is recognising that the denominator is a self-averaging quantity [98]. Under this assumption, the denominator is replaced with its mean,

$$\lim_{N\to\infty} \frac{1}{N} \sum_j 1 - (M_{ij}^{\ell})^2 \varphi^2(h_i^{\ell-1}) \tag{7.5}$$

$$= 1 - \mathbb{E}[(M_{ij}^{\ell})^2 \varphi^2(h_i^{\ell-1})] \tag{7.6}$$

$$= 1 - \sigma_m^2 \mathbb{E} \varphi^2(h_{j,a}^{l-1}) \tag{7.7}$$

where the property that the M_{ij}^{ℓ} and $h_i^{\ell-1}$ are each i.i.d. independent random variables at initialisation have been used [98]. The assumption is empirically verified in the numerical simulations section.

[1] That is, sample from a Gaussian then pass the sample through a function bounded on the interval $[-1, 1]$.

Following this self-averaging argument, expectations can be taken more readily, as follows.

$$q_{aa}^{\ell} = \mathbb{E}(h_i^{\ell})^2 = \frac{1}{2}\frac{\sum_j M_{ij}^{\ell}\varphi(h_i^{\ell-1}) + b_i^{\ell}}{1 + 2\sum_j[1 - (M_{ij}^{\ell})^2\varphi^2(h_i^{\ell-1})]} \tag{7.8}$$

$$= \frac{1}{2}\frac{N(\sigma_m^2\mathbb{E}\varphi^2(h_{j,a}^{l-1}) + \sigma_b^2)}{1 + 2(N - N\sigma_m^2\mathbb{E}\varphi^2(h_{j,a}^{l-1}))} \tag{7.9}$$

$$= \frac{1}{2}\frac{\sigma_m^2\mathbb{E}\varphi^2(h_{j,a}^{l-1}) + \sigma_b^2}{2(1 - \sigma_m^2\mathbb{E}\varphi^2(h_{j,a}^{l-1}))} \tag{7.10}$$

For more details consult the appendices. By similar steps, it can be shown in the deterministic binary neuron case the same expression is obtained, albeit with a different scaling constant. This can be easily seen by inspection of the field term in the deterministic neuron case,

$$h_i^{\ell} = \frac{1}{\sqrt{2}}\frac{\sum_j M_{ij}^{\ell}\varphi(h_i^{\ell-1}) + b_i^{\ell}}{\sqrt{\sum_j[1 - (M_{ij}^{\ell})^2\varphi^2(h_i^{\ell-1})]}} \tag{7.11}$$

In either case, using the derived expression for q_{aa}^{ℓ}, the correlation recursion can be written as

$$c_{ab}^{\ell} = \frac{1}{\sqrt{q_{aa}^{\ell}q_{bb}^{\ell}}}\frac{\sigma_m^2\mathbb{E}\varphi(h_{j,a}^{l-1})\varphi(h_{j,b}^{l-1}) + \sigma_b^2}{1 - \sigma_m^2\mathbb{E}\varphi^2(h_{j,a}^{l-1})} \tag{7.12}$$

The slope of the correlation mapping from layer to layer, when the normalized length of each input is at its fixed point $q_{aa}^{\ell} = q_{bb}^{\ell} = q^*(\sigma_m, \sigma_b)$, denoted as χ, is given by:

$$\chi = \frac{\partial c_{ab}^{\ell}}{\partial c_{ab}^{\ell-1}} = \frac{1 + q^*}{1 + \sigma_b^2}\sigma_m^2\int Dz_1 Dz_2 \varphi'(u_a)\varphi'(u_b) \tag{7.13}$$

where u_a and u_b are defined exactly as in the continuous case. Refer to the appendices for full details of the derivations. As in the standard continuous case, within several layers the variance approaches its asymptotic value, thus $q_{aa} = q_{bb}$ for two different inputs. This approximation is justified based on simulations. The recursive equations derived for this model and the continuous neural network are qualitatively similar, and by observation allow for the calculation of depth scales, just as in the continuous case [29].

7.2.2 Asymptotic expansions and depth scales

In the continuous case, the system approaches criticality as χ approaches 1, and thus the rate of convergence to any fixed point slows. The depth scales, as derived in [29] provide a quantitative indicator to the number of layers correlations will survive for, and thus how trainable a network is. It is shown here that similar depth scales can be derived for these Gaussian-binary networks.

According to [29] it should hold asymptotically that $|q_{aa}^{\ell} - q^*| \sim \exp(-\frac{\ell}{\xi_q})$ and $|c_{ab}^{\ell} - c^*| \sim \exp(-\frac{\ell}{\xi_c})$ for sufficiently large ℓ (the network depth), where ξ_q and ξ_c define the depth scales over which the variance and correlations of signals may propagate. Writing $q_{aa}^{\ell} = q^* + \epsilon^{\ell}$, it is shown in the appendix that:

$$\epsilon^{\ell+1} = \frac{\epsilon^{\ell}}{1 + q^*}\left[\chi_1 + \frac{1 + q^*}{1 + \sigma_b^2}\sigma_w^2\int Dz\varphi''(\sqrt{q^*}z)\varphi(\sqrt{q^*}z)\right] + \mathcal{O}((\epsilon^{\ell})^2) \tag{7.14}$$

One can similarly expand for the correlation $c^{\ell}_{ab} = c^* + \epsilon^{\ell}$, and if it is assumed that $q^{\ell}_{aa} = q^*$, then

$$\epsilon^{\ell+1} = \epsilon^{\ell}\big[\frac{1+q^*}{1+\sigma_b^2}\sigma_m^2 \int Dz\varphi'(u_1)\varphi'(u_2)\big] + \mathcal{O}((\epsilon^{\ell})^2) \tag{7.15}$$

The depth scales of interest are given by the log ratio $\log \frac{\epsilon^{\ell+1}}{\epsilon^{\ell}}$,

$$\begin{aligned}\xi_q^{-1} &= \log(1+q^*) \\ &\quad - \log\big[\chi_1 + \frac{1+q^*}{1+\sigma_b^2}\sigma_m^2 \int Dz\varphi''(\sqrt{q^*}z)\varphi(\sqrt{q^*}z)\big] \qquad (7.16)\\ \xi_c^{-1} &= -\log\big[\frac{1+q^*}{1+\sigma_b^2}\sigma_m^2 \int Dz\varphi'(u_1)\varphi'(u_2)\big] \\ &= -\log\chi \qquad (7.17)\end{aligned}$$

The arguments used in the original derivation [29] carry over to the Gaussian-binary case in a straightforward manner, albeit with more tedious algebra.

7.2.3 Jacobian mean squared singular value and mean field gradient backpropagation

As mentioned in the introduction to this Chapter, an equivalent perspective on this work is that controlling the forward propagation dynamics corresponds to controlling the mean squared singular value of the input-output Jacobain matrix of the entire network. This is because the input-output Jacobain matrix can be decomposed into the product of the single layer Jacobian matrices,

$$J = \prod_{\ell=1}^{L} J^{\ell}, \quad J^{\ell}_{ij} = \frac{\partial h^{\ell}_{i,a}}{\partial h^{\ell-1}_{j,a}} \tag{7.18}$$

In standard networks, the single layer Jacobian mean squared singular value is equal to the derivative of the correlation mapping χ as established in [28],

$$\mathbb{E}_{u,W^{\ell},h^{\ell}} \frac{||J^{\ell}u||_2^2}{||u||_2^2} = \chi \tag{7.19}$$

where the average is over the weights, Gaussian distribution of $h_i^{\ell-1}$ and a random perturbation u. For the Gaussian model studied here this is not true, and corrections must be made to calculate the true mean squared singular value. This can be seen by observing the terms arising from denominator of the pre-activation field[2],

$$\begin{aligned}J^{\ell}_{ij} = \frac{\partial h^{\ell}_{i,a}}{\partial h^{\ell-1}_{j,a}} &= \frac{\partial}{\partial h^{\ell}_j}\Big(\frac{\bar{h}^{\ell}_{i,a}}{\sqrt{\Sigma^{\ell}_{ii}}}\Big) \\ &= \varphi'(h^{\ell}_{i,a})\big[\frac{M^{\ell}_{ij}}{\sqrt{\Sigma^{\ell}_{ii}}} + (M^{\ell}_{ij})^2\frac{\bar{h}^{\ell}_{i,a}}{(\Sigma^{\ell}_{ii})^{3/2}}\varphi(h^{\ell}_{i,a})\big] \qquad (7.20)\end{aligned}$$

[2]The 'mean field' notation is dropped from Σ_{MF} for readability.

Since Σ_{ii} is a quantity that scales with the layer width N_ℓ, it is clear that when squared quantities are considered, such as the mean squared singular value, the second term due to the derivative of the denominator will vanish in the large layer width limit. Thus the mean squared singular value of the single layer Jacobian approaches χ. As such, the rest of this section proceeds as if χ is the exact quantity to be controlled.

The analysis involved in determining whether the mean squared singular value is well approximated by χ essentially goes through the mean field gradient backpropagation theory as described in [29]. This idea provides complementary depth scales for gradient signals travelling backwards. The next section moves on to simulations of random networks, verifying that the theory accurately predicts the average behaviour of randomly initialised networks.

7.2.4 Simulations

In Figure 7.1 it is seen that the average behaviour of random networks are well predicted by the mean field theory. The estimates of the variance and correlation from simulations of random neural networks provided some input signals are plotted. The dotted lines correspond to empirical means, the shaded area corresponds to one standard deviation, and solid lines are the theoretical prediction. Strong agreement is seen in both the variance and correlation plots.

Finally, in Figure 7.2 the variance and correlation depth scales are presented, as functions of σ_m, with different curves corresponding to different values of the bias variance σ_b. It is clear that similarly to continuous networks, σ_b and σ_m compete to effect the depth scale, which *appears* to only diverge with $\sigma_m \to 1$. Notice that contrary to standard networks where σ_b is scaled within one order of magnitude, σ_b must be changed across orders of magnitude to produce an effect, due to the scaling with the width of the network. Importantly, it is seen that the depth scale appears to only diverge as σ_m^2 approaches one value, whereas for continuous networks there are a continuous range of such points. This edge of chaos is studied more carefully in section 7.4.

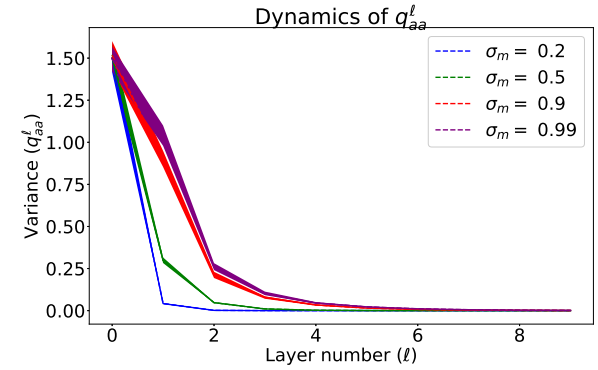
7.2.5 Remark: Valdity of the CLT for the first level of mean field

A legitimate immediate concern with initialisations that send $\sigma_m^2 \to 1$ may be that the binary stochastic weights $\mathbf{S}_{ij}^\ell$ are no longer stochastic, and that the variance of the Gaussian under the central limit theorem would no longer be correct. First recall the CLT's variance is given by $\mathrm{Var}(\mathbf{h}_{\mathrm{SB}}^\ell) = \sum_j (1 - m_j^2 x_j^2)$. If the means $m_j \to \pm 1$ then variance is equal in value to $\sum_j m_j^2(1 - x_j^2)$, which is the central limit variance in the case of only Bernoulli neurons at initialisation. Therefore, the applicability of the CLT is invariant to the stochasticity of the weights. This is not so of course if both neurons and weights are deterministic, for example if neurons are just tanh() functions.

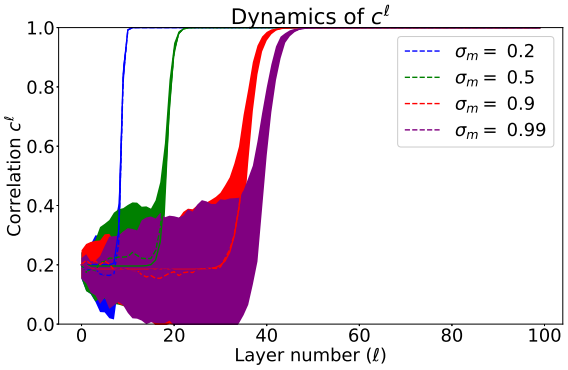
7.3 Experimental results

This section presents experimental tests of the mean field theory's predictions, by training networks to overfit a dataset in the supervised learning setting, having arbitrary depth and different initialisations. The performance of the networks is studied for various network depths and different mean variances $\sigma_m^2 \in [0, 1)$, fixing the bias variance to be close to zero, $\sigma_b^2 = 10^{-20}$.

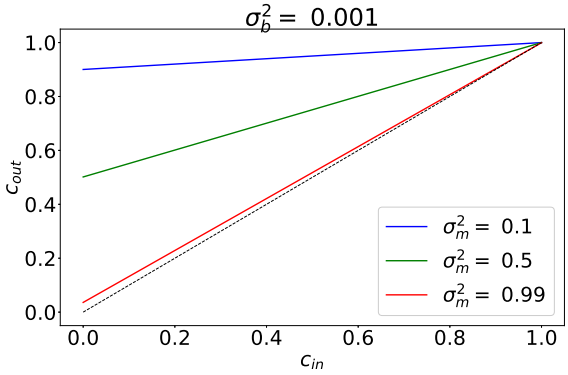
Both the continuous surrogate network and its binary network counterparts are evaluated. These include both the deterministic binary network and the stochastic binary network. In the deterministic binary case each binary weight is taken to be the sign of the mean, $\S_i j = \mathrm{sign}(M_{ij})$



(a)

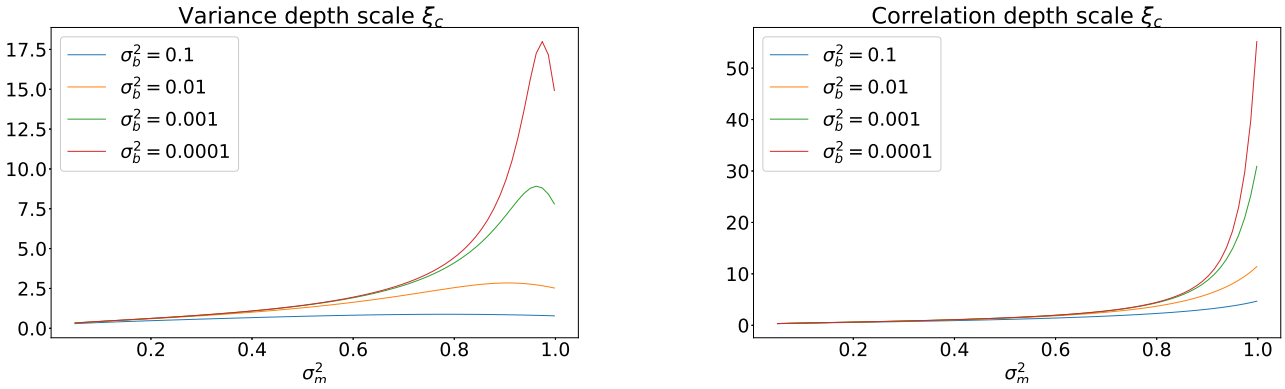


(b)



(c)

Figure 7.1: Dynamics of the variance and correlation maps, with simulations of a network of width $N = 1000$, 50 realisations, for various hyperparameter settings: $\sigma_m^2 \in \{0.2, 0.5, 0.99\}$ (blue, green and red respectively). (a) variance evolution, (b) correlation evolution. (c) correlation mapping (c_{in} to c_{out}), with $\sigma_b^2 = 0.001$



(a) (b)

Figure 7.2: Depth scales as σ_m^2 is varied. (a) The depth scale controlling the variance propagation of a signal (b) The depth scale controlling correlation propagation of two signals. Notice that the correlation depth scale ξ_c only diverges as $\sigma_m^2 \to 1$, whereas for standard continuous networks, there are an infinite number of such points, corresponding to various combinations of the weight and bias variances.

and where each neuron is the sign$(\cdot)$ non-linearity. In the stochastic binary case, the networks are evaluated by sampling several times. Recall the objective function is

$$\mathcal{L}_{\mathcal{D}}(f; M, b) = \sum_{\mu \in \mathcal{D}} \log \mathbb{E}_{\mathbf{S},\mathbf{x}}\big[p(y_\mu = f(x_\mu; \mathbf{S}, b, \mathbf{x}))\big] \tag{7.21}$$

It is then straightforward to obtain a Monte Carlo estimate of the probability for each input example μ

$$\mathbb{E}_{\mathbf{S},\mathbf{x}}\big[p(y_\mu = f(x_\mu; \mathbf{S}, b, \mathbf{x}))\big] \approx \sum_{\nu=1}^{N} p(y_\mu = f(x_\mu; S^\nu, b, x^\nu)) \tag{7.22}$$

where samples ν are denoted by S^ν, x^ν. The number of samples used in each simular is made clear in the figure captions, as these vary.

