The Rôle of Biases in Neural Network Models

Ansgar Heinrich Ludolf West

Doctor of Philosophy
University of Edinburgh
1997

Acknowledgments

This Ph.D. has been a long journey with some minor complications and there are many people to thank that this journey has come to a (successful) end. First of all, I would like to express my gratitude towards my main supervisor, David Saad. His constant encouragement and advice have been of invaluable help during the last years. His depth of knowledge of the field and his insight have often proved invaluable. At the end, he helped me to pull it all through, reading many drafts of papers and thesis chapters and even providing some financial assistance once my grant ran out.

I am also extremely grateful to my second supervisor, Chris Bishop. After the Neural Network Group of the Physics Department at Edinburgh University ceased to exist at the end of my second year, he provided me with the opportunity to stay with the Neurocomputing Research Group at Aston University, opening the door to an academically very stimulating environment and making funds available to cover both living expenses in the third year and the costs for attending two NIPS conferences. I will also be forever thankful to him and his wife Jenna for opening their home to me during the first few weeks in Birmingham. Furthermore, I would like to thank Alastair Bruce, who has been my guardian in Edinburgh, whenever I had difficulties with computing access or funding problems. In this respect, I also have to thank the Condensed Matters Group in Edinburgh, who gave me access to their machines after the Theory Group felt unable to accommodate me.

During my Ph.D., I was also inspired, academically and/or personally, by many scientist I met during conferences and other occasions such as Ronnie Meir, Ton Coolen, Jonathan Shapiro, Andreas Engel, and Sara Solla; I thank them all. I especially have to mention Andreas Engel, with whom I had extensive discussions via email over some of the issues covered in Chapter 3, and Ian Nabney, whose universal approximation proof for the soft-committee machine partially sparked off the idea for the conducting the research in Chapter 5. Further special gratitude is reserved for Sara Solla and Bennie Laudrup, for inviting me to stay at CONNECT, Copenhagen — unfortunately this stay did not materialize into any tangible results. However, the discussion with people at CONNECT (besides the above mentioned) such as John Hertz, Ole Winther, and Adam Prügel-Bennett have made the stay a success. I would also like to acknowledge financial support by the Physics Department of Edinburgh University through a Dewar fellowship, by the EPSRC through covering the University fees, and by the EU through travel support under grant No. ERB CHRX-CT92-0063.

During my first two years in Edinburgh, the small but very productive NN group provided a friendly, collaborative yet relaxed research environment. Special thanks

during this time goes to my fellow Ph.D. students and friends Pete, David B., and Glenn, who were always willing to share their insights and with whom I had also many memorable moments off work, including evenings together with David Saad and his wife Chris watching slides and enjoying Chris' cooking.

During my last two years in Birmingham, I have to thank all members of the Neurocomputing Group for their hospitality, especially I would like to mention my fellow Ph.D. mates Markus, Mehdi, Neil, Francesco, and Ragnar, who were always a good laugh. Special thanks goes to David B., who accompanied me from Edinburgh, for his friendship, reading parts of this thesis, and many enlightening discussions; Pete, for some great "supervisions" via phone and email; David D., with whom I shared a house and who always brought me down to earth from the heights of theoretical physics.

I also have to thank my friends both in Great Britain and in Germany for their affection and their loyalty. I would especially like to mention Mathias & Bernd who came all the way to surprise me on my 30. birthday, and Helen & Cath, with whom I spend several relaxing and fun weekends in London.

Finally, my greatest thank has to go to my family for their unflagging support, morally and financially, and whose patience never wavered notwithstanding my consistent underestimates of the time required to complete this thesis.

Publications

Some of the work presented in this thesis has been published or submitted for publication as listed below.

- A. H. L. West and D. Saad, *Adapative back-propagation in on-line learning of multilayer networks.* In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems*, (MIT Press, Cambridge, MA) Vol. 8, p. 323–329 (1996).

- A. H. L. West and D. Saad, *Threshold induced phase transitions in perceptrons. Journal of Physics A: Mathematics and General* Vol. **30**, No. 10, p. 3471-3496 (1997).

- A. H. L. West, D. Saad, and I. T. Nabney, *The learning dynamics of a universal approximator.* In M. C. Mozer, M. I. Jordan, and T. Petsche (Eds.), *Advances in Neural Information Processing Systems*, (MIT Press, Cambridge, MA) Vol. 9, p. 288–294 (1997).

- A. H. L. West and D. Saad, *The storage capacity of the upstart algorithm.* In S. W. Ellacott, J. C. Mason, and I. J. Anderson (Eds.), *Mathematics of Neural Networks: Models, Algorithms and Applications*, (Kluwer Academic Publishers, Boston) *Operations Research/Computer Science Interfaces Series* Vol. 65, Chap. 65, p. 372–377

- A. H. L. West and D. Saad, *On-line learning with adaptive back-propagation in two-layer networks. Physical Review E* Vol. **56**, p. 3426–3445 (1997).

- A. H. L. West and D. Saad, *The rôle of biases in on-line learning of two-layer networks.* Accepted for publication in *Physical Review E* (1997).

- A. H. L. West and D. Saad, *The statistical mechanics of constructive algorithms.* Submitted to *Journal of Physics A: Mathematics and General* (1997).

Nomenclature

It has been difficult to find a notation which is internally consistent throughout this thesis, which essentially consists of a variety of work I have conducted throughout the last four years related only by a certain but not entire focus on the rôle of biases in neural network models. I have adopted some principles, which should hold for most of the symbols used throughout the thesis although exceptions may occur, usually in order to comply with the notation prevalent in the field.

Bold *italic* letters (both upper and lower case), for example $\boldsymbol{W}$, $\boldsymbol{\xi}$, or $\boldsymbol{x}$ are used to denote vectors, whereas bold upper-case *Roman* letters, such as $\mathbf{M}$, denote matrices, with the identity matrix written as $\mathbf{1}$. Related variables are labelled by sub- or super-script *Roman* letters, as λ^{opt} is the optimum of λ. In order to stress specific transformations, relations or properties, accents are used, such as $\hat{q}$ for the to q conjugate order parameter or $R^{\star}$ for a fixed point of R. Elements of vectors or matrices on the other hand are indexed by sub- or super-script *italic*, as x_i, or in the case of replicas and patterns *Greek* letters, as $Q_{\sigma\rho}$ or ζ^{μ}.

In both text and mathematical equations, delimiter, such as braces, brackets, and parenthesis have usually been used in the following order: $\{[(\ )]\}$. However, this ordering has been ignored if the brackets have a special meaning. For instance, $\langle G \rangle_{\zeta}$ ($\langle\!\langle Z \rangle\!\rangle_{\boldsymbol{\xi}}$) denotes to a (quenched) average over variables or $\{\boldsymbol{W}_i\}$ denotes to a set of variables.

For summation over indices, the Einstein convention of summing over repeated indices is usually used. For vector and matrix notation in linear algebra, I have adopted the style most common, vectors are considered column vectors with the corresponding row vectors denoted by a superscript $^{\mathrm{T}}$ indicating the transpose and used those for vector-matrix multiplications besides the simple scalar product of two vectors, where the notation $\boldsymbol{W}_i \cdot \boldsymbol{\xi}$ was used. Similarly, $\mathbf{M}^{\mathrm{T}}$ denotes the transpose of a matrix $\mathbf{M}$. The notation $\mathbf{M} = (M_{ij})$ is used to denote the fact that the matrix $\mathbf{M}$ has elements M_{ij}, whereas $M_{ij} = (\mathbf{M})_{ij}$ is used for the ij element of a matrix $\mathbf{M}$. The norm of a vector x is denoted by $||x||$, while the magnitude of a scalar x is denoted by $|x|$. The determinant of a matrix $\mathbf{M}$ is written as $|\mathbf{M}|$.

Probabilities and probability distributions are denoted by P. Conditioning on a variable is expressed as $y|x$, when y is dependent of x, e.g., conditional probabilities are written as $P(y|x)$. For a function f dependent on parameters $\boldsymbol{\Omega}$, I write $f(\boldsymbol{\xi}; \boldsymbol{\Omega})$.

Furthermore, the notation $\mathcal{O}(N)$ is used to denote that a quantity is *of order* N, i.e., given two functions $f(N)$ and $g(N)$, we say that $f = \mathcal{O}(g)$ if $f(N) < Cg(N)$ for $N \to \infty$, where C is a constant. Similarly, we will say that $f \simeq g$ if the ratio

$f(N)/g(N) \to 1$ for $N \to \infty$. The convention was used that log $\equiv$ ln, if no specific base for the logarithm is given.

Special Symbols

Although most symbols are defined when they first occur in the thesis (or often even a chapter), a list of those symbols used for the most frequently occuring quantities is given below. The usage of a few (some occuring frequently) symbol varies between chapters. This is rather unfortunate, but the limit of the amount of symbols available makes this inevitable. Most conflicting symbols either have been used in order to be consistent with the literature or they are used to provide a more compact notation for some of the more tedious equations. For the most notable and frequently used such symbols, the conflicting definition are given below, with the chapter to which the definition applies in parenthesis.

Roman letters:

W	network (neuron, student) weight
E, V	error (energy) function
f	free energy per degree of freedom (3,4), general function (5,6)
h	network (neuron, student) activation
K	number of student hidden units
N	input dimension
M	magnetization (3,4), number of teacher hidden units (5,6)
p	number of patterns (in some cases probabilities)
P	Probability (distribution, density)
Q	student-student overlap
R	student-teacher overlap
s	entropy per degree of freedom
T	temperature (3,4), teacher-teacher overlap (5,6)
x	student activation (also general function variable)
y	teacher activation
Z	partition function

Greek letters:

α	example load
β	inverse ["formal" (6)] temperature
ϵ	error measure or rate
ζ	target variable or teacher output
η	learning rate
θ, ϑ	network (neuron, student) bias (threshold)
κ	pattern stability
λ	eigenvalue of a matrix (5,6)
μ	pattern index
ξ	input pattern variable

ϱ	teacher bias (5,6)
σ	network (neuron, student) output (also replica index and variances)
$\mathbf{\Omega}$	network (neuron, student) parameters

Special "functions"

$$\delta_{ij} = \begin{cases} 1 & \text{for } i = j; \\ 0 & \text{for } i \neq j, \end{cases} \qquad \text{(Kronecker delta symbol)}$$

$$\int_{-\infty}^{\infty} dx\, \delta(x - x_0) f(x) = f(x_0) \qquad \text{[defines Dirac delta-"function" } \delta(x)\text{]}$$

$$\Theta(x) = \begin{cases} 1 & \text{for } x > 0; \\ 0 & \text{for } x < 0, \end{cases} \qquad \text{(Heaviside step-function)}$$

$$\operatorname{sgn}(x) = \begin{cases} 1 & \text{for } x \geq 0; \\ -1 & \text{for } x < 0, \end{cases}$$

$$\tanh(x) = \frac{\mathrm{e}^{x} - \mathrm{e}^{-x}}{\mathrm{e}^{x} + \mathrm{e}^{-x}}$$

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x dt \exp(-t^2)$$

$$Dx = \frac{dx}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right)$$

$$H(x) = \int_x^{\infty} Dt = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$$

See Appendix A for some frequently used mathematical identities.

Abstract

Neural networks have attracted considerable interest in recent years due to their ability to learn complicated maps from examples, an ability termed universal approximation. Statistical mechanics has over the years proved to provide powerful tools for analysing the capabilities and learning behaviour of neural network models in the average case, which is distinguished from the PAC (Probably Approximately Correct) framework of computational learning theory which studies worst case scenarios. This thesis analyses the rôle of biases, also termed thresholds in binary systems, in the learning process of neural network models within a statistical mechanics framework.

In order to make progress in the understanding of neural networks, the physics community has started out in the study of simple models, especially the binary and linear perceptrons, which consist of a single layer and can therefore only implement linearly separable rules and linear mappings respectively. Another simplification, made in many works, was to ignore the threshold or bias in order to facilitate the calculation. The underlying assumption was usually either that the bias can be treated as just another weight, albeit with a constant input, or that the dynamics are most likely dominated by the ordinary weights in the thermodynamic limit of infinite input dimension usually employed. This thesis questions the validity of these assumptions for two of the most commonly studied problems in the neural network community: the network storage capacity and supervised learning problems.

In the capacity problem one calculates the number of examples with random output that can be stored perfectly on average by a certain network architecture, which is related to the VC (Vapnik-Chervonenkis) dimension in computational learning theory, and how many errors a network will typically make once its capacity has been saturated. For this problem, we have studied two cases: the binary perceptron above saturation and the capacity of architectures generated by constructive algorithms. For the binary perceptron, we find that the threshold induces a phase transition between a solution with zero threshold below and a solution with finite threshold above a certain number of examples, if the examples are required to be stored with a non-zero stability.

The capacity problem for multi-layer networks has proven especially elusive. Our calculation of the capacity of multi-layer networks built by constructive algorithms relies heavily on the existence of biases in the basic building block, the binary perceptron. It is the first time where the capacity is explicitly evaluated for large networks and finite stability. One finds that the constructive algorithms studied, a tiling-like algorithm and variants of the upstart algorithm, do not saturate the known Mitchison-Durbin bound.

In supervised learning, a student network is presented with training examples in the form of input-output pairs, where the output is generated by a teacher network. The central question to be answered is the relation between the number of examples presented and the typical performance of the student in approximating the teacher rule, which is usually termed generalisation. The influence of biases in such a student-teacher scenario has been assessed for the two-layer soft-committee architecture, which is a universal approximator and already resembles applicable multi-layer network models, within the on-line learning paradigm, where training examples are presented serially.

One finds that adjustable biases dramatically alter the learning behaviour. The sub-optimal symmetric phase, which can easily dominate training for fixed biases, vanishes almost entirely for non-degenerate teacher biases. Furthermore, the extended model exhibits a much richer dynamical behaviour, exemplified especially by a multitude of (attractive) suboptimal fixed points even for realizable cases, causing the training to fail or to be severely slowed down. In addition, in order to study possible improvements over gradient descent training, an adaptive back-propagation algorithm parameterised by a "temperature" is introduced, which enhances the ability of the student to distinguish between teacher nodes. This algorithm, which has been studied in the various learning stages, provides more effective symmetry breaking between hidden units and faster convergence to optimal generalisation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Neural networks have generated exceptional interest in recent years in such diverse fields as medical neuroscience, biology, cognitive science, computer science, mathematics, and physics. The diversity of the encompassing fields regrettably created a Babel of languages in the beginning which continues to a lesser extent today. Nevertheless, neural network research has emerged as a truly multidisciplinary field studying adaptive learning from examples in natural and artificial systems.

Although many of todays artificial neural networks (ANN) may have more in common with statistical models than with plausible models of biological neural tissue, the fact remains that they have been inspired by the relative ease with which the human brain solves complex tasks such as handwritten character, speech, or face recognition, which pose great difficulty to conventional von-Neumann computers. This is even more intriguing considering the fact that the microscopic processing units of the brain, *neurons*, are simple, slow, unreliable, and noisy elements performing only very basic computations in comparison to the complex, fast, and deterministic CPUs of present-day computers. The most plausible explanation for the computational superiority of the brain (for most problems beyond those consisting of simple number crunching) must therefore lie in the vast number of strongly interacting neurons it contains (estimated at around 10^{11} neurons, where each neuron is connected via synaptic couplings to around 10^4 of its neighbours).

ANNs essentially try to imitate these features of the brain. The simplest ANN, the *perceptron*, which is also a very abstract neuron model, simply performs a *weighted* sum of all input signals adjusted by a *bias* or *threshold*, and outputs a result-dependent signal. Through combining a collection of these perceptrons, the aim is to build network

architectures which can perform similar tasks to the brain.

Historically, two generic architectures have developed: *recurrent* architectures, which are dynamical systems and in which the information is processed back and forth between neurons until they settle in a stable state on a limit cycle (dynamical attractor), and *feedforward* architectures, which are deterministic systems and in which the information is processed unidirectionally from the input to the output neurons. Networks of this latter architecture, where the information is fed through one or several hidden processing layers are usually called multilayer perceptrons (MLPs) as they are an extension of (single-layer) perceptrons.

The resulting networks (in some implementations exceeding 10^4 neurons and 10^6 weights) can today solve such diverse tasks as handwritten character recognition (Le Cun et al. 1989), playing backgammon (Tesauro 1990) or controlling the magnetic plasma in an experimental fusion reactor (Bishop et al. 1993) and are on the verge of becoming an alternative computing paradigm — neurocomputing.

1.2 Neural networks — just that little bit different

Besides the fact that neural networks (NN), both artificial and natural, process information in a highly parallel manner in comparison to the mainly serial information processing in traditional computers, the main difference between these two computing paradigms is that NNs obviates the need for a detailed program — humans learn from examples provided to them and the feedback they receive from the environment. Hebb (1949) suggested that learning consists of the adjustment of the strength of the synaptic couplings between neurons in response to pattern stimuli. ANN imitate this procedure by adjusting their internal parameters according to an error signal representing the mismatch between actual and desired responses on a set of examples, the *training set*, which is usually given in the form of input-output pairs.

Unlike a computer program, where all procedures have to be made explicit, the procedures in the trained NN are encoded in all its parameters. This distribution of knowledge makes NNs very fault tolerant in comparison to conventional computers, even if some weights or even whole neurons are removed (thousands of neurons die in our brain every day) a neural network still functions almost identically, whereas a single wrong bit can make a computer (program) crash. On the other hand, since the knowledge of the ANN is encoded in its parameters, it makes it difficult for a human to understand what rules it actually follows.

In principle, there are many ways how to categorize the type of problems neural networks can learn, here we will coarsely group them into two categories:

Memory tasks: In some cases, such as learning the vocabulary of another language, NNs are trained to memorize a large amount of data. A further aim of training is that the NN is able to *restore* a memorized pattern, if it is presented with a corrupted version, such as the correction of a misspelled word. These abilities combined are usually termed *associative memory*.

Generalization tasks: In most cases, such as the classification of handwritten characters, the aim of training a *student* network is to avoid simple memorization (although that might well be how humans proceed initially), but to generalize[1], i.e., to extract the underlying rules, representing an unknown functional relationship between inputs and output, from a set of examples. The goal is to be able to predict the output of hitherto unseen examples. The preferred architecture for such *supervised learning* tasks is currently feedforward.

Both type of problems are hard to solve within the traditional computing paradigm and ANNs provide a valid and often superior alternative approach. In traditional computing memory (RAM), memory access is very fast but traditionally only by address only, leading to serious errors if an address bit is corrupted. Although attempts are being made to make memory *content addressable*, this is a very challenging problem and ANN provide a useful alternative. Similarly, in generalization problems, such as handwritten character recognition, rules are often so complex, that they are not compactly expressible in terms of a programming language.

1.3 The need for a theoretical framework

Although ANN have come a long way since the study of simple perceptrons and adalines in the 1950's and 60's, there are still many basic issues, which are not well understood. Furthermore, it remains to develop truly intelligent ANNs with similar abilities as the human brain. Although there is some merit in empirical studies and subsequent heuristic rule development, a thorough theoretical understanding of machine (and human) learning may be able to accelerate progress towards this goal immensely.

In view of the vast problems plaguing scientists attempts to model and understand the inner workings of biological neural networks, the difficulties in understanding the much simpler ANN models may seem minute. However, questions revolving around the three main issues involved in using ANNs — training, capacity, and generalization — are, in many cases, still not answered satisfactorily or open to debate.

[1]We usually use the term training or learning for the suitable adjustment of free parameters of the network, whereas generalization is reserved for the ability to predict the output of new inputs.

1.3.1 Training issues

Training an ANN is necessary for anybody that wants to use an ANN either as an associative memory or as a predictor. If the network cannot reproduce the training examples with some sort of accuracy, there is no point of using it. In fact the problem of finding suitable training algorithms for ANN models was so severe that for many years no algorithm was known that was able to train network architecture more complicated than simple perceptrons. The book by Minsky and Papert (1969), proving the severe limitation of mappings simple perceptrons can implement, subdued the field for a decade.

The introduction of the powerful idea of a training error (Hopfield 1982) and the use of networks which perform smooth mappings (i.e., differentiable with respect to their parameters) enabled the formulation of training as an optimization problem for which well established methods exist. The problem of training feed-forward networks with hidden layers has been significantly alleviated[2] by the back-propagation algorithm (Werbos 1974; Rumelhart et al. 1986a) and subsequent developments (Bishop 1995). Although, there remain considerable problems and open issues, such as

- Almost all algorithms have "fiddle parameters" (e.g., learning rate) and their optimal settings are in general unknown.
- Training is prone to getting stuck in bad local minima, being attracted to fixed points, or being severly slowed down by flat areas in the energy surface. This behaviour may depend on the "fiddle parameters", parameter initialization, or the algorithm used.
- One can either train by updating the network parameter after the presentation of a single (*on-line learning*) or all examples (*batch learning*) and there is some argument over which paradigm to prefer[3].
- Although practical training algorithms are known, they are in many cases still slow and the design of better, or even in some sense optimal, error functions or algorithms is highly desirable.

For recurrent networks the problems are particularly severe, since practical training algorithm, i.e., algorithms with acceptable training time, have still to be devised for many architectures such as Boltzmann machines (Hinton and Sejnowski 1986).

[2]It could be argued that the emergence of fast traditional computers that can actually implement the training of these models was also of significant importance.

[3]Batch learning seems generally faster, but also seems to be more prone to local minima.

1.3.2 Capacity issues

To solve memory tasks with ANN, the main issue of interest is the size and the architecture necessary to store a certain amount of information. Furthermore one would like to know the efficiency of architectures, i.e., how many patterns it can store for a fixed size in comparison to other models. Other issues concern the size of the basin of attraction of the stored patterns, i.e., how much a pattern can deviate from its stored version and is still restorable, and the number of errors a network is going to make once its memory capacity is saturated. Finally, the memory capacity of an ANN is related to the number of examples necessary to achieve valid generalization (see below) and is therefore a relevant quantity also for generalization tasks.

1.3.3 Generalization issues

Using ANNs to solve generalization problems is probably their most common use and the variety of issues that have arisen over the years reflects how much the field has matured. Initially, many scientists were absorbed with the problem of training networks and achieving a low error on the training set. Today it is generally accepted, that the performance measure to be minimized should be the *test error*, i.e., the expected error on an unknown example[4]. The difference between the two comes from the fact, that the training data are usually corrupted by noise, e.g., due to measurement errors, and the student network should not memorize the noise but generalize to the underlying rule.

This difficulty can be tackled heuristically by training different networks on a training set and then testing them on yet unseen data set aside and choosing the one with the lowest empirical test error. This may be a fix for this particular data set and this particular problem, however this is not an approach that can provide informed answers to the issues arising from the above observations such as

- The number of training examples typically necessary to achieve a small deviation between the training and the test error for a network of fixed size and architecture, i.e., when does the network stop memorizing and start generalizing. The heuristic approach is to demand at least as many examples as free parameters in the network.

- The size and architecture of a network which compromises optimally between its

[4]This is typically approximated by the empirical test error, a sample estimate of this expectation based on examples left aside during the training process.

desirable ability to approximate and its undesirable ability to memorize. Historically, three approaches have been employed. Growing or shrinking the network during training or penalizing the size of the parameters.

Informed answers to these questions can be provided in a theoretical framework, in which the conclusions are independent of the particular instance of the data set or the particular rule to be learnt. Ideally, such a framework would be powerful enough to objectively analyse methods employed in practice.

1.4 What's physics got to do with it

The reason why physics can provide a natural theoretical framework in which these issues can be addressed lies in the large number of interacting neurons involved and the non-linearity of their interactions. Statistical physics provides us with tools which can compactify the laws governing the microscopic behaviour of many particles, such as spins in a magnetic material or atoms in a gas, into a macroscopic description depending only on a few variables, usually referred to as *order parameters*, such as the magnetization of a magnetic substance or the volume of a gas.

As with every physical theory, its strength and/or usefulness depends on several factors. Initially, a mathematical model describing the microscopic details has to be defined. Similar to the simplifications made to real gases or natural magnetic materials which result in the idealized gas and Ising spin models studied in statistical mechanics, it is necessary to limit the complexity of the neural network models studied to the bare essentials. Only those microscopic degrees of freedom and associated dynamical laws which are actually relevant for the emergence of macroscopically observed phenomena are included. These simplifications in ANN models[5] take place not at the stage of modelling individual neurons (since these are already idealised) but in the interactions between neurons, which usually have to be restricted to simplified architectures in order to make the models solvable.

For feedforward networks, which will be our concern, the simplest architecture is just a single neuron, the perceptron. The breakthrough in the ability to analyse the behaviour of the perceptron was achieved by the classic paper by Gardner (1988), in which the capacity of the perceptron was calculated. This solution required some more mathematical approximations such as the thermodynamic limit of infinite systems size,

[5]For biological neurons, however, it is self-evident that, for example, the inclusion of all the intricate details of a biological neuron may make a model biologically more plausible but introduce so many degrees of freedom and complicated interactions that such a model is virtually unsolvable.

mean-field theory, solving integrals by saddle-point methods and the replica technique, all more or less standard techniques developed and used over many years for spin glasses (large disordered interacting spin models). Furthermore, a set of relevant macroscopic order parameters was introduced in order to reduce the number of variables as described above. Instead of replacing the spin variables by their average magnetization, as in ferromagnets, Gardner used statistics of the perceptron parameters, which in this case emerged naturally in the course of the calculation. That the resulting mathematical theory can actually reproduce the observed macroscopic properties is the yardstick for any theory and the success of her approximations could be justified *a posteriori*, i.e., the resulting equations actually reproduced the results which had been known before for a special case (Cover 1965).

One could argue, that the simple perceptron is not a very interesting model to study, since its computational limitations have been well known (Minsky and Papert 1969) and practical ANNs consist of much more complicated architectures involving many perceptrons. However, as with any physical theory, even if a model is much simpler than the real world, it may deliver instructive insights into the principles governing reality. Furthermore, a solvable model may be modified and expanded by adding more realistic features as the tools available become more sophisticated, improving its efficacy by determining further possible relevant degrees of freedom and/or identifying short-comings of the considered model.

A host of papers followed Gardner's hugely influential study and we therefore restrict ourselves to mentioning a few representative ones (more or less relevant for the context of this thesis). For the capacity problem, the extension to the perceptron above its capacity limit has been performed in (Gardner and Derrida 1988; Krauth and Mézard 1989; Majer et al. 1993), whereas a review of retrieval properties can be found in (Kinzel and Opper 1991). In parallel, the generalization properties of the perceptron have been studied, initially for realizable rules, i.e., a student perceptron learning a teacher perceptron of the same type but unknown parameters, such that the student can realize the rule perfectly. Later these calculations were expanded to unrealizable rules due to noise or due to a mismatch between student and teacher perceptrons [for an overview see (Seung et al. 1992; Watkin et al. 1993) and references therein].

The extension to more realistic multilayer networks has, however, proved much more difficult. The advance came to a long halt after the initially promising papers by Barkai et al. (1990) for capacity and similarly by Schwarze and Hertz (1992) for generalization calculations, due to the inherent difficulties of the techniques employed. Only recently has significant progress been made, either due to exploiting a new tech-

nique (Monasson and O'Kane 1994) for capacity calculations (but still following the equilibrium statistical mechanics framework predominantly used) or by the study of on-line rather than batch learning (Saad and Solla 1995a) for generalization calculations, employing techniques from non-equilibrium statistical mechanics.

1.5 Motivation and outline

The motivation of the research conducted in this thesis falls into two categories. One is provided by the necessity to scrutinize simplifications typically made by dropping degrees of freedom deemed less relevant or even irrelevant and/or the selection of the relevant degrees of freedom. In the case of this thesis, this concerns the threshold or bias of the individual perceptron units to which little or no attention has been paid.

This has been caused partly by the fact that in Gardner's particular calculation the threshold could be absorbed by another order parameter. Since then only few calculations have included biases. In the case of feedforward networks, which are our sole interest, the most notable exceptions are [Wendemuth (1994b, 1995c, 1994a)], where non-trivial effects were reported already for simple perceptrons in both capacity and generalization problems such as improved generalization ability in the early stages of learning. We mention in passing that for recurrent networks there have also been some studies that included thresholds, e.g., (Engel et al. 1989; Engel et al. 1990; Rau and Sherrington 1990; Rau et al. 1991; Yau and Wallace 1991), where it has been shown that biases (or external fields as they are usually called in this area) can improve the retrieval properties of the studied network models. In multilayer network calculations, however, we are not aware of any previous studies that include biases.

The second motivation is provided by the need to study realistic models, i.e., models which are of the order of the complexity of models used in the real world — multilayer networks[6]. The goal in this case is to increase the envelope of knowledge and the usefulness of the statistical mechanics framework by addressing issues which are also of relevance to the practitioner such as understanding the behaviour of these models and their training algorithms in order to improve their generalization ability.

1.5.1 Outline

This thesis can be roughly grouped into four parts. In Chapter 2, we give a more extended introduction into simple neural network models, assumptions and concepts

[6]Instead of listing previous work on multilayer networks, we refer the reader to the relevant chapters, where our efforts will be set in context to previous work.

relevant for the understanding of the thesis, which allow a reader with little or no knowledge of neural networks to gain an idea of the issues involved. It will approximately mirror the structure of this chapter but in a more detailed fashion. A reader familiar with the area may skip this chapter in its entirety.

The second part, consisting of Chapters 3 and 4, studies capacity related problems. In Chapter 3, we revisit the issue of the perceptron above saturation by adding a threshold. Even in this simple model, the addition of a threshold leads to interesting new effects, most notably a phase transition in the solution space not previously present. That allows us to argue that the choice of order parameters in (Gardner 1988) is not suitable above saturation. In Chapter 4, the results of the preceding chapter are used in order to calculate the capacity of a class of networks built by network growing algorithms, usually termed *constructive* algorithms. This allows us to circumvent, to some extent, the technical difficulties of the multilayer capacity calculations for fixed architecture and to compare various constructive algorithms with each other and with fixed architecture models.

The third part, consisting of Chapters 5 and 6, studies generalization issues within the on-line learning paradigm. Whereas the capacity chapters employed techniques from equilibrium statistical mechanics, here the model dynamics are studied in an non-equilibrium framework. In particular, in Chapter 5 we re-examine the on-line learning dynamics of the *soft-committee machine*, a slightly simplified multilayer network architecture, for gradient descent learning, which has been and is being studied extensively (Saad and Solla 1995a). As for the perceptron above saturation, we find that the inclusion of biases can alter the generalization behaviour as a function of training time qualitatively, leading either to significantly shorter training times than anticipated previously or to the emergence of new (attractive) fixed points (which can trap training either for long times or indefinitely in suboptimal network parameter configurations exhibiting symmetries not present in the teacher task). In Chapter 6, a closer look is taken at the reasons for the slow training times for on-line gradient descent in soft-committee machines. We introduce an algorithm which consists of a slight modification to the gradient descent rule whose magnitude is controlled by a single parameter. This algorithm allows us to study some of the shortcomings of gradient descent and leads the way for further improvement.

We close with a summary of our main results, a discussion of their implications, and an outline of open questions which either have been raised in its course or have been cast aside. Finally, I would like to remark that as this thesis consists of a variety of works spanning the field, I felt it is appropriate to write the thesis in such a way that each chapter can be read more or less on its own but without excessive overlap with the material from other chapters.

Chapter 2

Neural Networks — A Primer

The aim of this primer is to introduce simple neural network models, and some of the issues and concepts insofar they are of relevance to later chapters. This primer does not attempt to be exhaustive or to provide an overview of the field of artificial neural networks (also termed neurocomputing, connectionism, parallel distributed processing, machine learning, or computational learning theory by different communities), and I refer the reader to such excellent textbooks and reviews as (Hertz et al. 1991; Watkin et al. 1993) for a physicists view on neural networks and (Bishop 1995; Ripley 1996) for more recent and statistical perspectives (from which some of the ideas of this primer have been taken). Furthermore, since the models investigated in later chapters require quite diverse statistical mechanics techniques these are introduced almost entirely in the relevant chapters, although readers not familiar with basic concepts of spin-glass and non-equilibrium statistical mechanics may benefit from consulting (Mézard et al. 1987) for general replica techniques and (Gardiner 1983) for stochastic processes [see (Seung et al. 1992) and (Mace and Coolen 1997) respectively for their application to neural networks].

The primer is divided as follows: Section 2.1, introduces neural network models; Section 2.2, deals with fundamental issues of learning and generalization; Section 2.3, deals with the basic theoretical foundations and alternative frameworks to the statistical mechanics approach; Section 2.4, briefly gives a flavour of statistical mechanics techniques applied to neural networks.

2.1 From neuron models to ANN

The basic neuronal building blocks used in many ANNs have been directly inspired from the simplest models of biological neurons and we will use this path to introduce

ANNs. Roughly speaking, a biological neural network is believed to operate as follows. In their active state a neuron sends signals down its *axon*, which branches off to and ending at *synapses*. *Dendrites* of other neurons receive signals from these synapses and their *post-synaptic potentials* are lowered or increased, depending upon whether the synapse is excitatory or inhibitory. Whether a neuron is active depends on the size of this potential. Above a certain *threshold* a previously quiescent neuron starts *firing* spontaneously, where the intensity and probability of firing usually depends on the strength of the potential but saturates at a maximal value.

2.1.1 A neuron model

A simple model for a neuron was proposed by (McCulloch and Pitts 1943), whose generalization is most commonly used in ANN. In this model, the post-synaptic potential h_i (usually termed *activation* or *internal field* in ANNs) of a neuron i (or *unit*) is just the sum of inputs Ξ_j $(j = 1, \ldots, N)$ weighted by their respective synaptic couplings W_{ij} (or *weights*) subtracted by a *threshold* θ_i (also termed *bias*)

$$h_i = \frac{1}{\sqrt{N}} \boldsymbol{W}_i \cdot \boldsymbol{\Xi} - \theta_i, \tag{2.1}$$

where the normalization with $\sqrt{N}$ is introduced for convenience. Note that in general an input Ξ_j can be either the output σ_j of some other neuron j or a proper external input ξ_j from a source j.

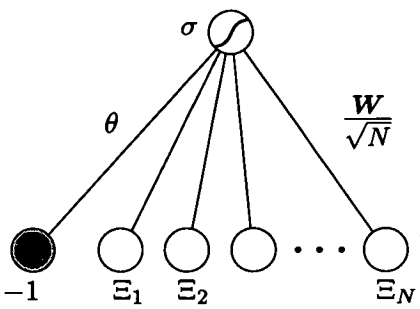

Figure 2.1. The simplest neuron model. As a network on its own it is usually referred to as the (simple) perceptron.

The output σ_i of the neuron in the generic model is assumed to be

$$\sigma_i = g(h_i), \tag{2.2}$$

where $g(x)$ is some (deterministic) transfer function. A network consisting only of one such unit is often termed the *(simple) perceptron* in the ANN community and is depicted in Figure 2.1.

In the original model of McCulloch and Pitts (1943), the neuron is a threshold unit and the transfer function becomes $g(x) = \Theta(x)$. In the context introduced above, it is referred to as the *Boolean perceptron*. In a statistical mechanics framework it is more convenient (Little 1974) to treat the neurons as Ising spins with symmetric states $\{-1, 1\}$ rather than asymmetric thresholded states $\{0, 1\}$. This is without loss of

generality (w.l.o.g.), since an asymmetric representation can be mapped to a symmetric one[1] by redefining $\sigma' = (2\sigma - 1)$. In this case, the transfer function just becomes $g(x) = \mathrm{sgn}(x)$.

In both artificial and biological neural networks, more general transfer functions are often considered which take real rather than binary values. The biological motivation for this can be taken either from the graded response of the neuron, dependent on its activation, or can be given a probabilistic interpretation as the neuron firing probability. In ANNs, the main reasons for using a real valued function are the utility of gradient-based techniques for continuous parameter estimation, the perceived increased computational abilities, and its applicability for both regression and classification tasks [where for (multi-class) classification problems one uses logistic discrimination, i.e., models class probabilities, instead of hard classification].

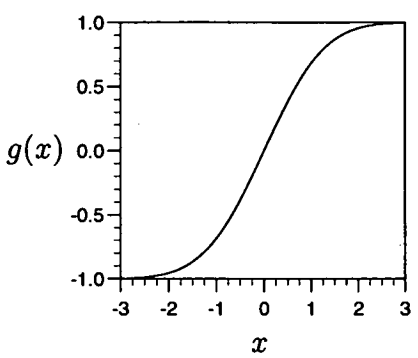

Figure 2.2. A typical sigmoidal transfer function in the range -1 to 1, here $g(x) = \mathrm{erf}(x/\sqrt{2})$ has been chosen.

In most cases these transfer functions are *sigmoidal* bounded squashing functions, usually operating in the intervals $]-1, 1[$ or $]0, 1[$, an example of which is shown in Figure 2.2. The exact mathematical form of such sigmoids seems not particular relevant[2], and popular choices in the ANN community for either range are $g(x) = \tanh(x)$ and the logistic sigmoid $g(x) = 1/(1 + e^{-x})$. For an analysis within a statistical mechanics framework it is more convenient to use the *error function* $g(x) = \mathrm{erf}(x/\sqrt{2})$ depicted in Figure 2.2. For the output neuron of a whole network in regression problem a linear transfer function $g(x) = x$ is usually used in order not to restrict the range of possible outputs. A network consisting of only one such neuron is termed the linear perceptron.

2.1.2 Network models

The simplest network model, the perceptron, consisting just of one neuron, has obviously serious computational limitations; it is essentially a *linear model*. The Boolean perceptron, $g(x) = \mathrm{sgn}(x)$ corresponds to a linear discriminant function in statistics, whose linear decision boundary, or hyperplane, can only implement a linearly separable

[1] This also leads to a transformation of the weights and thresholds (Hopfield 1982).

[2] Although there is both theoretical (Le Cun et al. 1991) and empirical evidence (Bishop 1995), that the range $]-1, 1[$ has some practical advantages over $]0, 1[$ in hidden layers of MLPs introduced below.

classification[3]. The linear perceptron, $g(x) = x$, performs a linear regression.

Considering the number of neurons in the brain and their connectivity, it seems clear that the brain's computational power lies in their numbers and interactions and that more powerful ANN models should reflect this. This poses three problems. These are essentially the question of architecture, i.e., the number and orientation of the connections in the network (or graphical structure), the question of size, i.e., the number of units in the network, and the question of learning ability, i.e., can a suitable training algorithm be developed for the architecture and size chosen.

In terms of architecture, ANN researchers have developed two architecture classes, recurrent and feedforward, which have been motivated by perceived brain structures. In *recurrent networks*, such as the Hopfield model (Hopfield 1982; Hopfield 1984) or Boltzmann machines (Hinton and Sejnowski 1986), the connectivity is high and the neurons drive one another collectively and repetitively, introducing feedback into the system[4]. They are essentially dynamical systems with many *attractors* and are therefore also a model for associative memory (Hopfield 1982). Although these models are computationally very powerful[5], they are notoriously difficult to train and have consequently decreased in popularity over recent years. Such models will not be considered in the remainder of this thesis.

Recurrent networks have subsequently been superseded by *feedforward networks* in popularity. In these the connectivity is more restricted and a signal is processed through the network from the input neurons, purely unidirectionally, towards the output neurons. No feedback is introduced and the state of each neuron is deterministic[6]. The most popular feedforward network is the multi-layer perceptron (MLP) shown in Figure 2.3, where the neurons are organized in layers, with connections only from one layer to the next, never backwards or sideways in a layer. Other, more complicated, feedforward networks architectures are also found, e.g., in networks built by constructive algorithms, as considered in Chapter 4.

The MLP is often restricted to only one *hidden* layer, as in Figure 2.3, an architecture termed two-layer network since the input layer is not counted. The realized

[3]A perceptron with a sigmoidal activation has in this case just the corresponding probabilistic interpretation of the class probability.

[4]In graph theory, neural networks can be described as directed graphs and recurrent networks can be defined as graphs with loops or cyclic graphs.

[5]This type of connectivity is also found in the "higher" region of the brain, i.e., the cortex, where cognitive functions are performed.

[6]In graph theory, the associated graph has no loops (acyclic graph).

mapping then becomes

$$\sigma = f\left\{\left[\frac{1}{\sqrt{K}}\sum_{i=1}^{K} w_i\, g\left(\frac{1}{\sqrt{N}}\boldsymbol{W}_i\cdot\boldsymbol{\xi} - \theta_i\right)\right] - \vartheta\right\}, \tag{2.3}$$

where w_i denotes the hidden-output weights from the K hidden units, ϑ the bias of the output unit[7], and f is a second generic transfer function, which is commonly a linear transfer function for regression (or function approximation). All the adjustable parameters of the model are therefore $\boldsymbol{\Omega} = \{\boldsymbol{W}_i, \theta_i, w_i, \vartheta\}$.

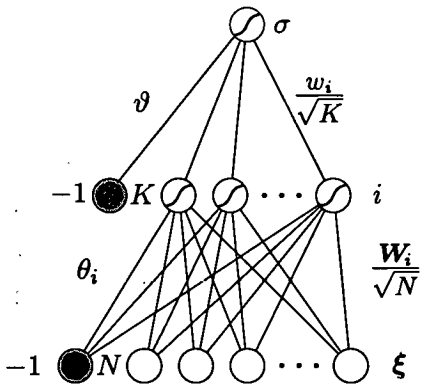

Figure 2.3. A multilayer perceptron with a single hidden layer.

The restriction to purely feedforward signal processing[8] and a single hidden layer may seem severe, however, it has been shown that this architecture is a *universal approximator* (Cybenko 1989), i.e., can in principle *approximate* a sufficiently "smooth" function to any required degree of accuracy, provided that the number of hidden units K is not restricted. Numerous papers have since been published generalizing this result [see e.g., (Hornik 1991)] and in particular giving rates of the approximation error with the number of hidden units in the network [see e.g., (Barron 1993)]. Furthermore, it has recently been shown (West et al. 1997), that an even simpler model, the soft-committee machine (Biehl and Schwarze 1995), where the hidden-output weights are fixed ($w_i \equiv 1$), is a universal approximator provided they have biases to the hidden layer. The number of hidden units required to achieve a similar approximation error, however, may be much larger than an equivalent general two-layer architecture.

These results concern the *approximation* ability (or error) of MLPs, i.e., how well we can expect the model to perform if we knew the optimal parameters. An entirely different problem concerns finding or *learning* these optimal parameters (or at least a good approximation) on the basis of a limited example set. Some of the issues and paradigms involved in the learning of network parameters are presented in the following section.

[7]Note that in the case of a linear output unit, the output bias ϑ can be removed w.l.o.g. since the outputs can be readjusted by their mean value.

[8]In the brain such regular layered and feedforward signal processing is mainly found in evolutionary older sections of the brain, e.g., cerebellum.

2.2 Learning and generalization

Besides the differences between the processing units of neural networks and conventional computers, the most striking difference is that humans and neural networks learn from examples. To introduce the different paradigms and issues involved in human and machine learning, we will use an example, closely related to a hypothetical (or real as in my case) move to Scotland and the subsequent introduction to whisky below.

2.2.1 An example — whisky classification

After having moved to Scotland, we feel it would be appropriate to learn something about whisky. We could take the approach that, as we do not know anything about whisky, we may as well ask the barman to pour us a measure out of a random bottle. As all whisky (to a first approximation of our current knowledge) tastes the same to us, we would also not bother to ask to see the label of the bottle in order to identify its brand. We may well discover eventually, that whiskies do have different aroma and that one group in particular reminds us of the smell of peat and seaweed (if we had an experience of these), but it would need the barman to tell us that the common factor is that they are made on islands off the West Coast of Scotland. The learning paradigm associated with the discovery of such groups or structure in data is called *unsupervised learning*.

A more common (and sensible if the labels on the bottles and a willing barman are readily available) approach to learning to become a whisky connoisseur (able to distinguish between the whisky regions of Scotland) would be to ask the barman to be our *teacher*. We still would ask him to serve us randomly a particular whisky out of the selection he has in his bar, our *training set*, but after tasting we would try to guess from which region of Scotland the whisky originates. A grumpy barman may only be willing to tell us whether our guess was right, a paradigm called *reinforcement learning*, whereas a friendly one will be telling us the actual region, a paradigm termed *supervised learning*, which will be the topic of this thesis.

Typically after gaining some experience, we will realize that we are especially bad at classifying, say, Highland whiskies and we may specifically ask the barman to pour one of those (which may require our barman to purchase a new bottle as we have already exhausted his selection of Highland whiskies!). This form of learning is usually referred to as *active* or *query* learning. In this thesis we will only consider learning from *random examples*.

After quite exhaustive tasting, we are actually able to classify (almost) all whisky brands in our regular bar into their respective regions. However, it is self-evident that

this ability should not make us content; only when we are actually able to predict the class label of a previously untasted whisky with some accuracy will we have generalized and be a true connoisseur.

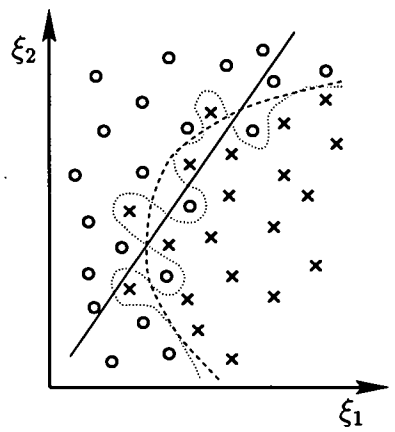

Figure 2.4. A schematic example of vectors in two dimensions (ξ_1, ξ_2) belonging to two classes represented by crosses and circles. The lines show the decision boundary of a model with low (——), medium (- - - -), and high (· · · · ·) complexity. Without some notion how well the classes are separated, i.e., what the teacher and the noise are, it is impossible to decide which should be the preferred model.

Failure in this respect can have several reasons. Initially there has to be an underlying *rule*, otherwise only memorization is possible and learning is futile. The size of our data set (the selection of whiskies in the bar) is necessarily limited and may also be intrinsicly *noisy*, e.g., single malt whiskies of the same distillery can vary in taste significantly between years (and barrels) or the barman could have been fraudulent and provided us with the wrong labels.

Both of these facts should lead us to consider what the *complexity* for our whisky classification model should be; a complex model with many free parameters may be able to classify all the training points correctly but could *overfit* and generalize poorly. On the other hand a too simple model with few free parameters may not overfit but miss essential parts of the rule and subsequently generalize poorly as well. These define the so-called bias and variance dilemma (Geman et al. 1992), and the trade-off is dependent on the size and noise of the data set and the underlying rule. Note that, in principle, one can take the extreme view that a student can never expect to perform better than random guesses without some *a priori* knowledge about the teacher (rule and noise) (Wolpert 1995).

However, experience tells us that the world is to a certain extent regular. We usually prefer simpler explanations (smooth models) over more complex ones if the results are comparable, a principle often referred to as *Occam's Razor*. With this notion in mind we would probably prefer the model with medium complexity in Figure 2.4 (in terms of the given example, we could think of these as three artificial nose models we have trained to distinguish between Island and Highland whiskies).

2.2.2 Supervised learning

Let us frame the above described method of supervised learning and considerations in a mathematical framework suitable to machine learning. A *student* performing a mapping $\sigma(\boldsymbol{\xi};\boldsymbol{\Omega})$ from an input space $\mathbb{I}$ to an output space $\mathbb{O}$, $\mathbb{I} \to \mathbb{O}$ (e.g., $\mathbb{R}^N \to \mathbb{R}$, in the future we will assume only one output node, although the results are easily generalizable to multiple outputs), is presented with a training set (or *data set*) $\mathcal{D}$ of p examples of input-output pairs $(\boldsymbol{\xi}^\mu, \zeta^\mu)$ $(\mu = 1, \ldots, p)$, where the inputs $\boldsymbol{\xi}^\mu$ are drawn independently and identically from some distribution $P(\boldsymbol{\xi})$. The output labels ζ^μ are generated by a *teacher* rule $\zeta_0(\boldsymbol{\xi})$ which has, in general, been corrupted by some noise process γ such that a label ζ^μ is generated by a probability $P(\zeta^\mu|\boldsymbol{\xi}, \zeta_0, \gamma)$ where we assume[9] $\langle \zeta^\mu(\boldsymbol{\xi}) \rangle_\gamma = \zeta_0(\boldsymbol{\xi})$ and $\left\langle [\zeta^\mu(\boldsymbol{\xi}) - \zeta_0(\boldsymbol{\xi})]^2 \right\rangle_\gamma = \sigma_\gamma^2$. Training consists of the adaption of the student parameters $\boldsymbol{\Omega}$ in such a way that the student's outputs reproduce the training labels ζ^μ of the training set at least approximately. The ultimate goal of learning is to achieve good generalization, i.e., infer the true teacher ζ_0, and we will return to this point shortly.

The problem machine learning faced initially, and still faces today for some architectures, was the lack of suitable training algorithms for networks more complicated than single-layer networks. The introduction of the concept of an *energy function*, or *training error*, (Hopfield 1982) proved particularly fruitful for devising training algorithms and is partly responsible for the resurgence of neural network interest in the 1980's. The training error $E(\boldsymbol{\Omega}|\mathcal{D})$ is defined to be the sum over all example pairs $(\boldsymbol{\xi}^\mu, \zeta^\mu)$ over a suitable *cost function* or *error measure* $\epsilon[\boldsymbol{\Omega}|(\boldsymbol{\xi}, \zeta)]$ measuring how accurate the answer of the student is on a particular example

$$E(\boldsymbol{\Omega}|\mathcal{D}) = \sum_{\mu=1}^{p} \epsilon[\boldsymbol{\Omega}|(\boldsymbol{\xi}^\mu, \zeta^\mu)] \tag{2.4}$$

For regression a popular choice for the cost function is quadratic loss[10]

$$\epsilon_{\mathrm{LMS}}[\boldsymbol{\Omega}|(\boldsymbol{\xi}, \zeta)] = \frac{1}{2}\left[\zeta - \sigma(\boldsymbol{\xi};\boldsymbol{\Omega})\right]^2 . \tag{2.5}$$

[9]For some noise models the noise is not unbiased, i.e., $\langle \zeta^\mu(\boldsymbol{\xi}) \rangle_\gamma \neq \zeta_0(\boldsymbol{\xi})$, and the above description is not general. However, it is obvious that a student could never distinguish between the true teacher and the noise-averaged biased teacher. In fact, if the aim is to predict the teacher output, the prediction error of the student is minimized by learning the biased teacher. Therefore one could argue that the true teacher should be defined by $\langle \zeta^\mu(\boldsymbol{\xi}) \rangle_\gamma$. Since, we neglect the effect of noise in the rest of the thesis, these fine differences are immaterial.

[10]This cost function can be motivated by maximum likelihood consideration for an additive Gaussian noise model on the teacher outputs (Bishop 1995).

For classification problems, this cost function can still be used for sigmoidal units (or by training on the activation h of a thresholded output unit), although better choices exist, as discussed in (Bishop 1995). For simple binary perceptrons, a family of cost functions parameterized by an exponent n is often considered (Griniasty and Gutfreund 1991)

$$\epsilon_n[\Omega|(\xi,\zeta)] = \Theta(\kappa - \zeta h)(\kappa - \zeta h)^n, \tag{2.6}$$

where κ is the stability with which the patterns are required to be stored.

Training then consists of minimizing the training error by a suitable technique, the simplest of which is gradient descent which, for finite step size $t \to t+1$, becomes

$$\Omega_i^{t+1} = \Omega_i^t - \eta \frac{\partial E(\Omega|\mathcal{D})}{\partial \Omega_i}, \tag{2.7}$$

with a suitable learning rate η controlling the size of an update step. In the case of MLPs and quadratic loss (2.5), Eq. (2.7)is commonly identified with the *back-propagation* algorithm (Werbos 1974; Rumelhart et al. 1986a), which is studied in Chapter 5 for a simplified MLP architecture in an idealized scenario. Similarly, by differentiating[11] Eq. (2.6), we can recover the perceptron learning rule (Rosenblatt 1962) with Eq. (2.6) and $n = 1$, or AdaTron learning (Anlauf and Biehl 1989) with $n = 2$ [besides trivial differences such as the enforcement of weight vector normalization (Griniasty and Gutfreund 1991)]. In practice such deterministic training is usually performed by computationally much more efficient second-order gradient-based algorithms such as scaled conjugate gradient (Bishop 1995). In the case of MLPs, such deterministic algorithms are prone to falling in poor local minima, i.e., minima with unacceptable high training error.

One way of avoiding local minima is to use stochastic methods, which correspond to additive thermal noise in Eq. (2.7). For *exhaustive learning* ($t \to \infty$), the resulting Gibbs distribution takes the form

$$P(\Omega|\mathcal{D}) = \frac{1}{Z} P(\Omega) \exp\left[-\beta E(\Omega|\mathcal{D})\right], \tag{2.8}$$

where β is the inverse temperature of the thermal noise[12]. The probability distribution

[11] Error functions such as the Gardner–Derrida cost function [$n = 0$ in Eq. (2.6)] are not differentiable and only Bayesian techniques can be applied (see below).

[12] For finite step size, one assumes that the updates in t are Poisson distributed (Heskes 1994) and the inverse temperature β of the thermal noise is set by the variance of the added white noise process F_i^t with $\left\langle F_i^t F_j^{t'} \right\rangle = 2\eta/\beta \delta_{ij} \delta_{tt'}$.

$P(\Omega)$ represents prior constraints on the student and Z is the normalization constant or *partition function*

$$Z = \int d\Omega\, P(\Omega) \exp(-\beta E). \tag{2.9}$$

This framework, closely resembles (Tishby et al. 1989; Levin et al. 1990) the Bayesian statistics viewpoint (Neal 1996), where the Gibbs distribution corresponds to the *posterior distribution*, β is interpreted as the inverse variance of the assumed additive Gaussian noise distribution on the outputs, and the partition function is termed the *evidence* of a model. As for many problems in statistical physics, it is very difficult to calculate the posterior distribution in a Bayesian framework analytically and almost always Monte Carlo (Neal 1992) techniques have to be employed to approximate the posterior distribution by a representative sample.

The final ($t \to \infty$) solutions for this mode of training, usually called *batch learning*, can therefore be analysed within equilibrium statistical mechanics. An alternative paradigm for training is called *on-line learning*, where single examples are presented serially and the training algorithm adjusts the parameters after the presentation of each example. On-line learning is usually applied in cases where the whole training set is either very large or training data is continually produced by a teacher, making batch training methods infeasible due to the associated memory requirements. On-line learning is also used in application domains where rules vary with time, such as encountered in many time-series prediction problems. Furthermore, on-line learning is less susceptible than batch learning to being trapped in local minima in the error surface due to the stochasticity in the training process induced by the single example stream. On-line learning is therefore a dynamical stochastic process which has to be analysed within a non-equilibrium statistical mechanics framework. We will study batch learning in Chapters 3 and 4, whereas on-line learning is considered in Chapters 5 and 6.

2.2.3 Generalization issues

Having trained our model successfully, let us now consider the main issues which may influence its ability to generalize. For simplicity we will consider only regression problems and the quadratic loss function, although similar considerations also hold for classification problems and other error functions. The standard approach to measuring the generalization ability (in a frequentist approach) would be to split the total data available in a training set and a *test set* $\mathcal{D}_t$ of size p_t. After training our model, we measure its performance by the *empirical test error*, defined as the average error over

the test set

$$\tilde{\epsilon}_t(\Omega|\mathcal{D}_t) = \frac{1}{2p_t}\sum_{\nu=1}^{p_t}[\zeta^\nu - \sigma(\xi^\nu;\Omega)]^2 . \tag{2.10}$$

In the limit $p_t \to \infty$, we recover the *test error* ϵ_t, also called total risk, defined as the average of the error measure ϵ over the input and output (i.e., noise process) distributions

$$\epsilon_t(\Omega) = \frac{1}{2}\left\langle[\zeta - \sigma(\xi;\Omega)]^2\right\rangle_{\xi,\gamma}. \tag{2.11}$$

It is useful to decompose the test error into two parts using the true teacher ζ_0 (see Section 2.2.2) and the assumption that the corruption process is additive white noise to yield

$$\epsilon_t(\Omega) = \frac{1}{2}\left\langle[\zeta_0(\xi) - \sigma(\xi;\Omega)]^2\right\rangle_{\xi} + \sigma_\gamma^2. \tag{2.12}$$

The first term in Eq. (2.12), the average of the error with respect to the true teacher, defines the *generalization function* $\epsilon_f(\Omega)$. The approximation error is then defined as the minimum of the generalization function with respect to the student parameters

$$\epsilon_a = \min_{\Omega}\epsilon_f(\Omega). \tag{2.13}$$

The approximation error is a decreasing function of the model complexity and reaches zero if the model can *realize* the teacher mapping, i.e., if the teacher space is a subspace of the student space[13]. If this is the case we speak of *realizable rules* otherwise of *unrealizable rules*. Note that the term "unrealizable rules" is also often applied in the case where the data are corrupted by noise. To distinguish between the two kinds of unrealizabilities, we speak of structural or noise-induced unrealizability.

The fact that the test error can never be smaller than the variance of the noise process on the data has an important implication for practical training: we should never attempt to train to a training error which is smaller than $p\sigma_\gamma^2$ (which obviously needs some kind of belief, knowledge or estimate of the noise level). To achieve a final training error of $E \approx p\sigma_\gamma^2$ we need to find a model that avoids both underfitting ($E \gg p\sigma_\gamma^2$) and overfitting ($E \approx 0$). For (generalized) linear models, such as polynomials, the number of parameters in the model must therefore be somewhat smaller than the number of

[13]If the teacher space is a proper subspace of the student space one also speaks of an *overrealizable* rule.

examples. For neural networks we can also use the number of hidden units or weights as an approximate measure of model complexity, although the number of layers, the connectivity and the size of the weights have an influence as well, which results in a reduced effective number of parameters.

Assuming that we have a good idea of the noise level and are willing to set aside a large test set for measuring our performance, we still have to search for a model with an appropriately small test error. Naturally, we could train a range of models with different architectures (different number of layers, weights, or connectivity) to find the right complexity, but a more principled approach seems advisable and historically several methods have evolved.

Constructive, or growing algorithms, are based on the idea of starting with a simple model, adding more units only when deemed appropriate as determined either by the training error or some heuristic rule[14]. *Pruning* algorithms follow the contrary idea of starting with a large network and shrinking its size by pruning weights (or whole units) deemed unnecessary during training, based on some criterion. Both of these methods can be termed structural stabilization. A conceptionally different, but effectively similar approach, is *regularization*, which adds a *penalty term* C to the training error penalizing weights which do not significantly contribute to the reduction in training error. The total energy function (or Hamiltonian) to be minimized then becomes

$$H = E + \lambda C, \tag{2.14}$$

where λ is a multiplicative constant controlling the degree of regularization. The most common penalty term is a quadratic penalty term for all "ordinary" weights (i.e., excluding thresholds) which, for a two-layer MLP, is

$$C = \frac{1}{2}\sum_i \lambda_i \boldsymbol{W}_i \cdot \boldsymbol{W}_i + \frac{1}{2}\lambda_0 \boldsymbol{w} \cdot \boldsymbol{w}, \tag{2.15}$$

where we potentially have allowed for different constants λ_i for the different units and layers. This penalty term is often called weight-decay as it leads to a linear decay of all weights in gradient descent and can also be identified with a Gaussian prior on the student weights in a Bayesian framework.

As one can see, these algorithms often have some constants, or "fiddle factors" such as λ in Eq. (2.14), which have to be set by the user. This can either be done by some insight or *prior belief* (when viewed within a Bayesian approach) on the smoothness of

[14] As the memory capacity of constructive algorithms will be the focus of Chapter 4, we will later provide a more detailed treatment of these algorithms.

the expected function and/or the noise level, or by statistical techniques such as *cross-validation*, where parts of the training set are used as a *validation set* on which these constants are optimized[15]. In this latter *frequentist* framework, the initial amount of data is therefore usually split in two sets to determine the best model.

Using a fraction of the training data as a validation set may be unsatisfactory, especially if data are scarce. An alternative viewpoint is given by Bayesian statistics where no validation set is needed. In a Bayesian framework, prior beliefs about the expected kind of function are incorporated in the student prior, as indicated in Eq. (2.8), which can be interpreted as a probabilistic version of the regularization term in Eq. (2.14). Furthermore, beliefs about the noise level and the extent of smoothness are usually expressed in *hyper-priors* for β and $\tilde{\lambda} = \lambda/\beta$. In principle, the size of the network has not to be restricted *a priori*. However, since the posterior distribution cannot be calculated analytically, the computational cost of Monte Carlo techniques (although much more efficient than any other integration techniques) practically limits the network size. This highlights a further drawback of Bayesian techniques; there is in principle no guarantee that we have sampled from the true equilibrium distribution, especially if this has a complicated structure.

After employing one of the methods above, we may have achieved an excellent solution (expressed by a very low error on a large test set) in comparison to other algorithms or models we have tried. This may be satisfactory for a particular application, however, the scientific value of such a result is (close to) zero. First of all the result may not be reproducible, as we have used particular "fiddle factor" values for our algorithm (even cross-validation techniques and Monte Carlo algorithms have these, although here they play a less crucial rôle) or we may have been simply lucky to have found a good solution. Furthermore, the validity of our conclusion holds only for that particular data set and the particular rule we have tried to learn. To extend our knowledge about the capabilities of an algorithm or model, we need a notion of how they are going to work for other data sets for the same problem or other rules. The need for theoretical frameworks which provide answers to questions, such as the number of training examples typically needed for a particular model to achieve a required performance on a particular problem, is therefore self-evident. Frameworks (statistical mechanics being one them) which can provide some answers to these types of questions are discussed in the following section.

[15] If we were to use the test set for this purpose, the error measured on the test set would cease to be an unbiased estimator of the test error.

2.3 Theoretical machine learning frameworks

The aim of any theoretical framework is to give either a bound on, or an expected value of, the (test error) performance we can expect when training a model. For a learning algorithm applied to a given class of teachers, the performance as a function of the number of examples provided is called the *learning curve*. The calculation of learning curves requires us to make the space of possible student and teachers explicit. In a theoretical framework it is convenient to define the teacher also as a neural network with parameters $\boldsymbol{\Omega}_0$ from some defined set. This assumption is not too restrictive if a very general network class is assumed, e.g., the class of functions implemented by all MLPs is very general as they are universal approximators. Furthermore, as explained in Section 2.2.1, without a notion of the rule to be learnt we cannot expect to be able to generalize. If upper performance bounds are desired, i.e., if we want to safeguard us against the *worst case*, less restrictive assumptions must be made than if we are interested in the expected performance. These two cases, their respective frameworks, merits and drawbacks are discussed below.

2.3.1 Worst case and the PAC framework

As mentioned above, we may in principle be interested in two kind of learning curves depending on our view of the world. A company that would like to use a neural network that controls some safety-critical operation, like the position of fuel rods in a nuclear fission reactors, would probably like to bound the error of the network with a very high confidence (the worst possible case). On the other hand, a company that would like to use a neural network for predicting costumer demand for their call centre would probably be content with the average error the network will make.

Worst-case results are usually studied with the probably approximately correct (PAC) framework (Valiant 1984) [for a good introduction and overview see (Kearns and Vazirani 1994)] of computational learning theory. The PAC framework gives bounds on the number of examples p needed by a learning algorithm to produce a student which, with high probability $(1 - \delta_c)$, has test error smaller than a specified accuracy ϵ_c. This requirement is to hold for any teacher drawn from the class of considered functions for any arbitrary but fixed input distribution. The main result of PAC learning is that p is dependent only on the Vapnik–Chervonenkis (VC) dimension (Vapnik and Chervonenkis 1971; Vapnik 1982), a measurement of the complexity of the function space, and the confidence and accuracy parameter δ_c, ϵ_c. The VC dimension d_{VC} of a function class is defined as the largest size of an input set which can be mapped to any desired output set. Intuitively, d_{VC} is the point where the "memory" of the

function class is saturated and beyond this, generalization can begin (Opper 1994). The VC dimension has some similarity to the capacity limit, an alternative definition of "memory", in the statistical mechanics framework (which will be the focus of Chapters 3 and 4). Both VC dimension and capacity are, however, very difficult to calculate for more complex function classes (in the case of ANNs more complex than the binary perceptron) and often only bounds can be found.

The advantages of the PAC framework are that it is distribution free and gives bounds explicitly dependent on the confidence and accuracy parameters. However, the approach also has severe disadvantages. First, it assumes that the problem is realizable, i.e., teacher and student functions spaces are identical[16]. Realistically, we often encounter problems where this may not be the case, e.g., we may not have very much data and we need to use a simpler student than teacher model in order to make the problem well posed. Second, the PAC framework was not designed to deal with noisy rules, although this possibility has been incorporated recently to some extent. Third, the PAC framework is mainly suited to classification and not regression tasks. The most serious drawback, however, is practical; the PAC framework gives bounds in the required number of examples which are very conservative in comparison to the results found in real applications. This may have several explanations, for example, only upper bounds are known on the VC dimension. However, the most reasonable explanation is that the worst case is just so atypical that it is extremely unlikely to be encountered in practice.

2.3.2 Average case analysis and generalization error

An average case framework is appropriate in many cases, especially if we are interested in typical results of our learning curve. In this case, we have to make some more restrictive assumptions on the data generating distributions, the selection process and distribution for the training inputs, and the noise process corrupting the teacher output. The extra restrictions in the average case have both general advantages and disadvantages. The need for more explicit assumptions makes the results seem less general[17] than the PAC results. Such assumptions, on the other hand, often make the resulting learning curves much more realistic. Furthermore, noise and a mismatch between teacher and student spaces can be incorporated more easily than in the PAC framework and both regression and classification problems can be investigated.

[16] In the VC theory, however, one can bound the difference between training and test error for unrealizable rules (Vapnik 1982).

[17] Although one could in principle use a family of parameterized distributions and average over its parameters.

The main aim of all average case analyses is the calculation of the average learning curve, measured by the expected performance of the student as a function of the number of examples, termed the *generalization error*. This is defined by averaging the generalization function introduced in Section 2.2.3 over the instances of the data set $\mathcal{D}$ and the (possible) randomness of the training procedure, i.e., over the distribution of possible training inputs and outputs (noise) $(\boldsymbol{\xi}^\mu, \zeta^\mu)$ and over the posterior student distribution (2.8) given a particular instance of the data set.

In statistical mechanics terms the average over the instance of the training data (input and noise distribution) corresponds to a quenched average, $\langle\!\langle\cdot\rangle\!\rangle_{\mathcal{D}}$, over the random disorder associated with the choice of a particular data set $\mathcal{D}$ $[(\boldsymbol{\xi}^\mu, \zeta^\mu), (\mu = 1, \ldots, p)]$. The average over the posterior student distribution corresponds to a thermal average, $\langle\cdot\rangle_T$, over the equilibrium Gibbs distribution, i.e., the ensemble of student parameters $\boldsymbol{\Omega}$, for a particular instance of the data set $\mathcal{D}$. Similar to the test error, one is interested in the average of the student-teacher mismatch for a random test example $\boldsymbol{\xi}$, and hence includes a further average, $\langle\cdot\rangle_{\boldsymbol{\xi}}$, over the distribution of test examples. The *generalization error* is then formally defined as

$$\epsilon_g = \frac{1}{2}\left\langle\!\!\left\langle\!\!\left\langle \left[\zeta_0(\boldsymbol{\xi}) - \sigma(\boldsymbol{\xi};\boldsymbol{\Omega})\right]^2 \right\rangle_T \right\rangle\!\!\right\rangle_{\mathcal{D}}\right\rangle_{\boldsymbol{\xi}}, \tag{2.16}$$

Note that one can also average the test error in a similar manner, which defines the *prediction error*. The difference between these two definitions results in only a constant additive noise error term. One can decompose the generalization error into two parts by inserting $\langle\sigma(\boldsymbol{\xi};\boldsymbol{\Omega})\rangle_T$ in (2.16) yielding

$$\epsilon_g = \frac{1}{2}\left\langle\!\!\left\langle\!\!\left[\zeta_0(\boldsymbol{\xi}) - \langle\sigma(\boldsymbol{\xi};\boldsymbol{\Omega})\rangle_T\right]^2 \right\rangle\!\!\right\rangle_{\mathcal{D}}\right\rangle_{\boldsymbol{\xi}} + \frac{1}{2}\left\langle\!\!\left\langle\!\!\left\langle \left[\langle\sigma(\boldsymbol{\xi};\boldsymbol{\Omega})\rangle_T - \sigma(\boldsymbol{\xi};\boldsymbol{\Omega})\right]^2 \right\rangle_T \right\rangle\!\!\right\rangle_{\mathcal{D}}\right\rangle_{\boldsymbol{\xi}}. \tag{2.17}$$

The second part measures the variance of the student solution induced by the algorithm (averaged over all data sets), which vanishes in a Bayesian framework, where the prediction is made by the mean student. This is a simple example of a bias and variance decomposition of a squared error in statistics[18].

Although the generalization error, as defined above, is now independent on the instance of the data set and the noise process corrupting the teacher rule, it is still dependent on a particular fixed teacher and it is desirable to remove this stochasticity by (quenched) averaging over a distribution of teachers. This is usually not part of the

[18]This should not be confused with the bias and variance dilemma (Geman et al. 1992), where the bias and variance terms are split with respect to the quenched disorder of the training data.

definition of the generalization error, however.

The calculation of the generalization error has proved to be very difficult, even for simple perceptrons and idealized Gaussian input and additive noise distributions. In the traditional statistics framework, only results in the asymptotic limit of a large number of examples p for realizable scenarios and smooth network mappings have been calculated (Amari et al. 1992; Amari and Murata 1993). This limit is not particularly interesting for real world problems where the data are limited. The intractability of calculating the generalization error is due to the complicated structure of the posterior distribution (2.8) for small example sets. Only in the limit $p \to \infty$ for realizable scenarios and smooth network mappings[19] is the asymptotic theory of statistics applicable where the posterior of the student parameters is approximately normally distributed around the true values of the teacher (corresponding to an annealed approximation in statistical mechanics).

Another approach, arguably more fruitful, has been to use statistical mechanics, since techniques developed to calculate the partition function of large disordered interacting particle systems, spin glasses, can be brought to bear in the non-asymptotic data size region. The main drawback from the view point of practitioners is the use of the infinite input dimension limit, $N \to \infty$. Since neural networks are viewed in this thesis from a statistical mechanics perspective, we give a basic flavour of the techniques employed in the following section.

2.4 Statistical mechanics framework

The purpose of this section is not to provide a detailed introduction of the techniques of later chapters but to add some background understanding as to what techniques can be applied and how they are related to each other. Let us therefore briefly recapitulate the facts established in the course of this primer and the introduction.

In *supervised* learning, a training set, consisting of p input-output pairs $\{(\boldsymbol{\xi}^\mu, \zeta^\mu)\}$, is given to the student network with parameters $\boldsymbol{\Omega}$. We distinguish two task types, as introduced in Section 1.2, the memorization and generalization tasks. In the memorization task, the outputs are labelled randomly (by a teacher), and the aim is to determine the expected memory capacity of a network. In the generalization task, the outputs are labelled by a teacher network, with parameter $\boldsymbol{\Omega}_0$, (possibly corrupted by noise) and the interest is in the average generalization error the student will typically achieve after being trained on p examples. This assumes implicitly that we are

[19]For example, for Boolean output units the posterior of the student does not become normally distributed in this limit.

not interested in the training dynamics, but in the final outcome for infinite training time, and an equilibrium statistical mechanics framework is appropriate, otherwise a non-equilibrium framework must be employed.

As mentioned in Section 1.4, the seminal paper by Gardner (1988) employed techniques developed for the study of spin glasses, since both spin glasses and neural networks can be seen as strongly interacting systems of particles with a quenched disorder. To make this analogy more clear, let us compare an Ising spin glass with a linear perceptron (see Section 2.1.1) that adjusts its parameters $\boldsymbol{\Omega} = \{\boldsymbol{W}, \theta\}$, learning either random examples or examples labelled by a noiseless teacher of the same architecture $\boldsymbol{\Omega}_0 = \{\boldsymbol{B}, \varrho\}$.

2.4.1 The spin glass analogy

In spin glasses, each of the N spins S_i interacts with a number K of its neighbours, dynamically exploring phase space on a short time scale. The interaction between the spins is determined by their couplings J_{ij}, which are assumed to be random, leading to frustration and disorder in the system. The change of these random couplings takes place over a much longer time scale than the spins need to relax to their thermal equilibrium; this is termed *quenched* disorder. The Hamiltonian of such a system is

$$H = -\sum_{i=1}^{N} \sum_{j \sim i} J_{ij} S_i S_j - \sum_{i=1}^{N} H_i S_i - H_0, \tag{2.18}$$

where H_i are external fields, H_0 some constant energy offset and $j \sim i$ denotes all indices j of spins interacting with spin i.

For neural networks, the approach is exactly the opposite. Here, the examples (or spins) are fixed, whereas the weights are the dynamic variables which are stochastically trained (strongly interacting via the p examples), relaxing to the posterior distribution, equivalent to the equilibrium Gibbs distribution. The change of the data set (spins) takes place on a time scale much slower than the training process of the weights, again inducing a quenched disorder. To make the analogy more clear, let us write the Hamiltonian (2.14) for the above mentioned linear perceptron student (2.1). We further assume a Gaussian weight prior (2.15) and training with quadratic loss (2.5) on the p examples. The inputs are (for simplicity) independently identically distributed samples from $\{-1, 1\}^N$. For a general teacher output ζ^μ the Hamiltonian equates to

$$H = \frac{1}{2} \sum_{\mu=1}^{p} \left[\zeta^\mu - \left(\frac{1}{\sqrt{N}} \boldsymbol{W} \cdot \boldsymbol{\xi}^\mu - \theta \right) \right]^2 + \frac{\lambda}{2} \boldsymbol{W} \cdot \boldsymbol{W} \tag{2.19a}$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{N}\left[\frac{\lambda}{2}\mathbf{1} + \frac{1}{2N}\sum_{\mu=1}^{p}\boldsymbol{\xi}^{\mu}(\boldsymbol{\xi}^{\mu})^{\mathrm{T}}\right]_{ij} W_i W_j$$
$$- \sum_{i=1}^{N}\left[\frac{1}{\sqrt{N}}\sum_{\mu=1}^{p}(\zeta^{\mu}+\theta)\,\xi_i^{\mu}\right] W_i + \frac{1}{2}\sum_{\mu=1}^{p}(\zeta^{\mu}+\theta)^2\,, \qquad (2.19b)$$

where we have rearranged the terms in Eq. (2.19b) such that the correspondence to Eq. (2.18) becomes more obvious. It seems that the only rôle of the student bias is to shift the training labels ζ^{μ}; indeed it can be shown that for the linear perceptron, the student (and teacher) bias[20] can be neglected w.l.o.g. by considering the transformation $(\zeta^{\mu})' = \zeta^{\mu} + \theta$. We will see in Chapter 3, however, that this not the case in a similar simple learning scenario. The Hamiltonian in the case of a teacher perceptron then simplifies to

$$H = \sum_{i=1}^{N}\sum_{j=1}^{N}\left[\frac{\lambda}{2}\mathbf{1} + \frac{1}{2N}\sum_{\mu=1}^{p}\boldsymbol{\xi}^{\mu}(\boldsymbol{\xi}^{\mu})^{\mathrm{T}}\right]_{ij} W_i W_j - \sum_{i=1}^{N}\sum_{n=1}^{N}\left[\frac{1}{N}\sum_{\mu=1}^{p}\boldsymbol{\xi}^{\mu}(\boldsymbol{\xi}^{\mu})^{\mathrm{T}}\right]_{ni} B_n W_i$$
$$+ \sum_{n=1}^{N}\sum_{m=1}^{N}\left[\frac{1}{2N}\sum_{\mu=1}^{p}\boldsymbol{\xi}^{\mu}(\boldsymbol{\xi}^{\mu})^{\mathrm{T}}\right]_{nm} B_n B_m. \qquad (2.19c)$$

2.4.2 Mean-field order parameter

A further central result from Eq. (2.19) is that each student weight interacts with all other student weights, i.e., neural networks model have infinite interaction range similar to the SK spin-glass model (Sherrington and Kirkpatrick 1975), and the physical system can be described exactly by a set of macroscopic *order parameters* of the mean-field type[21] in the thermodynamic limit of infinite input dimension $N \to \infty$. These order parameters appear naturally in the course of the averaging of the random disorder and typically measure the overlaps either between different student weight vector solutions (similar to the Edwards-Anderson order parameter and its generalizations in spin

[20] The marginal distribution of the student bias has a Gaussian distribution with a mean centred at the average output and a variance of $\mathcal{O}(1/N)$ induced by the randomness of the (unbiased) inputs, i.e., the bias is self-averaging in the thermodynamic limit. Additionally, the analysis in (Sollich and Barber 1997d) shows that the bias can be learnt from $\mathcal{O}(1)$ training examples.

[21] The equivalence obviously holds only for Ising weights, however, one can verify that the condition $W_i = \mathcal{O}(1)$ is sufficient [see (Mace and Coolen 1997)].

glasses) or between student and teacher weights (in the case of a teacher perceptron)

$$Q_{\sigma\rho} = \frac{1}{N}\boldsymbol{W}_\sigma \cdot \boldsymbol{W}_\rho \qquad \text{and} \qquad R = \frac{1}{N}\boldsymbol{W}\cdot\boldsymbol{B} \tag{2.20}$$

which will be introduced in detail later. These order parameters allow us to reduce the number of free weight parameters from $\mathcal{O}(N)$ to typically $\mathcal{O}(1)$ and are (usually)[22] *self-averaging* for $N \to \infty$, i.e., their probability distributions with respect to the randomness of the examples become Gaussian with variance of $\mathcal{O}(1/N)$.