7.3.1 Experimental details

The networks are trained on the classic benchmark MNIST dataset of handwritten digits, where the task to correctly classify the digits. The general phenomena observed on MNIST is consistent across different, more computationally intensive datasets. Since the networks are trained for large network depths, a reduced MNIST training set size of 12,500 images is considered (25% of the usual training set), while maintaining the same number of test images, 6000. The training and test performance are recorded (that is, the percentage of the images correctly labelled) after several so-called "epochs" of gradient descent. An epoch is defined as a single pass over the training set. The optimiser used was a variant of gradient descent known as Adam [119] with learning rate of 1×10^{-3} chosen after simple grid search, and a batch size of 64. Results were similar for other optimizers, including SGD, SGD with momentum, and RMSprop. Note that these networks were trained without dropout, batchnorm or any other heuristics.

The program used to implement the algorithms is known as PyTorch, written in the Python scintific computing language. Further details on the experimental settings, and further results, are contained in Appendix [**?**]. All code for the experiments will be released online, accompanying publications resulting from this thesis.

7.3.2 Training and test performance for different mean initialisation σ_m^2

It is seen that the experimental results match the correlation depth scale derived for the surrogate model, with a similar proportion to the standard continuous case of $6\xi_c$ being the maximum possible attenuation in signal strength before trainability becomes difficult, as described in [29].

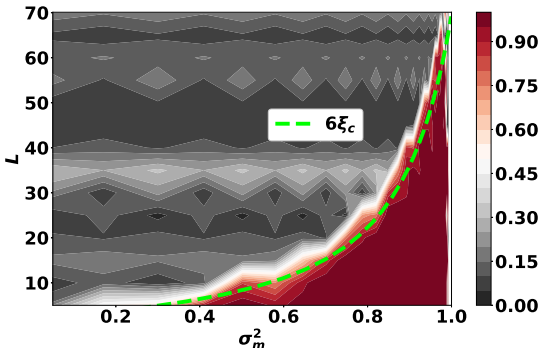
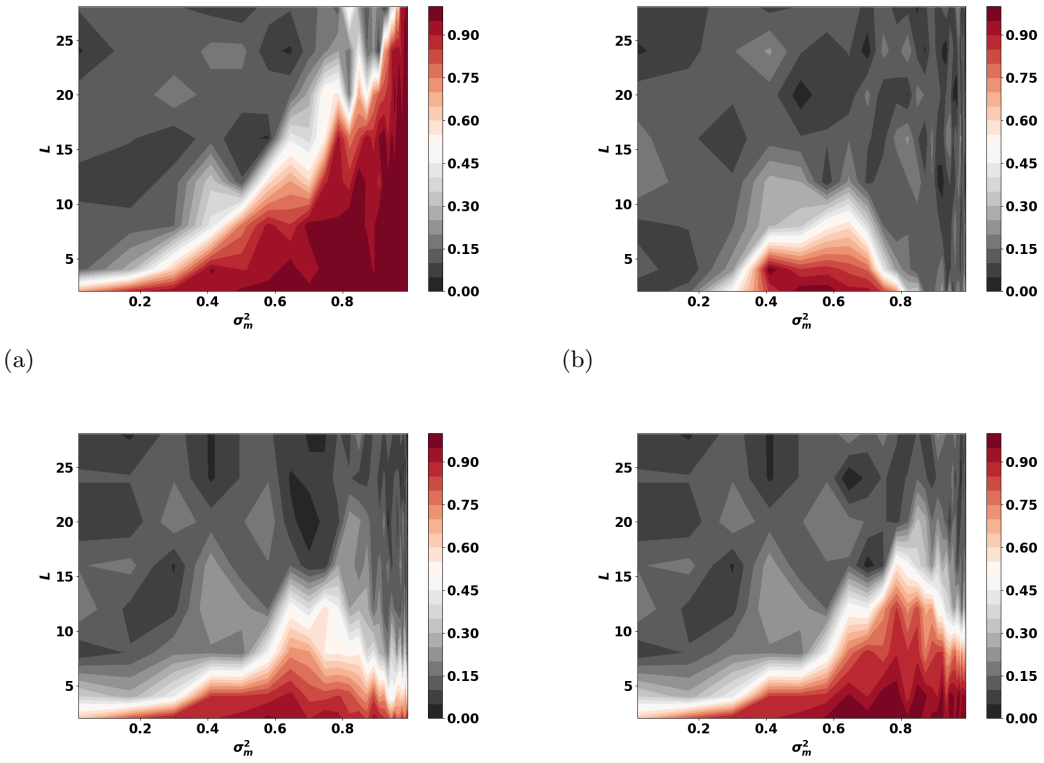


Figure 7.3: Training performance of the deterministic surrogate networks of different depth (in steps of 5 layers, up to $L = 100$), against the variance of the means σ_m^2. The performance plotted is that of the continuous surrogate, not its binary counterparts. Overlaid is a curve proportional to the correlation depth scale, matching the experimental results closely.

The reason the trainability is not seen to diverge in Figure 7.3 is that training time increases with depth, on top of requiring smaller learning rates for deeper networks, as described in detail in [26]. The experiment here used the same number of epochs regardless of depth, meaning shallower networks actually had an advantage over deeper networks, and yet still the initial variance σ_m^2 is the determining factor for trainability. A slight drop in trainability can be noticed for the continuous surrogate as the variance σ_m^2 approaches very close to one. As argued previously it is not likely that this is due to a violation of the CLT at the first level of mean field theory, however the input layer neurons are deterministic, so this may be an issue in the first CLT applied.

Figure 7.4 presents the training performance for the deterministic surrogate and its counterpart binary networks, both deterministic and stochastic. Once again, the algorithms are tested on the MNIST dataset and the results after 5 epochs are plotted. It can be seen that the performance of the stochastic network matches more closely the performance of the continuous surrogate, especially as the number of samples increases, from $N = 5$ to $N = 100$ samples.

The number of samples necessary to achieve better classification, at least for more shallow networks, depends on the number of training epochs. In some way, this is a sensible relationship, since during the course of training one might expect the means of the weights to polarise, moving closer to the bounds ± 1. Likewise, from experience continuous with neural networks,



(a) (b)

(c) (d)

Figure 7.4: Training performance of the continuous surrogate and its binary counterparts after training on the reduced MNIST dataset for 5 epochs. Figure (a) shows the performance of the continuous model. Figure (b) shows the performance of the deterministic binary network. Figures (c) and (d) show the performance of the stochastic binary network, averaged over 5 and 100 Monte Carlo samples respectively.

the neurons, which initially have zero mean pre-activations, are expected to "saturate" during training - that is, they become either always "on" $(+1)$ or "off" (-1). Being close to deterministic would require fewer samples overall, and this phenomena is observed.

The theory presented does not specify for how many steps of training the effects of the initialisation will persist, that is, for how long the network remains close to criticality. Therefore, the number of steps the network is trained for is an arbitrary choice, and in turn the experiments validate the theory in a limited way, since the theory itself is not specific enough.

7.4 Determining the edge of chaos

Recent literature on signal propagation in neural networks has taken to more carefully analysing the dynamical system properties of the mean field model of the networks [117] [120] [118]. A perspective from differential geometry is provided in [120], which also assumes as neuron non-linearity the erf() function to obtain closed form coupled equations. In [117], characterisations are given for neural networks with activation functions which have rectified linear units, revealing log-normal behaviour of pre-activations as the depth increases, assuming the width is held constant. The implication of this work is that for such networks the width must scale with the depth for the central limit theorem to hold to a certain degree of accuracy. A similar treatment is given in [118], which provides an instructive guide to the forward propagation properties of the system, in particular the rates of convergence of the correlations to their fixed points. This section follows the work in [118] and derives the edge of chaos conditions explicitly.

The edge of chaos condition is $\chi_1 = 1$, since this determines the stability of the correlation map fixed point $c^* = 1$. Note that for the deterministic surrogate this is always a fixed point. The hyper-parameters (σ_b^2, σ_m^2) that satisfy this condition can be found by solving the dynamical equations of the network.

Claim: *The points (σ_b^2, σ_m^2) corresponding to the edge of chaos are given by $\sigma_m^2 = 1/\mathbb{E}[\big(\varphi'(\sqrt{q^*}z)\big)^2] + \mathbb{E}[\varphi^2(\sqrt{q^*}z)]$ and finding σ_b^2 that satisfies*

$$q_{aa}^{\ell} = \sigma_b^2 + (\sigma_b^2 + 1)\frac{\mathbb{E}\varphi^2(h_{j,a}^{l-1})}{\mathbb{E}[\big(\varphi'(\sqrt{q^*}z)\big)^2]}$$

This can be established as follows. From the equation $\chi_1 = 1$,

$$\chi_1 = \frac{\sigma_m^2 \mathbb{E}[\big(\varphi'(\sqrt{q^*}z)\big)^2]}{1 - \sigma_m^2 \mathbb{E}[\varphi^2(\sqrt{q^*}z)]} = 1 \tag{7.23}$$

$$\implies \sigma_m^2 = \frac{1}{\mathbb{E}[\big(\varphi'(\sqrt{q^*}z)\big)^2] + \mathbb{E}[\varphi^2(\sqrt{q^*}z)]} \tag{7.24}$$

This can be substituted into the expression for the variance map, to obtain the expression in Claim 1.

Thus, in order to find the edge of chaos, as a function of the parameters σ_m^2 and σ_b^2, one must simply find a value of σ_b^2 which satisfies the variance map. This value for σ_b^2 is found numerically, as shown in Figure 7.5, for different neuron noise models and hence non-linearities $\varphi(\cdot)$. The critical initialisation for any of these design choices is found to be close to the point $(\sigma_m^2, \sigma_b^2) = (1, 0)$. However, it is not just the singleton point, as for example in [118] for the ReLu case for standard networks.

It is straightforward to numerically calculate the edges of chaos in the (σ_m^2, σ_b^2) plane. First consider both the stochastic and deterministic binary neurons, for both the $\tanh(\cdot)$ and $\text{erf}(\cdot)$ functions.

7.4.1 Stochastic binary weights and binary neurons

In Figure 7.5 it is clear that for the $\varphi(z) = \tanh(\kappa\, z)$ non-linearity, for values of $\kappa \leq 1$ appears to approach criticality for $(\sigma_m^2, \sigma_b^2) \to (1.0, 0.0)$. The maximum for σ_m^2 is of course one, since the weights considered are binary. Therefore, the 'edge' of chaos appears to exist only about the point $(1, 0)$. Note there is indeed a line and not a singular point, according the numerical solutions.

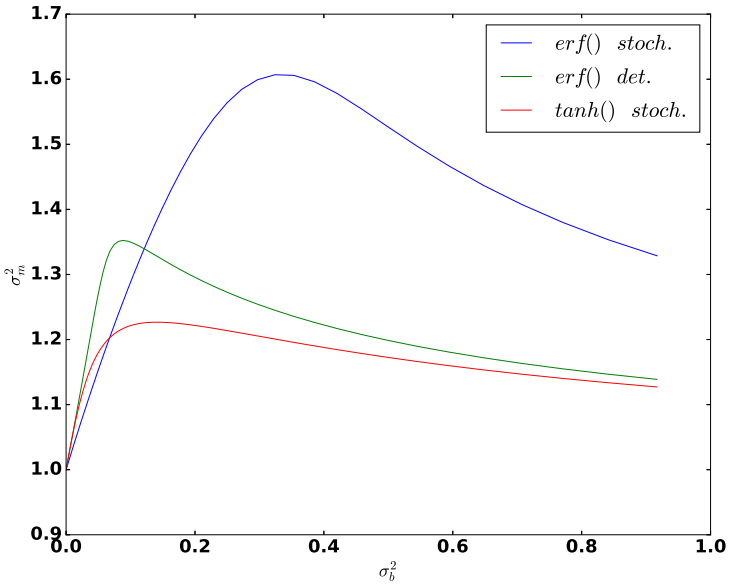


Figure 7.5: Edges of chaos for the deterministic surrogate model, for stochastic binary weights and stochastic or deterministic binary neurons. Presented is the edge of chaos in the (σ_m^2, σ_b^2), for both the a) stochastic neuron case with $\varphi(z) = \text{erf}(\frac{1}{4}\, z)$, b) the deterministic sign neuron case with $\varphi(z) = \text{erf}(\frac{1}{2}\, \cdot)$, and (c) the logistic based stochastic neuron, with tanh() approximation (see Chapter 5 for details). All edges are above $\sigma^2 = 1$ for all but small $\sigma_b^2 << 1$.