2.4.3 The free energy

Having ascertained that the equivalence between some neural network models and spin glasses holds, we can use the same techniques employed for spin glasses to calculate the observables we are interested in. Many relevant quantities, such as the average training error per example or the generalization error, can be calculated from the free energy (per degree of freedom) defined as

$$f = -\frac{1}{\beta N}\langle\langle \log Z \rangle\rangle_{\mathcal{D}}\,, \tag{2.21}$$

which is assumed to be self-averaging in the thermodynamic limit $N \to \infty$ since the unnormalized free energy is an extensive quantity[23]. The observables of interest can then be calculated as derivatives of the free energy with respect to suitable Lagrange multipliers, e.g., the average training error becomes $\partial(\beta f)/\partial\beta = p/N\epsilon_{\text{train}}$. Unfortunately, the calculation of the free energy is a difficult problem as it involves averaging over the logarithm of the partition function.

A simple but still fairly interesting approximation for calculating the free energy is the high-temperature limit, where both the temperature $T = 1/\beta$ and the example number divided by the input dimension p/N go to infinity simultaneously with their ratio fixed. In this limit the quenched disorder induced by the finite training set disappears simplifying the calculation of the free energy considerably. Although this limit is not completely trivial, it still exhibits unrealistic features such as the already mentioned infinite example set and the equivalence of training and generalization error (Seung et al. 1992).

[22] This is strictly true only for ergodic systems. If ergodicity is broken, as experienced in many occasions such as the problem studied in Chapter 3, there is some argument about the correct interpretation. For a detailed discussion see (Binder and Young 1986; Mézard et al. 1987).

[23] A simple argument (strictly holding only for finite range spin-glass) is given in (Binder and Young 1986).

A more advanced but still relatively simple techniques for calculating the free energy is the annealed approximation $\beta N f \approx -\log \langle\langle Z \rangle\rangle_{\mathcal{D}}$, where the quenched average is taken inside the logarithm. Since the logarithm is a convex function this gives a lower bound on the free energy, which is typically adequate for higher temperatures and realizable rules. However, for neural network learning we are usually interested in low temperature learning where the annealed approximation can be inadequate giving even qualitatively incorrect results. Additionally, the annealed approximation also fails for unrealizable (noisy) rules (Seung et al. 1992).

A more complicated but also more accurate technique (valid for all temperatures) for calculating this average is the *replica method*. Here a mathematical identity for the logarithm, the *replica trick*, (Edwards and Anderson 1975) is used to circumvent the direct average, but introduces replicas of the physical system (i.e., the network) representing different solutions to the same problem. However, further simplifications about the symmetries of the solution space have typically to be made in order to solve the resulting equations, and can only be justified *a posteriori*. This technique will be used in Chapters 3 and 4. However, other techniques for equilibrium calculation exists, which we will briefly review.

2.4.4 Alternative equilibrium approaches

Several alternative methods exist to calculate the observables of interest which do not use the free energy as the fundamental quantity.

Gardner Volume: In the original paper (Gardner 1988) the logarithmic volume in phase space whose weight correctly implement all examples was calculated. This approach can be interpreted as the zero-temperature entropy of networks with zero error (see below). It is mainly used in capacity calculations (Barkai et al. 1990; Barkai et al. 1992; Engel et al. 1992) but can also be adopted to generalization calculations (Opper 1994). Replica techniques are used to evaluate $\langle\langle \log V \rangle\rangle$.

Microcanoncial ensemble: The microcanonical ensemble can be used instead of the macrocanonical for calculating the numbers of networks with a certain energy, i.e., evaluating the entropy. This approach proved useful for networks with Ising weights, since the zero-entropy line can be used to calculate the minimal training error (Fontanari and Meir 1993). Again, replica techniques are needed to evaluate the entropy.

Multifractal analysis: This recently developed technique (Monasson and O'Kane 1994) calculates the distribution of relative weight cell sizes with multifractal

techniques. A weight cell is defined by the volume of all weights that lead to the same output representation for a given input pattern set. From this distribution both capacity and generalization[24] results can be inferred in principle (Monasson and Zecchina 1995; Monasson and Zecchina 1996; Cocco et al. 1996; Urbanczik 1997; Engel and Weigt 1996; Weigt and Engel 1997; Malzahn et al. 1997). The calculation of the multifractal spectrum requires replica techniques.

Cavity methods: This technique also originates from spin glass theory (Mézard et al. 1987) as a generalization from the TAP mean-field equations[25] (Thouless, Anderson, and Palmer 1977) and represents an alternative to replica calculations. The main idea of this approach is to introduce just one new example to the already trained network, and to study the average reaction of the network to this perturbation. (Mézard 1989; Griniasty 1993; Bouten et al. 1995; Opper and Winther 1996; Gerl and Krey 1994; Gerl and Krey 1995; Gerl and Krey 1997; Wong 1995; Wong 1997).

Response Function: For linear perceptron (student and teacher) the averaged *response function*, consisting of the averaged trace of the inverse example correlation matrix, has been applied successfully to calculate the observables (Krogh and Hertz 1992; Sollich 1994). In fact, the response function also provides the training dynamics as well. However, this method is essentially limited to a linear perceptron student since it seems otherwise intractable.

These methods can all be applied to calculate equilibrium observables, however, if one is interested in the training dynamics of non-linear networks, techniques from non-equilibrium statistical mechanics have to be applied.

2.4.5 Non-equilibrium approaches

By nature, the dynamics of learning are usually more difficult to solve than the static Gibbs distribution as they are a Markov process on the weight probability distribution in discrete time. The transition probability density between student parameters is determined by their update rule, e.g., gradient descent on the error (2.7), and therefore depends on the current student parameter, the error function, the teacher rule, and the set of examples (over which a quenched average has to be performed). An exact

[24] This is for the standard approach only true for noiseless teachers, however, recently the approach has been extended to noisy rules (Weigt 1997).

[25] Unlike the replica method, which formally first performs the quenched average and then constructs a mean-field theory, this approach first constructs a (self-consistent) mean-field theory and then averages over the disorder.

calculation is usually infeasible and several approaches have been attempted to simplify the above to make the dynamics solvable.

The to date most successful approach has been to study on-line learning, where a stream of examples is presented to the network and patterns are not recycled. In this case the quenched average over a whole example set becomes an annealed average over a single example. Furthermore, in the thermodynamic limit one can identify a set of macroscopic order parameters (similar to the ones of equilibrium calculations) that are self-averaging and thus replace the student parameters as the dynamic variables. That is, the probability distribution of the student parameters evolving stochastically in time is replaced by order parameters evolving deterministically [for early papers see, e.g., (Biehl and Schwarze 1992; Biehl and Riegler 1994; Biehl and Schwarze 1995; Kinouchi and Caticha 1992; Kinouchi and Caticha 1995; Copelli and Caticha 1995)].

This approach will be taken in Chapters 5 and 6, where it will be introduced in more detail. However, let us briefly mention those alternatives not pursued.

Stochastic approximation theory: The student parameters remain the dynamic variables and, in order to control the variance of their distribution, the limit of small learning rate is taken and the Markov process can be approximated by Fokker-Planck equations. No restrictions on the input dimension have to be made and the training set can be finite. However, this approach is valid only for learning close to attractive fixed points of the student parameter dynamics (Heskes and Kappen 1991; Heskes 1994; Radons 1993; Hansen et al. 1993; Leen and Moody 1993; Orr and Leen 1993; Leen and Orr 1994)

Dynamic mean-field theory: An approach originating from spin-glass theory with stochastic training and quenched disorder (finite training set). It employs sophisticated (and notoriously difficult) techniques such as generating functionals and path integrals and has to date only been solved for Boolean perceptrons with Ising weights (Horner 1992b; Horner 1992a; Horner 1993).

Dynamic replica theory: A recent technique stemming from spin-glass theory (Coolen et al. 1996), where replicas are used to calculate "sub-shell averages[26]", that allow the microscopic dynamics to be rewritten in terms of macroscopic order parameters (functions). This method is the natural extension of the techniques used in Chapters 5 and 6 to finite training sets (and both on-line and batch learning). A program has been proposed but not yet carried out (Mace and Coolen 1997).

[26] An integration over all student parameters resulting in the same set of order parameters.

2.4.6 Some concluding remarks

Let us just briefly summarize the above. In the average case analysis, from a statistical mechanics perspective, it is not necessary to take the number of examples per network parameter to infinity as in classical statistics. However, this is paid for by the limit of infinite input dimension. This is necessary in order to make the quantities of interest, such as the generalization and training error (which are usually calculated indirectly via the free energy and/or order parameters) self-averaging, i.e., the quantities of interest are deterministic functions since fluctuations depending on the data instances are only of $\mathcal{O}(N^{-\frac{1}{2}})$ which decay in this limit. In other words, the probability distributions of these quantaties are highly peaked Gaussians with variance of $\mathcal{O}(1/N)$ which become δ-functions for $N \to \infty$. Furthermore, the thermodynamic limit allows the introduction of (self-averaging) macroscopic order parameters which are functions of the microscopic student parameters and describe the system exactly.

Finally, we distinguish between equilibrium approaches, where we are interested in the properties after exhaustive learning (i.e., infinite training time) and non-equilibrium approaches, where training dynamics are studied.

Chapter 3

Threshold-Induced Phase Transitions in Perceptrons

Abstract

Error rates of a Boolean perceptron with threshold and either spherical or Ising constraint on the weight vector are calculated for storing patterns from biased input and output distributions derived within a one-step replica symmetry breaking (RSB) treatment. For unbiased output distribution and non-zero stability of the patterns, we find a critical load, α_p, above which two solutions to the saddlepoint equations appear; one with higher free energy and zero threshold and a dominant solution with non-zero threshold. We examine this second-order phase transition and the dependence of α_p on the required pattern stability, κ, for both 1RSB and replica symmetry (RS) in the spherical case and for 1RSB in the Ising case.

3.1 Introduction

Since the ground-breaking work by Gardner (1988) on the storage capacity of the Boolean perceptron, the replica (Mézard et al. 1987) and other techniques of statistical mechanics have been successfully employed to investigate many aspects of the performance of simple neural network models. While most of the research concentrated on exploring the learning ability and network capacity below saturation [for a review see (Watkin et al. 1993; Seung et al. 1992) and references therein], this chapter will concentrate on the errors of a Boolean perceptron above its saturation limit, or capacity limit α_c, working within a replica framework. Earlier studies (Erichsen and Thuemann 1993; Majer et al. 1993; Krauth and Mézard 1989) have particularly examined the cases of zero stability of the stored patterns, the effect of different error

functions on the error rates, and the distribution of pattern stabilities. Here, this work will be extended (and their results scrutinized) by allowing for a threshold and biased input and output distributions and investigate both real valued (spherical constraint) and binary weights (Ising constraint).

Even in this simple network, the Boolean perceptron, the extra degree of freedom introduced by the threshold offers new insights and triggers new phenomena which have not been observable previously. In the case of arbitrary input and output distributions, the threshold can always compensate for a ferromagnetic bias in the weights but not vice versa, which will allow us to argue that the paradigm of eliminating the threshold in favour of a ferromagnetic bias in the weights, which has been adopted in some papers [e.g., (Gardner 1988; Gutfreund and Stein 1990; Wendemuth et al. 1993)], should be reconsidered. The introduction of a threshold enables the elimination of the input distribution bias by suitably rescaling the threshold and stability.

Especially intriguing is the rôle of the threshold for non-zero stability and unbiased output distributions; above some critical pattern load α_p, two solutions to the saddle-point equations are found: one has a non-zero threshold and a lower free energy with an asymptotic error rate of 50%, the other is identical to that of a perceptron without threshold and exhibits a higher free energy with an asymptotic error rate above 50%. The order parameters show a second-order phase transition at the bifurcation point and have different asymptotic values.

The results gained in this chapter are also of interest since we can apply them to calculate the storage capacity of a class of networks with variable architecture produced by constructive algorithms in Chapter 4. This problem is especially interesting since so far explicit results for the capacity of multi-layer networks are restricted to zero stability and zero output bias. Furthermore, it is intriguing to compare the capacity of fixed architecture models with unconstrained optimization with variable architecture models with constrained optimization.

The chapter is structured as follows. In Section 3.2 the capacity and saturation problems are explained and the model, the Boolean perceptron with threshold (and spherical or Ising constraint), is introduced for correlated output and input distributions. In Section 3.3, the replica framework is explained briefly and the one-step replica symmetry breaking (1RSB) calculations is outlined for the two constraints for both the free energy and distribution of pattern stabilities. This is followed in Section 3.4 by a discussion of the error rate and the pattern stability distribution of the two Boolean perceptron models. We finish with a discussion of the significance of the results and some concluding remarks in Section 3.5.

3.2 The capacity and saturation problem

In this section, we will briefly define the problem of learning random dichotomies together with the capacity and the VC dimension of a network. We will then introduce the simplest neural network model, the Boolean perceptron and subsequently elucidate the difference between the two approaches. The section closes with a short explanation of the saturation problem.

3.2.1 The capacity problem

Both the capacity and VC dimension problem consider whether a learner, e.g., a neural network, can implement a set of $p = \alpha N$ random dichotomies given as a (training) set of input-output pairs $(\boldsymbol{\xi}^\mu, \zeta^\mu)$ $(\mu = 1, \ldots, p)$ with $\boldsymbol{\xi}^\mu \in \{-1, 1\}^N$ and $\zeta^\mu \in \{-1, 1\}$, where both the inputs $\boldsymbol{\xi}^\mu$ and the outputs ζ^μ are drawn independently from their respective probability distributions $P(\boldsymbol{\xi}^\mu)$ and $P(\zeta^\mu)$. Note, that one can use a symmetric $\{-1, 1\}$ output representation without loss of generality (w.l.o.g.), since an asymmetric representation $\{0, 1\}$ can be mapped to a symmetric one by redefining $\zeta^{\mu\prime} = (2\zeta^\mu - 1)$.

The difference between the capacity and the VC dimension definition is roughly that the former is probabilistic and distribution dependent, whereas the latter is not. The VC dimension d_{VC} (Vapnik and Chervonenkis 1971; Vapnik 1982) is formally defined as the maximal set size p for which an input example set can be shattered, i.e., mapped to any desired output set. The capacity limit is defined as the set size p for which a random input example set can be correctly mapped to a random output set with probability 1/2, i.e., when taking the *quenched* average over input and output sets (Cover 1965; Hertz et al. 1991). In the thermodynamic limit of infinite input dimension, $N \to \infty$, this probability can be conveniently redefined as arbitrarily close to 1 as the probability of implementability becomes a step function. Furthermore, the capacity limit α_{c} is usually defined not in terms of the set size p but as the ratio between p and the number of free parameters in the network, which, for example, for a two-layer network in the thermodynamic limit is NK, where N is the input dimension and K the number of hidden units. For the distributions it is generally assumed that the binary input distribution is independent of the pattern and site indices μ and j

$$P(\xi_j^\mu) = P(\xi) = \frac{1}{2}(1 + m_{\mathrm{i}})\delta(1 - \xi) + \frac{1}{2}(1 - m_{\mathrm{i}})\delta(1 + \xi). \tag{3.1a}$$

The random output distribution is also chosen to be independent of the pattern index

$$P(\zeta^\mu) = P(\zeta) = \frac{1}{2}(1 + m_{\mathrm{o}})\delta(1 - \zeta) + \frac{1}{2}(1 - m_{\mathrm{o}})\delta(1 + \zeta), \tag{3.1b}$$

where m_{i} and m_{o} represent the input and output biases respectively.

3.2.2 The Boolean perceptron

The simplest neural network, the perceptron (see Figure 2.1), is parameterized by its synaptic weight vector $\boldsymbol{W} \in \mathbb{R}^N$ and threshold $\theta \in \mathbb{R}$, performing the mapping

$$\sigma^\mu = \operatorname{sgn}\left(\frac{1}{\sqrt{N}}\boldsymbol{W}\cdot\boldsymbol{\xi}^\mu - \theta\right) = \operatorname{sgn}(h^\mu) \tag{3.2}$$

where $\operatorname{sgn}(x)$ is the sign of x and h^μ is termed the activation of the perceptron.

A further property, which has a strong influence on the capacity limit is the error measure used to train the perceptron. Here it is defined as

$$E = \sum_\mu \Theta\left(\kappa - \zeta^\mu h^\mu\right), \tag{3.3}$$

where $\Theta(x)$ is the Heaviside step function, which is 1 for $x > 0$ and 0 otherwise and κ is the stability with which the patterns are required to be stored. The choice of the stability κ has a significant impact on the capacity limit since it fixes the minimal allowed distance between a pattern and the decision hyperplane of the perceptron. This error function, often referred to as the Gardner–Derrida cost function, counts the number of patterns which are implemented with a stability less than κ, i.e., all misclassified patterns but also some correctly classified patterns for $\kappa > 0$. The Gardner–Derrida cost function leads to the least number of errors theoretically achievable by any "practical" learning algorithm [e.g., (Frean 1992)].

3.2.3 The capacity and VC dimension of the Boolean perceptron

The capacity and the VC-dimension of the perceptron can be calculated quite straightforwardly by looking at the *growth function* $C(p, N)$, defined as the number of different binary functions that can be implemented by the network on any set of p examples $\{\boldsymbol{\xi}^\mu\}$ in N input dimension. For the Boolean perceptron this is just (Cover 1965; Hertz et al. 1991)

$$C(p,N) = 2\sum_{i=0}^{N-1}\binom{p-1}{i} \approx 2^{2N}\frac{1}{2}\left\{1 + \operatorname{erf}\left[\sqrt{\frac{p}{2}}\left(\frac{2}{N} - 1\right)\right]\right\}, \tag{3.4}$$

where the approximation is valid for large N due to the Gaussian limit of the binomial coefficients. The capacity can therefore by directly read off as ($\alpha_c = 2N/N = 2$). For the VC dimension d_{VC}, one can exploit the binomial formula for $2^n = (1+1)^n$ to show that $d_{\mathrm{VC}} = N$ (Hertz et al. 1991) for the perceptron without threshold. Note that for a perceptron with threshold the result is $d_{\mathrm{VC}} = N + 1$ (Vapnik 1982). Intuitively, this

result is clear since for $p > N + 1$ there will always be a Boolean function which is not linearly separable. The generalization of the capacity of the perceptron to arbitrary stability is in comparison much less trivial and has been calculated by Gardner (1988) within a replica framework.

3.2.4 The saturation problem

Once the memory capacity of a neural network, here the perceptron, is reached, it is obvious that loading with further examples must necessarily lead to imperfect storage of the training set. Some patterns may have erroneous outputs others may have a stability below the required stability κ. The saturation problem therefore aims at finding the fraction of errors a network makes above its capacity limit. A calculation of the distribution of pattern stabilities furthermore allows for a more detailed investigation into the strategies employed by the network in this task.

3.3 Replica calculation of Boolean perceptron

In this section we outline the replica calculation for the Boolean perceptron trying to learn a set of random dichotomies above its saturation limit α_c. The calculation is similar to (Erichsen and Thuemann 1993; Majer et al. 1993) for real valued weights and a spherical constraint and to (Krauth and Mézard 1989) for binary weights, i.e., an Ising constraint; however, we allow for a threshold and biased output and input distributions. In the following the real valued weight Boolean perceptron will be referred to as the spherical (Boolean) perceptron, whereas the binary valued weight Boolean perceptron will be referred to as the Ising (Boolean) perceptron. This section is divided into six parts. In Section 3.3.1, the replica framework and the calculation for the free energy of the perceptron above saturation is introduced briefly, followed by the explicit evaluation within the replica symmetric (RS) and the one-step replica symmetry breaking (1RSB) ansätze in Sections 3.3.2–3.3.4. In Section 3.3.5, the same framework is then extended to calculate the distribution of pattern stabilities for the perceptron. In Section 3.3.6, we outline the differences for the calculations of the Ising perceptron and present the resulting equations.

3.3.1 Free energy of the spherical perceptron

The calculation for the perceptron above saturation will be performed within an equilibrium statistical mechanics approach, where we calculate the free energy (per input)

$$f = -\frac{1}{N\beta} \log Z, \tag{3.5}$$

which is assumed to be self-averaging in the thermodynamic limit $N \to \infty$ with finite example load $\alpha = p/N$. In the following, we will be interested only in the minimum error possible and will therefore consider zero-temperature Gibbs learning ($\beta \to \infty$). Hence

$$\langle\langle f \rangle\rangle = -\lim_{\beta\to\infty} \lim_{N\to\infty} \frac{1}{N\beta} \langle\langle \log Z \rangle\rangle = -\lim_{\beta\to\infty} \lim_{N\to\infty} \frac{1}{N\beta} \left\langle\!\!\left\langle \log \int d\mu(\boldsymbol{W}) e^{-\beta E} \right\rangle\!\!\right\rangle \tag{3.6}$$

where $\langle\langle\cdot\rangle\rangle$ is the quenched average over the distribution of patterns, consisting of integrations over biased input and output distributions (3.1) and E is the training error (3.3). Furthermore, in the case of real valued weights, a spherical constraint is enforced on the weight vector

$$d\mu(\boldsymbol{W}) = \delta(\boldsymbol{W}\cdot\boldsymbol{W} - N) \prod_{i=1}^{N} dW_i \,, \tag{3.7}$$

to avoid the invariance $(\boldsymbol{W}, \kappa) \to (\lambda\boldsymbol{W}, \lambda\kappa)$. To be able to pick out the two possible error sources (wrongly-on, where the requested target is $\zeta^\mu = -1$ but the output is $\sigma^\mu = 1$ and wrongly-off, where $\zeta^\mu = 1$ but $\sigma^\mu = -1$), auxiliary variables, ϵ^+ and ϵ^-, are introduced in the error function (3.3)

$$E = \sum_\mu \Theta(\kappa - \lambda^\mu) [\epsilon^- \Theta(\zeta^\mu) + \epsilon^+ \Theta(-\zeta^\mu)] = \sum_\mu V(\lambda^\mu, \kappa, \zeta^\mu), \tag{3.8}$$

where $\lambda^\mu = \zeta^\mu h^\mu$ and V is the error measure for a single example both introduced for convenience.[1] The derivatives of the free energy with respect to ϵ^+ or ϵ^- at $\epsilon^+ = \epsilon^- = 1$ will give us the wrongly-on and wrongly-off errors respectively.

To be able to perform the quenched average we make use of the replica trick (Edwards and Anderson 1975)

$$\langle\langle \log Z \rangle\rangle = \lim_{n\to 0} \frac{\langle\langle Z^n \rangle\rangle - 1}{n} = \lim_{n\to 0} \frac{1}{n} \log \langle\langle Z^n \rangle\rangle \,, \tag{3.9}$$

[1] This is also consistent with earlier work (Majer et al. 1993) and allows in principle a calculation for an arbitrary cost function.

calculating $\langle\langle Z^n \rangle\rangle$ for integer n (which can be seen as n replicas of the same physical system) and continuing analytically to $n = 0$. The calculation is performed by employing standard replica techniques [see (Gardner 1988; Majer et al. 1993) for details]. The integrals over the identically distributed inputs and outputs can be decomposed into p-fold (in the examples) and N-fold (in the input dimensions) products by introducing the auxiliary variable $\lambda^\sigma = \zeta h^\sigma$ in the error function (3.8) and its Lagrange multiplier $\hat{\lambda}^\sigma$ through the integral representation (A.1) of the resulting δ-functions. Through the subsequent integration over the inputs, natural order parameters[2] emerge

$$Q^{\sigma\rho} = \frac{1}{N} \boldsymbol{W}^\sigma \cdot \boldsymbol{W}^\rho \qquad (\text{for } \sigma < \rho), \qquad M^\sigma = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} W_i^\sigma, \tag{3.10}$$

together with their Lagrange multipliers[3], $\hat{Q}^{\sigma\rho}$ and $\hat{M}^\sigma$, created by the integral representation of the respective δ-functions. Similarly, the spherical constraint (3.7) is also rewritten as an integral over the Lagrange multiplier $\hat{E}^\sigma$. After some more algebraic manipulations the replicated partition function can be simplified to

$$\langle\langle Z^n \rangle\rangle = \int_{-\infty}^{\infty} \int_{-\mathrm{i}\infty}^{\mathrm{i}\infty} \left(\prod_\sigma \frac{dM^\sigma d\hat{E}^\sigma}{2\pi} \right) \left(\prod_{\sigma<\rho} \frac{dQ^{\sigma\rho} d\hat{Q}^{\sigma\rho}}{2\pi} \right)$$
$$\times \exp\left\{ N \left[G_0(\hat{Q}^{\sigma\rho}, \hat{E}^\sigma) + \alpha G_\mathrm{r}(Q^{\sigma\rho}, \theta^\sigma, M^\sigma) + \frac{1}{2} \sum_\sigma \hat{E}^\sigma - \sum_{\sigma<\rho} Q^{\sigma\rho} \hat{Q}^{\sigma\rho} \right] \right\}, \tag{3.11}$$

where

$$G_0(\hat{Q}^{\sigma\rho}, \hat{E}^\sigma) = \log \left\{ \int_{-\infty}^{\infty} \prod_\sigma dW^\sigma \exp\left[-\frac{1}{2} \sum_\sigma \hat{E}^\sigma W^\sigma W^\sigma + \sum_{\sigma<\rho} \hat{Q}^{\sigma\rho} W^\sigma W^\rho \right] \right\} \tag{3.12}$$

is the prior constraint Hamiltonian and

$$G_\mathrm{r}(Q^{\sigma\rho}, \theta^\sigma, M^\sigma) = \log \left\langle \int_{-\infty}^{\infty} \left(\prod_\sigma \frac{d\lambda^\sigma d\hat{\lambda}^\sigma}{2\pi} \right) \exp\left\{ -\beta V(\lambda^\sigma, \kappa, \zeta) - \mathrm{i} \sum_\sigma \hat{\lambda}^\sigma \lambda^\sigma \right.\right.$$
$$\left.\left. -\mathrm{i}\zeta \sum_\sigma \hat{\lambda}^\sigma (\theta^\sigma - m_\mathrm{i} M^\sigma) - \frac{1}{2}(1 - m_\mathrm{i}^2) \left[\sum_\sigma \hat{\lambda}^\sigma \hat{\lambda}^\sigma + 2 \sum_{\sigma<\rho} \hat{\lambda}^\sigma \hat{\lambda}^\rho Q^{\sigma\rho} \right] \right\} \right\rangle_\zeta \tag{3.13}$$

is the replicated Hamiltonian, and where $\langle\cdot\rangle_\zeta$ denotes an average over the output distribution.

[2] One could also allow $\rho = \sigma$. In this case $Q^{\sigma\sigma} = 1$ and $\hat{Q}^{\sigma\sigma} = \hat{E}^\sigma$ due to the spherical constraint.

[3] The contribution of $\hat{M}^\sigma$ actually vanishes in the thermodynamic limit.

3.3.2 The replica symmetric ansatz

To make further progress one has to make an assumption for the structure of the replica space. The simplest assumption is that replica symmetry holds (which is believed to correspond usually to a connected solution space):

$$\begin{array}{rcl} Q^{\sigma\rho} = q_1 \quad \text{and} \quad \hat{Q}^{\sigma\rho} = \hat{q}_1 & & (\text{for } \forall\ \sigma < \rho), \\ M^\sigma = M, \quad \theta^\sigma = \theta, \quad \text{and} \quad \hat{E}^\sigma = \hat{E} & & (\text{for } \forall\ \sigma). \end{array} \tag{3.14}$$

Inserting the above ansätze into Eqs. (3.12) and (3.13) and taking the $n \to 0$ limit yields

$$G_0^{\mathrm{RS}} = \frac{1}{2}\frac{\hat{q}_1}{\hat{E}+\hat{q}_1} - \frac{1}{2}\log(\hat{E}+\hat{q}_1), \tag{3.15a}$$

$$G_{\mathrm{r}}^{\mathrm{RS}} = \left\langle \int Dt\, \log\left[\mathcal{F}_{\mathrm{RS}}(t,\beta,q_1,\kappa,\zeta\theta)\right] \right\rangle_\zeta, \tag{3.15b}$$

where all integrals without explicit limits are from $-\infty$ to $+\infty$, Dt is the Gaussian measure $Dt = dt \exp(-t^2/2)/\sqrt{2\pi}$ and the function $\mathcal{F}_{\mathrm{RS}}$ is given by

$$\mathcal{F}_{\mathrm{RS}}(t,\beta,q_1,\kappa,\zeta\theta) = \int \frac{d\lambda}{\sqrt{2\pi(1-q_1)(1-m_{\mathrm{i}}^2)}} \exp\left(-\beta\mathcal{E}_{\mathrm{RS}}\right), \tag{3.16a}$$

with the auxilliary function $\mathcal{E}_{\mathrm{RS}}$

$$\mathcal{E}_{\mathrm{RS}}(\lambda,t,q_1,\kappa,\zeta\theta) = V(\lambda,\kappa,\zeta) + \frac{[\psi(\lambda)+\sqrt{q_1}\,t]^2}{2x}, \tag{3.16b}$$

where $x = \beta(1-q_1)$ and

$$\psi(\lambda) = \frac{\lambda + \zeta(\theta - m_{\mathrm{i}}M)}{\sqrt{1-m_{\mathrm{i}}^2}}. \tag{3.17}$$

When taking the $\beta \to \infty$ in order to access the ground state with least errors only, one has to distinguish two regimes. Below the capacity limit, α_{c} (above which the training error becomes strictly positive), $q_1 < 1$ even for $\beta \to \infty$. At and above the capacity limit, $q_1 \to 1$ for $\beta \to \infty$, because the volume of the individual solution spaces vanishes. Therefore, the self-consistent ansatz is made for $\alpha \geq \alpha_{\mathrm{c}}$ that $x = \beta(1-q_1)$ remains finite in the zero-temperature limit. In this case, the integral over λ in (3.16) can be calculated by the saddlepoint method; the exponentional is evaluated at $\lambda = \lambda^{\mathrm{opt}}$, where λ^{opt} minimizes $\mathcal{E}_{\mathrm{RS}}$ for given t. After calculating $\lambda^{\mathrm{opt}}(t)$ for the Gardner–Derrida

cost function and eliminating $\hat{q}_1$ and $\hat{E}$, the RS free energy at $\epsilon^+ = \epsilon^- = 1$ simplifies to:

$$\langle\langle f_{\mathrm{RS}} \rangle\rangle = \alpha \left\langle \int_{-\tau}^{\sqrt{2x}-\tau} Dt \frac{(t+\tau)^2}{2x} + H\left(\sqrt{2x} - \tau\right) \right\rangle_{\zeta} - \frac{1}{2x}, \tag{3.18}$$

where

$$\tau = \psi(\kappa) = \frac{\kappa + \zeta(\theta - m_{\mathrm{i}} M)}{\sqrt{1 - m_{\mathrm{i}}^2}}. \tag{3.19}$$

The free energy has to be evaluated at the saddlepoints with respect to the variables x and θ. The capacity limit, α_{c}, can be calculated from the saddlepoint equations by taking the limit $x \to \infty$. A more detailed examination of the free energy and the saddlepoint equations is deferred to Section 3.3.4.

Above the capacity limit α_{c}, it may be argued that different solutions can misclassify different patterns and that consequently the solution space may in general be disconnected. It has been previously shown that in the case of the Gardner–Derrida cost function the replica symmetric saddlepoint is locally unstable above saturation (Gardner and Derrida 1988), and the Parisi scheme of successive steps of replica symmetry breaking (RSB) (Mézard et al. 1987) must be employed.

3.3.3 The 1RSB ansatz

Here, we will restrict ourselves to a 1RSB calculation. We note that it has been shown recently that, for the spherical perceptron with the Gardner–Derrida cost function, infinitely many RSB steps are necessary to derive the correct result (Whyte and Sherrington 1996). Although 1RSB is, therefore, incorrect it is a very good approximation, as a two-step RSB calculation carried out for the spherical perceptron without threshold yielded only minor corrections in the free energy (Whyte and Sherrington 1996).

The ansatz for the 1RSB is that $Q^{\sigma\rho}$ is a $n \times n$ matrix

$$(Q^{\sigma\rho})_{nn} = \begin{pmatrix} Q_1 & Q_0 & \cdots & Q_0 \\ Q_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & Q_0 \\ Q_0 & \cdots & Q_0 & Q_1 \end{pmatrix}_{nn}, \tag{3.20a}$$

where Q_0 is a $m \times m$ matrix with elements q_0 and Q_1 is a $m \times m$ matrix with 0 on the diagonal and q_1 elsewhere. The ansatz for $\hat{Q}^{\sigma\rho}$ has the same block structure as for $Q^{\sigma\rho}$

with matrices $\hat{Q}_0$ and $\hat{Q}_1$. One further assumes

$$M^\sigma = M, \quad \theta^\sigma = \theta, \quad \text{and} \quad \hat{E}^\sigma = \hat{E} \qquad (\text{for } \forall\, \sigma), \tag{3.20b}$$

similar to the RS case (3.14). The order parameters q_1 and q_0 can be interpreted as the typical overlap between pairs of weight vectors in the same and different solution spaces, respectively. Clearly, if the solution space is connected $q_0 \equiv q_1$, which is the case for $\alpha \leq \alpha_c$, and we recover replica symmetry. Again using the above ansätze in Eqs. (3.12) and (3.13) and taking the $n \to 0$ limit yields

$$G_0^{\text{RSB}} = \frac{1}{2}\frac{\hat{q}_0}{(\hat{E}+\hat{q}_1) - m(\hat{q}_1 - \hat{q}_0)} - \frac{1}{2}\log(\hat{E}+\hat{q}_1) - \frac{1}{2m}\log\left(1 - \frac{\hat{q}_1 - \hat{q}_0}{\hat{E}+\hat{q}_1}\right), \tag{3.21a}$$

$$G_r^{\text{RSB}} = \left\langle \int Dt \frac{1}{m} \log\left[\mathcal{F}_{\text{RSB}}(t, m, \beta, q_0, q_1, \kappa, \zeta\theta)\right] \right\rangle_\zeta, \tag{3.21b}$$

where the function $\mathcal{F}_{\text{RSB}}$ is given by

$$\mathcal{F}_{\text{RSB}}(t, m, \beta, q_0, q_1, \kappa, \zeta\theta) = \int Dz \left[\int \frac{d\lambda}{\sqrt{2\pi(1-q_1)(1-m_i^2)}} \exp\left(-\beta \mathcal{E}_{\text{RSB}}\right) \right]^m, \tag{3.21c}$$

with the auxilliary function $\mathcal{E}_{\text{RSB}}$

$$\mathcal{E}_{\text{RSB}}(\lambda, t, z, q_0, q_1, \kappa, \zeta\theta) = V(\lambda, \kappa, \zeta) + \frac{[\psi(\lambda) + \sqrt{q_0}\, t + \sqrt{q_1 - q_0}\, z]^2}{2\beta\sqrt{1-q_1}} \tag{3.21d}$$

where ψ as in (3.17).

Similar to the RS case, we are interested in the $\beta \to \infty$ limit where $q_1 \to 1$ with $x = \beta(1-q_1)$ finite. The λ-integral in (3.21c) can again be evaluated at the saddlepoint $\lambda = \lambda^{\text{opt}}$, where λ^{opt} minimizes $\mathcal{E}_{\text{RSB}}$ for given z and t. Furthermore, the replica space dimension $m \to 0$ ($\beta \to \infty$) as only one solution is accessed and it becomes exponentially unlikely that any other solution is visited (Mézard et al. 1987). We therefore make a second self-consistent ansatz that $w = m/(1-q_1)$ remains finite in the zero-temperature limit. After some algebra, including determining $\lambda^{\text{opt}}(z,t)$ for the Gardner–Derrida cost function and elimination of $\hat{q}_1$, $\hat{q}_0$, and $\hat{E}$, the 1RSB free energy for $\epsilon^+ = \epsilon^- = 1$ is given by

$$\langle\langle -f_{\text{RSB}} \rangle\rangle = \frac{\alpha}{wx}\left\langle \int Dt \log\left[\mathcal{F}_{\text{RSB}}(t, w, x, q_0, \kappa, \zeta\theta)\right] \right\rangle_\zeta + \frac{q_0}{2x(1+w\Delta q)} + \frac{\log(1+w\Delta q)}{2wx}, \tag{3.22a}$$

where τ is as before (3.19), $\Delta q = 1 - q_0$, and the function $\mathcal{F}_{\mathrm{RSB}}$ has simplified to

$$\mathcal{F}_{\mathrm{RSB}}(t, w, x, q_0, \kappa, \zeta\theta) = \int_{-\frac{\mu}{\sqrt{\Delta q}}}^{\frac{\sqrt{2x}-\mu}{\sqrt{\Delta q}}} Dz \exp\left[-\frac{w}{2}\left(\sqrt{\Delta q}\, z + \mu\right)^2\right] + H\left(\frac{\mu}{\sqrt{\Delta q}}\right) + e^{-wx} H\left(\frac{\sqrt{2x}-\mu}{\sqrt{\Delta q}}\right), \quad (3.22b)$$

with $\mu = \tau + \sqrt{q_0}\, t$. The free energy has to be evaluated at the saddlepoints with respect to the variables w, x, q_0, and θ.

3.3.4 Saddlepoint equations and training error

Examining both the RS (3.18) and the 1RSB (3.22) free energies more closely, one sees that the ferromagnetic bias, M, of the weight vector (3.10) appears only in the definition of τ (3.19) and can be set to zero without loss of generality (w.l.o.g.)[4]. The order parameter M is therefore superfluous, i.e., any ferromagnetic bias in the couplings can be compensated by an adjustment of the threshold θ. This is in contrast to the usual paradigm, which eliminates θ in favour of M [e.g., (Gardner 1988; Gutfreund and Stein 1990; Wendemuth et al. 1993)], and therefore reduces the number of actual perceptron parameters. However, this is clearly only possible if $m_{\mathrm{i}} \neq 0$ and will lead to large absolute values of M for small m_{i}. Note, it has been remarked in (Wendemuth et al. 1993; Wendemuth 1995c), that for finite size systems $|m_{\mathrm{i}}| \gg 2p^{-\frac{1}{2}}$, because otherwise M will not be able to yield the required saddlepoint value, whereas no such problem exists when allowing for a threshold.

We further note that the bias of the input distribution, m_{i}, appears only in the definition of τ (3.19) also and its sole influence is a rescaling of the threshold and the stability. Therefore, a biased input distribution has the same effect on the performance of the perceptron as the increase of the stability for an unbiased input distribution. This can be understood in geometric terms. If the input distribution is unbiased, input vectors lie randomly distributed on the edges of the unit hypercube and two distinct patterns have a typical overlap of zero. Biased patterns on the other hand are correlated and have a typical overlap of m_{i}^2 with each other, i.e., they concentrate on a "conelike" section of the hypercube. The typical distance between patterns is therefore reduced by $\sqrt{1 - m_{\mathrm{i}}^2}$. Any solution of the weight vector corresponds to a hyperplane which seperates the two kind of patterns. The achieved stability is half the distance of the two correctly classified patterns with the shortest seperation across this plane

[4]The fact that M is redundent is a direct consequence of the fact that the integral over $\hat{M}$ does not contribute in the thermodynamic limit.

and hence the stability decreases by $\sqrt{1-m_{\mathrm{i}}^2}$ as well. Only at zero stability does the increase of the input bias have no effect on the performance of the perceptron. In the following, we will therefore set $m_{\mathrm{i}} = 0$ w.l.o.g..

The saddlepoint equation of the derivative of the free energy with respect to θ at $\epsilon^+ = \epsilon^- = 1$ gives

$$0 = \left\langle \zeta \int_{-\tau}^{\sqrt{2x}-\tau} Dt\,(t+\tau) \right\rangle_\zeta \qquad \text{and} \tag{3.23a}$$

$$0 = \left\langle \zeta \int Dt\, t \log\left[\mathcal{F}_{\mathrm{RSB}}(t,w,x,q_0,\kappa,\zeta\theta)\right] \right\rangle_\zeta \tag{3.23b}$$

for RS and 1RSB, respectively. For zero bias, one can readily see that $\theta = 0$ is always a solution to this and the other saddlepoint equations; regaining the results of the perceptron without threshold. However, this does not necessarily imply that this is the only solution to the saddlepoint equations, as demonstrated in Section 3.4.

Taking the derivatives of the free energies with respect to ϵ^- and ϵ^+ at $\epsilon^+ = \epsilon^- = 1$ and dividing by α gives the error rate (i.e., the number of errors divided by the total number of patterns) of wrongly-off and wrongly-on patterns respectively

$$\epsilon_{\mathrm{RS}}^{\mathrm{off/on}} = \frac{1}{2}(1 \pm m_{\mathrm{o}}) H\left(\sqrt{2x} - \kappa \mp \theta\right), \tag{3.24a}$$

$$\epsilon_{\mathrm{RSB}}^{\mathrm{off/on}} = \frac{1}{2}(1 \pm m_{\mathrm{o}}) \int Dt \frac{e^{-wx} H\left(\sqrt{2x} - \sqrt{q_0}\,t - \kappa \mp \theta\right)}{\mathcal{F}_{\mathrm{RSB}}(t,w,x,q_0,\kappa,\pm\theta)}, \tag{3.24b}$$

where we have set $m_{\mathrm{i}} = 0$ w.l.o.g.. In the following, the convention is adopted to use ϵ for error rates and $\hat{\epsilon}^{\mathrm{off/on}}$ for fraction of error rates, i.e., the $\hat{\epsilon}^{\mathrm{off/on}} = \epsilon^{\mathrm{off/on}}/\epsilon$.

We note that numerically no difference between the total training error[5] and the free energy is found in the thermodynamic limit for both RS and 1RSB and conclude that the normalized entropy, $s = S/N$, must diverge sublinearly or logarithmically for $\beta \to \infty$. One can calculate the first-order finite temperature correction of the free energy for both RS and 1RSB analytically, and find that it is negative and proportional to $\log(\Delta q)$, and equal to the low-temperature entropy. Explicitly, one finds

$$s_{\mathrm{RS}} = \frac{1}{2}\log(\Delta q)\left[1 + \alpha\left\langle H(-\tau) - H\left(\sqrt{2x} - \tau\right)\right\rangle_\zeta\right], \tag{3.25a}$$

$$s_{\mathrm{RSB}} = \frac{1}{2}\log(\Delta q)\left[1 + \alpha\left\langle \int Dt \int_{-\frac{\mu}{\sqrt{\Delta q}}}^{\frac{\sqrt{2x}-\mu}{\sqrt{\Delta q}}} Dz \frac{\exp\left[-\frac{w}{2}\left(\sqrt{\Delta q}\,z + \mu\right)^2\right]}{\mathcal{F}_{\mathrm{RSB}}(t,w,x,q_0,\kappa,\zeta\theta)} \right\rangle_\zeta\right]. \tag{3.25b}$$

[5] That is the error rates multiplied by α.

Unlike in the binary case, where a negative entropy is physically impossible and therefore an indication that the employed replica ansatz breaks down, a negative entropy has no such physical meaning in the real-valued case, due an arbitrary entropy offset.

3.3.5 Pattern stability distribution (PSD)

The pattern stability distribution (PSD), $P(\Lambda)$, is of interest as it provides the distance of stabilized ($\Lambda \geq \kappa$) and unstabilized patterns ($\Lambda < \kappa$) to the given threshold stability κ, i.e., it gives an idea how seriously patterns are misclassified. This extra information will be quite helpful in examining the already mentioned bifurcation point in order-parameter space in Section 3.4. For other error functions than the Gardner–Derrida cost function (e.g., the perceptron or AdaTron cost function), the integration of the probability density, $P(\Lambda)$, over the unstabilized patterns yields the error rate ϵ (Griniasty and Gutfreund 1991; Majer et al. 1993), which is otherwise inaccessible. The PSD is further of great importance to the dynamics of related attractor neural networks, by determining the basin of attraction of the memory states (Kepler and Abbott 1988; Gardner 1989).

The PSD $P(\Lambda|D)$ is in general dependent on the instances of the data set $D = \{(\boldsymbol{\xi}^\mu, \zeta^\mu) \,|\, \mu = 1, \ldots, p\}$. As we are interested in its average value $P(\Lambda) = \langle\!\langle P(\Lambda|D) \rangle\!\rangle$, we quench over the instances of the examples

$$P(\Lambda) = \langle\!\langle P(\Lambda|D) \rangle\!\rangle = \left\langle\!\!\left\langle \frac{1}{Z} \int d\mu(\boldsymbol{W}) \exp\left[-\beta \sum_\mu V(\lambda^\mu, \kappa, \zeta^\mu)\right] \delta(\Lambda - \lambda^1) \right\rangle\!\!\right\rangle, \quad (3.26)$$

where the pattern stability of pattern 1 is calculated w.l.o.g. as the pattern distribution is independent of the pattern index μ. Here, $d\mu(\boldsymbol{W})$ is the spherical constraint (3.7), but the above equation holds for any weight prior. In the thermodynamic limit, one can calculate this average using the replica trick.

$$P(\Lambda) = \lim_{n \to 0} \lim_{N \to \infty} \left\langle\!\!\left\langle Z^{n-1} \int d\mu(\boldsymbol{W}^1) \exp\left[-\beta \sum_\mu V(\lambda_1^\mu, \kappa, \zeta^\mu)\right] \delta(\Lambda - \lambda_1^1) \right\rangle\!\!\right\rangle,$$

where the superscript 1 in the integral given explicitly refers to the first replica index. The ensuing calculation is very similar to the one of the free energy except for the average over the first pattern (Kepler and Abbott 1988) and the special rôle of the first

replica index. After some algebra one finds

$$P(\Lambda) = \lim_{n\to 0}\lim_{N\to\infty} \int_{-\infty}^{\infty}\int_{-\mathrm{i}\infty}^{\mathrm{i}\infty} \left(\prod_{\sigma}\frac{dM^\sigma d\hat{E}^\sigma}{2\pi}\right)\left(\prod_{\sigma<\rho}\frac{dQ^{\sigma\rho}d\hat{Q}^{\sigma\rho}}{2\pi}\right) \mathrm{e}^{N\mathcal{G}(Q^{\sigma\rho},\hat{Q}^{\sigma\rho},\theta^\sigma,M^\sigma,\hat{E}^\sigma)}$$
$$\times\left\langle \int_{-\infty}^{\infty}\left(\prod_{\sigma}\frac{d\lambda^\sigma d\hat{\lambda}^\sigma}{2\pi}\right)\delta(\Lambda-\lambda^1)\exp\left\{-\beta V(\lambda^\sigma,\kappa,\zeta)-\mathrm{i}\sum_\sigma\hat{\lambda}^\sigma\lambda^\sigma\right.\right.$$
$$\left.\left.-\mathrm{i}\zeta\sum_\sigma\hat{\lambda}^\sigma(\theta^\sigma-m_\mathrm{i}M^\sigma)-\frac{1}{2}(1-m_\mathrm{i}^2)\left[\sum_\sigma\hat{\lambda}^\sigma\hat{\lambda}^\sigma+2\sum_{\sigma<\rho}\hat{\lambda}^\sigma\hat{\lambda}^\rho Q^{\sigma\rho}\right]\right\},\right\rangle_\zeta \qquad (3.27)$$

where $\mathcal{G}$ is up to $\mathcal{O}(N^{-1})$ identical to the Hamiltonian in Eq. (3.11). In the $N\to\infty$ limit, this integral is evaluated at the dominating saddlepoint of $\mathcal{G}$ corresponding to the saddlepoint of the free energy. However, since the exponentional term vanishes when taking $n\to 0$, only the intensive second part actually contributes to $P(\Lambda)$.

RS ansatz

For the RS ansatz one finds after some further algebra similar to the free energy calculation including taking the $n\to 0$ limit

$$P_{\mathrm{RS}}(\Lambda) = \left\langle \int Dz \frac{\exp\left[-\beta\mathcal{E}_{\mathrm{RS}}(\Lambda,t,q_1,\kappa,\zeta\theta)\right]}{\sqrt{2\pi(1-q_1)(1-m_\mathrm{i}^2)}\mathcal{F}_{\mathrm{RS}}(t,\beta,q_1,\kappa,\zeta\theta)}\right\rangle_\zeta, \qquad (3.28)$$

where $\mathcal{F}_{\mathrm{RS}}$ and $\mathcal{E}_{\mathrm{RS}}$ are taken from Eq. (3.16). For $\beta\to\infty$ above the capacity limit α_c, $P_{\mathrm{RS}}(\Lambda)$ can be simplified along the lines of (Amit et al. 1990; Griniasty and Gutfreund 1991) as the λ-integral in $\mathcal{F}_{\mathrm{RS}}$ can be evaluated at its saddlepoint λ^{opt}.

Since for $\beta\to\infty$ the exponentional is dominated by λ^{opt}, it also follows that the argument of the Dt-integral in Eq. (3.28) only contributes with a non-zero (unit) weight for $\Lambda=\lambda^{\mathrm{opt}}$ yielding the simple result

$$P_{\mathrm{RS}}(\Lambda) = \left\langle \int Dt\,\delta\left[\Lambda-\lambda^{\mathrm{opt}}(x,\zeta\theta,t)\right]\right\rangle_\zeta.$$

Calculating λ^{opt} for the Gardner–Derrida cost function equates the RS probability density

$$P_{\mathrm{RS}}(\Lambda) = \left\langle \delta(\Lambda-\kappa)\int_{-\tau}^{\sqrt{2x}-\tau} Dt + \frac{\Theta(\Lambda-\kappa)}{\sqrt{2\pi(1-m_\mathrm{i}^2)}}\exp\left[-\frac{1}{2}\phi^2\right]\right.$$
$$\left. + \frac{\Theta\left(\kappa-\Lambda-\sqrt{1-m_\mathrm{i}^2}\sqrt{2x}\right)}{\sqrt{2\pi(1-m_\mathrm{i}^2)}}\exp\left[-\frac{1}{2}\phi^2\right]\right\rangle_\zeta, \qquad (3.29)$$

where x and θ have to be evaluated at the saddlepoint of the free energy (3.18) as mentioned above and $\phi = \psi(\Lambda)$ with ψ as in (3.17). The PSD has three terms, a δ-function contribution for $\Lambda = \kappa$, i.e., at the error boundary, and two Gaussian contributions, which leave a gap of width $\sqrt{1-m_i^2}\sqrt{2x}$.

1RSB ansatz

For the 1RSB ansatz one finds similarly

$$P_{\mathrm{RSB}}(\Lambda) = \left\langle \int Dt \frac{1}{\mathcal{F}_{\mathrm{RSB}}(t,m,\beta,q_0,q_1,\kappa,\zeta\theta)} \int Dz \frac{\exp\left[-\beta\mathcal{E}_{\mathrm{RSB}}(\Lambda,t,z,q_0,q_1,\kappa,\zeta\theta)\right]}{\sqrt{2\pi(1-q_1)(1-m_i^2)}} \right.$$
$$\left. \times \left[\int \frac{d\lambda}{\sqrt{2\pi(1-q_1)(1-m_i^2)}} \exp\left(-\beta\mathcal{E}_{\mathrm{RSB}}(\lambda,t,z,q_0,q_1,\kappa,\zeta\theta)\right) \right]^{m-1} \right\rangle_\zeta , \quad (3.30)$$

where $\mathcal{F}_{\mathrm{RSB}}$ and $\mathcal{E}_{\mathrm{RSB}}$ are taken from Eqs. (3.21c) and (3.21d). Similarly to RS, $P_{\mathrm{RSB}}(\Lambda)$ can be simplified along the lines of (Majer et al. 1993) for $\beta \to \infty$ as the λ-integrals can be evaluated at their common saddlepoint λ^{opt}. Similar to RS there is only a contribution to the Dz-integral for $\Lambda = \lambda^{\mathrm{opt}}$ (noting that $m \to 0$ for $\beta \to \infty$). Inserting λ^{opt} for the Gardner–Derrida cost function simplifies the PSD of 1RSB to

$$P_{\mathrm{RSB}}(\Lambda) = \left\langle \int Dt \frac{\mathcal{N}_{\mathrm{RSB}}(t,w,x,q_0,\kappa,\zeta\theta)}{\mathcal{F}_{\mathrm{RSB}}(t,w,x,q_0,\kappa,\zeta\theta)} \right\rangle_\zeta , \quad (3.31a)$$

where the denominator $\mathcal{F}_{\mathrm{RSB}}$ is identical to (3.22b) and the numerator is given by

$$\mathcal{N}_{\mathrm{RSB}}(t,w,x,q_0,\kappa,\zeta\theta) = \delta(\Lambda-\kappa) \int_{-\frac{\mu}{\sqrt{\Delta q}}}^{\frac{\sqrt{2x}-\mu}{\sqrt{\Delta q}}} Dz \exp\left[-\frac{w}{2}\left(\sqrt{\Delta q}\,z+\mu\right)^2\right]$$
$$+ \frac{\Theta(\Lambda-\kappa)}{\sqrt{2\pi\Delta q(1-m_i^2)}} \exp\left[-\frac{\varrho^2}{2\Delta q}\right]$$
$$+ \frac{\Theta\left(\kappa-\Lambda-\sqrt{1-m_i^2}\sqrt{2x}\right)}{\sqrt{2\pi\Delta q(1-m_i^2)}} \exp\left[-\frac{\varrho^2}{2\Delta q} - wx\right], \quad (3.31b)$$

where $\varrho = \phi + \sqrt{q_0}\,t$ and the values of the order parameters x, w, q_0, and θ are again determined by the saddlepoint of the free energy (3.22).

Comparing the 1RSB with the RS PSDs, one finds three similar contributions, a δ-peak at the stability κ and two exponentional terms, separated by a gap-width of

$\sqrt{1-m_i^2}\sqrt{2x}$. In general, one finds (Majer et al. 1993) that 1RSB has a smaller gap, which is formally due to a reduced saddlepoint value of x, and a reduced weight of the δ-contribution. The 1RSB distribution has also lost the Gaussian form of the RS distribution, due to the presence of the denominator and the integration over t. One further finds a correction to the third contribution, which represents unstabilized, i.e., erroneous, patterns, which has acquired an extra supressive exponentional term e^{-wx}. As already pointed out in Section 3.3.4, the rôle of a non-zero input bias is the rescaling of the threshold θ, the stability κ and the pattern stability Λ with a factor of $\sqrt{1-m_i^2}$, and can therefore be set to zero w.l.o.g..

It is worth mentioning that the gap and the δ-peak are a feature of training algorithms above saturation employing the Gardner–Derrida cost function (Wendemuth 1995a). This is due to the fact that an algorithm achieving least errors attempts to stabilize the least unstabilized pattern, until any movement of the hyperplane will destabilize a pattern lying on the threshold decision boundary, leading to a fraction of patterns exactly on the decision boundary and leaving a gap between stabilized and unstabilized patterns. The above work has been complemented by a numerical study (Wendemuth 1995b), where the numerical PSD exhibits a gap and a δ-peak which are both finite but smaller than the theoretical 1RSB predictions within the accuracy of the simulations. This is consistent with a recent proof (Whyte and Sherrington 1996) which showed that any model exhibiting a gap in the PSD necessitates infinitely many RSB steps.

3.3.6 Ising perceptron

In the case of the Ising perceptron the calculation is very similar. In fact, the calculation of the replicated Hamiltonian G_r (3.13) is exactly the same as it only depends on the quenched average over the training examples. The difference is therefore mainly in the prior constraint Hamiltonian G_0 (3.12), where the integration over weight space is performed. Since the weight vector of the Ising perceptron is binary, i.e., $\boldsymbol{W} \in \{-1,1\}^N$, the measure in weight space [see Eq. (3.7)] becomes a sum $\int d\mu(\boldsymbol{W}) = \prod_{i=1}^N \sum_{W_i=\pm 1}$, and all terms with the Lagrange multiplier $\hat{E}^\sigma$ associated with the spherical constraint vanish in Eq. (3.11). The prior constraint Hamiltonian equates to

$$G_0^{\mathrm{I}}(\hat{Q}^{\sigma\rho}) = \log\left\{\prod_\sigma \exp\left[-\sum_{\sigma<\rho}\hat{Q}^{\sigma\rho}W^\sigma W^\rho\right]\right\}. \tag{3.32}$$

Again, using two ansätze for the structure in replica space, RS and 1RSB identical to those made in Section 3.3.1, one finds

$$G_0^{\mathrm{IRS}}(\hat{q}_1) = -\frac{\hat{q}_1}{2} + \int Dt \log\left[2\cosh\left(t\sqrt{\hat{q}_1}\right)\right], \tag{3.33a}$$

$$G_0^{\mathrm{IRSB}}(\hat{q}_1, \hat{q}_0) = -\frac{\hat{q}_1}{2} + \frac{m}{2}(\hat{q}_1 - \hat{q}_0) + \frac{1}{m}\int Dt \log\left[\int Dz\, 2\cosh\left(t\sqrt{\hat{q}_0} + z\sqrt{\hat{q}_1 - \hat{q}_0}\right)\right]^m, \tag{3.33b}$$

where IRS(B) stands for the RS or 1RSB ansatz for the Ising perceptron.

Great care has to be taken in the $\beta \to \infty$ limit, which is discussed in detail in (Krauth and Mézard 1989), here we will only outline the main results. One finds that the entropy s (per input) of the RS solution is negative for $\alpha > \alpha_{\mathrm{S}}^{\mathrm{I}}$ with $q_1 < 1$ in the zero-temperature limit and is therefore incorrect above $\alpha_{\mathrm{S}}^{\mathrm{I}}$. Studying the 1RSB solutions identifies $\alpha_{\mathrm{S}}^{\mathrm{I}}$ as the capacity limit $\alpha_{\mathrm{c}}^{\mathrm{I}}$. The capacity limit can therefore be calculated from the root of the RS free energy, which is identical to the temperature adjusted entropy $\beta f = -s$ since the training error is identical to zero and is given by

$$\langle\!\langle s_{\mathrm{IRS}} \rangle\!\rangle = -\frac{1}{2}(1 - q_1)\hat{q}_1 + \int Dt \log\left[2\cosh\left(t\sqrt{\hat{q}_1}\right)\right] + \alpha \left\langle \int Dt \log\left[H\left(\frac{\tau + \sqrt{q_1}\, t}{\sqrt{1 - q_1}}\right)\right]\right\rangle_\zeta, \tag{3.34}$$

and has to be evaluated at its saddlepoint with respect to q_1, $\hat{q}_1$, and θ.

The RS free energy only becomes strictly positive for $\alpha > \alpha_{\mathrm{E}}^{\mathrm{I}}$ where $q_1 \to 1$ with $x = \beta(1 - q_1)$ finite and the RS free energy of the Ising perceptron can be simplified (Gardner and Derrida 1988), resulting in

$$\langle\!\langle f_{\mathrm{IRS}} \rangle\!\rangle = \alpha \left\langle \int_{-\tau}^{\sqrt{2x}-\tau} Dt\, \frac{(t+\tau)^2}{2x} + H\left(\sqrt{2x} - \tau\right)\right\rangle_\zeta - \frac{1}{\pi x}, \tag{3.35}$$

which is identical to the RS free energy of the spherical perceptron (3.18) but for a constant $2/\pi$ in the last α-independent term. The RS solution of the Ising perceptron at α is consequently the same as the RS solution of the spherical perceptron at $\tilde{\alpha} \equiv \pi\alpha/2$, which holds also for error rates and the distribution of pattern stabilities. The RS solution of the Ising perceptron will therefore not be discussed further.

However, as already mentioned above, the RS solution is incorrect for $\alpha > \alpha_{\mathrm{S}}^{\mathrm{I}}$ and $\beta > \beta_{\mathrm{c}}$, where one finds 1RSB solutions, which are characterized by $q_1 = 1$ and $\hat{q}_1 = \infty$ for finite β. One further finds $m = \beta_{\mathrm{c}}/\beta$, $\hat{q}_0 \to 0$ and makes the self-consistent ansätze

that $v = m\beta$ and $y = m\sqrt{\hat{q}_0}$ are finite in the zero-temperature limit. Inserting these ansätze back into (3.33), one finds $G_0^{\mathrm{IRSB}}(\infty, \hat{q}_0) = G_0^{\mathrm{IRS}}(y^2)/m$. The replicated Hamiltonian G_r^{IRSB} (3.21) is calculated similarly to the spherical perceptron, with the above ansätze becoming equivalent to $x \to 0$ and $w \to \infty$ with wx finite. The 1RSB free energy of the Ising perceptron is therefore given by

$$\langle\langle -f_{\mathrm{IRSB}} \rangle\rangle = \frac{\alpha}{v} \left\langle \int Dt \log \left[\mathcal{F}_{\mathrm{IRSB}}(t, v, y, q_0, \kappa, \zeta\theta) \right] \right\rangle_\zeta + \frac{1}{v} \int Dt \log \left[2\cosh(yt) \right] - \frac{\Delta q y^2}{2v}, \quad (3.36a)$$

for $\epsilon^+ = \epsilon^- = 1$ and the function $\mathcal{F}_{\mathrm{IRSB}}$ is

$$\mathcal{F}_{\mathrm{IRSB}}(t, v, y, q_0, \kappa, \zeta\theta) = e^{-v} + (1 - e^{-v}) H\left(\frac{\mu}{\sqrt{\Delta q}} \right), \quad (3.36b)$$

with μ as before. The free energy has to be evaluated at its saddlepoint with respect to the variables v, y, q_0, and θ. The normalized entropy of the Ising perceptron can be shown to be identical to zero (Krauth and Mézard 1989).

Identical to the spherical perceptron the ferromagnetic bias on the weights M and the bias of the input distribution m_i can be set to zero w.l.o.g.. One also finds as before that $\theta = 0$ is always a solution to the saddlepoint equation for zero output bias and the error rates of wrongly-off and wrongly-on patterns are given respectively by

$$\epsilon_{\mathrm{IRSB}}^{\mathrm{off/on}} = \frac{1}{2}(1 \pm m_o) \int Dt \frac{e^{-v} H\left(\sqrt{2x} - \sqrt{q_0}\, t - \kappa \mp \theta \right)}{\mathcal{F}_{\mathrm{IRSB}}(t, v, y, q_0, \kappa, \pm\theta)}. \quad (3.37)$$

The pattern stability distribution (PSD) density $P_{\mathrm{IRSB}}(\Lambda)$ of the Ising perceptron within a 1RSB ansatz can be calculated similarly to the spherical perceptron in Section 3.3.5. In the zero-temperature limit, $x \to 0$ and $w \to \infty$ with wx finite is employed to find

$$P_{\mathrm{IRSB}}(\Lambda) = \left\langle \int Dt \frac{\mathcal{N}_{\mathrm{IRSB}}(t, v, y, q_0, \kappa, \zeta\theta)}{\mathcal{F}_{\mathrm{IRSB}}(t, v, y, q_0, \kappa, \zeta\theta)} \right\rangle_\zeta, \quad (3.38a)$$

where the denominator $\mathcal{F}_{\mathrm{IRSB}}$ is identical to (3.36b) and the numerator is given by

$$\mathcal{N}_{\mathrm{IRSB}}(t, v, y, q_0, \kappa, \zeta\theta) = \frac{\left[\Theta(\Lambda - \kappa) + e^{-v} \Theta(\kappa - \Lambda) \right]}{\sqrt{2\pi \Delta q (1 - m_i^2)}} \exp\left[-\frac{\varrho^2}{2\Delta q} \right], \quad (3.38b)$$

with ϱ as in (3.31b) and the values of the order parameters y, v, q_0, and θ are eval-

uated at the saddlepoint of the free energy (3.36). The PSD of the Ising perceptron has a common Gaussian numerator centered around ϱ, but for an extra exponential surpression of the unstabilized patterns $\Lambda < \kappa$ proportional to e^{-v}.

Comparing the PSDs of the spherical and the Ising perceptron shows no difference within the RS ansatz besides the already mentioned rescaling of α. However, one finds striking differences within the 1RSB treatment: the gap in the distribution as well as the δ-peak contribution at the threshold boundary κ have vanished in the PSD of the Ising perceptron in contradiction to (Wendemuth 1995a) (see Section 3.3.5). However, this could be explained by the fact that the Ising perceptron cannot adjust its decision boundary continuously due to the discreteness of the weights. Therefore, one may expect that unstabilized patterns lie arbitrarily close to the decision boundary and that patterns do not accumulate at the threshold stability.

Whereas it has been shown previously that the 1RSB ansatz for the spherical perceptron is not exact (Whyte and Sherrington 1996), which is formally due to the gap in the PSD, the Ising perceptron does not exhibit this gap and there has been some argument whether 1RSB in the macrocanonical approach is exact for this model[6]. Krauth and Mézard (1989) have carried out a second RSB step and have found no solution different to the 1RSB result, although one should mention that most of their numeric work was carried out around the capacity limit. Fontanari and Meir (1993) have calculated the entropy of the Ising perceptron in a microcanonical approach and found that their RS solution is identical to the 1RSB solution in the canonical approach. They calculated that the microcanonical RS saddlepoint is locally stable for all α, which also suggests that the ansatz is correct, as a breakdown would require that the RS saddlepoint is locally stable but globally unstable even for $\alpha \to \infty$. A third approach by Horner (1992b) investigating the learning dynamics using dynamic mean field theory which does not rely on the replica trick, indicates a slightly different picture. He finds that the fluctuation dissipation theorem (FDT) holds for high temperatures and the dynamics are ergodic validating the use of RS. For lower temperatures ergodicity is broken but one finds that a quasi FDT (QFDT) holds, parameterized by a variable m, which has a similar rôle as the 1RSB parameter m but has to be chosen inconsistently to the choice of m in replica theory. These dynamics were found to be strictly stable for infinite times indicating that no further RSB steps are necessary in this regime. Furthermore, there exists a third regime with additional diverging time scales which

[6] 1RSB has been proved to be exact for several models, e.g., for the generalized Sherrington-Kirkpatrick (SK) spin glass with $p = \infty$ spin interactions, which is equivalent to the random energy model and can be solved exactly (Gross and Mézard 1984).

corresponds to further breaking of replica symmetry[7]. However, the relevance of dynamic mean field theory for validating replica ansätze is debatable. Recently, Weigt and Engel (1997) have revisited the problem by studying the multi-fractal distribution of weight cells (Monasson and O'Kane 1994). Within their approach they find for $\alpha > \alpha_S^I$ a disontinuous transition from RS to 1RSB in parts of the distribution where RS is still locally stable. For even larger[8] $\alpha \approx 1.245$, the 1RSB solution reverts back to the RS solution and RS is again correct. There has been some speculation, whether RS in the multi-fractal approach corresponds approximately to 1RSB in the macrocanonical approach as taken here. This would corroborate the view that 1RSB is correct for the Ising perceptron for large α, however, does contradict Krauth and Mézard (1989) findings of no numerical solution for two-step RSB around the capacity limit. However, since the equivalence is not perfect, this remains an open question.

3.4 Discussion

Calculating the saddlepoint solutions for the order parameters and the error rates as a function of the normalized example number α for a range of stabilities κ and output biases m_0, one finds striking differences in the solution space to the case of a perceptron without threshold even for zero (output) bias (Majer et al. 1993; Krauth and Mézard 1989). Since the zero bias results are the most intriguing, most of the discussion will be limited to this special case, where just the introduction of a single free parameter to the perceptron, a threshold, changes the space of solutions accessible to the perceptron radically even for unbiased input and output distributions.

First, the order-parameter solution space and the total error rates of the spherical perceptron and the Ising perceptron are examined in Sections 3.4.1 and 3.4.2, respectively. This is followed by a discussion of the pattern stability distribution (PSD) in Section 3.4.3 and the assessment of the influence of a biased output distribution in Section 3.4.4. As discussed in Section 3.3.4, a biased input distribution can be absorbed through rescaling of the stability and therefore need not be discussed in more detail. Finally, the dependence of the phase transition in parameter solution space is investigated as a function of the stability κ in Section 3.4.5.

[7] An explicit phase diagram is given only for the perceptron and adatron cost functions.

[8] This corresponds to the discontinuous transition to perfect generalization in supervised learning with Ising perceptron student and teacher.

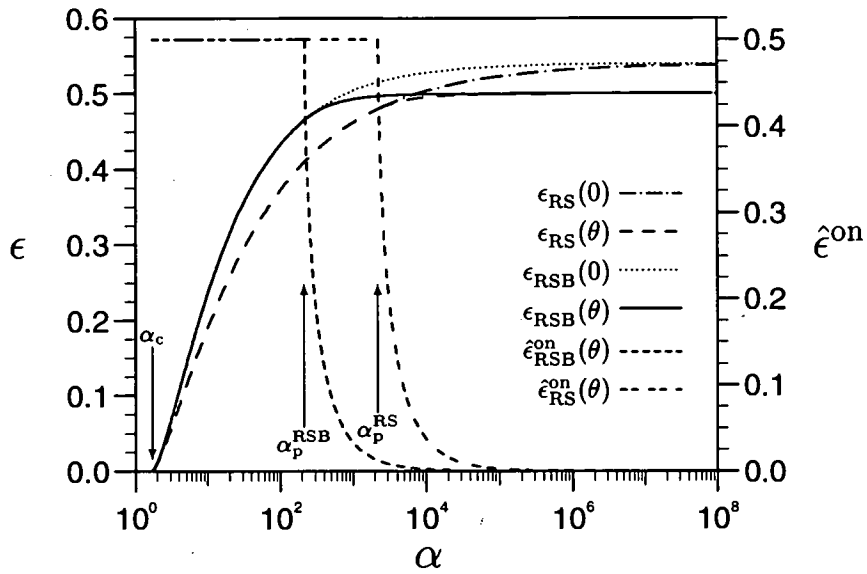

Figure 3.1. The total error rate ϵ of the spherical perceptron as a function of α for $\kappa = 0.1$ is predicted by 1RSB to be larger than the estimate of RS. For $\alpha > \alpha_c$ both theories initially predict a portion of $1/2$ for wrongly-on errors $\hat{\epsilon}^{on}$ indicating zero threshold (see also Figure 3.2). Above a critical α_p, $\hat{\epsilon}^{on}$ decreases abruptly and quickly approaches zero signalling a solution with non-zero threshold. This solution exhibits a lower asymptotic error rate than a perceptron without threshold. The predicted value of α_p is smaller for 1RSB than for RS.

3.4.1 Error rates and order-parameter solution space of the spherical perceptron

In Figure 3.1 the total error rates ϵ and the percentage of wrongly-on errors $\hat{\epsilon}^{on}$ is shown for the spherical perceptron in both the RS and the 1RSB ansatz, $m_o = 0$, and $\kappa = 0.1$ as a function of α. Below the capacity limit α_c the error rate $\epsilon(\alpha)$ is identically zero. For $\alpha > \alpha_c$, the RS estimate of the error rate is always below the 1RSB estimate for all $\alpha > \alpha_c$ and replica symmetry is broken as expected (Gardner and Derrida 1988). In Figure 3.2 the 1RSB overlap q_0 is plotted as a function of α in the same scenario, indicating the degree of replica symmetry breaking.

In Figure 3.1 one can also see that for $\alpha > \alpha_c$, the proportion of wrongly-on errors $\hat{\epsilon}^{on}$ is initially $1/2$. This corresponds to the threshold θ being identical to zero as one can see in Figure 3.2. This solution, for both RS and 1RSB, could have been expected from examining Eqs. (3.23). However, above a critical value of the normalized example number $\alpha > \alpha_p$, a second solution to the saddlepoint equations exists, which is characterized by a non-zero threshold and a fraction of wrongly-on errors smaller than

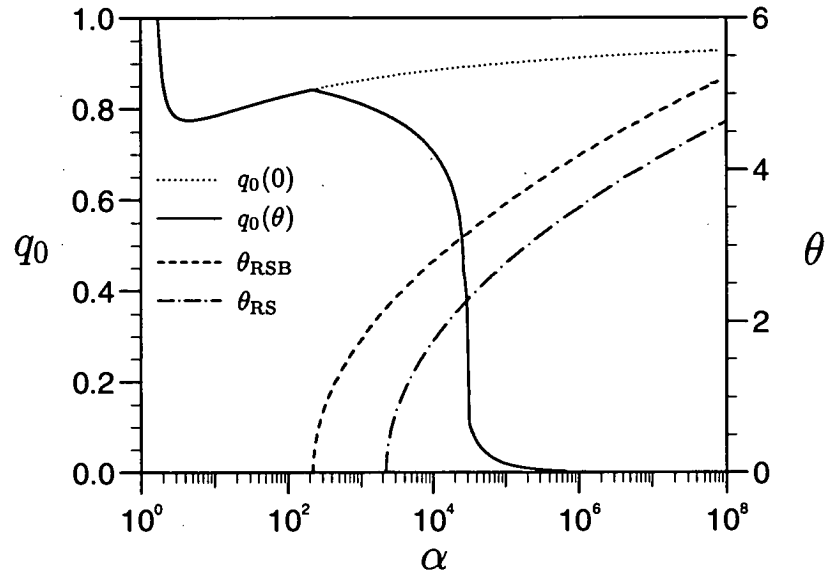

Figure 3.2. The prediction of the 1RSB overlap q_0 for the solution goes to zero as $\alpha \to \infty$ for the perceptron with threshold, whereas it approaches one with zero threshold. The threshold as a function of α in the 1RSB and the RS ansatz is also included.

1/2 (see Figures 3.1 and 3.2). The value of α_p can be seen to be significantly smaller for 1RSB than for RS. This is found to be true for all finite stabilities, which will be examined in more detail in Section 3.4.5 where the phase transition is examined as a function of the threshold stability κ.

One should note that although a zero threshold solution (to which we will refer as θ_0) still exists and is identical to the solution of a perceptron without threshold, it is, however, not a physically viable solution for the perceptron with threshold as it exhibits a higher free energy (i.e., larger error rate, as shown in Figure 3.1) than the non-zero threshold solution (which will be referred to as θ) and is therefore to be neglected in the thermodynamic limit. This illustrates that a solution to the saddlepoint equations found for any given replica ansatz is not necessarily unique.

Going back to Figure 3.1, one finds for further increasing $\alpha \to \infty$ the error rate of the θ_0 solution approaches an asymptotic error rate which is higher than 1/2, the asymptotic error rate of the θ solution. The qualitative difference between the error rates can be better understood by examining the PSD and will therefore be deferred to the discussion of the error limit in Section 3.4.3.

The bifurcation point in solution space is a second-order phase transition as all order parameters [see e.g., $\theta(\alpha)$ and $q_0(\alpha)$ in Figure 3.2] are continuous but non-differentiable

for $\alpha = \alpha_p$. In particular, for the threshold the numerical data strongly indicates the functional relationship

$$\theta \propto [\log(\alpha) - \log(\alpha_p)]^\gamma \tag{3.39}$$

for both RS and 1RSB theory with an exponent γ which is in very good agreement with the mean-field theory exponent of $1/2$, and a prefactor which is κ-dependent and consistently larger for 1RSB. One further finds spontaneous symmetry breaking in the space of thresholds θ as the solution is invariant under sign change of θ. The threshold θ therefore corresponds to the magnetization in ferromagnetic systems. Similarly, $1/\log(\alpha)$ plays the rôle of the temperature T. The external field in this case is the output bias m_o as it breaks the symmetry in θ-space and smoothes out the phase transition, as will be studied more closely in Section 3.4.4.

The phase transition at α_p stems from the competition between optimising the weights (or hyper plane angle) and a deterministic bias in the output of the perceptron which is controlled by the threshold. Whereas it is self-evident that for a biased output distribution it is also sensible to bias the output of the student with a non-zero threshold, this is only the case for an unbiased output distribution when the error rate becomes large enough for a given stability κ. To understand this more clearly, the distribution of pattern stabilities together with the total error rate is studied around the phase transition in Section 3.4.3.

In order-parameter space one finds qualitatively very different solutions, as can be seen in Figure 3.2 for the order parameters q_0 and θ. For the θ solution, the threshold increases towards infinity following the above functional relationship of Eq. (3.39) and q_0 decaying to zero, where $q_0 \propto 1/\alpha$ numerically with the possibility of minor logarithmic corrections. For the θ_0 solution on the other hand q_0 approaches one. To investigate the functional behaviour of the θ_0 solution in more detail, one can expand the free energy using the numerically justified ansätze $x \propto 1/\alpha$ and $w \propto \sqrt{\alpha}$ for $\alpha \to \infty$ leading to a power-law behaviour. The prefactors reported previously (Majer et al. 1993) are inconsistent with our analytical solutions and the numerical data. In particular, the solutions of the order parameters are to leading order

$$\Delta q = \frac{2}{\log(\alpha)}, \quad x = \frac{9}{4}\frac{e^{\kappa^2/2}}{\alpha[\log\alpha]^{3/2}}, \quad \text{and} \quad w = \frac{4}{9}\sqrt{\pi}\, e^{-\kappa^2/4}\sqrt{\alpha}\,[\log\alpha]^{9/4}. \tag{3.40}$$

These solutions are, however, only good approximations provided Δq is small and $\log\alpha \gg \log(\log\alpha)$, i.e., in general $\alpha \gg 10^{10}$ and is therefore not very accurate in the region where numerical solutions were obtained. The solutions suggest that for

increasing α the degree of RSB becomes more severe as m $[m = wx/\beta]$ and $(1 - q_1)$ $[1 - q_1 = x/\beta]$ decay to zero faster than the temperature.

For the solution with $\theta \neq 0$, we have not been able to find closed form asymptotic solutions to the saddlepoint equations. In fact, closed form asymptotic solution are even infeasible for the much simpler RS theory. The numerical analysis is quite difficult for both x and w; w and wx may at most diverge algebraically in $\log \alpha$ with powers smaller than one, whereas x seems to have a similar $\log \alpha$-behaviour, but the power is even smaller in magnitude and its sign seems to be κ dependent. As the error in the numerical calculation of the order-parameter solutions increase with α and the prefactor in the power laws in $\log \alpha$ are very small, we were not able to determine the value of the powers accurately. A divergent behaviour of wx indicates that the degree of RSB becomes less severe for increasing α, which should be contrasted to the θ_0 solution where the degree of RSB becomes worse.