7.4.2 Continuous weights and stochastic binary neurons

In the case of stochastic binary neurons and continuous weights, it is not possible to manipulate the equations as before to solve for the weight variance, denoted as σ_w^2 in this case (w for a continuous weight, rather than a mean of a binary weight). This can be seen from the equations for this surrogate,

$$h_i^\ell = \frac{\sum_j W_{ij}^\ell \varphi(h_i^{\ell-1}) + b_i^\ell}{\sqrt{\sum_j (W_{ij}^\ell)^2 [1 - \varphi^2(h_i^{\ell-1})]}} \tag{7.25}$$

The correlation map for this surrogate is given by

$$c_{ab}^\ell = \frac{\sigma_w^2 \mathbb{E}\varphi(h_{j,a}^{l-1})\varphi(h_{j,b}^{l-1}) + \sigma_b^2}{\sigma_w^2(1 - \mathbb{E}\varphi^2(h_{j,a}^{l-1}))} \tag{7.26}$$

and when taking the derivative, it is clear that the weight variance σ_w^2 cancels,

$$\chi = \frac{\mathbb{E}\varphi'(h_{j,a}^{l-1})\varphi'(h_{j,b}^{l-1})}{(1 - \mathbb{E}\varphi^2(h_{j,a}^{l-1}))} \tag{7.27}$$

It is still possible to determine an edge of chaos condition, as shown in the appendix, however the numerical solutions return a diverging variance length q_{aa}, a problem which motivated the introduction of signal propagation theory to neural networks in the first place. It therefore appears that it is not possible to train networks with stochastic binary neurons and continuous weights, to arbitrary depth, via this surrogate network. Experimental results confirm this, though these experiments are not included here, since it suffices to report the lack of trainability.

7.5 Chapter conclusion

This chapter has presented a theoretical study of a binary neural network algorithm using dynamic mean field theory, following the analysis recently developed for standard continuous neural networks [28], [29]. Based on self-averaging arguments, it was possible to derive equations which govern signal propagation in wide, random neural networks, and obtained depth scales that limit trainability. Directly from the calculation of the signal propagation equations it is clear that the choice of neurons being either deterministic or stochastic binary variables makes little difference. Numerical simulations were presented which validate the theory of signal propagation in randomly initialised networks. Experimental results presented in turn validate the theoretical predictions around trainability.

Interesting experimental results were uncovered for the binary neural networks corresponding to the trained surrogate. This includes both deterministic binary and stochastic binary network. It was seen that during training, when evaluating the deterministic and stochastic binary counterparts concurrently with the surrogate, the performance of both binary networks is worse than the continuous model, especially as depth increases. The stochastic binary network was seen to outperform the deterministic binary network, which makes sense since the objective optimised is the expectation over an ensemble of stochastic binary networks. In either case, the difference between the continuous surrogate and the binary networks appears to decrease as training progresses.

The next chapter studies how signals propagate in untrained, random binary networks, both deterministic and stochastic. From this study it is possible to make more informed guesses as to why the binary networks may not perform well despite their surrogate training well to arbitrary depth.

This chapter also developed the signal propagation theory to explicitly calculate the edge of chaos conditions for deterministic surrogates for different combinations of stochastic weights or neurons. That is, for networks with either stochastic binary weights or neurons, or both. From this it was possible to categorise whether an edge of chaos exists for each model. It was found that surrogates for networks with continuous neurons and stochastic binary weights are easier to train, in the sense that the edge of chaos exists for a wider range of values of the hyper-parameters than in the case where weights and neurons are both stochastic binary variables.

Interestingly, it was found that a surrogate network for the case of stochastic binary neurons but continuous weights has no edge of chaos. This is a counter intuitive result. One might expect that the continuous weights would make for an easier problem than having stochastic binary weights, in the sense of critical initialisation. However, it appears the combination of both stochastic binary weights and neurons is beneficial. This shifts more weight to the notion that it is the neuron non-linearity (stochastic or not), that determines the properties of the dynamical system implemented by the neural network [118].

This study of this deterministic continuous surrogate network has provided considerable insight into the training of binary neural networks, which should inform further theoretical studies. It has also yielded results of practical significance, which should inform the development of new algorithms.

Chapter 8

Signal propagation for perturbed surrogates and binary networks

This chapter investigates the Gaussian Monte Carlo based approximation for binary neural networks, the so called "perturbed" surrogate, defined in Chapter 5. This approximation is also based on a Gaussian central limit assumption, but rather than integrating over each neuron as in the algorithm studied in Chapter 7, the pre-activation is instead sampled, since the resulting (stochastic) function is also differentiable. For this class of algorithm, the dynamic mean field theory can also be applied and is shown to accurately describe the propagation of signals through the surrogate networks, just as in the deterministic surrogate.

The resulting signal propagation equations are quite different to those of the deterministic surrogates, in line with the very different nature of the approximation. The analysis reveals that divergence in the derived depth scales depends on the various combinations of stochastic binary weights or neurons that one chooses, with only the continuous neuron case having a divergent depth scale. The derivations of all the signal propagation equations to all of these models are presented in this section, and subsequently the edges of chaos for all the models are presented as well.

In addition to studying the perturbed surrogate model, the signal propagation theory for both deterministic binary networks and stochastic binary networks is also presented. That is, networks that cannot be trained by gradient descent, but whose dynamic mean field behaviour can be studied readily. The motivation for this study is that it may provide some explanation for the poor performance in the early stages of training, of the binary networks that are trained via a given surrogate network. This point is elaborated on in the discussion.

In summary this chapter makes the following contributions:

- Derives analogous equations for signal propagation in "perturbed" surrogate networks, based on the Gaussian Monte Carlo approximation to stochastic binary neural networks.

- Proves that for certain surrogates there is no divergence in the corresponding depth scale by showing that there is no edge of chaos, depending on the combinations of weights and neurons being either continuous or binary stochastic variables

- Determines numerically the edges of chaos for the surrogates that have such an edge, revealing the limitations on the trainability of the models

- Derives signal propagation equations for deterministic binary networks, that is, with binary weights and sign() neuron non-linearity, establishing that there is no edge of chaos for such a network
- Derives signal propagation equations for stochastic binary networks, under the combinations of the original binary network's weights and neurons being either continuous or binary stochastic variables (but not both continuous)

8.1 Perturbed surrogate: stochastic binary weights and neurons

8.1.1 Signal propagation equations

As in previous Chapter, the objective is to compute the variance map for the perturbed algorithm. The pre-activation field for the perturbed surrogate with both stochastic binary weights and neurons is given by,

$$\mathbf{h}^l_{i,a} = \frac{1}{\sqrt{N}} \sum_j M^l_{ij} \phi(\mathbf{h}^{l-1}_{j,a}) + b^l_i + \boldsymbol{\epsilon}^\ell_{i,a} \frac{1}{\sqrt{N}} \sqrt{\sum_j 1 - (M^l_{ij})^2 \phi^2(\mathbf{h}^{l-1}_{j,a})} \tag{8.1}$$

recalling that $\boldsymbol{\epsilon} \sim \mathcal{N}(0,1)$. The non-linearity $\phi(\cdot)$ can of course be derived from any valid binary Bernoulli neuron model, as before (eg. $\tanh(\cdot)$ or $\text{erf}(\cdot)$).

Again, the variance of interest is defined as

$$q^l_{aa} = \frac{1}{N_l} \sum_i (\mathbf{h}^l_{i,a})^2 = \mathbb{E}\left[(\mathbf{h}^l_{i,a})^2\right] \tag{8.2}$$

Assuming again that

$$m_{ij} \sim N(0, \sigma^2_m) \tag{8.3}$$
$$b_i \sim N(0, \sigma^2_b) \tag{8.4}$$

and appealing to the same self-averaging arguments used in the previous section,

$$\lim_{N\to\infty} \frac{1}{N} \sum_j 1 - (M^\ell_{ij})^2 \phi^2(h^{\ell-1}_i) = 1 - \sigma^2_m \mathbb{E}\phi^2(\mathbf{h}^{l-1}_{j,a}) \tag{8.5}$$

the variance map is then found to be

$$\mathbb{E}\left[(\mathbf{h}^l_{i,a})^2\right] = \mathbb{E}\left[\left(\frac{1}{\sqrt{N}} \sum_j m^l_{ij} \phi(\mathbf{h}^{l-1}_{j,a}) + b^l_i + \frac{1}{\sqrt{N}} \boldsymbol{\epsilon}^\ell_{i,a} \sqrt{\sum_j 1 - (m^l_{ij})^2 \phi^2(h^{l-1}_{j,a})}\right)^2\right] \tag{8.6}$$
$$= \sigma^2_m \mathbb{E}\phi^2(\mathbf{h}^{l-1}_{j,a}) + \sigma^2_b + (1 - \sigma^2_m \mathbb{E}\phi^2(\mathbf{h}^{l-1}_{j,a})) \tag{8.7}$$
$$= 1 + \sigma^2_b \tag{8.8}$$

Interestingly, the variance map does not depend on the variance of the means of the binary weights. This is a counter intuitive result, not immediately obvious from the pre-activation field definition.

In the covariance map however, there is no such simplification, since the perturbation $\boldsymbol{\epsilon}_{i,a}$ is uncorrelated between examples a and b,

$$q^l_{ab} = \mathbb{E}\left[\mathbf{h}^l_{i,a}\mathbf{h}^l_{i,b}\right] \tag{8.9}$$

$$= \sigma^2_m \mathbb{E}\phi(\mathbf{h}^{l-1}_{j,a})\phi(\mathbf{h}^{l-1}_{j,a}) + \sigma^2_b \tag{8.10}$$

and thus the correlation map is given by

$$c^l_{ab} = \frac{\sigma^2_m \mathbb{E}\phi(\mathbf{h}^{l-1}_{j,a})\phi(\mathbf{h}^{l-1}_{j,a}) + \sigma^2_b}{1 + \sigma^2_b} \tag{8.11}$$

and the derivative of the correlation map is given by

$$\chi = \sigma^2_m \mathbb{E}\phi'(\mathbf{h}^{l-1}_{j,a})\phi'(\mathbf{h}^{l-1}_{j,b}) \tag{8.12}$$

8.1.2 Determining the edge of chaos

Since the mean variance σ^2_m does not appear in the variance map, there are different conditions for existence of the edge of chaos.

Claim: *There is no edge of chaos for the perturbed surrogate for a network with stochastic binary weights and stochastic binary neurons.*

Proof: The conditions for a critical initialisation are that $c^* = 1$ to be a fixed point and $\chi_1 = 1$. No such fixed point exists. A fixed point $c^* = 1$ exists if and only if $\sigma^2_m = \frac{1}{\mathbb{E}[\phi^2(h^{l-1}_{j,a})]}$. Note that $\sigma^2_m \leq 1$. For any $\phi(z)$ which is the mean of the stochastic binary neuron, the expectation $\mathbb{E}[\phi^2(z)] \leq 1$. For example, consider $\phi(z) = \tanh(\kappa z)$ for any finite kappa. Note that if $\kappa \to \infty$ corresponds to $\phi(z) = \text{sign}(z)$ and $c* = 1$ is in fact always a fixed point, but the sign(z) function does not have a derivative defined appropriately for a gradient descent procedure.

8.2 Perturbed surrogate: stochastic binary weights and continuous neurons

8.2.1 Signal propagation equations

As shown in the appendix, the signal propagation equations for the case of continuous neurons and stochastic binary weights yields the variance map,

$$q_{aa} = \mathbb{E}\phi^2(\mathbf{h}^{l-1}_{j,a}) + \sigma^2_b \tag{8.13}$$

Thus, once again, the variance map does not depend on the variance of the means of the binary weights. The covariance map however does retain a dependence on σ^2_m,

$$q^l_{ab} = \sigma^2_m \mathbb{E}\phi(\mathbf{h}^{l-1}_{j,a})\phi(\mathbf{h}^{l-1}_{j,a}) + \sigma^2_b \tag{8.14}$$

with the same expression as before. The correlation map is given by

$$c^l_{ab} = \frac{\sigma^2_m \mathbb{E}\phi(\mathbf{h}^{l-1}_{j,a})\phi(\mathbf{h}^{l-1}_{j,a}) + \sigma^2_b}{\mathbb{E}\phi^2(\mathbf{h}^{l-1}_{j,a}) + \sigma^2_b} \tag{8.15}$$

and the derivative of the correlation map is given by

$$\chi = \sigma^2_m \mathbb{E}\phi'(\mathbf{h}^{l-1}_{j,a})\phi'(\mathbf{h}^{l-1}_{j,b}) \tag{8.16}$$

8.2.2 Determining the edge of chaos

Claim: *The edge of chaos for the perturbed surrogate, for the case of continuous* $\tanh(\cdot)$ *neurons and stochastic binary weights is the singleton* $(\sigma_b^2, \sigma_m^2) = (0, 1)$.

Proof: From the correlation map there is a fixed point $c^* = 1$ if and only if $\sigma_m^2 = 1$, by inspection. In turn, the edge of chaos condition $\chi_1 = 1$ holds if

$$\mathbb{E}[(\phi'(\mathbf{h}_{j,a}^{l-1}))^2] = \frac{1}{\sigma_m^2} = 1 \tag{8.17}$$

Thus to find the critical initialisation one needs to find a value of $q_{aa} = \mathbb{E}\phi^2(\mathbf{h}_{j,a}^{l-1}) + \sigma_b^2$ that satisfies this final condition. In the case that $\phi(\cdot) = \tanh(\cdot)$, then the function $(\phi'(\mathbf{h}_{j,a}^{l-1}))^2 \leq 1$, taking the value 1 at the origin only, this requires $q_{aa} \to 0$. Therefore the only solution is the singleton $(\sigma_b^2, \sigma_m^2) = (0, 1)$.

Finding the edge of chaos corresponds to finding a value of $q_{aa} = \mathbb{E}\phi^2(\mathbf{h}_{j,a}^{l-1}) + \sigma_b^2$ that satisfies this final condition. In the case that $\phi(\cdot) = \tanh(\cdot)$, then the function $(\phi'(\mathbf{h}_{j,a}^{l-1}))^2 \leq 1$, taking the value 1 at the origin only, this requires $q_{aa} \to 0$. Thus the 'edge of chaos' is the singleton point $(\sigma_b^2, \sigma_m^2) = (0, 1)$. This is confirmed by experiment, as reported in the subsequent sections.

8.2.3 Experiments

As seen in Figures 8.1 and 8.2, the edge of chaos for the $\tanh(\cdot)$ non-linearity occurs only at the singleton point $(\sigma_b^2, \sigma_m^2) = (0, 1)$. Presented are simulations for varying σ_m^2, with fixed $\sigma_b^2 = 0$.

As in the previous chapter, the performance continuous surrogate and binary network are compared, at training time, in Figure 8.1, and at test time, in Figure 8.2. Once again, the divergence in the depth scale is observed for the continuous surrogate, but the binary network corresponding to the adapted means does not perform to similar depths. This effect was observed for different conditions, such as longer training time, larger network width and different gradient step sizes.