We find the different asymptotic behaviours for the two sets of order-parameter solutions puzzling; especially, the asymptotics of the order parameter q_0 — the typical overlap between two replicas in different solution spaces. Whereas q_0 decays algebraically in α to zero for the θ solution, i.e., weight-vector solutions become totally uncorrelated, it approaches 1 logarithmically for the θ_0 solution, i.e., the weight-vector solutions become absolutely correlated. It has been argued before (Majer et al. 1993) that this asymptotic behaviour for the spherical perceptron without threshold is incorrect (and 1RSB must therefore be inexact at least for high storage level), since one should expect q_0 to approach 0 for $\alpha \to \infty$ as in this limit any weight vector should perform equally well on the training data. More precisely, for loads α greater than the capacity limit α_c, the perceptron classifies only a subset of the examples correctly and misclassifies the rest. For moderate loads and small error rates, there must be a significant overlap between the sets of examples two weight-vector solutions classify correctly. Therefore, the average overlap between weight-vector solutions should be non-zero and hence, $q_0 > 0$. For very large α and large error rates ϵ, the smallest possible overlap between two sets of correctly classified examples should decrease[9] and since the patterns are uncorrelated, the correlations between their respective weight-vector solutions should decrease similarly. Hence, the smallest average overlap scale in the replica ansatz should approach 0 for $\alpha \to \infty$.

We will later come back to this argument and the issue of the breakdown of 1RSB in the light of the asymptotics of the order parameter q_0, especially in comparison with

[9] In fact, for the perceptron with zero threshold and $\kappa > 0$, one finds $\epsilon > 1/2$ for α large enough and the sets of correctly classified patterns for two solutions could be disjoint.

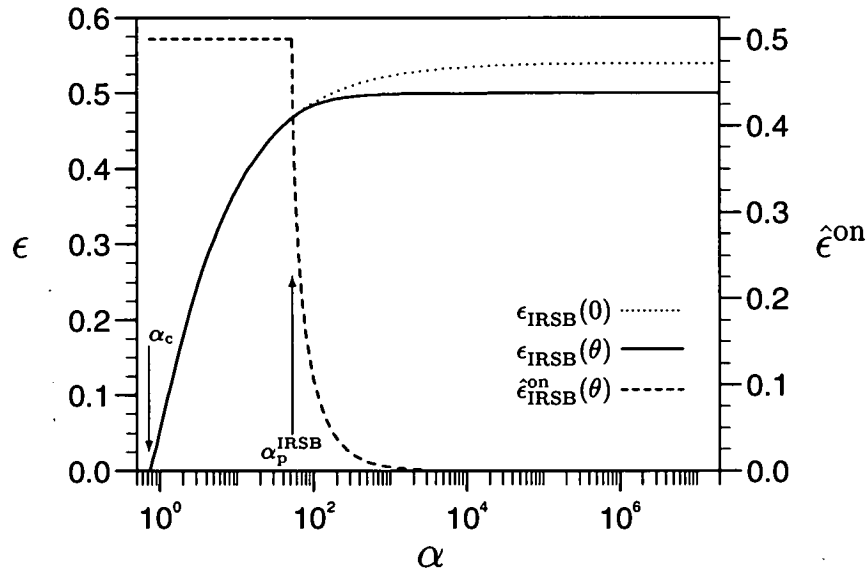

Figure 3.3. The error rates ϵ are shown as a function of α with $\kappa = 0.1$ for the Ising perceptron within the 1RSB ansatz. Similar to the spherical perceptron there is initially only one solution with a fraction of $1/2$ for wrongly-on errors $\hat{\epsilon}^{\rm on}$ and zero threshold (see Figure 3.4). Again one finds a bifurcation point in solution space at a critical $\alpha_{\rm p}$, which is smaller than for the spherical perceptron and similar behaviour of the fraction of $\hat{\epsilon}^{\rm on}$ errors.

the asymptotic solutions of the Ising perceptron, which will be presented below.

3.4.2 Order-parameter solution space of the Ising perceptron

As mentioned in Section 3.3.6, whereas it has been established that 1RSB is not exact for the spherical perceptron there has been some argument whether 1RSB is exact for the Ising perceptron, and it is therefore useful to compare the solution in order-parameter space and their asymptotics for the two weight priors.

In Figures 3.3 and 3.4 the evolution of the error rates and the fractions of wrongly-on errors and the corresponding values of the order parameters q_0 and θ are shown for the Ising perceptron in the 1RSB ansatz for the same scenario, i.e., for $m_{\rm o} = 0$ and $\kappa = 0.1$. One finds certain similarities but also striking differences to the results for the spherical perceptron. At the capacity limit $\alpha_{\rm c}^{\rm I}$, q_0 does not approach 1 as in the spherical perceptron, indicating a single solution in weight space, but a finite value $q_0 < 1$, i.e., several correlated solutions exist at $\alpha_{\rm c}^{\rm I}$. As for the spherical perceptron, the solution to the saddlepoint equations is initially unique and exhibits a zero threshold. As the error increases for growing α, one finds a similar second-order phase transition in order-parameter space, with the emergence of a second solution to the saddlepoint

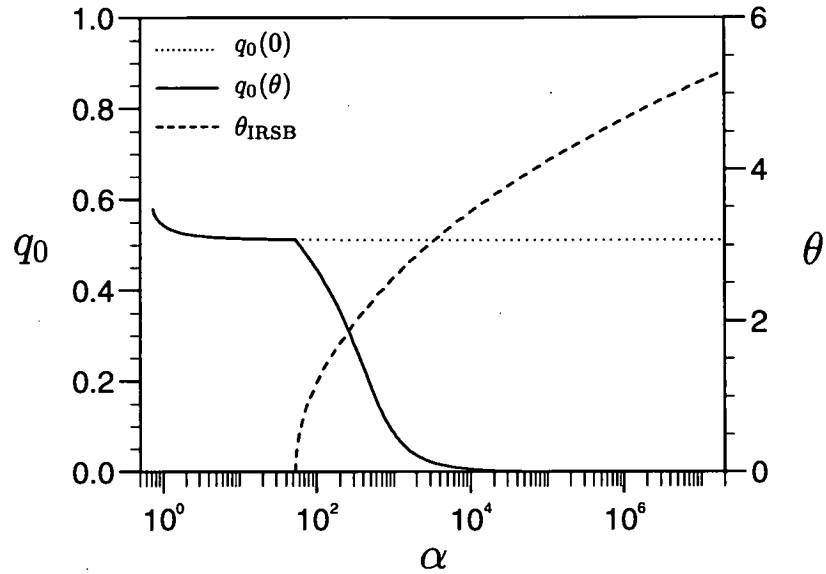

Figure 3.4. The 1RSB overlap q_0 of the Ising perceptron for the θ solution goes to zero as $\alpha \to \infty$, whereas it approaches a finite value ($q_0 = 0.51114$) for the θ_0 solution. The threshold as a function of α grows logarithmically to infinity.

equations characterized by a non-zero threshold at $\alpha = \alpha_{\mathrm{p}}$. For the threshold, the numerical data supports the same mean field power-law behaviour of Eq. (3.39).

In the asymptotic limit of infinite example load, the RSB overlap q_0 again approaches a finite limit for the θ_0 solution, which is κ-dependent but always strictly less than 1, whereas it converges against zero for the θ solution, following a power-law decay $q_0 \propto \alpha^{-1}$. One further finds for the Ising perceptron without threshold that the order parameter y approaches a finite value as q_0, whereas v, which is the equivalent of wx in the spherical case, decays as $v \propto 1/\sqrt{\alpha}$, similar to the spherical perceptron, indicating that the degree of RSB becomes more severe for increasing α.

We would like to point out that the asymptotic result of q_0 violates the qualitative argument in (Majer et al. 1993), which demands $q_0 \to 0$ for $\alpha \to \infty$, although it has been argued that 1RSB may be exact for the Ising perceptron. In order to exclude with certainty that no solution to the saddlepoint exists which is characterized by $q_0 \to 0$, substantial numerical and analytical work has been carried out for the special case $\kappa = 0$ even for $\alpha > 10^{10}$, where the numerical solutions to the saddlepoint equations of Eq. (3.36) become unreliable due to the inherent inaccuracy of the numerical integrations. The saddlepoint equations were expanded in a Taylor series in v, for which the dominant terms of all integrals can be solved analytically for $\kappa = 0$. This expansion was in excellent agreement with previous results and also provided accurate results

for α values, where the solutions to the full equations were inaccurate. However, an extensive numeric search for solutions with q_0 and y small was unsuccessful even for $\alpha > 10^{200}$. This could be confirmed by the fact that algebraic saddlepoint equations, obtained by expanding the equations further for small q_0 and y, have only unphysical complex roots.

In the numerical analysis for the θ solution, it is again difficult to find the exact power-law exponents and possible logarithmic corrections. However, exact relationships between order parameters can be established. The conjugate order parameter y decays as $1/\sqrt{\alpha}$. This suggests a relationship with q_0 as $y^2 \propto \hat{q}_0$, and indeed $q_0/y^2 \sim 1$ holds for large α. The order parameter v, diverges logarithmically in α and one finds $v/\theta \sim 2\kappa$ asymptotically, again indicating that the degree of RSB of the θ solution decreases for large α.

These functional relationships can be confirmed by a series expansion of the free energy around $q_0 = 0$ and $y = 0$, followed by an asymptotic expansion in θ and v assuming[10] w.l.o.g. $\theta > 0$. The later expansion is, however, only valid in the region where $\theta - \kappa \gg 1$. The saddlepoint equations of $\partial f/\partial y$ and $\partial f/\partial \theta$ give to leading order $q_0 = y^2$ and $v = 2\kappa\theta$, in agreement with the numerical data. Inserting $\partial f/\partial v$ in $\partial f/\partial q_0$ gives

$$\sqrt{q_0} = y = \frac{\log(2)}{\kappa\sqrt{\alpha}} \tag{3.41a}$$

The remaining saddlepoint equation $\partial f/\partial v$, determining θ,

$$\exp\left[-\frac{1}{2}(\theta-\kappa)^2\right] - \exp\left[-\frac{1}{2}(\theta+\kappa)^2\right] = \frac{\sqrt{2\pi}\log(2)}{\kappa\alpha}, \tag{3.41b}$$

does not have a closed form solution. However, for $\theta\kappa \gg 0$ an approximate solution can be found

$$\theta \approx \kappa + \sqrt{2}\left[\log\left(\frac{\kappa\alpha}{\sqrt{2\pi}\log(2)}\right)\right]^{1/2}. \tag{3.41c}$$

Whereas the analytical equations for y and q_0 and the solution of θ, obtained by solving Eq. (3.41b) numerically, fit the numerical solutions of the full saddlepoint equations very well even for moderate values of $2 \leq \theta \leq 6$, the closed form solution for θ (3.41c) is only a good approximation for $\kappa \geq 1$ in this region.

[10] For $\theta < 0$, one has to replace θ by $|\theta|$ in all the equations.

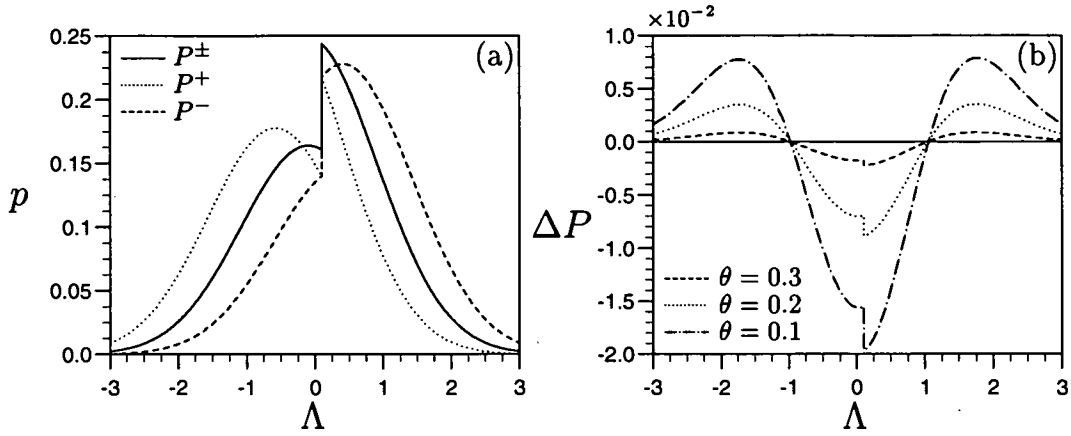

Figure 3.5. (a) The PSDs $P(\Lambda)$ of the Ising perceptron are shown as a function of the pattern stability Λ for $\kappa = 0.1$ for an example load $\alpha(\theta = 0.5) = 59.492$ close to the phase transition point $[\alpha_p(\kappa = 0.1) = 53.021]$. The θ_0 solution predicts the same PSD $P^{\pm}$ for both $\zeta = +1$ and $\zeta = -1$ patterns. For the θ solution this symmetry is broken. (b) The difference in the total PSD ($\Delta P \equiv P^{+} + P^{-} - 2P^{\pm}$) as a function of Λ for various values of α: $\alpha(0.1) = 53.266$, $\alpha(0.2) = 54.008$, and $\alpha(0.3) = 55.266$. The asymmetry of $\Delta P(\Lambda)$ caused by the discontinuity at the decision boundary leads to the reduction in the error rate of the θ solution.

3.4.3 Pattern stability distribution (PSD)

The phase transition in order parameter space is driven by the increase of the error rate ϵ for increasing example load α. It is therefore natural to examine the change in the pattern stability distribution (PSD) of the perceptron around the critical load α_p. The PSD of the Ising perceptron is examined first as it has a simpler structure (it lacks the gap and the δ-contribution of the spherical case).

In Figure 3.5(a) PSDs of the Ising perceptron for patterns with targets $\zeta = +1$ and $\zeta = -1$ are plotted for both the θ_0 and θ solution for stability $\kappa = 0.1$. The example load α was chosen slightly larger than α_p and determined as a function of the value of threshold θ, e.g., in Figure 3.5 $\alpha(\theta = 0.5) = 59.492$ [for comparison $\alpha_p(\kappa = 0.1) = 53.021$]. The $\zeta = \pm 1$ PSDs $P^{\pm}$ of the θ_0 solution are identical. For the θ solution this symmetry is broken and the PSDs P^{+} and P^{-} are distorted around the former. For $\theta > 0$ the probability in the unstabilized region $\Lambda < \kappa$ has increased for $\zeta = +1$ patterns whereas it has reduced for $\zeta = -1$ patterns, and vice versa for the stabilized region $\Lambda \geq \kappa$.

All three distributions exhibit a discontinuity at the threshold stability κ which is formally due to the exponentional factor e^{-v} in Eq. (3.38b). Although the functional form of the PSDs (3.38) is quite complicated, the PSDs have almost conserved the Gaussian form of the numerator. The means are shifted and dependent on $\zeta\theta$.

To assess the change in total error, it is more accurate to study the difference of the total PSD [$\Delta P \equiv (P^+ + P^-) - 2P^{\pm}$]. In Figure 3.5(b) ΔP is shown for three values of α even closer to the critical point. One can see that the shift of the means of the θ PSDs removes probability mass from the region close to the decision threshold κ. Furthermore $\Delta P(\Lambda)$ is almost symmetric around κ. If this symmetry were perfect, the total error rate could not be different for the θ_0 and θ solution. However, a distortion is found in the region $\Lambda \approx \kappa$, which can be most easily depicted by the discontinuity at κ, which grows for increasing $\alpha(\theta)$.

One finds quite similar results in the case of the spherical perceptron, although the gap and the δ-contribution in the PSD lead to a more complex behaviour. To make the effect of these extra features more obvious, a larger threshold stability, $\kappa = 1$, was chosen for the spherical case. In Figure 3.6(a) the 1RSB PSDs of patterns with targets $\zeta = +1$ and $\zeta = -1$ is shown for both solutions and an example load of $\alpha(\theta = 0.5) = 2.0901$ [for comparison $\alpha_p(\kappa = 1) = 1.8706$]. Again one finds that the $\zeta = \pm 1$ PSDs of the θ solution are distorted around the PSD of the θ_0 solution.

The distributions have three components. For $\Lambda < \kappa$, the distribution looks similar to a Gaussian hump with means which vary with the value of $-\theta$. This regime is seperated by a visible gap to the stabilized patterns, with a gap width which is widened for the θ solution. One further finds that the contribution of the δ-functions at $\Lambda = \kappa$ has increased for the θ solution. The main probability mass of the stabilized patterns is found in the Gaussian-like tail for $\Lambda > \kappa$.

To study the differences of the PSDs, $\Delta P(\Lambda)$ is shown for three values of α closer to α_p in Figure 3.6(b). One finds less symmetry in ΔP than for the Ising perceptron, but total probability mass has also been removed from the vicinity of $\Lambda = \kappa$. The main reduction in the error rate in this case seems to come from the widening of the gap. This difference in probability mass has been partly shifted to the δ-contributions. The increase of probability mass at the δ-peaks and the decrease of probability mass at the widened gap is, however, between a factor of 10–100 larger (and increasing for $\alpha \to \alpha_p$) than the reduction in the error rate for the α values studied in Figure 3.6(b).

It is of further interest to study the limit $\alpha \to \infty$ as the error rate of the θ_0 solution approaches its asymptotic value, which is larger than the asymptotic error rate of the θ solution of 1/2 as was shown in both Figures 3.1 and 3.3. The θ solutions in the limit of infinite example load has been shown to be characterized by a threshold increasing to infinity and the portion of wrongly-on errors decreasing rapidly to zero (see e.g., Figures 3.1 and 3.2).

To study this limit more closely, the PSDs of the spherical perceptron are shown in the 1RSB ansatz for $\kappa = 1$ and increasing α separately for the θ_0 and θ solutions

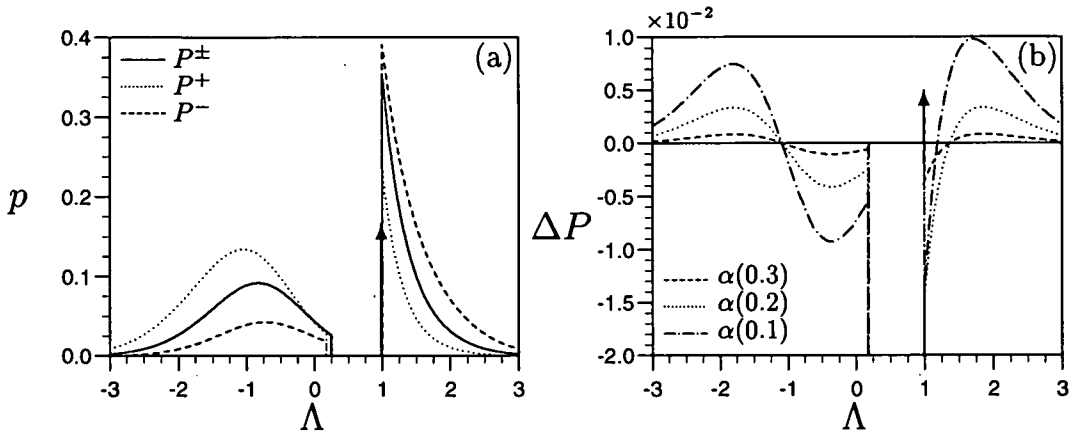

Figure 3.6. (a) The PSDs $P(\Lambda)$ of the spherical perceptron as a function of the pattern stability Λ for $\kappa = 1$ for an example load $\alpha(\theta = 0.5) = 2.0901$ close to the phase transition point $[\alpha_p(\kappa = 1) = 1.8706]$. Again the θ_0 solution predicts the same PSD $P^{\pm}$ for both $\zeta = \pm 1$ patterns, whereas this symmetry is broken for the θ solution. The δ-peaks are indicated by the arrow and their probability masses are given by $P_\delta^{\pm} = 9.5251 \cdot 10^{-2}$, $P_\delta^{+} = 9.1652 \cdot 10^{-2}$, and $P_\delta^{-} = 1.0563 \cdot 10^{-1}$. (b) The difference in the total PSD ($\Delta P \equiv P^{+} + P^{-} - 2P^{\pm}$) as a function of Λ for various values of α: $\alpha(\theta = 0.1) = 1.8790$ $[\Delta P_\delta = 2.3490 \cdot 10^{-4}]$, $\alpha(0.2) = 1.9076$ $[\Delta P_\delta = 1.1915 \cdot 10^{-3}]$, and $\alpha(0.3) = 1.9469$ $[\Delta P_\delta = 2.6226 \cdot 10^{-3}]$. The reduction in the error rate of the θ solution seems to be caused mainly by the increase of the gap.

in Figure 3.7. For the θ_0 solution (which is equivalent to the perceptron without threshold), both PSDs approach half the probability mass of a Gaussian distribution with zero mean and unit variance. This is as expected, since the examples are uniformly distributed spatially and a random weight vector on the hypersphere has an average overlap (activation) with the examples which is Gaussian distributed. As all examples with absolute activation smaller than κ are always counted as erroneous, the error rate approaches $\epsilon = 1 - H(\kappa) \geq 1/2$ in the $\alpha \to \infty$ limit.[11]

For the θ solution on the other hand, both PSDs also approach (half the probability masses of) unit variance Gaussian distributions but with means centred around $\zeta\theta$. Although any weight vector will have a Gaussian-distributed overlap, the activation is shifted due to large threshold. This means that for infinite α, the θ solution classifies the examples deterministically as either all $+1$ or -1 depending on the sign of the (infinite) threshold, resulting in an total error rate of $1/2$ irrespective of the stability κ.

One can assess the convergence rate of the the error rate of the perceptron against the asymptotic error rate ϵ^{∞} from the numerical solutions of the saddlepoint equations.

[11] This means that any random weight vector on the hypersphere has the same error for $\alpha = \infty$. In the case of $\kappa = 0$ this corresponds to random guessing of the output with 50% chance of success.

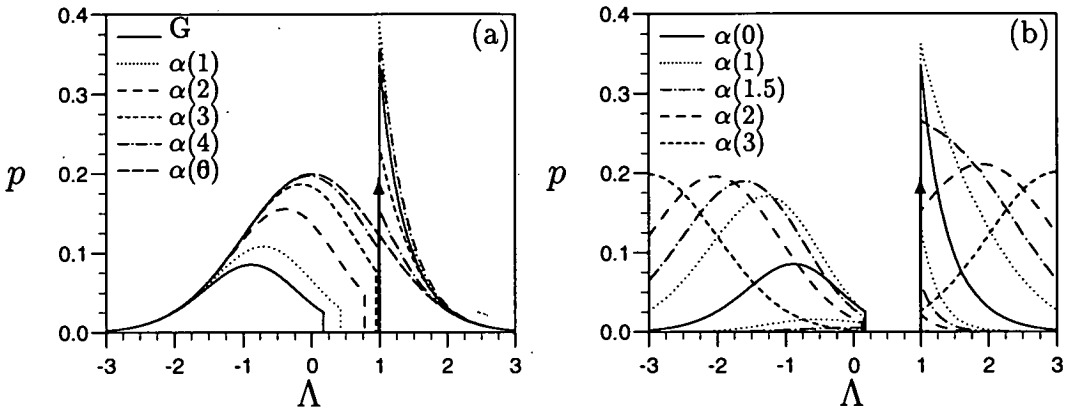

Figure 3.7. The PSDs $P(\Lambda)$ of the spherical perceptron as a function of the pattern stability Λ for $\kappa = 1$ and increasing example load α. (a) The PSD of the θ_0 solution and $\alpha(\theta = 0) = \alpha_p = 1.8706$ $[P_\delta^\pm = 1.1029{\cdot}10^{-1}]$, $\alpha(1) = 2.8878$ $[P_\delta^\pm = 6.1194{\cdot}10^{-2}]$, $\alpha(2) = 10.800$ $[P_\delta^\pm = 1.1017{\cdot}10^{-2}]$, $\alpha(3) = 85.059$ $[P_\delta^\pm = 9.2650{\cdot}10^{-4}]$, $\alpha(4) = 1385.9$ $[P_\delta^\pm = 5.1225{\cdot}10^{-5}]$, and $\alpha(\theta = 6) = 6.2488{\cdot}10^5$ $[P_\delta^\pm = 3.4349{\cdot}10^{-8}]$. The total PSD of both $\zeta = \pm 1$ patterns approaches the zero mean unit variance Gaussian distribution. (b) Both PSDs of the θ solution for a range of α values [see above and $\alpha(\theta = 1.5) = 4.0890$]. The δ-contributions to the $\zeta = \pm 1$ PSDs for $\theta > 0$ are given by (in order of increasing threshold): $[P_\delta^+ = 6.0682{\cdot}10^{-2};\ P_\delta^- = 8.0595{\cdot}10^{-2}]$, $[P_\delta^+ = 3.2087{\cdot}10^{-2};\ P_\delta^- = 4.9147{\cdot}10^{-2}]$, $[P_\delta^+ = 1.3589{\cdot}10^{-2};\ P_\delta^- = 2.4065{\cdot}10^{-2}]$, $[P_\delta^+ = 1.1555{\cdot}10^{-3};\ P_\delta^- = 2.7075{\cdot}10^{-3}]$. Both PSDs approach half of the probability mass of a unit variance Gaussian distribution centred at $\zeta\theta$.

For the θ_0 solution, one finds within the RS ansatz (independent of the weight prior), and within the 1RSB ansatz for spherical and Ising perceptron respectively

$$\epsilon^\infty - \epsilon_{\rm RS} \propto \alpha^{-0.3333\pm1}, \quad \epsilon^\infty - \epsilon_{\rm RSB} \propto \alpha^{-0.490\pm5}, \quad \text{and} \quad \epsilon^\infty - \epsilon_{\rm IRSB} \propto \alpha^{-0.500\pm1},$$

where the error indicates the uncertainty in the last significant digit only. The different exponent in the power law for Ising and spherical perceptron in the 1RSB ansatz is due to a logarithmic correction in the spherical case, as can be confirmed by using the results for the expansions of the saddlepoint equations (3.40) to calculate the asymptotic error of the spherical percepetron in the RS and similarly the 1RSB ansatz

$$\epsilon^\infty - \epsilon_{\rm RS} = \frac{1}{2}\left[\frac{12e^{-\kappa^2}}{\pi\alpha}\right]^{1/3} \quad \text{and} \quad \epsilon^\infty - \epsilon_{\rm RSB} = \frac{e^{-\kappa^2/4}}{\sqrt{\pi}}\frac{[\log\alpha]^{1/4}}{\sqrt{\alpha}}. \tag{3.42}$$

For the θ solution one finds similarly for the total error rate

$$\frac{1}{2} - \epsilon_{\rm RS} \propto \alpha^{-1.0002\pm2}, \quad \frac{1}{2} - \epsilon_{\rm RSB} \propto \alpha^{-1.04\pm4}, \quad \text{and} \quad \frac{1}{2} - \epsilon_{\rm IRSB} \propto \alpha^{-1.04\pm4},$$

for the three cases, respectively. For the θ solution it was again difficult to measure the powers in the 1RSB cases very accurately due to possible logarithmic corrections. This is supported by comparing the numerical predictions to our analytical results for the Ising perceptron, where one finds to leading order

$$\frac{1}{2} - \epsilon_{\mathrm{IRSB}} = \frac{\log(2)}{2\kappa\theta_{\mathrm{s}}} \frac{1}{\alpha}, \tag{3.43}$$

where θ_{s} is the solution for the threshold from Eq. (3.41b) or its approximation (3.41c), which gives a logarithmic correction to the power law with exponent 1.

Comparing the predictions of the power-law decay of the error rate between θ_0 and θ solution, one notes two important differences. First, the exponent of the decay is twice as large for the θ solution, where the error decays linearly with α, and a slower convergence for the θ_0 solution with $\sqrt{\alpha}$. Second, the correction of the θ solution going from RS to 1RSB is only minor, a logarithmic term, whereas it is substantial for the θ_0 solution, a change in the exponent from $1/3$ to $1/2$. This suggests that the effect of RSB for large α is more severe for the perceptron without threshold than with threshold. It also may indicate that the effect of further RSB breaking should be less pronounced for the θ solution than for the θ_0 solution.

3.4.4 Non-zero output bias m_{o}

For non-zero output bias m_{o}, the symmetry in the space of thresholds θ is broken and only solutions with $\theta \neq 0$ are found for all α, characterized by $\theta > 0$ for $m_{\mathrm{o}} < 0$ and vice verse. Due to the symmetry of the solutions for $m_{\mathrm{o}} \to -m_{\mathrm{o}} \Rightarrow \theta \to -\theta$, one can assume $m_{\mathrm{o}} < 0$ and $\theta > 0$ w.l.o.g.. Below, only the Ising perceptron will be discussed as the behaviour for both binary and real weights is quite generic.

In Figure 3.8 the threshold of the Ising perceptron is shown as a function of α for various values of the output bias m_{o} at fixed stability $\kappa = 0.1$. In Figure 3.8(a) one sees that for very small magnitude of the bias, the evolution of the threshold closely approaches the curve for zero bias. Similar behaviour can also be found for the other order parameters. The largest deviations between the zero-bias solution and the finite-bias solution can always be found around the point of the phase transition at α_{p}. In this sense, the output bias m_{o} can be seen as an external field which smoothes out the phase transition.

In Figure 3.8(b) the evolution of the threshold θ is shown for larger magnitudes of the bias over a wider range of loads α. For large α the threshold tends to infinity, whereas the left-hand starting point of each curve depicts the capacity limit α_{c} increasing with increasing magnitude of the bias.

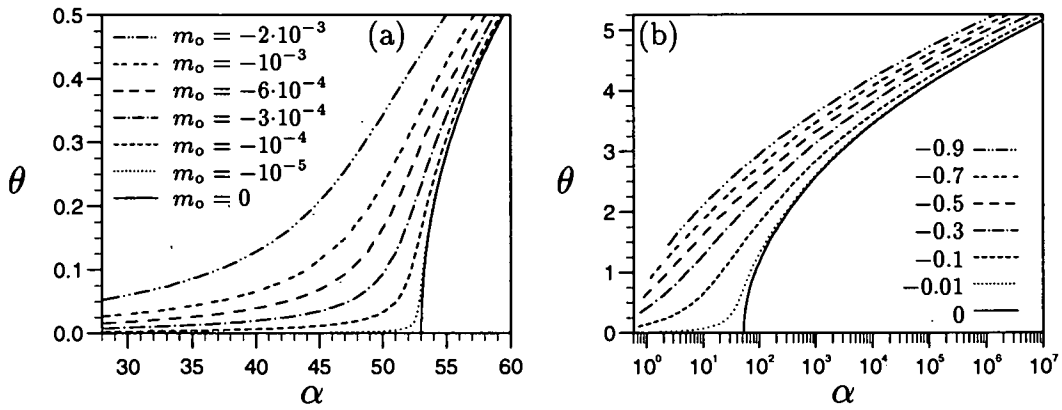

Figure 3.8. (a) The evolution of the threshold θ with the example load α is shown for several small values of the bias (see the legend) around the critical load α_p with constant stability $\kappa = 0.1$. The phase transition is increasingly smoothed out for growing magnitude of the bias. (b) The evolution of $\theta(\alpha)$ over a wide range of α for larger magnitudes of the bias m_o shows the same effect. The left-hand starting point of each curve depicts the capacity limit α_c increasing with growing magnitude of the bias.

For large α, one can expand the free energy of the Ising perceptron, similarly to the zero-bias case. One finds that the leading order of $\partial f/\partial y$ gives $q_0 = y^2$ as for the zero-bias case. The leading order of $\partial f/\partial\theta$ implies

$$v = 2|\theta_s|\left[\kappa + \frac{1}{2|\theta_s|}\log\left(\frac{1+|m_o|}{1-|m_o|}\right)\right] = 2|\theta_s|\kappa^*, \tag{3.44a}$$

where θ_s is the solution of the threshold for given load α and κ^* is a modified effective stability, which depends on the bias and on the solution of the threshold (i.e., ultimately on α). Further inserting $\partial f/\partial v$ in $\partial f/\partial q_0$ yields

$$\sqrt{q_0} = y = \frac{\log(2)}{\sqrt{1-m_o^2\kappa^*}}\frac{1}{\sqrt{\alpha}}. \tag{3.44b}$$

The remaining saddlepoint equation $\partial f/\partial v$ to determine θ_s is given by

$$(1+|m_o|)\exp\left[-\frac{1}{2}(|\theta|-\kappa)^2\right] - (1-|m_o|)\exp\left[-\frac{1}{2}(|\theta|+\kappa)^2\right] = \frac{\sqrt{2\pi}\log(2)}{\kappa^*\alpha}. \tag{3.44c}$$

and cannot be solved for θ in closed form. The approximation used in the zero-bias case in Eq. (3.41) [see Section 3.4.2], which neglects the less dominant term on the left-hand side of Eq. (3.44c), still does not make a closed-form solution feasible, due to

the θ-dependence of κ^*.

For the asymptotic error rate one finds $\epsilon^\infty = \frac{1}{2}(1 - |m_0|)$ irrespective of the stability κ — the intuitive result if one classifies the larger class of example correctly and misclassifies the smaller example class by using a threshold of infinite absolute value. The asymptotic error rate is approached via

$$\epsilon^\infty - \epsilon_{\text{IRSB}} = \frac{\log(2)}{2|\theta_s|\kappa^*}\frac{1}{\alpha}. \tag{3.45}$$

As both θ_s and κ^* are dependent on α, the asymptotic behaviour deviates from a pure power-law behaviour.

3.4.5 The stability dependence of the phase transition

In this section the dependence of the phase-transition point in order-parameter solution space on the threshold stability κ is examined. In Figure 3.9 α_p is plotted versus κ on a log–log scale for both spherical and Ising perceptron in the RS and 1RSB ansätze. The critical point α_p in solution space increases for decreasing stability but exists for all non-zero stabilities, and exhibits a power-law dependence on κ for small stabilities with $\alpha_p \to \infty$ as $\kappa \to 0$. The numerical data predicts the exponents of the power laws as

$$\alpha_p^{\text{RS}} \propto \kappa^{-3.000\pm1}, \quad \alpha_p^{\text{RSB}} \propto \kappa^{-2.04\pm2}, \quad \text{and} \quad \alpha_p^{\text{IRSB}} \propto \kappa^{-2.0000\pm1},$$

where the RS theory of the Ising perceptron only rescales the prefactor with the constant $2/\pi$.

From Figure 3.9 one can further conclude that the phase transition exists for all finite stabilities $\kappa > 0$. The limits $\kappa \to 0$ and $\alpha \to \infty$ are therefore *not* interchangeable, which leads to some interesting effects. Although, $\kappa = 0$ has an error rate of $1/2$ at $\alpha = \infty$ irrespective of the threshold, only the θ_0 solution is accessible to the perceptron for any finite α and it has no access to the θ solution for $\alpha \to \infty$. Similarly, allowing for a finite output-distribution bias m_0 (see Section 3.4.4), i.e., a non-zero external field, leads to thresholds with infinite magnitude but opposite signs for $m_0 \to \pm 0$ at $\alpha = \infty$, since for any finite m_0 it is advantageous to classify the larger target class correctly deterministically. This behaviour could be interpreted as a first-order phase transition, as one finds a discontinuous jump in θ, and the "point" $\{\kappa = 0, \alpha = \infty\}$ could be seen as a *tricritical* point.

These phenomena can partly be understood in an analogy with the ferromagnet, where the critical temperature T_c is proportional to the coupling strength J, broadly analogous to the decrease of $1/\log(\alpha_p)$ with κ. Therefore, the $\kappa \to 0$ and $\alpha \to \infty$ limit

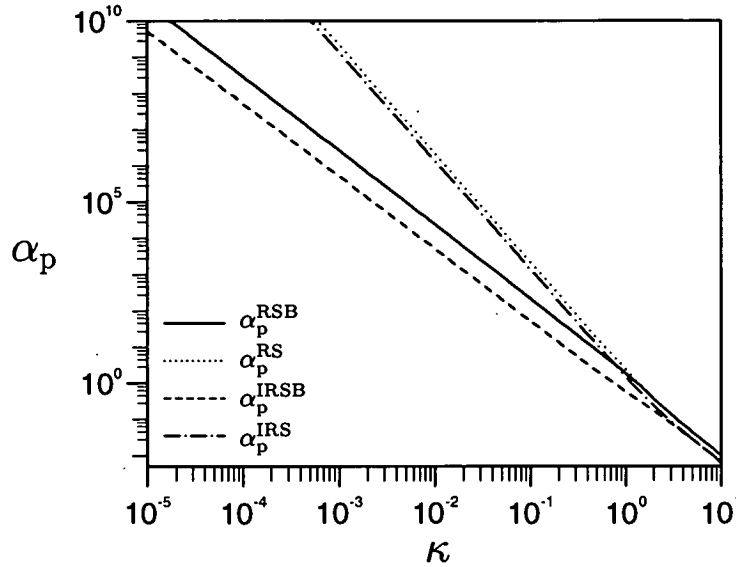

Figure 3.9. The critical normalized example number α_p as a function the stability κ on a log–log scale shows a power-law behaviour for small stability. The predicted power-law behaviour using 1RSB is significantly different to the one predicted from RS. This phase diagram separates points with zero and finite threshold below and above the line of second-order phase transitions at α_p.

corresponds to $J \to 0$ and $T \to 0$. In the ferromagnet, it is obvious that these two limits do not commute; the resulting magnetization depends only on the ratio T/J. Here, the behaviour is quantitatively much more complicated but qualitatively similar. The $m_o = 0$ phase diagram (with parameters κ and α) shows the phase boundary between zero and non-zero threshold which ends in the point $\{\kappa = 0; \alpha = \infty\}$; depending on how the limit to this point is taken, one can move along either side of this boundary, which subsequently results in different limits for the threshold.

As the phase transition seems to be triggered by the increase of the error rate above a critical value, the error rate $\epsilon_p = \epsilon(\alpha_p)$ is also shown at the critical load, together with its deviation from the asymptotic error rate 1/2 in Figure 3.10. One can see that the stability has a dominant influence on the occurrence of the phase transition through the error rate. For large stabilities the θ_0 solution becomes already unstable for small error rates, with the limit $\epsilon_p \to 0$ for $\kappa \to \infty$. The difference in the critical error rate between the Ising and spherical perceptron is greatest for moderate stabilities $\kappa \approx 1$, which may be attributed to the gap and the δ-contribution in the PSD of the spherical perceptron.

The RS theory does not only underestimate the error for a given load α, and,

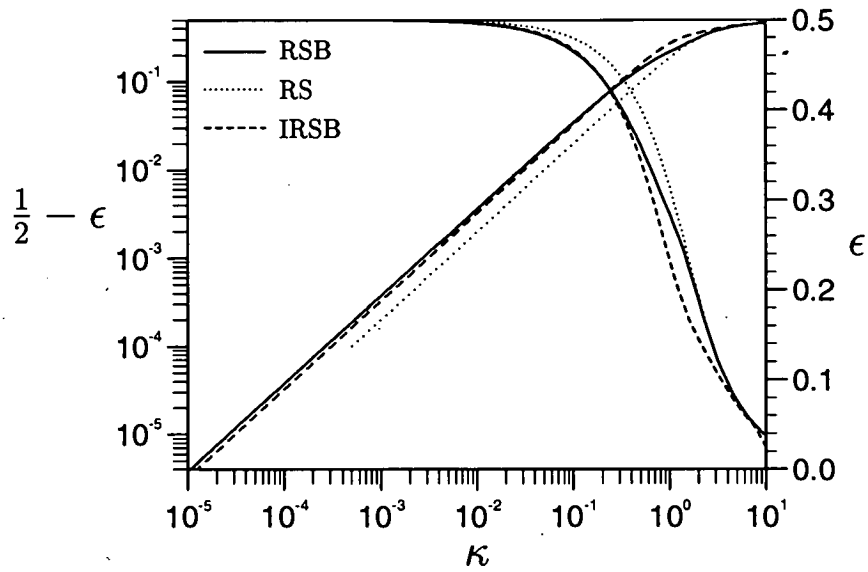

Figure 3.10. The error rate $\epsilon(\alpha_p)$ and its deviation from the asymptotic error rate $1/2$ is shown as a function of the stability κ on log–lin and log–log scales respectively. The remnant error rate $1/2 - \epsilon$ shows a power-law decay for small κ. For larger stabilities, the phase transition occurs for increasingly small error rates.

therefore, gives the incorrect power law for α_p, but also fails to predict the correct critical error rate. RS fails especially for smaller stabilities, i.e., large α, as expected. This is especially obvious by looking at the remnant error rate in Figure 3.10 which decays with a power law. The exponents can be also evaluated from the numerical data:

$$\frac{1}{2} - \epsilon_p^{\mathrm{RS}} \propto \kappa^{1.000\pm1}, \quad \frac{1}{2} - \epsilon_p^{\mathrm{RSB}} \propto \kappa^{0.993\pm2}, \quad \text{and} \quad \frac{1}{2} - \epsilon_p^{\mathrm{IRSB}} \propto \kappa^{1.00000\pm1}.$$

Although RS seems to give a reasonable power-law decay of the error, the prefactor is blatantly incorrect. An asymptotic expansion for small thresholds and stabilities for the RS theory gives

$$\alpha_p^{\mathrm{RS}} = \frac{8}{9}\frac{\sqrt{2\pi}}{\kappa^3} \qquad \text{and} \qquad \frac{1}{2} - \epsilon_p^{\mathrm{RS}} = \frac{\kappa}{2\sqrt{2\pi}} \tag{3.46a}$$

Of more interest is the functional behaviour of the 1RSB solution for small stabilities, as the numerical solutions indicate a deviation from the pure power-law behaviour in both the point of the phase transition as well as the asymptotic error. A similar analytic

expansion gives

$$\frac{\alpha_{\rm p}^{\rm RSB}}{\sqrt{\log \alpha_{\rm p}^{\rm RSB}}} = \frac{1}{2\kappa^2} \tag{3.46b}$$

for which a closed form solution does not exist. However, one can see that the deviation from the pure κ^{-2} power-law behaviour of $\alpha_{\rm p}$ is due to the additional logarithmic term in $\alpha_{\rm p}$.

For the Ising perceptron it is not possible to expand all of the equations as the order parameters y and q_0 have finite limits. However, the numerical solutions themselves give us some insight. For the Ising perceptron there is no numerical indication that the critical load $\alpha_{\rm p}$, or its error $\epsilon_{\rm p}$, deviate from pure power-law behaviours in contrast to the spherical case, which exhibit logarithmic corrections. Furthermore, for large stabilities the phase transition occurs at a smaller error rate for the Ising than for the spherical perceptron, whereas this characteristic is reversed for small stabilities, where the phase transition occurs at a larger error rate. These differences between the two weight priors could either be attributed to their respective weight-space structures, or it may indicate that 1RSB is correct in the Ising and incorrect in the spherical case.

3.5 Summary and conclusions

In this chapter the threshold Boolean perceptron above saturation has been investigated for both spherical and binary weight priors. Even for unbiased input and output distributions, one finds that the introduction of a threshold triggers interesting phenomena for finite stabilities $\kappa > 0$ which are not otherwise present. Namely, a second-order phase transition in order-parameter space is found at a stability-dependent critical load $\alpha_{\rm p}(\kappa)$, with spontaneous symmetry breaking in the space of thresholds θ. This phase transition is driven by the error rate as the perceptron without threshold exhibits a higher asymptotic error $[\epsilon^{\infty} = 1 - H(\kappa)]$ than the perceptron with threshold $[\epsilon^{\infty} = 1/2]$.

One can furthermore identify the bias of the output distribution $m_{\rm o}$ with the external magnetic field in spin systems that breaks the symmetry in θ space and smoothes out the phase transition. Whereas a non-zero output bias has, therefore, a profound effect on the performance of perceptrons, we find that a non-zero input bias can always be absorbed by a rescaling of the target stability κ. These results also suggest that one should not remove the threshold in favour of a ferromagnetic bias in the couplings as we have found that a threshold can always compensate for this bias but not vice versa.

Zero stability, $\kappa = 0$, constitutes a special case, as one does not find a phase transition for finite α and the limits $\kappa \to 0$ and $\alpha \to \infty$ are *not* interchangeable. One could argue that the "point" $\{\kappa = 0, \alpha = \infty\}$ is in fact a first-order phase transition, as one finds discontinuous jumps in order-parameter space when taking the $m_0 \to \pm 0$ limit from either side.

In the asymptotic limit $\alpha \to \infty$ and finite stability $\kappa > 0$, we not only find unequal values for the asymptotic error rate but strikingly different solutions in order-parameter space for the perceptron with and without threshold, especially, for the asymptotics of the 1RSB overlap q_0. In the case of the spherical weight constraint, we find that q_0 approaches 1 for the perceptron without threshold, whereas q_0 decays to 0 for the perceptron with threshold. For the Ising perceptron we find a similar behaviour: the solution with non-zero threshold is characterized by a vanishing overlap q_0 for increasing α and the solution with zero threshold exhibits a finite limit of q_0 for infinite load which is stability dependent and strictly smaller than 1.

It has been argued previously (Majer et al. 1993) that the above asymptotic behaviour for the spherical perceptron without threshold indicates that 1RSB cannot be exact at high load. For a correct solution one would expect the smallest overlap scale q_0 to approach 0 for $\alpha \to \infty$ as in this limit any weight vector should perform equally well. Recently, it has been shown by performing a two-step RSB calculation (Whyte and Sherrington 1996) that 1RSB is indeed inexact for the spherical perceptron without threshold. Furthermore, it has been proved (Whyte and Sherrington 1996) that any model with a gap in the PSD (such as the spherical perceptron with or without threshold and Gardner–Derrida cost function) necessitates infinitely many RSB steps to yield the exact result. These findings give some support to the validity of the qualitative argument made above. A strict application of this argument would imply that 1RSB is also inexact for the Ising constraint, which has been the source of some debate (Krauth and Mézard 1989; Fontanari and Meir 1993; Horner 1992b; Weigt and Engel 1997). As the PSD of the Ising perceptron with the Gardner–Derrida cost function does not exhibit a gap, the proof in (Whyte and Sherrington 1996) is not able to resolve this issue.

We have some doubts if one can have enough confidence in the qualitative argument of (Majer et al. 1993) to argue that 1RSB is incorrect in the Ising model. First, we believe that one should be very careful to apply such an intuitive argument to models with discrete weights. For example, whereas all overlaps in the spherical model converge to 1 at the capacity limit, leaving just a single solution, the smallest overlap scale q_0 remains finite but strictly smaller than 1 for the Ising model, which is initially not really intuitive [see (Krauth and Mézard 1989) for a plausible explanation], as it suggests

several solutions at the capacity limit. A similar effect may be present in the limit $\alpha \to \infty$. Second, one may argue, that the argument of (Majer et al. 1993) can demand $q_0 = 0$ strictly only at $\alpha = \infty$, whereas it implicitly assumes a smooth transition of $q_0 \to 0$ for $\alpha \to \infty$, which does not take into account the possibility of a discontinuous transition. We have arguably found a possibility for such a discontinuous transition for the case $\kappa = 0$ at $\alpha = \infty$, from the θ_0 solution with $q_0 = 1$ to the θ solution with $q_0 = 0$. To resolve the issue of the exactness of 1RSB in the Ising perceptron with Gardner–Derrida cost function, it may be worthwhile to re-examine the two-step RSB solution in (Krauth and Mézard 1989) numerically for large α and/or to calculate the stability of the 1RSB solution.

Nevertheless, results concerning the asymptotic behaviour of the error rate and the order parameters presented here suggest that the effect of further RSB breaking may be even smaller for both the Ising and the spherical perceptron with threshold in the regime of the θ solution than has been found for the θ_0 solution of the spherical perceptron in (Whyte and Sherrington 1996). The 1RSB solution may therefore remain sufficiently accurate for many practical purposes like calculating the capacity of multilayer networks produced by constructive algorithms, which will be carried out in the following Chapter 4, where a treatment with a two-step RSB solution is computationally infeasible.

Chapter 4

The Statistical Mechanics of Constructive Algorithms

Abstract

After investigating the perceptron above saturation in a replica framework in the previous chapter, these results are applied to investigate the storage capacity of multilayer networks with overlapping receptive fields for constructive algorithms using Boolean perceptrons as their basic building block. The assumption of weak coupling between subsequently constructed perceptrons is verified within a replica symmetric (RS) ansatz and shown to be negligible in most cases in comparison to correction due to replica symmetry breaking (RSB) in individual perceptrons. The capacities of a tiling-like and variants of the upstart algorithm are then calculated within RS and one-step RSB with the quenched average taken over the individual units separately for networks with up to $K = 4000$ and $K = 600$ units respectively. Within this treatment, the storage capacity α_c^K seems to exhibit a power-law behaviour in $\log K$ with an exponent n that may depend on the algorithm and the stability. However, due to finite size effects in K reliable estimates of n could not be extracted. Nevertheless, the results strongly indicate that n should be strictly smaller than 1 within 1RSB, whereas within RS the Mitchison–Durbin bound is violated for finite K and $n > 1$ may hold asymptotically.

4.1 Introduction

In the last chapter, we have studied the simplest neural network, the Boolean perceptron, above its saturation limit within a replica framework following the groundbreaking papers (Gardner 1988; Gardner and Derrida 1988). Many papers using the

replica techniques and the framework Gardner had developed followed, investigating many aspects of the performance of simple neural network models. Whereas initially research focussed on the extensions of the storage capacity problem by either calculating properties above saturation as has been performed in the previous chapter or the size of the basin of attraction, the attention has shifted recently mainly towards the understanding of the supervised learning problem within the student-teacher scenario, which calculates the generalization capability of a neural network model with the number of training examples available. Such problems will be studied later in Chapters 5 and 6.

The capacity problem does, however, remain relevant due to its relation to the Vapnik–Chervonenkis (VC) dimension (Blumer et al. 1989) of computational learning theory and the PAC framework. As explained in Chapter 3, the difference broadly speaking being that statistical mechanics analyses the average case, whereas the PAC framework analyses the worst case. Within the PAC framework, the VC–dimension enables one to determine an upper bound on the examples needed to achieve a certain generalization error (Vapnik and Chervonenkis 1971; Baum and Haussler 1989), broadly reflecting the view that generalization can only begin once the storage capability has been exceeded (Opper 1994). The capacity of a network model therefore influences both its flexibility of implementing complicated mappings and its generalization ability for a training set of given size.

Substantial work has therefore been carried out in both communities in order to calculate these storage quantities, however, the problem has proved to be very hard for multilayer perceptrons (MLPs) and most success has been reserved to the estimation of upper and lower bounds (Baum and Haussler 1989; Mitchison and Durbin 1989; Bartlett 1993; Wendemuth 1995d; Maass 1994; Sakurai 1995) of two-layer networks, resulting in the well known lower and upper bounds of the storage capacity (per adjustable weight) of 1 and $\log_2 K$ [Mitchison–Durbin (MD) bound (Mitchison and Durbin 1989)] respectively, where K is the number of units in the hidden layer. Attempts for the direct calculation of the capacity limit of MLPs have been hampered by the inherent difficulties of the replica calculation needed to perform the quenched average of the training set[1]. In the capacity calculations of the parity[2] (Barkai et al. 1990) and committee[3] (Barkai et al. 1992; Engel et al. 1992) machines, replica symmetric (RS) treatments violate the MD bound derived by information theory or counting

[1]Such difficulties can be avoided in the generalization problem by studying on-line learning (Saad and Solla 1995b), which will be considered in Chapters 5 and 6.

[2]The output of a parity machine is the product of all hidden unit outputs.

[3]The output of a committee machine is the majority of all hidden unit outputs.

arguments similar to (Cover 1965), whereas a replica symmetry breaking (RSB) calculation (Barkai et al. 1990) saturate the bound in the $K \to \infty$ limit for the tree parity machine[4]. Other efforts (Saad 1994) break the symmetry of the hidden units explicitly prior to the actual calculation, but the resulting equations are approximations and difficult to solve for large networks. Recently, the introduction of a new technique (Monasson and Zecchina 1995), focusing on the number of implementable internal representations instead of on the Gardner volume, has been used to calculate the capacity of the tree (Monasson and Zecchina 1996) and fully-connected committee machine (Urbanczik 1997; Xiong et al. 1997). In the $K \to \infty$ limit, the committee machine does not saturate the MD bound but still diverges with $\sqrt{\log K}$.

What follows in this chapter avoids these problems by addressing the capacity of a class of networks with variable architecture produced by constructive algorithms. In this case, the basic building blocks are simple Boolean perceptron, which are trained individually and the results derived in Chapter 3 can be applied iteratively to yield the storage capacity of two-layer networks.

A multitude of constructive algorithms have been proposed over the years, e.g., (Gallant 1990; Ash 1989; Mézard and Nadal 1989; Nadal 1989; Fahlman and Lebiere 1990; Frean 1990a; Campbell and Perez Vicente 1995). They are all loosely based on the idea that in general it is *a priori* unknown how large a network must be to perform a certain regression or classification task. It seems therefore appealing to start off with a simple network, e.g., a Boolean or sigmoidal perceptron, and to increase its complexity by adding further units only when needed, thereby eliminating the cumbersome search for the right network size.

However, the constructive algorithms proposed differ in several aspects. Some of them are applicable to regression (Ash 1989; Fahlman and Lebiere 1990) others to classification (Mézard and Nadal 1989; Nadal 1989; Frean 1990a; Campbell and Perez Vicente 1995) tasks, which is often reflected in the type of units they use [Boolean, sigmoid, RBF (Radial Basis Function), or HON (Higher Order Network)]. They also produce several typical architectures, e.g., hierarchical tree type (Frean 1990a), list type (Campbell and Perez Vicente 1995), cascade type (Nadal 1989; Fahlman and Lebiere 1990), self-organising cell-type (Fritzke 1994), multilayer with either fixed (Frean 1990a; Marchand et al. 1990; Martinez and Estève 1992) or problem driven deepness (Mézard and Nadal 1989). Some algorithms also have several versions, which usually result in different architectures.

[4]In a tree archcitecture, the receptive fields of the hidden units are non-overlapping, i.e., share no common inputs, wheras in fully connected models the hidden units share all inputs.

Another important difference lies in the training procedure performed once a new unit has been added. Some algorithms require only the training of the newly added unit fixing the weights of previously constructed units, while others require some sort of retraining of connections involving output weights and/or unit weights. The great advantage of the former is that the training time of the whole network is usually relatively short, since training involves only small units, typically single-layer networks, for which fast training algorithms are available even for Boolean units (Rosenblatt 1962; Minsky and Papert 1969; Diederich and Opper 1987; Krauth and Mézard 1987; Anlauf and Biehl 1989; Frean 1992; Wendemuth 1995a)[5]. These constructive algorithms can therefore avoid the difficulty of learning internal representation of units, e.g., by back-propagation for sigmoid units (Werbos 1974; Rumelhart et al. 1986a) or by the CHIR (Learning by Choice of Internal Representations) algorithm for Boolean units (Grossman et al. 1989), unlike other proposed constructive algorithms (Fritzke 1994; Ash 1989) and general MLPs with an *a priori* fixed architecture. This training process is especially difficult for Boolean units, where powerful second-order gradient-based techniques (Bishop 1995) are not available.

A further advantage of some constructive algorithms, which train only single layer units, is the existence of convergence proofs, i.e., one can show that training will converge in finite steps, unlike conventional networks which can be trapped in bad local minima and often have to be restarted many times before an acceptable solution is found. For some algorithms this convergence is to zero training error, a feature which leads to undesirable over-fitting and subsequent poor generalization for noisy data. However, this problem can be addressed by including some kind of penalty term on the creation of new units to the training error and/or by training with negative stability allowing for errors close to the decision boundary. That constructive algorithm can in principle be very good generalizers has been shown in (Schapire 1990), where it has been proven within the PAC framework, that any *weak learner*, a machine which only achieves a generalization error just below random guessing, can be used to constructively build a *strong learner*, a machine which achieves any arbitrary small generalization error. This *boosting* algorithm and its improved variants (Freund 1995; Freund and Schapire 1995) have shown very promising results in real world applications (Drucker et al. 1994), along other constructive algorithms which have been tested on noisy problems (Littmann and Ritter 1996).

Other approaches, which aim at automating the choice of appropriate network size,

[5]Some of these algorithms have to be stabilized for non-linear separable problems by the *pocket* algorithm (Gallant 1990).

are based on starting with large networks and then attempt to optimize performance by identifying and removing unnecessary individual weights and/or units according to some predefined rules, e.g., (Mozer and Smolensky 1989; Chauvin 1989; Le Cun et al. 1990; Hassibi and Stork 1993; Levin et al. 1994). These procedures usually require computationally expensive calculation and further retraining of the pruned network. A conceptionally different but effectively similar approach is to add penalty terms (Weigend et al. 1990; Nowlan and Hinton 1992; Setiono 1997) to the energy function to be minimized, often also termed weight decay, or regularization, which practically eliminate weights which do not significantly contribute to the reduction of the training error. Within a Bayesian framework (Neal 1996), it is also not necessary to restrict the number of units *a priori*; however, Bayesian methods can be prohibitively expensive in many situations.

Overall, constructive algorithm seem therefore rather appealing, but the abilities of different algorithms have neither been compared heuristically in a systematic way on real world problems nor has any attempt been made to understand their properties within a theoretical framework. The aim of this chapter is, therefore, to introduce a framework in which one aspect of the performance, the learning of random dichotomies or capacity problem, of a class of constructive algorithm can by analysed and compared objectively. This should give us some indication of how effective different constructive algorithms use their weights in comparison to each other and to the upper bounds known for unconstrained MLPs.

The class of constructive algorithm susceptible to this framework consists of algorithms which use only Boolean perceptrons as the basic building block and where later generations of units receive no input from previously constructed units, unlike e.g., cascade networks such as (Mézard and Nadal 1989; Fahlman and Lebiere 1990). This is due to the fact that our treatment relies on iteratively using results derived for individual Boolean perceptrons above their saturation limit derived in Chapter 3. We therefore rely on the approximation that the quenched average over the training set can be taken separately for each individual perceptron, i.e., correlations between the output of previous hidden units need not be taken into account and the correlations between the errors of the units are small.

Here, we will investigate in particular variants of the upstart algorithm (Frean 1990a) and a tiling-like algorithm (Biehl and Opper 1991), although our calculations can easily be extended to many other algorithms, such as (Marchand and Golea 1993; Zollner et al. 1992; Campbell and Perez Vicente 1995). For the algorithms studied here, the corrections to the decoupled approximation have been calculated for two consecutive units within an RS ansatz and turn out to be small in most regimes in

comparison to correction due to RSB in each individual perceptrons, rendering errors due to the decoupling assumption small.

For these algorithms, we calculate the capacity within the replica symmetric (RS) and one-step replica symmetry breaking (1RSB) ansatz for networks with up to $K = 4000$ (RS) and $K = 600$ (1RSB) units. Within this treatment, the numerical results strongly indicate that the storage capacity α_c^K exhibits a power-law behaviour in $\log K$ with an exponent n, which may be stability and algorithm dependent. The exponent has been measured locally showing slight systematic shifts with the number of units K, so that reliable upper bounds or estimates of n for $K \to \infty$ could not be extracted. Therefore, recent asymptotic capacity results may be interesting theoretically, however, finite K effects may render them irrelevant for practical considerations. For all constructive algorithms studied, the finite K results further indicate that n is strictly smaller than 1 when accounting for 1RSB. Within the simpler RS treatment, the Mitchison–Durbin bound is violated for large finite K and $n > 1$ may hold asymptotically.

The chapter is structured as follows. In Section 4.2 the capacity problem is introduced and the investigated constructive algorithms described. In Section 4.3, an introduction to the replica framework used for the capacity calculation is given and the mechanism for employing results derived for simple perceptrons to obtain results for the capacity limit of constructive algorithms is explained. This will be complemented by a brief presentation of results for a single and two coupled perceptrons, which give insight into the numerical results of the iterative calculation of the capacity limit of networks built by constructive algorithms. In Section 4.4 the numerical capacity data is presented and analysed by calculating the local power-law exponent $n(K)$. The chapter finishes with a discussion and some concluding remarks in Section 4.6.

4.2 Constructive algorithms

As explained in detail in Section 3.2, in the capacity problem a network aims at implementing a pattern set consisting of $p = \alpha N$ input-output pairs $\{(\boldsymbol{\xi}^\mu, \zeta^\mu)\}$, where both inputs and outputs are taken from random distributions (3.1). It has been known for many years (Minsky and Papert 1969), that the mapping of the Boolean perceptron (3.2) is a linear decision boundary. Therefore, if the set of examples is not linearly separable with a minimum distance κ of all patterns to the hyperplane, a perceptron will not be able to classify all patterns without errors.

In this case, a learning machine, such as a MLP, that is able to learn non-linear decision boundaries needs to be trained on the example set in order to achieve perfect

storage. An alternative approach is taken by many constructive algorithms, which add new perceptrons in such a way that the combination of all their linear decision boundaries leads to a non-linear decision boundary that can perform the required task. The constructive algorithms vary significantly in the way these extra decision boundaries are trained and how they are combined to yield the overall output. Below, this will be explained for the constructive algorithms analysed in detail in this chapter, a tiling-like algorithm (Biehl and Opper 1991; Biehl and Opper 1993), which has previously been analysed numerically within an RS ansatz, and the upstart algorithm (Frean 1990a), whose original ideas are introduced and compared with the versions, termed upstart II-III, which have been used in the capacity calculations.

4.2.1 The tiling-like algorithm

The basic idea of the tiling-like algorithm (Biehl and Opper 1991) and of most other constructive algorithms such as (Mézard and Nadal 1989; Nadal 1989; Zollner et al. 1992) is to constructively build a faithful internal representation in the hidden layer, i.e., all patterns with the same internal representation share the same target output. The remaining problem is then to devise a way to map the internal representation to the desired output, which can be solved in many different ways. The tiling-like algorithm achieves this by constructively building a very specific set of internal representations, which is automatically mapped to the desired output by a hardwired parity function as a fixed hidden-output mapping, where the output is just the product of the individual outputs of all constructed units, leading to a chequered partition of the input-space.

The faithful representation in the hidden layer is achieved in the following way. The first perceptron, $\mathcal{U}_1$, is trained on the original Boolean targets $\zeta^\mu \in \{-1, 1\}$. If this unit makes any errors, ϵ_1, on the training set, a second unit, $\mathcal{U}_2$, is created which is trained on the complete training set but with modified targets exploiting the property of the parity function: whereas the output of the whole network would remain unchanged for an output of $+1$ by $\mathcal{U}_2$, it would be reversed for -1. Hence, the targets of $\mathcal{U}_2$ are $+1$ for previously correctly classified and -1 for misclassified patterns. This procedure is iterated until the current unit $\mathcal{U}_i$ classifies all patterns correctly (according to its targets). It can be shown that this algorithm will eventually converge as $\mathcal{U}_i$ corrects at least one previously incorrectly classified pattern without upsetting any correctly classified ones. Note, that it is sufficient to train each perceptron with stability κ to ensure that all examples are finally implemented with the desired stability, a property which also holds for most other constructive algorithms including the upstart algorithm.

4.2.2 The upstart algorithm

Although the basic idea of building a faithful internal representation by adding new units holds also for the upstart algorithm (Frean 1990a), the technical details of the algorithm are somewhat different to many other constructive algorithms. First of all, it uses a asymmetric $\zeta^\mu \in \{0,1\}$ instead of the usual symmetric (Ising) representation $\zeta^\mu \in \{-1,1\}$ for the outputs. Similar to other algorithms, one starts with a single perceptron, the *mother* unit $\mathcal{M}$, and further units are created only if erroneous patterns exist. However, in this algorithm potentially two *daughter* units, $\mathcal{U}^+$ and $\mathcal{U}^-$, are created to specifically correct one of the two possible type of errors: *wrongly-off* errors, where the target was 1 but the output is 0, and *wrongly-on* errors, where the target was 0 but the actual output is 1. $\mathcal{U}^+$ and $\mathcal{U}^-$ are connected to their mother unit $\mathcal{M}$ by a large enough positive or negative weight, respectively, so that they overrule any decision by $\mathcal{M}$ when they are active ($\sigma = 1$).

Consider, for example, the new training set and targets that would be assigned to $\mathcal{U}^-$, which will be connected with a large negative weight to $\mathcal{M}$, i.e., whose rôle will be to inhibit $\mathcal{M}$. $\mathcal{U}^-$ should be active ($\sigma = 1$) for patterns where $\mathcal{M}$ is currently wrongly-on and inactive ($\sigma = 0$) for patterns where $\mathcal{M}$ is correctly-on. However, $\mathcal{U}^-$ does not have to be trained on patterns for which $\mathcal{M}$ is correctly-off, since an active $\mathcal{U}^-$ would only reinforce $\mathcal{M}$'s already correct response. The remaining patterns, for which $\mathcal{M}$ is wrongly-off, need special consideration. They have to be included in $\mathcal{U}^-$'s training set with target 0, in order to avoid inhibiting the pattern further which would lead frustration when combined with the output of $\mathcal{U}^+$, which is trying to correct the wrongly-off patterns. Similar arguments can be applied to $\mathcal{U}^+$, and the resulting targets and training sets for both unit types are summarized in Table 4.1.

If $\mathcal{U}^+$ and $\mathcal{U}^-$ can correct all erroneous patterns, the algorithm has achieved its objective and terminates. Otherwise, various possibilities exist for its continuation, of which several have already been reported in the original works (Frean 1990a; Frean 1990b). In the original algorithm (termed here upstart I), those daughter units with non-zero training errors in turn become the mothers of the daughters of the next generation, leading to a hierarchical network architecture as shown in Figure 4.1(a). Consequently, this allows for a parallelization of the local training procedure, but also tends to lead to a exponentional increase of the number of hidden units (and hence unit specific training sets) with each generation, which may potentially make extremely wasteful use of hidden units. It has already been pointed out in the original publication (Frean 1990a), that this hierarchical tree can be squashed into an equivalent more conventional two-layer architecture, where all units (including the original unit $\mathcal{M}$, which

can then be seen as $\mathcal{U}_1^+$) are connected to a master output unit $\mathcal{M}$, with positive or negative weights whose magnitude increases with each generation in order to guarantee that erroneous decisions are actually corrected by the following generation as shown in Figure 4.1(b). These two-layer networks show only a linear increase of the number of units with each generation, which may use each unit more efficiently and which is also easier to analyse for the purposes of this chapter.

In this case, one has to decide what criteria to use for the creation of new units. One could obviously create both types of units, $\mathcal{U}_i^+$ and $\mathcal{U}_i^-$, simultaneously with each generation as before if both types of errors are made, but this may be wasteful, if, for example, there are much more wrongly-off than wrongly-on errors.

A more efficient variant, here termed upstart II, is therefore to create both type of units if the probability of both error types is identical and only one unit correcting the error of higher frequency otherwise. Two obvious choices exist for evaluating these probabilities. One could either use the marginal probabilities of wrongly-on and wrongly-off errors or one could condition the probabilities on the original target, i.e., compare the probability of a wrongly-on error given a pattern with original target $\zeta = 0$ with that of a wrongly-off error given a pattern with original target $\zeta = 1$. We will denote the former as criterion (a) (i.e., upstart IIa) and the latter as criterion (b) (i.e., upstart IIb). Obviously, these two variants are identical for the case when the initial output distribution is unbiased, i.e., the number of targets for each class is identical.

Table 4.1. The targets of the original upstart algorithm and its variants depending on the targets ζ and the output σ of the current mother unit (or master output unit) $\mathcal{M}$. The target "$*$" means that the pattern is not included in the training set of $\mathcal{U}_i^{\pm}$ for all algorithms, whereas a bracket $[\cdot]$ around a target has the same meaning for upstart III where only one type of unit, $\mathcal{U}_i^+$ or $\mathcal{U}_i^-$, is created per generation.

Output σ	Target ζ	
	$\zeta = 1$	$\zeta = 0$
$\sigma = 1$	correctly-on $\mathcal{U}_i^+$ $*$ $\mathcal{U}_i^-$ 0	wrongly-on $\mathcal{U}_i^+$ [0] $\mathcal{U}_i^-$ 1
$\sigma = 0$	wrongly-off $\mathcal{U}_i^+$ 1 $\mathcal{U}_i^-$ [0]	correctly-off $\mathcal{U}_i^+$ 0 $\mathcal{U}_i^-$ $*$

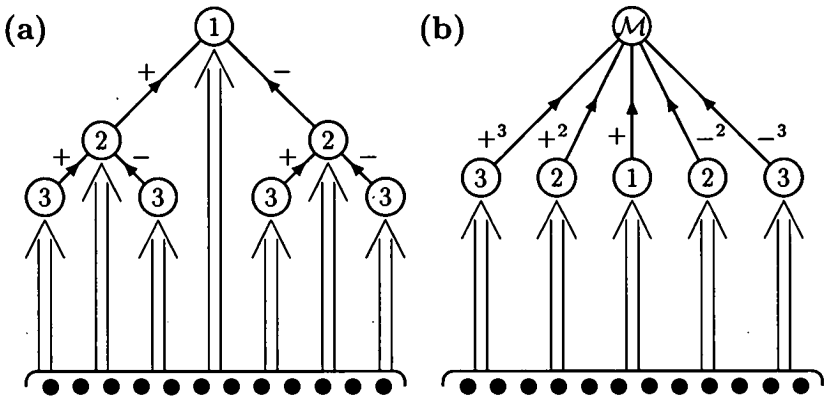

Figure 4.1. Networks of three generations produced by the original upstart algorithm (a) and the modified upstart II algorithm (b). The number of units of the original algorithm grows exponentially with each generation whereas the modified version grows only linearly. The black dots represent the input units. The open circles are hidden and output units created by the two version of the upstart algorithm numbered after their generation. The wide arrows symbolize input weights from all the input units, whereas the normal arrows represent single weights between hidden units and to the output unit $\mathcal{M}$. The plus and minus signs are the sign of the connecting weights and the powers give an indication of their magnitude.

Note, that in principle it would not be necessary to include the wrongly-off patterns in $\mathcal{U}_i^-$'s training set (and similarly wrongly-on patterns in $\mathcal{U}_i^+$'s training set) if only one unit is constructed per generation, since in this case frustration of the output unit is not an issue that has to be addressed. In order to investigate possible efficiency gains, a further variant, upstart III, is proposed, which always creates only one type of unit and can therefore implement above mentioned reductions in training sets. As for upstart II, we again consider both criteria (a,b).

The formal definition of these versions of the upstart algorithm is as follows:

Step 0: Create an asymmetric Boolean $\{0,1\}$ output unit $\mathcal{M}$ with threshold 1, to which all other units, forming the hidden layer, will be attached to. Subsequently, create the initial processing unit $\mathcal{U}_1^+$ train it on the original targets ζ^μ, freeze its weights, and connect it to $\mathcal{M}$ with a $+1$ weight, i.e., $\mathcal{M}$ has initially the same outputs as $\mathcal{U}_1^+$. Initialize the generation index $i = 1$ and the index for $\mathcal{U}^+$ and $\mathcal{U}^-$ units to $p = 1$ and $m = 0$

Step 1: Evaluate the number of wrongly-off and wrongly-on errors, ϵ^{on} and ϵ^{off}, made by $\mathcal{M}$ in generation i. Terminate if all patterns are correct, otherwise calculate the error probabilities to be applied, i.e., $p^{\text{off}} = P(\epsilon^{\text{off}})$ and $p^{\text{on}} = P(\epsilon^{\text{on}})$ for

criterion (a) or $p^{\text{off}} = P(\epsilon^{\text{off}}|\zeta^\mu = 1)$ and $p^{\text{on}} = P(\epsilon^{\text{on}}|\zeta^\mu = 0)$ for criterion (b). Then create unit(s) according to the employed variant, i.e., if $p^{\text{on}} > p^{\text{off}}$ then $\mathcal{U}_{i+1} := \mathcal{U}^-_{m+1}$, if $p^{\text{on}} < p^{\text{off}}$ then $\mathcal{U}_{i+1} := \mathcal{U}^+_{p+1}$, and otherwise ($p^{\text{on}} = p^{\text{off}}$) $\mathcal{U}_{i+1} := \mathcal{U}^-_{m+1} + \mathcal{U}^+_{p+1}$ for upstart II and $\mathcal{U}_{i+1} := \mathcal{U}^+_{p+1}$ for upstart III. The targets and training sets of the new unit(s) are given by Table 4.1.

Step 2: The new unit(s) are trained on their new training set(s) and their weights are frozen.

Step 3: The new unit(s), $\mathcal{U}^+_{p+1}$ and/or $\mathcal{U}^-_{m+1}$, are connected with positive respectively negative weight of identical magnitude to the output unit $\mathcal{M}$. The magnitude is adjusted so that previous decisions are overruled if one new unit is active. Go back to **Step 1**.

Similarly to the tiling-like algorithm, these versions of the upstart converge eventually, as each new generation reduces the total error by at least one pattern per created unit.

4.3 Calculation of the capacity in a replica framework

As mentioned in Section 3.2.1, the capacity limit is defined in probabilistic terms, as the property of being able to realize a mapping on "average" over all possible training sets. In the statistical mechanics community this problem has been addressed in several ways, all of them using the same basic technique. The initial approach (Gardner 1988), calculates the average (logarithm of the) volume in parameter space, which implements the training set perfectly. The reason for calculating the logarithm of the volume (often termed *Gardner volume*) rather than the volume itself, is that the first is assumed to be *self-averaging* in the thermodynamic limit of infinite input dimension ($N \to \infty$), whereas the second is not. The capacity limit is reached, when this volume vanishes. The second approach (Gardner and Derrida 1988), calculates the average free energy (corresponding to the error for the Gardner–Derrida cost function) the network makes for a given training set size and training temperature. In the limit of zero training temperature, the capacity limit is determined by the largest training set size with zero error, which has been the technique we employed in Chapter 3. The third approach (Monasson and Zecchina 1995), calculating the (logarithmic) number (and weight volume size) of implementable faithful internal representation, is similar to the Gardner-volume approach and is especially applicable in MLPs. This approach also allows the calculation of the cell size spectrum in weight space (Engel and Weigt 1996; Weigt and Engel 1997) providing further insights into the structure of the weight

solutions. All approaches have, however, in common, that the average of the logarithmic quantity over the input and output distributions is in all cases performed by using the replica trick, which replaces the average over the logarithm at the expense of introducing replicated network parameters.

These replica calculations are notoriously difficult for MLPs. Furthermore, it is not clear how the above frameworks can be extended to the case of constructive algorithms, where the architecture is not fixed *a priori*, but evolves during training and two instances of the same training set size could, for example, lead to different architectures. However, in the case of constructive algorithms, the optimization of each perceptron is performed individually, severely constraining the interaction between the different perceptrons. The influence of the previous perceptrons is only via the modification of the training set, by redefining the targets and/or selecting a subset of the patterns. Since the original targets are random, it could be argued that the redefined targets are approximately random as well, but coming from a distribution with modified bias. In this case, the quenched average over the patterns decouples and the capacity of networks built by constructive algorithms can be calculated from results derived for simple perceptrons: the errors made be the perceptron(s) of the current generation determine the example load α and the bias of the output distribution m_o for the next generation.

4.3.1 Assessing the influence of correlations

In order to assess whether this approximation is justifiable, a replica calculation for two perceptrons created in consecutive steps of the considered constructive algorithms has been carried out within a RS ansatz. These correlations should be dominant in comparison to correlations between perceptrons which are more than one generation apart. The details of the calculation and the results are reported in Appendix 4.A, here, only the main implication will be reported. For both the upstart and the tiling-like algorithm, the effect of correlations between consecutive units on the capacity limit or the error rate is usually insignificant in comparison to the effects of 1RSB in the individual perceptrons. For the tiling-like algorithm, the effects of correlations are usually smaller than for the upstart algorithm and one even finds situations where the correlations are non-existent (zero bias and small stability). Although, one can identify one region, small (but finite) m_o, large κ and α around the capacity limit, where correlations are substantial, this region should be only relevant for small networks and will have no bearing on the results presented here.

These correlation results may be compared to capacity results for fixed two-layer architectures with unconstrained optimization. For the parity machine with fixed architecture, which is somewhat related to the tiling-like algorithm, one finds that the

correlations (in terms of overlaps) between units are zero for any number of hidden units K (Engel et al. 1992) (for zero stability and unbiased output distribution), leading to the same capacity for tree and fully-connected architectures. For the fully-connected committee machine, which is in spirit more similar to the upstart algorithm, one finds (anti-) correlations in the capacity calculation for finite K, which vanish proportional to K^{-1} in the limit of large K (Urbanczik 1997; Xiong et al. 1997), leading only to a correction in the prefactor when compared to the tree architecture (Monasson and Zecchina 1995; Monasson and Zecchina 1996).

The two most relevant features found for the correlations hold for all constructive algorithms considered here and should carry over at least qualitatively to a more accurate 1RSB calculation (which is beyond the scope of this work). First, any correlation between the perceptrons leads to a decreased capacity of the combined network or to an increased error rate above saturation[6]. The uncoupled approximation should, therefore, constitute at least an upper bound to the true capacity, if this result holds qualitatively when accounting for RSB[7]. Furthermore, the correlations vanish in the region where both units are highly saturated, which could be considered the most relevant region, since most units operate in this regime for large networks. Hence, we believe that the upper bound calculated should be relatively tight, especially for the tiling-like algorithm.

4.3.2 Capacity and error rates for single perceptrons

After having assessed the influence correlations, and decided that their impact is less significant to negligible in comparison to 1RSB in the single perceptron, the purpose of this section is to briefly review the results for the capacity and the error rates for simple perceptrons insofar they are relevant for the ensuing capacity calculation and have not been covered in Chapter 3. We limit ourselves to the case of the spherical perceptron as the Ising perceptron behaves generically similar.

The capacity limit, α_c, of a simple perceptron is only a function of two parameters[8]: the output bias m_o and the stability κ. It is evident, that an increase in stability leads to a decrease in the capacity, whereas an increase in output bias m_o leads to an increase

[6]This may seem somewhat surprising initially, since the anti-correlations in the committee-machine result in an increase of the capacity of the fully-connected in comparison to the tree committee-machine. However, this effect may be explained by the constrained optimization for constructive algorithms and/or the increase in functional flexibility when going from a tree to on overlapping architecture.