8.3 Perturbed surrogate: continuous weights and stochastic binary neurons

8.3.1 Signal propagation equations

As shown in the appendix, the signal propagation equations for the case of stochastic binary neurons and continuous weights yields the variance map,

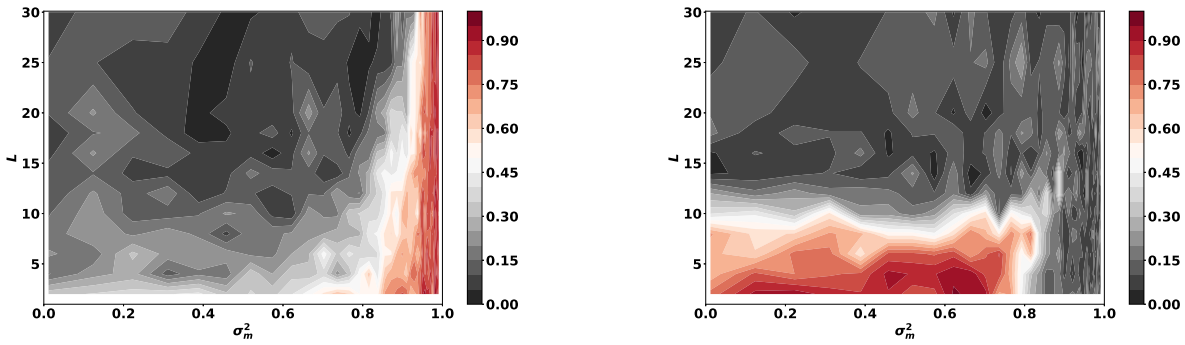
$$q_{aa} = \sigma_w^2 + \sigma_b^2 \tag{8.18}$$

where it can be seen the variance map does depend on the variance of the continuous weights. The covariance map is given by,

$$q_{ab}^l = \sigma_w^2 \mathbb{E}\phi(\mathbf{h}_{j,a}^{l-1})\phi(\mathbf{h}_{j,a}^{l-1}) + \sigma_b^2 \tag{8.19}$$

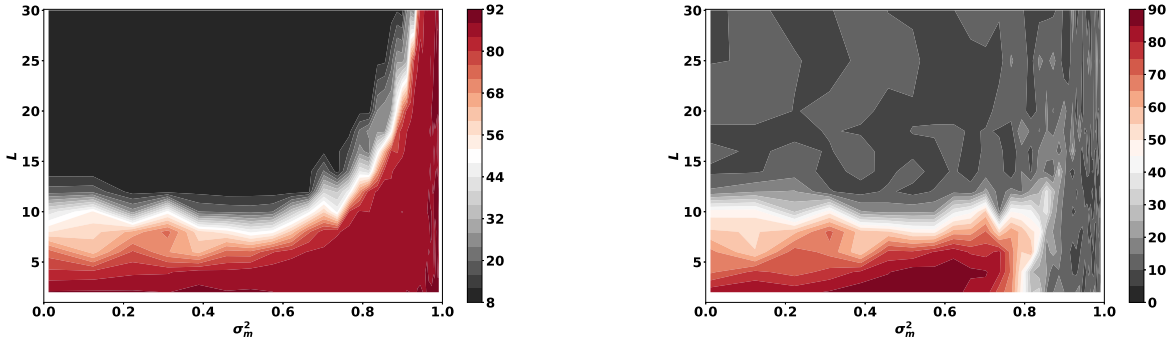
as is standard for this surrogate (and indeed continuous networks). The correlation map is given by

$$c_{ab}^l = \frac{\sigma_w^2 \mathbb{E}\phi(\mathbf{h}_{j,a}^{l-1})\phi(\mathbf{h}_{j,a}^{l-1}) + \sigma_b^2}{\sigma_w^2 + \sigma_b^2} \tag{8.20}$$



(a) (b)

Figure 8.1: Training performance of the perturbed surrogate networks: a) evaluation of continuous surrogate, b) evaluation of corresponding the binary model (non-stochastic). Maximum depth $L = 30$, steps of $L = 2$, after ten epochs on reduced MNIST training set (10%), using SGD with momentum. Non-linearity used was $\tanh(\cdot)$ (with $\kappa = 1$), and divergence in trainability of continuous surrogate is observed for hyperparameter setting of $(\sigma_m^2, \sigma_b^2) = (1, 0)$



(a) (b)

Figure 8.2: Test performance of the perturbed surrogate networks: a) evaluation of continuous surrogate, b) evaluation of corresponding the binary model (non-stochastic). Results corresponding to experiment presented in Figure 8.1.

and the derivative of the correlation map given by

$$\chi = \sigma_w^2 \mathbb{E}\phi'(\mathbf{h}_{j,a}^{l-1})\phi'(\mathbf{h}_{j,b}^{l-1}) \tag{8.21}$$

From the correlation map, it is clear that $c^* = 1$ if and only if $\mathbb{E}\phi^2(\mathbf{h}_{j,a}^{l-1}) = 1$. This is not possible however, since the function $\phi^2(z) \leq 1$ for all z. Therefore, this surrogate does not have an edge of chaos[1].

8.4 Signal propagation for deterministic and stochastic binary neural networks

This section presents the signal propagation equations for deterministic binary and stochastic networks. The case of deterministic binary networks considers random binary weights and sign$(\cdot)$ neuron non-linearities. The case of stochastic binary networks considers weights and neurons being either continuous or stochastic and binary (but not both continuous). In all cases, the equations are either identical to, or special cases of, the equations for the perturbed surrogate models.

The study of how signals propagate in such binary networks, despite not being directly relevant to the optimisation methods considered here, is nonetheless important for at least two reasons.

First of all, when evaluating any binary network which is trained by some algorithm (eg. gradient descent on a given surrogate model), signals will of course propagate forwards through the corresponding binary network. This network will either be deterministic or stochastic. In either case, it makes sense that the closer one is to the early stages of the training process, the closer the signal propagation behaviour is to the randomly initialised case. A theoretical description of this propagation is thus desirable.

A second reason for studying deterministic binary networks is that such networks have been used in popular heuristic algorithms for training binary neural networks [23], based on the so called "straight-through estimator" [84]. This heuristic, which is not derived as an estimator of any function, or derivative, propagates signals forward through a deterministic binary network, whose neurons are the sign$(\cdot)$ non-linearity, and whose weights are taken to be the sign() of some auxiliary parameter. In applying gradient descent, the non-linearities are "replaced" by a smooth function, such as tanh$(\cdot)$, and the auxiliary parameters are updated ignoring the sign() computation of the forward pass. While this heuristic is yet to be understood, given it is not clear what function or derivative it is an estimator of, it is important to understand the dynamics of the signals during the early stages of training.

8.4.1 Forward signal propagation

In this neural network, it should be understood that all neurons are simply sign$(\cdot)$ functions of their input, and all weights $W_{ij}^{\ell} \in \{\pm 1\}$ are randomly distributed according to

$$P(W_{ij}^{\ell} = +1) = 0.5 \tag{8.22}$$

$$\tag{8.23}$$

[1]The noiseless case, $\kappa \to \infty$, where the neuron is the sign function, does satisfy this condition, but the gradient of this function is not defined.

thus maintaining a zero mean.

The pre-activation field is given by

$$h_i^\ell = \frac{1}{\sqrt{N_{\ell-1}}} \sum_j W_{ij}^\ell \operatorname{sign}(h_j^{\ell-1}) + b_i^\ell \tag{8.24}$$

So, the length map is:

$$q_{aa}^\ell = \int Dz (\operatorname{sign}(\sqrt{q_{aa}^{\ell-1}} z)^2) + \sigma_b^2 \tag{8.25}$$

$$= 1 + \sigma_b^2 \tag{8.26}$$

Interestingly, this is the same value as for the perturbed Gaussian with stochastic binary weights and neurons.

The covariance evolves as

$$q_{ab}^\ell = \int Dz_1 Dz_2 \operatorname{sign}(u_a) \operatorname{sign}(u_b) + \sigma_b^2 \tag{8.27}$$

and the corresponding correlation map evolves as

$$c_{ab}^\ell = \mathcal{C}(c_{ab}^{\ell-1}, q_{aa}^{\ell-1}, q_{bb}^{\ell-1}, b, \sigma_b) = \frac{\int Dz_1 Dz_2 \operatorname{sign}(u_a) \operatorname{sign}(u_b) + \sigma_b^2}{\sqrt{q_{aa}^{\ell-1} q_{bb}^{\ell-1}}} \tag{8.28}$$

This correlation can be written in closed form. The first step is to rewrite the integral over h, for a joint density $p(h_a, h_b)$, and then rescale the h_a such that the variance is 1, so that $dh_a = \sqrt{q_{aa}} dv_a$

$$\int dh_a dh_b \operatorname{sign}(h_a) \operatorname{sign}(h_b) p(h_a, h_b) = \int dv_a dv_b \operatorname{sign}(v_a) \operatorname{sign}(v_b) p(v_a, v_b) \tag{8.29}$$

$$= \big(2P(v_1 > 0, v_2 > 0) - 2P(v_1 > 0, v_2 < 0)\big) \tag{8.30}$$

where $p(v_a, v_b)$ is a joint with the same correlation c_{ab} (which is now equal to its covariance), and the capital $P(v_1, v_2)$ corresponds to the (cumulative) distribution function. A standard result for standard bivariate normal distributions with correlation ρ,

$$P(v_1 > 0, v_2 > 0) = \frac{1}{4} + \frac{\arcsin(\rho)}{2\pi}, \qquad P(v_1 > 0, v_2 < 0) = \frac{\cos^{-1}(\rho)}{2\pi} \tag{8.31}$$

So then,

$$\int dh_a dh_b \phi(h_a) \phi(h_b) p(h_a, h_b) = \sqrt{q_{aa} q_{bb}} \big(\frac{1}{2} + \frac{\arcsin(c_{ab}^{\ell-1})}{\pi} - \frac{\cos^{-1}(c_{ab}^{\ell-1})}{\pi}\big) \tag{8.32}$$

Thus the correlation map is:

$$c_{ab}^\ell = \frac{\big(\frac{1}{2} + \frac{\arcsin(c_{ab}^{\ell-1})}{\pi} - \frac{\cos^{-1}(c_{ab}^{\ell-1})}{\pi}\big) + \sigma_b^2}{\sqrt{q_{aa}^{\ell-1} q_{bb}^{\ell-1}}} \tag{8.33}$$

$$= \frac{\frac{2}{\pi} \arcsin(c_{ab}^{\ell-1}) + \sigma_b^2}{\sqrt{q_{aa}^{\ell-1} q_{bb}^{\ell-1}}} \tag{8.34}$$

From before $q_{aa} = 1 + \sigma_b^2$, and so

$$c_{ab}^{\ell} = \frac{\frac{2}{\pi}\arcsin(c_{ab}^{\ell-1}) + \sigma_b^2}{1 + \sigma_b^2} \tag{8.35}$$

Recall that $\arcsin(1) = \frac{\pi}{2}$. Therefore $c^* = 1$ is a fixed point always.

The slope of the correlation map, denoted as usual by $\chi = \frac{\partial c_{ab}^{\ell}}{\partial c_{ab}^{\ell-1}}$, is calculated by first integrating over the $\phi() = \text{sign}()$ non-linearities, and then taking the derivative. This proceeds as follows,

$$\chi = \frac{\partial c_{ab}^{\ell}}{\partial c_{ab}^{\ell-1}} = \frac{2}{\pi}\frac{1}{\sqrt{q_{aa}^{\ell-1}q_{bb}^{\ell-1}}}\frac{1}{\sqrt{1-(c_{ab}^{\ell-1})^2}} = \frac{2}{\pi}\frac{1}{(1+\sigma_b^2)}\frac{1}{\sqrt{1-(c_{ab}^{\ell-1})^2}} \tag{8.36}$$

It is clear that the derivative χ diverges at $c_{ab}^{\ell} = 1$, meaning that there is no 'edge of chaos' for this system. This of course means that correlations will not propagate to arbitrary depth in deterministic binary networks, as one might have expected.

8.4.2 Stochastic weights and neurons

Beginning again with the variance map,

$$q_{aa}^{l} = \mathbb{E}\left[(h_{i,a}^{l})^2\right] \tag{8.37}$$

where in this the field is given by

$$h_{i,a}^{l} = \frac{1}{\sqrt{N}}\sum_j W_{ij}^{l} x_{h_{j,a}^{l-1}} + b_i^{l} \tag{8.38}$$

where $x_{h_{j,a}^{l-1}}$ denotes a Bernoulli neuron whose natural parameter is the pre-activation from the previous layer.

The expectation for the length map is defined in terms of nested conditional expectations, since the idea is to average over all random elements in the forward pass,

$$q_{aa}^{\ell} = \mathbb{E}_h\mathbb{E}_{x|h}x_{h_{j,a}^{l-1}} + \sigma_b^2 \tag{8.39}$$

$$= 1 + \sigma_b^2 \tag{8.40}$$

Once again, this is the same value as for the perturbed Gaussian with stochastic binary weights and neurons.

Similarly, the covariance map gives us,

$$q_{ab}^{l} = \mathbb{E}\left[h_{i,a}^{l}h_{i,b}^{l}\right] = \mathbb{E}_{h_a,h_b}\mathbb{E}_{x_b|h_a}\mathbb{E}_{x_b|h_b}x_{h_{j,a}^{l-1}}x_{h_{j,b}^{l-1}} + \sigma_b^2 = \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2 \tag{8.41}$$

with $phi(\cdot)$ being the mean function, or a shifted and scaled version of the cumulative distribution function for the Bernoulli neurons, just as in previous Chapters. This expression is equivalent to the perturbed surrogate for stochastic binary weights and neurons, with a mean variance of $\sigma_m^2 = 1$. Following the arguments for that surrogate, no edge of chaos exists.

8.4.3 Stochastic binary weights and continuous neurons

In this case, as it is shown in the appendix, the resulting equations are

$$q_{aa}^{\ell} = \mathbb{E}\phi^2(h_{j,a}^{l-1}) + \sigma_b^2 \tag{8.42}$$

$$q_{ab}^{l} = \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2 \tag{8.43}$$

which are, once again, the same as for the perturbed surrogate in this case, with $\sigma_m^2 = 1$. This means that this model *does* have an edge of case, at the point $(\sigma_m^2, \sigma_b^2) = (1, 0)$.

8.4.4 Continuous weights and stochastic binary neurons

Similar arguments to the above show that the equations for this case are exactly equivalent to the perturbed surrogate model. This means that no edge of chaos exists in this case either.

8.5 Chapter conclusion

This Chapter has studied the signal propagation properties of binary neural networks and a perturbed surrogate network for training binary neural networks. In the case of the surrogate model, based on a Monte Carlo sample of a Gaussian approximation, it was possible to apply the mean field theory and validate the signal propagation equations, once again the derivation was based on the self-averaging arguments. From these equations it was determined that the surrogates would have diverging depth scales by the direct calculation of the edges of chaos. The only model to have an edge of chaos is the case of stochastic binary weights and continuous neurons.

In the case of the deterministic binary and stochastic binary networks, the signal propagation description, under the dynamic mean field theory, revealed important theoretical results. In particular, in the case of either networks with both deterministic binary weights and neurons, and stochastic binary weights and neurons, it was seen that there is no edge of chaos initialisation.

This result is relevant for understanding the training of binary networks using the surrogate networks, in particular the evaluation of the trained binary counterparts. Consider for a moment the signal propagation behaviour of a continuous network that has been trained, and this is not in its initially random state. This means that, as far as the mean field theory is concerned, the self-averaging behaviour, including any central limit behaviour, cannot be assumed to hold. However, clearly the networks are still performing some useful information processing, and thus are not in either the completely ordered case (asymptotic correlation $c^{\infty} = 1$) nor the chaotic case ($c^{\infty} = 0$).

It makes sense that the closer one is to the early stages of the training process, the closer the signal propagation behaviour will reflect the randomly initialised case. That is, correlations do not propagate, since there is no edge of chaos condition. However, it is possible that as training progresses the signal propagation behaviour binary counterparts of these surrogates might approach the signal propagation of the trained surrogate model. This may explain the difference in the performance between the surrogate model and its binary counterparts (deterministic or stochastic) early in training, a difference which appears to decrease as training progresses.

Chapter 9

Conclusion

This thesis has studied autonomous decision making systems in two parts. The first part was concerned with a model known as a reciprocal chain, a generative statistical model of target dynamics, used in decision making for target tracking systems. The proposed advantage of a reciprocal chain over existing models is the ability to capture higher level target behaviour such as proceeding to a destination. The second part was concerned with neural networks, used as discriminative models for classification tasks, when their parameters are constrained to have low precision.

In the introduction the thesis was placed within both a domain of application, and an academic discipline. In terms of a domain of application, the two parts of the thesis are jointly motivated by the spectre of advanced computation and decision making in real world applications, by devices processing information "at the edge". So called edge processing refers to local decision making on devices that are typically resource constrained. Tracking systems equipped with target dynamics modelled with reciprocal chains may be able to make decisions locally at the site of a device, such as a camera, since it processes information on a higher level of abstraction. Neural networks with low precision parameters, in particular those with binary parameters, are able in principle to operate on fast, low memory and low power hardware, removing the infeasible demands of the current full precision computation of regular neural networks.

In terms of a discipline, the two parts can be placed within the emerging field of machine learning, which sees the intersection of established disciplines such as statisics, optimisation and computer vision, to name a few. The two parts are distinguished according to the separation long established in statistics, of generative and discriminative modelling approaches for decision making. One advantage of generative statistical models is that they are considered to be more interpretable by human users than discriminative models. The disadvantage they face is that they are typically less flexible, whereas discriminative models, such as neural networks, can be applied to a wider set of problems, provided there is sufficient data.

This final chapter is devoted to summarising the contributions made in both parts of the thesis and connecting results between various chapters, as well as the two parts themselves. A discussion of future avenues for research is also included in this chapter.

9.1 Conclusion for decision making with reciprocal chains

The first part of the thesis was concerned with a generative statistical model of "intention", or "destination awareness" of target dynamics in a tracking context. The model studied is

known as a reciprocal chain, a stochastic process that posits a joint probability distribution over the initial and final states of target trajectories. The use of this model was motivated by considering the idea that, in more complex domains such as through camera networks or through road networks, targets exhibit behaviours not explained by Newtonian dynamics. It was argued that a human tasked with tracking an intelligent agent, might model behaviour using some approximation resembling Bayesian inference applied to observations of a Markov decision process (MDP). This is a problem known as inverse planning and is a computationally prohibitive exercise not required for tracking. Furthermore, it was argued the dynamics for reciprocal models qualitatively resemble those from a MDP without this computational overhead. The reciprocal model was investigated and measured against other candidate tracking models, for a detection task known as track extraction, using a novel observation model.

Reciprocal chains are stochastic processes on a fixed interval, which one can interpret as a time interval. As reviewed in the background material presented in Chapter 2, a destination can be encoded into target dynamics by prescribing a joint end-points distribution for a trajectory with given local Markov dynamics. This means that reciprocal chains are non-causal generalisations of Markov chains, since they do not satisfy the Markov property. Several important results were reviewed in this chapter, in particular the Markov or Schrödinger bridge construction of a reciprocal chain. Based on this construction, it is possible write down hidden reciprocal chain filters for state estimation and detection algorithms. The computational cost of using such models, at inference time, is N-times that of standard Markov chain filtering complexity, where N is the dimension of the state space.

Chapter 2 concluded with a discussion of the fixed time interval nature of reciprocal chains and similar models, and the potential issues that such a constraint might impose. As discussed, popular mathematical model for an agent operating in an environment with some goal is a MDP. In a tracking setting, one could model a destination as the "goal" of the agent, and consider agent behaviour on an infinite time horizon. This motivated an exploration of the closest time homogenous Markov process to a given reciprocal chain, with the measure of closeness being in terms of Kullback-Leibler divergence. The resulting stochastic process produced qualitatively similar dynamics to a MDP with a prescribed goal state. This is an interesting point of view to take. If pursued, one can relate Schrödinger bridge models to a branch of control theory known variously as path integral control theory or Kullback-Liebler control, which can be defined on finite or infinite time horizons. Such connections are worthwhile to explore in the tracking context, since explicit models of "agency", such as a MDP, are likely to be desirable as tracking systems will be expected to process and appropriately handle more complex human behaviour.

In order to assess the utility of modelling destination awareness with reciprocal chains, this required first a design of a simulation environment comparable to the real-world domains where reciprocal chains may find use. Targets with dynamics that incorporate a notion of intent may be more appropriate in domains such as tracking a target through a road network, or through a network of cameras, rather than a single camera or along a road. Therefore, one of the first contributions of Chapter 3 was to propose a novel simulation environment exemplary of such a domain. This particular environment included an observation model incorporating "clutter", defined to be observations of uncertain origin. These observations do not relate to the target of interest but may interfere with the performance of the tracker.

Within this simulation environment, the problem of track extraction was studied. Track extraction is a detection problem where the task is to decide whether a set of observations originated from a target, or not. A likelihood ratio test was constructed based on normalised

optimal hidden reciprocal chain filters.

In order to understand, via simulation, the potential advantages of reciprocal chains, several viable models within the class of Markov processes were compared to reciprocal chains. The included models were; (i) a Markov chain with given local dynamics, (ii) the Schrödinger bridge model, and (iii) a reciprocal chain with a uniform joint endpoints distribution. Assuming the ground truth to be that targets exhibit intentional behaviour codified by a joint endpoints distribution, the benefit to the detector performance for track extraction from incorporating this information faithfully was studied. Based on simulations, it was found that the benefit to tracking, as measured by the area under the receiver-operator-characteristic curve, was in an approximately linear relationship with the Kullback-Liebler divergence between the true joint distribution and the joint distributions implied by each of the comparison models. This insight would help to guide the application of reciprocal models, indicating where a benefit to tracking performance could be expected.

9.1.1 Directions of future research

Tracking algorithms based around optimal Bayesian inference, such as those considered in the thesis, do not scale well to complex tracking scenarios such as multi-target tracking. This is a fundamental problem due to the combinatorial explosion in the number of possible tracks (or hypotheses), as the number of measurements and targets increases. Scaling issues can be partially alleviated by the application of approximate inference techniques such as particle filtering, or heuristic techniques such as "pruning" tracks based on thresholding track likelihoods.

A question worthy of exploration is whether the incorporation of higher level information improves multi-target tracking. The results from Chapter 3 suggest this might be the case, since reciprocal chains essentially re-weight the probabilities of trajectories through a system. In combination with the particle filter or pruning techniques, this could yield dividends for tracking systems, and should be explored in future.

Another line of future research, for the tracking of targets equipped with some model of intent, could step towards inverse planning problems. That is, the modelling of targets as agents that evolve dynamically according to a Markov decision process. In particular, the class of path integral or Kullback-Liebler control problems are especially interesting, since they are very closely related to the Schrödinger bridge models that motivated reciprocal processes. An example of the potential application of such Markov processes is that one can condition the process on attaining several states (or in general, distributions) at several different times. This idea of targets visiting "waypoints" along a trajectory was explored in the context of non-Markov Gaussian reciprocal processes [121] recently. It was also argued that accentuating the connections between the Markov reciprocal processes and the Markov decision processes studied under Kullback-Leibler control theory would be valuable. For example, the paper of [52] and related works have introduced advanced importance sampling methods for approximate Bayesian estimation of these processes. As described previously, such approximate methods are likely to be necessary in complex scenarios encountered in tracking problems.

9.2 Conclusions for for statistical learning with neural models

Neural networks are highly flexible discriminative models for decision making. In recent years a significant engineering and economic effort has seen the development of software and computational tools that have enabled researchers to satisfactorily solve tasks once thought to be

quite difficult. However, the most successful neural network designs are both memory and power hungry, being continuous functions represented with full precision variables, and having been optimised on dedicated hardware such as graphical processing units. In the context of processing on devices "at the edge" of a computer network, this makes the use of standard neural networks infeasible.

The second part of the thesis therefore studied the problem of optimising neural networks of low precision, in particular binary (± 1) networks. One of the primary motivation for networks entirely composed of binary variables in particular is the low power consumption and high speed computation they enable, as compared to standard full precision neural networks.

Introducing discrete variables in neural networks creates challenges for the optimisation processes typically applied to full precision neural networks. Specifically, since the neural networks are not differentiable, one cannot directly apply popular continuous optimisation techniques. Therefore, researchers resort to various approximations in order to obtain differentiable surrogate networks that, when trained, produce binary neural networks that perform well.

The latter chapters of the thesis focused on *stochastic* binary neural networks, which use the stochasticity to develop differentiable surrogates. Several surrogate networks were developed and explored, with the study questioning the role of parameter initialisation in particular. The theoretical results obtained, based on tools from statistical physics, provide insight into the optimisation process, as well as practical advice for those wishing to train binary neural networks.

In broader terms, the second part of this thesis is encouraging as it starts to "open" the black box of neural networks. It is encouraging that the theoretical tools to do this are relatively simple ideas borrowed from statistical physics and dynamical systems. While not a substitute to the transparency of generative models such as those studied in the first part, theoretical descriptions of these complex decision making systems is crucial for their reliable use in more diverse applications.

9.2.1 Summary of contributions for statistical learning with binary neural networks

The contributions made in the second part of the thesis included both an extensive review of the background to the binary neural network learning problem, presented in Chapters 4, 5 and 6, as well as new algorithms in 5, and a theoretical analysis of several algorithms and binary networks in Chapters 7 and 8.

The review of the background theory was presented with the aim of providing a unified treatment of the elements composing the optimisation algorithms under consideration. This contribution resulted in a broad discussion, covering both the technical details of the problem required in surrogate design in Chapters 4 and 5, as well as an introduction to the theoretical tools being borrowed from statistical physics in Chapter 6.

Chapter 5 also introduced new approximations for deriving surrogate models, based on a novel Markov chain based derivation. This Markov chain representation encompasses all existing and new surrogates for stochastic binary neural networks. Of particular note was the development of a Monte Carlo based approximation for neural networks with both stochastic binary weights and neurons, yielding what was referred to as a perturbed surrogate network.

The theoretical contributions presented in Chapters 7 and 8 applied several ideas from statistical physics to analytically study properties of binary neural networks and the surrogate networks used in their training. Specifically, the technical contributions investigated the typical behaviour of signals propagating forward and backward through these network models, at

initialisation. A key technical step was to apply self-averaging arguments to derive so called dynamic mean field equations which govern signal propagation in random neural networks, in the limit of infinite network width. These equations, which are coupled scalar equations, are functions of initialisation hyperparameters, that is, the variance of the weights' means and the bias variance. The dynamic mean field description, and its theoretical predictions, were confirmed by numerical simulations and experiment using specialised neural network software framework PyTorch [122].

From the signal propagation equations depth scales were derived that provide quantitative guides to the limits of trainability. This line of work led to the derivation of the "edge of chaos" conditions for various network surrogates. The conditions allow one to determine, analytically or numerically, whether or not an edge of chaos exists for a given surrogate model. This means binary network algorithms can be categorised into those which have critical initialisation and those which do not.

Chapter 7 considered a popular deterministic surrogate network [33], [85]. The derivations revealed that the choice of deterministic or stochastic binary neuron had little impact on the equations, with only a scaling constant difference in one of the two mean field equations. Further results include categorising the edge of chaos (EOC) properties for different choices of stochastic binary weights or neurons. For the case of stochastic binary weights and neurons, there exists an EOC. In the case of stochastic binary neurons and continuous weights, there exists no EOC, a counter intuitive result, since generally one would expect a network with higher precision weights to train more easily than the stochastic binary weight counterpart. There exists no deterministic surrogate for the case of stochastic binary weights and continuous neurons.

Chapter 8 considered a surrogate network of a different nature, involving a Monte Carlo sampling approximation in order to attain a differentiable function. For this surrogate, the EOC only exists for the case of continuous neurons and stochastic binary weights, with all other choices having no EOC.

Practically, the solutions to the edge of chaos equations, if such an edge exists, provide the values of the hyper-parameters for which the depth scales diverges. This analysis is the first major theoretical study of multi-layer binary neural network algorithms. In the context of neural network theory more broadly, it provides an analogue to the analyses recently developed for standard continuous neural networks [28], [29], [117], [118]. The basic dynamic mean field theory also holds for these surrogates, although in certain cases there exists an extra hyper-parameter dependence, arising from the stochastic neuron noise model.

A pressing question, of both theoretical and practical significance, is on the nature of the relationship between the continuous surrogate and its binary network counterpart (either deterministic or stochastic). The experimental results in Chapter 7 revealed that the trained binary networks do not perform well at large depths, despite the continuous surrogate attaining excellent performance on both the training and test sets. Another observation from the experiments was that earlier in training, the stochastic networks, with a sufficient number of samples, tends to outperform the deterministic binary networks.

The difference between the surrogate and the binary networks appears to decrease as training progresses. Based on the analysis of binary networks in Chapter 8, it was established that these networks have no edge of chaos, and are always in the chaotic regime at initialisation. This means that, in turn, correlations do not propagate to arbitrary depth in binary networks, and all correlations asymptotically approach zero. As such, it makes sense that the closer one is to the start of training, the worse the performance of the binary networks will be, since their

signal propagation behaviour is chaotic. This is in contrast to continuous networks such as the surrogates with edges of chaos, where signals propagate well at the start of training, by design. Based on this logic, it was argued in 8 that as training progresses, the signal propagation properties of the binary networks, deterministic and stochastic, might approach those of the trained continuous surrogate.

9.2.2 Directions for future research

The second part of the thesis raises open questions. In terms of further theoretical tools for analysis, there are several different routes to consider. The dynamic mean field theory presented and used here can be the starting point of two traditionally distinct theories. The first is the theory of critical phenomena, culminating in the so called renormalisation group methods. A recent formulation of such a theory for standard continuous networks was recently proposed [114]. A different route is to study the path integral approach to studying the same questions, which is based on stochastic differential equations [31], [113]. One advantage of these more advanced statistical physics theories is that in deriving the same dynamic mean field equations, it is possible to also derive corrections to account for the fluctuations about the mean predictions, as seen in the random network simulations. These corrections include, for instance, taking into account finite size effects in the width of the network, which in some continuos neural networks has seen to become an issue [117]. It is important to note that the information geometric approach of [116] proposes to study finite size effects as well.