[7]The breakdown of 1RSB in the individual perceptrons does not constitute a problem since further RSB steps increase the error rate (Whyte and Sherrington 1996).

[8]The third potential parameter, the input bias m_i, can be absorbed by a suitable rescaling of the stability κ and can therefore be set to zero w.l.o.g..

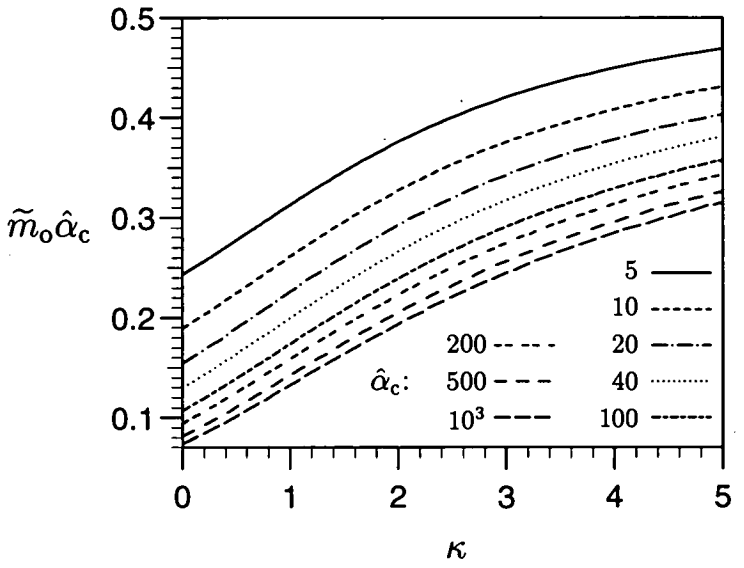

Figure 4.2. The bias $\widetilde{m}_o$ (or rather $\widetilde{m}_o\hat{\alpha}_c$ to highlight the scaling of α_c with $\widetilde{m}_o$) is shown as a function of the stability κ for several fixed normalized capacities $\hat{\alpha}_c(m_o,\kappa) \equiv \alpha_c(m_o,\kappa)/\alpha_c(0,\kappa)$ (see the legend). The increase of $\widetilde{m}_o$ with κ shows that for larger κ a smaller bias m_o is sufficient to realize the same increase in the normalized capacity. The normalization of $\widetilde{m}_o$ with $\hat{\alpha}_c$ shows the capacity grows slightly slower than linearly in $\widetilde{m}_o^{-1}$.

in the capacity, with $\alpha_c \to \infty$ for $m_o \to \pm 1$. It is therefore convenient to introduce the "bias" $\widetilde{m}_o \equiv (1-|m_o|)$ for large magnitude of the bias. In Figure 4.2, $\widetilde{m}_o$ is shown as a function of the stability κ for several fixed normalized capacities $\hat{\alpha}_c$, here defined as the ratio between the capacity for non-zero and zero bias $\hat{\alpha}_c(m_o,\kappa) \equiv \alpha_c(m_o,\kappa)/\alpha_c(0,\kappa)$. The increase of $\widetilde{m}_o$ with κ shows that for larger κ a smaller bias m_o is sufficient to realize the same normalized capacity. The other important feature, can be seen by the normalization of the curves in $\widetilde{m}_o$ with $\hat{\alpha}_c$, which demonstrates that the capacity α_c grows slightly slower than linearly in $\widetilde{m}_o^{-1}$ for all κ. Extending the asymptotic result for $\widetilde{m}_o \to 0$ from (Gardner 1988) to finite stability κ, one finds to leading orders

$$\alpha_c(\widetilde{m}_o) = \frac{1}{\widetilde{m}_o}\left\{\log(\widetilde{m}_o^{-1}) + 2\sqrt{2}\sqrt{\log(\widetilde{m}_o^{-1})}\,\kappa + \mathcal{O}(\log[\log(\widetilde{m}_o^{-1})])\right\}^{-1}, \tag{4.1}$$

shows that the sublinearity of α_c is dominated for $\widetilde{m}_o \to 0$ by the term $1/\log(\widetilde{m}_o^{-1})$ independent of the stability κ for $\sqrt{\log(\widetilde{m}_o^{-1})} \gg \kappa$. The increase of $\widetilde{m}_o$ with κ for constant $\hat{\alpha}_c$ can therefore be explained by the decreasing relevance of κ in determining the capacity for large bias m_o (or large α_c), when compared to $m_o = 0$, where one finds

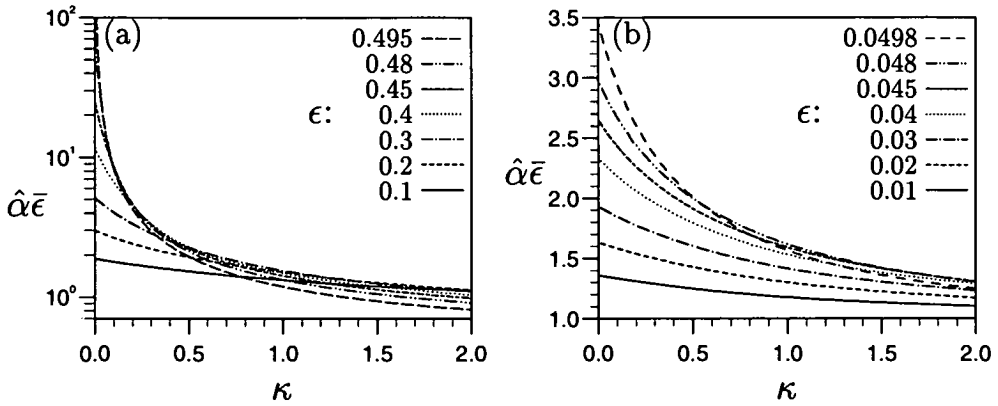

Figure 4.3. The normalized load $\hat{\alpha} \equiv \alpha/\alpha_c$ [or rather $\hat{\alpha}\bar{\epsilon}$ (see the text) to highlight the scaling of ϵ with α] is shown as a function of the stability κ for several fixed error rates ϵ (see the legend) and two output bias values (a) $m_o = 0$ and (b) $m_o = 0.9$. The decrease of $\hat{\alpha}$ with κ shows that the error increases more quickly for larger stability even if accounting for the decrease of the capacity. The normalization of $\hat{\alpha}$ with $\bar{\epsilon} \equiv (\epsilon^\infty - \epsilon)/\epsilon^\infty$ helps to highlight the deviation of the scaling of $\bar{\epsilon}$ from α^{-1}.

for $\kappa \to \infty$ to leading order $\alpha_c(\kappa) = (1 + \kappa^2)^{-1}$. The capacity of the Ising perceptron in the limit $\tilde{m}_o \to 0$ can be calculated along similar lines using self-consistent ansätze for the order parameters (justified by numerical results) yielding the identical result up to leading order in κ as in Eq. (4.1) but for a rescaling of α_c by $2/\pi$.

Above the capacity limit, let us just briefly review the behaviour of the perceptron as described in detail in Chapter 3. In this case, the perceptron has to misclassify a certain fraction of the example set, expressed in the error rate, ϵ, which depends on the output bias m_o, the stability κ, and on the example load α. It is self-evident that an increase in α beyond the capacity limit is followed by an increase in ϵ. In order to assess how this increase relates to the stability κ, Figure 4.3 shows the normalized load $\hat{\alpha} \equiv \alpha/\alpha_c$ as a function of κ for various fixed error rates ϵ. The dependence on the output bias m_o is illustrated by the choice of $m_o = 0$ and $m_o = 0.9$ in Figures 4.3(a) and 4.3(b), respectively. In order to highlight the scaling behaviour of $\hat{\alpha}$ with the error rate, $\hat{\alpha}$ was adjusted by the normalized remnant error $\bar{\epsilon} \equiv (\epsilon^\infty - \epsilon)/\epsilon^\infty$, where ϵ^∞ is the asymptotic error rate for $\alpha \to \infty$.

For both output bias values, one finds that $\hat{\alpha}$ increases (for given $\bar{\epsilon}$) for decreasing stability κ, an effect which is somewhat reverse to the observation made for the capacity limit as described in Figure 4.2. That this effect is more pronounced for $m_o = 0$ than for $m_o \neq 0$ has its root in the changing structure of solution space. For $m_o \neq 0$, the solutions are always characterized by a non-zero threshold θ, reflecting the non-

zero output bias. The asymptotic error rate (in α) is given by $\epsilon^{\infty} = (1 - |m_0|)/2$, which corresponds to deterministically classifying the larger example class correctly and misclassifying the smaller example class by using a threshold of infinite absolute value. The asymptotic error rate is approached by a power law of α^{-1}, corresponding to the rescaling used in Figure 4.3, modified by κ-dependent logarithmic corrections, which explain the κ-dependence found in Figure 4.3. For $m_0 = 0$, the behaviour is more complex. Initially, the solutions for all κ are characterized by zero threshold and $\epsilon^{\text{on}} = \epsilon^{\text{off}} = \epsilon/2$ as expected by the pattern symmetry. However, for any finite stability exists a critical load α_p (with $\alpha_p \to \infty$ for $\kappa \to 0$) at which a phase transition to a solution with non-zero threshold occurs, breaking the symmetry of the patterns ($\epsilon^{\text{on}} \neq \epsilon^{\text{off}}$). This is caused by the fact that the zero-threshold solution has a κ-dependent asymptotic error rate of $\epsilon^{\infty} = 1 - H(\kappa) \geq 1/2$, which is strictly larger than $1/2$ for any finite stability, making it advantageous to adopt the strategy of the non-zero bias case to classify the examples deterministically for $\alpha \to \infty$. The asymptotic error rate is approached by a power laws of $1/\alpha$ (with logarithmic corrections) for the non-zero threshold solution as for the non-zero bias case, but of $1/\sqrt{\alpha}$ (with logarithmic corrections) for the zero threshold solution, applicable for very small κ.

4.3.3 Employing results for the simple perceptron

In this section it is shown how the results for the capacity limit and the error rates of simple perceptrons demonstrated above can be used to calculate the capacity of the considered constructive algorithms. As an example, consider the capacity limit of a network with K units constructed by the tiling-like algorithm for given initial output bias, m_0, and stability, κ. For convenience, the example load on the whole network and capacity of individual perceptron units are expressed in terms of $\alpha \equiv p/N$ and α_c, respectively, whereas the capacity of the whole network is defined as $\alpha_c^K \equiv \alpha/K$ (and $\alpha_c^1 = \alpha_c$).

Assume that the current guess of the network capacity at iteration j is α_j^K resulting in an initial example load of $\alpha_1 = K\alpha_j^K$ with output bias $m_1 = m_0$. These parameters together with the desired stability κ determine the error rate ϵ_1 made by the first perceptron $\mathcal{U}_1$. The example load α_2 and bias m_2 of the second perceptron $\mathcal{U}_2$ result by simply applying the rules of the algorithm: the load $\alpha_2 = \alpha_1$, since the complete training set is used, and the bias $m_2 = 1 - 2\epsilon_1$ (or the more natural parameterization $\tilde{m}_2 = 1 - m_2 = 2\epsilon_1$), since the target of all examples but the erroneous ones is $+1$. Obviously, these parameters determine ϵ_2 and this procedure is repeated until the last perceptron $\mathcal{U}_K$ which is supposed to have reached its capacity $\alpha_K = \alpha_1$ for $\tilde{m}_K = 2\epsilon_{K-1}$. Therefore, the actual capacity limit α_c for $\tilde{m}_K$ is calculated and compared

to α_K: if the last perceptron is below its capacity limit, the true capacity $\alpha_c^K > \alpha_j^K$ otherwise $\alpha_c^K < \alpha_j^K$ and a root solving routine can be employed to solve

$$[\alpha_c(\widetilde{m}_K) - \alpha_K] = 0, \tag{4.2}$$

as a function of α_c^K. A symbolic program for the procedure called by the root solver can then be written as outlined in Figure 4.4.

```
tile_cap := proc(α_c^K)                    % begin procedure
global K, κ, m_o;                          % global variables
local ···;                                 % local variables (here unspecified)
α_1 := K α_c^K; m̃_1 := 1 − m_o;            % initialize algorithm
for (i = 1, K − 1) do                      % loop over the erroneous perceptrons
    ε_i := error_calc(α_i, m̃_i, κ);        % error rate calculating procedure
    α_{i+1} := α_i; m̃_{i+1} := 2ε_i;       % calculate new parameters
od;                                        % have reached last perceptron
α_c := perc_cap(m̃_K, κ)                    % calculate capacity limit
RETURN(α_c − α_K);                         % return difference between capacity and load
end;
```

Figure 4.4. A symbolic capacity calculation procedure of the tiling-like algorithm called by an all-purpose root solving routine.

It is now more clear, what the relevance of the results for the simple perceptron presented in Section 4.3.2 is. The error rate of the previous perceptron determines the output distribution bias of the current perceptron and the curves of constant error in Figure 4.3 are curves of constant bias for the next generation. With each step of the algorithm, the decreased "bias" $\widetilde{m}_o$ leads to a reduced error rate, until the capacity limit for the current bias is larger than the current load α. The influence of the stability is therefore twofold. The increase in $\widetilde{m}_o$ for constant $\hat{\alpha}_c$ with κ (observed in Figure 4.2) should have the effect of increasing the normalized capacity limit, $\hat{\alpha}_c^K \equiv \alpha_c^K/\alpha_c$, with κ of the whole network, whereas the decrease of $\hat{\alpha}$ for fixed error with κ should have qualitatively the opposite effect.

For the upstart algorithm similar consideration are applied to yield a procedure for a root solver that calculates the capacity of the created networks. The resulting procedures are much more complicated and summarized in Appendix 4.B.

Note, that the numerical uncertainty in the solution of the order parameter, mainly caused by the numerical integration inaccuracy, results in an error in the error rate calculation. Propagating the upper and lower bound of the error rate in each genera-

tion separately through the whole network gives estimated error bars in the capacity[9]. Although the relative capacity error increases with network size it never exceeded 10^{-4} and could therefore be neglected.

4.4 Numerical capacity results

Within the uncorrelated approximation, the procedures described within Section 4.3.3 (and Appendix 4.B) have been used to calculate the capacity of the considered constructive algorithms. This section will roughly fall into three parts. In Section 4.4.1 numerical capacity results for an unbiased random mapping, usually considered in capacity calculations, will be presented as a function of K using the various algorithms, different stabilities κ, and employing either the RS or 1RSB ansatz. In Section 4.4.2 biased output distributions will be considered for $\kappa = 0$ and the different algorithms compared, followed by a an investigation into the capacity for the Ising weight prior instead of the spherical weight prior in Section 4.4.3. The functional behaviour of these numerical results will then analysed for finite K and suggestions are made concerning the asymptotic limits of the capacity for $K \to \infty$ in Section 4.5.

4.4.1 Capacity for unbiased outputs

Previous capacity calculation for MLPs have only investigated zero stability for unbiased output distributions. In this section, one aim is therefore to assess the influence of finite stability on the asymptotic functional form of the capacity limit for unbiased output distributions. Another goal is to compare the capacity[10] between the networks built by the different constructive algorithms considered[11].

In Figure 4.5, the capacity limit of networks constructed by the upstart II algorithm is shown as a function of the number of hidden units for various stabilities κ for the uncoupled RS [Figure 4.5(a)] and 1RSB [Figure 4.5(b)] ansätze. Note, that the capacity curves have been normalized by $\alpha_c^1(\kappa)$ for presentational reasons. For both ansätze, one finds that the capacity grows monotonically for small stabilities. for larger stabilities

[9]This technique was compared with the change in the capacity resulting from relaxing the accuracy requirements for the numercial integration over several order of magnitude and it was found that the error propagation method overestimates the capacity error by about two orders of magnitude.

[10]For brevity, the capacity limit of networks constructed by an algorithm will often be referred to just as the capacity of the algorithm.

[11]Note, that for unbiased output distributions, the two selection criteria considered for the variants of the upstart algorithm are identical, and the two variants will therefore be referred to as upstart II and III in this section.

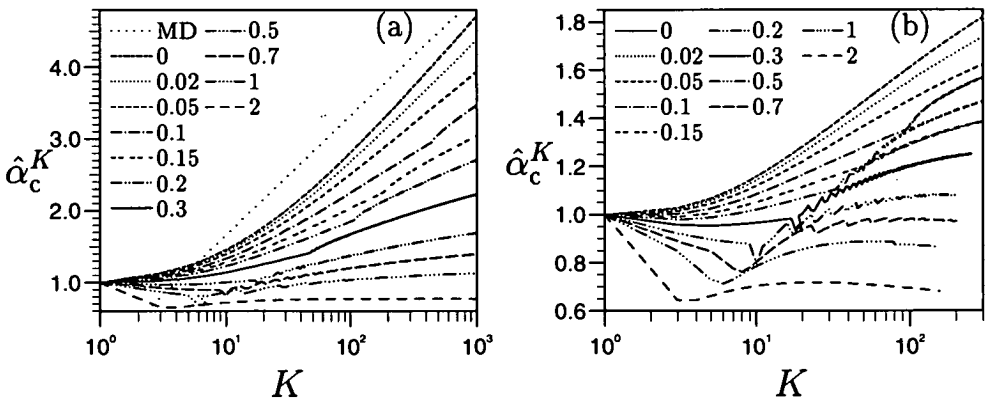

Figure 4.5. The normalized capacity limit $\hat{\alpha}_c^K \equiv \alpha_c^K/\alpha_c^1$ [where $\alpha_c^1(\kappa)$ is the capacity of a simple perceptron] of networks constructed by upstart II is shown as a function of the number of hidden units K for several stabilities κ (see the legend) for the uncoupled (a) RS and (b) 1RSB ansätze. The RS-capacity violates the superimposed MD bound for small stabilities and large enough K in the range of hidden units investigated.

the capacity decreases initially for small K but increases for larger K. This non-monotonic behaviour can be explained by the initial inefficiency of the upstart algorithm by tackling the two type of errors with two different unit types, such that the network grows directly from one to three units. For even larger stabilities, one finds that the capacity actually decreases in the $K \to \infty$ limit[12].

Notice that for finite stability κ and large enough K, one finds a kink in the capacity curve due to the phase transition in the solution of the first perceptron from zero to finite threshold, which leads to a breaking of the error symmetry and consequent network symmetry for upstart II. The asymmetry in the error also leads to the jags in the capacity as the two types of units cease to saturate simultaneously. Furthermore, due to the increasingly deterministic classification of the first perceptron (for an explanation see Section 4.3.2), wrongly-off errors become increasingly more common than wrongly-on errors, resulting in the upstart II algorithm constructing much more $\mathcal{U}^+$ than $\mathcal{U}^-$ nodes[13]. In fact, for certain stabilities, one finds that a $\mathcal{U}^-$ node can actually vanish before a new $\mathcal{U}^+$ node needs to be created, when increasing the load α on the first perceptron, leading to a decrease in total network size (e.g., $\kappa = 0.3$ for the 1RSB ansatz, where the decrease in network size at $K = 20$ is apparent).

In comparison to the bounds and capacity limits known for fixed-architecture MLPs,

[12]Note that the decrease is in the capacity per weight of the networks, the capacity of the network still increases linearly in K to leading order.

[13]Or vice versa due to random symmetry breaking in the first perceptron.

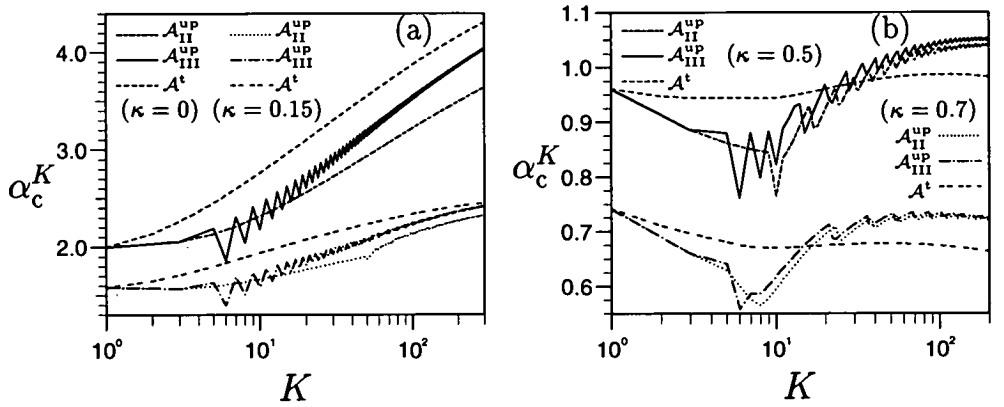

Figure 4.6. The capacity limit α_c^K of networks constructed by the upstart II ($\mathcal{A}_{\rm II}^{\rm up}$), upstart III ($\mathcal{A}_{\rm III}^{\rm up}$), and tiling-like ($\mathcal{A}^{\rm t}$) algorithms is shown as a function of the number of hidden units K for several stabilities: (a) $\kappa = 0$ and $\kappa = 0.15$; (b) $\kappa = 0.5$ and $\kappa = 0.7$.

one finds that the capacity curves for networks built by upstart II violate the MD bound within the uncoupled RS ansatz for large enough number of hidden units; in the curve shown the slope of the RS curve is larger than the MD bound for $K \approx 80$[14]. However, within the uncoupled 1RSB ansatz the MD bound is not saturated. Although, the 1RSB results initially seem to predict a logarithmic increase of the capacity[15] $\alpha_c^K \propto \log K$ for small stabilities and network sizes $20 \lessapprox K \lessapprox 150$ as reported in preliminary work (West and Saad 1997), this functional description is inadequate for larger stabilities and/or larger network sizes and a detailed analysis will be carried out in Section 4.5.

In Figure 4.6, the capacity resulting from the different constructive algorithms considered is investigated for a few stabilities. For both small and large stabilities, shown in Figures 4.6(a) and 4.6(b) respectively, one finds that the tiling-like algorithm has a larger capacity for small network size and for small stabilities than both variants of the upstart algorithm, which may be mainly attributed to the fact that the tiling-like algorithm attempts to correct both error types simultaneously.

However, asymptotically the tiling-like algorithm is less efficient than both variants of the upstart algorithm for larger stabilities, but even for small stabilities the capacity curves for upstart networks (in particular for upstart III) approach those of the tiling-like network. This behaviour may be explained by the fact that the upstart algorithm

[14]Within RS the slope of the tiling-like and the upstart III algorithms violates the MD bound above $K \approx 20$ and $K \approx 50$, respectively. Note that this result only holds for the range of $K \leq 4000$ investigated since the asymptotic values could not be calculated self-consistently in Section 4.5.

[15]The estimated prefactors are significantly smaller than the $(\log 2)^{-1}$ of the MD bound.

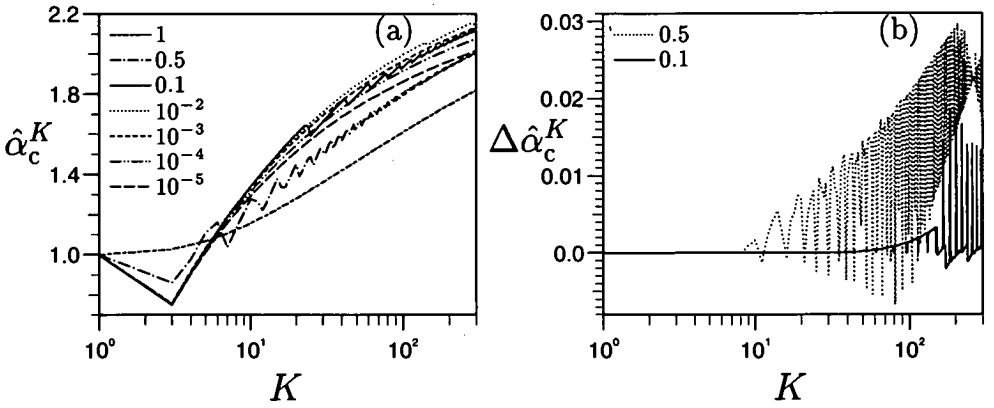

Figure 4.7. (a) The normalized capacity limit $\hat{\alpha}_c^K \equiv \alpha_c^K/\alpha_c^1$ [where $\alpha_c^1(\widetilde{m}_o)$ is the capacity of a simple perceptron] of networks constructed by upstart IIb is shown as a function of the number of hidden units K for several "biases" $\widetilde{m}_o$ (see the legend) for the uncoupled 1RSB ansatz. (b) To highlight the influence of the selection criteria, the difference between the normalized capacity limits $\Delta\hat{\alpha}_c^K = \hat{\alpha}_c^K(\mathcal{A}_{\text{IIa}}^{\text{up}}) - \hat{\alpha}_c^K(\mathcal{A}_{\text{IIb}}^{\text{up}})$ of the two selection criteria is shown for $\widetilde{m}_o = 0.5$ and $\widetilde{m}_o = 0.1$, suggesting that on average upstart IIa outperforms upstart IIb.

eliminates part of the original training data in the training set of consecutive units.

This argument can also explain the fact that upstart III is more efficient than upstart II for all stabilities as it eliminates in general more patterns from the training set. This advantage, however, becomes less significant for large κ, where almost all errors are wrongly-off and consequently almost all units are of the $\mathcal{U}^+$ type beyond the phase transition of the first perceptron. The fraction of wrongly-on errors eliminated from the training sets of $\mathcal{U}^+$ units is therefore small and the two versions behave similarly.

4.4.2 Capacity for biased outputs and zero stability

Similarly to exploring finite stabilities as above, it is interesting to address the influence of biased output distributions on the capacity limit of MLPs for which no results are a known in the case of fixed architectures. Due to the symmetry it is sufficient to study $m_o < 0$ w.l.o.g., for $m_o > 0$ the rôles of of wrongly-on and wrongly-off errors reverse and consequently the rôles of $\mathcal{U}^-$ and $\mathcal{U}^+$ units for the upstart algorithm. Again, we would like to compare the capacity between the different considered constructive algorithms and for the variants of the upstart algorithm also the criteria selecting the next unit type.

In Figure 4.7(a), the capacity limit of networks constructed by the upstart IIb

algorithm is shown as a function of the number of hidden units for various "biases" $\widetilde{m}_o \equiv 1 - |m_o|$ for the uncoupled 1RSB ansatz. Again, the capacity curves have been normalized by $\alpha_c^1(\widetilde{m}_o)$ for presentational reasons. Although, the normalized capacity limit for biased output distributions is initially larger than less biased or unbiased output distribution for $K > 6$, one finds asymptotically (in K) that the (normalized) slope[16] seems to decrease for increasing bias (see below), suggesting that the constructive algorithms are less efficient (when compared to a single perceptron) in exploiting the bias of the output distribution. The curves are jagged for finite but small bias, since both units type are constructed and the units do not saturate simultaneously. For very large bias, the larger example class is almost deterministically classified correctly and only one type of unit is constructed to correctly classify the smaller class, again leading to smooth capacity curves.

The influence of the unit creation criterion is therefore only important for small bias and its influence on the capacity limit is depicted in Figure 4.7(b) and shows that upstart IIa is slightly more efficient than upstart IIb. A more detailed examination of the constructed networks shows that both criteria are not ideal. For unbiased output distribution, one finds that after the breaking of the network symmetry (i.e., beyond the phase transition of the solution of the first perceptron for finite κ), the algorithm initially only creates $\mathcal{U}^+$ units until both error types have the same frequency after which it alters between the unit types. We find this creation scheme the most natural and believe that it is probably also optimal. For finite output bias, criterion (a), using the number of errors as the decision criterion, builds $\mathcal{U}^-$ units too early into the network, i.e., instead of alternating between unit types at the end the $\mathcal{U}^-$ units are dispersed less frequently over a wider unit number range. This may be considered a wasteful use of $\mathcal{U}^-$ units.

Criterion (b), basing its selection on the number of errors made normalized by the size of the its target class, alleviates this problem leading to networks with fewer $\mathcal{U}^-$ units for fixed total network size, however, in this case we find that the creation of $\mathcal{U}^-$ units tends to be left too late, leading to extra $\mathcal{U}^+$ units that have to be created at the very end to correct the few wrongly-on errors the $\mathcal{U}^-$ units make. In fact, for both creation criteria and for the last few units, we find sometimes that the algorithm actually decides on building a unit type which is below its capacity limit whereas the other unit type is above its capacity, i.e., criterion (a) selects a $\mathcal{U}^-$ although it should have selected a $\mathcal{U}^+$ and vice versa for criterion (b)[17].

[16] For larger bias the actual slope is still much larger due to the normalization factor $\alpha_c^1(\widetilde{m}_o)$ that scales with Eq. (4.1).

[17] To cater for those few cases, we have decided to amend both criteria such that the algorithm always

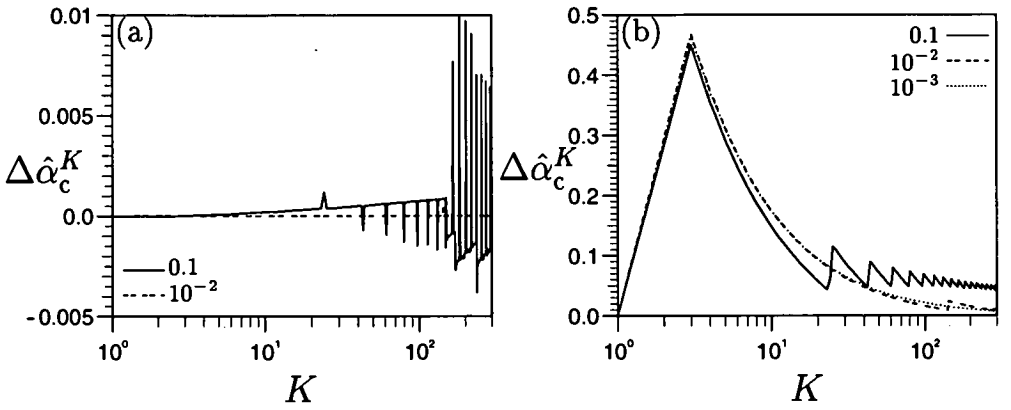

Figure 4.8. The influence of the constructive algorithm $\mathcal{A}$ is assessed by plotting the difference between the normalized capacity limits $\Delta\hat{\alpha}_c^K = \hat{\alpha}_c^K(\mathcal{A}) - \hat{\alpha}_c^K(\mathcal{A}_{\mathrm{IIb}}^{\mathrm{up}})$ for (a) where $\mathcal{A}$ is the upstart IIIb and (b) where $\mathcal{A}$ is the tiling-like algorithm and several bias values (see the legend).

Both criteria are therefore not optimal, criterion (a) selects $\mathcal{U}^-$ units too early and criterion (b) selects $\mathcal{U}^-$ units to late. A better criterion should therefore compromise somewhat between these two; however, we were not able to devise a more suitable objective criterion.

In Figure 4.8, upstart IIb is compared to upstart IIIb and the tiling-like algorithm. For the versions of the upstart algorithm [Figure 4.8(a)], the difference in capacity is very small and decays rapidly for increasing bias, since the difference in the training sets becomes negligible as the fraction of wrongly-on errors goes to zero and only one $\mathcal{U}^-$ unit is created in all network sizes investigated.

The difference of the upstart IIb to the tiling-like capacity [Figure 4.8(b)] is significant for small networks, due to the separate treatment of each error type. For large bias, the tiling-like capacity approaches the upstart IIb capacity rapidly, as almost all errors become wrongly-off and only few $\mathcal{U}^-$ units are created, consequently leading to almost identical training sets and networks besides the extra $\mathcal{U}^-$ unit built by the upstart algorithm.

This should be contrasted to large stabilities κ, where the upstart algorithm also constructs almost entirely a single unit type; however, in this case, the difference in the training sets between the upstart and tiling-like algorithm does not vanish asymptotically in κ. The fraction of correctly-on patterns excluded from the training set of all $\mathcal{U}^+$ units in the upstart algorithm approaches 1/2 of all patterns for later units in the

selects a unit above its capacity limit first.

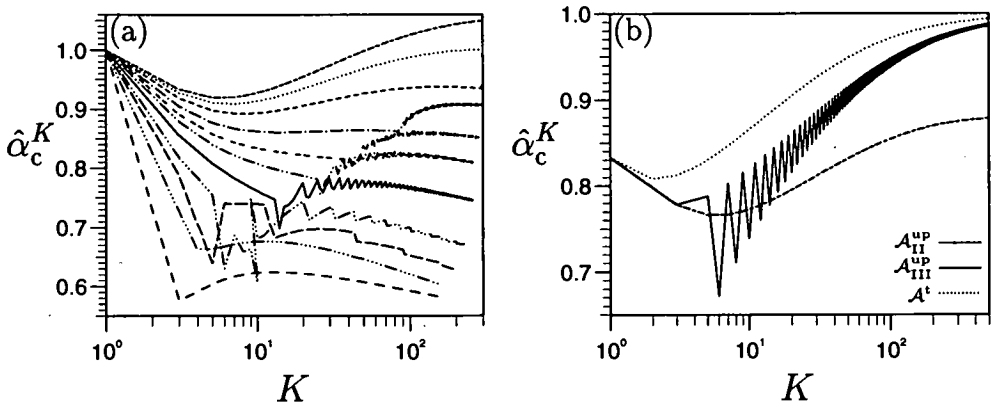

Figure 4.9. This figure illustrates the differences when using the Ising rather than the spherical perceptron as the basic building block of the constructive algorithms. (a) The normalized capacity limit $\hat{\alpha}_c^K \equiv \alpha_c^K/\alpha_c^1$ for networks constructed by upstart II is shown as a function of the number of hidden units K for several stabilities κ (from top to bottom $\kappa = 0$, 0.02, 0.05, 0.1, 0.15, 0.2, 0.3, 0.5, 0.7, 1.0, and 2.0, as in Figure 4.5) for the uncoupled 1RSB ansatz. (b) The capacity limit α_c^K of networks constructed by the upstart II ($\mathcal{A}_{II}^{up}$), upstart III ($\mathcal{A}_{III}^{up}$), and tiling-like ($\mathcal{A}^t$) algorithms is shown as a function of the number of hidden units K for $\kappa = 0$.

unbiased output distribution case, whereas this fraction approaches $\widetilde{m}_o/2$ in the biased output distribution case — vanishing for $\widetilde{m}_o \to 0$.

For small bias, the picture is less clear. The tiling-like capacity seems to decay to a value which has approximately a constant difference to the upstart capacity, although we have found for zero bias, that at least the upstart III capacity curve approaches that of the tiling-like algorithm. This difference may be explained by the suboptimality of both upstart selection criteria for $m_o \neq 0$.

4.4.3 Capacity for the Ising perceptron

Up to now, we have only considered the constructive algorithms using spherical perceptron with real valued weights of arbitrary accuracy as their basic building block. In realistic implementations weights are only stored up to a certain accuracy and especially for VLSI implementations Ising (binary $\{-1, 1\}$) weights are a often considered alternative. In this section, we therefore investigate the influence of an Ising weight prior for the perceptron, usually referred to as the *Ising perceptron* (in contrast to the *spherical* perceptron with real weights), on the capacity of the resulting networks. For brevity we will only consider unbiased output distributions and mainly networks constructed by the upstart II algorithm.

In Figure 4.9(a) the normalized capacity limit of networks constructed by the upstart II algorithm is shown as a function of the number of hidden units for various stabilities κ for the uncoupled 1RSB ansatz. As for the spherical perceptron, we find that not only the capacity but also the slope of the normalized capacity curves decrease with increasing stability. In comparison to the spherical perceptron, the normalized capacity is, however, much smaller; in fact, for all stabilities the normalized capacity decreases initially for small K. Although the capacity increases for very small κ for larger K, the curves flatten out asymptotically. For larger κ, the decrease of $\hat{\alpha}_c^K$ is only abated briefly due to the phase transition in the solution of the first perceptron (which occurs for much smaller α in the Ising perceptron), leading to the already observed kink in the capacity curves and the possibility of a decrease of the number of units for increasing load α.

In Figure 4.9(b) the dependence of the capacity curves on the constructive algorithm is investigated for $\kappa = 0$. As for small stability in the spherical perceptron, the tiling-like algorithm has the largest capacity for the range of hidden units investigated. Whereas, the upstart III capacity closes the gap in the capacity to the tiling-like algorithm, the upstart II algorithm seems to be asymptotically less efficient for this stability.

4.5 Analysis of the capacity

Although the visual inspection of the capacity curves already reveals some information about the efficiency of the considered constructive algorithms as a function of K and κ, it would be more useful to be able to model the capacity curves at least for large K with a reasonable functional form.

As mentioned above, a functional form of the capacity limit α_c^K linear in $\log K$ cannot fit the curves adequately for large κ and/or K. For $\kappa = 0$ and unbiased output distributions, the capacity results for committee and parity machines within the replica framework (Barkai et al. 1990; Monasson and Zecchina 1996; Urbanczik 1997; Xiong et al. 1997) yield power laws in $\log K$

$$\alpha_c^K \simeq c[\log K]^n, \tag{4.3}$$

with $n = \frac{1}{2}$ and $n = 1$ respectively. This suggests that power laws may hold also for constructive algorithms at least for $\kappa = 0$ and $m_o = 0$ which we will investigate first in this section. Later, we will extend our analysis to finite m_o and finite κ.

4.5.1 Zero stability and unbiased output distributions

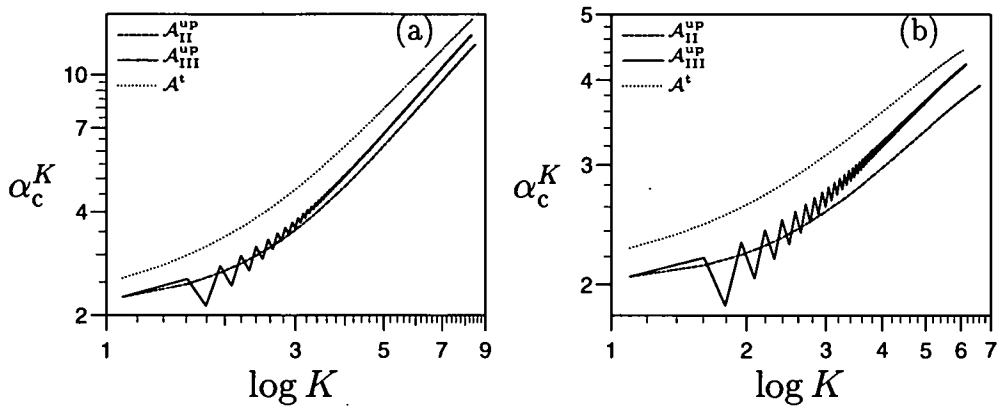

Figure 4.10. The power-law relationship between α_c^K and $\log K$ (4.3) is shown to hold approximately for all considered constructive algorithms [upstart II ($\mathcal{A}_{II}^{up}$), upstart III ($\mathcal{A}_{III}^{up}$), and tiling-like ($\mathcal{A}^{t}$)] for $\kappa = 0$ and (a) RS; (b) 1RSB.

In Figure 4.10, the power-law ansatz is scrutinized for the capacity curves of all considered algorithms for both RS and 1RSB, by plotting α_c^K vs. $\log K$ on a log-log scale. We find that the ansatz is reasonable for both cases, although the slope for 1RSB drops slightly for larger $\log K$. The determination of accurate exponents n (which are arguably more relevant than the prefactor c), however, is difficult for two reasons. As the power law is in $\log K$, the range for fitting the exponent is relatively short, since the calculation of the capacity is computationally quite expensive[18], making it impossible to calculate capacities over several decades of $\log K$. Furthermore, the power-law behaviour in $\log K$ seems impure, as locally (i.e., around a particular K value) calculated exponent values $n(K)$ exhibit small constant shifts. This may, for example, be caused by superimposed corrections decaying in $\log(\log K)$. This shift may be a cumulative effect of the error calculations propagated through the perceptrons as well as of the capacity calculation for the last perceptron(s).

These difficulties will become more apparent in the course of this analysis. For the time being, we would just like to express our reservations about the accuracy of the asymptotically derived exponent values. These should rather be seen as a local snapshot. Bearing this in mind, finite K measurements $n(K)$ were derived using a moving regression window. The resulting curve of $n(K)$ was then either averaged for the largest K available resulting in a local mean approximation n_l or extrapolated to $1/K \to 0$ using linear and quadratic regression models yielding n_e. This extrapolated estimate n_e should be seen as an indication in which direction n_l is moving rather than

[18]E.g., the calculation of the capacity limit for the tiling-like algorithm to up to $K = 300$ takes approximately 2 days (10 min) of CPU on a state-of-the-art workstation within the 1RSB (RS) ansatz.

a true extrapolation to n for $K \to \infty$.

These calculations still depends on the length of the regression window and the number of resulting $n(K)$ values included in the mean and extrapolated estimates, due to the noisy capacity curves for the upstart algorithms and the small constant shifts in $n(K)$. To minimize these effects, an average was taken over different lengths of the regression windows for local $n(K)$ values as well the number of $n(K)$ values included in the extrapolation. The values of n_l and n_e were then determined by a weighted mean of the individual resulting models by their respective likelihoods.

The final results for n_l and n_e are shown in Table 4.2 for the various constructive algorithms and both RS and 1RSB ansätze for the spherical perceptron. For this case, the power-law exponents were calculated once for the usual range of hidden units explored ($K = 300$ for 1RSB and $K = 1000$ for RS) and in most cases reevaluated for larger K ($450 \leq K \leq 750$ for 1RSB and $K = 4000$ for RS) in order to verify the size of the systematic errors.

For all algorithms, we observe that the resulting extrapolated estimate n_e is smaller than the local mean n_l, suggesting that n_l could be an upper bound to the true exponent n. As expected, the extrapolated estimates n_e themselves are not very accurate; the reevaluated local mean n_l for the larger K value shows a strong shift and is in many cases smaller than the extrapolation n_e for smaller K. It may be argued that this is

Table 4.2. The estimated power-law exponents n_l and n_e for $\kappa = 0$, the considered algorithms, and RS and 1RSB for two values of K in order to spot systematic errors.

$\mathcal{A}$	K^*	RSB		RS	
		n_l	n_e†	n_l	n_e
$\mathcal{A}_{\mathrm{II}}^{\mathrm{up}}$	300	0.554(2)	0.48(2)	1.291(0)	1.279(0)
	750	0.479(1)	0.36(2)	1.264(0)	1.243(0)
$\mathcal{A}_{\mathrm{III}}^{\mathrm{up}}$	300	0.602(1)	0.54(1)	1.305(0)	1.290(0)
	450	0.557(1)	0.47(1)	1.272(0)	1.247(0)
$\mathcal{A}^{\mathrm{t}}$	300	0.468(1)	0.36(3)	1.174(0)	1.167(0)
	450	0.428(1)	0.31(2)	1.158(0)	1.145(0)

*The numbers given for K apply for 1RSB; for RS $K = 1000$ and $K = 4000$ were used instead (lower and upper limit respectively).

†The error in the exponents is usually given for the leading digit in brackets only, i.e., 0.49(2) is equivalent to 0.49 ± 0.02. The error is either given as (0) when smaller than the last significant digits given.

due to the extrapolation carried out in $1/K$ rather than in $1/\log K$, however, such an extrapolation is not advisable within the values of K explored. From Table 4.2, we can therefore conclude that the K values explored are not in the asymptotic regime.

Comparing the exponents between the different algorithm may suggest that both upstart algorithms are asymptotically more efficient than the tiling-like algorithm and upstart III outperforms upstart II, with the reservation that the large finite size effects may undermine this observation. Furthermore, one can speculate upon whether the exponents n for RS are asymptotically larger than 1, i.e., violate the MD bound, as their finite K estimates suggest.

Finally, note that the exponents were calculated under the assumption of subsequent perceptrons being uncorrelated and the effect of correlation (and also higher RSB step) may cause a shift to smaller (local) n values, similarly to the transition from RS to 1RSB. Since RSB in the single perceptron is believed to be more relevant than correlations, this correction should, however, be significantly smaller and likely negligible in comparison to finite size effects in K.

One way to probe further into the asymptotic regime for fixed K is by studying large bias. For $\widetilde{m}_o \to 0$, the relevant α values are much larger, and the error rates and the capacity of most units are closer to their asymptotic expansions [see Chapter 3 and Eq. (4.1)].

4.5.2 Analysis for biased outputs and zero stability

To assess the influence of the output bias more objectively and also to gain some qualitative insight into the likely behaviour for much larger K in the more interesting case of 1RSB, we have calculated n_l for $K = 300$ and n_e along the same lines as above and present the results in Figure 4.11(a).

For all algorithms, we find that n initially decreases with increasing bias m_o, before both estimates level off. This result is inconsistent since n must either be constant or increase with m_o, otherwise the capacity curve for a smaller bias would eventually cross that for a larger bias. This result could have been anticipated from the decreasing slope for large bias observed in the raw capacity curve [see Figure 4.7(a)]. Evidently, this contradiction causes no actual violation for any practical range of hidden units[19], since the prefactor estimates c scale to leading order[20] with $c \propto \widetilde{m}_o^{-1}$ [see Figure 4.11(b)].

The question remains open whether the asymptotic exponents n will have any

[19]E.g., for the upstart III algorithm the capacity curve for $\widetilde{m}_o = 1$ were to cross the one for $\widetilde{m}_o = 10^{-5}$ for $K \approx \exp(10^{17})$.

[20]Note that the logarithmic correction to this leading behaviour for c resembles the result for the perceptron (4.1).

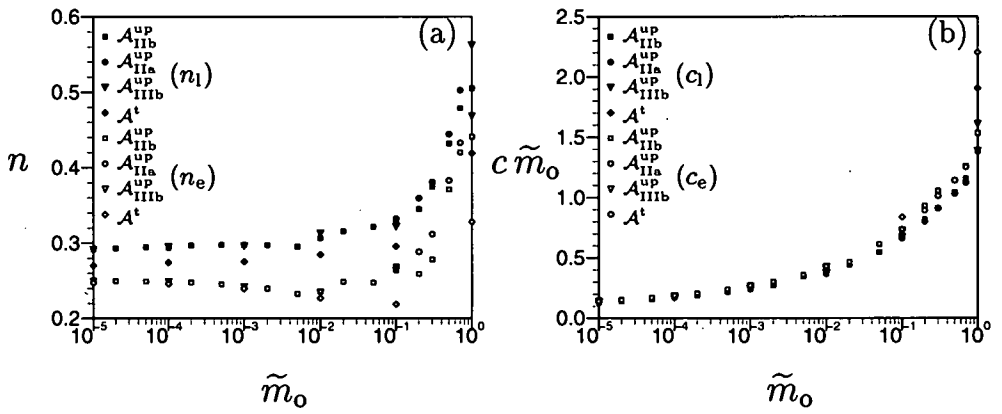

Figure 4.11. (a) The power-law exponent estimates n_l and n_e for the capacity limit $\alpha_c^K \propto [\log K]^n$ are shown as a function of the "bias" $\widetilde{m}_0 \equiv 1 - |m_0|$ for the various algorithms [upstart IIa ($\mathcal{A}^{up}_{IIa}$), upstart IIb ($\mathcal{A}^{up}_{IIb}$),upstart IIIb ($\mathcal{A}^{up}_{IIIb}$), and tiling-like ($\mathcal{A}^t$)] for the uncoupled 1RSB ansatz. (b) The corresponding prefactor c_l and c_e, where the values were adjusted by the dominant linear scaling in $\widetilde{m}_0^{-1}$. The local values were determined for $K = 300$ and are denoted by filled symbols, whereas the extrapolation estimates are represented by open symbols (see the legend). The estimation error for all estimates does roughly not exceed more than five times the size of the symbols, and is about their size in many cases, especially for small $\widetilde{m}_0$.

functional dependence on m_0. If n were independent of m_0, n would also have to be independent of the considered constructive algorithms for any bias since for $\widetilde{m}_0 \to 0$ their differences vanish as explained in Section 4.4.2. The performance of the constructive algorithms studied would then only vary in the prefactor, in contrast to the case of fixed architectures where n can be architecture dependent [see Eq. (4.3)].

Finally, it is worthwile illustrating how such inconsistency can arise. Consider the asymptotics of the perceptron capacity for $\widetilde{m}_0 \to 0$, for which an asymptotic expansion can be derived explicitly (4.1). A numerical determination of the leading asymptotic behaviour in $\log(\widetilde{m}_0^{-1})$ would require extremely small $\widetilde{m}_0$ values. For the κ-corrections to become negligible $\sqrt{\log(\widetilde{m}_0^{-1})} \gg 2\sqrt{2}\kappa$, i.e., representing the inequality as a small factor δ requires $\widetilde{m}_0 \approx \exp(-8\delta^2\kappa^2)$. The true κ-independent exponent (-1) is therefore numerically almost impossible to predict. In fact, for any small but finite $\widetilde{m}_0$, an numerical evaluation of the exponent for finite κ is always strictly larger than -1 and increasing with κ. At face value, this result would also inconsistently predict that asymptotically the capacity curves for larger κ cross those of smaller κ. Considering, that such numerical difficulties already mar asymptotic results for a simple perceptron, it may be of no surprise that consistent and accurate asymptotic results could not be

extracted in the case studied here.

4.5.3 Analysis for finite stability

Although we have shown in the previous section that consistent power-law exponents cannot be determined for the range of hidden units available, it is nevertheless interesting to study the local exponents for finite stabilities to gain a qualitative insight.

For finite stability, the fit of a power-law model on the capacity curves themselves deteriorates for increasing stability. This is mainly due to the non-negligible corrections to the asymptotic capacity limit of the last perceptron as described above. For example, if a power-law model is fit to the capacity limit of a simple perceptron as a function of $\log(\tilde{m}_o^{-1})$ for $\kappa = 2$ and the range of $\tilde{m}_o^{-1}$ relevant for the bias of the last perceptron within the range of hidden units explored, the estimated exponent is around $-\frac{1}{2}$ and increasing systematically towards the correct value -1.

In order to be able to extend the analysis to finite stability, it is therefore necessary to separate the cumulative effect of the errors and the capacity of the last perceptron. Consider, for example the tiling-like algorithm. The capacity of the complete network in terms of the capacity of the last perceptron is according to Eq. (4.2)

$$\alpha_c^K = \frac{1}{K}\alpha_c(\tilde{m}_K) \tag{4.4}$$

Since $\tilde{m}_K \to 0$ for $K \to \infty$, one can use the asymptotic expansion (4.1) of α_c to express the capacity solely in terms of $\tilde{m}_K$ (to leading order for $\tilde{m}_K \to 0$)

$$\alpha_c^K \simeq \frac{1}{K}\frac{1}{\tilde{m}_K \log(\tilde{m}_K^{-1})}. \tag{4.5}$$

For $\tilde{m}_K$ we make the (numerically justified) ansatz

$$\tilde{m}_K \simeq \frac{b}{K[\log K]^l}, \tag{4.6}$$

with $l = l(\kappa, K)$ and $b = b(\kappa, K)$ being weakly dependent on K. The final result for the capacity then becomes to leading order

$$\alpha_c^K \simeq \frac{1}{b}[\log(K)]^{l-1} \equiv \tilde{c}[\log(K)]^{\tilde{n}}, \tag{4.7}$$

where the exponent $\tilde{n}$ (and the prefactor $\tilde{c}$) are now purely determined from the error calculation avoiding the slowly vanishing terms in the perceptron capacity (4.1) for finite κ. For the variants of the upstart algorithm similar considerations can be

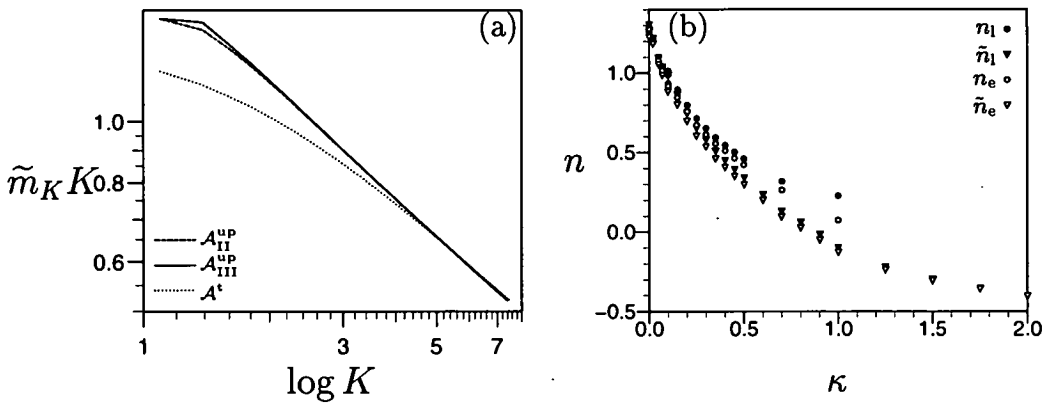

Figure 4.12. (a) The power-law relationship between $\widetilde{m}_K K$ and $\log K$ (4.6) is shown to hold approximately for $\kappa = 2$ and RS (and all constructive algorithms). Similar results can be obtained for other stabilities and 1RSB. (b) The exponent values extracted either from the capacity curve itself (n) or from the bias on the last unit ($\tilde{n}$) are very similar for small stability confirming that the two methods are equally suited in this regime. For larger stability the two approaches differ mainly due to the slowly decaying corrections to the asymptotic capacity limit of the last perceptron for $\widetilde{m}_o \to 0$ (see the text).

applied, leading asymptotically to the same equation for the exponent n, but with a prefactor which asymptotically depends on the initial output-distribution bias, the exact derivation of which can be found in Appendix 4.C.

The adequacy of this approach is scrutinized in Figure 4.12(a), where Eq. (4.6) is shown to hold well for $\kappa = 2$ and RS, by plotting $\widetilde{m}_K K$ vs. $\log K$ on a log-log scale for all considered algorithms. In Figure 4.12(b), the differences between the RS power-law exponent estimates resulting from the capacity curve itself (n) and indirectly via the bias on the last unit ($\tilde{n}$) are depicted.

For small stability κ, the two estimates almost coincide. The small deviations can be explained by two factors. First, higher order terms in the expansion of the last perceptron's capacity limit have been neglected in (4.5). Second, the indirect calculation of the exponent value suffers from some systematic errors in the case of the upstart algorithm. For small finite stabilities the capacity limit can be reached purely due to a change in architecture without either units ever being very close to their respective saturation limits. Furthermore, the capacity as a function of $\widetilde{m}_K$ has further corrections which only strictly vanish for $\widetilde{m}_K = 0$ (see Appendix 4.C). The reevaluated exponents are explicitly listed in Table 4.3 for $\kappa = 0$ to allow a comparison with the original values Table 4.2. Studying both tables, the largest systematic differences

can be found in the case of the tiling-like algorithm[21], whereas the deviations for the upstart algorithm are very small, which may be explained by the systematic corrections cancelling neglected higher order terms in the capacity. Again, these differences show, that we cannot expect quantitatively accurate results.

Returning to Figure 4.12(b), for large stability κ, the n estimates differ significantly between the calculation methods. The difference approximately corresponds to the expected correction from neglecting the slowly decaying systematic shifts of the last perceptrons asymptotic ($\widetilde{m}_K \to 0$) capacity for finite κ. For large stabilities, where the slope of the raw capacity curves becomes very small or even changes sign, any reliable exponent cannot be determined from the capacity curve itself, which can be seen by the diverging n_e and n_l estimates for $\kappa = 1$. Below, we will therefore use the indirect method of determining estimates for n. Unfortunately, $\tilde{n}_\mathrm{l}$ is not a local estimate of n and therefore cannot be compared to the raw capacity curve in Section 4.4.

Keeping these restrictions in mind, the behaviour of the power-law exponent estimates $\tilde{n}_\mathrm{l}$ and $\tilde{n}_\mathrm{e}$ as a function of the stability κ is shown in Figure 4.13 for all considered constructive algorithms within the RS [Figure 4.13(a)] and 1RSB [Figure 4.13(b)] ansatz. In all cases the behaviour of $\tilde{n}$ mirrors the qualitative observations made from Figures 4.5 and 4.6; the exponent estimates of n decrease monotonically for increasing

[21]The difference between n and $\tilde{n}$ of around 0.08 to 0.1 correspond almost exactly to the expected finite $\widetilde{m}_\mathrm{o}$ corrections to the asymptotic perceptron capacity limit for $\kappa = 0$ {of $\mathcal{O}(\log[\log(\widetilde{m}_\mathrm{o}^{-1})])$}.

Table 4.3. The reestimated power-law exponents $\tilde{n}_\mathrm{l}$ and $\tilde{n}_\mathrm{e}$ for $\kappa = 0$, the considered algorithms, and several replica ansätze for two values of K in order to highlight the occurance of systematic errors.

$\mathcal{A}$	K^*	RSB		RS		IRSB	
		$\tilde{n}_\mathrm{l}$	$\tilde{n}_\mathrm{e}$	$\tilde{n}_\mathrm{l}$	$\tilde{n}_\mathrm{e}$	$\tilde{n}_\mathrm{l}$	$\tilde{n}_\mathrm{e}$
$\mathcal{A}_\mathrm{II}^\mathrm{up}$	300	0.583(1)	0.49(2)	1.305(0)	1.271(5)	−0.014(4)	−0.112(6)
	750	0.495(1)	0.36(2)	1.258(0)	1.225(0)	−0.070(0)	−0.084(1)
$\mathcal{A}_\mathrm{III}^\mathrm{up}$	300	0.622(2)	0.53(2)	1.312(0)	1.276(6)	0.035(2)	−0.035(17)
	450	0.571(1)	0.47(1)	1.259(0)	1.223(1)	−0.004(1)	−0.064(2)
$\mathcal{A}^\mathrm{t}$	300	0.367(1)	0.29(1)	1.079(0)	1.070(0)	−0.128(0)	−0.152(1)
	450	0.330(1)	0.23(2)	1.062(0)	1.049(0)	−0.136(0)	−0.147(1)

*The numbers given for K apply to 1RSB in the case of the spherical perceptron. For RS $K = 1000$ and $K = 4000$ were used as before and for the Ising perceptron the larger network sizes were $K = 1000$ for $\mathcal{A}_\mathrm{II}^\mathrm{up}$ and $K = 500$ for $\mathcal{A}_\mathrm{III}^\mathrm{up}$ and $\mathcal{A}^\mathrm{t}$.

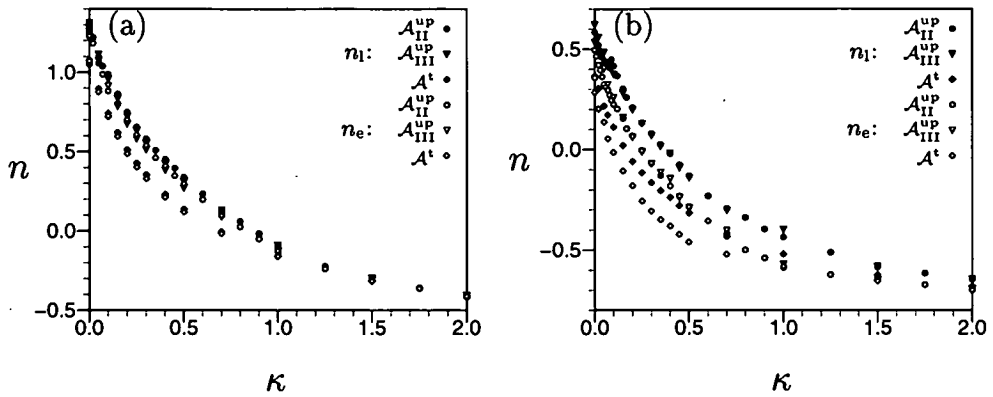

Figure 4.13. The power-law exponent estimates $\tilde{n}_l$ and $\tilde{n}_e$ for the capacity limit $\alpha_c^K \propto [\log K]^n$ is shown as a function of the stability κ for (a) RS and (b) 1RSB for the various algorithms [upstart II ($\mathcal{A}_{II}^{up}$), upstart III ($\mathcal{A}_{III}^{up}$), and tiling-like ($\mathcal{A}^t$)]. The local values $\tilde{n}_l$ were determined for $K = 1000$ for RS and for $150 \leq K \leq 300$ for 1RSB and are denoted by filled symbols, whereas the extrapolation estimate $\tilde{n}_e$ are represented by open symbols (see the legend). The estimation error for $\tilde{n}_l$ and $\tilde{n}_e$ does roughly not exceed more than three times the size of the symbols, and is about their size in many cases especially for RS or 1RSB and very large κ.

stability κ and above a critical stability κ_c, the slope becomes negative. For large stabilities, the power-law exponent estimates seem to converge to a finite limit $\tilde{n}(\kappa \to \infty)$. It is a very interesting question, whether the shape of the functional dependence of n on κ is preserved for $K \to \infty$.

Within the RS ansatz, depicted in Figure 4.13(a), the estimated exponent $\tilde{n} > 1$ for small stabilities and all algorithms suggesting that the MD bound is violated. This is in contrast to the original work for the tiling-like algorithm (Biehl and Opper 1991), where $n = 1$ was reported within the RS ansatz based on smaller network sizes. However, it is not entirely clear whether $n > 1$ actually holds asymptotically. It is worth mentioning that the simple RS treatment in the Gardner-volume calculation for the committee machine (Barkai et al. 1992; Engel et al. 1992) predicts an asymptotic capacity limit proportional to $\sqrt{K}$ instead of the correct $\sqrt{\log K}$ (Monasson and Zecchina 1996; Urbanczik 1997; Xiong et al. 1997), i.e., predicts K rather than $\log K$ as the relevant quantity. The RS ansatz in the present case seems to lead at least to the correct scaling.

For the 1RSB ansatz shown in Figure 4.13(b), the picture is similar, with the noticeable difference of a shift of n such that $n < 1$ for $\kappa = 0$. Note that for 1RSB the errors for the individual exponent estimates are much larger since $150 \leq K \leq 300$ instead of $K > 1000$ for RS. Especially large error bars are obtained for $\kappa \approx 0.1$

and upstart II, where the kink in the curve can be explained by the phase transition at $100 \leq K \leq 300$. Note that for upstart III the estimates of n [see also both Table 4.2 and 4.3] are close to $\frac{1}{2}$, which would suggest that networks constructed by the upstart III algorithm have a similar performance as the committee machine. Since the committee machine uses unconstrained optimization of the internal representations, this may be seen as a further indication that the available n estimates are significantly too large.

It would be further interesting to compare the performance of the three algorithms. Although, we find that the $\tilde{n}$ estimates somewhat mirror the observations made from the visual inspection of the capacity curves in Section 4.4.1, a significant difference is that the tiling-like algorithm is estimated to perform asymptotically worse than both upstart versions. However, due to the larger finite size effects and the systematic errors for the exponents of the upstart algorithm, this may not hold asymptotically. Furthermore, this calculation does not account for the influence of correlations between the perceptrons on n. We expect these corrections to be larger for the upstart algorithm.

Note that the difference between $\tilde{n}_\mathrm{l}$ and $\tilde{n}_\mathrm{e}$ approximately decreases monotonically with κ, although K was at least decreased for 1RSB from $K = 300$ for $\kappa = 0$ to $K = 150$ for $\kappa = 2$. We have identified two possible causes. First, the n itself may have a finite limit $n(\kappa \to \infty)$. Second, for large stability the asymptotic error regime of the individual perceptrons (where the error approaches the asymptotic error in a simple power law with logarithmic corrections) is reached faster, which may dampen higher order corrections to the measured power law in $\widetilde{m}_K$.

In conclusion, we believe that it is plausible to assume that $\tilde{n}_\mathrm{l}$ as well as $\tilde{n}_\mathrm{e}$ constitute practical upper bounds for $n(\kappa)$. For smaller stabilities significant corrections are to be expected as has already been highlighted by the inconsistency found for non-zero bias. For increasing stabilities, these bounds become arguably tighter.

4.5.4 Analysis for the Ising perceptron

In this section, we briefly assess whether the observations made for the spherical perceptron also hold for the Ising perceptron. In Figure 4.14 both estimates of the power-law exponent, $\tilde{n}_\mathrm{l}$ and $\tilde{n}_\mathrm{e}$ for upstart II, show the same characteristic decay to a finite limit for large stability. The main difference is that the curves are shifted to much smaller n values. In fact for all algorithms (see also Table 4.3) both $\tilde{n}_\mathrm{l}$ and $\tilde{n}_\mathrm{e}$ are predicted to be slightly smaller than 0 for large enough K, i.e., asymptotically the capacity is expected to decrease for all stabilities.

For the Ising perceptron, the error in the $\tilde{n}(\kappa)$ estimates is especially large for $\kappa \approx 0.08$, where the kink in the capacity for relatively large K makes the measurements of $\tilde{n}$ more difficult. For large κ, the non-monotonic behaviour may be an artefact caused

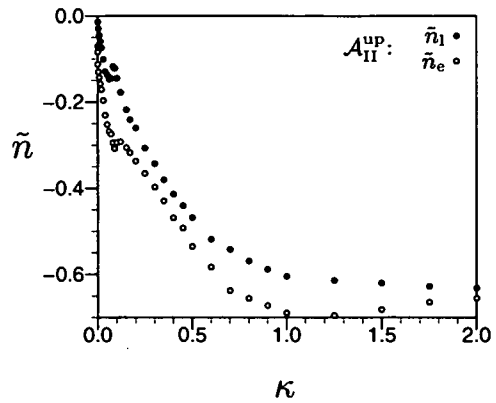

Figure 4.14. For upstart II and the uncoupled 1RSB ansatz, employing the Ising perceptron as the basic building block results in $\tilde{n}_{\rm l}$ and $\tilde{n}_{\rm e}$ being shifted to smaller values for all stabilities κ. The local values $\tilde{n}_{\rm l}$ were determined for $150 \leq K \leq 300$ and are denoted by filled circles, whereas their extrapolation $\tilde{n}_{\rm e}$ are represented by open circles (see the legend).

by the decrease of K with κ.

4.6 Summary and conclusions

The main thrust of this chapter has been the study of the capacity limit of MLPs built by constructive algorithms. Here we have concentrated on two algorithms, a tiling-like algorithm ($\mathcal{A}^{\rm t}$) for a parity machine (Biehl and Opper 1991), inspired by the tiling algorithm (Mézard and Nadal 1989), and variants of the upstart algorithm (Frean 1990a) ($\mathcal{A}_{\rm IIa}^{\rm up}$, $\mathcal{A}_{\rm IIb}^{\rm up}$, $\mathcal{A}_{\rm IIIa}^{\rm up}$, and $\mathcal{A}_{\rm IIIb}^{\rm up}$) which are accessible to a statistical mechanics framework. The variants of the upstart algorithm differ in the make-up of the training set (II,III) and in the selection criteria (a,b) used for the creation of new units, allowing us to assess the impact of small changes in the algorithm to the resulting capacity.

In order to calculate their capacity explicitly, the approximation has been made that the quenched average over the training sets can be taken separately for each perceptron generated by the algorithms, effectively assuming that the perceptrons are uncorrelated. This approximation allows the capacity being calculated employing results for simple perceptrons derived within a replica framework. The validity of this "uncoupled" approximation has been assessed within a RS ansatz for the case of two perceptrons being generated in successive steps of the algorithms. The corrections to the capacity (and the errors made above saturation) due to the correlations turn out to be in most cases negligible in comparison to the effect of RSB in the individual perceptrons, suggesting that the results derived are a good approximation.

For the case of zero stability and unbiased output distributions for which exact asymptotic capacity results are known for the parity ($\alpha_c^K \propto \log K$) (Barkai et al. 1990) and the committee machine ($\alpha_c^K \propto \sqrt{\log K}$) (Monasson and Zecchina 1996; Urbanczik 1997; Xiong et al. 1997), we find that all constructive algorithm considered are likely to exhibit a power-law behaviour[22] in $\log K$, $\alpha_c^K \simeq c[\log K]^n$. The power-law ansatz modifies preliminary results (West and Saad 1997) and earlier work (Biehl and Opper 1991).

Visual inspection of the capacity curves suggest that the prefactor c is dependent on the constructive algorithm employed. For a more objective analysis, local estimates of the exponent $n(K)$ for finite K have been obtained, providing a reasonable fit for the capacity curves and then extrapolated to $1/K \to 0$. However, two sources for systematic errors were identified. First and probably most important, due to the small range of $\log K$ values accessible to such an approach, significant finite K effects were encountered. A reestimation of n using larger networks shows that the n estimates systematically shift to smaller values. Furthermore, the K values explored are too small to make the extrapolation accurate — the local estimate of n at a larger K value were often found to be significantly smaller than the extrapolation from smaller K values to $1/K = 0$. Second, further RSB breaking and the ignored correlation between perceptrons should lead to systematic shifts of n to smaller values — although the magnitude of these corrections is unknown, they should be significantly smaller than the corrections going from the uncoupled RS to 1RSB. In summary, the exponent values cannot be estimated reliably enough to decide whether n is algorithm dependent. Nevertheless, the extracted exponents may still provide practical upper bounds for the true exponents since the extrapolated values were in all cases smaller than their corresponding local estimates.

Within these restrictions, the exponent n for the RS ansatz has been estimated to be greater than 1 for all constructive algorithms within the K values studied, which would violate the MD bound (Mitchison and Durbin 1989). If this violation holds asymptotically, it should be compared to the failure of the RS ansatz in fixed architecture cases (Barkai et al. 1992; Engel et al. 1992), where power-laws in K instead of $\log K$ are predicted.

For the uncoupled 1RSB ansatz, we find $0.23 \lessapprox n \lessapprox 0.47$ for various estimation methods and constructive algorithms. Especially the result for the upstart III algorithm predicting values close to $\frac{1}{2}$ (the exponent for the committee machine using

[22] Although it may be argued that the size of K explored does not allow us to rule out any other functional ansätze, a power-law was the only simple functional form which held reasonably across all cases studied.

unconstrained optimization) seems to confirm that these finite K predictions are too optimistic.

This work has furthermore extended the study of the capacity limit to finite stabilities and biased output distributions, issues which, to our knowledge, have not been addressed previously for multilayer networks. In both cases, the most reasonable functional form of the capacity limit remains $\alpha_c^K \simeq c[\log K]^n$ for large K and the constructive algorithms studied.

In the case of biased output distributions (but zero stability), the limitations of the validity of the extracted exponents have been made more apparent as no consistent n values could be determined. Theoretically n must either increase or remain constant for $m_0 \to 1$; our numerical results, however, suggest otherwise. This contradiction may be explained by the fact that networks are pushed further into the asymptotic error and capacity regimes for increasing bias, increasing the "effective" K value. The estimated exponents for large m_0 may therefore provide a tighter upper bound for the true exponents than could be achieved for zero output bias. It remains an interesting open question whether the true exponent n is a function of m_0 or a constant. Constant n implies that the exponent must also be independent of the constructive algorithms studied since it can be shown that they become equivalent in the $m_0 \to 1$ limit. A visual inspection for finite K values relevant in practice reveals some performance difference between the algorithms for small but finite bias. The tiling-like algorithm outperforms both upstart variants, which is partly due to the fact that only suboptimal unit creation selection criteria could be identified for the upstart algorithm.

For finite stability κ but unbiased output distributions, we find that the $n(\kappa)$ estimates decay monotonically in κ for all algorithms to finite limits $n(\kappa \to \infty)$. For both RS and 1RSB, we find that for stabilities beyond a "critical" stability κ_c (defined by a K independent constant capacity for large K) the capacity (per network weight) decreases asymptotically (as a function of K). This effect has also been observed from a visual inspection of the capacity curves, however, the analysis suggests that the critical stability may asymptotically be much smaller than anticipated from the curves themselves. For the Ising perceptron (and 1RSB), the n estimates are smaller than 0 for all stabilities, whereas numerically we find this transition for small but finite stability for the K values explored.

In all cases, it is of considerable interest whether the true exponent n is dependent on the stability κ. Within the limitations of our analysis this question cannot be answered with certainty. However, it may be argued that, although the n estimates themselves are not accurate, the generic shape of the dependence between n estimates and κ has consistently carried over across different perceptron weight models and replica ansätze,

making a stability dependent exponent a reasonable conjecture.

Of further interest is whether the exponent n is dependent on the constructive algorithm employed analogous to the functional dependence of n on the architecture type found for conventional networks. Comparing the various constructive algorithms for all κ, the estimates predict consistently $n(\mathcal{A}_{\mathrm{III}}^{\mathrm{up}}) > n(\mathcal{A}_{\mathrm{II}}^{\mathrm{up}}) > n(\mathcal{A}^{\mathrm{t}})$, suggesting that the upstart algorithm is asymptotically more efficient. However, this result completely neglects the issue of the size of systematic errors in the n estimates, which are considerably larger for the upstart algorithm. Especially for small stability, the difference between the n estimates for the tiling-like and upstart algorithms seem grossly exaggerated considering the numerical results. The above discussion of this issue for large bias gives also some credit to the conjecture that the performance difference between (the considered) constructive algorithms may lead asymptotically only to algorithm (and stability) dependent prefactors.

For the prefactors, the numerical capacity results predict, that upstart III always outperforms upstart II. This performance difference is caused by the fact that the design of upstart III allows for the elimination of more training patterns from the training set of units constructed consecutively (although this difference vanishes for $\kappa \to \infty$). The comparison between upstart and tiling-like algorithm is less straightforward. For small stability the tiling-like capacity remains above both upstart capacities for all K values and calculation ansätze (RS, RSB, IRSB) studied, although the upstart III capacity closes the gap for increasing K. For larger stability, both upstart algorithms exhibit a higher capacity than the tiling-like algorithm for large enough K. This behaviour reflects two competing effects. The design of the upstart algorithm includes pattern elimination for latter units (increasing its efficiency) but also features the creation of specialized units correcting only one error type (decreasing its efficiency).

In conclusion, we believe it is reasonable to assume that all considered constructive algorithm use their hidden units less efficient than a fixed architecture multilayer network with unconstrained optimization. This is due to the fact that the constructive algorithm use their hidden units to overrule previous decision and can therefore explore only a much smaller space of internal representation than a general two-layer network.

It would be very desirable to investigate the effect of finite stability and non-zero output distribution bias for fixed architectures where the $K \to \infty$ limit can be taken analytically. Of special interest would be whether the exponent of the power-law is actually dependent on the stability κ and/or the bias m_0 of the output distribution — two issues which have been addressed but could not be answered with sufficient certainty in the framework employed in this chapter.

Appendix 4

4.A Replica calculation for two coupled perceptrons

In this appendix, we will briefly outline the replica calculation for two Boolean perceptrons which have been successively constructed by variants of the upstart algorithm or the tiling-like algorithms introduced in Section 4.2 for the case of learning a set of random dichotomies. The calculation is in spirit similar to the single perceptron as calculated in Chapter 3. We have restricted ourselves to the case of real valued weights and a spherical constraint and the replica symmetric (RS) ansatz for simplicity, although the calculation is in principle also extendible to one-step replica symmetry breaking (RSB) or/and to binary $\{-1,1\}$ weights (Ising constraint).

4.A.1 Free energies of the coupled perceptrons

As mentioned in Section 3.2.1, the task of the learner in the capacity problem is to implement a given set of $p = \alpha N$ random dichotomies $(\boldsymbol{\xi}^\mu, \zeta^\mu)$, where the inputs $\boldsymbol{\xi}^\mu$ and outputs ζ^μ are taken from the distributions (3.1). In general, it is useful to extend this capacity problem beyond saturation, where the learner cannot implement all examples but has to misclassify some of them. The aim of training is then to minimize the training error, which is given by summing over a suitable *cost function* for each example. Below, we will investigate the minimal error and the capacity achievable for a learner consisting of two perceptrons created consecutively by the constructive algorithms in question and for cost functions which penalize the number of misclassifications.

The first perceptron with parameters $\boldsymbol{\Omega}_1 = \{\boldsymbol{W}_1, \theta_1\}$, performing the mapping in Eq. (3.2), aims to minimize the number of misclassifications irrespective of the constructive algorithm and is therefore trained on the error function equivalent to the one introduced in Eq. (3.3)

$$E_1 = \sum_\mu \Theta\left(\kappa - \lambda_1^\mu\right) = \sum_\mu V_1(\lambda_1^\mu, \kappa), \tag{4.A.1}$$

where $\lambda_1^\mu = \zeta^\mu h_1^\mu$ and V_k is the error measure for a single example and has been introduced for convenience for the same reasons as given in Chapter 3: It enables one to carry out most of the calculation without specifying a particular error function and also allows the introduction of the auxiliary term $[\epsilon^- \Theta(\zeta^\mu) + \epsilon^+ \Theta(-\zeta^\mu)]$ picking out

the wrongly-on and wrongly-off errors when taking derivatives with respect to ϵ^+ or ϵ^-.

The inputs $\boldsymbol{\xi}^\nu$ for the training set of the second perceptron, with parameters $\boldsymbol{\Omega}_2 = \{\boldsymbol{W}_2, \theta_2\}$, are a subset of the original inputs while the targets ζ^ν are defined according to the rules of the specific constructive algorithm and depend on the original target and the output of the first perceptron. For the tiling-like algorithm, the training set consists of all previous inputs and the target is $+1$ for correctly and -1 for erroneously classified patterns. The error function to be minimized is therefore

$$E_2^{\mathrm{t}} = \sum_\mu \Theta(\kappa - \lambda_1^\mu)\,\Theta(\kappa + \lambda_2^\mu) + [1 - \Theta(\kappa - \lambda_1^\mu)]\,\Theta(\kappa - \lambda_2^\mu) \tag{4.A.2a}$$

$$= \sum_\mu V_2^{\mathrm{t}}(\lambda_1^\mu, \lambda_2^\mu, \kappa), \tag{4.A.2b}$$

where $\lambda_2^\mu = h_2^\mu$ and V_2^{t} is the generic error measure.

The error function for the upstart algorithm depends on the variant used and whether the second unit is constructed to correct wrongly-off or wrongly-on errors. However, it is self-evident by symmetry arguments that one has only to investigate the case of wrongly-off error correction: the result for wrongly-on error correction can be obtained by flipping the random output target $\zeta^\mu \to -\zeta^\mu$, which corresponds to changing the sign of the output distribution bias $m_{\mathrm{o}} \to -m_{\mathrm{o}}$. Following similar considerations as above concerning the training set and targets of the daughter unit (see Table 4.1), one finds

$$E_2^{\mathrm{up},\gamma} = \sum_\mu \Theta(\zeta^\mu)\Theta(\kappa - \lambda_1^\mu)\,\Theta(\kappa - \lambda_2^\mu) + \Theta(-\zeta^\mu)\,[1 + (\gamma - 1)\Theta(\kappa - \lambda_1^\mu)]\,\Theta(\kappa + \lambda_2^\mu) \tag{4.A.3a}$$

$$= \sum_\mu V_2^{\mathrm{up},\gamma}(\lambda_1^\mu, \lambda_2^\mu, \kappa), \tag{4.A.3b}$$

where γ switches between upstart II ($\gamma = 1$) and upstart III ($\gamma = 0$).

The total training error function therefore becomes for either algorithm using the generic E_2 for the second perceptron

$$E = E_1 + \delta E_2, \tag{4.A.4}$$

where we have introduced a weighting factor $\delta = \beta_2/\beta$ in the total energy where β_2 acts as a "quasi" temperature for the second perceptron in the total energy function. This is necessary because the minimization of the error is not unconstrained, i.e., the weights of the first perceptron are trained first and subsequently frozen during the training of

the second perceptron.

As in Chapter 3, the calculation will be performed in the thermodynamic limit $N \to \infty$ with finite example load $\alpha = p/N$, where the free energy per input $N\beta f = \log Z$ is assumed to be self-averaging. Again, we will only consider zero-temperature Gibbs learning for both perceptrons, as we are interested only in the minimum error possible. The separation of the training is achieved by first taking the $\delta \to 0$ limit, keeping terms up to $\mathcal{O}(\delta)$, and subsequently taking the $\beta \to \infty$ limit[23]. The free energy splits into two parts as a consequence: f_1, which is of $\mathcal{O}(1)$, and represents the free energy of the first perceptron and f_2, which is of $\mathcal{O}(\delta) = \mathcal{O}(\beta_2/\beta)$, and is the free energy of the second perceptron. Hence

$$\begin{aligned} \langle\langle f \rangle\rangle = \langle\langle f_1 + \delta f_2 \rangle\rangle &= -\lim_{\beta\to\infty}\lim_{\delta\to 0}\lim_{N\to\infty} \frac{1}{N\beta} \langle\langle \log Z \rangle\rangle \qquad (4.A.5) \\ &= -\lim_{\beta\to\infty}\lim_{\delta\to 0}\lim_{N\to\infty} \frac{1}{N\beta} \left\langle\!\!\left\langle \log \int d\mu(\boldsymbol{W}_2) d\mu(\boldsymbol{W}_1) \exp\left[-\beta\left(E_1 + \delta E_2\right)\right] \right\rangle\!\!\right\rangle \end{aligned}$$

where $\langle\langle \cdot \rangle\rangle$ is the quenched average over the distribution of patterns, consisting of integrations over the biased input and output distributions (3.1).

Similar to Chapter 3, we enforce spherical constraints on the weight vectors $d\mu(\boldsymbol{W}_i) = \delta(\boldsymbol{W}_i \cdot \boldsymbol{W}_i - N) \prod_{k=1}^{N} dW_k$ to avoid the invariance $(\boldsymbol{W}_k, \kappa) \to (\lambda \boldsymbol{W}_k, \lambda\kappa)$ and perform the quenched average using the replica trick (3.9). Note, that we treat the two perceptrons as one physical system with parameters $\boldsymbol{\Omega} = \{\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2\}$ when replicating the partition function. We apply the same standard techniques as in Section 3.3.1 including the introduction of order parameters for the single perceptrons and order parameters describing the cross-overlaps between the two perceptrons

$$Q_i^{\sigma\rho} = \frac{1}{N} \boldsymbol{W}_i^{\sigma} \cdot \boldsymbol{W}_i^{\rho} \qquad (\text{for } \forall\ \sigma < \rho), \qquad (4.A.6a)$$

$$M_i^{\sigma} = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} W_{i_k}^{\sigma} \qquad (\text{for } \forall\ \sigma), \qquad (4.A.6b)$$

$$R^{\sigma\rho} = \frac{1}{N} \boldsymbol{W}_1^{\sigma} \cdot \boldsymbol{W}_2^{\rho} \qquad (\text{for } \forall\ \sigma, \rho), \qquad (4.A.6c)$$

with their Lagrange multipliers $\hat{Q}_i^{\sigma\rho}$, $\hat{M}_i^{\sigma}$, $\hat{R}^{\sigma\rho}$ and the Lagrange multiplier $\hat{E}_i^{\sigma}$ associated with the spherical constraints[24]. After some algebra, the replicated partition

[23] Alternatively, one can see this procedure as taking the $\beta \to \infty$ with β_2 constant and then taking the $\beta_2 \to \infty$ limit.

[24] The contributions of $\hat{M}_i^{\sigma}$ vanish in the thermodynamic limit similar to single perceptron calculations.

function becomes

$$\langle\!\langle Z^n \rangle\!\rangle = \int_{-\infty}^{\infty} \int_{-\mathrm{i}\infty}^{\mathrm{i}\infty} \left(\prod_{i=1}^{2} \prod_{\sigma} \frac{dM_i^\sigma d\hat{E}_i^\sigma}{2\pi} \right) \left(\prod_{i=1}^{2} \prod_{\sigma<\rho} \frac{dQ_i^{\sigma\rho} d\hat{Q}_i^{\sigma\rho}}{2\pi} \right) \left(\prod_{\sigma,\rho} \frac{dR^{\sigma\rho} d\hat{R}^{\sigma\rho}}{2\pi} \right)$$
$$\times \exp\left\{ N \left[G_0(\hat{Q}_i^{\sigma\rho}, \hat{E}_i^\sigma, \hat{R}^{\sigma\rho}) + \alpha G_{\mathrm{r}}(Q_i^{\sigma\rho}, \theta_i^\sigma, M_i^\sigma, R^{\sigma\rho}) \right.\right.$$
$$\left.\left. + \frac{1}{2} \sum_{\sigma} \hat{E}_i^\sigma - \sum_{\sigma<\rho} Q_i^{\sigma\rho} \hat{Q}_i^{\sigma\rho} - \sum_{\sigma,\rho} R^{\sigma\rho} \hat{R}^{\sigma\rho} \right] \right\}, \quad (4.A.7)$$

where we use the Einstein convention for summations over repeated indices. The two terms in the integral are the prior constraint Hamiltonian,

$$G_0(\hat{Q}_i^{\sigma\rho}, \hat{E}_i^\sigma, \hat{R}^{\sigma\rho}) = \log \left\{ \int_{-\infty}^{\infty} \prod_{\sigma} dW_1^\sigma dW_2^\sigma \exp\left[-\frac{1}{2} \sum_{\sigma} \hat{E}_i^\sigma W_i^\sigma W_i^\sigma \right.\right.$$
$$\left.\left. + \sum_{\sigma<\rho} \hat{Q}_i^{\sigma\rho} W_i^\sigma W_i^\rho + \sum_{\sigma,\rho} \hat{R}^{\sigma\rho} W_1^\sigma W_2^\rho \right] \right\}, \quad (4.A.8a)$$

which depends on the weight prior and the replicated Hamiltonian

$$G_{\mathrm{r}}(Q_i^{\sigma\rho}, \theta_i^\sigma, M_i^\sigma, R^{\sigma\rho}) = \log \left\langle \int_{-\infty}^{\infty} \left(\prod_{i=1}^{2} \prod_{\sigma} \frac{d\lambda_i^\sigma d\hat{\lambda}_i^\sigma}{2\pi} \right) \right.$$
$$\times \exp\left\{ -\beta \left[V_1(\lambda_1^\sigma, \kappa) + \delta V_2(\lambda_1^\sigma, \lambda_2^\sigma, \kappa) \right] - \mathrm{i} \sum_{\sigma} \hat{\lambda}_i^\sigma \lambda_i^\sigma \right.$$
$$- \mathrm{i} \sum_{\sigma} \left[\zeta \hat{\lambda}_1^\sigma (\theta_1^\sigma - m_{\mathrm{i}} M_1^\sigma) + \hat{\lambda}_2^\sigma (\theta_2^\sigma - m_{\mathrm{i}} M_2^\sigma) \right]$$
$$\left.\left. - \frac{1}{2}(1 - m_{\mathrm{i}}^2) \left[\sum_{\sigma} \hat{\lambda}_i^\sigma \hat{\lambda}_i^\sigma + 2 \sum_{\sigma<\rho} \hat{\lambda}_i^\sigma \hat{\lambda}_i^\rho Q_i^{\sigma\rho} + 2\zeta \sum_{\sigma,\rho} \hat{\lambda}_1^\sigma \hat{\lambda}_2^\rho R^{\sigma\rho} \right] \right\} \right\rangle_\zeta, \quad (4.A.8b)$$

where $\langle \cdot \rangle_\zeta$ denotes an average over the output distribution.

4.A.2 The replica symmetric ansatz

To make further progress we have to make an assumption for the structure of the replica space as in Chapter 3. Here, we will only pursue the simplest replica symmetric (RS)

ansatz, which assumes

$$
\begin{aligned}
& & Q_i^{\sigma\rho} &= q_i \quad \text{and} \quad \hat{Q}_i^{\sigma\rho} = \hat{q}_i \quad (\text{for } \forall\, i \text{ and } \sigma < \rho), \\
M_i^\sigma = M_i, \quad & & \theta_i^\sigma &= \theta_i, \quad \text{and} \quad \hat{E}_i^\sigma = \hat{E}_i \quad (\text{for } \forall\, \sigma), \\
& & R^{\sigma\sigma} &= r \quad \text{and} \quad \hat{R}^{\sigma\sigma} = \hat{r} \quad (\text{for } \forall\, \sigma), \\
& & R^{\sigma\rho} &= s \quad \text{and} \quad \hat{R}^{\sigma\rho} = \hat{s} \quad (\text{for } \forall\, \sigma \neq \rho).
\end{aligned}
\tag{4.A.9}
$$

The physical interpretation of q_i is the same as for a single perceptron calculation: the typical overlap of two solutions within the version space of the individual perceptrons. The overlap s and r both describe the overlap between the two perceptrons, but r describes the overlap of the second perceptron with the first perceptron on whose errors it has been trained, whereas s describes the overlap of the second perceptron with any other first perceptron from the version space.

We note that replica symmetry is broken in this scenario. However, the aim of this calculation is to assess whether the effect of coupling two perceptrons in a capacity calculation is stronger than that of replica symmetry breaking in the individual perceptrons. A 1RSB calculation would result in 4-dimensional integrals, which are difficult to evaluate numerically accurate enough to find solutions to the saddlepoint equations.

Inserting the above ansätze into Eqs. (4.A.8a) and (4.A.8b) and taking the $n \to 0$ limit yields

$$
\begin{aligned}
G_0^{\mathrm{RS}} &= \frac{1}{2}\frac{(\hat{E}_1+\hat{q}_1)\hat{q}_2 + (\hat{E}_2+\hat{q}_2)\hat{q}_1 + 2\hat{s}(\hat{r}-\hat{s})}{(\hat{E}_1+\hat{q}_1)(\hat{E}_2+\hat{q}_2) - (\hat{r}-\hat{s})^2} \\
&\quad -\frac{1}{2}\log\left[(\hat{E}_1+\hat{q}_1)(\hat{E}_2+\hat{q}_2) - (\hat{r}-\hat{s})^2\right] \qquad (4.A.10a) \\
G_{\mathrm{r}}^{\mathrm{RS}} &= \left\langle \int D\mu(t_1,t_2)\, \log\left[\mathcal{F}_{\mathrm{RS}}(t,\beta,\kappa,\zeta,q_i,\theta_i,r,s)\right] \right\rangle_\zeta , \qquad (4.A.10b)
\end{aligned}
$$

where all integrals without explicit limits are from $-\infty$ to $+\infty$. The measure $D\mu(t_1,t_2)$ is given by

$$
D\mu(t_1,t_2) = \frac{dt_1 dt_2}{2\pi\sqrt{q_1q_2-s^2}} \exp\left[-\frac{1}{2(q_1q_2-s^2)}\begin{pmatrix}t_1\\t_2\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}q_2 & -\zeta s\\ -\zeta s & q_1\end{pmatrix}\begin{pmatrix}t_1\\t_2\end{pmatrix}\right], \tag{4.A.11}
$$

and the function $\mathcal{F}_{\rm RS}$ is given by

$$\mathcal{F}_{\rm RS}(t,\beta,\kappa,\zeta,q_i,\theta_i,r,s) = \int \frac{d\lambda_1 d\lambda_2}{2\pi(1-m_i^2)\sqrt{(1-q_1)(1-q_2)-(r-s)^2}} \times \exp\left\{-\beta\left[V_1(\lambda_1,\kappa)+\delta V_2(\lambda_1,\lambda_2,\kappa) + \frac{1}{2}\frac{x_2(\psi_1+t_1)^2+\delta x_1(\psi_2^+ + t_2)^2 - 2\delta\zeta z(\psi_1+t_1)(\psi_2^+ + t_2)}{x_1x_2-\delta z^2}\right]\right\}, \quad (4.A.12)$$

where $x_1 = \beta(1-q_1)$, $x_2 = \beta_2(1-q_2)$, $z = \beta(r-s)$, and

$$\psi_1(\lambda_1) = \frac{\lambda_1 + \zeta(\theta_1 - m_i M_1)}{\sqrt{1-m_i^2}} \quad \text{and} \quad \psi_2^{\pm}(\lambda_2) = \frac{\lambda_2 \pm (\theta_2 - m_i M_2)}{\sqrt{1-m_i^2}}. \quad (4.A.13)$$

Here, we have introduced the self-consistent scaling ansätze for the order parameters of the two perceptrons when taking the $\delta \to 0$ and $\beta \to \infty$ limits with $\delta\beta = \beta_1 \to \infty$ in order to access the ground states with least errors only: The volume of the individual solution spaces of the two perceptrons above their capacity limits shrink proportional to the applied "temperature", which is β for the first and β_2 for the second perceptron. Since the version space of the first perceptron induces the difference between r and s, $r-s$ should scale with $1/\beta$.

For $\beta \to \infty$, the integrals over λ_1 and λ_2 in (4.A.12) can be calculated by the saddlepoint method; the exponentional is evaluated at $\lambda_1 = \lambda_1^{\rm opt}$ and $\lambda_2 = \lambda_2^{\rm opt}$, where $\lambda_1^{\rm opt}$ and $\lambda_2^{\rm opt}$ jointly minimize the square bracket for given t_1 and t_2. The $\delta \to 0$ limit effectively constraints this minimization as required[25]: the dominant term of $\mathcal{O}(1)$ in (4.A.12) is independent of λ_2 and can therefore only determine $\lambda_1^{\rm opt}$, which optimizes the first perceptron. The inclusion of $\mathcal{O}(\delta)$ terms determines $\lambda_2^{\rm opt}$, which corresponds with the optimization of the second under the constraint of the first perceptron. Whereas the calculation of $\lambda_1^{\rm opt}(t_1)$ is identical for both upstart and tiling-like, $\lambda_2^{\rm opt}(t_1,t_2)$ is determined algorithm dependent. We furthermore eliminate the conjugate order parameters $\hat{q}_i$, $\hat{E}_i$, $\hat{r}$, and $\hat{s}$, keeping only the terms up to $\mathcal{O}(\delta)$.

The free energy f_1 of $\mathcal{O}(1)$, i.e., for the first perceptron, then simplifies to the

[25] It is fairly straightforward but cumbersome to calculate the free energy for $\delta = 1$, i.e., unconstrained minimization, in order to assess the performance degradation due to constraining the optimization in the constructive algorithms. Such a study is, however, beyond the scope of this thesis has set itself.

already known result (3.18)

$$\langle\langle f_1 \rangle\rangle = \alpha \left\langle \int_{-\tau_1}^{\sqrt{2x_1}-\tau_1} Dt \, \frac{(t+\tau_1)^2}{2x_1} + H\left(\sqrt{2x_1} - \tau_1\right) \right\rangle_\zeta - \frac{1}{2x_1}, \tag{4.A.14}$$

where

$$\tau_1 = \psi_1(\kappa) = \frac{\kappa + \zeta(\theta_1 - m_i M_1)}{\sqrt{1 - m_i^2}}. \tag{4.A.15}$$

The free energy f_1 is be evaluated at the saddlepoints with respect to the variables x_1 and θ_1.

The free energy of the second perceptron for the tiling-like algorithm f_2^t simplifies to

$$\begin{aligned}
\langle\langle f_2^t \rangle\rangle = \alpha \Bigg\langle & \int_{\sqrt{2x_1}-\tau_1}^{\infty} dt_1 \left[\int_{\tau_2^- - \sqrt{2x_2}}^{\tau_2^-} dt_2 \mu(t_1, t_2) \, \frac{(t_2 - \tau_2^-)^2}{2x_2} + \int_{-\infty}^{\tau_2^- - \sqrt{2x_2}} dt_2 \mu(t_1, t_2) \right] \\
& + \int_{-\tau_1}^{-\sqrt{2x_1}-\tau_1} dt_1 \left[\int_{-\tilde{\tau}_2^+}^{\sqrt{2x_2}-\tilde{\tau}_2^+} dt_2 \mu(t_1, t_2) \, \frac{(t_2 + \tilde{\tau}_2^+)^2}{2x_2} + \int_{\sqrt{2x_2}-\tilde{\tau}_2^+}^{\infty} dt_2 \mu(t_1, t_2) \right] \\
& + \int_{-\infty}^{-\tau_1} dt_1 \left[\int_{-\tau_2^+}^{\sqrt{2x_2}-\tau_2^+} dt_2 \mu(t_1, t_2) \, \frac{(t_2 + \tau_2^+)^2}{2x_2} + \int_{\sqrt{2x_2}-\tau_2^+}^{\infty} dt_2 \mu(t_1, t_2) \right] \Bigg\rangle_\zeta - \frac{1 - 2\hat{z}r}{2x_2},
\end{aligned} \tag{4.A.16}$$

where the measure $\mu(t_1, t_2)$ is derived from Eq. (4.A.11) by taking the appropriate limits

$$\mu(t_1, t_2) = \frac{1}{2\pi\sqrt{1 - r^2}} \exp\left[-\frac{t_1^2 - 2\zeta r t_1 t_2 + t_2^2}{2(1 - r^2)} \right], \tag{4.A.17}$$

and the following variables were introduced for convenience

$$\tau_2^\pm = \psi_2^\pm(\kappa) = \frac{\kappa \pm (\theta_2 - m_i M_2)}{\sqrt{1 - m_i^2}}, \quad \tilde{\tau}_2^\pm = \tau_2^\pm \mp \zeta \hat{z}(t_1 + \tau_1), \quad \text{and} \quad \hat{z} = \frac{z}{x_1}. \tag{4.A.18}$$

The free energy of the second perceptron for the versions of the upstart algorithm $f_2^{\mathrm{up},\gamma}$

can be simplified similarly

$$\langle\langle f_2^{\mathrm{up},\gamma}\rangle\rangle = \alpha \Bigg\langle \int_{\sqrt{2x_1}-\tau_1}^{\infty} dt_1 \Bigg\{ \Theta(\zeta) \Bigg[\int_{-\tau_2^+}^{\sqrt{2x_2}-\tau_2^+} dt_2 \mu(t_1,t_2) \frac{(t_2+\tau_2^+)^2}{2x_2} + \int_{\sqrt{2x_2}-\tau_2^+}^{\infty} dt_2 \mu(t_1,t_2) \Bigg] + \gamma\Theta(-\zeta) \Bigg[\int_{\tau_2^- -\sqrt{2x_2}}^{\tau_2^-} dt_2 \mu(t_1,t_2) \frac{(t_2-\tau_2^-)^2}{2x_2} + \int_{-\infty}^{\tau_2^- -\sqrt{2x_2}} dt_2 \mu(t_1,t_2) \Bigg] \Bigg\}$$

$$+ \Theta(-\zeta) \Bigg\{ \int_{-\tau_1}^{-\sqrt{2x_1}-\tau_1} dt_1 \Bigg[\int_{\tilde{\tau}_2^- -\sqrt{2x_2}}^{\tilde{\tau}_2^+} dt_2 \mu(t_1,t_2) \frac{(t_2-\tilde{\tau}_2^+)^2}{2x_2} + \int_{-\infty}^{\tilde{\tau}_2^- -\sqrt{2x_2}} dt_2 \mu(t_1,t_2) \Bigg]$$

$$+ \int_{-\infty}^{-\tau_1} dt_1 \Bigg[\int_{\tau_2^- -\sqrt{2x_2}}^{\tau_2^-} dt_2 \mu(t_1,t_2) \frac{(t_2-\tau_2^-)^2}{2x_2} + \int_{-\infty}^{\tau_2^- -\sqrt{2x_2}} dt_2 \mu(t_1,t_2) \Bigg] \Bigg\} \Bigg\rangle_{\zeta} - \frac{1-2\hat{z}r}{2x_2}, \quad (4.A.19)$$

Since the free energy f_2 for either algorithm is only of $\mathcal{O}(\delta)$, it is evaluated with respect to the variables x_2, θ_2, r, and $\hat{z}$, with x_1 and θ_1 fixed by Eq. (4.A.14). The capacity limit α_c (here not normalized with respect to the number of units) of the combination of the two perceptrons can be calculated from the saddlepoint equations by taking the $x_2 \to \infty$ limit, i.e., the second perceptron does not make any errors. Note, that for the upstart calculation this is only a formal capacity limit, since the wrongly-on errors still need to be corrected.

4.A.3 Solutions of the saddlepoint equations

The saddlepoint solutions for the order parameters and the error rates as a function of the normalized example number α were evaluated for the different constructive algorithms and a range of stabilities κ and output biases m_o. For brevity, only the most relevant effects for the purpose of this chapter will be reported graphically, especially the size of the correlations and their impact on the capacity limit (and the error rate) in comparison to the impact of 1RSB in the individual perceptrons.

For the tiling-like algorithm the order parameters r and $\hat{z}$, describing the correlations between the perceptrons, are shown in Figure 4.15 as a function of α for $\kappa = 1$ and various m_o values. For zero output-distribution bias, the correlations are initially identical to zero above the capacity limit, before their magnitude rises abruptly corresponding to $\alpha = \alpha_p$, i.e., at the phase transition of the first perceptron from the zero to the non-zero threshold solution. Consequently for zero stability and zero bias, the

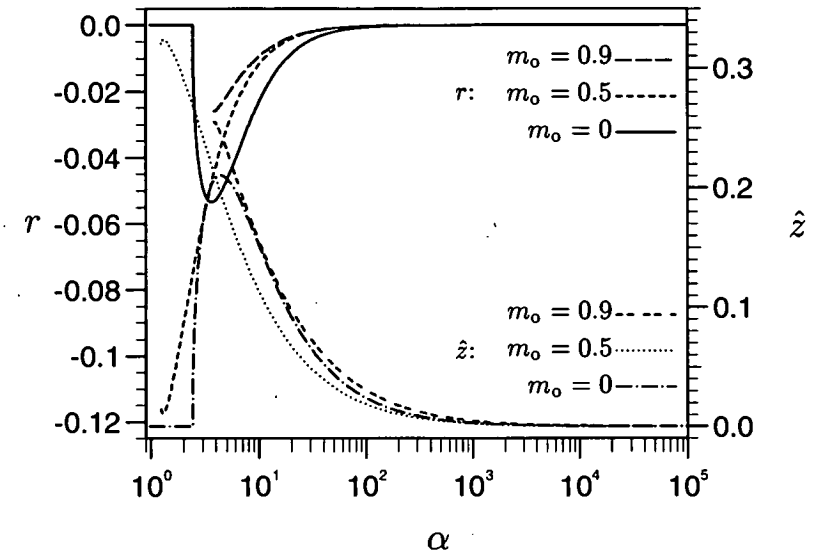

Figure 4.15. The correlations of two consecutive perceptrons created by the tiling-like algorithm are shown via the two order parameters r and $\hat{z}$ as a function of α for $\kappa = 1$ and three bias values (see the legend).

perceptrons remain uncorrelated for all α. After the magnitude of the correlations has passed through a maximum, the order parameters decay to zero asymptotically with power laws whose exponents are approximately

$$r \propto \alpha^{-1.9\pm1} \qquad \text{and} \qquad \hat{z} \propto \alpha^{-0.95\pm5}, \tag{4.A.20}$$

where the error indicates the uncertainty in the last significant digits only. The uncertainty is most likely caused by logarithmic corrections to power laws with theoretical exponents of -2 and -1, respectively.

For non-zero bias, the correlations are largest at the capacity limit and the order parameters decay with the same power laws for $\alpha \to \infty$. Note, that a non-zero overlap r between the perceptrons is always negative, i.e., the perceptrons are anti-correlated. The physical interpretation of the order parameter $\hat{z}$ is less clear. In Figure 4.16 the order parameters r and $\hat{z}$ are shown for the upstart algorithm with either $\gamma = 1$ ($\mathcal{A}_{\mathrm{II}}^{\mathrm{up}}$) or $\gamma = 0$ ($\mathcal{A}_{\mathrm{III}}^{\mathrm{up}}$) and $\kappa = 1$, $m_0 = 0$. In comparison to the tiling-like algorithm, several differences and similarities are remarkable. First, the correlations for the upstart algorithm are always finite for zero bias. Second, one finds $r\hat{z} < 0$ as previously, however, the sign of the order parameters is reversed, i.e., $r > 0$ and the perceptrons are positively correlated. Third, the magnitude of the correlations for both variants of the

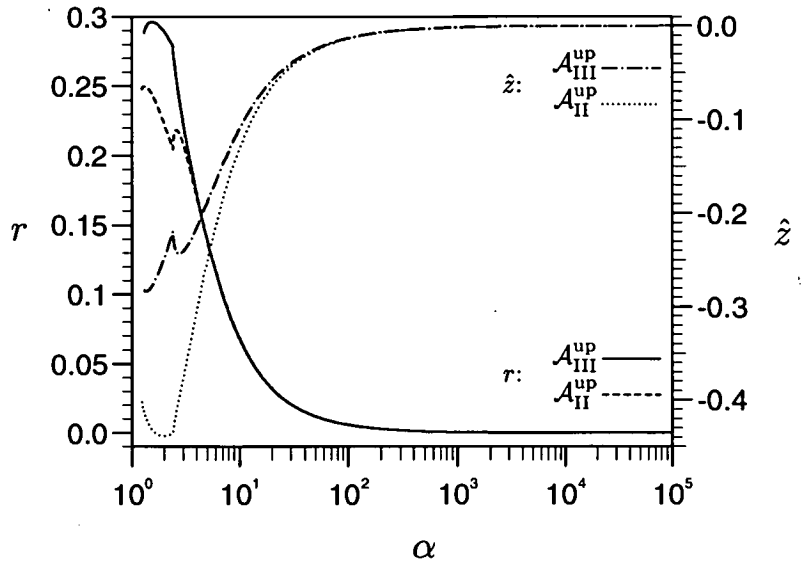

Figure 4.16. The correlations of two consecutive perceptrons created by two variants of the upstart algorithm (see the legend) are shown via the two order parameters r and $\hat{z}$ as a function of α for $\kappa = 1$ and $m_0 = 0$.

upstart algorithm are always significantly larger than for the tiling-like algorithm. Note, that the overlap r of the perceptrons for upstart III is larger than for upstart II, but the reverse is true for the magnitude of $\hat{z}$. The correlations for both variants and their differences are largest around the capacity limit. For $\alpha \to \infty$ the differences between the two variants vanish, since the difference in the training sets becomes negligible, and the correlations decay to zero with identical power-law exponents

$$r \propto \alpha^{-0.995\pm10} \qquad \text{and} \qquad \hat{z} \propto \alpha^{-0.96\pm5}, \tag{4.A.21}$$

where the deviation from -1 may be caused by logarithmic corrections. Note furthermore, that at the phase transition point α_p, the order parameters are non-differentiable as for the tiling-like algorithm with the possibility of local maxima for the magnitude of the correlation order parameters.

In general, one finds that any correlations decrease the capacity limit of the combined perceptrons or increase the error rate above saturation. The correlations are usually the largest around the capacity limit (besides for the tiling-like algorithm, where the maximum can be found around α_p for finite κ and zero or very small bias) and decay algebraically above it. In general, one also finds that the correlations grow with increasing stability and diminish for very large bias after passing through a maximum

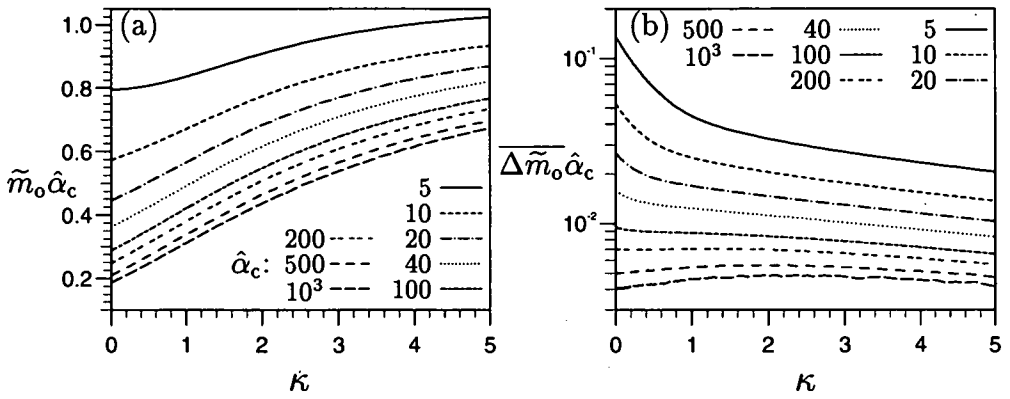

Figure 4.17. (a) The bias $\widetilde{m}_o$ (or rather $\widetilde{m}_o\hat{\alpha}_c$ to highlight the scaling of α_c with $\widetilde{m}_o$) is shown as a function of the stability κ for several fixed normalized capacities $\hat{\alpha}_c(m_o) \equiv \alpha_c(m_o)/\alpha_c(0)$ (see the legend) for two perceptrons built by the tiling-like algorithm . (b) For the upstart III algorithm, the results are quite similar for large $\hat{\alpha}_c$ and to highlight the differences, $\overline{\Delta\widetilde{m}_o} \equiv [\widetilde{m}_o(\mathcal{A}_{\mathrm{III}}^{\mathrm{up}}) - \widetilde{m}_o(\mathcal{A}^t)]/\widetilde{m}_o(\mathcal{A}^t)$ is shown normalized by $\hat{\alpha}_c$.

for non-zero bias.

In order to assess the impact of the correlations in comparison to RSB in the individual perceptrons above saturation more quantitatively, we concentrate on the capacity, the region where the correlations have generally the largest impact. The most meaningful comparison is found by calculating and comparing $\widetilde{m}_o$ as a function of κ for various fixed normalized capacities $\hat{\alpha}_c$ as in Figure 4.2 for the tiling-like and the variants of the upstart algorithm and for correlated RS as well as uncorrelated RS and 1RSB ansätze. Note, that $\widetilde{m}_o$ [defined as $\widetilde{m}_o \equiv (1 - |m_o|)$] is ambiguous for the upstart algorithm as the symmetry of the capacity under $m_o \leftrightarrow -m_o$ is broken in the above calculation and a true capacity limit would also need three perceptrons. This ambiguity is resolved by postulating that the "capacity limit" for bias m_o is given by the unit type that saturates first and leading to the constraint $m_o < 0$ for the upstart calculation w.l.o.g.. Note furthermore, that for decreasing $\widetilde{m}_o$ the portion of wrongly-on and correctly-off patterns, which cause the difference between the versions of the upstart algorithm and between the upstart and the tiling-like algorithm, become smaller, leading to similar results for large $\hat{\alpha}_c$.

In Figure 4.17, $\widetilde{m}_o$ is therefore only shown for two perceptrons coupled via the tiling-like algorithm in the uncorrelated 1RSB ansatz [Figure 4.17(a)], whereas $\overline{\Delta\widetilde{m}_o} = [\widetilde{m}_o(\mathcal{A}_{\mathrm{III}}^{\mathrm{up}}) - \widetilde{m}_o(\mathcal{A}^t)]/\widetilde{m}_o(\mathcal{A}^t)$ is shown for the upstart III algorithm to magnify the differences [Figure 4.17(b)]. Both quantities are multiplied by $\hat{\alpha}_c$ in order to eliminate

the most dominant scaling. In comparison to the single perceptron (Figure 4.2), $\widetilde{m}_o$ is larger due to the addition of the second perceptron and grows slower with κ due to the error rate of the first perceptron increasing more quickly for larger stability. This already indicates that the tendency of $\widetilde{m}_o$ growing with κ reverses eventually for larger networks. The comparison of $\widetilde{m}_o$ in Figure 4.17(b) for the two constructive algorithms shows that $\widetilde{m}_o$ for the upstart algorithm is always larger, although the difference in $\widetilde{m}_o$ vanish very quickly $\hat{\alpha}_c \to \infty$ (or $\widetilde{m}_o \to 0$), especially considering the fact that the difference in $\widetilde{m}_o$ has been magnified additionally by $1/\widetilde{m}_o(\mathcal{A}^t)$.

In Figure 4.18, the impact of 1RSB and correlations (within an RS) ansatz ($\mathrm{RS_c}$) is compared by plotting $\Delta\widetilde{m}_{o\,\mathrm{RS_c}}^{\mathrm{RSB}} \equiv (\widetilde{m}_o^{\mathrm{RSB}} - \widetilde{m}_o^{\mathrm{RS}})/(\widetilde{m}_o^{\mathrm{RS_c}} - \widetilde{m}_o^{\mathrm{RS}})$ in the same scenario as Figure 4.17 for the tiling-like [Figure 4.18(a)] and the upstart III algorithm [Figure 4.18(b)] (the differences to upstart II are insignificant). For both tiling-like and upstart algorithm, one finds that the corrections due to correlations are usually smaller than those due to RSB and become insignificant for large $\hat{\alpha}_c$ ($\Delta\widetilde{m}_{o\,\mathrm{RS_c}}^{\mathrm{RSB}} \gg 1$), corresponding to large bias m_o (as found for the last perceptrons of a large network). Furthermore, one finds that the impact of the correlations increases in general with the stability κ. For small $\hat{\alpha}_c$ and large stability κ, one actually finds a region where correlations are more significant than RSB ($\Delta\widetilde{m}_{o\,\mathrm{RS_c}}^{\mathrm{RSB}} < 1$), suggesting corrections of the capacity results for small networks and slightly biased output distributions. Note, that the correlation corrections of the capacity are usually at least twice as large for the upstart than for the tiling-like algorithm, which has already been suggested by the larger magnitude of the upstart than the tiling-like correlation order parameters shown in Figures 4.16 and 4.15, respectively.

Although, the correlations can therefore be significant in some regions of m_o, κ, and α space, the area is confined to small m_o, large κ, and α around the capacity limit, which is not extremely relevant for large networks, where most units are highly saturated, and small stabilities, which are of most interest. An open question is the effect of correlation over several generations of the constructive algorithm. It seems, however, natural that the correlations between consecutive perceptrons should be dominant.

4.B Propagation of errors for the upstart algorithm

Similarly to the tiling-like algorithm, the capacity of networks built by variants of the upstart algorithm can be calculated by propagating the errors of the individual perceptrons. Consider for example the procedure for upstart IIIa. Again the rules of the algorithm as described in Section 4.2 are followed and the similar considerations to Section 4.3.3 are applied. The following notable differences are caused by the upstart

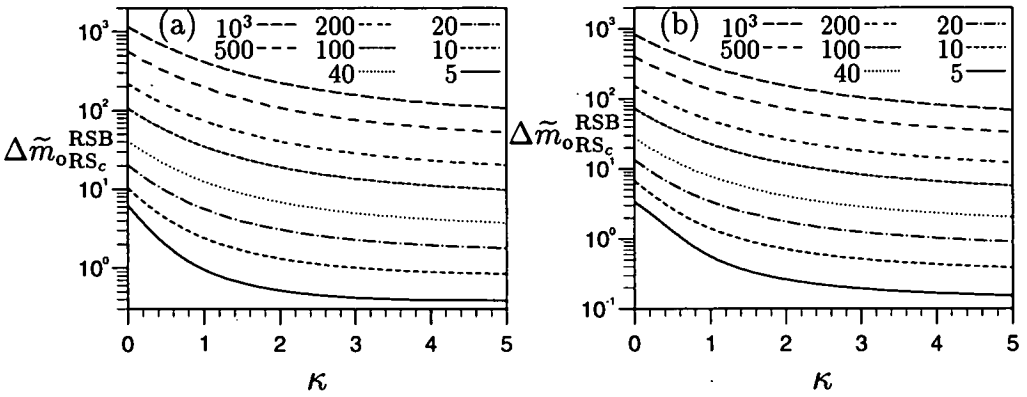

Figure 4.18. The size of corrections to $\widetilde{m}_o$ from the RS ansatz due to RSB and correlations within the RS ansatz (RS_c) is compared by plotting $\Delta\widetilde{m}_{o\,RS_c}^{RSB} \equiv (\widetilde{m}_o^{RSB} - \widetilde{m}_o^{RS})/(\widetilde{m}_o^{RS_c} - \widetilde{m}_o^{RS})$ in the same scenario as Figure 4.17 for (a) the tiling-like and (b) the upstart III algorithm.

algorithm creating two different types of units, $\mathcal{U}^+$ and $\mathcal{U}^-$. First, the procedure becomes dependent on the type of error made, i.e., is written in terms of ϵ^{on} and ϵ^{off} instead of $\epsilon = \epsilon^{on} + \epsilon^{off}$. Furthermore, since $\mathcal{U}^-$ is connected to the output with a negative weight, the rôles of *off* and *on* are reversed, e.g., wrongly-on patterns of $\mathcal{U}^-$ are actually wrongly-off patterns for $\mathcal{M}$. Second, the example load α_i decreases from the initial load α_0 over subsequent generations, since patterns can be omitted from the training sets and it is useful to introduce the quantities $\bar{\alpha}_i \equiv \alpha_i/\alpha_0$, which is the probability of an example being in the training set and will be referred to as load fractions. Third, $\widetilde{m}_i$ is in general negative since the target of the majority of the patterns is 0 (or -1) and not $+1$ as in the tiling-like algorithm[26], and $\widetilde{m}_i \equiv 1 + m_i$. It is furthermore useful to restrict the bias of the original output distribution to $m_o < 0$ w.l.o.g., and to introduce the fraction of training patterns $\bar{\alpha}_0^{\zeta^\pm} \equiv (1 \pm m_o)/2$ with $\zeta^\pm = \pm 1$. Fourth, it becomes necessary to propagate errors over several generations, since only one type of unit is created with each generation and subsequently only one type of error is dealt with. A symbolic program for the procedure called by the capacity root solver is outlined in Figure 4.19. The procedure for upstart IIIb is identical to upstart IIIa, but for a change in the creation criterion, which is changed to $(\epsilon_i^{on}\bar{\alpha}_0^{\zeta^-} > \epsilon_i^{off}\bar{\alpha}_0^{\zeta^+})$ to account for conditioning the error type probability on the initial target probability.

[26]Due to the symmetry of the equations for $m_o \to -m_o$, $\zeta^\mu \to -\zeta^\mu$, and $\theta \to -\theta$, this effectively only changes $\epsilon^{on} \leftrightarrow \epsilon^{off}$ and the sign of the threshold

```
upstartIIIa_cap := proc(α_c^K)                                  % begin procedure
global K, κ, m_o;                                               % global variables
local ···;                                                      % local variables (here unspecified)
α_0 := Kα_c^K; m̃_o := 1 + m_o; ᾱ_0^{ζ+} := m̃_o/2; ᾱ_0^{ζ−} := 1 − ᾱ_0^{ζ+};   % more global
m_flag:= TRUE;                                                  % flag for unit type
ᾱ_1 := 1; m̃_1 := m̃_o;                                          % initialize first perceptron U_1^+
ε_0^on:=0; ε_0^off := 0;                                        % initialize errors
for (i = 1, K − 2) do                                           % loop over the erroneous perceptrons
    α_i := α_0 ᾱ_i;                                             % calculate load
    (ε^on, ε^off) := error_calc(α_i, m̃_i, κ);                   % calculate error rates
    if (m_flag) then ε^on ↔ ε^off; fi;                          % swap error types for U_i^−
    ε_i^on := ε_{i−1}^on + ᾱ_i ε^on;                            % accumulate error rates
    ε_i^off := ε_{i−1}^off + ᾱ_i ε^off;                         % (with respect to α_0)
    if (ε_i^on > ε_i^off) then                                  % apply creation criterion
        ᾱ_{i+1} := ε_i^on + ᾱ_0^{ζ+} − ε_i^off;                 % wrongly-on and correctly-on patterns
        m̃_{i+1} := 2ε_i^on/ᾱ_{i+1};                             % calculate new bias
        m_flag:= TRUE; ε_i^on := 0;                             % create U_{m+1}^− and reset ε^on
    else
        ᾱ_{i+1} := ε_i^off + ᾱ_0^{ζ−} − ε_i^on;                 % wrongly-off and correctly-off patterns
        m̃_{i+1} := 2ε_i^off/ᾱ_{i+1};                            % calculate new bias
        m_flag:= FALSE; ε_i^off := 0;                           % create U_{p+1}^+ and reset ε^off
    fi;
od;                                                             % have reached last two perceptrons
if (m_flag) then                                                % have already created U_{m+1}^−
    p := p + 1;                                                 % additionally create U_{p+1}^+
    ᾱ_K := ε_{K−1}^off + ᾱ_0^{ζ−}; m̃_K := 2ε_{K−1}^off/ᾱ_K;
else                                                            % have already created U_{p+1}^+
    m := m + 1;                                                 % additionally create U_{p+1}^−
    ᾱ_K := ε_{K−1}^on + ᾱ_0^{ζ+}; m̃_K := 2ε_{K−1}^on/ᾱ_K;
fi;
α_{K−1} := α_0 ᾱ_{K−1}; α_K := α_0 ᾱ_K;                         % calculate loads
α_c^a := perc_cap(m̃_{K−1}, κ)                                   % calculate the two capacity limit
α_c^b := perc_cap(m̃_K, κ)
Δα^a := α_c^a − α_{K−1}; Δα^b := α_c^b − α_K;                   % difference of capacity and load
RETURN(min(Δα^a, Δα^b));                                        % return more saturated perceptron
end;
```

Figure 4.19. A symbolic capacity calculation procedure of the upstart IIIa algorithm called by an all-purpose root solving routine.

For the procedure of upstart II, two major changes have to be made. The first is induced by the different training set criteria, i.e., the inclusion of wrongly-off patterns into $\mathcal{U}_i^-$'s (and wrongly-on patterns from $\mathcal{U}_i^+$'s) training set. The load fractions for $\mathcal{U}_{i+1}^{\pm}$ are therefore changed to $\bar{\alpha}_{i+1} := \epsilon_i^{\text{off/on}} + \bar{\alpha}_0^{\zeta\mp}$. Since these incorrect pattern are included in the training set, although no attempt is been made to correct them, it is self-evident, that it is possible to make an error on an example which is already labelled as incorrect. Therefore, the calculated error rates have to be corrected in order to avoid

such multiple counting of errors. Employing the assumption that the perceptrons are uncorrelated, the overall errors after the creation of a $\mathcal{U}_i^{\pm}$ unit in the current generation become

$$\epsilon_i^{\text{off/on}} = \bar{\alpha}_i \epsilon^{\text{off}} \qquad \text{and} \qquad \epsilon_i^{\text{on/off}} = \epsilon_{i-1}^{\text{on/off}} + \epsilon^{\text{on}} \left[1 - \frac{\epsilon_{i-1}^{\text{on/off}}}{\bar{\alpha}_0^{\zeta^{\mp}}} \right] \bar{\alpha}_i. \tag{4.B.1}$$

The second major changes caters for cases where both types of perceptrons are created simultaneously, e.g., allowing for the possibility of two error rate calculations in the inner loop. This case necessitates the introduction of unit specific quantities, such as $\widetilde{m}_i^{\pm}$ and $\bar{\alpha}_i^{\pm}$, where i is now purely a generation index[27]. Note, that in this case the errors are not propagated over generations, since both type of errors are corrected simultaneously, but the simultaneous training combined with the non-zero overlap of patterns between the two training sets again leads to an overlap of the erroneous patterns. Following similar consideratins as above, one finds

$$\epsilon_i^{\text{off/on}} = \epsilon_{\pm}^{\text{off}} \bar{\alpha}_i^{\pm} + \epsilon_{\mp}^{\text{on}} \left[1 - \epsilon_{\pm}^{\text{off}} \frac{\bar{\alpha}_i^{\pm}}{\bar{\alpha}_0^{\zeta^{\mp}}} \right] \bar{\alpha}_i^{\mp}. \tag{4.B.2}$$

A closer inspection of Eq. (4.B.2) reveals that its symmetry leads to identical wrongly-on and wrongly-off errors for all generations of the algorithm if the initial output distribution bias is zero ($\bar{\alpha}_0^{\zeta^{\pm}} = 1/2$) and the first perceptron makes the identical fraction of error types, i.e., the solution with zero threshold (see discussion in Section 4.3.2) applies. In this case, the computational load for the capacity calculation is reduced by a factor of two and the symmetry leads to much smoother capacity curves as the two last units saturate simultaneously.

4.C Derivation of the asympotitic upstart algorithm capacity

In the case of upstart algorithm variants, the capacity of the complete network is a function of the capacity α_c^K of the last perceptron $\mathcal{U}_K^{\pm}$ and its load fraction $\bar{\alpha}_K^{\pm}$, due to some patterns being eliminated from its training set. For upstart II, $\bar{\alpha}_K^{\pm}$ can be expressed in terms of the applied bias $\widetilde{m}_K^{\pm}$, and the initial output-distribution bias m_o [which is for convenience again written in terms of the initial load fractions $\bar{\alpha}_0^{\zeta^{\pm}} \equiv$

[27] The number of units created have therefore to be counted by a separate label.

$(1 \pm m_o)/2]$ to yield

$$\alpha_c^K = \frac{\alpha_c(\tilde{m}_K^{\pm})}{K} \frac{(2 - \tilde{m}_K^{\pm})}{2\bar{\alpha}_0^{\zeta^{\mp}}}. \tag{4.C.1}$$

For $K \to \infty$ and consequently $\tilde{m}_K^{\pm} \to 0$, the capacity becomes asymptotically

$$\alpha_c^K \simeq \frac{1}{b\bar{\alpha}_0^{\zeta^{\mp}}} [\log(K)]^{m-1}, \tag{4.C.2}$$

following the derivation for the tiling-like algorithm in Section 4.5.

For upstart III, the derivation becomes slightly more complicated due to the additional exclusion of one class of incorrect patterns from the training set of the saturated unit. In this case, $\bar{\alpha}_K^{\pm}$ can only be bounded in terms of $\tilde{m}_K^{\pm}$ and $\bar{\alpha}_0^{\zeta^{\pm}}$, leading to bounds on the capacity which are specific on the unit creation selection criterion applied. For upstart IIIa, the capacity bounds are

$$\frac{\alpha_c(\tilde{m}_K^{\pm})}{K} \frac{(2 - \tilde{m}_K^{\pm})}{2\bar{\alpha}_0^{\zeta^{\mp}}} \leq \alpha_c^K \leq \frac{\alpha_c(\tilde{m}_K^{\pm})}{K} \frac{1}{\bar{\alpha}_0^{\zeta^{\mp}}}, \tag{4.C.3}$$

and similarly for upstart IIIb, one finds

$$\frac{\alpha_c(\tilde{m}_K^{\pm})}{K} \frac{(2 - \tilde{m}_K^{\pm})}{2\bar{\alpha}_0^{\zeta^{\mp}}} \leq \alpha_c^K \leq \frac{\alpha_c(\tilde{m}_K^{\pm})}{K} \frac{[2 - (2 - 1/\bar{\alpha}_0^{\zeta^{\mp}})\tilde{m}_K^{\pm}]}{2\bar{\alpha}_0^{\zeta^{\mp}}}. \tag{4.C.4}$$

However, for $K \to \infty$ ($\tilde{m}_K^{\pm} \to 0$), these bounds become tight and the asymptotic capacity again reduces to Eq. (4.C.2).

Chapter 5

The Rôle of Biases in On-Line Learning of Two-Layer Networks

Abstract

Whereas previous chapters have studied capacity problems, the influence of biases in generalization problems is studied in this chapter. In particular, the learning dynamics of a two-layer neural network, a normalized soft-committee machine, is studied for on-line gradient descent learning. Within a non-equilibrium statistical mechanics framework, numerical studies show that the inclusion of adjustable biases dramatically alters the learning dynamics found previously. The symmetric phase which has often been predominant in the original model all but disappears for a non-degenerate bias task. The extended model furthermore exhibits a much richer dynamical behaviour, e.g., attractive suboptimal symmetric phases even for realizable cases and noiseless data.

5.1 Introduction

In the two previous chapters we investigated the storage capacity of ANN, i.e.,the ability of a network to memorize random input-output patterns, and their behaviour above saturation. In this and the next chapter, we will address the problem of generalization in ANN, i.e., their ability to learn a mapping from examples. These examples are again given in the form of input-output pairs, in this case, however, the output labels are generated by a teacher whose mapping the student aims to infer.

Similarly to the capacity problem, the generalization problem is relatively well understood for simple perceptrons, but progress for architectures with hidden layers has

been hampered by the inability to perform the necessary quenched average over the training set in order to study their performance independent of the particularities of an individual training set. Some progress has been made in cases, where the transfer function of the hidden-layer is binary (Schwarze and Hertz 1992; Schwarze 1993; Urbanczik 1995). However, such network models are rarely used in practice since they cannot be trained by gradient-based minimization techniques.

Of much more interest is the theoretical understanding of the learning dynamics of MLPs with sigmoid activation function due to their paramount use in practical applications and their universal approximation ability (Cybenko 1989; Cybenko 1992). For these type of networks an equilibrium mechanics calculation has so far proved evasive, however, a method to overcome this problem has been introduced recently by Saad and Solla (1995b) using techniques from non-equilibrium statistical mechanics. It studies *on-line* learning in two-layer networks with an arbitrary number of hidden unit, allowing insight into the learning behaviour of neural network models whose complexity is of the same order as those used in real-world applications.

The on-line learning paradigm, whereby the network parameters are updated serially after the presentation of each single example, allows one to avoid the difficulties of averaging over a whole (finite) training set necessary for the more commonly studied *batch* learning algorithm, where all examples are used simultaneously to update the network parameters. The network model studied in particular, the soft-committee machine (Biehl and Schwarze 1995), consists of a single hidden layer with adjustable input-hidden, but fixed hidden-output weights (see Figure 2.3 for a general MLP architecture). The average learning dynamics of these networks are calculated in the thermodynamic limit of infinite input dimensions and in a student-teacher scenario where a *student* network is presented with training examples $(\boldsymbol{\xi}^\mu, \zeta^\mu)$. The input vectors $\boldsymbol{\xi}^\mu$ are Gaussian random variables and the outputs ζ^μ are labelled by a *teacher* network of the same architecture but possibly with a different number of hidden units. Although the framework allows in principle for any on-line learning algorithm to update the student parameter; gradient descent on the squared example error is studied in this chapter. In the following chapter, we will investigate how a simple modification to the gradient descent learning algorithm can improve learning times in many situations.

The above learning scenario is already quite similar to the problems faced in the real world, but the approach still suffers from several drawbacks. First, the analysis of the mean learning dynamics relies on the thermodynamic limit of infinite input dimension — a problem which has been addressed in (Barber et al. 1996), where finite size effects have been studied and it was shown that the thermodynamic limit is relevant in most cases. Second, the analysis also relies on the fact that the number

of hidden units remain finite and do not scale with the number of input dimensions, which is necessary for the universal approximation result to hold. A theory that is able to cope with this issue remains to be developed. Third, examples are not resampled, describing a scenario with an unrealistically large training set compared to most real cases, where training examples are scarce and therefore repeatedly cycled over. This problem provides a further yet unsolved technical challenge, although the issue has been considered at least for the linear perceptron (Sollich and Barber 1997a; Sollich and Barber 1997b). Fourth, the hidden-output weights are kept fixed, a constraint which has been relaxed in (Riegler and Biehl 1995; Riegler 1997), where it has been shown that the learning dynamics are usually dominated by the input-hidden weights. Fifth, the biases of the hidden units are fixed to zero, a constraint which is actually more severe than fixing the hidden-output weights. One can show (West et al. 1997) that soft-committee machines (without restricted number of hidden units) are universal approximators provided one allows for adjustable biases to the hidden layer.

In this chapter, we address the last limitation by studying the model of a normalized soft-committee machine with dynamic biases following the framework set out in (Saad and Solla 1995b). In Section 5.2 the model is defined and the calculation of the differential equations governing the training evolution is derived. In Section 5.3 numerical studies of a few typical learning scenarios are presented to show the qualitative difference in the dynamics to the model with fixed biases, most notably the emergence of attractive suboptimal network configurations. These and their dependence on the teacher task, the influence of weight and bias initialization, and the choice of the learning rates for weights and biases will be studied in Section 5.4. We will also set our results in context to previous works on weight initialization which devised heuristic rules. In Section 5.5 the optimal learning rates are calculated analytically for arbitrary network size and a range of teacher tasks for the convergence phase, where the student network is close to the optimal solution. In Section 5.6 we will outline possible extensions of this framework and in particular briefly assess the impact of unrealizable teacher rules. This is followed by a summary and discussion of the main results in Section 5.7.

5.2 Dynamical equations

The student network considered is a normalized soft-committee machine of K hidden units with adjustable biases. Each hidden unit i consists of a bias θ_i and a weight vector $\boldsymbol{W}_i$ which is connected to the N-dimensional inputs $\boldsymbol{\xi}$. All hidden units are connected to a linear output unit with arbitrary but fixed gain γ by couplings of fixed

strength. The activation of any unit is normalized (by the inverse square root of the number of weight connections into the unit) allowing all weights to be of $\mathcal{O}(1)$ magnitude, independent of the input dimension or the number of hidden units. Note that this is in contrast to most other on-line learning literature (e.g., (Biehl and Schwarze 1995)); however, this makes the necessary scaling of the learning rates more explicit and leads to more elegant results for optimal learning rates. The implemented mapping of a student with parameters $\mathbf{\Omega} = \{\boldsymbol{W}_i, \theta_i\}$ is therefore

$$\sigma(\boldsymbol{\xi}; \mathbf{\Omega}) = \frac{\gamma}{\sqrt{K}} \sum_{i=1}^{K} g\left(\frac{1}{\sqrt{N}} \boldsymbol{W}_i \cdot \boldsymbol{\xi} - \theta_i\right) = \frac{\gamma}{\sqrt{K}} \sum_{i=1}^{K} g\left(x_i - \theta_i\right), \tag{5.1}$$

where $x_i = \boldsymbol{W}_i \cdot \boldsymbol{\xi}/\sqrt{N}$ is the student activation and $g(\cdot)$ is a sigmoidal transfer function. Note, although the biases add only K degrees of freedom to the network, their influence on the hidden unit response is still of the same order as the complete weight vector.

The map to be learned is defined by a teacher network of the same architecture except for a possible difference in the number of hidden units M and is defined by the weight vectors $\boldsymbol{B}_n$ and biases ϱ_n $(n = 1, \ldots, M)$. Training examples are of the form $(\boldsymbol{\xi}^\mu, \zeta^\mu)$, where the components of the input vectors $\boldsymbol{\xi}^\mu$ are drawn independently from a zero-mean Gaussian distribution with arbitrary variance σ^2. The outputs are labelled by the teacher with parameters $\mathbf{\Omega}_0 = \{\boldsymbol{B}_n, \theta_i\}$ according to

$$\zeta^\mu = \zeta_0(\boldsymbol{\xi}^\mu; \mathbf{\Omega}_0) = \frac{\gamma}{\sqrt{M}} \sum_{n=1}^{M} g\left(\frac{1}{\sqrt{N}} \boldsymbol{B}_n \cdot \boldsymbol{\xi}^\mu - \varrho_n\right) = \frac{\gamma}{\sqrt{M}} \sum_{n=1}^{M} g\left(y_n^\mu - \varrho_n\right), \tag{5.2}$$

where $y_n^\mu = \boldsymbol{B}_n \cdot \boldsymbol{\xi}^\mu/\sqrt{N}$ is the activation of teacher hidden unit n. For simplicity, the labels are not corrupted by noise although this can be implemented straightforwardly (Saad and Solla 1997). Note that we will use indices i, j, k, l to refer to units in the student network and n, m for units in the teacher network.

In on-line learning the student parameters $\mathbf{\Omega}$, are modified to reduce the error the student makes on a presented single example $(\boldsymbol{\xi}^\mu, \zeta^\mu)$

$$\epsilon(\mathbf{\Omega}, \boldsymbol{\xi}^\mu) = \frac{1}{2}[\zeta^\mu - \sigma(\boldsymbol{\xi}^\mu; \mathbf{\Omega})]^2. \tag{5.3}$$

Gradient descent on the error (5.3), in this scenario commonly identified with *back-propagation* (Werbos 1974; Rumelhart et al. 1986a), results in updates of the student

parameters

$$\boldsymbol{W}_i^{\mu+1} - \boldsymbol{W}_i^{\mu} = \eta_w \delta_i^{\mu} \frac{\boldsymbol{\xi}^{\mu}}{\sqrt{N}} \tag{5.4a}$$

$$\theta_i^{\mu+1} - \theta_i^{\mu} = -\frac{\eta_\theta}{N} \delta_i^{\mu} \tag{5.4b}$$

with

$$\delta_i^{\mu} = \delta^{\mu} g'(x_i^{\mu} - \theta_i) = [\zeta^{\mu} - \sigma(\boldsymbol{\xi}^{\mu}; \boldsymbol{\Omega})]\, g'(x_i^{\mu} - \theta_i), \tag{5.4c}$$

where g' is the derivative of the activation function g. The two learning rates, η_w for the weights and η_θ for the biases (which has been rescaled explicitly by $1/N$), have to be set by the user to ensure both fast training and convergence to a minimum of the generalization error. Note, that δ^{μ} is the linear error for the present example and δ_i^{μ} is often viewed as the back-propagation of the error through the hidden node j (Hertz, Krogh, and Palmer 1991). This back-propagation scheme has the advantage of being both local and of having a computational complexity linear in the number of weights.

The above Markovian stochastic dynamics (5.4) are hard to solve generally since this necessitates solving a master equation for the time evolution of the weight and bias probability distributions. The initial approximations was the use of small learning rates in order to be able to expand the master equation and approximate it by a Fokker-Planck equation. However, this approach still fails when applied to the whole (global) learning process and can only be used close to attractive fixed points of the dynamics (Heskes 1994).

However, one is ultimately interested mainly in the typical performance of the student network on a randomly selected input example given by the *generalization error*

$$\epsilon_g(\boldsymbol{\Omega}) = \langle \epsilon(\boldsymbol{\Omega}, \boldsymbol{\xi}) \rangle_{\boldsymbol{\xi}} \tag{5.5}$$

Since the dependence of the inputs enter only through the student and teacher activations $\boldsymbol{x} = (x_1, \ldots, x_K)$ and $\boldsymbol{y} = (y_1, \ldots, y_M)$, the probability of $\boldsymbol{\xi}$ can be rewritten in terms of a joint probability distribution in the activations. The resulting distribution is Gaussian with zero mean as $\langle x_i \rangle_{\boldsymbol{\xi}} = \langle y_n \rangle_{\boldsymbol{\xi}} = 0$ and a covariance matrix $\boldsymbol{C}$ whose components are given by the order parameters describing the overlaps between student and teacher nodes:

$$\langle x_i\, x_j \rangle_{\boldsymbol{\xi}} = \frac{\sigma^2}{N} \boldsymbol{W}_i \cdot \boldsymbol{W}_j \equiv \sigma^2 Q_{ij}, \tag{5.6a}$$

$$\langle x_i\, y_n\rangle_{\boldsymbol{\xi}} = \frac{\sigma^2}{N}\boldsymbol{W}_i\cdot\boldsymbol{B}_n \equiv \sigma^2 R_{in}, \tag{5.6b}$$

$$\langle y_n\, y_m\rangle_{\boldsymbol{\xi}} = \frac{\sigma^2}{N}\boldsymbol{B}_n\cdot\boldsymbol{B}_m \equiv \sigma^2 T_{nm}, \tag{5.6c}$$

As also the weights solely enter through the activations, the generalization error must be a function of these order parameters and the biases θ_i and ϱ_n only. It would therefore be a major simplification if the dynamics could be rewritten solely in a macroscopic set of order parameters and the biases. This approach can be formalized in terms of a master equation and Fokker-Planck approach (Mace and Coolen 1997) for both on-line and batch dynamics for finite and infinite training sets. In the case of on-line learning with a formally infinite training set (i.e., examples are not recycled), the result is identical to the more intuitive approach we are providing here.

Taking a look at Eq. (5.4), we realize that the difference equations for the weights $\boldsymbol{W}_i$ can be rewritten as difference equations for Q_{ij} and R_{in}, by either squaring or multiplying equations with each other or by taking the scalar products with the various teacher weight vectors and including the normalization by $1/N$

$$Q_{ij}^{\mu+1} - Q_{ij}^{\mu} = \frac{\eta_w}{N}\left(\delta_i^\mu x_j^\mu + \delta_k^\mu x_i^\mu\right) + \frac{\eta_w^2}{N}\delta_i^\mu\delta_k^\mu\frac{1}{N}\boldsymbol{\xi}^\mu\cdot\boldsymbol{\xi}^\mu, \tag{5.7a}$$

$$R_{in}^{\mu+1} - R_{in}^{\mu} = \frac{\eta_w}{N}\eta_w\delta_i^\mu y_n^\mu, \tag{5.7b}$$

$$\theta_i^{\mu+1} - \theta_i^{\mu} = -\frac{\eta_\theta}{N}\delta_i^\mu. \tag{5.7c}$$

The order parameters Q_{ij} and R_{in} replace the $\boldsymbol{W}_i$ as dynamical variables, whereas the order parameters remain fixed T_{nm} and are defined by the task. In the thermodynamic limit ($N\to\infty$), the dynamical order parameters Q_{ij} and R_{in} become self-averaging with respect to the randomness in the training data, i.e., their probability distributions become δ-functions at their mean value, and it is sufficient to study their mean evolution by averaging over the input distribution or rather the joint Gaussian distribution of the activations.

Although it is known that self-averaging holds for overlap-type order parameter dynamics, this is not entirely self-evident for the bias dynamics. This has been addressed by scaling the learning rate for the biases in Eq. (5.4) by $1/N$, such that the updates of the biases becomes of the same order as those of the order parameters in Eq. (5.7). Furthermore, extensive simulations for a number of finite system sizes N conclusively confirm that the bias dynamics are also self-averaging and their variances exhibit a $1/N$ scaling behaviour. For the details of the simulations we refer the reader to Section 5.3. In the case of adjustable hidden-output weights, a rigorous proof (which can be ex-

tended to apply to biases) for self-averaging for $\mathcal{O}(1/N)$ updates is given in (Riegler 1997).

If one further interprets the normalized example number $\alpha = \mu/N$ as a continuous time variable, the difference equations can be conveniently rewritten as first-order coupled differential equations

$$\frac{dQ_{ij}}{d\alpha} = \eta_w \langle \delta_i x_j + \delta_j x_i \rangle_{\boldsymbol{\xi}} + \eta_w^2 \langle \delta_i \delta_j \rangle_{\boldsymbol{\xi}}, \tag{5.8a}$$

$$\frac{dR_{in}}{d\alpha} = \eta_w \langle \delta_i y_n \rangle_{\boldsymbol{\xi}}, \tag{5.8b}$$

$$\frac{d\theta_i}{d\alpha} = -\eta_\theta \langle \delta_i \rangle_{\boldsymbol{\xi}}. \tag{5.8c}$$

The scaling of the bias learning rate with $1/N$ may suggest that the dynamics of the biases and the weights are mismatched in this framework for at least some of the learning stages, leading to an optimal learning rate for the biases at infinity. This effect has already been observed in the case of adaptive hidden-output weights (Riegler 1997).

For dynamics on different time scales or different order of learning rates, it is natural to apply the method of adiabatic elimination (Gardiner 1983) to the fast variables, here the hidden-output weights or biases. In this approximation, it is assumed that the fast variables driven by the large learning rates are forced to relax to an attractive fixed point of the their dynamics assuming the slow variables, i.e., input-hidden weight order parameters, to be constant. This method has already been employed successfully for adaptive hidden-output weights (Riegler 1997), where it has been shown also that the ensuing dynamics for the order parameters are again self-averaging. One can further show (Rattray and Saad 1997a), that adiabatic elimination for the hidden-output weights is not only locally optimal by minimizing the generalization error with respect to the hidden-output weights instantly but also globally optimal. In the case of adiabatic elimination of the bias dynamics, neither can be shown since the equilibrium values of the biases are calculated from a set of nonlinear equations, whereas the equilibrium of the hidden-output weights is given by a set of linear equations. Furthermore, the solution of the nonlinear set of equations does not necessarily need to be unique, a problem which can be removed by demanding that the bias dynamics should relax dynamically to an attractive solution from their previous equilibrium values. A detailed treatment would go beyond the scope this thesis has set itself although we will present some results derived by this approximation where deemed appropriate.

Most integrations in Eqs. (5.8) can be performed analytically for the choice of the error function $g_\nu(x) = \mathrm{erf}(\nu x/\sqrt{2})$ (see Figure 2.2) as the sigmoidal transfer function, but for single Gaussian integrals remaining for η_w^2-terms and the generalization error.

For the exact form of the dynamical equations and the generalization error the reader is referred to Appendix 5.A. We only mention in passing that the variance of the input distribution σ^2 merely rescales the weight order parameters and the weight learning rates by σ^2. The sigmoidal gain ν rescales the weight order parameters and weight learning rate by ν^2 and the biases and bias learning rate by ν. The output gain γ rescales all learning rates by γ^2. In the following these parameters are therefore set to one without loss of generality.

Before we will present some typical results for the training evolution by numerically integrating the differential equations (5.8), we would like to classify the huge variety of learning scenarios in this framework into some distinct generic tasks. In the original model with fixed biases (Saad and Solla 1995b), it has been found useful to classify a learning scenario according to the isotropy of its teacher weight vectors. Tasks with very similar norms of the hidden unit weight vectors exhibit a much longer training time than tasks with strongly graded norms, which can especially be attributed to the problem of symmetry breaking in the space of the student hidden units. This may somewhat be caused by the identical output distributions of the individual teacher hidden units with the same norm. Only the differences in the initial student-teacher overlaps R_{in} introduced by the random initial conditions, allow the student hidden units to distinguish between the teacher hidden units in this case. For graded teacher lengths, the hidden unit output distributions still have zero mean but differ in the variance and higher cumulants. In this case, asymmetric initialization of the student-student overlaps Q_{ij} is sufficient to break student node symmetry.

The extra degrees of freedom introduced by the biases should have similar symmetry breaking effects. For simplicity, assume for the moment that the teacher weight vectors are isotropic. In the case that all teacher biases are degenerate ($\varrho_n = \varrho$), the identical hidden unit output distributions are shifted, with means

$$\langle g(y_n - \varrho_n) \rangle_\xi = -g\left(\frac{\varrho_n}{\sqrt{1+T_{nn}}}\right). \tag{5.9}$$

Again, one finds that only asymmetric initial conditions of the student-teacher overlaps R_{in} can break the symmetry. If, however, the teacher biases are non-degenerate, the teacher hidden unit output distributions are all different, e.g., have shifted means. In this case, asymmetric initial values of the student biases are sufficient to break the student hidden-unit symmetry. We will later see, that this symmetry breaking effect is stronger than that introduced by graded teacher lengths. For graded teachers, the only obvious choice for "degenerate" teacher biases is $\varrho_n = 0$. For non-zero teacher biases, the mean of the output distribution will shift according to Eq. (5.9). The choice

$\varrho_n = \varrho$ leads to student hidden unit symmetry breaking even for identical initial weight vectors as long as the initial student biases are not identical as well; clearly a sign of "non-degenerate" biases when compared to isotropic teacher weights. Two other possible scaling ansätze for "degenerate" teacher biases in the case of graded teacher lengths are

$$\hat{\varrho} = \frac{\varrho}{\sqrt{1+T}}, \tag{5.10a}$$

$$\check{\varrho} = \frac{\varrho}{\sqrt{T}}, \tag{5.10b}$$

where $\hat{\varrho}$ restores identical means of the individual teacher hidden unit output distributions, whereas $\check{\varrho}$ restores identical distances of the decision hyperplane (in the following termed *abscissa*) of the sigmoidal transfer function to the origin. Neither of these ansätze (or any other ansatz inspired by numerical results) seems to restore "degenerate" teacher biases perfectly, reflecting the fact that it is impossible to preserve output distribution symmetries for non-zero means, due to the skewed distributions induced by the nonlinearity. However, once the teacher lengths and one teacher bias is fixed, one can numerically always find a set of teacher biases which exhibit at least a very slow learning progress. Unfortunately, we have not been able to find a consistent ansatz that can predict these correctly, although they are in many cases close to the values given by the ansatz (5.10a). In general, we have found this ansatz more useful in most cases and we will therefore term $\hat{\varrho}$ the *effective bias*.

Summarizing the above argument, it makes sense to classify teacher tasks according to the following two criteria:

- Degree of isotropy in the teacher norms. Isotropic teacher tasks are defined by similar weight vector lengths ($T_{nm} = T\delta_{nm}$), whereas graded teachers tasks feature norms with different values. These are referred to as $\mathcal{T}^i$ and $\mathcal{T}^g$, respectively.

- Degree of degeneracy in the student biases. For isotropic teacher weights, degenerate teachers tasks are defined by similar biases ($\varrho_n = \varrho$), whereas non-degenerate teachers tasks exhibit biases with distinct values. These tasks are referred to as $\mathcal{T}_d$ and $\mathcal{T}_n$, respectively.

For graded teacher weights, degenerate biases as such are only given for $\varrho_n = 0$, although one can also find sets of non-zero biases numerically that are approximately "degenerate."

5.3 Typical evolution of the dynamical equations

The differential equations can only be solved accurately in moderate times for smaller student networks ($K \leq 5$) but any teacher size M due to the required numerical integrations. For small learning rates, where η_w^2-terms can be neglected, the differential equations can be solved for any K. For the remainder of the chapter, we would like to focus on the influence of different bias scenarios and the influence of the learning rates. We therefore restrict ourselves otherwise mainly to small realizable networks ($K = M$ with $K = 2, 3$) and uncorrelated isotropic teacher weight vectors of arbitrary length ($T_{nm} = T\delta_{nm}$).

The dynamical evolution of the overlaps Q_{ij}, R_{in} and the biases θ_i follows from integrating the equations of motion (5.8) from initial conditions determined by the (random) initialization of the student weights $\boldsymbol{W}_i$ and biases θ_i. For random initialization the resulting norms Q_{ii} of the student vector will be $\mathcal{O}(1)$, while the overlaps Q_{ij} between different student vectors, and student-teacher vectors R_{in} will be only $\mathcal{O}(1/\sqrt{N})$. A random initialization of the weights and biases can therefore be simulated by initializing the norms Q_{ii}, the biases θ_i and the normalized overlaps $\hat{Q}_{ij} = Q_{ij}/\sqrt{Q_{ii}Q_{jj}}$ and $\hat{R}_{in} = R_{in}/\sqrt{Q_{ii}T_{nn}}$ from uniform distributions in the $[0, 1]$, $[-1, 1]$, and $[-10^{-12}, 10^{-12}]$ intervals, respectively. We find that the results of the numerical integration are sensitive to these random initial values which has not been the case to this extent for fixed biases. To study the effect of different weight initialization, we have fixed the initial values of the student-student overlaps Q_{ij} and biases θ_i for some of the numerical examples, as these can be manipulated freely in any learning scenario. The initial student-teacher overlaps R_{in} are always randomized as suggested above.

In our first example (Figure 5.1), we demonstrate the potential influence of the adjustable biases in the learning dynamics of the soft-committee machine model, by comparing two typical realizable learning tasks ($K = M = 2$) with isotropic teacher weight vectors $\mathcal{T}^i$ ($T_{nm} = \delta_{nm}$). The student parameters denoted by $*$ represent a learning scenario in the original model, where both student and teacher lack biases, i.e., $\theta_i = 0$ and $\varrho_n = 0$. The other scenarios feature student networks from the extended model, i.e., with adjustable biases. They are trained by an isotropic teacher task with small non-degenerate biases ($\varrho_{1,2} = \mp 0.1$). For both scenarios, the learning rate and the initial conditions were judiciously chosen to be $\eta_0 = 2.0$, $Q_{11} = 0.1$, $Q_{22} = 0.2$, $\hat{R}_{in} = \hat{Q}_{12} = U[-10^{-12}, 10^{-12}]$ with $\theta_1 = 0.0$ and $\theta_2 = 0.5$ for the student with adjustable biases.

In both cases, the student weight vectors [Figure 5.1(a)] are drawn quickly from their initial values into a suboptimal symmetric phase, characterized by the lack of

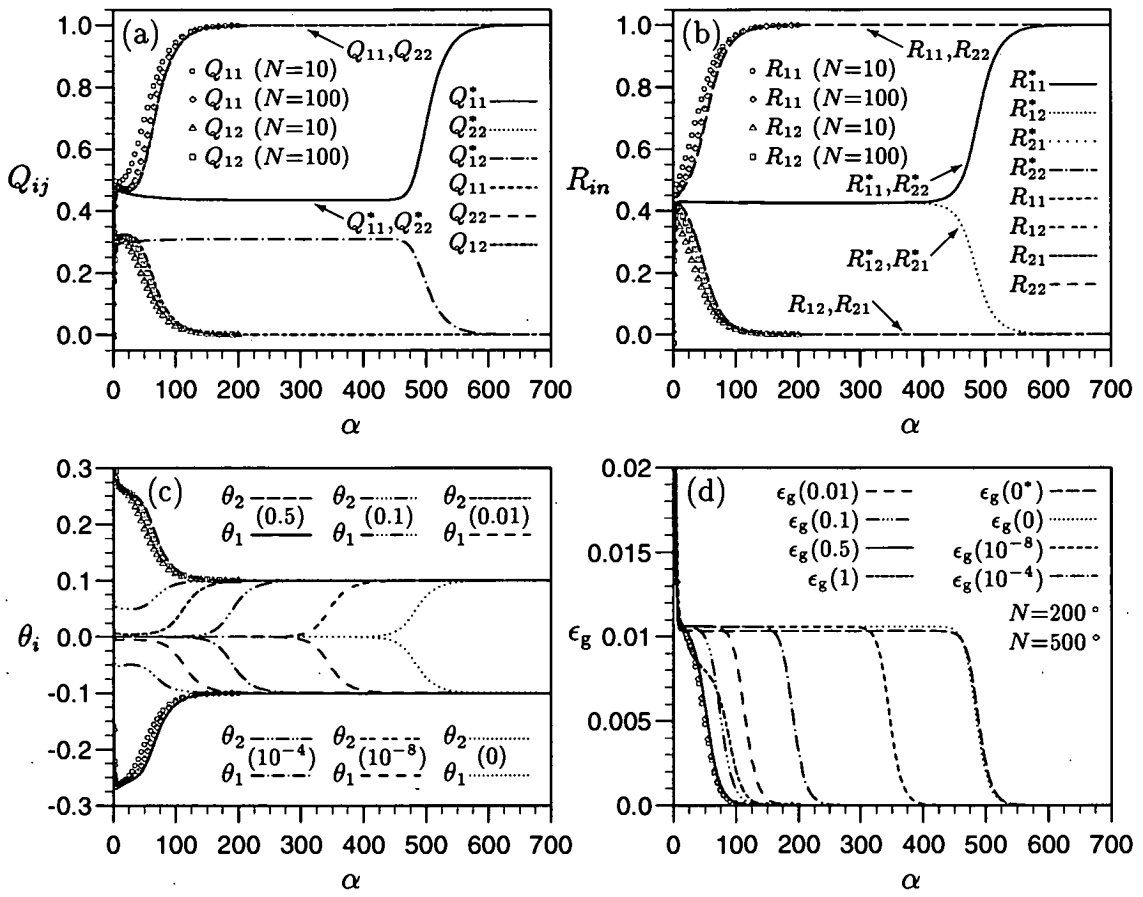

Figure 5.1. The dynamical evolution of (a) the student-student overlaps Q_{ij} and (b) the student-teacher overlaps R_{in} as a function of the normalized example number α is compared for two student-teacher scenarios. One student network (denoted by $*$) has fixed zero biases and is trained using examples generated by a bias-less teacher network. Other student networks have adjustable biases and are learning to imitate a teacher task with non-zero biases. The influence of the symmetry in the initialization of the biases on the dynamics is shown for (c) the student biases θ_i and (d) the generalization error ϵ_g. The initial value of $\theta_1 = 0$ is kept for all runs, but θ_2 varies and is given in brackets in the legends. Finite size simulations for input dimensions $N = 10 \ldots 500$ show that the dynamical variables are self-averaging. For all order parameters and the biases the mean trajectories for $N = 10$ and $N = 100$ are shown for the relevant order parameters {see the legends, for biases: θ_1 [$N = 10$ ($\circ$), $N = 100$ ($\diamond$)]; θ_2 [$N = 10$ ($\triangle$), $N = 100$ ($\square$)]}. For the generalization error we show the results for $N = 200$ and $N = 500$ for comparison.

specialization of the student hidden units on a particular teacher hidden unit, as can be depicted from the similar values of R_{in} in Figure 5.1(b). This symmetry is bro-

ken almost immediately in the learning scenario with adjustable student biases and non-degenerate teacher biases. The student converges quickly to the optimal solution, characterized by the evolution of the overlap matrices $\mathbf{Q}$, $\mathbf{R}$ and biases $\boldsymbol{\theta}$ [see Figure 5.1(c)] to their optimal values $\mathbf{T}$ and $\boldsymbol{\varrho}$ (up to the permutation symmetry due to the arbitrary labeling of the student nodes). Likewise, the generalization error ϵ_g decays to zero in Figure 5.1(d). The student with fixed biases is trapped for most of its training time in the symmetric phase before it converges eventually.