The perspective of controlling the spectrum of input-output Jacobian matrix first proposed in [26] is also a compelling one, especially if one is interested purely in the optimisation of neural networks, since the spectral properties of the Jacobian matrix control much of the gradient descent process. This line of work has been extensively developed using random matrix theory in [81] [30], from the original proposals of [26] regarding orthogonal initialisation, which allows for large training speed gains. For example, orthogonal initialisations were recently defined for convolutional neural networks, allowing for the training of networks with tens of thousands of layers [123]. Whether a sensible orthogonal initialisation can be defined for binary neural network algorithms, and if it is possible to apply the random matrix calculations are important questions. The study here provides an important first step in this direction.

Finally, as a brief note, it is expected that the results presented in the second part of this thesis may be of interest to researchers of Bayesian approaches to deep learning. A Bayesian approach is desirable as one might like a measure of uncertainty over the parameters over the neural network. Both the perturbed and deterministic Gaussian approximations presented here have been used as the basis of approximate variational Bayesian algorithms [124], [119]. Therefore the results presented here on signal propagation at initialisation may prove interesting for a problem of Bayesian inference over neural networks with discrete variables.

Bibliography

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct 2016.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, 2000.

[5] C. L. Baker, J. B. Tenenbaum, and R. R. Saxe, "Bayesian models of human action understanding," in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, ser. NIPS'05. Cambridge, MA, USA: MIT Press, 2005, pp. 99–106. [Online]. Available: http://dl.acm.org/citation.cfm?id=2976248.2976261

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, September 2016, pp. 3464–3468.

[7] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957. [Online]. Available: http://www.jstor.org/stable/2098689

[8] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[9] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion A Handbook of Algorithms /—cYaakov Bar-Shalom, Peter K. Willett, Xin Tian*. Storrs, CT : YBS Publishing, 2011, includes bibliographical references and index.

[10] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3047–3055.

[11] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 4696–4704.

[12] "sotabench.com," https://sotabench.com/benchmarks/image-classification-on-imagenet, 2019, accessed: 22-10-2019.

[13] S. Mittal, "A survey on optimized implementation of deep learning models on the nvidia jetson platform," *Journal of Systems Architecture*, vol. 97, pp. 428 – 442, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1383762118306404

[14] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," 2016.

[15] C. L. Baker, R. R. Saxe, and J. B. Tenenbaum, "Bayesian theory of mind: Modeling joint belief-desire attribution," in *In Proceedings of the Thirtieth Third Annual Conference of the Cognitive Science Society*, 2011, pp. 2469–2474.

[16] M. Fanaswala and V. Krishnamurthy, "Spatiotemporal trajectory models for metalevel target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 1, pp. 16–31, Jan 2015.

[17] M. Fanaswala, V. Krishnamurthy, and L. White, "Destination-aware target tracking via syntactic signal processing," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 3692–3695.

[18] M. Fanaswala and V. Krishnamurthy, "Detection of anomalous trajectory patterns in target tracking via stochastic context-free grammars and reciprocal process models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 76–90, Feb 2013.

[19] G. Stamatescu, A. Dick, and L. B. White, "Multi-camera tracking of intelligent targets with hidden reciprocal chains," in *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Nov 2015, pp. 1–8.

[20] B. Pannetier, K. Benameur, V. Nimier, and M. Rombaut, "Vs-imm using road map information for a ground target tracking," in *2005 7th International Conference on Information Fusion*, vol. 1, July 2005, pp. 8 pp.–.

[21] Chih-Chung Ke, J. G. Herrero, and J. Llinas, "Comparative analysis of alternative ground target tracking techniques," in *Proceedings of the Third International Conference on Information Fusion*, vol. 2, July 2000, pp. WEB5/3–WEB510 vol.2.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 5 2015.

[23] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4107–4115. [Online]. Available: http://papers.nips.cc/paper/6573-binarized-neural-networks.pdf

[24] R. Banner, I. Hubara, E. Hoffer, and D. Soudry, "Scalable methods for 8-bit training of neural networks," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 5145–5153. [Online]. Available: http://papers.nips.cc/paper/7761-scalable-methods-for-8-bit-training-of-neural-networks.pdf

[25] N. Wang, J. Choi, D. Brand, C.-Y. Chen, and K. Gopalakrishnan, "Training deep neural networks with 8-bit floating point numbers," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 7675–7684. [Online]. Available: http://papers.nips.cc/paper/7994-training-deep-neural-networks-with-8-bit-floating-point-numbers.pdf

[26] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6120

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html

[28] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli, "Exponential expressivity in deep neural networks through transient chaos," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 3360–3368.

[29] S. S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein, "Deep information propagation," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. [Online]. Available: https://openreview.net/forum?id=H1W1UN9gg

[30] J. Pennington, S. Schoenholz, and S. Ganguli, "The emergence of spectral universality in deep networks," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Storkey and F. Perez-Cruz, Eds., vol. 84. Playa Blanca, Lanzarote, Canary Islands: PMLR, 09–11 Apr 2018, pp. 1924–1932. [Online]. Available: http://proceedings.mlr.press/v84/pennington18a.html

[31] A. Crisanti and H. Sompolinsky, "Path integral approach to random neural networks," *Phys. Rev. E*, vol. 98, p. 062120, Dec 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.98.062120

[32] A. Braunstein and R. Zecchina, "Learning by message passing in networks of discrete synapses," *Phys. Rev. Lett.*, vol. 96, p. 030201, Jan 2006. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.96.030201

[33] D. Soudry, I. Hubara, and R. Meir, "Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 963–971.

[34] G. Stamatescu, L. B. White, and R. Bruce-Doust, "Track extraction with hidden reciprocal chains," *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 1097–1104, April 2018.

[35] D. A. Castanon, B. C. Levy, and A. S. Willsky, "Algorithms for the incorporation of predictive information in surveillance theory†," *International Journal of Systems Science*, vol. 16, no. 3, pp. 367–382, 1985. [Online]. Available: https://doi.org/10.1080/00207728508926680

[36] J. E. Kulkarni and L. Paninski, "State-space decoding of goal-directed movements," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 78–86, 2008.

[37] B. Jamison, "Reciprocal processe," *Z. Warscheinlichkeitstheorie verw. Gebiete*, vol. 41, no. 30, pp. 65–86, 1970. [Online]. Available: https://doi.org/10.1007/BF00532864

[38] A. Beghi, "On the relative entropy of discrete-time markov processes with given end-point densities," *IEEE Transactions on Information Theory*, vol. 42, no. 5, pp. 1529–1535, Sep. 1996.

[39] B. C. Levy, R. Frezza, and A. J. Krener, "Modeling and estimation of discrete-time gaussian reciprocal processes," *IEEE Transactions on Automatic Control*, vol. 35, no. 9, pp. 1013–1023, Sep. 1990.

[40] L. B. White and F. Carravetta, "Optimal smoothing for finite state hidden reciprocal processes," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2156–2161, Sep. 2011.

[41] ——, "New normalised bayesian smoothers for signals modelled by non-causal compositions of reciprocal chains," in *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, June 2014, pp. 205–208.

[42] ——, "Normalized optimal smoothers for a class of hidden generalized reciprocal processes," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6489–6496, Dec 2017.

[43] L. B. White and H. X. Vu, "Maximum likelihood sequence estimation for hidden reciprocal processes," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2670–2674, Oct 2013.

[44] R. J. Elliott, J. B. Moore, and L. Aggoun, *Hidden Markov models : estimation and control / Robert J. Elliott, Lakhdar Aggoun, John B. Moore.* Springer-Verlag New York, 1995.

[45] T. Georgiou and M. Pavon, "Positive contraction mappings for classical and quantum schrödinger systems," *Journal of Mathematical Physics*, vol. 56, no. 3, 3 2015.

[46] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing).* New York, NY, USA: Wiley-Interscience, 2006.

[47] F. Carravetta and L. B. White, "Modelling and estimation for finite statereciprocal processes," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2190–2202, Sep. 2012.

[48] R. Bruce-Doust, "Forgetting properties of finite-state reciprocal processes," Master's thesis, 2017. [Online]. Available: http://hdl.handle.net/2440/113380

[49] P. Dai Pra, "A stochastic control approach to reciprocal diffusion processes," *Applied Mathematics and Optimization*, vol. 23, no. 1, pp. 313–329, Jan 1991. [Online]. Available: https://doi.org/10.1007/BF01442404

[50] H. J. Kappen, "Path integrals and symmetry breaking for optimal control theory," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 11, pp. P11 011–P11 011, nov 2005.

[51] E. Todorov, "Linearly-solvable markov decision problems," pp. 1369–1376, 2007. [Online]. Available: http://papers.nips.cc/paper/3002-linearly-solvable-markov-decision-problems.pdf

[52] J. Bierkens and H. J. Kappen, "Explicit solution of relative entropy weighted control," *Systems and Control Letters*, vol. 72, pp. 36 – 43, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167691114001583

[53] G. van Keuk, "Sequential track extraction," *IEEE Transactions on Aerospace Electronic Systems*, vol. 34, pp. 1135–1148, Oct. 1998.

[54] M. Wieneke and W. Koch, "On sequential track extraction within the pmht framework," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, p. 276914, Oct 2007. [Online]. Available: https://doi.org/10.1155/2008/276914

[55] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, Jan 2004.

[56] J. Black and T. Ellis, "Multi camera image tracking," *Image and Vision Computing*, vol. 24, no. 11, pp. 1256 – 1267, 2006, performance Evaluation of Tracking and Surveillance. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885605000806

[57] A. R. Dick and M. J. Brooks, "A stochastic approach to tracking objects across multiple cameras," in *Australian Conference on Artificial Intelligence*, 2004, pp. 160–170.

[58] H. Detmold, A. van den Hengel, A. Dick, A. Cichowski, R. Hill, E. Kocadag, K. Falkner, and D. S. Munro, "Topology estimation for thousand-camera surveillance networks," in *2007 First ACM/IEEE International Conference on Distributed Smart Cameras*, Sep. 2007, pp. 195–202.

[59] A. Cichowski, C. Madden, H. Detmold, A. Dick, A. van den Hengel, and R. Hill, "Tracking hand-off in large surveillance networks," in *2009 24th International Conference Image and Vision Computing New Zealand*, Nov 2009, pp. 276–281.

[60] T. Kirubarajan, Y. Bar-Shalom, K. R. Pattipati, and I. Kadar, "Ground target tracking with variable structure imm estimator," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 26–46, Jan 2000.

[61] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[62] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[63] D. J. Hand, "Measuring classifier performance: a coherent alternative to the area under the roc curve," *Machine Learning*, vol. 77, no. 1, pp. 103–123, Oct 2009. [Online]. Available: https://doi.org/10.1007/s10994-009-5119-5

[64] M. Jordan, "Why the logistic function? a tutorial discussion on probabilities and neural networks," Massachusetts Institute of Technology, Tech. Rep., 1995.

[65] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[66] F. Ribeiro and M. Opper, "Expectation propagation with factorizing distributions: A gaussian approximation and performance results for simple models," *Neural Computation*, vol. 23, no. 4, pp. 1047–1069, April 2011.

[67] M. Mezard and A. Montanari, *Information, Physics, and Computation*. New York, NY, USA: Oxford University Press, Inc., 2009.

[68] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, pp. 1303–1347, 2013. [Online]. Available: http://jmlr.org/papers/v14/hoffman13a.html

[69] C. P. Robert, *The Bayesian choice: a decision-theoretic motivation*. Springer-Verlag, 1994.

[70] J. O. Berger, *Statistical decision theory and Bayesian analysis; 2nd ed.*, ser. Springer Series in Statistics. New York: Springer, 1985. [Online]. Available: https://cds.cern.ch/record/1327974

[71] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002.

[72] L. Bottou, F. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018. [Online]. Available: https://doi.org/10.1137/16M1080173

[73] P. Bhlmann and S. van de Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[74] J. W. T. Peters and M. Welling, "Probabilistic binary neural networks," *CoRR*, vol. abs/1809.03368, 2018. [Online]. Available: http://arxiv.org/abs/1809.03368

[75] V. N. Vapnik, "An overview of statistical learning theory," *Trans. Neur. Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999. [Online]. Available: https://doi.org/10.1109/72.788640

[76] P. L. Bartlett, M. I. Jordan, and J. D. Mcauliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006. [Online]. Available: http://www.jstor.org/stable/30047445

[77] T. Zhang, "Statistical behavior and consistency of classification methods based on convex risk minimization," *Ann. Statist.*, vol. 32, no. 1, pp. 56–85, 02 2004. [Online]. Available: https://doi.org/10.1214/aos/1079120130

[78] P. Whittle, "Risk Sensitivity, A Strangely Pervasive Concept," *Macroeconomic Dynamics*, vol. 6, no. 01, pp. 5–18, February 2002.

[79] G. K. Dziugaite and D. M. Roy, "Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data," in *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*, 2017. [Online]. Available: http://auai.org/uai2017/proceedings/papers/173.pdf

[80] P. Chaudhari and S. Soatto, "Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks," *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–10, 2017.

[81] J. Pennington, S. Schoenholz, and S. Ganguli, "Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4785–4795. [Online]. Available: http://papers.nips.cc/paper/7064-resurrecting-the-sigmoid-in-deep-learning-through-dynamical-isometry-theory-and-practice.pdf

[82] M. C. Fu, *Stochastic Gradient Estimation.* New York, NY: Springer New York, 2015, pp. 105–147.

[83] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, May 1992. [Online]. Available: https://doi.org/10.1007/BF00992696

[84] Y. Bengio, N. Léonard, and A. C. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *CoRR*, vol. abs/1308.3432, 2013, accessed: Mon, 13 Aug 2019 16:47:35 +0200. [Online]. Available: http://arxiv.org/abs/1308.3432

[85] C. Baldassi, F. Gerace, H. J. Kappen, C. Lucibello, L. Saglietti, E. Tartaglione, and R. Zecchina, "Role of synaptic stochasticity in training low-precision neural networks," *Phys. Rev. Lett.*, vol. 120, p. 268103, Jun 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.120.268103

[86] O. Shayer, D. Levi, and E. Fetaya, "Learning discrete weights using the local reparameterization trick," *CoRR*, vol. abs/1710.07739, 2017. [Online]. Available: http://arxiv.org/abs/1710.07739

[87] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1-2, pp. 1–305, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1561/2200000001

[88] D. P. Kingma and M. 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. [Online]. Available: http://arxiv.org/abs/1312.6114

[89] C. Maddison, A. Mnih, and Y. Teh, "The concrete distribution: A continuous relaxation of discrete random variables." International Conference on Learning Representations, 2017.

[90] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. [Online]. Available: https://openreview.net/forum?id=rkE3y85ee

[91] G. Tucker, A. Mnih, C. J. Maddison, D. Lawson, and J. Sohl-Dickstein, "Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Curran Associates Inc., 2017, pp. 2624–2633.

[92] A. F. Murray and P. J. Edwards, "Enhanced mlp performance and fault tolerance resulting from synaptic weight noise during training," *IEEE Transactions on Neural Networks*, vol. 5, no. 5, pp. 792–802, Sep. 1994.