Before analysing the differences between the original soft-committee and the extended model further, we would like to briefly assess the influence of finite input dimension N on the dynamics, especially in order to confirm that the dynamic variables are self-averaging. In Figure 5.1 we therefore also compare the theoretical evolution of the overlaps, the biases and the generalization error with the simulation results for input dimensions $N = 10 \ldots 500$, for the above student and teacher scenario with adjustable biases. The initialization for the simulations are identical to the theory for the student norms and biases, but the overlaps were scaled appropriately with input dimension ($\hat{R}_{in} = \hat{Q}_{12} = U[-N^{-\frac{1}{2}}, N^{-\frac{1}{2}}]$).

Since the learning trajectory for finite N is stochastic, there is a probability for a student node permutation in the specialization process leading to multimodal probability distributions of the dynamic variables. To be able to calculate meaningful mean trajectories and variances, student nodes were therefore relabelled a posteriori. However, this permutation probability decreases in the simulations with $1/N^3$, leading to a well defined deterministic behaviour in the thermodynamic limit, i.e., the probability distributions of the dynamic variables become asymptotically unimodal. The resulting mean trajectories of the dynamic variables are shown for two input dimensions ($N = 10, 100$) in Figures 5.1(a–c), where some of the order parameters (Q_{22}, R_{22}, and R_{21}) were omitted as they have very similar values to others (Q_{11}, R_{11}, and R_{12}) due to the symmetry in the learned task. The size of the symbols is only a guide to the eye, but is generally much larger than the standard deviation in the mean. Even for the smallest input dimension of $N = 10$, the agreement of the simulations with the theoretical predictions is qualitatively good but the trajectories exhibit a systematic shift to smaller α values. For $N = 100$ the finite size effects on the mean trajectory are already very small. For comparison, the simulated value of the generalization error in Figure 5.1(d) for larger input dimensions ($N = 200, 500$) are already virtually indistinguishable from the theoretical predictions. In general, one finds that the deviations of the mean from their thermodynamic predictions and the variances of the dynamical fluctuations scale with $1/N$ as expected (Barber et al. 1996).

One of the most striking differences between the soft-committee machine with and

without biases is the length of the symmetric phase for non-degenerate teacher biases. In the model with fixed biases, the symmetric phase seems to dominate the overall training time in Figure 5.1. This issue will be discussed in detail in Chapter 6, where a simple extension of back-propagation training is proposed to alleviate this problem and compared to standard back-propagation (gradient descent) for the fixed bias model. Here, we will only mention that in this case the training time for back-propagation in isotropic teacher scenarios grows faster than K^2 in the symmetric phase in comparison to K in the convergence phase even for locally or globally optimized learning rates [see Chapter 6 and (Saad and Rattray 1997a; Saad and Rattray 1997b; Rattray and Saad 1997a)]. For small learning rates the trapping time is furthermore linearly extended with η_0. The influence of the initial conditions is only logarithmic through the differences in the initial student-teacher overlaps R_{in} (Biehl et al. 1996) which are typically of $\mathcal{O}(1/\sqrt{N})$ and cannot be influenced in real scenarios without *a priori* knowledge. The initialization of the biases, however, can be controlled by the user and its influence on the learning dynamics is shown in Figures 5.1(c) and 5.1(d) for the biases and the generalization error, respectively. For initially identical biases ($\theta_1 = \theta_2 = 0$), the evolution of the order parameters and hence the generalization error is almost indistinguishable from the fixed biases case. A breaking of this symmetry leads to a decrease of the symmetric phase linear in $\log(|\theta_1 - \theta_2|)$ until it has all but disappeared. The dynamics are again slowed down for very large initialization of the biases [see Figure 5.1(d)], where the biases have to be modified significantly before reaching their optimal values.

The influence of bias dynamics in the case of degenerate teacher biases is demonstrated in Figure 5.2; here we show the evolution of the overlaps, the biases and the generalization error from random initial conditions for $K = 3$ and a common learning rate ($\eta_0 = \eta_\theta = \eta_w = 2$) for a realizable task ($M = 3$) with isotropic weight vectors ($T_{nm} = \delta_{nm}$) and degenerate but non-zero biases ($\varrho_n = 1$). As before the student-student overlaps [Figure 5.2(a)] are quickly drawn into a symmetric subspace, characterized by similar overlaps R_{in} [Figure 5.2(b)] between each student node and all teacher nodes. The student biases [Figure 5.2(c)] take values which are symmetrically grouped around the true degenerate teacher biases. The breaking of the symmetry occurs in two stages. First, the third hidden unit, whose single student bias is located closest to the true bias value, begins to specialize on the third teacher unit. The other two student units decorrelate from the third and its associated teacher unit, but remain strongly correlated with each other and the two other teacher units. The two biases keep their symmetry around the true teacher bias value. These symmetries are eventually also broken and the student finally converges to the optimal solution. Although the evolution is therefore still characterized by three learning stages, transient

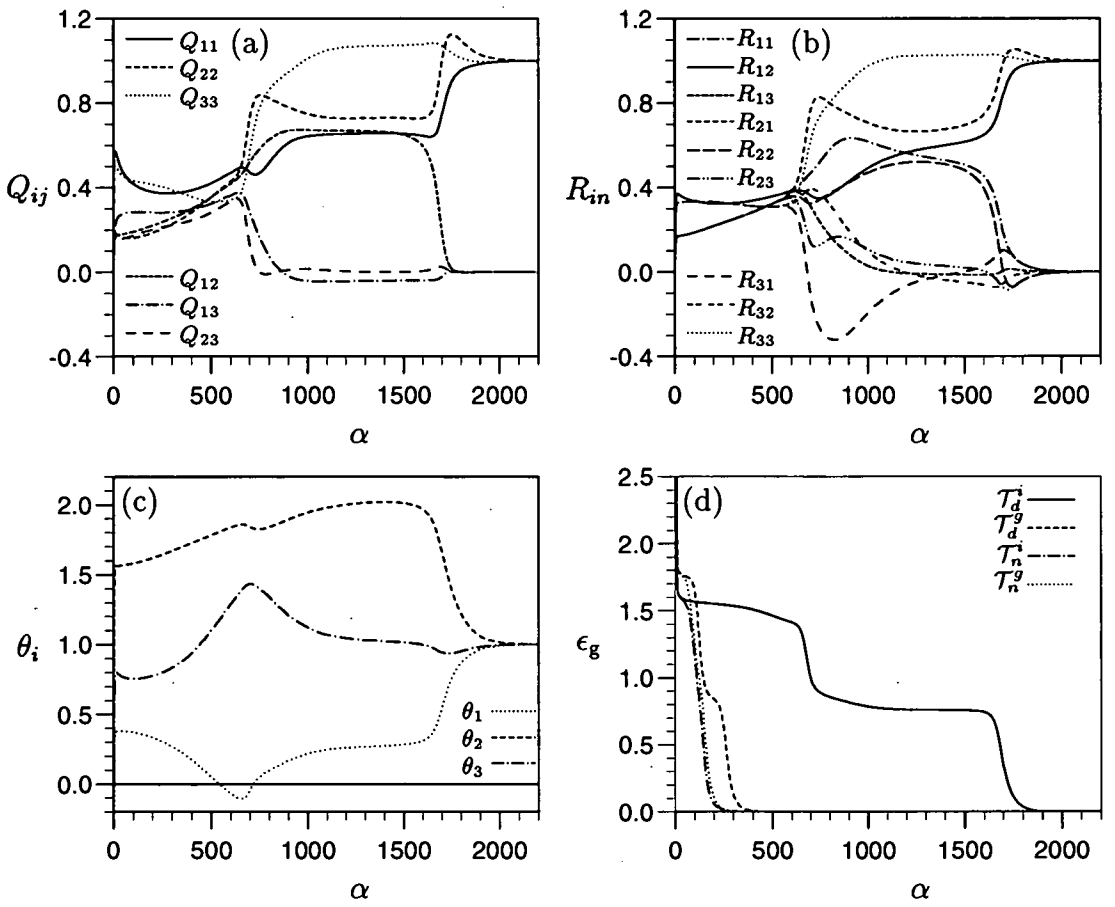

Figure 5.2. The dynamical evolution of the student-student overlaps Q_{ij} (a), the student-teacher overlaps R_{in} (b), the student biases θ_i (c) and the generalization error ϵ_g (d) as a function of the normalized example number α is shown for a realizable scenario $K = M = 3$ and $\eta_0 = \eta_\theta = \eta_w = 2$. The teacher tasks $\mathcal{T}_d^i$ large degree of symmetry ($T_{nm} = \delta_{nm}$ and $\varrho_n = 1$) is responsible for the very slow specialization process that takes place in two identifiable stages. Training time is shortened considerably when the teacher vector isotropy or bias degeneracy is broken.

to the symmetric phase, breaking of the symmetry and final convergence, similar to the evolution of the model with fixed biases, the extra degrees of freedom introduced by the biases enrich the dynamical evolution considerably.

To contrast the training behaviour in this very symmetric task $\mathcal{T}_d^i$ with the three other generic tasks which exhibit less symmetry, we introduce small deviations from the original symmetry by choosing $T_{nm} = (1 + 0.1n)\delta_{nm}$ instead of $T_{nm} = \delta_{nm}$ for teacher overlaps and/or $\varrho_n = 0.8 + 0.1n$ instead of $\varrho_n = 1$ for the biases. These deviations have a dramatic effect on the evolution of the generalization error in Figure 5.2(d). The

task $\mathcal{T}_d^i$ has by far the slowest training behaviour, with the sequential specialization process already described above for the order parameters. This is followed by the approximation to the task $\mathcal{T}_d^g$ which also features a sequential breaking of the symmetry but on a much shorter time scale. The fastest training times are exhibited for tasks $\mathcal{T}_n^g$ and $\mathcal{T}_n^i$ with no measurable speed up for the graded task, suggesting that non-degenerate biases affect the breaking of node symmetry more significantly than graded weight vectors. The strong symmetry breaking effect of the biases is arguably due to a steep minimum in the generalization error surface along the direction of the biases caused by the shift of the means of the individual hidden unit output distributions. This picture can be confirmed by the fact that the trajectories of the biases do not cross, i.e., the rank ordering according to the value of the bias is preserved at all times, whereas the ordering according to the norms is not. We have found this to be true for a range of other learning scenarios studied, including larger networks and more strongly graded teachers, provided that the biases were not initialized highly symmetrically. This seems to promote initialization schemes where the biases of the student hidden units are spread evenly across the input domain as has been suggested previously on a heuristic basis (Nguyen and Widrow 1990).

For the cases of degenerate teacher biases, the grouping of student biases found above is typical for all cases studied. For an even number of degenerate teacher biases, the student units combine in pairs. Each pair is characterized by its two biases having the same distance to the true teacher bias value with opposite sign and by its weight vectors being highly correlated. For an odd degeneracy, as above, the behaviour is similar but for a single remaining student bias which is stabilized around the true teacher bias value. The breaking of the symmetries in these cases can take a lot longer than for fixed biases and can be extremely complicated. It is often broken in stages as in the example given above, but can also occur simultaneously. We also find a strong influence of the training outcome on the initial conditions and the learning rate chosen, in some cases not all symmetries are broken and the student remains trapped in a suboptimal configuration, i.e., some of the symmetric fixed points are attractive.

To illustrate this point, the dynamics of the student biases θ_i are shown in Figure 5.3 for $K = M = 2$, $\eta_0 = 1$ and random initial conditions, and an isotropic teacher with degenerate biases ($\varrho_n = 0$). The student was initialized identically for the different runs (i.e., the same seed was used for the random number generator), but for a change in the range of the random initialization of the biases ($U[-b, b]$). We find that the student progress is inversely related to the magnitude of the bias initialization until a critical value of b is reached, where the student fails to converge at all. It remains in a suboptimal phase characterized by biases of the same large magnitude but oppo-

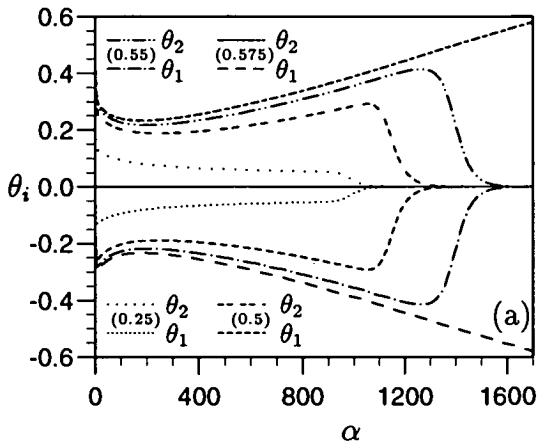

Figure 5.3. The dynamical evolution of the biases θ_i for a student imitating an isotropic teacher with zero biases reveals symmetric dynamics for θ_1 and θ_2. The student was randomly initialized identically for the different runs, but for a change in the range of the random initialization of the biases ($U[-b, b]$), with the value of b given in the legend. Above a critical value of b the student remains trapped in a suboptimal phase.

site sign and highly positively correlated weight vectors which have identical overlap with all respective teacher vectors. This behaviour may be explained by the fact that the generalization error decreases with increasing magnitude of the symmetric bias arrangement in the symmetric phase, suggesting the possibility of a local minimum in the generalization error surface. This may cause the dynamic competition between the specialization process of the student hidden units and the increase in magnitude of the biases observed in Figure 5.3, where the basin of attraction is determined by the initial conditions and the learning rates. Fastest convergence for this scenario is achieved for $b = 0$ and a reasonable bias initialization strategy seems therefore almost opposite to the above case of non-degenerate teacher biases.

In order to devise an initialization strategy which can cope well with all learning scenarios, we explore the influence of the initial conditions and the learning rate on the learning process more systematically in the following section.

5.4 Attractive fixed points

Although attractive symmetric fixed points have been found also for the soft-committee machine model with fixed biases (Biehl et al. 1996), these needed careful preparation of the initial conditions and were restricted to over-realizable cases. In the case of

adaptive biases, one finds a multitude of attractive sub-optimal fixed points for realizable cases with, in some cases, large basins of attraction. They exist not only in cases where both teacher weight vectors are isotropic and the biases degenerate but also for graded teachers and non-degenerate biases, although in these cases, the basins of attraction tend to shrink with increasing task asymmetry. In real world problems, the problem of poor local minima and the influence of the initial conditions on these is well known for back-propagation training. One can find numerous examples in the literature [e.g., (Kim and Ra 1991; van Ooyen and Nienhuis 1992)] which produce training error dynamics that look very similar to the evolution of the generalization error found in this chapter.

Subsequently, many algorithms [see e.g., (Lehtokangas et al. 1995) and references therein] have been proposed that aim to find good initial conditions. However, we are aware only of two (Nguyen and Widrow 1990; Kim and Ra 1991) which do not rely on information extracted from an *a priori* known training set and are therefore the only ones applicable in the framework studied. Below, we will therefore try to gain a qualitative understanding of how the initial conditions and the learning rates can be chosen to avoid becoming attracted to suboptimal network solutions. Our findings are then compared to the heuristicly based suggestions in (Nguyen and Widrow 1990; Kim and Ra 1991).

Due to the quadratic increase in the number of dynamic variables with the system size K, we restrict ourselves to the the smallest network size $K = 2$, although we have verified the validity of the drawn conclusions for larger networks. In particular, three elements which influence the size of the basin of attraction for given initial conditions were investigated: the task asymmetry (in terms of the teacher lengths and biases), the initial conditions and the learning rates.

Since the initialization space and hence the basins of attraction are still of high dimensionality, we have restricted ourselves to one-dimensional slices in one of the biases, θ_2, parameterized by a further variable. The remaining variables of the student were chosen to be $\eta_\theta = \eta_w = 2.0$, $Q_{11} = 0.1$, $Q_{22} = 0.2$, $\theta_1 = 0.0$, and $\hat{R}_{in} = \hat{Q}_{12} = U[-10^{-12}, 10^{-12}]$ (with a fixed random seed). The teacher task was usually chosen to be of the form $\mathcal{T}_n^i$ with $T_{nm} = \delta_{nm}$ and $\varrho_n = 0$, if not otherwise stated. The convergence time α_c was defined as the example number at which the generalization error has decayed to a small value, here judiciously chosen to be 10^{-8} requiring the student to have broken the symmetries in weight space successfully. The convergence time diverges in the case that the student is attracted to a suboptimal fixed point.

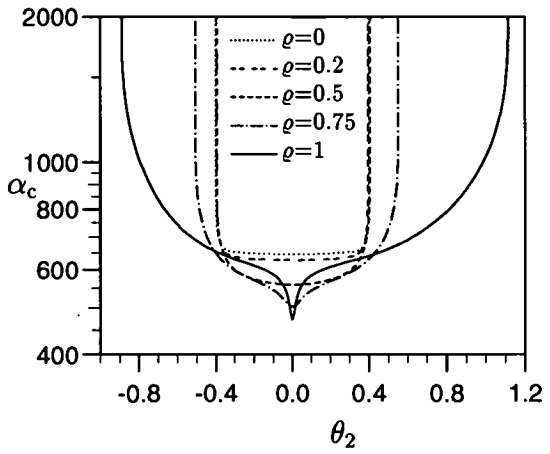

Figure 5.4. The convergence time $\alpha_c(\theta_2)$ (see the text) is shown for several values of the common teacher bias for the degenerate teacher bias task $\mathcal{T}_d^i$ ($\varrho_n = \varrho$). α_c diverges for large enough initial magnitude of θ_2 for all values of ϱ (see the legend). For increasing ϱ the basin of attraction to the optimal solution becomes asymmetric and larger.

5.4.1 Task asymmetry

In Figures 5.4–5.6 we compare the influence of the initialization of θ_2 on the convergence time α_c and the resulting basin of attraction for three different teacher tasks of the form $\mathcal{T}_d^i$, $\mathcal{T}_n^i$ and $\mathcal{T}_d^g$, where some sort of asymmetry was applied gradually to the original teacher task ($T_{nm} = \delta_{nm}$ and $\varrho_n = 0$).

In the case of degenerate teacher biases $\mathcal{T}_d^i$ (Figure 5.4) for which the biases were chosen to be $\varrho_n = \varrho$, the convergence time diverges beyond some critical absolute values $\theta_c^{\pm}$ of θ_2 and the basin of attraction to the optimal solution is restricted to $-\theta_c^- < \theta_2 < \theta_c^+$. For small ϱ this basin is symmetric ($\theta_c^- = \theta_c^+$) and almost constant in size, whereas for large ϱ, the basin is skewed and increases in size. The fastest convergence is always achieved for $\theta_2 = \theta_1 = 0$, i.e., when the teacher task degeneracy is reflected in the bias initialization. This effect becomes increasingly more pronounced for larger teacher bias values ϱ, which also generally show shorter convergence times. This effect may be explained by the fact that for small ϱ most examples are drawn from the region where the sigmoidal transfer function is linear, making the symmetry breaking process more difficult.

This behaviour is to be contrasted to the case of non-degenerate teacher bias tasks $\mathcal{T}_n^i$ characterized by $\varrho_n = \pm\varrho$ shown in Figure 5.5. Here, one finds that the basin of attraction to the optimal solution already increases substantially for very small values

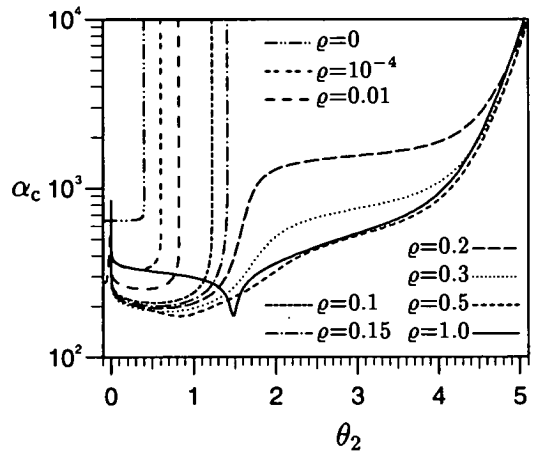

Figure 5.5. The convergence time $\alpha_c(\theta_2)$ is shown in terms of the asymmetry in the teacher biases T_n^i ($\varrho_n = \pm\varrho$). These tasks also exhibit an attractive suboptimal fixed point for small ϱ, but with a smaller basin of attraction. Above a critical value the suboptimal fixed point becomes unstable although it still can influence the learning process considerably. For very large initial values θ_2 (and large enough ϱ), the learning process is slowed down exponentionally, but the student is still able to converge to the optimal solution eventually.

of ϱ, although we still find that the student is drawn into a suboptimal solution for large enough initial θ_2. However, above a certain value in the teacher bias asymmetry $\varrho_c \approx 0.174$, the suboptimal solution ceases to be an attractive fixed point, although the dynamics can still be slowed down considerably due to the influence of the symmetric fixed point. Above ϱ_c and very large initial values θ_2, one finds that the convergence time increases exponentionally with θ_2, arguably due to the fact that the student hidden unit is initially highly saturated and the gradient decreases exponentionally.

We further find that the basin of attraction is always perfectly symmetric, unlike in the degenerate case since the hidden unit symmetry is broken by the biases and not the weights. This also explains the sharp peak in the convergence time for initial values around $\theta_2 = 0$ with

$$\alpha_c(\theta_2') - \alpha_c(\theta_2) \propto \log\left(\frac{|\theta_2|}{|\theta_2'|}\right) \tag{5.11}$$

for small initial values θ_2' and θ_2, as already shown in Figure 5.1(d). Eq. (5.11) holds exactly in the limit $\theta_2 \to 0$ only for $R_{in} = 0$, in which case the the convergence time diverges as only the biases can break the symmetry. Otherwise, the convergence time is affected by the specialization process triggered by the asymmetric initial conditions

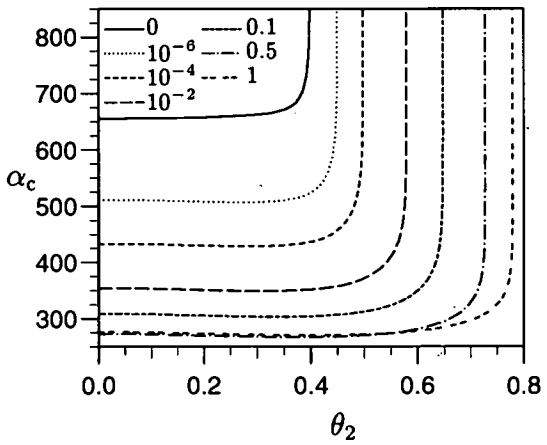

Figure 5.6. The convergence time α_c is shown as a function of difference in the teacher lengths $\delta T = T_{22} - T_{11}$ (see the legend). α_c is also reduced as for the asymmetric bias case (Figure 5.5), but the basin of attraction does not grow as significantly for the tasks $\mathcal{T}_d^g$.

in R_{in}. This is also true for the other laws [Eqs. (5.12) and (5.13)] found below.

Similarly, the shortest possible convergence time decreases initially with increasing task asymmetry according to

$$\alpha_c^{\rm opt}(\varrho') - \alpha_c^{\rm opt}(\varrho) \propto \log\left(\frac{\varrho}{\varrho'}\right), \tag{5.12}$$

and the minimum becomes sharper in terms of θ_2 for large ϱ. This minimum defines the optimal initial value $\theta_2^{\rm opt}(\varrho)$, which increases as expected with increasing ϱ, but is always considerably larger than ϱ. This effect is especially remarkable when taking the initial student norm into account, comparing the actual effective bias or alternatively the abscissa of the hidden units (i.e., $\varrho/\sqrt{1+T}$ and $\theta_2/\sqrt{1+Q_{22}}$ or $\varrho/\sqrt{T}$ and $\theta_2/\sqrt{Q_{22}}$).

The graded teacher task $\mathcal{T}_d^g$ also speeds up the breaking of hidden unit symmetry as shown in Figure 5.6 and reduces the optimal convergence time $\alpha_c^{\rm opt}$ substantially. The difference in convergence time due to a small task asymmetry is given in terms of the teacher length difference $\delta T = T_{22} - T_{11}$ by

$$\alpha_c^{\rm opt}(\delta T') - \alpha_c^{\rm opt}(\delta T) \propto \log\left(\frac{\delta T}{\delta T'}\right). \tag{5.13}$$

The total reduction in α_c for a given asymmetry is smaller when compared to $\mathcal{T}_n^i$. This confirms the observation made in Section 5.3 that the biases have a stronger symmetry breaking effect than the weights. This is also mirrored in the basin of

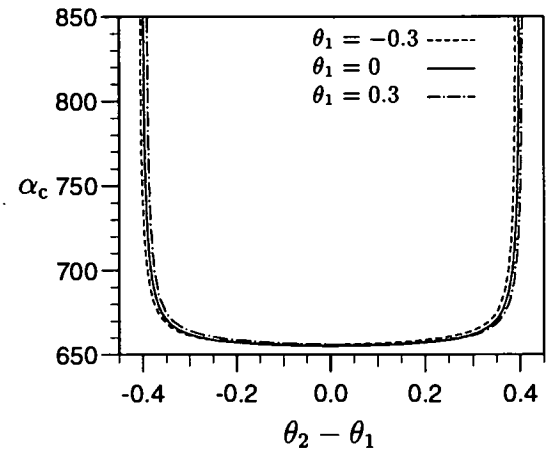

Figure 5.7. The basin of attraction for initial θ_2 shown for several values of θ_1 depends almost solely on the difference $\theta_2 - \theta_1$.

attraction increase, which is not as substantial as in the case of asymmetric biases, and the critical bias θ_c follows approximately $\theta_c(\delta T) - \theta_c(0) \propto \delta T^{0.141(3)}$.

We have found qualitatively similar results for larger networks, where the basin of attraction to the optimal solution also grows with the teacher task asymmetry. However, one also finds that the range of initial conditions attracted to the optimal solution shrinks with network size for a given teacher task asymmetry (e.g., $\varrho_n - \varrho_{n-1} = 0.1$) and the number of suboptimal attractive fixed points grows significantly. We have found this to be true especially where the asymmetry is purely in the weight vectors.

5.4.2 The initial conditions

Since the largest basin of attraction to the suboptimal fixed point is found for learning scenarios with degenerate teacher biases, we will investigate the influence of the other initial conditions and the learning rates for the task $T_{nm} = \delta_{nm}$ and $\varrho_n = 0$.

In Figure 5.7 it is shown that the influence of the initialization of the first bias θ_1 consists almost exclusively of a linear shift in the range of initial θ_2 values that lead to convergence of the training. In particular, we find that the results become invariant under the transformation $\theta_2' = \theta_2 - 0.9745(9) \times \theta_1$, i.e., the basin of attraction depends almost solely on the difference $\theta_2 - \theta_1$. This is somewhat surprising since one may have assumed that the basin of attraction should depend on the individual abscissas or the effective biases of the student.

In Figure 5.8 the basin of attraction for different initial student lengths is shown. All

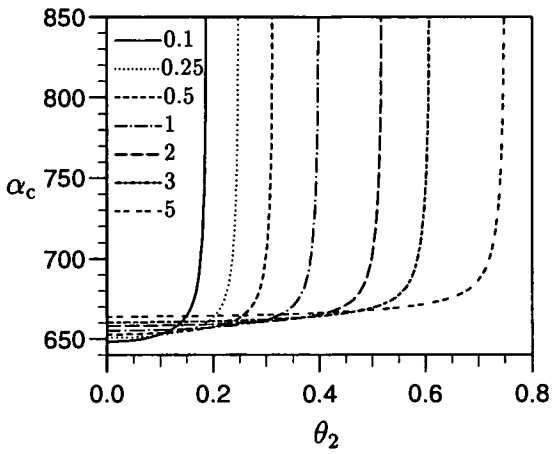

Figure 5.8. The basin of attraction for initial θ_2 shown for several magnification factors M of the initial student-student overlaps Q_{ij} (see the legend) increases with the size of these initial values.

the initial student-student overlaps were magnified from their original values[1] by factors M given in the legend. The influence of the student lengths is clearly twofold. First, the basin of attraction in θ_2 grows approximately with $0.068(5) + 0.331(6) \times M^{0.445(8)}$, making the training process less sensitive to the initial bias values. However, this growth translates into a decrease of the critical abscissa since Q grows with M, which could be interpreted as another sign that the raw initial values are the crucial parameters and not the abscissas. Second, the optimal convergence time is slowed down slightly for increasing M and one finds approximately $\alpha_c^{\rm opt} = 643(1) + 12(1) \times M^{0.34(3)}$.

Similarly in Figure 5.9, we assess the influence of finite size effects on the basin of attraction through the typical initial normalized student-teacher overlaps $\hat{r} = \mathcal{O}(1/\sqrt{N})$ (ignoring other stochastic finite size effects). As predicted in (Biehl et al. 1996), the optimal convergence time is reduced linearly in $\log(\hat{r})$ [$\alpha_c = 187.70(7) - 16.923(4) \times \log(\hat{r})$]. More relevant for the purpose of this investigation is the increase in the basin of attraction to the optimal solution with the critical initial bias $\theta_c = 0.370(1) + 0.507(5) \times \hat{r}^{0.103(1)}$.

The results found for $K = 2$ again carry over qualitatively to larger networks with the decrease in the basin of attraction with network size as already mentioned in Section 5.4.1. Especially interesting in this respect is, that even for $K = 2$, the maximal initial abscissas that guarantee convergence for the case of degenerate teacher biases are

[1]The increase in Q_{ii} leads to a rescaling of the overlaps R_{in} since the normalized overlaps $\hat{R}_{in}$ were randomly fixed. Note also, that similar results are obtained when increasing the initial student lengths individually.

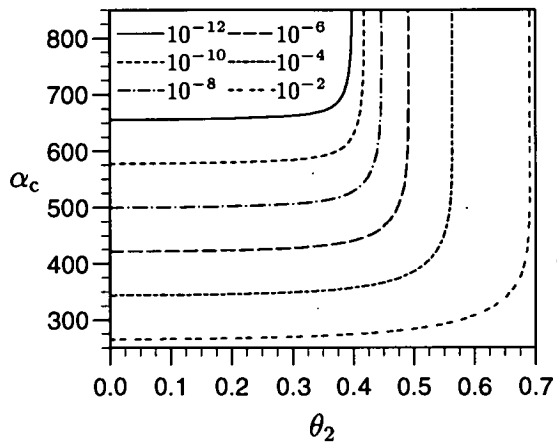

Figure 5.9. Although the basin of attraction for initial θ_2 grows with the range of initial student-teacher overlaps $\hat{r}$ (for values see the legend) the dynamics still get trapped in a suboptimal configuration for large enough θ_2. Since $\hat{r} \sim 1/\sqrt{N}$, this gives some indication how finite size systems may behave.

generally smaller than the size of the input domain, a tendency which becomes more emphasized for larger networks. These results therefore contradict heuristics presented by Nguyen and Widrow (1990), where it has been suggested to spread the abscissas across the input domain. Nguyen and Widrow (1990) also have assumed implicitly that the abscissas are the relevant quantities, whereas the theoretical framework applied here indicates that the raw bias values are more important in determining the basin of attraction.

5.4.3 The learning rates

Beside the initial conditions and the teacher task to be learned, the learning rates used also strongly influence the learning process. In Figure 5.10 the convergence time as a function of θ_2 is shown for a range of common learning rates η_0. For convenience, the convergence time has been normalized with $1/\eta_0$. One finds that the convergence time diverges for all learning rates, above a critical initial value of θ_2. For increasing learning rates, this transition first becomes sharper and occurs at smaller θ_2 until the learning rate is reached that provides the fastest convergence to the optimal solution for small θ_2, beyond which the basin of attraction widens again.

The increase of the basin of attraction has been postulated by Kim and Ra (1991), however, the functional relationship given ($\eta_0 < Q_{ii} + \theta_i^2$) cannot be supported by our findings. It is not only quantitatively incorrect, it also fails to predict a finite

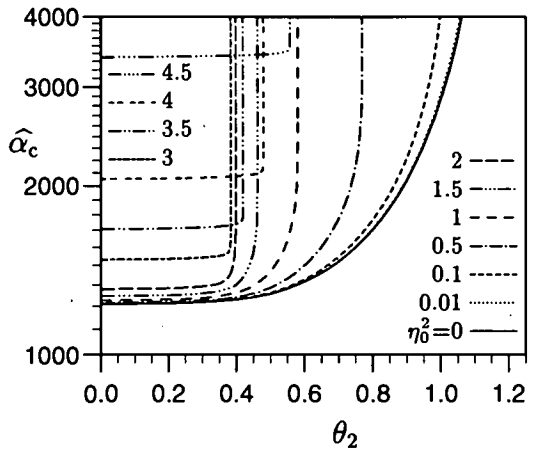

Figure 5.10. The normalized convergence time $\widehat{\alpha_c} \equiv \eta_0 \alpha_c$ is shown as a function of the initialization of θ_2 for various learning rates η_0 (see the legend, $\eta_0^2 = 0$ represents the dynamics neglecting η_0^2 terms.).

boundary for an infinitesimal small learning rate. This work further does not account for interaction between the hidden units and the different rôles of weights and biases in determining the basin of attraction (see Section 5.4.2).

In Figures 5.11 and 5.12 it is shown that it can be beneficial to separate the weight and bias learning rates. In Figure 5.11 the normalized convergence time $\widehat{\alpha_c}(\theta_2)$ is plotted for fixed bias learning rate ($\eta_\theta = 2$) but allowing for variations in the weight learning rate η_w. One can readily see that the basin of attraction increases when the weight and bias learning rates are well separated. This advantage, however, is relative as a very small weight learning rate increases the convergence time linearly.

Similarly in Figure 5.12, the convergence time $\alpha_c(\theta_2)$ is shown for fixed weight learning rate ($\eta_w = 2$) but variable bias learning rate η_θ. Again, the basin of attraction is clearly enlarged when separating the time scale for the training of biases and weights. Whereas training is slowed down for small bias learning rates, this is not the case for large η_θ where the basin of attraction increases to very large values. It is therefore more reasonable to achieve the desirable separation of the learning rates by choosing a large bias learning rate. In fact, a maximal bias learning rate does not exist in this scenario, suggesting a possible different scaling. It further poses the question whether in this case the basin of attraction encompasses the whole space of initial conditions.

Unfortunately, a closer inspection using larger networks and other learning tasks reveals several limitations of large bias learning rates and adiabatic elimination. First of all, the use of adiabatic elimination for very small α leads to extremely large initial

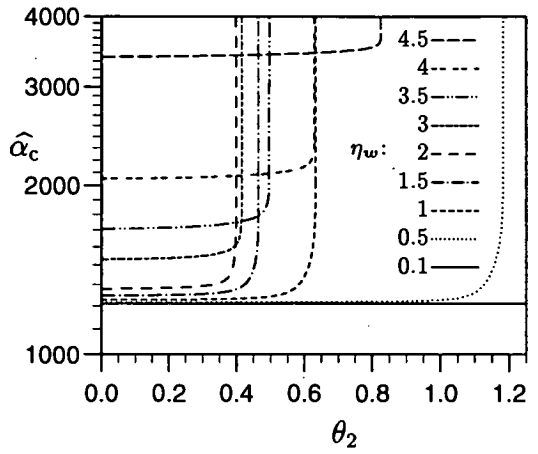

Figure 5.11. The normalized convergence time as a function of θ_2 is shown for various weight learning rates η_w (see the legend) with the bias learning rate fixed at $\eta_\theta = 2$. For very small weight learning rate the basin of attraction increases quickly (for $\eta_w = 0.1$ the training diverges for $\theta_{\rm crit} = 5.415$).

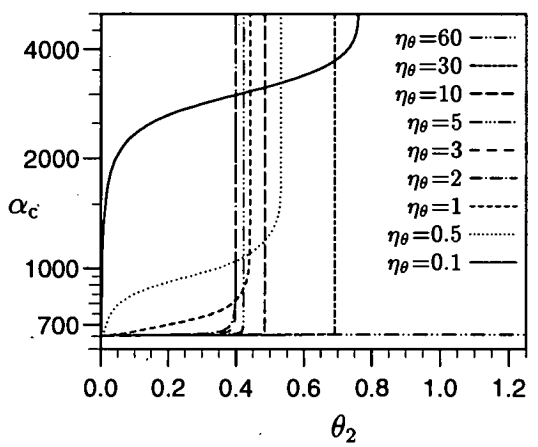

Figure 5.12. The convergence time $\alpha_c(\theta_2)$ is plotted for various bias learning rates η_θ (see the legend) with the weight learning rate fixed at $\eta_w = 2$. For very large bias learning rate the basin of attraction extends to very large values, e.g., to $\theta_{\rm crit} = 5.735$ for $\eta_\theta = 60$, although the training is still eventually slowed down exponentionally for very large initial values of θ_2.

equilibrium values of opposing signs for the biases, effectively cancelling the outputs of pairs of hidden units. This effect can be attributed to the initial lack of information about the teacher, reflected by the inherently small values of the student-teacher overlaps R_{in} favouring the hidden units to be switched off effectively. Consequently, the

progress of the student weights is inhibited to such an extent that training does not converge in finite time for all practical purposes[2]. Similarly, very large but finite bias learning rates also slow down the training time due to the biases blowing up in the very early stages of learning. It is therefore necessary to restrict the bias learning rate for very small α, i.e., for the initial transient, to a finite value. It is unclear, whether this is also a problem for finite size systems where adiabatic elimination corresponds to a bias learning rate of $\mathcal{O}(1)$ instead of $\mathcal{O}(1/N)$.

Even when adiabatic elimination or a very large bias learning rate are only triggered once training has reached the stable symmetric plateau, their usefulness in terms of basin of attraction enlargement is, in general, not pronounced for larger networks. In fact, using large bias learning rates can actually decrease the basin of attraction to the optimal network parameters especially in degenerate bias tasks with isotropic weight vectors, e.g., training with a bias learning rate above $\eta_\theta = 3$ in the learning scenario of Figure 5.2 converges to a suboptimal fixed point.

However, once all hidden unit symmetries have been broken, adiabatic elimination or a very large bias learning rate can be employed in all circumstances and generally results in slightly faster training when compared to using a finite learning rate. This will be investigated analytically in more detail in the following section.

5.5 Analysis of the convergence phase

For a more thorough analysis of the dependence of the learning curves with respect to the number of hidden units, the teacher tasks and the chosen learning rate, it is necessary to restrict the number of parameters in the model such that an analytical treatment becomes feasible. In the case of the soft-committee machine model with fixed zero biases, it was suggested in (Saad and Solla 1995b) to study the case of realizable learning scenario ($K = M$), and isotropic teachers ($T_{nm} = T\delta_{nm}$), where the order parameter space can be very well characterized throughout the learning process by similar diagonal and off-diagonal elements of the overlap matrices $\mathbf{Q}$ and $\mathbf{R}$, simplifying a linear analysis around the symmetric and zero generalization error fixed points considerably since the number of dynamic variables can be reduced to at most four.

For the model with dynamic biases this dimensionality reduction for the equivalent teacher task with isotropic weights and degenerate biases is in general not a good

[2]For adiabatic elimination of the hidden-output weights one finds similarly that the outputs of the student hidden units are suppressed initially by an equilibrium of the output weights close to 0 (Rattray and Saad 1997a; Rattray 1997). However, this does not inhibit the progress of the student as in the case of the biases.

approximation as can be clearly seen in Figure 5.2. However, if the student biases are initialized quite symmetrically, we find the ansatz

$$Q_{ij} = Q\delta_{ij} + C(1-\delta_{ij}), \quad R_{in} = R\delta_{in} + S(1-\delta_{in}), \quad \text{and} \quad \theta_i = \theta \tag{5.14}$$

to be justified for the student-student overlaps, (apart from a relabelling of the student nodes) student-teacher overlaps, and the student biases in both the symmetric and convergence phase. For the symmetric phase an analysis is still infeasible, since analytic expressions for the symmetric fixed points cannot be solved in closed form. However, since the symmetric phase is especially a problem for the teacher with fixed biases, this phase will be analysed in detail in Chapter 6.

For the convergence phase, the reduction of the number of order parameters from $\mathcal{O}(K^2)$ to just five allows us to analyse the learning dynamics as a function of the network size K, the length of the teacher hidden units T, the size of the teacher biases ϱ, and the user adjustable learning rates η_0 and η_θ.

5.5.1 The eigenvalue spectrum

In order to predict the optimal learning rates for the convergence phase, we linearize the equations of motion (5.A.4) in $\{R, Q, C, S, \theta\}$ around the zero generalization error fixed point $R^* = Q^* = T$, $S^* = C^* = 0$ and $\theta^* = \varrho$ (see Appendix 5.B). The matrix $\mathbf{M}$ of the resulting system of five coupled linear differential equations in $r = T - R$, $q = T - Q$, $s = S$, $c = C$ and $\vartheta = \varrho - \theta$ has two sets of eigenvalues.

Two eigenvalues ($\lambda_{1,2}$) are the solutions to a quadratic equation (5.B.3) consisting of the same matrix elements of $\mathbf{M}$ as in the fixed bias case and are therefore independent of the bias learning rate η_θ. These eigenvalues are nonlinear in the learning rate η_w and λ_1 becomes positive for large enough η_w. The other three eigenvalues ($\lambda_{3,4,5}$) are the solution to a cubic equation (5.B.4). These eigenvalues depend on both learning rates and are negative for all values of η_w and η_θ. These eigenvalues are minimized with respect to η_θ in the limit $\eta_\theta \to \infty$, i.e., the optimal bias learning rate in the convergence phase is at infinity (for a more detailed discussion see Appendix 5.B). Below, we will therefore restrict ourselves to the study of two learning rate parameterizations: a common learning rate $\eta_0 = \eta_w = \eta_\theta$ or the weight learning rate η_w with the bias learning rate η_θ eliminated by taking the limit $\eta_\theta \to \infty$. We will adopt the convention to use a generic learning rate η and eigenvalues λ whenever a statement is applicable for both parameterizations, whereas parameterization dependent symbols denoted by superscripts or subscripts are used otherwise.

The behaviour of the eigenvalues described above is graphically illustrated for both

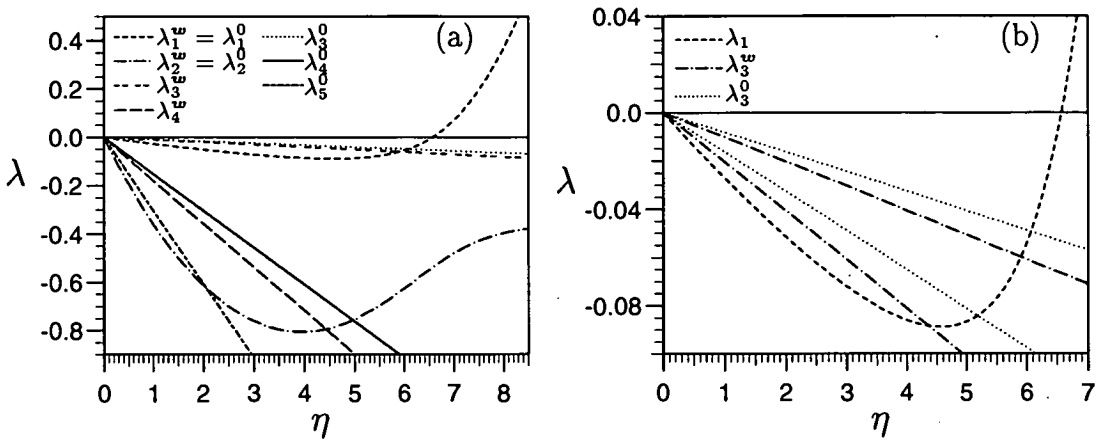

Figure 5.13. (a) The eigenvalues λ_i^0 and λ_i^w are shown as a function of the applicable learning rate η for $K = 5$, $T = 1$ and $\varrho = 1$ for the cases $\eta_\theta = \eta_w = \eta_0$ and $\eta_\theta \to \infty$, respectively. (b) The two relevant eigenvalues (see the text) λ_1 and λ_3 are magnified for the same scenario. For comparison we plot $2\lambda_3$ and find that the optimal learning rate $\eta^{\rm opt}$ is given by the minimum of λ_1 for $\eta_\theta \to \infty$ but by the root of $\lambda_1 - 2\lambda_3$ for $\eta_\theta = \eta_w$.

learning rate parameterizations in Figure 5.13(a) for $K = 5$, $T = 1$, and $\varrho = 1$. Within these parameterizations, the eigenvalues $\lambda_{3,4,(5)}$ are linear in η, whereas $\lambda_{1,2}$ have higher orders in η. $\lambda_{1,2}$ are identical for both parameterizations since they are functions of η_w only, whereas the slopes of $\lambda_{3,4}$ are clearly minimized for the parameterization $\eta_\theta \to \infty$ (λ_5^w is omitted since $\lambda_5 \to -\infty$ for $\eta_\theta \to \infty$). One can further distinguish between two slow modes associated with eigenvalues λ_1 and λ_3 and three fast modes associated with eigenvalues λ_2 and $\lambda_{4,5}$, which are negative for all learning rates and whose magnitude is significantly larger in the region of interesting η. The fast modes decay quickly and their influence on the long-time dynamics is negligible. The dependence of the two relevant eigenvalues λ_1 and λ_3 on η is more closely illustrated in Figure 5.13(b) in the same learning scenario. As mentioned, the eigenvalue λ_3 is negative and linear in η, whereas the eigenvalue λ_1 is a nonlinear function of η and negative for small η. For large η, λ_1 becomes positive and training does not converge to the optimal solution defining the maximum learning rate $\eta_{\rm max}$ as $\lambda_1(\eta_{\rm max}) = 0$. For all $\eta < \eta_{\rm max}$ the generalization error decays exponentionally to $\epsilon_g^* = 0$.

5.5.2 The optimal dynamics

In order to identify the optimal convergence eigenvalue $\lambda^{\rm opt}$, which is the eigenvalue associated with the slowest decay mode, we expand the generalization error to second

order in r, q, s, c, and $\dot{\vartheta}$ (5.B.8). Numerically, we find that the eigenvector associated with the linear eigenvalue λ_3 is orthogonal to the first-order terms in the generalization error and can therefore not contribute to their decay, but controls only the decay of second-order term with $2\lambda_3$.

The learning rate η^{opt} which provides the fastest asymptotic decay rate λ^{opt} of the generalization error is therefore given by the condition

$$\lambda^{\mathrm{opt}} = \left| \min_{\eta} \left[\max \left(\lambda_1, 2\lambda_3 \right) \right] \right| . \tag{5.15}$$

This means either $\lambda_1(\eta_{\mathrm{r}}^{\mathrm{opt}}) = 2\lambda_3(\eta_{\mathrm{r}}^{\mathrm{opt}})$ or $\min_\eta(\lambda_1)$ if $\lambda_1(\eta_{\mathrm{m}}^{\mathrm{opt}}) > 2\lambda_3(\eta_{\mathrm{m}}^{\mathrm{opt}})$, where $\eta_{\mathrm{m}}^{\mathrm{opt}}$ is the learning rate at the minimum of λ_1. In Figure 5.13(b) one finds that for this particular case the fastest decay is achieved at the minimum of λ_1 for $\eta_\theta \to \infty$ but at the root of $\lambda_1 - 2\lambda_3$ for $\eta_\theta = \eta_w$.

Unfortunately, the calculation of λ^{opt} (and η_0 or η_w) via Eq. (5.15) and the determination of the kind of optimum is analytically infeasible for general K, T and ϱ. However, for some special cases further analytical progress can be made: $K \to \infty$, $T \to \infty$ and $T \to 0$. For the T limits, it is necessary to adopt a scaling for the teacher bias ϱ, and we have used both natural scaling ansätze [see Eq. (5.10) in Section 5.2]. These analytic limits are studied in detail in Appendices 5.B.1–5.B.5 and the main results will be referred to in the discussion of the appropriate figures and are summarized in Table 5.1.

The critical teacher length T^{crit}

We find that in the small-T limit, the optimum is always given by the minimum of λ_1 and both learning rate parameterizations are identical, whereas for the large-T limit, the root solution ($\lambda_1 = 2\lambda_3$) applies resulting in a faster decay for $\eta_\theta \to \infty$. For finite T there exists a $T^{\mathrm{crit}}(K, \varrho)$, which depends on the kind of learning rate parameterization and divides these two solution regimes. The functional dependence of T_0^{crit} and T_w^{crit} is graphically illustrated in Figure 5.14 as a function of ϱ for a range of K values including the $K \to \infty$ limit, where it is implicitly assumed that $\exp \varrho^2 \ll K$.

In Figure 5.14(a) T_0^{crit} decreases monotonically with ϱ. The $K \to \infty$ limit exhibits a finite limit ($T_0^{\mathrm{crit}} \approx 0.21$) for $\varrho \to \infty$, but acquires a power-law decay $T_0^{\mathrm{crit}} \propto \varrho^{-2}$ for all finite K [see inset of Figure 5.14(a)]. For $T > T_0^{\mathrm{crit}}(K, 0) \approx 1.278$, the root solution applies for all ϱ due to monotonously decreasing T_0^{crit}, whereas for all other T values the solution type changes from the minimum to the root above a T and K dependent value of ϱ. The dependence of T_0^{crit} on K is relatively weak and varies with ϱ. For small ϱ ($\varrho \lessapprox 0.45$), T_0^{crit} increases with K, whereas for medium ϱ ($0.45 \lessapprox \varrho \lessapprox 1.64$),

T_0^{crit} decreases with K. Above $\varrho \gtrapprox 1.64$, T_0^{crit} increases again with K and reaches the qualitatively different solution for finite and infinite K.

On the other hand, T_w^{crit} does not behave monotonically in ϱ (with the exception of $K = 2$) as shown in Figure 5.14(b). It also decreases initially like T_0^{crit} up to $\varrho \approx 1.3$, but then increases up to a maximum whose height and position increases in K, before it falls towards the asymptotic value of $T_w^{\text{crit}}(K, \infty) = 1/2$ for all finite K. We again find a qualitatively different behaviour for $K \to \infty$ as T_w^{crit} grows unabatedly with ϱ. Depending on the value of K and T, the type of solution can therefore change up to three times for increasing ϱ. Similar to T_0^{crit}, we also find that the T_w^{crit} grows with K initially ($\varrho \lessapprox 0.52$), then decreases ($0.52 \lessapprox \varrho \lessapprox 1.97$) and then increases again.

It is also clear from the graphs and from the fact that $\lambda_3^w \leq \lambda_3^0$, that T_w^{crit} must be greater than T_0^{crit} for all K and ϱ besides $\varrho = 0$ where $T_w^{\text{crit}} = T_0^{\text{crit}}$. We can therefore

Table 5.1. For $T \to 0$ and $T \to \infty$ the optimized dynamics in the convergence phase show power-law behaviour in leading order (for more detail including higher-order terms consult Appendix 5.B) for both learning rate parameterizations $\eta_\theta = \eta_w$ and $\eta_\theta \to \infty$. The table shows the power laws and the $\hat{\varrho} = \varrho/\sqrt{1+T}$ dependence of the optimal learning parameters η_w^{opt} and η_0^{opt}, their respective optimal convergence eigenvalue λ_w^{opt} and λ_0^{opt} and the normalized difference between maximal and optimal learning rate $\Delta\eta_{\max}^{\text{opt}} = (\eta_{\max} - \eta^{\text{opt}})/\eta^{\text{opt}}$. Note that for the $T \to 0$ limit both learning rate parameterizations are identical. In this limit, an alternative scaling for the biases ($\check{\varrho} = \varrho/\sqrt{T}$) has been investigated as well.

	$T \to \infty$ (K finite)		$T \to \infty$ $[TK^{-1} = \mathcal{O}(1)]$	
	$\eta_\theta = \eta_w$	$\eta_\theta \to \infty$	$\eta_\theta = \eta_w$	$\eta_\theta \to \infty$
η^{opt}	$\pi\sqrt{2}K$	$\pi\sqrt{2}K$	$T^{\frac{1}{2}}\,\mathrm{e}^{\frac{1}{2}\hat{\varrho}^2}$	$T^{\frac{1}{2}}\,\mathrm{e}^{\frac{1}{2}\hat{\varrho}^2}$
$\Delta\eta_{\max}^{\text{opt}}$	$[T(1+\hat{\varrho}^2)]^{-1}$	T^{-1}	$[T(1+\hat{\varrho}^2)]^{-1}$	T^{-1}
λ^{opt}	$T^{-\frac{3}{2}}(1+\hat{\varrho}^2)^{-1}\,\mathrm{e}^{-\frac{1}{2}\hat{\varrho}^2}$	$T^{-\frac{3}{2}}\,\mathrm{e}^{-\frac{1}{2}\hat{\varrho}^2}$	$[TK(1+\hat{\varrho}^2)]^{-1}$	$(TK)^{-1}$

	$T \to 0$	
	$\eta_\theta \geq \eta_w$ $(\check{\varrho})$	$\eta_\theta \geq \eta_w$ $(\hat{\varrho})$
η^{opt}	π	$\pi\,\mathrm{e}^{\hat{\varrho}^2}$
$\Delta\eta_{\max}^{\text{opt}}$	$T\sqrt{1+2\check{\varrho}^2}$	$\sqrt{\hat{\varrho}^2 T}$
λ^{opt}	$T^2K^{-1}(1+2\check{\varrho}^2)$	$TK^{-1}\hat{\varrho}^2$

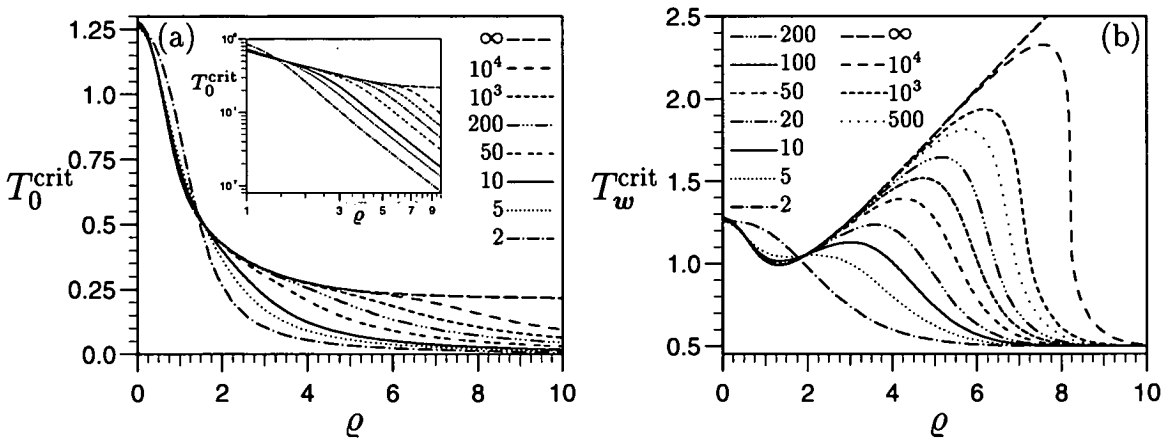

Figure 5.14. The critical teacher lengths $T_0^{\rm crit}$ (a) for $\eta_\theta = \eta_w$ and $T_w^{\rm crit}$ (b) for $\eta_\theta \to \infty$ as a function of $\hat{\varrho}$ for several K values given in the legend (∞ represents the $K \to \infty$ limit). $T^{\rm crit}$ defines the transition between the optimal convergence given by the minimum of λ_1 and by the root of $\lambda_1 - 2\lambda_3$. Notice that for given T, the solution type can change for increasing $\hat{\varrho}$ at most once for $\eta_\theta = \eta_w$, whereas it can change up to three times for $\eta_\theta \to \infty$. The inset in (a) shows the power-law decay of $T_0^{\rm crit} \propto \hat{\varrho}^{-2}$.

divide the optimal convergence behaviour for all K, T, and ϱ into three regimes:

1. $T \leq T_0^{\rm crit}(K,\varrho) \leq T_w^{\rm crit}(K,\varrho)$: The minimum of λ_1 defines the optimum and both learning rate parameterizations behave identically ($\lambda_w^{\rm opt} = \lambda_0^{\rm opt}$ and $\eta_w^{\rm opt} = \eta_0^{\rm opt}$).

2. $T_0^{\rm crit}(K,\varrho) < T < T_w^{\rm crit}(K,\varrho)$: The optimal solution is different for both parameterization. The minimum of λ_1 is still optimal for $\eta_\theta \to \infty$, but $\lambda_1 - 2\lambda_3 = 0$ defines the optimum for $\eta_\theta = \eta_w$. The optimal convergence rates and learning rates are different with $\lambda_w^{\rm opt} > \lambda_0^{\rm opt}$ and $\eta_w^{\rm opt} < \eta_0^{\rm opt}$.

3. $T_0^{\rm crit}(K,\varrho) \leq T_w^{\rm crit}(K,\varrho) < T$: Although the optimal solution is now the root of $\lambda_1 - 2\lambda_3$ for both parameterizations, we still find $\lambda_w^{\rm opt} \geq \lambda_0^{\rm opt}$ and $\eta_w^{\rm opt} \leq \eta_0^{\rm opt}$ since $\lambda_3^w \leq \lambda_3^0$.

Since the 3-dimensional parameter space is difficult to visualise, we study the optimal convergence exemplary for two slices, K-ϱ and ϱ-T, since we are mainly interested in the dependence of the convergence dynamics on ϱ. A more thorough investigation into the K-T space is deferred to Chapter 6 for the special case of $\varrho = 0$.

Optimal dynamics in K-ϱ space

In Figure 5.15 we show the convergence behaviour of the parameterization $\eta_\theta = \eta_w = \eta_0$ [Figures 5.15(a–c)] in comparison to $\eta_\theta \to \infty$ [Figures 5.15(d–f)] as a function of K for $T = 1$ and a range of ϱ values. In Figure 5.15(a), one can see that the optimal learning rate $\eta_0^{\rm opt}$ is hardly K dependent for small ϱ (beside the inherent rescaling with $1/K$ implied by the normalization of the soft-committee machine), but increases proportionally to K for large ϱ before it eventually levels off at a ϱ dependent value. The $K \to \infty$ analysis suggests a scaling of the optimal learning rate with $\log \eta_0^{\rm opt} \propto \varrho^2$ since the maximal learning rate scales in this fashion. This is mirrored in the behaviour of the optimal convergence rate in Figure 5.15(b) (for graphical purposes multiplied by K) which exhibits the expected $1/K$ behaviour for small ϱ. For large ϱ, however, the increase in $\eta_0^{\rm opt} \propto K$ for small K causes $\lambda_0^{\rm opt}$ to be constant until $\eta_0^{\rm opt}$ levels off, when $\lambda_0^{\rm opt}$ reverts back to the $1/K$ decay. We further note that the absolute value of the convergence rate $\lambda_0^{\rm opt}$ initially increases for small ϱ for all values of K, which is a T-dependent effect we will study in more detail below. In Figure 5.15(c) we further show the normalized difference between the maximal and optimal learning rate defined as

$$\Delta\eta_{\rm max}^{\rm opt} = \frac{\eta_{\rm max} - \eta^{\rm opt}}{\eta^{\rm opt}}.$$

We find that $\Delta\eta_{0\,\rm max}^{\rm opt}$ initially increases with ϱ for all K, which is again a feature dependent on T, before it decreases monotonically, reflecting a steeper and more skewed curve for λ_1.

To compare the two learning rate parameterizations, the ratio of the optimal learning rates $\eta_w^{\rm opt}$ and $\eta_0^{\rm opt}$ shown in Figure 5.15(d) shows that for small ϱ the ratio is identical since $T = 1 < T_0^{\rm crit} < T_w^{\rm crit}$. For increasing ϱ the ratio falls below 1 since $\eta_0^{\rm opt}$ is now determined by the root of $2\lambda_3 - \lambda_1$ ($T_0^{\rm crit} < T < T_w^{\rm crit}$). Increasing ϱ even further, one finds that also $\eta_w^{\rm opt}$ is determined initially by the root solution ($T_0^{\rm crit} < T_w^{\rm crit} < T$). For larger K one finds kinks in the curves when the ratio approaches $1/2$. A ratio of $1/2$ suggests for an assumed quadratic eigenvalue λ_1, that $\eta_0^{\rm opt}$ is close to the maximal learning rate $\eta_{\rm max}$, whereas $\eta_w^{\rm opt}$ is close to the minimum located at $\eta_{\rm max}/2$. The kinks therefore coincide with a change to $T_0^{\rm crit} < T < T_w^{\rm crit}$ above a value of K dependent on ϱ [e.g., for $\varrho = 6$ the kink is at $K \approx 100$, which coincides with $T^{\rm crit}(100, 6) \approx 1$ as can be seen in Figure 5.14(b)]. For even larger ϱ this solution change is pushed out to larger values of K.

The ratio of the optimal convergence rates $\lambda_w^{\rm opt}$ and $\lambda_0^{\rm opt}$ shown in Figure 5.15(e) reflects above observations. For small ϱ the minimum of λ_1 is optimal and the ratio is 1. Even for larger T values, where the root solutions apply for $\varrho = 0$, ratios very close to 1

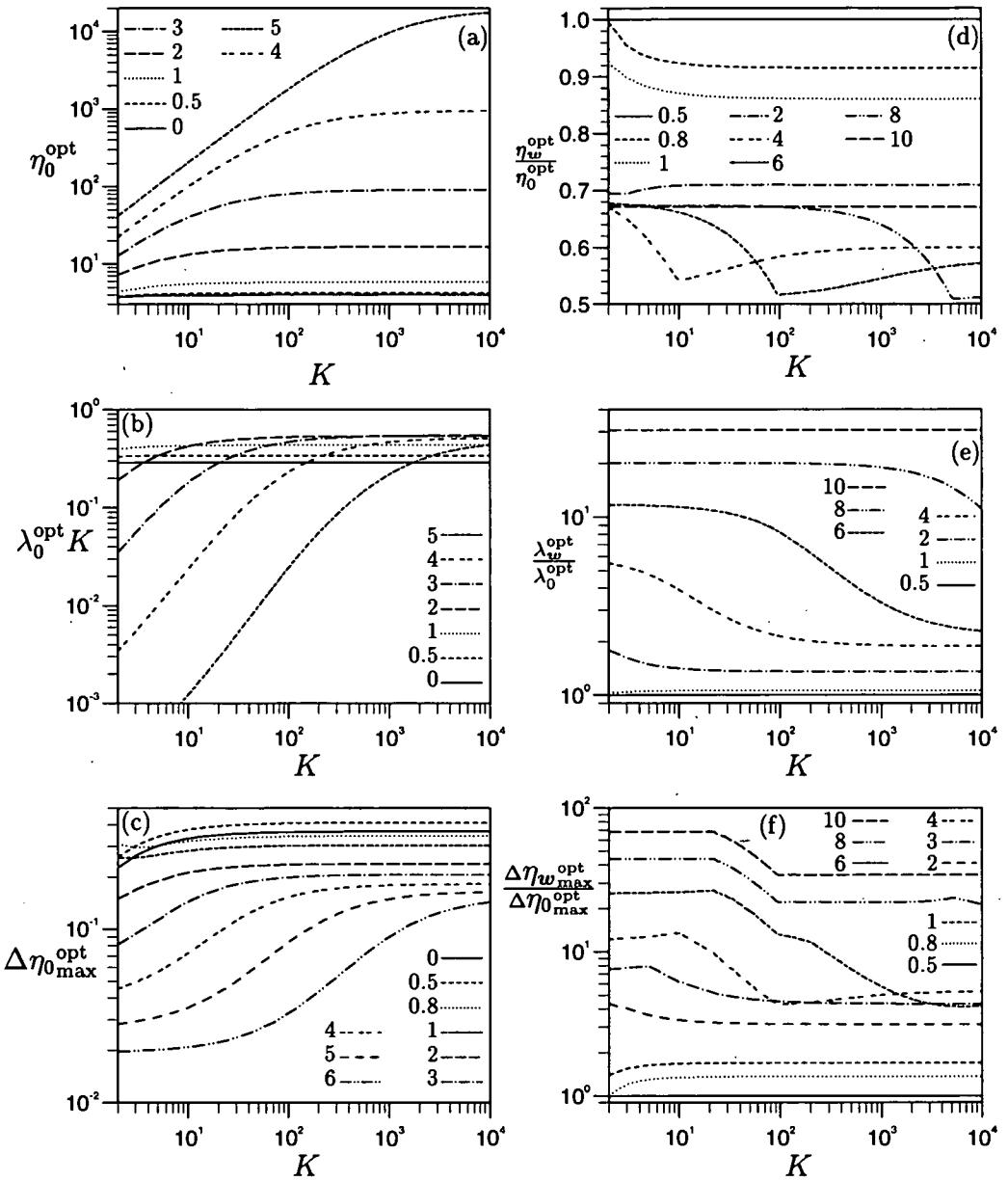

Figure 5.15. The convergence scenario as a function of K for $T = 1$ and various ϱ given in the legends. (a) Optimal learning rate $\eta_0^{\rm opt}$ for $\eta_\theta = \eta_0$. (b) Optimal convergence rate $\lambda_0^{\rm opt}$, multiplied by K for convenience. (c) The normalized difference between the optimal and maximal learning rates $\Delta\eta_{0\rm max}^{\rm opt}$. Ratio of the optimal learning rates $\eta_w^{\rm opt}$ and $\eta_0^{\rm opt}$ (d), the optimal convergence rates $\lambda_w^{\rm opt}$ and $\lambda_0^{\rm opt}$ (e), and the normalized differences $\Delta\eta_{w\rm max}^{\rm opt}$ and $\Delta\eta_{0\rm max}^{\rm opt}$ (f).

are observed for small ϱ. For larger ϱ, however, the root solutions apply either for both learning rate parameterizations or at least for $\eta_\theta = \eta_w$ and the widening gap between λ_1 for the two learning rate parameterizations leads to ratios above 1 increasing with ϱ. The benefit achievable is, however, limited eventually for large K when the optimal convergence of the $\eta_\theta \to \infty$ parameterization reverts back to the minimum of λ_1.

This behaviour holds similarly for the ratio of the normalized separation of maximal and optimal learning rates $\Delta\eta_{w\,\mathrm{max}}^{\mathrm{opt}}$ and $\Delta\eta_{0\,\mathrm{max}}^{\mathrm{opt}}$ [Figure 5.15(f)]. The widening gap between λ_1 increases the ratio significantly above 1, once η_0^{opt} is given by the root solution. The non-monotonic behaviour for some of the lines in Figure 5.15(f) can be explained by the change in the degree of skewness of λ_1 away from a parabolic form when the minimum solution applies for η_w^{opt}.

Optimal dynamics in ϱ-T space

When considering the optimal dynamics as a function of ϱ and T, two natural scaling ansätze for the bias ϱ present themselves (see discussion in Section 5.2), which become especially relevant in the limits $T \to \infty$ and $T \to 0$. The first ansatz ($\varrho = \hat{\varrho}\sqrt{1+T}$), here termed effective bias, fixes the mean hidden unit output independent of T, the other ansatz ($\varrho = \check{\varrho}\sqrt{T}$), here termed abscissa, keeps the distance of the decision hyperplane to the origin constant. For large $T \gg 1$, both ansätze become identical to leading orders. For small T, however, there are significant differences. In this section we have adopted $\hat{\varrho}$ as the preferred variable since it results in the more universal behaviour for finite T, but we will discuss their differences in detail in Section 5.5.3.

In Figure 5.16 the influence of different teacher length values T is studied, where the convergence behaviour of the parameterization $\eta_\theta \to \infty$ [Figures 5.16(a–c)] is shown as a function of $\hat{\varrho}$ for $K = 10^2$ and a range of T values (including theoretical predictions from asymptotic analyses when useful). Figure 5.16(a) shows that the optimal learning rate increases exponentionally in $\hat{\varrho}^2$. For small $\hat{\varrho}$, the prefactor of the exponentional increase approaches $1/2$ for large T, whereas it approaches 1 for small T, in agreement with the prediction from the $K \to \infty$ and $T \to 0$ analyses [included in Figure 5.16(a)]. For larger $\hat{\varrho}$, however, one finds a prominent change in the slope of the η_w^{opt} curves, where the position of the transition and its significance is dependent on T. For very small but finite T this transition is beyond the range of the graph and the change in the slope becomes less significant. The limiting behaviour is in agreement with the $T \to 0$ analysis [included in Figure 5.16(a)]. For finite T, η_w^{opt} still increases exponentionally in $\hat{\varrho}^2$ after the transition, but the constant prefactor in the exponent is altered and decreases for large T. The limiting behaviour is in agreement with the findings of the $T \to \infty$ analysis for finite K in Appendix 5.B.5, which predicts a finite limit of η_w^{opt} for

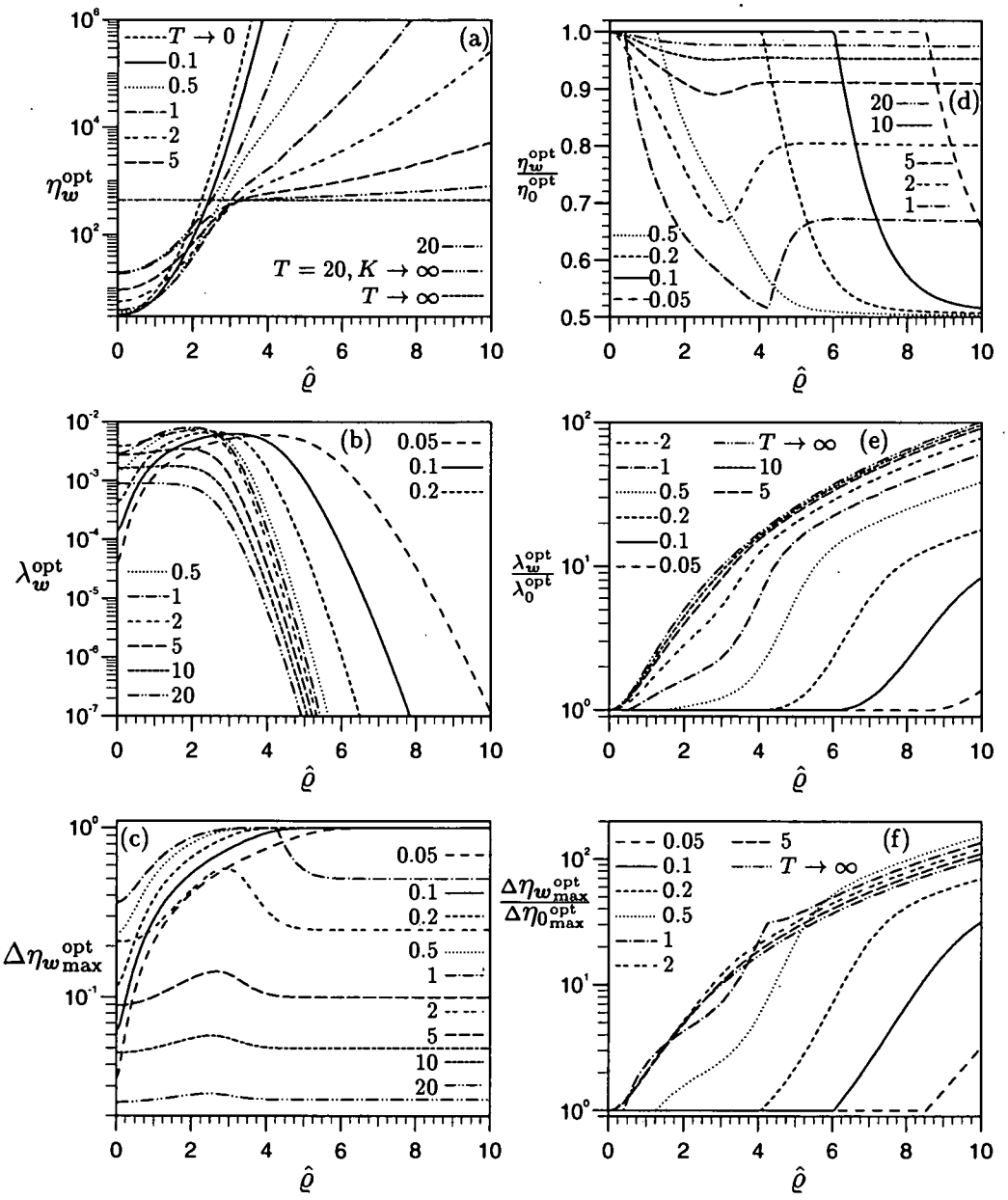

Figure 5.16. The convergence scenario as a function of $\hat{\varrho}$ for $K = 10^2$ and various T given in the legends including predictions from expansions for $T \to 0$, $T \to \infty$. (a) Optimal learning rate η_w^{opt} for $\eta_\theta = \infty$. (b) Optimal convergence rate λ_w^{opt}. (c) The normalized difference between the optimal and maximal learning rates $\Delta\eta_{w\,max}^{opt}$. Ratio of the optimal learning rates η_w^{opt} and η_0^{opt} (d), the optimal convergence rates λ_w^{opt} and λ_0^{opt} (e), and the normalized differences $\Delta\eta_{w\,max}^{opt}$ and $\Delta\eta_{0\,max}^{opt}$ (f).

large $\hat{\varrho}$ also shown in Figure 5.16(a).

The dependence of the optimal convergence eigenvalue $\lambda_w^{\rm opt}$ shown in Figure 5.16(b) is similarly intriguing. One finds that the convergence rate increases initially with $\hat{\varrho}$ up to maximum, whose position shifts to larger $\hat{\varrho}$ values for decreasing T and becomes flatter for increasing T. Beyond the maximum, $\lambda_w^{\rm opt}$ decreases exponentionally in $\hat{\varrho}^2$, with the prefactor in the exponentional increasing with T, but saturating at $1/2$ as predicted from the $T \to \infty$ analysis. The small T expansion predicts the steep initial increase in $\lambda_w^{\rm opt}$ correctly, as the order of the optimal convergence rate for non-zero $\hat{\varrho}$ is not $\mathcal{O}(T^2/K)$ as for zero $\hat{\varrho}$ but $\mathcal{O}(T/K)$. The expansion is a good approximation for small finite T and small $\hat{\varrho}$ but breaks down for larger $\hat{\varrho}$, where the optimal convergence rate $\lambda_w^{\rm opt}$ reaches a almost T independent maximum of $\mathcal{O}(1/K)$ and can also not account for the eventual exponentional decrease of $\lambda_w^{\rm opt}$ with $\hat{\varrho}$ beyond the maximum. This failure is caused by the implicit assumption $\hat{\varrho}^2 \ll -\log T$ in the $T \to 0$ limit which shifts the maximum in $\lambda_w^{\rm opt}$ to $\hat{\varrho} = \infty$. For larger network sizes K not shown here, one finds that the position of the maximum shifts to larger $\hat{\varrho}$ and becomes flatter. This effect leads to the shift of the maximum to $\hat{\varrho} = \infty$ in the $K \to \infty$ expansion.

The behaviour of the normalized separation $\Delta\eta_w{}_{\rm max}^{\rm opt}$ in Figure 5.16(c) reflects the kind of solution present. For small $T < T_w^{\rm crit}$, the minimum of λ_1 is optimal and $\Delta\eta_w{}_{\rm max}^{\rm opt}$ increases monotonically towards 1, i.e., λ_1 becomes parabolic for large $\hat{\varrho}$. For $T = 1$, we find the same behaviour for small $\hat{\varrho}$, but find a prominent kink at $\hat{\varrho} \approx 4.25$ [i.e., $\varrho \approx 6$, see Figure 5.14(b)], which coincides with $T^{\rm crit} = 1$. For $\hat{\varrho} > 4.25$, $T^{\rm crit} < 1$ and $\Delta\eta_w{}_{\rm max}^{\rm opt}$ falls to a constant below 1. For larger T, the behaviour is similar but smoother in comparison to $T = 1$, reflecting the fact that although the optimal solution is always given by the root, its distance to the minimum changes with $\hat{\varrho}$ as $T_w^{\rm crit}$ rises and falls.

The results for the parameterization $\eta_\theta = \eta_w$ are quite similar to $\eta_\theta \to \infty$ and to enhance the differences we show the ratios of the relevant quantaties in Figures 5.16(d–f). For the optimal learning rate $\eta_0^{\rm opt}$, we also find the change in the exponentional behaviour. For large enough $T > T_0^{\rm crit}$, the ratio of the $\eta_w^{\rm opt}/\eta_0^{\rm opt}$ falls below 1 [see Figure 5.16(d)] and approaches a constant limit for large $\hat{\varrho}$. For medium T (e.g., $T = 1$), the difference is most pronounced, reflecting the many changes in the type of solutions due to the variability of $T_w^{\rm crit}$ and $T_0^{\rm crit}$. For small $\hat{\varrho}$, the minimum solution of λ_1 is optimal for both learning rate parameterizations. In the range of $0.40 \lessapprox \hat{\varrho} \lessapprox 4.25$ (i.e., $0.55 \lessapprox \varrho \lessapprox 6$), $T_0^{\rm crit} < T < T_w^{\rm crit}$ and the ratio drops significantly[3] towards $1/2$ until also $T_w^{\rm crit} < T$ and the ratio rises again towards the asymptotic behaviour.

[3]Note that for $T = 1$, $T_w^{\rm crit}$ also falls briefly below T in the range $0.80 \lessapprox \hat{\varrho} \lessapprox 1.10$ ($1.15 \lessapprox \varrho \lessapprox 1.50$) and $T_0^{\rm crit} < T_w^{\rm crit} < T$. The ratio of the learning rates still drops due to the widenin cap between λ_3^w and λ_3^0 for increasing $\hat{\varrho}$.