[93] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, Jan. 1995. [Online]. Available: http://dx.doi.org/10.1162/neco.1995.7.1.108

[94] H. Touchette, "The large deviation approach to statistical mechanics," *Physics Reports*, vol. 478, no. 1, pp. 1 – 69, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0370157309001410

[95] H. Touchette and R. J. Harris, *Large Deviation Approach to Nonequilibrium Systems*. John Wiley & Sons, Ltd, 2013, ch. 11, pp. 335–360. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527658701.ch11

[96] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: an Introduction*. Oxford; New York: Oxford University Press, 2001.

[97] K. Binder and A. P. Young, "Spin glasses: Experimental facts, theoretical concepts, and open questions," *Rev. Mod. Phys.*, vol. 58, pp. 801–976, Oct 1986. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.58.801

[98] M. Mezard, G. Parisi, and M. Virasoro, *Spin Glass Theory and Beyond*, 01 1987, vol. 9.

[99] T. Castellani and A. Cavagna, "Spin-glass theory for pedestrians," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 05, p. P05012, may 2005.

[100] G. Györgyi, "Techniques of replica symmetry breaking and the storage problem of the mcculloch–pitts neuron," *Physics Reports*, vol. 342, no. 4, pp. 263 – 392, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0370157300000739

[101] E. Gardner, "The space of interactions in neural network models," *Journal of Physics A: Mathematical and General*, vol. 21, no. 1, pp. 257–270, jan 1988.

[102] Krauth, Werner and Mézard, Marc, "Storage capacity of memory networks with binary couplings," *J. Phys. France*, vol. 50, no. 20, pp. 3057–3066, 1989. [Online]. Available: https://doi.org/10.1051/jphys:0198900500200305700

[103] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová, "Gibbs states and the set of solutions of random constraint satisfaction problems," *Proceedings of the National Academy of Sciences*, vol. 104, no. 25, pp. 10 318–10 323, 2007. [Online]. Available: https://www.pnas.org/content/104/25/10318

[104] H. Huang, K. Y. M. Wong, and Y. Kabashima, "Entropy landscape of solutions in the binary perceptron problem," *Journal of Physics A: Mathematical and Theoretical*, vol. 46, no. 37, p. 375002, aug 2013.

[105] H. Horner, "Dynamics of learning for the binary perceptron problem," *Zeitschrift für Physik B Condensed Matter*, vol. 86, no. 2, pp. 291–308, Jun 1992. [Online]. Available: https://doi.org/10.1007/BF01313839

[106] C. Baldassi, C. Borgs, J. T. Chayes, A. Ingrosso, C. Lucibello, L. Saglietti, and R. Zecchina, "Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes," *Proceedings of the National Academy of Sciences*, vol. 113, no. 48, pp. E7655–E7662, 2016. [Online]. Available: https://www.pnas.org/content/113/48/E7655

[107] J. Kadmon and H. Sompolinsky, "Optimal architectures in a solvable model of deep networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4781–4789. [Online]. Available: http://papers.nips.cc/paper/6330-optimal-architectures-in-a-solvable-model-of-deep-networks.pdf

[108] G. Yang, J. Pennington, V. Rao, J. Sohl-Dickstein, and S. S. Schoenholz, "A mean field theory of batch normalization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=SyMDXnCcF7

[109] L. Xiao, Y. Bahri, J. Sohl-Dickstein, S. Schoenholz, and J. Pennington, "Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 5393–5402. [Online]. Available: http://proceedings.mlr.press/v80/xiao18a.html

[110] A. M. Saxe, J. L. McClelland, and S. Ganguli, "A mathematical theory of semantic development in deep neural networks," *Proceedings of the National Academy of Sciences*, vol. 116, no. 23, pp. 11 537–11 546, 2019. [Online]. Available: https://www.pnas.org/content/116/23/11537

[111] P. Chaudhari, A. M. Oberman, S. J. Osher, S. Soatto, and G. Carlier, "Deep relaxation: partial differential equations for optimizing deep neural networks," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1704.04932

[112] H. Sompolinsky and A. Zippelius, "Dynamic theory of the spin-glass phase," *Phys. Rev. Lett.*, vol. 47, pp. 359–362, Aug 1981. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.47.359

[113] J. Schuecker, S. Goedeke, D. Dahmen, and M. Helias, "Functional methods for disordered neural networks," *arXiv e-prints*, May 2016.

[114] S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "A correspondence between random neural networks and statistical field theory," *CoRR*, vol. abs/1710.06570, 2017, accessed: Mon, 13 Aug 2019. [Online]. Available: http://arxiv.org/abs/1710.06570

[115] S.-i. Amari, "A method of statistical neurodynamics," *Kybernetik*, vol. 14, no. 4, pp. 201–215, Dec 1974. [Online]. Available: https://doi.org/10.1007/BF00274806

[116] S. Amari, R. Karakida, and M. Oizumi, "Statistical neurodynamics of deep networks: Geometry of signal spaces," *CoRR*, vol. abs/1808.07169, 2018. [Online]. Available: http://arxiv.org/abs/1808.07169

[117] A. Labatie, "Characterizing well-behaved vs. pathological deep neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 3611–3621. [Online]. Available: http://proceedings.mlr.press/v97/labatie19a.html

[118] S. Hayou, A. Doucet, and J. Rousseau, "On the impact of the activation function on deep neural networks training," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 2672–2680. [Online]. Available: http://proceedings.mlr.press/v97/hayou19a.html

[119] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015, accessed: Wed, 20 Nov 2019. [Online]. Available: http://arxiv.org/abs/1412.6980

[120] S. Amari, R. Karakida, and M. Oizumi, "Statistical neurodynamics of deep networks: Geometry of signal spaces," *CoRR*, vol. abs/1808.07169, 2018, accesed: Sun, 02 Sep 2019. [Online]. Available: http://arxiv.org/abs/1808.07169

[121] L. B. White and F. Carravetta, "Stochastic realisation and optimal smoothing for gaussian generalised reciprocal processes," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 369–374.

[122] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[123] L. Xiao, Y. Bahri, J. Sohl-Dickstein, S. Schoenholz, and J. Pennington, "Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 5393–5402. [Online]. Available: http://proceedings.mlr.press/v80/xiao18a.html

[124] J. M. Hernández-Lobato and R. P. Adams, "Probabilistic backpropagation for scalable learning of bayesian neural networks," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 1861–1869. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045316

Appendix A

Signal propagation derivations for deterministic surrogates

A.1 Derivation of signal propagation equations in deterministic surrogate networks

Here we present the derivations for the signal propagation in the deterministic surrogate network models studied in the thesis in Chapter 7. The derivations are similar in all other surrogates, and thus not repeated for brevity.

A.1.1 Variance propagation

We first calculate the variance given a signal:

$$q_{aa}^{l} = \frac{1}{N_l} \sum_{i} \left(h_{i,a}^{l} \right)^2 = E\left[\left(h_{i,a}^{l} \right)^2 \right] \tag{A.1}$$

Where for us:

$$h_{i,a}^{l} = \frac{\sum_j m_{ij}^{l} \phi\left(h_{j,a}^{l-1} \right) + b_i^{l}}{\sqrt{\sum_j \left(1 - \left(m_{ij}^{l} \right)^2 \phi^2 \left(h_{j,a}^{l-1} \right) \right)}} \tag{A.2}$$

and

$$m_{ij} \sim N\left(0, \sigma_m^2\right) b_i \sim N\left(0, N_{l-1}\sigma_b^2\right) \tag{A.3}$$

$$\mathbb{E}\left[\left(h_{i,a}^{l}\right)^{2}\right]=\mathbb{E}\left[\left(\frac{\sum_{j}m_{ij}^{l}\phi\left(h_{j,a}^{l-1}\right)+b_{i}^{l}}{\sqrt{\sum_{j}\left(1-\left(m_{ij}^{l}\right)^{2}\phi^{2}\left(h_{j,a}^{l-1}\right)\right)}}\right)^{2}\right]=\frac{\mathbb{E}\left[\left(\sum_{j}m_{ij}^{l}\phi\left(h_{j,a}^{l-1}\right)+b_{i}^{l}\right)^{2}\right]}{N_{l-1}-\sum_{j}\left(m_{ij}^{l}\right)^{2}\phi^{2}\left(h_{j,a}^{l-1}\right)}$$
$$=\frac{\sum_{j}\sigma_{m}^{2}\mathbb{E}\phi^{2}\left(h_{j,a}^{l-1}\right)+N_{l-1}\sigma_{b}^{2}}{N_{l-1}\left(1-\frac{1}{N_{l-1}}\sum_{j}\left(m_{ij}^{l}\right)^{2}\phi^{2}\left(h_{j,a}^{l-1}\right)\right)}=\frac{N_{l-1}\sigma_{m}^{2}\mathbb{E}\phi^{2}\left(h_{j,a}^{l-1}\right)+N_{l-1}\sigma_{b}^{2}}{N_{l-1}\left(1-\sigma_{m}^{2}\mathbb{E}\phi^{2}\left(h_{j,a}^{l-1}\right)\right)}$$
$$=\frac{\sigma_{m}^{2}\mathbb{E}\phi^{2}\left(h_{j,a}^{l-1}\right)+\sigma_{b}^{2}}{1-\sigma_{m}^{2}\mathbb{E}\phi^{2}\left(h_{j,a}^{l-1}\right)} \tag{A.4}$$

Where, $\mathbb{E}\phi^{2}\left(h_{j,a}^{l-1}\right)$ can be written explicitly, taking into account that $h_{j,a}^{l-1}\sim N\left(0,q_{aa}\right)$:

$$\mathbb{E}\left[\phi^{2}\left(h_{j,a}^{l}\right)\right]=\int\mathcal{D}h_{j,a}^{l}\phi^{2}\left(h_{j,a}^{l}\right)=\int dh_{j,a}^{l}\frac{1}{\sqrt{2\pi}\mathbb{E}\left[\left(h_{j,a}^{l}\right)^{2}\right]}\exp\left(-\frac{\left(h_{j,a}^{l}\right)^{2}}{2\mathbb{E}\left[\left(h_{j,a}^{l}\right)^{2}\right]}\right)\phi^{2}\left(h_{j,a}^{l}\right)$$
$$=\int dh_{j,a}^{l}\frac{1}{\sqrt{2\pi q_{aa}^{l}}}\exp\left(-\frac{\left(h_{j,a}^{l}\right)^{2}}{2q_{aa}^{l}}\right)\phi^{2}\left(h_{j,a}^{l}\right) \tag{A.5}$$

We can now perform the following change of variable:

$$z_{j,a}^{l}=\frac{h_{j,a}^{l}}{\sqrt{q_{aa}^{l}}} \tag{A.6}$$

Then:

$$\mathbb{E}\left[\phi^{2}\left(h_{j,a}^{l}\right)\right]=\frac{1}{\sqrt{2\pi q_{aa}^{l}}}\sqrt{q_{aa}^{l}}\int dz_{j,a}^{l}\exp\left(-\frac{\left(z_{j,a}^{l}\right)^{2}}{2}\right)\phi^{2}\left(\sqrt{q_{aa}^{l}}z_{j,a}^{l}\right)$$
$$=\frac{1}{\sqrt{2\pi}}\int dz\exp\left(-\frac{z^{2}}{2}\right)\phi^{2}\left(\sqrt{q_{aa}^{l}}z\right)$$
$$=\int\mathcal{D}z\phi^{2}\left(\sqrt{q_{aa}^{l}}z\right) \tag{A.7}$$

$$q_{aa}^{l}=\mathbb{E}\left[\left(h_{i,a}^{l}\right)^{2}\right]=\frac{\sigma_{m}^{2}\int\mathcal{D}z\phi^{2}\left(\sqrt{q_{aa}^{l-1}}z\right)+\sigma_{b}^{2}}{1-\sigma_{m}^{2}\int\mathcal{D}z\phi^{2}\left(\sqrt{q_{aa}^{l-1}}z\right)} \tag{A.8}$$

In the first layer, input neurons are not stochastic: they are samples drawn from the Gaussian distribution $x^{0}\sim N\left(0,q^{0}\right)$:

Correlation propagation

To determine the correlation recursion we start from its definition:

$$c_{ab}^{l} = \frac{q_{a,b}^{l}}{\sqrt{q_{aa}^{l} q_{bb}^{l}}}, \tag{A.9}$$

where q_{ab}^{l} represents the covariance of the pre-activations $h_{i,a}^{l}$ and $h_{i,b}^{l}$, related to two distinct input signals and therefore defined as:

$$q_{ab}^{l} = \frac{1}{N_l} \sum_{i} h_{i,a}^{l} h_{i,b}^{l} = \mathbb{E}\left[h_{i,a}^{l} h_{i,b}^{l} \right]. \tag{A.10}$$

Replacing the pre-activations with their expressions provided in eq. (A.2) and taking advantage of the self-averaging argument, we can then write:

$$c_{ab}^{l} = \frac{\sigma_m^2 \mathbb{E}\left[\phi\left(h_{j,a}^{l-1} \right) \phi\left(h_{j,b}^{l-1} \right) \right] + \sigma_b^2}{\sqrt{q_{aa}^{l} \left(1 - \sigma_m^2 \mathbb{E}\left[\phi^2\left(h_{j,a}^{l-1} \right) \right] \right)} \sqrt{q_{bb}^{l} \left(1 - \sigma_m^2 \mathbb{E}\left[\phi^2\left(h_{j,b}^{l-1} \right) \right] \right)}}. \tag{A.11}$$

At this point, given that q_{aa}^{l} and q_{bb}^{l} quite quickly approach the fixed point, we can conveniently assume $q_{aa}^{l} = q_{bb}^{l}$. Moreover, exploiting eq.(A.8), we can finally write the expression for the correlation recursion:

$$c_{ab}^{l} = \frac{1 + q_{aa}^{l}}{q_{aa}^{l}} \frac{\sigma_m^2 \mathbb{E}\left[\phi\left(h_{j,a}^{l-1} \right) \phi\left(h_{j,b}^{l-1} \right) \right] + \sigma_b^2}{1 + \sigma_b^2}. \tag{A.12}$$

A.1.2 Derivation of the slope of the correlations at the fixed point

To check the stability at the fixed point, we need to compute the slope of the correlations mapping from layer to layer at the fixed point:

$$\begin{aligned} \chi|_{q_*} &= \frac{\partial c_{ab}^{l}}{\partial c_{ab}^{l-1}} \\ &= \frac{1 + q_*}{q_*} \frac{\sigma_m^2}{1 + \sigma_b^2} \frac{\partial}{\partial c_{ab}^{l-1}} \mathbb{E}\left[\phi\left(h_{j,a}^{l-1} \right) \phi\left(h_{j,b}^{l-1} \right) \right] |_{q_*} , \\ &= \frac{1 + q_*}{q_*} \frac{\sigma_m^2}{1 + \sigma_b^2} \frac{\partial}{\partial c_{ab}^{l-1}} \int \mathcal{D}z_a \mathcal{D}z_b \phi\left(u_a \right) \phi\left(u_b \right) |_{q_*} \end{aligned} \tag{A.13}$$

where we get rid of σ_b because independent from c_{ab}^{l-1}. Replacing the definition of u_a and u_b provided in the continuous model, we can explicitly compute the derivative with respect to c_{ab}^{l-1}:

$$\chi = \frac{1+q_*}{q_*}\frac{\sigma_m^2}{1+\sigma_b^2}\left(A-B\right), \tag{A.14}$$

where we have defined A and B as:

$$\begin{aligned} A &= \sqrt{q_*}\int \mathcal{D}z_a\mathcal{D}z_b\phi\left(\sqrt{q_{aa}^{l-1}}z_a\right)\phi'\left(\sqrt{q_{bb}^{l-1}}\left(c_{ab}^{l-1}z_a+\sqrt{1-\left(c_{ab}^{l-1}\right)^2}z_b\right)\right)z_a \\ B &= \sqrt{q_*}\int \mathcal{D}z_a\mathcal{D}z_b\phi\left(\sqrt{q_{aa}^{l-1}}z_a\right)\phi'\left(\sqrt{q_{bb}^{l-1}}\left(c_{ab}^{l-1}z_a+\sqrt{1-\left(c_{ab}^{l-1}\right)^2}z_b\right)\right)\frac{c_{ab}^{l-1}}{\sqrt{1-\left(c_{ab}^{l-1}\right)^2}}z_b. \end{aligned} \tag{A.15}$$