The improvement by using a large bias learning rate is reflected in the ratio $\lambda_w^{\rm opt}/\lambda_0^{\rm opt}$ [Figure 5.16(e)], which increases monotonically with $\hat{\varrho}$, for T or $\hat{\varrho}$ large enough so that $T > T_0^{\rm crit}$. In the $T > T_w^{\rm crit}$ region, the ratio $\lambda_w^{\rm opt}/\lambda_0^{\rm opt}$ increases with $a_0 + a_2\hat{\varrho}^2$, where a_0 and a_2 are T dependent constants which approach $a_0 = 1$ and $a_2 = 1$ for large T as predicted by the $T \to \infty$ analysis. Using large η_θ is similarly benefical in the same region of T and $\hat{\varrho}$ with respect to the separation of maximal and optimal learning rates as depicted in Figure 5.16(f). For larger T, we find the same regression behaviour of the ratio $\Delta\eta_{w\,\rm max}^{\rm opt}/\Delta\eta_{0\,\rm max}^{\rm opt}$ with $b_0 + b_2\hat{\varrho}^2$, where b_0 and b_2 are again T dependent constants with the asymptotic limit $1 + \hat{\varrho}^2$ for $T \to \infty$. In the curve for $T = 1$, one observes several swerves and a kink due to $T_0^{\rm crit}$ or $T_w^{\rm crit}$ crossing $T = 1$.

5.5.3 The impact of adaptive biases

The analysis of the convergence phase for non-zero biases has revealed several new insights, which could have not been inferred from the zero-fixed biases case [see (Saad and Solla 1995b) and Chapter 6 for comparison].

For small T, where the training for the zero-bias case is slowed down by a factor $1/T^2$, arguably due to the nearly linear network output making the distinction between different units difficult, one finds that the scaling assumption for the bias has a dramatic impact. This can be understood qualitatively by considering the network output distribution which can be calculated in closed form in the $T \to 0$ limit.

For finite abscissa (using the scaling $\varrho = \check{\varrho}\sqrt{T}$), the hidden unit output distribution is Gaussian with mean $\mu = -\sqrt{2K/\pi}\check{\varrho}\sqrt{T}$ and standard deviation $\sigma = \sqrt{2/\pi}\sqrt{T}$. The probability of a positive (and hence negative) output remains constant for $T \to 0$ and is equal to $H(\check{\varrho}\sqrt{K})$, where $H(x) = \int_x^\infty dx/\sqrt{2\pi}\exp(-x^2/2)$, i.e., even for small T the output of the hidden unit will have some probability of being both negative and positive, but the mean goes to zero. For this scaling, one finds a slight improvement in the convergence rate for non-zero bias by a factor $1 + 2\check{\varrho}^2$, suggesting that breaking the symmetry of the network output distribution around zero is beneficial, but a more significant improvement is not possible since the hidden unit outputs are mainly in the linear regime where the student cannot discriminate efficiently between the teacher hidden units and the convergence rate still decays with T^2.

For finite effective bias (using the scaling $\varrho = \hat{\varrho}\sqrt{1+T}$), the network output distribution is also Gaussian for small T, but with mean $\mu = -\sqrt{K}g(\hat{\varrho})$ and standard deviation $\sigma = \sqrt{2/\pi}\exp(-\hat{\varrho}^2/2)\sqrt{T}$. The probability of an output of opposite sign to the mean output vanishes for $T \to 0$. The single hidden unit output is concentrated in the nonlinear region of the sigmoidal activation function and one could argue that most information about a teacher parameters can be extracted by the student in this

region as long as the hidden units are not too saturated, leading to the improvement in the convergence rate by $\mathcal{O}(\hat{\varrho}^2/T)$.

One could further speculate, that the increase of the optimal learning rate matching the suppression of the gradient is facilitated by the exponentional decrease of the network output variance with $\hat{\varrho}$. For finite T and larger $\hat{\varrho}$, the results for $T \to 0$ expansion become inaccurate for $\hat{\varrho}^2 \not\ll -\log T$ and one finds that the optimal learning rate growth cannot be sustained, leading to the eventual exponentional decay of the convergence eigenvalue with $\hat{\varrho}^2$ as observed for finite K. Due to the T dependence of this breakdown, one even finds the anomaly that training can be momentarily improved when decreasing T slightly [see Figure 5.16(b)].

The unsustainability of the optimal learning rate growth is epitomized in the $T \to \infty$ limit, where the optimal learning rate stays constant for all $\hat{\varrho}$. However, if the $K \to \infty$ limit is taken simultaneously with $T \to \infty$, the convergence rate either remains constant for $\eta_\theta \to \infty$ or decays algebraically with $(1+\hat{\varrho})^2$ for $\eta_\theta = \eta_w$. Similar behaviour is also found for finite T and large K for small enough $\hat{\varrho}$.

The underlying reasons of this difference can be explained most easily for the infinite T case, where the hidden unit output becomes binary and the subsequent network output probability distribution is binomial, as teacher hidden units are uncorrelated. The probability of a single hidden output to be $+1$ parameterizes the binomial distribution and is $1/2[1 - g(\hat{\varrho})]$, i.e., $1/2$ for $\hat{\varrho} = 0$ and decays exponentionally fast for large $\hat{\varrho}$ ($\propto \mathrm{e}^{-\hat{\varrho}^2}$). The corresponding mean and standard deviation are $\mu = -\sqrt{K} g(\hat{\varrho})$ and $\sigma = \sqrt{1 - g^2(\hat{\varrho})}$, respectively. Since both student and teacher network are highly correlated, the error signal should be at most $\mathcal{O}(1/K)$, i.e., at most two hidden units disagree, leading to a possible increase of the learning rate with K. For large effective bias $\hat{\varrho}$, this event becomes exponentionally unlikely and the error signal is identically zero most of the time. The learning rate, however, cannot be increased accordingly since this would lead to an exponentionally large update step size in an error event. The convergence rate has therefore to decay exponentionally. For $K \to \infty$, the binomial output distribution becomes Gaussian with the above mean and variance, leading to smooth network outputs and error signals. Here, the learning rate can be increased exponentionally, which may be linked to the exponentional decrease of the output variance for large $\hat{\varrho}$ combined with the implicit assumption that $\hat{\varrho}^2 \ll \log K$. This behaviour carries over qualitatively to finite T and K for $\hat{\varrho}^2$ small enough, and can explain the initial matching increase of the optimal learning rate and the extension of the region of almost constant convergence rate for larger K.

5.6 Towards more realistic scenarios

The scope of this chapter has so far been restricted in several ways. One obvious restriction has been the fixed hidden-output weights. Although soft-committee machine with biases are universal approximators (West et al. 1997), in practice it is advantageous to use adjustable hidden-output weights. This extension is straightforward in terms of feasibility, but adds a further dimension to the space of parameters to be investigated. We expect our results to be at least qualitatively correct, but we cannot rule out that the dynamics become even richer with more suboptimal fixed points. Unfortunately, the works to date which have allowed for adjustable hidden-output units (Riegler and Biehl 1995; Riegler 1997) have not discussed the issue of hidden unit symmetry breaking.

We have furthermore restricted ourselves to realizable scenarios, where the student network can learn to imitate the teacher network perfectly. In real learning scenarios, one expects both structural unrealizability, due to a mismatch between the function space of the student and the task, as well as unrealizability due to corrupted training data. Both types of unrealizability can be incorporated in this framework, by studying $K \neq M$ and by allowing for noise on the teacher weights and/or outputs, respectively. Both have been addressed already for the soft-committee machine without biases [Saad and Solla (1995b, 1996, 1997)].

Here we will briefly assess the effects arising due to the introduction of adjustable biases in the case of structural unrealizability. In Figure 5.17 the evolution of the training is shown for $K = 3$ and $M = 4$, i.e., when the target function is more complicated than the mapping the student can achieve. The teacher overlaps are $T_{nm} = \delta_{nm}(n+1)/2$ for graded and $T_{nm} = \delta_{nm}$ for isotropic teachers. The teacher biases are $\varrho_n = (2n-5)/5\sqrt{1+T_{nn}}$ for non-degenerate and $\varrho_n = 0$ for degenerate teachers. The common learning rate is always $\eta_0 = 2$ and the weight initialization is $Q_{ii} = (18+n)/100$, $\theta_i = (n-2)/100$, and random overlaps as outlined in Section 5.3. The initialization was chosen quite symmetrically to make differences between the tasks more pronounced and to ensure convergence to a fixed point with the lowest generalization error for the most symmetric task $\mathcal{T}_d^i$.

The main focus will be on the $\mathcal{T}_n^i$ since for this task the effect of non-degenerate teacher biases can be separated from the effect of graded teacher norms. In Figures 5.17(a–c) the evolution of the overlaps Q_{ij}, R_{in} and the biases θ_i is shown. The student is initially drawn into a symmetric phase with similar values for student lengths Q_{ii} and correlations Q_{ij} [Figure 5.17(a)]. This is mirrored by similar student-teacher overlaps R_{in} shown in Figure 5.17(b), signalling the lack of significant specialization with a specific teacher node. The specialization is driven by the student biases depicted

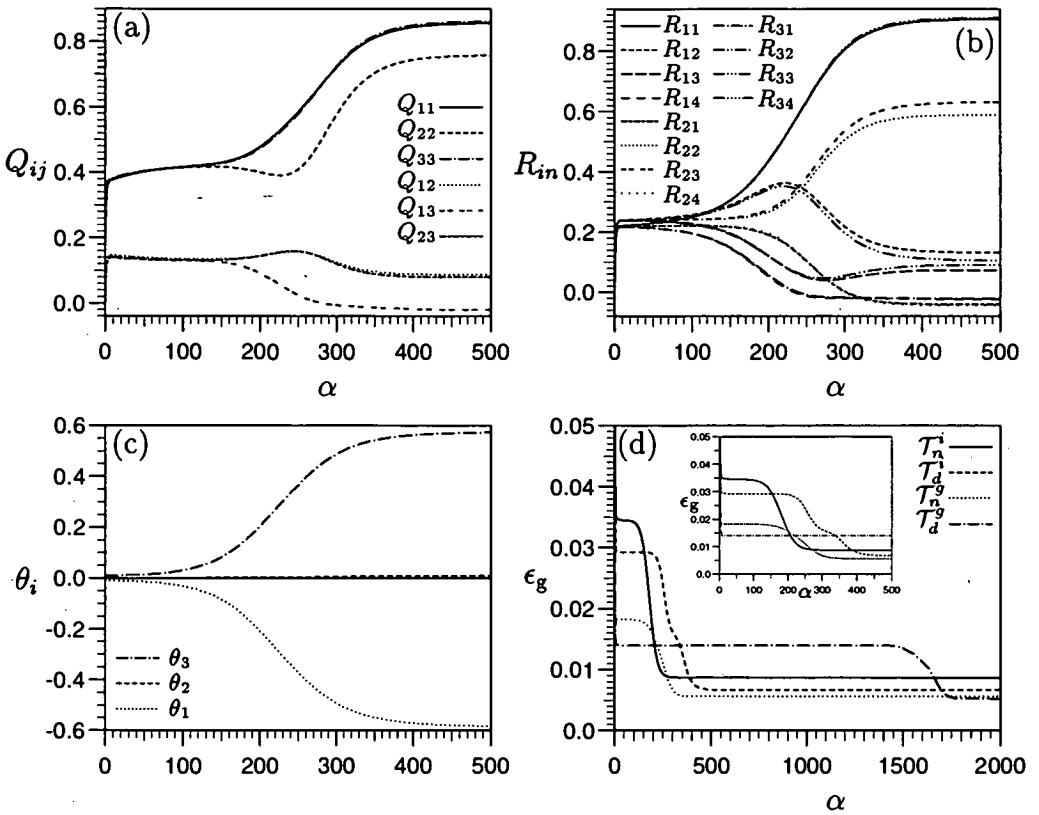

Figure 5.17. A typical training dynamics is shown as a function of α for an unrealizable case $K = 3$ and $M = 4$. The teacher task are of the form: $T_{nm} = \delta_{nm}(n+1)/2$ for graded and $T_{nm} = \delta_{nm}$ for isotropic teachers; $\varrho_n = (2n-5)/5\sqrt{1+T_{nn}}$ for non-degenerate and $\varrho_n = 0$ for degenerate teacher biases. The common learning is always $\eta_0 = 2$. The evolution of the student-student overlaps Q_{ij} (a), the student-teacher overlaps R_{in} (b), and the student biases θ_i (c) are shown for $\mathcal{T}_n^i$. The generalization error ϵ_g (d) is shown for all tasks, with the inset magnifying the escape out of the symmetric phase for the students learning the less symmetric tasks.

in Figure 5.17(c), whose symmetry is broken first and whose trajectories do not cross, although they were initialized quite symmetrically. Since the student network does not have enough resources to model the teacher task adequately, it chooses to dedicate two units (1 and 3) to specialise primarily on the teacher hidden units (1 and 4) with the largest absolute bias value; which is reflected by large R_{11} and R_{34} values and the proximity of the student biases θ_1 and θ_3 to the corresponding teacher biases ϱ_1 and ϱ_4. This seems sensible since these two units have on average the largest (absolute) output. The last student unit 2 specializes almost equally on the two remaining teacher units 2 and 3 (large R_{22}, R_{23} and θ_2 lies between ϱ_2 ϱ_3). The remaining student-teacher overlaps fall

roughly into two groups: The student units (1,4) which are highly specialized on one unit acquire a relatively large overlap with the remaining teacher units (2,3) for which no dedicated student unit exists, whereas they retain only small correlations of either positive or negative sign with those teacher units, which are already modelled almost entirely by another student unit. The size of the individual student-teacher overlaps is also highly correlated with the proximity of the associated student and teacher biases (e.g., $R_{23} > R_{13} > R_{33}$ for fixed teacher unit or $R_{34} > R_{33} > R_{32} > R_{31}$ for fixed student unit). One further notices that the student biases are positioned to ensure that the means of the student and teacher network output distributions (which is just the sum of the means of the individual hidden unit output distributions in a network) are very similar. Matching the mean of the teacher output distribution is obviously a necessary but not sufficient condition for achieving a small generalization error.

Obviously, the specialization process described above is dependent on the teacher task presented. For graded teacher tasks, the larger teacher hidden unit weight vectors lead to a larger variance of their output distributions (and ultimately the output distribution of the whole network). The student hidden unit have therefore to compromise between primarily modelling large variance by specialising on teacher units with large weight norms and large mean by specialising on teacher units with large (effective) biases. We still find that the student biases are positioned to ensure that the mean output is approximately identical, but the student also accounts for larger variances. For degenerate biases, one finds that the dynamics and the optimal attractive fixed point are very similar to the fixed bias case for both graded and isotropic teacher, with the student biases taking values close to the degenerate (effective) teacher bias position[4].

In Figure 5.17(d) the dynamics of the four different generic tasks are compared by following the evolution of the generalization error. As for realizable learning scenarios, one finds that the specialization process for the task $\mathcal{T}_d^i$ is by far the slowest due to the slow breaking of the symmetries. For the task $\mathcal{T}_d^g$ one finds more than one plateau in the generalization error [see inset of Figure 5.17(d)] characteristic of the sequential symmetry breaking for graded teacher lengths. The fastest training is exhibited by the tasks with non-degenerate biases $\mathcal{T}^n$, with a slight speed-up for graded teacher lengths $\mathcal{T}_g^n$. Unlike in realizable scenarios, the dynamics approach a non-zero asymptotic generalization error, which is smallest for the task $\mathcal{T}_d^i$ with most symmetries. For the tasks presented here, the breaking of the bias degeneracy results in a smaller increase

[4]For zero degenerate teacher biases, the student biases converge exactly to zero, whereas for non-zero "degenerate" teacher biases one finds the most self-consistent results for *effective* biases, i.e., degenerate effective teacher biases lead to approximately degenerate effective student biases

of the generalization error than the breaking of length isotropy. This feature, however, depends on the particular choice of teacher norms and biases.

Similar to the realizable case, we also find that the dynamics are sensitive to the initial conditions, especially for tasks with many symmetries such as $\mathcal{T}_d^i$, and the asymptotic network configuration can vary significantly in their generalization error. For the $\mathcal{T}_d^i$, the basin of attraction to the optimal solution described above is quite small and requires highly symmetric initial bias values. Otherwise the bias dynamics show the grouping around the true teacher bias value similar to the realizable case with the notable difference, that the bias values seem to diverge instead of converging to (sub-optimal) fixed values.

For non-degenerate biases, one also finds a multitude of stable network configuration depending on the initial conditions, which all feature quite similar generalization error. For the task $\mathcal{T}_n^i$ for example, a different set of initial conditions (changing only the norms $Q_{ii} = (1+n)/10$) leads to student unit 2 specializing primarily on teacher unit 3 instead of specializing almost equally on teacher units 2 and 3 and results in a slightly smaller generalization error. We find that the evolution of the dynamics to solutions with similar asymptotic generalization error are qualitatively similar, but one does not find a dominant basin of attraction to a particular solution as in the case of fixed biases. A more detailed investigation of these issues is, however, beyond the scope this thesis has set itself.

Finally we would like to point out, that in the case of student-teacher mismatch $K \neq M$, the difference between the normalized and unnormalized committee machine are substantial and the results are therefore quite different. For $K > M$, the unnormalized soft-committee machine is overrealizable and the excess nodes can be pruned away to achieve perfect generalization. This is obviously not possible for the normalized soft-committee machine due to the different normalization factor, and the task becomes unrealizable with a finite asymptotic generalization error. For $K < M$, the normalisation of the committee machine leads to generally lower asymptotic values of the order parameters with a resulting generalization error which is always lower than for the unnormalized case. This seems due to the normalization keeping the variance of the network output distribution of constant order (for uncorrelated teacher weight vectors) irrespective of the number of hidden units, whereas the order of the output variance is mismatched ($\sqrt{K}$ and $\sqrt{M}$) in the unnormalized model.

5.7 Summary and discussion

This research has been motivated by recent progress in the theoretical study of on-line learning in realistic two-layer neural network models — the soft-committee machine, trained with back-propagation (Saad and Solla 1995b). The studies so far have excluded biases to the hidden layers, a constraint which has been removed here in order to study the influence of the biases on the models learning behaviour. This extended model is in principle a universal approximator (West et al. 1997), although within the framework at issue the model can only be studied in a limit where the approximation proof does not hold as it may require the number of hidden units to scale with N. Nevertheless, the ensuing dynamics turn out to be very rich and more complex than the original model, although we had to restrict ourselves for computational reasons to small networks.

For non-degenerate teacher biases, one finds that the symmetry in the student hidden unit space can be broken almost immediately by the biases, provided the student biases were initialized asymmetrically, speeding up the learning process considerably in comparison to the fixed bias model where the training process can easily be dominated by the symmetric phase characterized by a lack of hidden unit specialization. These results suggest that student biases should in practice be initially spread evenly across the input domain if there is no *a priori* knowledge of the target function. For degenerate teacher biases, however, especially in combination with similar teacher lengths, such a scheme can be extremely counterproductive as asymmetric initial student biases severely prolong the training and can in many cases even trap the learning process permanently in attractive fixed points. Although attractive suboptimal fixed points were also found in the original soft-committee machine model (Biehl et al. 1996), these seem to have been restricted to over-realizable cases and the associated basins of attraction have been very small.

Unlike in the fixed bias case, the initial conditions, Q_{ij} and θ_i, which can manipulated in real scenarios, influence the training time considerably and can even cause complete training failure. To gain a qualitative understanding of the influence of the initial conditions, the basins of attraction to the optimal solution were therefore studied exhaustively for $K = M = 2$. One finds that attractive suboptimal fixed points exist for many training scenarios, including graded teachers and even non-degenerate teacher biases. The range of initial conditions attracted to these suboptimal network configurations diminishes with increasing asymmetry of the task, especially for non-degenerate teacher biases, where the attractive fixed point vanishes eventually. In the task with the smallest basin of attraction, isotropic teacher weight vectors and degenerate teacher biases, which was studied in great detail, one finds several unexpected

results. First, the basin of attraction is mainly dependent on the difference in the initial student biases, rather than their individual abscissas or the resulting mean. Second, the basin of attraction, with respect to the student biases θ_i, grows with increasing student norms, but the corresponding abscissa ($\hat{\theta} = \theta/\sqrt{Q}$) decreases. Third, the basin of attraction is enlarged by larger initial student-teacher overlaps and training should therefore be less prone to failure for smaller input dimension.

Additionally, the influence of the learning rates on the basin of attraction was studied for the same isotropic and degenerate task. For a common learning rate for biases and weights, the basin of attraction shrinks to a minimum in the region of fastest convergence, i.e., for the overall optimal learning rate. The basin of attraction increases especially for small learning rate but always remains finite. More effective in increasing the basin of attraction seems, however, the separation of the bias and weight learning rate. Whereas one must necessarily pay dearly for stability with a decrease in convergence speed when employing a small bias or weight learning rate, a large bias learning rate does not compromise training efficiency.

Although most of the results found for $K = 2$ also carry over qualitatively to larger networks, the size of the basin of attraction shrinks considerably with network size, which may partly be contributed to the substantial increase of the number of attractive suboptimal fixed points with different internal symmetries. In particular, we have found that the use of a large bias learning rate or the adiabatic elimination of the biases can actually decrease the basin of attraction for larger networks and degenerate biases.

Unlike preliminary results (West et al. 1997) which seemed to support the heuristic suggestion in an earlier work (Nguyen and Widrow 1990) to spread the abscissas across the input domain in order to speed up training; our more extensive work, clearly suggest that such an initialization scheme may in general not be advisable. Our results show that in terms of the initialization, the difference in the threshold values and not the individual abscissas are the more relevant variables. Furthermore, such a scheme will most likely fail to convergence to the optimal solution when some of the biases are degenerate, although one can only speculate how common these tasks are encountered in practice.

Other previous work (Kim and Ra 1991), which relates the basin of attraction of the weight initialization with the learning rate, seems also to be partially contradicted by our findings. Although the basin of attraction does grow with decreasing learning rate, as found in (Kim and Ra 1991), the functional relationship given for convergence in this work ($\eta_0 < Q_{ii} + \theta_i^2$) fails to predict a finite boundary for an infinitesimal learning rate. Furthermore, the treatment of the biases as just another weight parameter suggests

a growing basin of attraction with both increasing weights and biases, whereas we find that biases actually have the reverse effect. The work also neglects the strong interaction between the hidden units, e.g., the importance of the difference in initial thresholds or the shrinking of the basin of attraction for larger networks.

An initialization procedure which provides both stability and fast convergence speed for all tasks, seems therefore difficult to realize due to the inherently different requirements for tasks with degenerate and non-degenerate biases. The probably most successful approach is to opt for a combined approach of medium spread of the biases, large initial weights, a reasonable separation of weight and bias learning rate. This must be combined with a criteria which restarts network biases for hidden units trapped in an attractive suboptimal fixed point. Since for most attractive fixed points found, the student hidden units are not highly saturated, i.e., the absolute values of their mean output is reasonably less than 1, it is not sufficient to just select saturated units with large effective bias. This criteria must therefore account for the actual bias values in combination with correlations between the student hidden unit weight vectors. For persistently large correlation between a pair of weight vectors and very similar lengths, the biases could for example be reset to their mean value. If such a strategy works in all situations remains to be shown, which goes beyond the scope of this thesis. Possible difficulties are likely to be unrealizable scenarios, where persistent correlation are caused by a lack of student resources and a successful algorithm would have to be able to distinguish between the two. Its usefulness would then have to be further tested in finite size systems and real world problems. However, as already mentioned, in cases where the training set is known in advance, many algorithms are available that aim to infer good initial conditions from the training data [see e.g., (Lehtokangas et al. 1995) and references therein].

Unlike for the entire training process and general learning scenarios, where we had to restrict ourselves to small networks, the dynamics can be studied and optimized for all network sizes for the isotropic degenerate teacher task in the convergence phase, where hidden unit symmetry is already broken successfully and the student approaches the optimal solution. Since this type of task is not only the slowest in terms of overall training time, but also in the convergence phase itself, the results should give us a bound on the performance of other tasks.

One finds that optimal convergence is achieved for an infinite bias learning rate, suggesting that an $\mathcal{O}(1)$ rather than an $\mathcal{O}(1/N)$ bias learning rate is appropriate for finite systems once hidden unit symmetry is broken and that the input-hidden weights dominate the learning behaviour in this phase. The dependence of the optimal (weight) learning rate has been studied as a function of the number of hidden units K and the

teacher length T with special emphasis on the influence of non-zero (effective) bias $\hat{\varrho}$, which provides the most useful scaling of the bias in the convergence phase. We have restricted ourselves also to two learning rate parameterizations for the biases: $\eta_\theta = \eta_w$ and $\eta_\theta \to \infty$. One finds that both for small T or small $\hat{\varrho}$, there is either no or little difference between the two parameterizations. The advantage of an increased bias learning rate grows, however, for large enough T approximately proportionally to $\hat{\varrho}^2$.

The influence of the value of the effective teacher biases $\hat{\varrho}$ manifests itself for both parameterizations in the initially surprising effect that for most T values the learning performance actually improves for small non-zero bias. This can be explained by postulating that most information on the parameters of an individual hidden unit can be obtained in the region where the sigmoid is already nonlinear but not quite saturated. In this region one finds an exponentional increase in the optimal learning rate matching the suppression of the gradient. This increase, however, cannot be sustained for larger $\hat{\varrho}$ and leads to an eventual exponentional decay of the convergence speed in $\hat{\varrho}^2$ for any finite K. This exponentional decay is delayed to larger $\hat{\varrho}$ values for small teacher length T and large network size K, which may be attributed to the increasing smoothness of the error signals allowing for a larger learning rate. This fact is epitomized in the $T \to 0$ and $K \to \infty$ limits, where the convergence rate does increase unabatedly or decreases at most algebraically in $\hat{\varrho}$, respectively.

The choice of the learning rate is therefore important in both the symmetric phase, where it can help to avoid attractive fixed point as well as in the convergence phase, where the optimal value varies significantly in the relevant region of parameter space, making it difficult to choose good learning rates in practice. The problem of training is also exacerbated by the difficulty of student parameter initialization without *a priori* knowledge about the learning task present, which can change the basin of attraction to the optimal solution considerably.

Many future research directions will be interesting to pursue. Since the learning dynamics have shown to change significantly with the introduction of adjustable biases for realizable scenarios, it appears to be of obvious interest to investigate the influence of unrealizability more systematic than could be achieved within the scope of this chapter. It may be further interesting to investigate if noisy rules will have a significant impact on the qualitative behaviour of the system.

Of most interest and relevance, however, seems to be the understanding of the shortcomings of standard gradient descent in on-line learning of multilayer networks and the subsequent development of more sophisticated on-line learning algorithms. This problem can be addressed heuristically, by attempting to find good criteria for monitoring

the progress made in training by monitoring the student parameters, which may improve the setting of learning rates or aid in the identification of hidden units trapped in suboptimal fixed points. Theoretical studies may help to guide such development by analysing current algorithms and by analysing the effect of modification of algorithms motivated either *ad hoc* or by principle. This is the aim of the following chapter, where we study the impact of a simple alteration to standard gradient descent motivated by the dominance of the symmetric phase in learning scenarios with similar teacher biases or for symmetric initialization of the student biases. In order to be able to study the dynamics analytically for both the symmetric and the convergence phase in more detail, we restrict ourselves to the soft-committee machine with zero-fixed biases.

Appendix 5

5.A Dynamical equations

The generalization error is calculated by averaging the quadratic loss function (5.3) explicitly over the activations $\{x, y\}$ (and implicitly over all inputs) which are multivariate Gaussian distributed with zero mean and covariance matrix C given by

$$C = \sigma^2 \begin{bmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{T} \end{bmatrix}. \tag{5.A.1}$$

In the following all averages are taken with respect to this distribution and making use of the convention that indices i, j, k, l and n, m label student and teacher nodes, respectively.

The generalization error takes the form

$$\epsilon_g = \frac{\gamma^2}{2K} \left\{ \frac{K}{M} \sum_{n,m=1}^{M} J_2(n,m) - 2\sqrt{\frac{K}{M}} \sum_{i,n=1}^{K,M} J_2(i,n) + \sum_{i,j=1}^{K} J_2(i,j) \right\}, \tag{5.A.2}$$

with the integral $J_2(1,2) = \langle g(u_1)g(u_2) \rangle$, where u_i represent members of $\{x, y\}$ and the sigmoidal transfer function g is here taken to be the error function $g_\nu(u) = \mathrm{erf}(\nu u/\sqrt{2})$. We denote with I_d, J_d averages over d variables with one and two g terms, respectively. The integrals can be calculated by introducing an integral representation for g (A.2), which allows the integrals to be evaluated by Gaussian integration (A.4). Unlike in the case of fixed zero-biases, however, only for integrals involving a single g terms can the remaining single integral (caused by the integral representations introduced earlier)

be calculated analytically. For the integrals involving g^2 terms, the two remaining integrals have shifted arguments for which no analytical solution is known. However, these integrals can be simplified considerably to make a numerical integration feasible. There are several possible representations, e.g., the Kendall series expansion, but we have chosen one which consists of a single Gaussian integral of two error functions. We have found that this form has the advantage that the summation over units and the integration can be interchanged, greatly improving numerical accuracy for fixed computational cost.

In this form the integral $J_2(\cdot)$ is given by

$$J_2(1,2) = \int Dt\, g_\nu\Big(\sqrt{\sigma^2 C_{11}}\, t - \vartheta_1\Big)\, g_\nu\left(\frac{\sigma^2 C_{12} t - \vartheta_2}{\sqrt{\sigma^2 C_{11}\psi_2 - \nu^2\sigma^2 C_{12}^2}}\right), \tag{5.A.3}$$

where

$$\psi_i = 1 + \nu^2\sigma^2 C_{ii}, \quad \text{and} \quad Dt = \frac{dt}{\sqrt{2\pi}}\exp\left(-\frac{t^2}{2}\right)$$

is the Gaussian measure, with any integral without explicit limits is from $-\infty$ to $+\infty$. The dependence of the integral on the sigmoidal gain ν and the input variance σ^2 can be absorbed by redefining

$$\tilde{\vartheta}_i = \nu\vartheta_i, \quad \text{and} \quad \tilde{C}_{ij} = \nu^2\sigma^2 C_{ij},$$

a rescaling which also holds for the other integrals below. To evaluate an integral explicitly, the full covariance matrix $\mathcal{C}$ is projected into the relevant subspace. For example, the relevant elements for $J_2(i,n)$ are $C_{11} = Q_{ii}$, $C_{12} = R_{in}$, and $C_{22} = T_{nn}$. It is a property of multivariate Gaussian distributions (Saad and Solla 1995b) that integrals of reduced dimensionality such as $J_2(1,1)$ are generated from the general form $J_2(1,2)$ by the appropriate constraints (in this case $C_{11} = C_{12} = C_{22}$).

The differential equations for $\mathbf{Q}$, $\mathbf{R}$, and $\boldsymbol{\theta}$ are calculated similarly and take the form

$$\frac{dQ_{ij}}{d\alpha} = \frac{\eta_w\gamma^2}{K}\left\{\sqrt{\frac{K}{M}}\sum_{m=1}^{M} I_3(i,j,m) + I_3(j,i,m) - \sum_{k=1}^{K} I_3(i,j,k) + I_3(j,i,k)\right\} \tag{5.A.4a}$$

$$+\left(\frac{\eta\gamma^2}{K}\right)^2\left\{\frac{K}{M}\sum_{n,m=1}^{M} J_4(i,j,n,m) - 2\sqrt{\frac{K}{M}}\sum_{k,n=1}^{K,M} J_4(i,j,k,n) + \sum_{k,l=1}^{K} J_4(i,j,k,l)\right\},$$

$$\frac{dR_{in}}{d\alpha} = \frac{\eta_w\gamma^2}{K}\left\{\sqrt{\frac{K}{M}}\sum_{m=1}^{M} I_3(i,n,m) - \sum_{k=1}^{K} I_3(i,n,k)\right\}, \tag{5.A.4b}$$

$$\frac{d\theta_i}{d\alpha} = -\frac{\eta_\theta \gamma^2}{K}\left\{\sqrt{\frac{K}{M}}\sum_{m=1}^{M} I_2(i,n) - \sum_{k=1}^{K} I_2(i,k)\right\}, \tag{5.A.4c}$$

where the two integrals $I_2(1,2) = \langle g'(u_1)g(u_2)\rangle$ and $I_3(1,2,3) = \langle g'(u_1)u_2\, g(u_3)\rangle$ can be evaluated analytically, whereas $J_4(1,2,3,4) = \langle g'(u_1)g'(u_2)g(u_3)g(u_4)\rangle$ can be simplified to a form similar to $J_2(\cdot)$ and one finds

$$I_2(1,2) = \nu\sqrt{\frac{2}{\pi}}\frac{1}{\sqrt{\psi_1}}\exp\left(-\frac{1}{2}\frac{\tilde{\vartheta}_1^2}{\psi_1}\right) g_1(\Theta_{12}) \tag{5.A.5a}$$

$$I_3(1,2,3) = \sqrt{\frac{2}{\pi}}\frac{1}{\sqrt{\psi_1}}\exp\left(-\frac{1}{2}\frac{\tilde{\vartheta}_1^2}{\psi_1}\right) \tag{5.A.5b}$$
$$\times\left[\frac{\tilde{C}_{13}\tilde{\vartheta}_1}{\psi_1} g_1(\Theta_{12}) + \sqrt{\frac{2}{\pi}}\frac{\Psi_{12}\Gamma_{13}}{\sqrt{\Psi_{13}}\sqrt{\psi_1}}\exp\left(-\frac{1}{2}\Theta_{12}^2\right)\right],$$

$$J_4(1,2,3,4) = \nu^2\exp\left(-\frac{1}{2}\frac{\psi_2\tilde{\vartheta}_1^2 - 2\tilde{C}_{12}\tilde{\vartheta}_1\tilde{\vartheta}_2 + \psi_1\tilde{\vartheta}_2^2}{\Psi_{12}}\right) \tag{5.A.5c}$$
$$\times\frac{2}{\pi}\frac{1}{\sqrt{\Psi_{12}}}\int Dt\, g_1\left(\sqrt{\tilde{C}'_{33}}\, t - \tilde{\vartheta}'_3\right) g_1\left(\frac{\tilde{C}'_{34}t - \tilde{\vartheta}'_4}{\sqrt{\tilde{C}'_{33}\psi'_4 - \tilde{C}'^2_{34}}}\right),$$

where we conveniently define

$$\Psi_{ij} = \psi_i\psi_j - \tilde{C}_{ij}^2, \qquad \Theta_{ij} = \frac{\tilde{C}_{ij}\tilde{\vartheta}_i - \psi_i\tilde{\vartheta}_j}{\sqrt{\psi_i\psi_{ij}}},$$

$$\Gamma_{1i} = \frac{\psi_1\tilde{C}_{2i} - \tilde{C}_{12}\tilde{C}_{1i}}{\Psi_{12}}, \qquad \Gamma_{2i} = \frac{\psi_2\tilde{C}_{1i} - \tilde{C}_{12}\tilde{C}_{2i}}{\Psi_{12}},$$

and the primed variables

$$\tilde{C}'_{ij} = \tilde{C}_{ij} - (\tilde{C}_{1i}\Gamma_{2j} + \tilde{C}_{2i}\Gamma_{1j}), \qquad \tilde{\vartheta}'_i = \tilde{\vartheta}_i - (\tilde{\vartheta}_1\Gamma_{2i} + \tilde{\vartheta}_2\Gamma_{1i}),$$

with the obvious extensions, e.g., $\psi'_i = 1 + \tilde{C}'_{ii}$. Again, one infers the elements of the reduced covariance matrix using the unit labelling convention and the appropriate dimensionality reduction.

As mentioned above the gain ν rescales all order parameters and the biases explicitly and furthermore leads to an implicit rescaling of both learning rates by ν^2 in the differential equations (5.A.4). The learning rates are further rescaled by the linear output gain by γ^2. The total rescaling of any bias and the bias learning rate η_0 therefore

is

$$\tilde{\vartheta} = \nu\vartheta, \qquad \text{and} \qquad \tilde{\eta}_\theta = \frac{\nu^2\gamma^2}{(K)}\eta_\theta. \tag{5.A.6a}$$

For the weight order parameters and their learning rate η_w the result is

$$\tilde{C} = \nu^2\sigma^2 C \qquad \text{and} \qquad \tilde{\eta}_w = \frac{\nu^2\gamma^2\sigma^2}{(K)}\eta_w. \tag{5.A.6b}$$

In the remainder of the chapter we will therefore set $\nu = \gamma = \sigma = 1$ w.l.o.g..

5.B The analytical convergence dynamics

For a realizable isotropic teacher scenario characterized by $K = M$, $T_{nm} = T\delta_{nm}$, and degenerate biases $\varrho_n = \varrho$, the number of free parameters can be reduced with the ansatz (5.14), to just five variables R, S, Q, C, and θ, which gives an accurate description for the dynamics when the student biases were not initialized too unsymmetrically.

In the convergence phase one can expand the differential equations (5.A.4) in a Taylor series to first order around the zero generalization error fixed point, $Q^* = R^* = T$, $C^* = S^* = 0$, and $\theta^* = \varrho$,

$$\frac{dp_i}{d\alpha} = \sum_{j=1}^{4} m_{ij}p_j,$$

where $p_i = P_i - P_i^*$ and P_i are generic order parameters [we use the ordering $P_1 = R$, $P_2 = Q$, $P_3 = S$, $P_4 = C$, and $P_5 = \theta$ following the convention of earlier work (Saad and Solla 1995b)], and the eigenvalues and eigenvectors of the Jacobian matrix $\mathbf{M}$ of first derivatives determine the solution of the linearized differential equation.

The elements of the Jacobian matrix are explicitly given by

$$m_{11} = -\frac{2}{\pi}\frac{\eta_w}{K}\frac{1}{(1+2T)^{\frac{3}{2}}}\exp\left(-\frac{\varrho^2}{1+2T}\right)\left[(1+3T) - \frac{2T\varrho^2}{1+2T}\right], \tag{5.B.1a}$$

$$m_{12} = \frac{1}{\pi}\frac{\eta_w}{K}\left\{3\frac{[(1+2T)-2\varrho^2]T}{(1+2T)^{\frac{5}{2}}}\,\mathrm{e}^{-\frac{\varrho^2}{1+2T}} - (K-1)\frac{T\varrho^2}{(1+T)^3}\,\mathrm{e}^{-\frac{\varrho^2}{1+T}}\right\}, \tag{5.B.1b}$$

$$m_{13} = \frac{2}{\pi}\frac{\eta_w}{K}\frac{K-1}{(1+T)^2}\exp\left(-\frac{\varrho^2}{1+T}\right)\left[(1+2T) - \frac{T\varrho^2}{1+T}\right], \tag{5.B.1c}$$

$$m_{14} = -\frac{2}{\pi}\frac{\eta_w}{K}(K-1)\exp\left(-\frac{\varrho^2}{1+T}\right)\frac{[(1+T)-\varrho^2]T}{(1+T)^3}, \tag{5.B.1d}$$

$$m_{15} = \frac{2}{\pi}\frac{\eta_w}{K}\varrho T\left[2\frac{1}{(1+2T)^{\frac{3}{2}}}\,\mathrm{e}^{-\frac{\varrho^2}{1+2T}} + \frac{K-1}{(1+T)^2}\,\mathrm{e}^{-\frac{\varrho^2}{1+T}}\right], \tag{5.B.1e}$$

$$m_{21} = \frac{4}{\pi}\frac{\eta_w}{K}\left\{\frac{(1+T)(1+2T)+2T\varrho^2}{(1+2T)^{\frac{5}{2}}}\exp\left(-\frac{\varrho^2}{1+2T}\right)\right.$$
$$\left. -\frac{2}{\pi}\frac{\eta_w}{K}\left[\frac{1}{\sqrt{1+4T}}\,\mathrm{e}^{-\frac{2\varrho^2}{1+4T}}+\frac{K-1}{1+2T}\,\mathrm{e}^{-\frac{2\varrho^2}{1+2T}}\right]\right\}, \tag{5.B.1f}$$

$$m_{23} = -\frac{4}{\pi}\frac{\eta_w}{K}(K-1)\exp\left(-\frac{\varrho^2}{1+T}\right)\left[\frac{(1+T)+T\varrho^2}{(1+T)^3}-m'_{23}\right], \tag{5.B.1g}$$

$$m'_{23} = \frac{2}{\pi}\frac{\eta_w}{K}\left[\frac{2\,\mathrm{e}^{-\frac{\varrho^2}{(1+T)(1+3T)}}}{\sqrt{(1+T)(1+3T)}}+\frac{(K-2)}{(1+T)\sqrt{1+2T}}\,\mathrm{e}^{-\frac{\varrho^2}{1+2T}}\right], \tag{5.B.1h}$$

$$m_{31} = \frac{2}{\pi}\frac{\eta_w}{K}\frac{1}{1+T}\exp\left(-\frac{\varrho^2}{1+T}\right), \tag{5.B.1i}$$

$$m_{32} = -\frac{1}{\pi}\frac{\eta_w}{K}\frac{(1+T)-\varrho^2}{(1+T)^3}T\exp\left(-\frac{\varrho^2}{1+T}\right), \tag{5.B.1j}$$

$$m_{33} = -\frac{2}{\pi}\frac{\eta_w}{K}\left[\frac{(K-2)(1+T)^2-T\varrho^2}{(1+T)^3}\,\mathrm{e}^{-\frac{\varrho^2}{1+T}}+\frac{1}{\sqrt{1+2T}}\,\mathrm{e}^{-\frac{\varrho^2}{1+2T}}\right], \tag{5.B.1k}$$

$$m_{34} = -\frac{2}{\pi}\frac{\eta_w}{K}\frac{T\varrho^2}{(1+T)^3}\exp\left(-\frac{\varrho^2}{1+T}\right), \tag{5.B.1l}$$

$$m_{35} = -\frac{2}{\pi}\frac{\eta_w}{K}\frac{T\varrho}{(1+T)^2}\exp\left(-\frac{\varrho^2}{1+T}\right), \tag{5.B.1m}$$

$$m_{41} = -\frac{4}{\pi}\frac{\eta_w}{K}\exp\left(-\frac{\varrho^2}{1+T}\right)\left[\frac{1}{1+T}-m'_{23}\right], \tag{5.B.1n}$$

$$m_{43} = \frac{4}{\pi}\frac{\eta_w}{K}\left\{\frac{(K-2)(1+T)^2+T\varrho^2}{(1+T)^3}\,\mathrm{e}^{-\frac{\varrho^2}{1+T}}+\frac{1}{\sqrt{1+2T}}\,\mathrm{e}^{-\frac{\varrho^2}{1+2T}}\right.$$
$$\left. -\frac{2}{\pi}\frac{\eta_w}{K}\left[\frac{2}{1+2T}\,\mathrm{e}^{-\frac{2\varrho^2}{1+2T}}+(K-2)\,\mathrm{e}^{-\frac{\varrho^2}{1+T}}\left(\frac{4}{\sqrt{1+2T}}\,\mathrm{e}^{-\frac{\varrho^2}{1+2T}}+\frac{K-3}{1+T}\,\mathrm{e}^{-\frac{\varrho^2}{1+T}}\right)\right]\right\}, \tag{5.B.1o}$$

$$m_{51} = -\frac{2}{\pi}\frac{\eta_\theta}{K}\frac{\varrho}{(1+2T)^{\frac{3}{2}}}\exp\left(-\frac{\varrho^2}{1+2T}\right), \tag{5.B.1p}$$

$$m_{52} = -\frac{1}{\pi}\frac{\eta_\theta}{K}\varrho\left[\frac{1}{(1+2T)^{\frac{3}{2}}}\,\mathrm{e}^{-\frac{\varrho^2}{1+2T}}+\frac{K-1}{(1+T)^2}\,\mathrm{e}^{-\frac{\varrho^2}{1+T}}\right], \tag{5.B.1q}$$

$$m_{53} = \frac{2}{\pi}\frac{\eta_\theta}{K}(K-1)\frac{\varrho}{(1+T)^2}\exp\left(-\frac{\varrho^2}{1+T}\right), \tag{5.B.1r}$$

$$m_{55} = -\frac{2}{\pi}\frac{\eta_\theta}{K}\left[\frac{1}{\sqrt{1+2T}}\exp\left(-\frac{\varrho^2}{1+2T}\right)+\frac{K-1}{1+T}\exp\left(-\frac{\varrho^2}{1+T}\right)\right]. \tag{5.B.1s}$$

The remaining elements can be deduced by the matrix relations

$$\begin{aligned} m_{11} - \frac{1}{2}m_{21} &= m_{22} - 2m_{12}, \qquad m_{33} - \frac{1}{2}m_{43} = m_{44} - 2m_{34}, \\ m_{13} - \frac{1}{2}m_{23} &= m_{24} - 2m_{14}, \qquad m_{31} - \frac{1}{2}m_{41} = m_{42} - 2m_{32}, \\ m_{25} &= 2m_{15}, \qquad m_{45} = 2m_{45}, \qquad m_{54} = -m_{53}. \end{aligned} \tag{5.B.2}$$

The characteristic polynomial of such a Jacobian matrix, whose zeros define the eigenvalues, separates in a quadratic and a cubic equation. The two eigenvalues given by the quadratic equation correspond to those of the 4×4-matrix with fixed biases and are given by

$$\lambda_{1,2} = \frac{1}{2}\left[A_1 + B_1 \pm \sqrt{(A_1 - B_1)^2 + 4C_1D_1}\right], \tag{5.B.3a}$$

with

$$\begin{aligned} A_1 &= m_{11} - \frac{1}{2}m_{21}, \qquad B_1 = m_{44} - 2m_{34}, \\ C_1 &= m_{31} - \frac{1}{2}m_{41}, \qquad D_1 = m_{24} - 2m_{14}. \end{aligned} \tag{5.B.3b}$$

These eigenvalues are nonlinear in the learning rate η_w. The remaining eigenvalues are given by the solutions to the cubic equation

$$0 = \lambda^3 + a_2\lambda^2 + a_1\lambda + a_0, \tag{5.B.4a}$$

with coefficients

$$\begin{aligned} a_2 &= -(m_{55} + A_2 + B_2), \\ a_1 &= m_{55}(A_2 + B_2) + (A_2B_2 - C_2D_2) - E_2m_{15} - m_{54}m_{35}, \\ a_0 &= -m_{55}(A_2B_2 - C_2D_2) + m_{54}(m_{35}A_2 - m_{15}D_2) + E_2(m_{15}B_2 - m_{35}C_2), \end{aligned} \tag{5.B.4b}$$

where

$$\begin{aligned} A_2 &= m_{11} + 2m_{12}, \qquad B_2 = m_{44} + \frac{1}{2}m_{43}, \\ C_2 &= m_{31} + 2m_{32}, \qquad D_2 = m_{24} + \frac{1}{2}m_{23}, \qquad E_2 = m_{51} + 2m_{52}, \end{aligned} \tag{5.B.4c}$$

These eigenvalues are negative for all values of η_w and η_θ. For $\eta_w = \eta_\theta = \eta_0$, these eigenvalues are also linear in η_0.

This can be confirmed by finding the zeros of the determinant in the two learning rates η_w and η_θ, which correspond to an eigenvalue becoming zero and therefore define critical (maximal) learning rates. For the equations for the determinant roots

$$A_1 B_1 - C_1 D_1 = 0, \tag{5.B.5a}$$

$$a_2 = 0, \tag{5.B.5b}$$

we obtain only one non-trivial, i.e., non-zero, solution for Eq. (5.B.5a) and hence the weight learning rate η_w, coinciding with $\lambda_1 = 0$, and in particular no non-trivial solution for Eq. (5.B.5b) and hence the bias learning rate η_θ. This and numerical solutions suggest that the optimal bias learning rate is located at infinity.

This can be explicitly shown for the special case $\varrho = 0$, where the eigenvalue spectrum separates further. A closer inspection of the matrix elements reveals that all m_{5i} and m_{i5} for $i \neq 5$ become zero and the eigenvalues take the form

$$\lambda_{3,4} = \frac{1}{2}\left[A_2 + B_2 \pm \sqrt{(A_2 - B_2)^2 + 4C_2 D_2}\right], \tag{5.B.6a}$$

$$\lambda_5 = m_{55}, \tag{5.B.6b}$$

recovering the convergence dynamics of the weight order parameters in the isotropic case with fixed biases as studied in Chapter 6, but for an extra eigenvalue describing the decay of the student biases to their optimal value. Since only this eigenvalue depends (linearly) on η_θ, the optimal bias learning rate is at infinity.

To make progress in the general case of non-zero teacher bias, we restrict our study to two possible parameterizations $\eta_0 = \eta_\theta = \eta_w$ and a finite weight learning rate η_w with $\eta_\theta \to \infty$. In the following we use the convention that the (weight) learning rate will be denoted by η for the generic case or when a result is valid for both parameterizations.

For large η_θ, we expand the characteristic polynomial (5.B.4a) asymptotically with the two ansätze $\lambda = \mathcal{O}(\eta_\theta)$ and $\lambda = \mathcal{O}(1)$. One finds that the characteristic polynomial separates as expected into

$$\lambda_{3,4} = \frac{1}{2}(A_2 + B_2) - \frac{E_2 m_{15} + m_{54} m_{35}}{2 m_{55}} \pm \frac{1}{2}\left[(A_2 - B_2)^2 + \left(\frac{E_2 m_{15} + m_{54} m_{35}}{m_{55}}\right)^2 + 4C_2 D_2 - 2\frac{(A_2 - B_2)(E_2 m_{15} - m_{54} m_{35})}{m_{55}} - 4\frac{E_2 m_{35} C_2 + m_{54} m_{15} D_2}{m_{55}}\right]^{\frac{1}{2}}, \tag{5.B.7a}$$

$$\lambda_5 = m_{55}, \tag{5.B.7b}$$

which is similar to the zero bias case, but with corrections to the eigenvalues $\lambda_{3,4}$ due

to the finite biases. However, these eigenvalues become independent of the value of η_θ.

In order to study the optimal value of the learning rate η, which gives the fastest decay to zero generalization error, one has to assess which mode, i.e., eigenvalue and associated eigenvector, contributes to its decay. We therefore expand the generalization error (5.A.2) to second order in $\{q, r, s, c, \vartheta\}$

$$\epsilon_g = \frac{e^{-\frac{\varrho^2}{1+2T}}}{\pi\sqrt{1+2T}}\left[(2r-q) - \frac{1}{4}(2r-q)^2\frac{T(1+2T)+\varrho^2}{(1+2T)^2} + q(r-q)\frac{(1+2T)-2\varrho^2}{(1+2T)^2}\right.$$
$$\left. + \vartheta\varrho\frac{2r-3q}{1+2T} + \vartheta^2\right] - \frac{K-1}{\pi(1+T)}e^{-\frac{\varrho^2}{1+T}}\left[(2s-c) + q(s-c)\frac{(1+T)-\varrho^2}{(1+T)^2}\right.$$
$$\left. + \frac{1}{4}(4s^2-2c^2-q^2)\frac{\varrho^2}{(1+T)^2} + \vartheta(2s-2c+q)\frac{\varrho}{1+T} - \vartheta^2\right]. \quad (5.B.8)$$

Unfortunately, we were unable to find analytical solutions to the eigenvectors. Numerical solutions, however, show that the eigenvectors associated with the eigenvalues $\lambda_{3,4,5}$ are orthogonal to the first-order terms in the generalization error and thus cannot contribute to their decay. These modes are therefore only relevant for second-order terms in the generalization error with a decay rate of $2\lambda_{3,4,5}$. As discussed in Section 5.5, the fastest convergence is given by Eq. (5.15). This is usually achieved either for η_r^{opt}, where $2\lambda_3 = \lambda_1$, or for η_m^{opt}, which is defined by the minimum of λ_1.

It is in general infeasible to optimize the eigenvalues with respect to the learning parameter η (η_w or η_0) analytically for arbitrary K, T and ϱ. However, one can make some progress in certain limits of K, T, and ϱ which we will investigate below.

5.B.1 Large-K limit

The dominant terms for large number of hidden units for all relevant quantities can be extracted by an asymptotic series expansion under the self-consistent ansatz $\eta_w = \mathcal{O}(1)$. For the two relevant eigenvalues one makes the ansatz $\lambda_i = \mathcal{O}(K^{-1})$ and finds to leading order

$$\lambda_1 = -\frac{4}{\pi}\frac{\eta_w\mathcal{E}_1}{K}\frac{\left[(1+T)-\sqrt{1+2T}\mathcal{E}_2\right]\left(\pi\sqrt{1+2T}-\eta_w\mathcal{E}_1\right)}{(1+2T)\left[\pi(1+T)-\eta_w\mathcal{E}_1\mathcal{E}_2\right]}, \quad (5.B.9a)$$

$$\lambda_3 = -\frac{2}{\pi}\frac{1}{K}\frac{\eta_w\eta_\theta\mathcal{E}_1}{\eta_\theta(1+T)^2+\eta_w T\varrho^2}\left[\frac{(1+T)^2}{(1+2T)^{\frac{3}{2}}} + \frac{T\varrho^2}{(1+2T)^{\frac{5}{2}}} - \frac{\mathcal{E}_2}{1+T}\right], \quad (5.B.9b)$$

with the auxiliary variables

$$\mathcal{E}_1 = \exp\left(-\frac{\varrho^2}{1+2T}\right), \quad \text{and} \quad \mathcal{E}_2 = \exp\left(-\frac{T\varrho^2}{(1+T)(1+2T)}\right). \tag{5.B.9c}$$

These define two critical learning rates

$$\eta_w^{\max} = \pi\frac{\sqrt{1+2T}}{\mathcal{E}_1}, \tag{5.B.10a}$$

$$\eta_w^{\text{crit}} = \pi\frac{1+T}{\mathcal{E}_1\mathcal{E}_2} > \eta_w^{\max}, \tag{5.B.10b}$$

where λ_1 is identical to zero ($\eta_w^{\max}$) [corresponding to the maximal learning rate that can also be obtained by solving Eq. (5.B.5a)] and diverges (η_w^{crit}), respectively. Inspecting Eqs. (5.B.9) and (5.B.10) suggests that the natural rescaling for the learning rates for non-zero teacher bias in this limit is

$$\hat{\eta}_w = \eta_w\mathcal{E}_1 \quad \text{and} \quad \hat{\eta}_\theta = \eta_\theta\mathcal{E}_1. \tag{5.B.11}$$

We further mention in passing, that Eq. (5.B.9a) is only a valid expansion of λ_1 for $\eta_w < \eta_w^{\text{crit}}$, beyond which the ansatz $\lambda_1 = \mathcal{O}(K^{-1})$ breaks down, a fact that becomes important when optimizing the dynamics with respect to the learning rate.

For both of parameterizations ($\eta_0 = \eta_w = \eta_\theta$ and η_w with $\eta_\theta = \eta_\theta^{\text{opt}} \to \infty$) this optimization is performed by calculating both $\eta_{\text{r}}^{\text{opt}}$ and $\eta_{\text{m}}^{\text{opt}}$, i.e., solving $2\lambda_3 = \lambda_1$ and $d\lambda_1/d\eta = 0$, respectively. Since λ_1 is only a function of η_w, $\eta_{\text{m}}^{\text{opt}}$ is identical for both parameterizations, whereas $\eta_{\text{r}}^{\text{opt}}$ is in general different. The candidates for the optimal learning rate take the form

$$\begin{aligned}\eta_{0,\text{r}}^{\text{opt}} = &\left\{2(1+T)^2\left[(1+T)(1+2T)+2T\varrho^2\right] - (1+2T)^{\frac{5}{2}}(2+T+\varrho^2)\mathcal{E}_2\right\} \\ &\times \pi(1+T)T\mathcal{E}_1^{-1}\left\{(1+2T)^{\frac{3}{2}}(1+T)^2\left[(1+T)^2+T\varrho^2\right] + (1+2T)^{\frac{5}{2}}\mathcal{E}_2^2\right. \\ &\left. - 2(1+T)\left[2(1+2T)(1+T)^3 + T(1+2T+2T^2)\varrho^2\right]\mathcal{E}_2\right\}^{-1},\end{aligned} \tag{5.B.12a}$$

$$\begin{aligned}\eta_{w,\text{r}}^{\text{opt}} = &\pi(1+T)T\mathcal{E}_1^{-1}\left\{(1+T)\left[2(1+T)^2(1+2T)-\varrho^2\right] - (1+2T)^{\frac{5}{2}}(2+T)\mathcal{E}_2\right\} \\ &\times\left\{(1+2T)^{\frac{3}{2}}(1+T)^4 - (1+T)\left[2(1+2T)(1+T)^3+T\varrho^2\right]\mathcal{E}_2 + (1+2T)^{\frac{5}{2}}\mathcal{E}_2^2\right\}^{-1},\end{aligned} \tag{5.B.12b}$$

$$\eta_{\text{m}}^{\text{opt}} = \pi\frac{1}{\mathcal{E}_1\mathcal{E}_2}\left\{(1+T) - \sqrt{1+T}\left[(1+T) - \sqrt{1+2T}\mathcal{E}_2\right]^{\frac{1}{2}}\right\}. \tag{5.B.12c}$$

To decide on the correct optimal learning rate $\eta^{\rm opt}$, one has to evaluate whether $\eta^{\rm opt}_{\rm r,m} < \eta_{\rm crit}$ since the solution is otherwise spurious due to the breakdown of the ansatz for λ_1 above $\eta_{\rm crit}$. For the remaining valid candidates the optimal convergence rate is calculated. In general, one finds for given T and ϱ that $\eta^{\rm opt} = \eta^{\rm opt}_{\rm r}$ for $T > T^{\rm crit}(\varrho)$ and $\eta^{\rm opt} = \eta^{\rm opt}_{\rm m}$ for $T < T^{\rm crit}(\varrho)$, where $T^{\rm crit}(\varrho)$ is defined by $\eta^{\rm opt}_{\rm r} = \eta^{\rm opt}_{\rm m}$.

To make further progress in the $K \to \infty$ limit, one can look at several limits for T and ϱ. For the limits $T \to \infty$ and $T \to 0$, one has to consider scaling ansätze for the biases with T which ensure that the biases remain meaningful. As discussed in Section 5.2 and subsequently Section 5.5.3, one can adopt two possible interpretations of the influence of the biases which are identical to leading orders for $T \to \infty$ but qualitatively different for $T \to 0$. The *effective bias* ($\varrho = \hat{\varrho}\sqrt{1+T}$) keeps the mean hidden unit output constant for all T. The *abscissa* ($\varrho = \check{\varrho}\sqrt{T}$) keeps the distance of the decision hyperplane (or root) constant.

There are some further subtleties when studying various limits. The results for first taking the $K \to \infty$ limit and than the large-T limit turn out to be equivalent, to leading order in K and T, to results where both T and K go to their limits simultaneously, i.e., taking the limit $K \to \infty$ with $T = T_\infty K$, where T_∞ controls the significance between T and K. However, there is a significant difference to the case where the $T \to \infty$ limit is taken first, which will also be studied below. For small T on the other hand, the limits $K \to \infty$ and $T \to 0$ are interchangeable to third order. Below, we therefore only use those expansions which give us the more general solutions.

5.B.2 Small-T limit and $\hat{\varrho}$

In this limit, the slowest mode is associated with λ_1 and the optimal learning rate is determined by $\eta^{\rm opt}_{\rm m}$ which is identical for both learning rate parameterizations and the leading terms of the interesting quantities are

$$\eta_{\rm max} = \pi\, {\rm e}^{\hat{\varrho}^2} \left[1 + \left(1 - \frac{K+4}{K}\hat{\varrho}^2\right) T\right], \tag{5.B.13a}$$

$$\eta^{\rm opt} = \eta_{\rm max} - \pi\, {\rm e}^{\hat{\varrho}^2} \left[\sqrt{\frac{(K-1)\hat{\varrho}^2}{K}}\sqrt{T} - \frac{K-2}{K}\hat{\varrho}^2 T\right], \tag{5.B.13b}$$

$$\lambda^{\rm opt} = -4\frac{T}{K}\left\{\hat{\varrho}^2 - 2\sqrt{\frac{K-1}{K}}\hat{\varrho}^2\sqrt{T} + \frac{1}{2}\left[1 - 4\hat{\varrho}^2 + 5\frac{K-4}{K}\hat{\varrho}^4\right] T\right\}. \tag{5.B.13c}$$

The result for the model without biases can be recovered to leading order by simply setting $\hat{\varrho} = 0$. This shows that learning speed is improved by a factor of T for non-zero (finite) bias since the two leading terms of $\lambda^{\rm opt}$ vanish for $\hat{\varrho} = 0$. In this limit, the

effective bias $\hat{\varrho}$ dominates the dynamics. It is obvious, that this expansions suffers from two drawbacks. First, the limit of zero bias cannot be taken adequately for higher orders (this is especially obvious for higher-order terms in $\eta^{\rm opt}$, which have not been included here for brevity, where $\hat{\varrho}$ appear in the denominator). Second, the expansion predicts a unabated increase of the optimal convergence rate $\lambda^{\rm opt}$ with $\hat{\varrho}$, which is not the case for any finite T, where $\lambda^{\rm opt}$ levels off and eventually decays exponentionally. This is due to the implicit assumption in the $T \to 0$ expansion that $\hat{\varrho}^2 \ll -\log T$, i.e., the small T terms always dominate the solution over exponentional terms in $\hat{\varrho}$. Below, we will address the first of the inadequacies, by analysing the $T \to 0$ limit, with the scaling $\check{\varrho} = \varrho/\sqrt{T}$, i.e., ϱ vanishes with T.

5.B.3 Small-T limit and $\check{\varrho}$

As in the small-T limit with ϱ finite, the slowest mode is associated with λ_1 and both parameterizations are identical. In particular, one finds

$$\eta_{\rm max} = \pi \left[1 + (1+\check{\varrho}^2)T + \frac{\check{\varrho}^4}{2} T^2 - \frac{K+4}{2K}(1+2\check{\varrho}^2)T^2 \right], \tag{5.B.14a}$$

$$\eta^{\rm opt} = \eta_{\rm max} - \pi \sqrt{\frac{K-1}{2K}} \sqrt{1+2\check{\varrho}^2}\, T, \tag{5.B.14b}$$

$$\lambda^{\rm opt} = -2\frac{T^2}{K} \left\{ (1+2\check{\varrho}^2) - 2 \left[(1+3\check{\varrho}^2) + \sqrt{\frac{K-1}{2K}} (1+2\check{\varrho}^2)^{\frac{3}{2}} \right] T \right\}. \tag{5.B.14c}$$

In this case, the results for the model without biases are recovered for all orders for $\check{\varrho} = 0$. One can still see, that the learning is improved for non-zero biases, but for this scaling only by a factor of $1 + 2\check{\varrho}^2$ and not by $\mathcal{O}(T)$. This expansion holds only for $\check{\varrho}^2 \ll T$ due to the algebraic expansion of all exponentional terms.

5.B.4 Large-T and -K limit ($T = T_\infty K$):

For large T, the two scaling ansätze for ϱ are equivalent and the eigenvalue λ_3 has the smallest order. The optimal solution is therefore given by the solution of $\eta_{\rm r}^{\rm opt}$ and the leading terms of the relevant quantities become

$$\eta_{\rm max} = \pi\sqrt{2}\sqrt{T}\, {\rm e}^{\frac{1}{2}\hat{\varrho}^2} \left[1 - \frac{\sqrt{T}}{K} {\rm e}^{\frac{1}{2}\hat{\varrho}^2} + \frac{1 + 4T_\infty + 4T_\infty^2\, {\rm e}^{\hat{\varrho}^2} - \hat{\varrho}^2}{4T} \right], \tag{5.B.15a}$$

$$\eta_0^{\rm opt} = \eta_{\rm max} - \frac{\pi\sqrt{2}}{2\sqrt{T}(1+\hat{\varrho}^2)} {\rm e}^{\frac{1}{2}\hat{\varrho}^2}, \tag{5.B.15b}$$

$$\eta_w^{\text{opt}} = \eta_{\max} - \frac{\pi\sqrt{2}}{2\sqrt{T}}\,\mathrm{e}^{\frac{1}{2}\hat{\varrho}^2}, \tag{5.B.15c}$$

$$\lambda_0^{\text{opt}} = -\frac{2}{KT(1+\hat{\varrho}^2)}\left[1 - \frac{\sqrt{T}\,\mathrm{e}^{\frac{1}{2}\hat{\varrho}^2}}{K} + \frac{T_\infty^2\,\mathrm{e}^{\hat{\varrho}^2} + T_\infty}{T} + \frac{\hat{\varrho}^4 + 4\hat{\varrho}^2 - 2}{2T(1+\hat{\varrho}^2)}\right], \tag{5.B.15d}$$

$$\lambda_w^{\text{opt}} = -\frac{2}{KT}\left[1 - \frac{\sqrt{T}\,\mathrm{e}^{\frac{1}{2}\hat{\varrho}^2}}{K} + \frac{2T_\infty^2\,\mathrm{e}^{\hat{\varrho}^2} + 2T_\infty + \hat{\varrho}^2 - 2}{2T}\right]. \tag{5.B.15e}$$

The comparison for zero biases ($\hat{\varrho} = 0$) reveals that in this limit, the existence of biases slows down the training process to leading order only in the case where $\eta_\theta = \eta_w$. Furthermore, this decrease is surprisingly only algebraic in $\hat{\varrho}$. This can be explained by the exponential growth of the optimal learning rates matching the gradient decrease due to the saturation of the error function for large $\hat{\varrho}$. Again, this solution is only a good approximation for finite K and T as long as $\hat{\varrho}^2 \ll \log K$ and $\hat{\varrho}^2 \ll \log T$.

5.B.5 Large-T limit

Unlike for small T, the learning behaviour changes qualitatively in the $T \to \infty$ limit for K finite, as indicated by numerical solutions. Again λ_3 controls the convergence and one finds to leading orders

$$\eta_{\max} = \pi\sqrt{2}K\left[1 - \frac{K-1}{\sqrt{T}}\,\mathrm{e}^{-\frac{1}{2}\hat{\varrho}^2}\right], \tag{5.B.16a}$$

$$\eta_0^{\text{opt}} = \eta_{\max} - \frac{\pi\sqrt{2}K}{2(1+\hat{\varrho}^2)T}, \tag{5.B.16b}$$

$$\lambda_0^{\text{opt}} = -\frac{2}{(1+\hat{\varrho}^2)T^{\frac{3}{2}}}\,\mathrm{e}^{-\frac{1}{2}\hat{\varrho}^2}\left[1 - \frac{K-1}{\sqrt{T}}\,\mathrm{e}^{-\frac{1}{2}\hat{\varrho}^2}\right], \tag{5.B.16c}$$

$$\eta_w^{\text{opt}} = \eta_{\max} - \frac{\pi\sqrt{2}K}{2T}, \tag{5.B.16d}$$

$$\lambda_w^{\text{opt}} = -\frac{2}{T^{\frac{3}{2}}}\,\mathrm{e}^{-\frac{1}{2}\hat{\varrho}^2}\left[1 - \frac{K-1}{\sqrt{T}}\,\mathrm{e}^{-\frac{1}{2}\hat{\varrho}^2}\right]. \tag{5.B.16e}$$

In this case, the optimal learning rate is independent of $\hat{\varrho}$ to leading order in T. The exponentionally decreasing gradient therefore directly affects the optimal convergence rate.

Chapter 6

On-Line Learning with Adaptive Back-Propagation in Two-Layer Networks

Abstract

As seen in the previous chapter, training with on-line gradient descent (standard back-propagation) can severely slow down the learning process in multilayer networks, if the task to be learnt exhibits symmetries. In order to gain an understanding of this behaviour and to identify possible improvements, an adaptive back-propagation algorithm parameterized by an inverse temperature β is studied and compared with gradient descent which is recovered in the special case $\beta = 1$ for soft-committee machines with an arbitrary number of hidden units. In the framework developed in the previous chapter, we analyse these learning algorithms in both the symmetric and the convergence phase for finite learning rates in the case of uncorrelated teachers of similar but arbitrary length T. These analyses show that adaptive back-propagation results generally in faster training by breaking the symmetry between hidden units more efficiently and by providing faster convergence to optimal generalization than gradient descent.

6.1 Introduction

In the previous chapter, we have found that in the early stages of training the student network is drawn into a suboptimal symmetric phase, characterized by undifferentiated imitation, by student vectors, of parameter vectors related to the various teacher

hidden nodes. Although student node symmetry is eventually broken and student performance converges to the minimal achievable generalization error, a significant part of the training time may be spent with the system trapped in the symmetric subspace in the case where the teacher task exhibits many symmetries. Speeding up the escape from the symmetric phase is likely to improve the training efficiency significantly[1]. In this chapter, we suggest a simple modification of the basic back-propagation algorithm and analyse the resulting expected improvement in training efficiency.

The need for improved neural network training methods is clear as training efficiency is in the heart of the method itself and plays a significant rôle in determining the usefulness of the method as a whole; new tools may enable us to obtain better performance in shorter training times as well as to expand the envelope of feasible tasks. For batch training there is a variety of efficient training methods available, such as second-order methods (e.g., Newton-Raphson or conjugate gradient). However, as these methods are based on the entire training set they are not applicable to on-line learning. Several different methods have been employed for improving on-line training in both discrete and smooth networks, most of which are based on heuristics or on analysis in the asymptotic regime.

Among the most common modifications to the conventional back-propagation algorithm, for smooth systems, is training with momentum. An analysis using stochastic approximation theory (Leen and Orr 1994) shows that for learning large example sets it merely rescales the learning rate in the convergence phase. Similar trivial effects are also mirrored in the statistical mechanics framework (Prügel-Bennett 1996), unless different scaling is used for the learning rate term. Its usefulness is so far inconclusive. Other methods aimed at incorporating information about the curvature of the error surface into the learning rule have been proposed recently (Leen and Orr 1994; Amari 1997b; Amari 1997a). These rules are expected to be efficient asymptotically, although their effect on earlier stages of the learning process and especially on the length of the symmetric phase is not yet clear.

Several efficient methods have been suggested for on-line learning in discrete networks. Some of the methods are based on a greedy maximization of the local difference in generalization error (Kinouchi and Caticha 1992; Copelli and Caticha 1995; Copelli et al. 1996), while others are based on structured learning rules (Biehl and Riegler 1994; Kim and Sompolinsky 1996). It is, however, unclear whether these methods can be extended to accommodate smooth multilayer networks such as the soft-committee

[1]A suitable working definition for efficiency of training algorithms may be their speed of convergence to an "acceptable" generalization error, in terms of training time or the number of example presentations.

machine (Biehl and Schwarze 1995; Saad and Solla 1995b) and whether these extensions would be useful in devising an efficient method for escaping the symmetric phase, especially since applying local optimization in this phase is likely to fail [as demonstrated in (Saad and Rattray 1997a; Rattray and Saad 1997a)].

A method for breaking the symmetry of the student network in smooth machines by enforcing a weight-ordering penalty term on the space of hidden units has been suggested in (Barber et al. 1996), showing a considerable improvement in training time for a very simple network architecture. A more detailed numerical investigation, however, shows that this method fails completely in the case of isotropic teacher networks, with uncorrelated teacher weight vectors of similar length, where the student remains indefinitely trapped in a suboptimal symmetric phase[2]. In the case of a soft-committee machine where biases are applied to the hidden layer nodes, as is the case in realistic networks, it has become evident in Chapter 5 that the strongest symmetry-breaking effect is provided by the network biases, possibly leading to a stagnating competition in breaking the symmetry between biases and the weight-ordering penalty term.

The aim of this chapter is twofold. It gives some insight into the reasons for the short-comings of back-propagation and it furthermore investigates possible improvements by introducing an adaptive back-propagation algorithm. This algorithm features, besides the learning rate η, a second adaptable parameter, the inverse temperature β, which improves the ability of the student to distinguish between hidden nodes of the teacher for $\beta > 1$. We compare its efficiency with that of gradient descent in training two-layer networks following the framework developed in the previous chapter and present numerical studies and rigorous analyses of both the breaking of the symmetric phase and the asymptotic convergence. We note that although these analyses provide us with optimal values of the user adjustable parameters η and β for different stages of the training process in a range of learning scenarios, it remains an open question how these parameters can be optimized adaptively on-line without a priori knowledge of the training task[3]. Within this limitation, we find that the optimized adaptive back-propagation can significantly reduce training time in both regimes by efficiently breaking the symmetry between hidden units and by providing faster exponential convergence asymptotically.

[2] A presentation of this quite substantial numerical work goes beyond the scope of this thesis.

[3] The term *adaptive* in ABP therefore refers to the adjustability of β and to the subsequent deformation of the search space. It does not imply an ability of the algorithm to tune its adjustable parameters on-line.

6.2 Adaptive back-propagation and dynamical equations

Although the framework employed in this chapter is very similar to the previous one, let us briefly introduce the modified model. The considered student network is a fully-connected normalized soft-committee machine with K hidden units and parameters $\boldsymbol{\Omega} \equiv \boldsymbol{W} = \{\boldsymbol{W}_i\}$ implementing the mapping

$$\sigma(\boldsymbol{\xi}; \boldsymbol{W}) = \frac{\gamma}{\sqrt{K}} \sum_{i=1}^{K} g\left(\frac{1}{\sqrt{N}} \boldsymbol{W}_i \cdot \boldsymbol{\xi}\right) = \frac{\gamma}{\sqrt{K}} \sum_{i=1}^{K} g\left(x_i\right), \tag{6.1}$$

where $x_i = \boldsymbol{W}_i \cdot \boldsymbol{\xi}/\sqrt{N}$ refers to student activations and $g(\cdot)$ is a sigmoidal transfer function as before. Similarly, the teacher is defined by a network of the same architecture and M hidden units with parameters $\boldsymbol{\Omega}_0 \equiv \boldsymbol{B} = \{\boldsymbol{B}_n\}$, that provides noise-free output labels ζ^μ to the randomly drawn input pattern[4] $\boldsymbol{\xi}^\mu$, i.e.,

$$\zeta^\mu = \zeta_0(\boldsymbol{\xi}^\mu; \boldsymbol{B}) = \frac{\gamma}{\sqrt{M}} \sum_{n=1}^{M} g\left(\frac{1}{\sqrt{N}} \boldsymbol{B}_n \cdot \boldsymbol{\xi}^\mu\right) = \frac{\gamma}{\sqrt{M}} \sum_{n=1}^{M} g\left(y_n^\mu\right), \tag{6.2}$$

where $y_n^\mu = \boldsymbol{B}_n \cdot \boldsymbol{\xi}^\mu/\sqrt{N}$ denotes teacher activations[5] as previously.

A generic on-line training algorithm $\mathcal{A}$ can in general be defined by the update of each parameter in response to the presentation of a single example $(\boldsymbol{\xi}^\mu, \zeta^\mu)$, which in our case can take the general form

$$\boldsymbol{W}_i^{\mu+1} = \boldsymbol{W}_i^\mu + \mathcal{A}_i(\{\gamma\}, \boldsymbol{W}^\mu, \boldsymbol{\xi}^\mu, \zeta^\mu), \tag{6.3}$$

where $\{\gamma\}$ defines parameters adjustable by the user. In the case of gradient descent, or (standard) back-propagation, on the quadratic error function (5.3), as studied in the previous chapter, this results in

$$\mathcal{A}_i^{\mathrm{GD}}(\eta, \boldsymbol{W}^\mu, \boldsymbol{\xi}^\mu, \zeta^\mu) = \eta \delta_i^\mu \boldsymbol{\xi}^\mu, \tag{6.4a}$$

with

$$\delta_i^\mu = \delta^\mu g'(x_i^\mu) = [\zeta^\mu - \sigma(\boldsymbol{\xi}^\mu; \boldsymbol{W})]\, g'(x_i^\mu), \tag{6.4b}$$

[4]The components of $\boldsymbol{\xi}^\mu$ are again drawn independently from a zero mean Gaussian distribution with arbitrary variance σ^2.

[5]Note that we will again use indices i, j, k, l to refer to units in the student network and n, m for units in the teacher network.

where the only user adjustable parameter is the (weight) learning rate η. One can readily see that each of the three terms in the back-propagation weight update plays a different rôle. The difference δ^μ between the student output and the target together with the learning rate determines the overall size of the update of all weight parameters by specifying how closely student and teacher are matched. The input vector $\boldsymbol{\xi}^\mu$ discriminates between the weights leading to different inputs. However, only $g'(x_i^\mu)$, i.e., the derivative of the transfer function $g(\cdot)$, breaks the symmetry between different hidden units. The fact that a prolonged symmetric phase can exist indicates that this term is not significantly different over the hidden units for a typical input in the symmetric phase.

The rationale of the adaptive back-propagation algorithm defined below is therefore to alter the g'-term, in order to magnify small differences in activation between hidden units. A simple way of enhancing these differences is by altering $g'(x_i)$ to $g'(\beta x_i)$, where β plays the role of an inverse "temperature." Varying β changes the range of hidden unit activations relevant for training, e.g., for $\beta > 1$ learning is more confined to small activations, when compared to gradient descent ($\beta = 1$), i.e., the training process is effectively "frozen" for larger activations. One could also absorb this modification into gradient descent with a site and activation dependent learning rate, making it more obvious that adaptive back-propagation deforms the search space spatially. The adaptive back-propagation learning rule is therefore

$$\mathcal{A}_i^{\mathrm{ABP}}(\eta, \beta, \boldsymbol{W}^\mu, \boldsymbol{\xi}^\mu, \zeta^\mu) = \eta \delta^\mu g'(\beta x_i^\mu) \boldsymbol{\xi}^\mu = \eta \widetilde{\delta}_i^\mu \boldsymbol{\xi}^\mu, \tag{6.5}$$

with δ^μ as in Eq. (6.4b). To compare the adaptive back-propagation (ABP) algorithm with conventional gradient descent (GD), we follow the framework described in Section 5.2. After the introduction of the self-averaging dynamical order parameters Q_{ij} and R_{in}, and the task specific fixed order parameters T_{nm} (5.6), the averaged update equations can again be written in the thermodynamic limit $N \to \infty$ as first-order differential equations in $\alpha = \mu/N$, resulting in

$$\frac{dR_{in}}{d\alpha} = \eta \left\langle \widetilde{\delta}_i y_n \right\rangle_{\{\boldsymbol{x},\boldsymbol{y}\}}, \tag{6.6a}$$

$$\frac{dQ_{ij}}{d\alpha} = \eta \left\langle \widetilde{\delta}_i x_j + \widetilde{\delta}_j x_i \right\rangle_{\{\boldsymbol{x},\boldsymbol{y}\}} + \eta^2 \left\langle \widetilde{\delta}_i \widetilde{\delta}_j \right\rangle_{\{\boldsymbol{x},\boldsymbol{y}\}}. \tag{6.6b}$$

For the error function chosen in Chapter 5, all the integrals in Eqs. (6.6) and the generalization error (5.5) can be calculated analytically unlike for the case of non-zero bias. For the exact form of the dynamical equations and the generalization error, we refer the reader to Appendix 6.A. We only mention in passing that the same rescaling

for the sigmoidal gains ν, the output gain γ, and the variance of the input distribution σ^2 hold as in Chapter 5, such that in the following these will be set to one w.l.o.g..