We can focus on B first. Integrating by parts over z_b we get:

$$B = \sqrt{q_*}\int \mathcal{D}z_a\mathcal{D}z_b\phi\left(\sqrt{q_{aa}^{l-1}}z_a\right)\frac{\partial}{\partial z_a}\phi'\left(\sqrt{q_{bb}^{l-1}}\left(c_{ab}^{l-1}z_a+\sqrt{1-\left(c_{ab}^{l-1}\right)^2}z_b\right)\right). \tag{A.16}$$

Then, integrating by parts over z_a, we the get:

$$\begin{aligned} B = \sqrt{q_*}\int \mathcal{D}z_a\mathcal{D}z_b\phi\left(\sqrt{q_{aa}^{l-1}}z_a\right)\phi'\left(\sqrt{q_{bb}^{l-1}}\left(c_{ab}^{l-1}z_a+\sqrt{1-\left(c_{ab}^{l-1}\right)^2}z_b\right)\right)z_a+ \\ - q_*\int \mathcal{D}z_a\mathcal{D}z_b\phi'\left(\sqrt{q_{aa}^{l-1}}z_a\right)\phi'\left(\sqrt{q_{bb}^{l-1}}\left(c_{ab}^{l-1}z_a+\sqrt{1-\left(c_{ab}^{l-1}\right)^2}z_b\right)\right). \end{aligned} \tag{A.17}$$

Replacing A and B in eq. (A.14), we then obtain the closest expression for the stability at the variance fixed point, namely:

$$\chi|_{q_*} = \frac{1+q_*}{1+\sigma_b^2}\sigma_m^2\int \mathcal{D}z_a\mathcal{D}z_b\phi'\left(u_a\right)\phi'\left(u_b\right) \tag{A.18}$$

A.1.3 Variance depth scale

As pointed out in the main text, it should hold asymptotically that:

$$|q_{aa}^{l+1}-q_*| \sim \exp\left(-\frac{l+1}{\xi_q},\right) \tag{A.19}$$

with ξ_q defining the variance depth scale. To compute it we can expand over small perturbations around the fixed point, namely:

$$
\begin{aligned}
q_{aa}^{l+1} &= q_* + \epsilon^l \\
&= \frac{\sigma_m^2 \int \mathcal{D}z \phi^2 \left(\sqrt{q_* + \epsilon^l} z\right) + \sigma_b^2}{1 - \sigma_m^2 \int \mathcal{D}z \phi^2 \left(\sqrt{q_* + \epsilon^l} z\right)}.
\end{aligned} \tag{A.20}
$$

Expanding the square root for small ϵ^l, we can then write:

$$
q_{aa}^{l+1} \simeq \frac{\sigma_m^2 \int \mathcal{D}z \phi^2 \left(\sqrt{q_*} z + \frac{\epsilon^l}{2\sqrt{q_*}} z\right) + \sigma_b^2}{1 - \sigma_m^2 \int \mathcal{D}z \phi^2 \left(\sqrt{q_*} z + \frac{\epsilon^l}{2\sqrt{q_*}} z.\right)} \tag{A.21}
$$

We can now expand the activation function ϕ around small perturbations and then computing the square getting rid of higher order terms in ϵ^l, thus finally obtaining:

$$
q_{aa}^{l+1} \simeq q_* + \frac{1 + q_*}{\sqrt{q_*}} \frac{\sigma_m^2 \int \mathcal{D}z \phi\left(\sqrt{q_*} z\right) \phi'\left(\sqrt{q_*} z\right) z}{1 - \sigma_m^2 \int \mathcal{D}z \phi^2 \left(\sqrt{q_*} z\right)} \epsilon^l \tag{A.22}
$$

Comparing this expression with the one in eq. (A.20), we can then write:

$$
\epsilon^{l+1} \simeq \frac{1 + q_*}{\sqrt{q_*}} \frac{\sigma_m^2 \int \mathcal{D}z \phi\left(\sqrt{q_*} z\right) \phi'\left(\sqrt{q_*} z\right) z}{1 - \sigma_m^2 \int \mathcal{D}z \phi^2 \left(\sqrt{q_*} z\right)} \epsilon^l. \tag{A.23}
$$

Integrating by parts over z, we then obtain:

$$
\epsilon^{l+1} \simeq \left[(1 + q_*) \frac{\sigma_m^2 \int \mathcal{D}z \phi'\left(\sqrt{q_*} z\right) \phi'\left(\sqrt{q_*} z\right) + \int \mathcal{D}z \phi''\left(\sqrt{q_*} z\right) \phi\left(\sqrt{q_*} z\right)}{1 - \sigma_m^2 \int \mathcal{D}z \phi^2 \left(\sqrt{q_*} z\right)} \right] \epsilon^l. \tag{A.24}
$$

Given that it holds eq. (A.8), and noticing that χ evaluated at the correlation fixed point $c_* = 1$ is given by:

$$
\chi|_{c_*=1} = \frac{\sigma_m^2}{1 + \sigma_b^2} (1 + q_*) \int \mathcal{D}z \left[\phi'\left(\sqrt{q_*} z\right)\right]^2, \tag{A.25}
$$

we can finally get:

$$
\epsilon^{l+1} \simeq \left[\chi|_{c_*=1} + \frac{\sigma_m^2 (1 + q_*)}{1 + \sigma_b^2} \int \mathcal{D}z \phi''\left(\sqrt{q_*} z\right) \phi\left(\sqrt{q_*} z\right) \right] \frac{\epsilon^l}{1 + q_*}. \tag{A.26}
$$

Given that we expect (A.19) to hold asymptotically, that is:

$$
\epsilon^{l+1} \sim \exp\left(-\frac{l+1}{\xi_q}\right), \tag{A.27}
$$

we can finally obtain the variance depth scale:

$$
\xi_q^{-1} = \log(1 + q_*) - \log\left(\chi|_{c_*=1} + \frac{\sigma_m^2 (1 + q_*)}{1 + \sigma_b} \int \mathcal{D}z \phi''\left(\sqrt{q_*} z\right) \phi\left(\sqrt{q_*} z\right) \right). \tag{A.28}
$$

Appendix B

Signal propagation derivations for perturbed surrogates

B.1 Perturbed Gaussian surrogate: Stochastic neuron $\mathbb{E}\phi(h_i^\ell) = \tanh(h_i^\ell)$, SB weights

We first compute :

$$q_{aa}^l = \frac{1}{N_l}\sum_i (h_{i,a}^l)^2 = \mathbb{E}\left[(h_{i,a}^l)^2\right] \tag{B.1}$$

Where for us:

$$h_{i,a}^l = \frac{1}{\sqrt{N}}\sum_j m_{ij}^l \phi(h_{j,a}^{l-1}) + b_i^l + \epsilon_{i,a}^\ell \frac{1}{\sqrt{N}}\sqrt{\sum_j 1 - (m_{ij}^l)^2\phi^2(h_{j,a}^{l-1})} \tag{B.2}$$

we will use a different parameterisation for this study, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$, so that we can study the effect of the perturbation. If $\sigma_\epsilon^2 \to 0$ we have a deterministic continuous network (under the mean field model).

$$m_{ij} \sim N(0, \sigma_m^2) \tag{B.3}$$
$$b_i \sim N(0, \sigma_b^2) \tag{B.4}$$

$$\mathbb{E}\left[(h_{i,a}^l)^2\right] = \mathbb{E}\left[\left(\frac{1}{\sqrt{N}}\sum_j m_{ij}^l\phi(h_{j,a}^{l-1}) + b_i^l + \frac{1}{\sqrt{N}}\epsilon_{i,a}^\ell\sqrt{\sum_j 1 - (m_{ij}^l)^2\phi^2(h_{j,a}^{l-1})}\right)^2\right] \tag{B.5}$$
$$= \sigma_m^2\mathbb{E}\phi^2(h_{j,a}^{l-1}) + \sigma_b^2 + (1 - \sigma_m^2\mathbb{E}\phi^2(h_{j,a}^{l-1})) \tag{B.6}$$
$$= 1 + \sigma_b^2 \tag{B.7}$$

$$q_{ab}^l = \mathbb{E}\left[h_{i,a}^l h_{i,b}^l\right] \tag{B.8}$$
$$= \sigma_m^2\mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2 \tag{B.9}$$

$$c_{ab}^l = \frac{\sigma_m^2 \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2}{1+\sigma_b^2} \tag{B.10}$$

$c^* = 1$ if $\mathbb{E}\phi^2(h_{j,a}^{l-1}) = \frac{1}{\sigma_m^2}$.

$$\chi = \sigma_m^2 \mathbb{E}\phi'(h_{j,a}^{l-1})\phi'(h_{j,b}^{l-1}) \tag{B.11}$$

$\chi = 1$ if $\mathbb{E}\phi'(h_{j,a}^{l-1})\phi'(h_{j,b}^{l-1}) = \frac{1}{\sigma_m^2}$.
Perhaps there is a contradiction here...

B.2 Perturbed Gaussian surrogate: Continuous neuron $\phi() = \tanh()$, SB weights

We first compute :

$$q_{aa}^l = \frac{1}{N_l}\sum_i (h_{i,a}^l)^2 = \mathbb{E}\left[\left(h_{i,a}^l\right)^2\right] \tag{B.12}$$

Where for us:

$$h_{i,a}^l = \frac{1}{\sqrt{N}}\sum_j m_{ij}^l \phi(h_{j,a}^{l-1}) + b_i^l + \epsilon_{i,a}^\ell \frac{1}{\sqrt{N}}\sqrt{\sum_j (1-(m_{ij}^l)^2)\phi^2(h_{j,a}^{l-1})} \tag{B.13}$$

$$m_{ij} \sim N(0, \sigma_m^2) \tag{B.14}$$

$$b_i \sim N(0, \sigma_b^2) \tag{B.15}$$

$$\mathbb{E}\left[(h_{i,a}^l)^2\right] = \mathbb{E}\left[\left(\frac{1}{\sqrt{N}}\sum_j m_{ij}^l \phi(h_{j,a}^{l-1}) + b_i^l + \frac{1}{\sqrt{N}}\epsilon_{i,a}^\ell \sqrt{\sum_j (1-(m_{ij}^l)^2)\phi^2(h_{j,a}^{l-1})}\right)^2\right] \tag{B.16}$$

$$= \sigma_m^2 \mathbb{E}\phi^2(h_{j,a}^{l-1}) + \sigma_b^2 + \mathbb{E}\phi^2(h_{j,a}^{l-1}) - \sigma_m^2 \mathbb{E}\phi^2(h_{j,a}^{l-1}) \tag{B.17}$$

$$= \sigma_b^2 + \mathbb{E}\phi^2(h_{j,a}^{l-1}) \tag{B.18}$$

$$q_{ab}^l = \mathbb{E}\left[h_{i,a}^l h_{i,b}^l\right] \tag{B.19}$$

$$= \sigma_m^2 \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2 \tag{B.20}$$

$$c_{ab}^l = \frac{\sigma_m^2 \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2}{\sigma_b^2 + \mathbb{E}\phi^2(h_{j,a}^{l-1})} \tag{B.21}$$

$c^* = 1$ if $\sigma_m^2 = 1$!

$$\chi = \sigma_m^2 \mathbb{E}\phi'(h_{j,a}^{l-1})\phi'(h_{j,b}^{l-1}) \tag{B.22}$$

$\chi = 1$ if $\mathbb{E}\phi'(h_{j,a}^{l-1})\phi'(h_{j,b}^{l-1}) = \frac{1}{\sigma_m^2}$

B.3 Perturbed Gaussian surrogate: Continuous weights, stochastic neuron

We first compute :

$$q_{aa}^l = \frac{1}{N_l}\sum_i (h_{i,a}^l)^2 = \mathbb{E}\left[\left(h_{i,a}^l\right)^2\right] \tag{B.23}$$

Where for us:

$$h_{i,a}^l = \frac{1}{\sqrt{N}}\sum_j w_{ij}^l \phi(h_{j,a}^{l-1}) + b_i^l + \epsilon_{i,a}^\ell \frac{1}{\sqrt{N}}\sqrt{\sum_j (1-\phi^2(h_{j,a}^{l-1}))(w_{ij}^l)^2} \tag{B.24}$$

$$w_{ij} \sim N(0, \sigma_m^2) \tag{B.25}$$
$$b_i \sim N(0, \sigma_b^2) \tag{B.26}$$

$$\mathbb{E}\left[(h_{i,a}^l)^2\right] = \mathbb{E}\left[\left(\frac{1}{\sqrt{N}}\sum_j w_{ij}^l \phi(h_{j,a}^{l-1}) + b_i^l + \epsilon_{i,a}^\ell \frac{1}{\sqrt{N}}\sqrt{\sum_j (1-\phi^2(h_{j,a}^{l-1}))(w_{ij}^l)^2}\right)^2\right] \tag{B.27}$$
$$= \sigma_w^2 \mathbb{E}\phi^2(h_{j,a}^{l-1}) + \sigma_b^2 + \sigma_w^2(1-\mathbb{E}\phi^2(h_{j,a}^{l-1})) \tag{B.28}$$
$$= \sigma_b^2 + \sigma_w^2 \tag{B.29}$$

$$q_{ab}^l = \mathbb{E}\left[h_{i,a}^l h_{i,b}^l\right] \tag{B.30}$$
$$= \sigma_w^2 \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2 \tag{B.31}$$

$$c_{ab}^l = \frac{\sigma_w^2 \mathbb{E}\phi(h_{j,a}^{l-1})\phi(h_{j,a}^{l-1}) + \sigma_b^2}{\sigma_w^2 + \sigma_b^2} \tag{B.32}$$

$c^* = 1$ if $\mathbb{E}\phi^2(h_{j,a}^{l-1}) = 1$.

$$\chi = \sigma_w^2 \mathbb{E}\phi'(h_{j,a}^{l-1})\phi'(h_{j,b}^{l-1}) \tag{B.33}$$

$\chi = 1$ if $\mathbb{E}\phi'(h_{j,a}^{l-1})\phi'(h_{j,b}^{l-1}) = \frac{1}{\sigma_w^2}$.