6.3 Numerical Integration of the dynamical equations

The differential equations can easily be integrated numerically for any number of K student and M teacher hidden units, which is in contrast to the case of dynamic biases studied in Chapter 5, where the numerical integration of some of the averages restricted us to small networks. For the remainder of the chapter, we will focus on the realizable case ($K = M$) and uncorrelated isotropic teachers of arbitrary length $T_{nm} = T\delta_{nm}$ in order to be able to push some of the analyses carried out in Chapter 5 even further.

As before, the dynamical evolution of the overlaps Q_{ij} and R_{in} follows from integrating the equations of motion (6.6) from initial conditions determined by the (random) initialization of the student weights $\boldsymbol{W}_i$. Following the considerations of Section 5.3, the random initialization of the weights can again be simulated by initializing the norms Q_{ii}, and the normalized overlaps $\hat{Q}_{ij} = Q_{ij}/\sqrt{Q_{ii}Q_{jj}}$ and $\hat{R}_{in} = R_{in}/\sqrt{Q_{ii}T_{nn}}$ from uniform distributions in the $[0, 1]$ and $[-10^{-12}, 10^{-12}]$ intervals, respectively.

In Figure 6.1 we show a typical difference in the evolution of the overlaps and the generalization error for $\beta = 12$ and $\beta = 1$ (gradient descent) for $K = 3$ and $\eta = 0.03$. In both cases, the student is drawn quickly into a suboptimal symmetric phase, characterized by a finite generalization error [Figure 6.1(e)] and no differentiation between the hidden units of the student. The student norms Q_{ii} and overlaps Q_{ij} are similar [Figures 6.1(a,c)], i.e., the students are highly correlated with each other. The overlaps of each student node with all teacher nodes R_{in} are nearly identical [Figures 6.1(b,d)], i.e., each student unit imitates all teacher units with similar success. The student trained by GD [Figures 6.1(c,d)] is trapped in this unstable suboptimal solution for most of the training time, whereas ABP [Figures 6.1(a,b)] breaks the symmetry significantly earlier. The convergence phase is characterized by a specialization of each student nodes to a particular teacher node, which corresponds to an evolution of the overlap matrices $\mathbf{Q}$ and $\mathbf{R}$ to their optimal value $\mathbf{T}$, except for the permutational symmetry due to the arbitrary labelling of the student nodes.

Examining the decay of the generalization error in Figure 6.1(e) more closely, one can see that the choice $\beta = 12$ is suboptimal in this regime. The student trained with $\beta = 1$ converges faster to zero generalization error. In order to optimize both the learning temperature β and the learning rate η simultaneously for both phases of the learning process, the symmetric and the convergence phase, we will examine the equations of motions analytically in the following section.

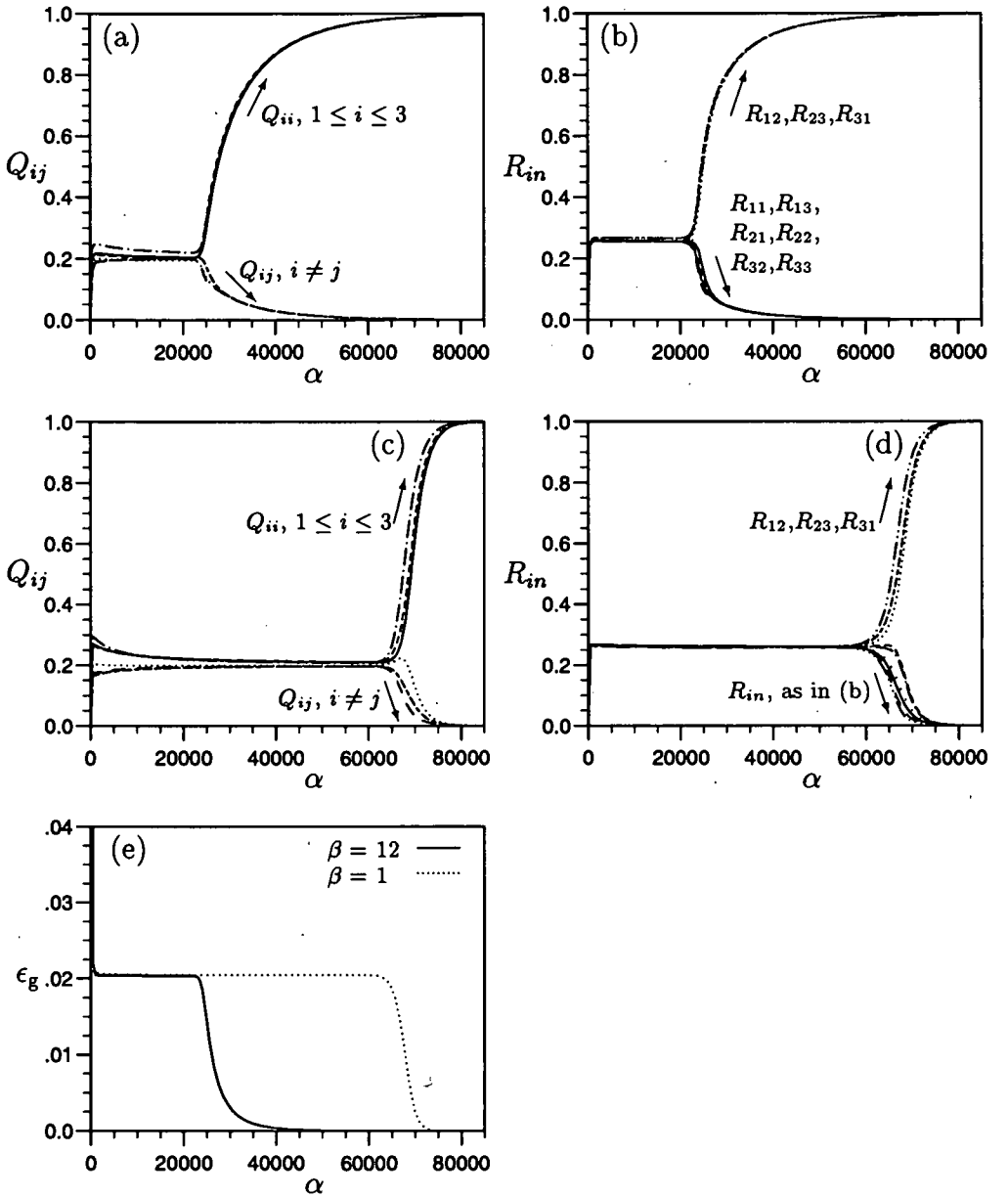

Figure 6.1. Dynamical evolution of the student-student overlaps Q_{ij} (a,c), the student-teacher overlaps R_{in} (b,d), and the generalization error (e) as a function of the normalized example number α for a student with three hidden nodes learning an isotropic three-node teacher (T_{nm}=δ_{nm}). The learning rate η=0.03 is fixed, but the value of the inverse temperature varies (a,b): β=12 and (c,d): β=1 (gradient descent).

6.4 Analysis of the dynamical equations

For the model with fixed-zero biases, realizable learning scenario ($K = M$) and isotropic teacher tasks ($T_{nm} = T\delta_{nm}$) the order parameter space can be very well characterized

by similar diagonal and off-diagonal elements of the overlap matrices $\mathbf{Q}$ and $\mathbf{R}$ for the whole learning process, justifying the ansatz

$$Q_{ij} = Q\delta_{ij} + C(1-\delta_{ij}) \quad \text{and} \quad R_{in} = R\delta_{in} + S(1-\delta_{in}) \tag{6.7}$$

for the student-student overlaps and (apart from a relabelling of the student nodes) student-teacher overlaps respectively. As one can see from Figure 6.1, this approximation is particularly good in the symmetric phase and during the final convergence to perfect generalization.

The reduction of the number of order parameters from $\mathcal{O}(K^2)$ to just four simplifies the differential equations and the generalization error significantly (see Appendix 6.B). This allows us to analyse the learning dynamics exactly as a function of the size of the network K, the length of the teacher hidden units T, and the user adjustable training parameters: the learning rate η and the learning temperature β.

6.4.1 Symmetric phase and onset of specialization

Numerical integration of the equations of motion for a range of learning scenarios shows that the length of the symmetric phase depends on the number of hidden units K, the anisotropy in the length of the teacher vectors, the choice of the user adjustable parameters η and β, and the anisotropy of the initial conditions. If we assume that the initial conditions are random and K is fixed, the trapping in the symmetric phase is especially prolonged by isotropic teachers and small learning rates η.

Initially, we will therefore study the dynamics (6.6) analytically in the symmetric phase for isotropic teachers in the small-η regime, where terms proportional to η^2 can be neglected. Later, the effect of a finite learning rate, i.e., including η^2 terms, will be studied analytically for small η and numerically for arbitrary η.

Truncated equations

The truncated equations of motion have only one physical fixed point, given by

$$Q_0^* = C_0^* = \frac{T}{K(1+T)-T}, \tag{6.8a}$$

$$R_0^* = S_0^* = \sqrt{\frac{Q^*T}{K}} = \frac{T}{\sqrt{K[K(1+T)-T]}}, \tag{6.8b}$$

which is independent of β and therefore identical to the one obtained by Saad and Solla (1995b) for $T = 1$. The fixed point can be understood in geometrical terms: the student weight vectors are confined to the subspace spanned by the teacher weight

vectors and their projection onto each teacher weight vector is identical. However, this symmetric solution is an unstable fixed point of the dynamics and the small perturbations introduced by the generically nonsymmetric initial conditions will eventually drive the student towards specialization.

To study the onset of specialization, we expand the truncated differential equations to first order in the deviations $q = Q - Q_0^*$, $c = C - C_0^*$, $r = R - R_0^*$, and $s = S - S_0^*$ from the fixed point values (6.8). The linearized equations of motion take the form $d\boldsymbol{v}/d\alpha = \mathbf{M}\cdot\boldsymbol{v}$, where $\boldsymbol{v} = (r, s, q, c)$ and $\mathbf{M}$ is a 4×4 matrix whose elements are the first derivatives of the truncated update equations (6.B.2) at the fixed point with respect to $\boldsymbol{v}$. Perturbations or *modes* which are proportional to the *eigenvectors* $\boldsymbol{v}_i$ of $\mathbf{M}$ will therefore decrease or increase exponentially depending on whether the corresponding *eigenvalue* λ_i is negative or positive. For the onset of specialization only the modes with positive eigenvalue are relevant, being amplified by the dynamics. For them we can identify the inverse eigenvalue as a typical escape time τ_i from the symmetric phase.

For the truncated equations of motion, we find only one relevant perturbation [see Appendix 6.C.1, Eqs. (6.C.4) and (6.C.5)] with an associated eigenvector implying $q = c = 0$ and $s = -r/(K-1)$, i.e., a pure rotation of the student weight vectors inside the subspace spanned by the teacher weight vectors towards the teacher unit they will specialize on. This can also be confirmed by a closer look at Figure 6.1. The onset of specialization is signalled by the breaking of the symmetry between the student-teacher overlaps, whereas significant differences from the symmetric fixed point values of the student norms and overlaps occur later. The escape eigenvalue is

$$\lambda_0(\beta) = \frac{2}{\pi}\frac{\eta\beta T^2}{\sqrt{K(1+T)-T}\,[K(1+T)+\beta T]^{3/2}}. \tag{6.9}$$

Maximization of $\lambda_0^{\mathrm{opt}}(\beta)$ with respect to β yields

$$\beta^{\mathrm{opt}} = 2\frac{K(1+T)}{T}, \tag{6.10}$$

i.e., the optimal β scales with the number of hidden units and also grows proportionally to $1/T$ for small teacher lengths. The optimized escape eigenvalue is

$$\begin{aligned}\lambda_0^{\mathrm{opt}}(\beta^{\mathrm{opt}}) &= \frac{4\sqrt{3}}{9\pi}\frac{\eta T}{\sqrt{K(1+T)}\sqrt{K(1+T)-T}} \\ &= \lambda_0^{\mathrm{opt}}(1)\frac{2\sqrt{3}}{9}\frac{[K(1+T)+T]^{3/2}}{T\sqrt{K(1+T)}}.\end{aligned} \tag{6.11}$$

Trapping in the symmetric phase is therefore for very small learning rates always inversely proportional to the learning rate η. It is interesting to study two limiting cases: $K \to \infty$, i.e., large networks, and $T \to 0$, i.e., small teacher weights or nearly linear functions. In these limits, one finds that the escape eigenvalue is $\lambda \propto 1/K^2$ ($\lambda \propto T^2$) for GD in contrast to $\lambda \propto 1/K$ ($\lambda \propto T$) for optimized ABP respectively, i.e., in these limits the time spent in the symmetric phase can be reduced by an order of K or $1/T$.

Small-η expansion

Numerical integrations of the differential equations (6.A.2) for larger learning rates indicate a reduced optimal value of β, with the ansatz (6.7) still valid. It is therefore desirable to analyse the symmetric phase for finite learning rates.

Analytically, we can expand the full set of equations (6.B.2) to first-order in $\boldsymbol{v} = (r, s, q, c)$ around the fixed point of zeroth order (6.8) and find its first order correction in η by solving the resulting set of linear equations. The new fixed point found is still characterized by $Q^* = C^*$ and $R^* = S^*$ [Eq. (6.C.6)]. This is in contradiction to the numerical results, which predict a fixed point with $Q^* > C^*$ and $R^* = S^*$. This contradiction can be resolved by studying the linear dynamics around the new fixed point. An eigenvalue that was marginal ($\lambda_2 = 0$) for the truncated equations of motions acquires a positive contribution of $\mathcal{O}(\eta^2)$ [Eq. (6.C.7)]. The mode associated with this eigenvalue increases differences between Q and C, leading primarily to a growth of the student weight vectors outside the subspace spanned by the teacher weight vectors (see Appendix 6.C.3) and no specialization. As these differences are typically large for random initial conditions (unlike differences in R and S), this mode will drive the student quickly away from the fixed point characterized by $Q^* = C^*$ to one with $Q^* > C^*$, where the student will be trapped until specialization between R and S will occur eventually. Unfortunately, this fixed point cannot be studied analytically, but can, however, be studied numerically.

Numerical finite-η analysis

In Figure 6.2(a) the order parameter values are shown at the fixed point, which are characterized by $Q^* > C^*$ and $R^* = S^*$ for finite-η values. Whereas R^* is nearly constant over a wide range of learning rates, the value of Q^* increases and C^* decreases rapidly. In fact, as η approaches a certain value, termed here η_D, the values of the order parameters diverge.

The eigenvalue spectrum: This behaviour can be understood by linearizing the dynamics around the fixed point and analysing its eigenvalues [see Figure 6.2(b)]. We

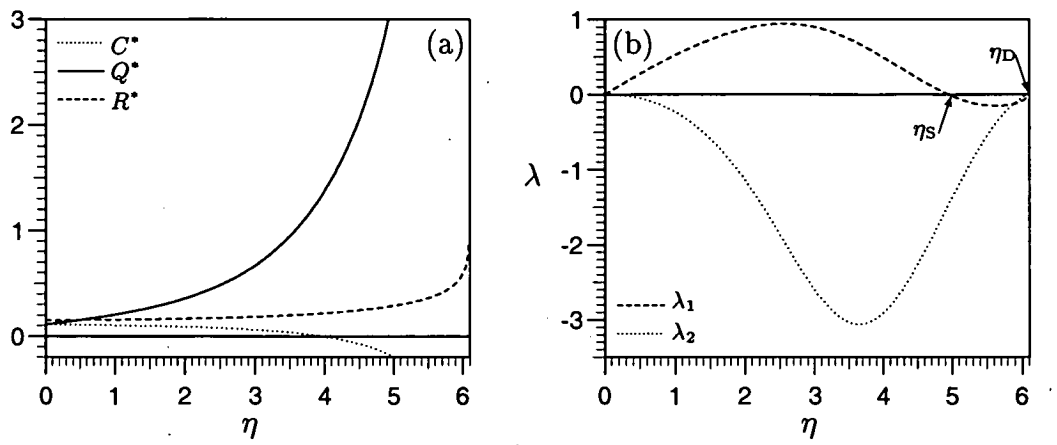

Figure 6.2. (a) The symmetric fixed-point values R^*, Q^*, and C^* of the order parameters are shown as a function of the learning rate η at $K = 5$ and $T = 1$ for $\beta = 1$. The values of the order parameters diverge for $\eta \to \eta_D$ (see the text). (b) For the same parameters, the relevant eigenvalues λ_1, λ_2 (see the text) of the linearized dynamics around the (learning-rate-dependent) symmetric fixed point explain the divergent behaviour as $\lambda_2(\eta_D) \to 0$. The maximum in λ_1, the eigenvalue that drives the specialization process, defines the optimal learning rate.

find two eigenvalues that are always negative and of large magnitude and are therefore irrelevant to the long-term behaviour of the dynamics. For the other two eigenvalues one finds that $\lambda_1 > 0$ and $\lambda_2 < 0$ for small to intermediate learning rates. The eigenvector associated with λ_1 is in fact identical to the one found for fixed points with $Q^* = C^*$ and corresponds to a pure rotation and instability in R-S space. The eigenvector of λ_2 is also very similar to the eigenvector of the eigenvalue that caused the instability of the $Q^* = C^*$ fixed point in the Q-C space. For increasing learning rate, we first find a global maximum for λ_1 at the optimal learning rate $\eta^{\text{opt}}(\beta)$. For even larger learning rates, we find different generic behaviours, depending on the values of the parameters K, T, and β. In general, there are two candidates for a maximal learning rate η_{max} identifiable in Figure 6.2(b). The first, η_D, corresponds to λ_2 becoming positive, causing an instability in Q-C space and diverging values of the order parameters. The other candidate is given by the learning rate η_S, where λ_1 turns negative and the fixed point becomes attractive. One can identify two phases: $\eta_S < \eta_D$ and $\eta_D > \eta_S$ (for which η_S does not actually exist since the fixed point vanishes above η_D). However, in the following we will not distinguish between these two phases, but simply define $\eta_{\text{max}} = \min(\eta_D, \eta_S)$.

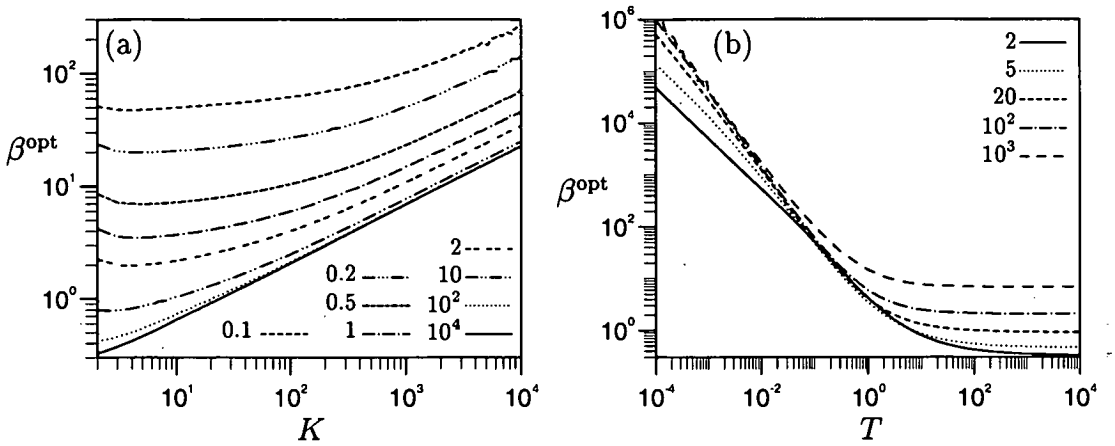

Figure 6.3. (a) The optimal inverse temperature β^{opt} is shown for various T values (see the legend) as a function of K. For sufficiently large values of TK, β^{opt} grows with $\sqrt{K}$. (b) Here β^{opt} is shown as a function of T for various K values (see the legend). For small T we find a power-law increase of β with $1/T$ with an exponent that approaches 1 for TK small enough.

Optimal inverse temperature β^{opt}: In order to estimate the potential gain by using ABP in the finite learning rate case, we optimize the dynamics with respect to the learning rate η under the constraint $\beta = 1$ (GD) and contrast it with results obtained by optimizing with respect to both the learning rate η and the inverse temperature β (ABP) for a range of K and T values. In Figure 6.3 the optimal value of β is shown as a function of both K and T. Figure 6.3(a) shows that β^{opt} increases for growing network size K, as is expected from the small learning rate analysis. However, the size of β^{opt} grows significantly slower and becomes dependent on the value of the product TK. For $TK \gg 1$ and $K \to \infty$ one finds $\beta^{opt} \propto \sqrt{K}$, which has to be contrasted with the previously predicted $\beta^{opt} \propto K$ [see Eq. (6.10)], due to the influence of finite learning rates. Similarly, as shown in Figure 6.3(b), β^{opt} grows for decreasing teacher lengths T but remains constant for large T as predicted previously. We find power laws for $T \to 0$, with exponents dependent on the value of TK. For $TK \ll 1$ however, the exponent approaches -1, which is identical to the theoretical prediction in the small-η regime.

Optimal learning rates and escape eigenvalues: Having identified the two interesting regimes, where the optimal inverse temperature deviates significantly from its GD value, small teacher weight vectors $T \to 0$ and large networks $K \to \infty$, we investigate the differences in optimal dynamics for GD and ABP further. In Figure 6.4

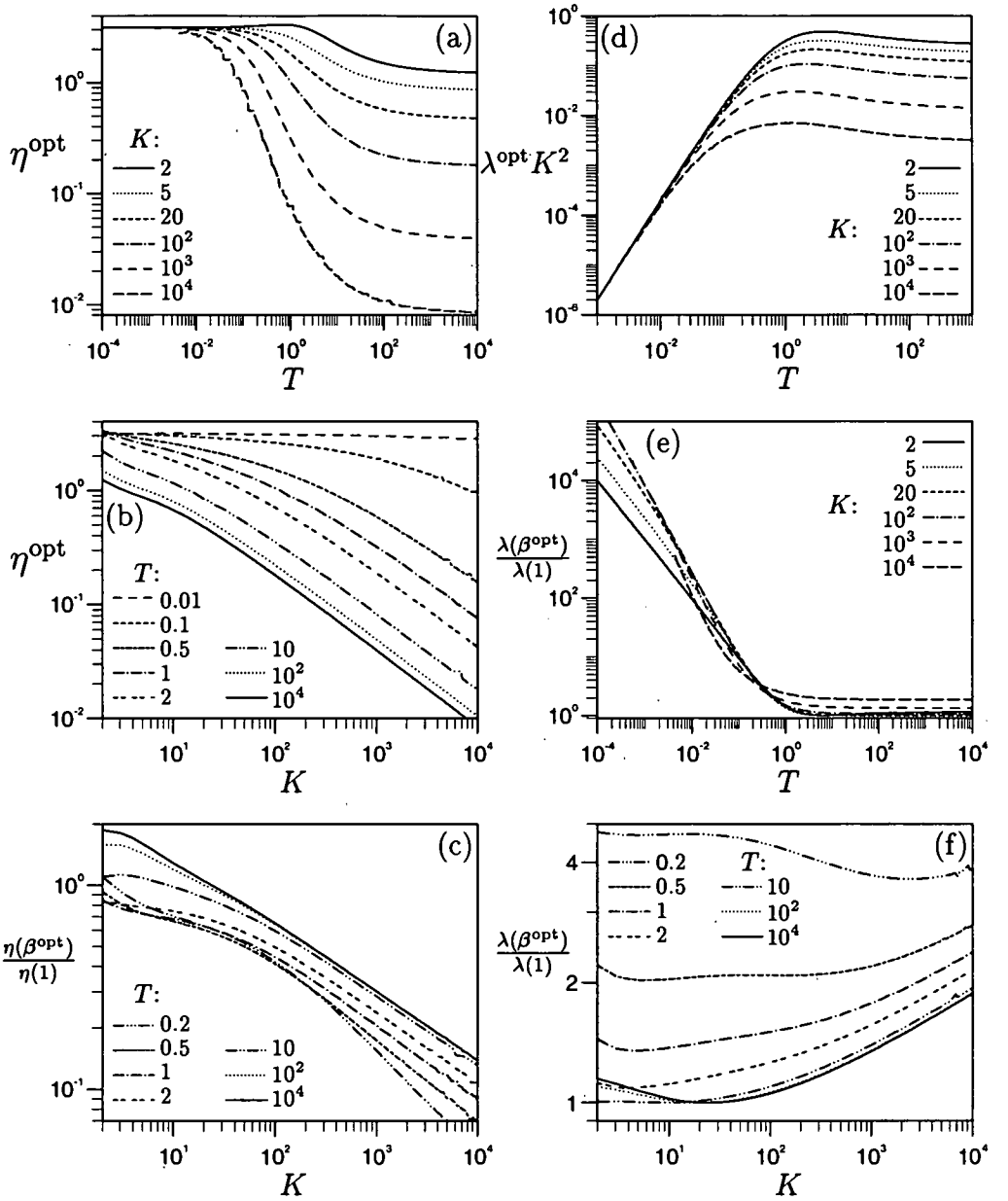

Figure 6.4. (a) The optimal learning rate $\eta^{\rm opt}(T)$ for GD shows the most volatile behaviour for $0.1 \leq T \leq 10$. (b) $\eta^{\rm opt}(K)$ shows a power-law decay for $TK \gg 1$. (c) The quotient of the optimal learning rates of ABP and GD shows that $\eta^{\rm opt}(\beta^{\rm opt})$ decays even faster for large K. (d) The optimal escape eigenvalue $\lambda^{\rm opt}(T)$ for GD (multiplied by K^2) collapses on a universal (K-independent) curve for small T, but acquires a further K dependence for large T. The possible gain by using ABP is shown by plotting the quotient of the optimal escape eigenvalue for the two training algorithms. The advantage of ABP is most impressive for small T (see the legend for K values) (e), but shows also a power-law gain for large K (see the legend for T values) (f).

we show the behaviour of both the optimal learning rate $\eta^{\rm opt}$ [Figures 6.4(a–c)] and the resulting optimal escape eigenvalue $\lambda^{\rm opt}$ [Figures 6.4(d–f)] for GD in comparison to ABP for various K-T scenarios.

The optimal learning rate $\eta^{\rm opt}(T)$ of GD, depicted in Figure 6.4(a), exhibits a strongly K-dependent limit for large T and a universal limit for small T. In general, $\eta^{\rm opt}(T)$ decreases for increasing T and shows its most volatile behaviour in the region $0.1 \leq T \leq 10$ and for large K. These teacher values are the most reasonable for real learning problems, i.e., in practice it will be generally difficult to choose a good learning rate especially for large networks. This picture can be confirmed by examining the influence of K on $\eta^{\rm opt}$ for GD as shown in Figure 6.4(b). For very small T, the learning rate exhibits hardly any dependence on K, whereas for TK large enough, one finds that $\eta^{\rm opt} \propto K^{-\frac{2}{3}}$.

The behaviour of the optimal learning rate for optimized ABP is quite similar to GD. The main difference from GD can be seen in Figure 6.4(c), which shows that $\eta^{\rm opt}(\beta^{\rm opt})$ decays faster for ABP, with $\eta^{\rm opt}(\beta^{\rm opt}) \propto K^{-1}$ for large TK. One also finds that the optimal learning rate saturates for large- and small-T values to K-dependent constants. For large T this may be explained by the fact that the error is limited by the saturation of all units.

The optimized escape eigenvalue, which largely determines the training time spent in the symmetric phase, is shown for GD in Figure 6.4(d), where we have multiplied $\lambda^{\rm opt}$ by K^2 for convenience. For small T, one finds that $\lambda^{\rm opt}(T)$ collapses on universal curve for all K and we find the same power-law behaviour as predicted in the small-η analysis ($\lambda^{\rm opt} \propto T^2/K^2$) [see Eq. (6.9)]. For large T, one also finds that $\lambda^{\rm opt}$ becomes increasingly weakly dependent on T as expected. However, it also shows a further K dependence due to the decay of the optimal learning rate and one finds $\lambda^{\rm opt} \propto \eta^{\rm opt}/K^2$.

To highlight the possible gains of using ABP, $\lambda^{\rm opt}(\beta^{\rm opt})/\lambda^{\rm opt}(1)$ is plotted as a function of T and K in Figures 6.4(e,f). In Figure 6.4(e), one finds for small T a gain[6] of $1/T$ for $TK \ll 1$, which was predicted from the small-η analysis [see Eq. (6.11)]. For large K [see Figure 6.4(f)] we also find a power-law gain in K for the optimized dynamics, but only for $TK \gg 1$ and with an exponent that is only $1/6$, much smaller than the value of 1 predicted previously in Eq. (6.11). This can be attributed to the slower than predicted increase in $\beta^{\rm opt}$ and to the smaller optimal learning rate for ABP in this regime.

[6] Although we find numerically exponents for T smaller than -1 for larger K, it remains unclear if these hold even for $TK \ll 1$. However, for $K \to \infty$, $T \to 0$, and $TK = \text{const.}$, the power law seems to approach $-3/2$.

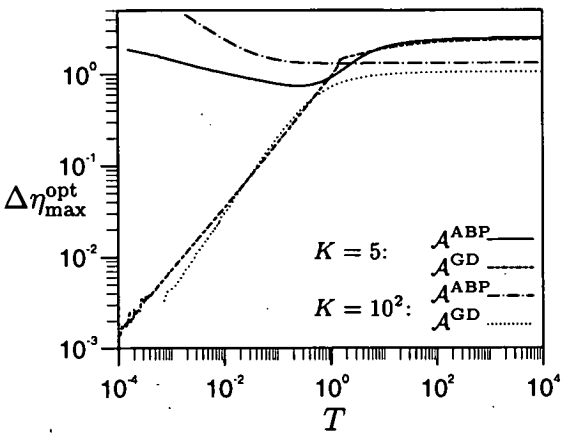

Figure 6.5. The normalized difference between the maximal and optimal learning rate $\Delta\eta_{\max}^{\text{opt}} = (\eta_{\max} - \eta^{\text{opt}})/\eta^{\text{opt}}$ is shown for both adaptive back-propagation $\mathcal{A}^{\text{ABP}}$ and gradient descent $\mathcal{A}^{\text{GD}}$ for $K = 5, 100$ as a function of T.

Maximal learning rates: Of arguably further importance for training is the sensitivity of the choice of the learning rate, especially in the sense of how well the maximal learning rate is separated from its optimal value. Therefore, the normalized difference between the maximal and optimal learning rate $\Delta\eta_{\max}^{\text{opt}} = (\eta_{\max} - \eta^{\text{opt}})/\eta^{\text{opt}}$ is compared for ABP and GD as a function of T for two K values in Figure 6.5. Whereas the optimal and maximal learning rates are well separated for all T (and K) for optimized ABP, this is not the case for small T for GD, where one finds a power-law decay of $\Delta\eta_{\max}^{\text{opt}}$ with an exponent that approaches $2/3$ for $TK \ll 1$ from above, making an optimal selection of the learning rate increasingly more difficult.

Fixed points: Finally, we would like to compare the symmetric fixed point for the optimized dynamics for finite learning rate with the theoretical values (6.8) for the truncated equations. Instead of illustrating the behaviour graphically, we have summarized the results in Table 6.1. We have found it most illuminating to compare the normalized difference $\hat{P}^* = (P^* - P_0^*)/P_0^*$ for all relevant order parameters (note that the identity $R^* = S^*$ is preserved for finite η) in the various limits. In general, one finds for both algorithms that $Q^* > Q_0^*$ and $R^* > R_0^*$. For C^*, however, one finds a T-dependent behaviour with $C^* < C_0^*$ for $T < T_s^{\text{crit}}(K)$ and $C^* > C_0^*$ for $T > T_s^{\text{crit}}(K)$, where $T_s^{\text{crit}} \propto K^{\frac{1}{3}}$ for GD and $T_s^{\text{crit}} \propto K^{\frac{1}{2}}$ for ABP. We furthermore find that the optimal symmetric fixed point for ABP is always significantly closer to the zero learning rate fixed point than for GD.

	$T \to 0$ $(TK \ll 1)$		$K \to \infty$ $(TK \gg 1)$	
	$\mathcal{A}^{\mathrm{GD}}$	$\mathcal{A}^{\mathrm{ABP}}$	$\mathcal{A}^{\mathrm{GD}}$	$\mathcal{A}^{\mathrm{ABP}}$
$\hat{Q}^*$	$c(K)$	$T^{0.33\pm3}$	$K^{0.64\pm2}$	$K^{0.48\pm2}$
$\hat{C}^*$	$-c(K)$	$-T^{0.33\pm3}$	$K^{-0.33\pm2}$	$K^{-0.50\pm1}$
$\hat{R}^*$	$T^{1.00\pm1}$	$T^{1.33\pm1}$	$K^{-0.35\pm2}$	$K^{-0.50\pm1}$
T_s^{crit}			$K^{0.31\pm2}$	$K^{0.50\pm1}$

Table 6.1. Symmetric fixed points of the optimized dynamics for both the gradient descent $\mathcal{A}^{\mathrm{GD}}$ and adaptive back-propagation $\mathcal{A}^{\mathrm{ABP}}$ are compared in the limits $T \to 0$ and $K \to \infty$ to the theoretical values for $\eta = 0$ by calculating their normalized difference $\hat{P}^* = (P^* - P_0^*)/P_0^*$. These differences exhibit either power-law behaviour, with algorithm-dependent exponents or saturate at constant limits, whose absolute value may be parameter-dependent and are referred to by $c(\cdot)$. In the limit $T \to \infty$ all parameters exhibit finite limits and are therefore omitted. $T_s^{\mathrm{crit}}(K)$ is defined by $C^* = C_0^*$.

A brief summary of the symmetric phase results

Before we turn our attention to the optimization of the dynamics in the convergence phase, we would like to summarize the results obtained so far and put them in the context of previous work. Unlike the small learning rate regime, which has been studied previously for GD (Saad and Solla 1995b), we find that the amount of training time spent in the symmetric phase actually scales worse than K^2 for the optimal choice of learning parameters (see Table 6.2 for an overview of the numerical values of the power laws). This seems to be mainly due to the need of reducing the learning rate η with increasing K. This reduction is arguably caused by the high correlations between student nodes inside and the (mainly uncorrelated) increase of the student lengths Q^* outside the space spanned by the teacher vectors, leading to a discrepency between student and teacher output that increases significantly faster than K for large enough T. For $K \to \infty$ $(TK \gg 0)$, one also finds that the gain, by using the optimal ABP choice of $\beta^{\mathrm{opt}} \propto \sqrt{K}$, is only a factor $K^{\frac{1}{6}}$ and not K as predicted previously.

We have furthermore relaxed the constraint $T = 1$ used in (Saad and Solla 1995b) and have found that the optimal learning parameter values change significantly in the most relevant region of teacher lengths, which makes it difficult in practice to choose optimal learning parameters without prior knowledge or estimation of the teacher lengths. For small T, which corresponds to nearly linear (but bounded) rules, one finds that the

	$T \to 0$ $(TK \ll 1)$		$K \to \infty$ $(TK \gg 1)$	
	$\mathcal{A}^{\mathrm{GD}}$	$\mathcal{A}^{\mathrm{ABP}}$	$\mathcal{A}^{\mathrm{GD}}$	$\mathcal{A}^{\mathrm{ABP}}$
β^{opt}	1	$T^{-1.00\pm1}$	1	$K^{0.50\pm2}$
η^{opt}	π	$c(K)$	$K^{-0.67\pm3}$	$K^{-1.00\pm1}$
$\Delta\eta^{\mathrm{opt}}_{\mathrm{max}}$	$T^{0.68\pm3}$	$c(K)$	$c(T)$	$c(T)$
λ^{opt}	$T^{2.00\pm1}K^{-2}$	$T^{1.00\pm1}K^{-2}$	$K^{-2.66\pm4}$	$K^{-2.50\pm1}$

Table 6.2. For $T \to 0$ and $K \to \infty$ the optimized dynamics in the symmetric phase show power-law behaviour for both the gradient descent $\mathcal{A}^{\mathrm{GD}}$ and adaptive back-propagation $\mathcal{A}^{\mathrm{ABP}}$. The table shows the optimal learning parameters η^{opt} and β, the optimal escape eigenvalue λ^{opt}, and the normalized difference between maximal and optimal learning rate $\Delta\eta^{\mathrm{opt}}_{\mathrm{max}} = (\eta_{\mathrm{max}} - \eta^{\mathrm{opt}})/\eta^{\mathrm{opt}}$. The errors in the exponent are given for the last significant digit only and $c(\cdot)$ refers to constant limits, whose value is dependent on a parameter.

specialization process is furthermore slowed down by a factor of $1/T^2$ for GD learning. This is arguably due to the fact that the symmetric fixed point is already a very good approximation to the true function and information about the nonlinearities is scarce. In this regime the optimal choice of $\beta^{\mathrm{opt}} \propto 1/T$ helps the student significantly in breaking the symmetry by reducing the region of hidden unit activation relevant for training and favouring rotational over longitudinal changes. The gain achievable in this regime is of order $1/T$.

6.4.2 Convergence to optimal generalization

In Section 5.5, the optimal convergence dynamics were already described for the case of GD and dynamic biases. Here, we will push the analysis for GD further in the space of T and K. Furthermore, besides predicting the optimal (weight) learning rate η^{opt}, we also study the influence of the inverse temperature β and identify its optimal value β^{opt}. As before, we linearize the reduced set of equations of motion (6.B.2) in $\{R, Q, C, S\}$ around the zero generalization error fixed point $R^* = Q^* = T$ and $S^* = C^* = 0$ (see Appendix 6.D).

The eigenvalue spectrum

The matrix $\mathbf{M}$ of the resulting system of four coupled linear differential equations in $r = T - R$, $q = T - Q$, $s = S$, and $c = C$ has two pairs of eigenvalues ($\lambda_{1,2}$ and $\lambda_{3,4}$)

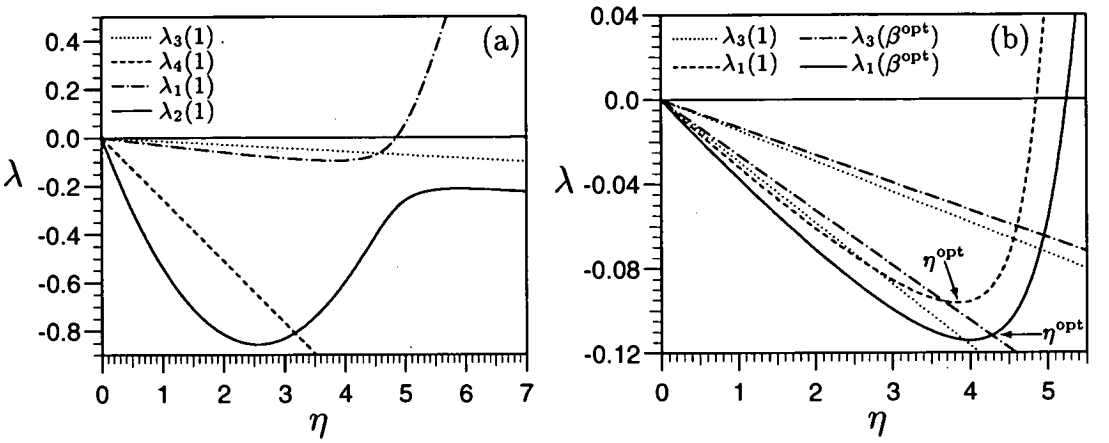

Figure 6.6. (a) The four eigenvalues λ_i for GD ($\beta = 1$) as a function of the learning rate η at $K = 3$ and $T = 1$. (b) The two relevant eigenvalues (see the text) λ_1 and λ_3 in the same scenario are shown for two values of β: $\beta = 1$ and $\beta = \beta^{opt} = 1.8314$. For comparison we plot $2\lambda_3$ and find that the optimal learning rate η^{opt} is given by the condition $\lambda_1 = 2\lambda_3$ for β^{opt}, but by the minimum of λ_1 for $\beta = 1$.

that are solutions of quadratic equations (6.D.4). The solutions for $\lambda_{1,2}$ reduce to the ones found in Section 5.5.1 for $\beta = 1$ (GD) since these eigenvalues are independent of the bias dynamics, whereas $\lambda_{3,4}$ have been modified for GD by the exclusion of the bias dynamics.

The dependence of these eigenvalues on the learning rate is illustrated in Figure 6.6(a) for $K = 3$ and $T = 1$. The eigenvalues $\lambda_{3,4}$ are linear in η, whereas $\lambda_{1,2}$ have higher orders in η as before. Again, one can distinguish further between two slow modes associated with eigenvalues λ_1 and λ_3 and two faster modes associated with eigenvalues λ_2 and λ_4, which are negative for all learning rates and whose magnitude is significantly larger in the relevant η region. As previously, the fast modes can be neglected therefore for the long-time dynamics, as they decay quickly. The dependence of the two relevant eigenvalues λ_1 and λ_3 on η and β is more closely illustrated in Figure 6.6(b) in the same learning scenario and for two β values. As described above, the eigenvalue λ_3 is negative and linear in η, whereas the eigenvalue λ_1 is a nonlinear function of η and negative for small η. For large η, λ_1 becomes positive and training cannot converge to the optimal solution consequently defining the maximum learning rate η_{max} as $\lambda_1(\eta_{max}) = 0$. For smaller learning rates ($\eta < \eta_{max}$) the generalization error decays exponentionally to $\epsilon_g^* = 0$.

The optimal dynamics

Although, the settings of the optimal dynamics was already described in Section 5.5.2, we will briefly reiterate the findings due to the introduction of the inverse temperature β. The generalization error is expanded to second in r, q, s, and c [Eq. (6.D.1)], in order to identify the optimal convergence eigenvalue λ^{opt}, which is the eigenvalue associated with the slowest decay mode. For all β, we find that the eigenvector (6.D.5) associated with the linear eigenvalue λ_3 is orthogonal to the first-order terms in the generalization error and therefore cannot contribute to their decay, but controls only the decay of second-order terms with $2\lambda_3$ as for GD. The learning rate η^{opt} that provides the fastest asymptotic decay rate λ^{opt} of the generalization error is therefore as before given by the condition

$$\lambda^{\text{opt}} = \left| \min_{\eta} \left[\max \left(\lambda_1, 2\lambda_3 \right) \right] \right| . \tag{6.12}$$

This means either $\lambda_1(\eta_{\text{r}}^{\text{opt}}) = 2\lambda_3(\eta_{\text{r}}^{\text{opt}})$ or $\min_\eta(\lambda_1)$ if $\lambda_1(\eta_{\text{m}}^{\text{opt}}) > 2\lambda_3(\eta_{\text{m}}^{\text{opt}})$, where $\eta_{\text{m}}^{\text{opt}}$ is the learning rate at the minimum of λ_1. Here, the existence of either solution type is strongly dependent on β as can be seen in Figure 6.6(b), where the minimum of λ_1 defines the optimal decay rate for GD but the root of $\lambda_1 - 2\lambda_3$ optimizes the dynamics for ABP with $\beta = \beta^{\text{opt}} = 1.8314$.

In general (given K), one finds that for GD ($\beta = 1$) the optimal learning rate is at the minimum of λ_1 for $T < T_c^{\text{crit}}(K)$ and by $\lambda_1 = 2\lambda_3$ otherwise, where $T_c^{\text{crit}}(K)$ is

Table 6.3. For $T \to 0$ and $T \to \infty$ the optimized dynamics in the convergence phase show power-law behaviour for both gradient descent $\mathcal{A}^{\text{GD}}$ and adaptive back-propagation $\mathcal{A}^{\text{ABP}}$. The table shows the optimal learning parameters η^{opt} and β^{opt}, the optimal convergence eigenvalue λ^{opt}, and the normalized difference between maximal and optimal learning rate $\Delta\eta_{\text{max}}^{\text{opt}} = (\eta_{\text{max}} - \eta^{\text{opt}})/\eta^{\text{opt}}$. $c(\cdot)$ refers to constant limits, whose value is dependent on a parameter.

	$T \to 0$		$T \to \infty$ (K finite)		$T \to \infty$ [$TK^{-1} = \mathcal{O}(1)$]	
	$\mathcal{A}^{\text{GD}}$	$\mathcal{A}^{\text{ABP}}$	$\mathcal{A}^{\text{GD}}$	$\mathcal{A}^{\text{ABP}}$	$\mathcal{A}^{\text{GD}}$	$\mathcal{A}^{\text{ABP}}$
β^{opt}	1	T^{-1}	1	$T^{-\frac{1}{3}}$	1	$\frac{1}{3}$
η^{opt}	π	$c(K)$	K^1	K^1	$T^{\frac{1}{2}}$	$T^{\frac{1}{2}}$
$\Delta\eta_{\text{max}}^{\text{opt}}$	T^1	$T^{\frac{1}{2}}$	T^{-1}	T^{-1}	T^{-1}	T^{-1}
λ^{opt}	T^2K^{-1}	T^1K^{-1}	$T^{-\frac{3}{2}}$	$T^{-\frac{3}{2}}$	$T^{-1}K^{-1}$	$T^{-1}K^{-1}$

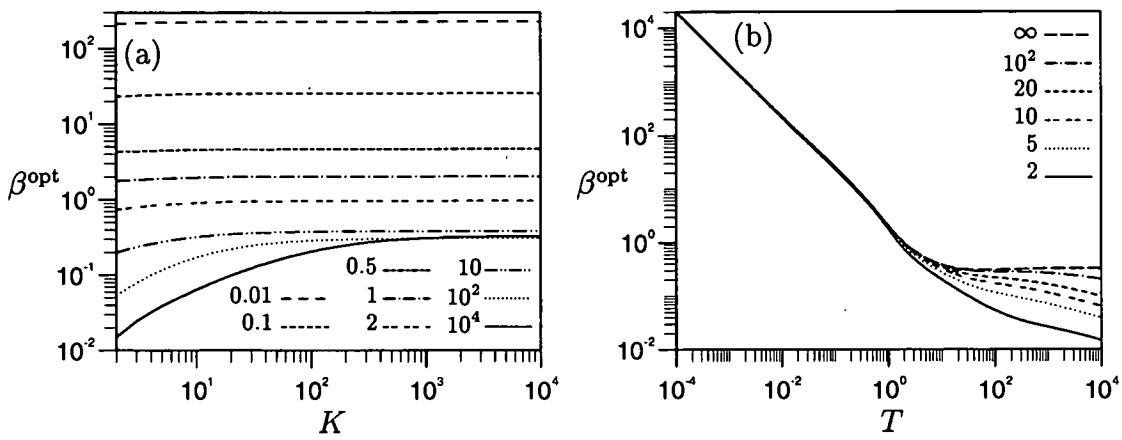

Figure 6.7. (a) The optimal inverse temperature $\beta^{\rm opt}$ is shown for various T values (see the legend) as a function of K. It exhibits only a significant K dependence for large T. (b) $\beta^{\rm opt}$ is shown as a function of T for various K values (see the legend), including the dominant term for $K \to \infty$. For small T, we find a power-law increase of $\beta^{\rm opt}$ with $1/T$ independent of K. For large T, the behaviour of $\beta^{\rm opt}$ strongly depends on K.

a function weakly dependent on K and $T_c^{\rm crit}(\infty) = 1.2780$ [see also Figure 6.8(c)] as already observed in Section 5.5.2. For optimized ABP, however, where the decay rate $\lambda^{\rm opt}(\beta)$ has been maximized with respect to β, the optimal learning rate is given by the root of $\lambda_1 - 2\lambda_3$ for all values of T.

Since both these optimizations are analytically infeasible for arbitrary K and T, we again study some special cases further where analytical progress can be made: $K \to \infty$, $T \to \infty$, and $T \to 0$. These limits are studied in detail in Appendices 6.D.1–6.D.5. The resulting power laws will be referred to in the discussion of the appropriate figures and are summarized for all relevant scenarios in Table 6.3. In order not to interrupt the flow of the argument, we will refrain below from referring back to the results in Chapter 5, since the main objective here is to compare the differences between the two algorithms.

Optimal inverse temperature $\beta^{\rm opt}$

As in the symmetric phase, one expects the largest gains by using ABP in regions of T-K space, where $\beta^{\rm opt}$ deviates significantly from 1. In Figure 6.7 the optimal value of β is shown as a function of both K and T. Figure 6.7(a) shows that $\beta^{\rm opt}$ is only a weak function of K and does not change its order for $K \to \infty$ unlike in the symmetric phase. The only significant K dependence is found for large T and small K.

This should be contrasted to the strong T dependence of $\beta^{\rm opt}$ depicted in Figure 6.7(b), where the theoretical results for $K \to \infty$ are included as well. For small T one finds to leading order $\beta^{\rm opt} = 2/T$, independent of K, whereas a strong dependence of K on $\beta^{\rm opt}$ is found for large T. For finite K or $T/K \gg 1$, one finds $\beta^{\rm opt} \propto T^{-\frac{1}{3}}$, whereas $\beta^{\rm opt} \approx 1/3$ for $T/K \leq \mathcal{O}(1)$. The qualitative difference of learning for finite and infinite K in the large-T limit will become clear later.

Optimal learning and convergence rates

Again, we would like to assess the potential benefits of ABP over GD. Note the discrepency between our results and those previously presented (Saad and Solla 1995b) for GD in the convergence phase for the special case $T = 1$, where an approximation by reducing the dynamics to the q-r space was employed, producing inaccurate results.

In Figure 6.8 we therefore show the behaviour of both the optimal learning rate $\eta^{\rm opt}$ [Figures 6.8(a,b)] and the resulting optimal convergence eigenvalue $\lambda^{\rm opt}$ [Figures 6.8(c,d)] for GD in comparison to ABP as a function of T for several values of K, including the dominant term for $K \to \infty$. The optimal learning rate $\eta^{\rm opt}(T)$ of GD depicted in Figure 6.8(a) has a universal limit of π for small T identical to the symmetric phase. For large T the limit becomes strongly dependent on K. Again, there exists a qualitative difference between finite K, where one finds analytically $\eta^{\rm opt} \propto K$ for $T \to \infty$ and infinite K where $\eta^{\rm opt} \propto \sqrt{T}$.

The quotient between the optimal learning rates of ABP and GD in Figure 6.8(b) shows no significant difference in stark contrast to results in the symmetric phase. In general, one finds that the learning rate for ABP is larger than for GD when $\beta^{\rm opt} > 1$ and vice versa. For small T the optimal learning rate approaches $\sqrt{3}\,\pi$ for infinite K [Eq. (6.D.13c)] with minor corrections for finite K [Eq. (6.D.17c)]. For large T, the difference is a factor of $1/\sqrt{2}$ for infinite K, whereas they are identical for finite K.

The kink in the curves around $T \approx 1$ can be explained by the fact that the condition that defines $\eta^{\rm opt}$ for GD changes at that point (see above). The corresponding critical teacher value $T^{\rm crit}(K)$ is shown in Figure 6.9.

The optimized convergence eigenvalue, which largely determines the training time spent achieving an acceptable generalization error, is shown for GD in Figure 6.8(c), where we have multiplied $\lambda^{\rm opt}$ by K for convenience. For small T, one finds that $\lambda^{\rm opt}$ collapses on a univeral curve ($\lambda^{\rm opt} \propto T^2/K$), similar to its symmetric phase behaviour. For large T, the behaviour for $\lambda^{\rm opt}$ depends significantly on the order of K as that of the learning rate. Analytically, one finds for K finite and $TK \gg 1$ that $\lambda^{\rm opt}$ is actually independent of K and decreases proportionally to $T^{\frac{3}{2}}$. For large T and $T/K = \mathcal{O}(1)$, on the other hand, the scaling is $\lambda^{\rm opt} \propto 1/(TK)$.

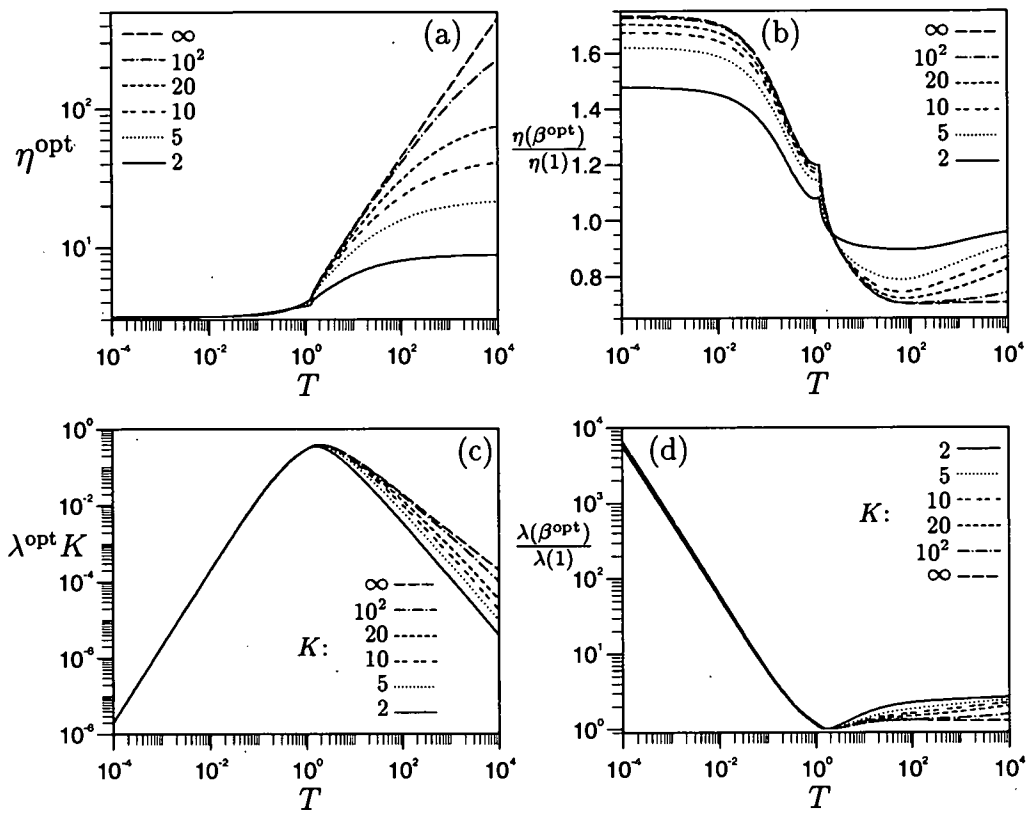

Figure 6.8. Optimal learning parameters in the convergence phase as a function of T for various K values (see the legends). (a) The optimal learning rate η^{opt} for GD shows a significant increase for large T and K. (b) The quotient of the optimal learning rates of ABP and GD shows no significant difference in the optimal learning rates of the two algorithms. (c) The optimal convergence rate for GD multiplied by K collapses on a universal (K-independent) curve for small T and decays rapidly with exponent 2 as in the symmetric phase. For large T the convergence rate also decays in T, but with an exponent that seems K-dependent. (d) The possible gain by using ABP is shown by plotting the quotient of the optimal convergence eigenvalue for the two training algorithms. The advantage of ABP is most impressive for small T, where one can gain a K independent factor $1/T$ in comparison to GD. For large T the gain is K-dependent but constant in T.

To highlight the possible gains from using ABP, $\lambda^{opt}(\beta^{opt})/\lambda^{opt}(1)$ is plotted as a function of T in Figure 6.8(d). For small T, one finds as in the symmetric phase that ABP gains a factor $1/T$, with only a very weak K dependence due to corrections in the $1/K$ dependence for ABP. For large T, one finds only a constant gain for ABP, which ranges between 1.299 and 2.828 depending on the values of T and K, although

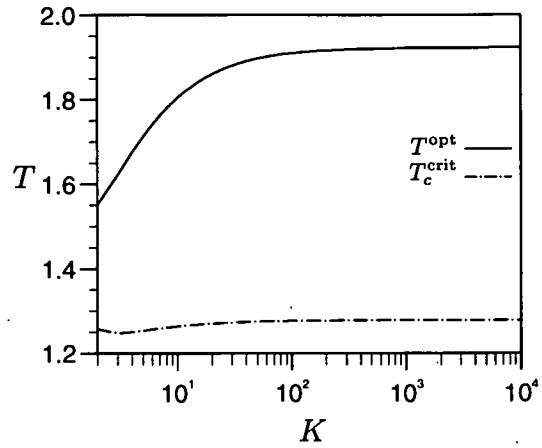

Figure 6.9. The teacher length $T_c^{\text{crit}}(K)$, where the optimal learning rate changes from the minimum of λ_1 to the root of $\lambda_1 - 2\lambda_3$, and the teacher length $T^{\text{opt}}(K)$, where the convergence rate λ takes its global minimum. The latter coincides with $\beta^{\text{opt}} = 1$ for all K. (f) The normalized difference between the maximal and optimal learning rate $\Delta\eta_{\text{max}}^{\text{opt}}$ is shown for both adaptive back-propagation $\mathcal{A}^{\text{ABP}}$ and gradient descent $\mathcal{A}^{\text{GD}}$ for $K = 5, \infty$ as a function of T. For both small and large T one finds power-law behaviour.

β^{opt} deviates significantly from 1 for finite K.

Other issues: Optimal/critical teacher lengths and maximal learning rates

A question one could ask is which teacher length T^{opt} maximized λ^{opt} for given K. This turns out to be identical for both algorithms [$\beta^{\text{opt}}(T^{\text{opt}}) = 1$] and its dependence on K is shown in Figure 6.9. Although only of academic interest as T is given by the rule to be learned, it nevertheless presents some interesting insights. ABP effectively deforms the search space via the single parameter β to compensate for the anisotropy of the generalization error surface. At T^{opt} no useful deformation can be obtained by using $\beta \neq 1$, leaving room for speculation whether isotropy is recovered. Other methods for deforming the search space based on information geometry have been introduced recently and involve more complicated learning rules, which may not always be tractable (Amari 1997b; Amari 1997a).

In Figure 6.10, the normalized separation between the maximal and optimal learning rate shows for both algorithms only a very weak dependence on K in comparison to T. The gap is largest for $T = \mathcal{O}(1)$, the region of most likely T values, with a maximal separation around 30% for both algorithms, which is significantly smaller than the separation in the symmetric phase. For both large and small T, we find decays of the normalized gap in T. For large T, the decay is proportional to $1/T$ for both algorithm,

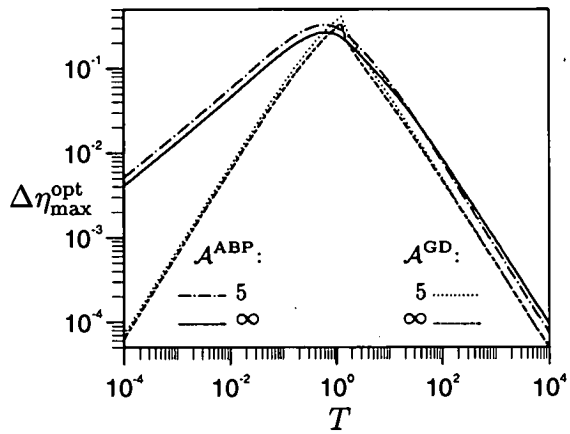

Figure 6.10. The normalized difference between the maximal and optimal learning rate $\Delta\eta_{\max}^{\mathrm{opt}}$ is shown for both adaptive back-propagation $\mathcal{A}^{\mathrm{ABP}}$ and gradient descent $\mathcal{A}^{\mathrm{GD}}$ for $K = 5, \infty$ as a function of T. For both small and large T one finds power-law behaviour.

with slight differences in the constant prefactor. For small T, however, the behaviour is algorithm dependent, with a decay proportional to T for GD proportional to $\sqrt{T}$ for ABP.

A brief summary of the convergence phase results

As in the symmetric phase, the extension of the analysis to the full *R-Q-S-C* space and arbitrary T values has revealed several insights. The normalization for the soft-committee machine chosen here leads to the optimal learning rate for both algorithms (and the optimal inverse temperature for ABP) being only weakly dependent on K in most practical learning scenarios, suggesting a similar scaling for applied networks. For large K one finds furthermore that the training time scales with K in almost all cases, in contrast to the symmetric phase, reflecting the fact that the student hidden units have already specialized on a particular teacher hidden unit.

For extreme values of T, one finds further interesting effects. For small T, GD training is slowed down by a further factor of $1/T^2$, which can be reduced to a factor of $1/T$ by the optimal choice of $\beta^{\mathrm{opt}} \propto 1/T$, similar to the symmetric phase.

For large T, one has to distinguish between two regimes. For finite K, both the mapping of the network and the error signal become increasingly discrete in this limit, leading to an architecture similar to a committee machine. In this case, the error signal is of $\mathcal{O}(1/K)$ leading to a rescaling of the learning rate with K, in order to keep the weight update constant for all network sizes, making the convergence rate independent

of K. The increasingly discrete nature of the error signal, however, seems responsible for the decrease in the convergence rate by $T^{-\frac{3}{2}}$ for both algorithms. The possible gain of ABP stays constant in this limit, in spite of the significant scaling of $\beta^{\rm opt} \propto T^{-\frac{1}{3}}$.

In the limit where K grows simultaneously with T, one finds a qualitatively different behaviour. This can be explained by the smoothness of the network output and the error signal in this case due to the fact that hidden units outputs are discrete but uncorrelated, giving rise to a Gaussian output distribution (central limit theorem).

6.5 Summary and discussion

The motivation for the research presented in this chapter has been initially provided by the dominance of the suboptimal symmetric phase in on-line learning of two-layer feed-forward networks trained by gradient descent studied in Chapter 5 in learning task with internal symmetries. We identified the possible reason for the short-comings of standard gradient descent and proposed an adaptive back-propagation training algorithm (6.5) parameterized by an inverse temperature β. For $\beta = 1$ standard back-propagation or GD is recovered, whereas $\beta = 0$ corresponds to a generalized Hebb rule. This algorithm has enabled us to confirm this intuition and to investigate possible improvements.

ABP is designed to deform search space using the single parameter β. For $\beta > 1$, the specialization of the student nodes is improved by enhancing differences in the activation between hidden units. In this region, the achievable learning rate is usually higher than for GD, leading effectively to favouring rotational changes of the weight vector over length changes. For $0 < \beta < 1$, we find the opposite effect, as the activation region of the student relevant for training is increased and the learning rate decreased, causing an enhancement of length changes.

Its performance has been compared to GD for the same architecture as in Chapter 5, a normalized soft-committee student network with K hidden units, but with zero-fixed biases in order to be able to push further the analysis of its performance in learning a rule defined by an isotropic teacher ($T_{nm} = T\delta_{nm}$) of the same architecture. The natural normalization of the soft-committee machine has again lead to more elegant results for the whole training process as it eliminates the unnatural scaling of the (weight) learning rate with the input dimension N and, in many cases, with the number of hidden units K, which is a feature of the unnormalized model, and combined with the results presented in Chapter 5 suggests a similar approach for real world networks.

In comparison to Chapter 5, we have not only been able to extend the analysis to both relevant phases of learning, the symmetric and convergence phase. This work extends furthermore previous results for the unnormalized model without biases (Saad

and Solla 1995b) substantially by addressing the influence of finite learning rates in the symmetric phase and the influence of the teacher length T on the dynamics. Although our analysis was only a linear analysis around the fixed points of the dynamics, the results for GD have been confirmed to hold extremely well for the whole lerning process by studies of global optimal learning rates within a variational framework [Saad and Rattray (1997a, 1997b); Rattray and Saad (1997a)]. The analysis identifies three interesting regimes: large K, small T, and large T.

6.5.1 Large K

For large K, the linear analysis of the equations of motion around the symmetric fixed point for small learning rates suggests that the trapping time is inversely proportional to the learning rate and grows $\tau \propto K^2$ for GD[7] and $\tau \propto K$ for optimized ABP with $\beta^{\text{opt}} \propto K$. This suggests that for increasing network size it seems to become harder for a student node to distinguish between the many teacher nodes and to specialize on one of them. This is reflected by the decrease in the squared student length $Q^* \propto 1/K$ at the symmetric fixed point, pushing the student hidden nodes into the linear region of the sigmoidal activation function, where differentiation is more difficult.

This picture is altered significantly when accounting for finite learning rate effects, due to the decrease in the optimal learning rate η^{opt} with K, beyond the rescaling implicit in the network normalization. This rescaling assumes an unnormalized network output of $\mathcal{O}(\sqrt{K})$ and a typical squared error of $\mathcal{O}(K)$, which is appropriate in the case when the hidden units of both the student and the teacher network are uncorrelated. However, in the symmetric phase this is not the case for the student network leading to errors that grow faster than $\mathcal{O}(K)$ and making a decrease in the learning rate necessary. The significant reduction of the learning rate may also be associated with the need to limit the proportion of the student length outside the space spanned by the teacher for large K.

The actual training time spent in the symmetric phase therefore scales $\tau \propto K^{\frac{8}{3}}$ for GD and $\tau \propto K^{\frac{5}{2}}$ for ABP, reducing the benefit of an adjustable temperature to $K^{\frac{1}{6}}$. One also finds that the scaling for the optimal temperature changes to $\beta^{\text{opt}} \propto \sqrt{K}$ in this limit.

For the convergence phase one finds that the training time scales with K in almost all cases, reflecting the fact that the learning rate must (implicitly) be rescaled by $1/K$

[7] This result only seems to differ from the result in (Saad and Solla 1995b) due to different scaling for η.

as the typical quadratic deviation between teacher and student output increases proportionally to K. The optimal inverse temperature and the optimal gain of using ABP in this regime are dependent on T but remain constant for large K due to the fact that each student hidden unit is already specialized on one teacher unit and the effect of other units in inhibiting further specialization is negligible.

These results mean that most of the training time is spent in the symmetric phase (or search regime) for large networks, at least in learning scenarios with a certain amount of symmetry. This suggests that considerably more effort should be directed towards developing algorithms, which can significantly reduce the training time in this phase, than towards fine tuning of the asymptotic convergence.

6.5.2 Small T

In the small T-limit, one finds very similar results for both the symmetric and the convergence phases, e.g., the optimal learning rate is universally π for GD, the optimal inverse temperature has the same scaling behaviour ($\beta^{\text{opt}} \propto 1/T$), and the optimal escape and the optimal convergence eigenvalue scale with T^2 for GD and with T for ABP in both learning phases. This results in a gain of order $1/T$, in using ABP, for the whole training process.

The universal slowdown of learning in the small-T limit may be explained by the fact that the learning rule becomes increasingly linear, resulting in a very flat (generalization) error surface between the symmetric and the zero-generalization error fixed point. The major difference is the scaling of the relevant eigenvalue with the number of hidden units K, reflecting the lesser degree of confusion once the hidden unit symmetry is broken.

6.5.3 Large T

For large T the picture is not as coherent, which can be explained by the increasingly binary nature of the hidden unit outputs. In the symmetric phase, the outputs of the student hidden units are highly correlated, whereas the outputs of the teacher hidden units are uncorrelated, leading to large errors between the student and teacher network output that scale with K but saturate for large T, explaining the large changes in the optimal learning parameters for medium T but also their indifference to further increases in T once T is sufficiently large.

In the convergence phase, a significantly different behaviour is observed for the two cases of finite K and infinite K, where the network output is discrete and continuous, respectively. For infinite K, the error remains smooth and actually decreases for large

T due to the increasingly binary hidden unit output, giving rise to an increase of $\eta^{\text{opt}} \propto T^{\frac{1}{2}}$. For finite K, one typically finds that at most one student hidden unit "misclassifies" the output of the corresponding hidden unit of the teacher, causing a discrete error of either 0 or $1/K$ and leading to a rescaling of the learning rate proportional to K.

It would be quite interesting to study this limit more closely due to its similarity to the committee machine. The possibility of tuning the weight function with β between a Hebb-like form for $\beta = 0$ and a Gaussian form for finite β may give some idea about successful training algorithms for binary networks.

However, throughout our analyses we have implicitely assumed that the decay or increase in the exponential terms outstrips any algebraic variation in the prefactors and all optimizations were carried out under this assumption. This is reasonable at least for medium values of T, which are most likely to be encountered practically, but probably also for any finite values of T. For infinite T, i.e., networks with discrete hidden units, this ansatz is, however, insufficient as the exponential term vanishes and the dynamics become algebraic in α.

In principle, one could encompass these limiting cases by incorporating second-order terms of the Taylor series around the fixed points and solving the resulting set of nonlinear differential equations by transforming them into matrix Riccati equations. Although this is in principle feasible, it goes beyond the scope of which has been achievable in this thesis.

6.5.4 Conclusions

This chapter has shown some of the learning performance limitations of gradient descent in the on-line learning paradigm. Within the model with fixed biases studied, one finds severe drawbacks of GD, especially in the symmetric phase, which dominates the learning process for large networks and symmetries in the task. It remains to be seen how common such types of tasks are, however, one could argue that for larger input dimensions symmetry in tasks may not be uncommon, since student hidden units may have to be located at similar bias values to model output variances in different directions of input space.

The suggested adaptive back-propagation algorithm generally speeds up the training process considerably if its extra parameter, the inverse temperature β, is chosen optimally. It has provided us also with some insight into the shortcomings of GD and has outlined possible further research directions.

The relaxation of the constraint $T = 1$ has shown that the optimal learning parameter values change significantly in the region of usually relevant teacher lengths and

between the symmetric and the convergence phase, confirming and extending results for the model with dynamic biases studied in Chapter 5. The choice of good learning parameters (in this case the learning rate η and the inverse temperature β) remains a paramount problem in practice without prior knowledge or estimation of the teacher lengths and the progress the student has made in learning. This should encourage more research into reliable on-line estimation of optimal learning parameters. It further suggests that the selection of individual learning parameters for each hidden node of the network may be potentially hugely beneficial (Saad and Rattray 1997a), although the ultimate goal remains the developement of on-line learning algorithms which can be considered globally "optimal," as will be discussed in more detail in Chapter 7.

Appendix 6

6.A Dynamical equations

The generalization error and the dynamical equations are calculated along the lines of Appendix 5.A by averaging over the covariance matrix $\mathcal{C}$ (5.A.1). The form of the generalization error and the dynamical equations are identical but with integrals which are analytically solvable and dependent on β. In order to avoid extensive cross-referencing we repeat their form briefly[8]. The generalization error takes the form

$$\epsilon_g = \frac{\gamma^2}{2K}\left\{\frac{K}{M}\sum_{n,m=1}^{M} J_2(n,m) - 2\sqrt{\frac{K}{M}}\sum_{i,n=1}^{K,M} J_2(i,n) + \sum_{i,j=1}^{K} J_2(i,j)\right\}, \qquad (6.A.1)$$

The differential equations for $\mathbf{R}$ and $\mathbf{Q}$ are

$$\frac{dQ_{ij}}{d\alpha} = \frac{\eta\gamma^2}{K}\left\{\sqrt{\frac{K}{M}}\sum_{m=1}^{M} I_3(i,j,m) + I_3(j,i,m) - \sum_{k=1}^{K} I_3(i,j,k) + I_3(j,i,k)\right\} \qquad (6.A.2a)$$

$$+\left(\frac{\eta\gamma^2}{K}\right)^2\left\{\frac{K}{M}\sum_{n,m=1}^{M} J_4(i,j,n,m) - 2\sqrt{\frac{K}{M}}\sum_{k,n=1}^{K,M} J_4(i,j,k,n) + \sum_{k,l=1}^{K} J_4(i,j,k,l)\right\},$$

$$\frac{dR_{in}}{d\alpha} = \frac{\eta\gamma^2}{K}\left\{\sqrt{\frac{K}{M}}\sum_{m=1}^{M} I_3(i,n,m) - \sum_{k=1}^{K} I_3(i,n,k)\right\}, \qquad (6.A.2b)$$

[8]We again make use of the convention that indices i, j, k, l and n, m label student and teacher nodes, respectively.

with the same integrals[9] $J_2(1,2) = \langle g(u_1)g(u_2)\rangle$, $I_3(1,2,3) = \langle g'(u_1)u_2 g(u_3)\rangle$ and $J_4(1,2,3,4) = \langle g'(u_1)g'(u_2)g(u_3)g(u_4)\rangle$. In this case these integrals can be evaluated analytically for $g_\nu(u) = \mathrm{erf}(\nu u/\sqrt{2})$ by introducing an integral representation for g (A.2) using (A.4) and if necessary applying (A.5a)

$$J_2(1,2) = \frac{2}{\pi}\arcsin\left(\frac{\tilde{C}_{12}}{\sqrt{1+\tilde{C}_{11}}\sqrt{1+\tilde{C}_{22}}}\right), \tag{6.A.3a}$$

$$I_3(1,2,3) = \frac{2}{\pi}\frac{\Psi_{12}(\beta)}{\sqrt{\Psi_{13}(1)}}\frac{\Gamma_{13}}{\psi_1(\beta)}, \tag{6.A.3b}$$

$$J_4(1,2,3,4) = \left(\frac{2}{\pi}\right)^2 \frac{\nu^2}{\sqrt{\Psi_{12}(\beta)}}\arcsin\left(\frac{\tilde{C}'_{34}}{\sqrt{1+\tilde{C}'_{33}}\sqrt{1+\tilde{C}'_{44}}},\right), \tag{6.A.3c}$$

where we have conveniently defined

$$\psi_i(\beta) = 1+\beta\tilde{C}_{ii}, \qquad \psi_{ij}(\beta) = \beta\tilde{C}_{ij}, \qquad \Psi_{ij}(\cdot) = \psi_i(\beta)\psi_j(\cdot) - \psi_{ij}(\beta)\psi_{ij}(\cdot),$$
$$\Gamma_{1i} = \frac{\psi_1(\beta)\tilde{C}_{2i} - \psi_{12}(\beta)\tilde{C}_{1i}}{\Psi_{12}(\beta)}, \qquad \Gamma_{2i} = \frac{\psi_2(\beta)\tilde{C}_{1i} - \psi_{12}(\beta)\tilde{C}_{2i}}{\Psi_{12}(\beta)},$$
$$\tilde{C}'_{ij} = \tilde{C}_{ij} - \beta\left[\tilde{C}_{1i}\Gamma_{2j} + \tilde{C}_{2i}\Gamma_{1j}\right],$$

with $(\cdot)$ representing either β or 1. As before, the rescaled covariance matrix elements take the form $\tilde{C}_{ij} = \nu^2\sigma^2 C_{ij}$ and the actual elements of a reduced covariance matrix are inferred by using the unit labelling convention and the appropriate dimensionality reduction[10]. The input variance σ^2 and the gains ν and γ can also be absorbed as in Eq. (5.A.6b), such that $\nu = \gamma = \sigma = 1$ can be set w.l.o.g..

6.B Reduced equations

Reducing the free parameters for $K = M$ and $T_{nm} = T\delta_{nm}$ with the ansatz (6.7) to just R, S, Q, and C simplifies the generalization error (6.A.1) to

$$\epsilon_g = \frac{1}{\pi}\left\{\arcsin\left(\frac{T}{1+T}\right) + (K-1)\arcsin\left(\frac{C}{1+Q}\right) + \arcsin\left(\frac{Q}{1+Q}\right)\right.$$
$$\left. -2\arcsin\left(\frac{R}{\sqrt{1+Q}\sqrt{1+T}}\right) - 2(K-1)\arcsin\left(\frac{S}{\sqrt{1+Q}\sqrt{1+T}}\right)\right\}. \tag{6.B.1}$$

[9] Again u_i represent members of $\{\boldsymbol{x}, \boldsymbol{y}\}$ and we denote with I_d, J_d averages over d variables with one and two g terms, respectively.

[10] For example, the relevant elements for $J_2(i,n)$ are $C_{11} = Q_{ii}$, $C_{12} = R_{in}$, and $C_{22} = T_{nn}$ and for $J_2(1,1)$ are $C_{11} = C_{12} = C_{22}$

The differential equations for R, S, Q, and C are determined from Eq. (6.A.2) similarly and take the form

$$\frac{dR}{d\alpha} = \frac{2}{\pi}\frac{\eta}{K}\frac{1}{\gamma_1}\left\{\frac{\mathcal{R}_0-\gamma_1}{\sqrt{\mathcal{R}_0}} - \frac{R}{\sqrt{\mathcal{Q}_0}} - (K-1)\left[\frac{\beta RS}{\sqrt{\mathcal{S}_0}} + \frac{S\gamma_1-\beta RC}{\sqrt{\mathcal{C}_0}}\right]\right\}, \tag{6.B.2a}$$

$$\frac{dS}{d\alpha} = \frac{2}{\pi}\frac{\eta}{K}\frac{1}{\gamma_1}\left\{\frac{\mathcal{S}_0-\gamma_1}{\sqrt{\mathcal{S}_0}} - \frac{R\gamma_1-\beta SC}{\sqrt{\mathcal{C}_0}} - \frac{\beta RS}{\sqrt{\mathcal{R}_0}} - \frac{S}{\sqrt{\mathcal{Q}_0}},\right.$$
$$\left. - (K-2)\left[\frac{\beta S^2}{\sqrt{\mathcal{S}_0}} + \frac{S\gamma_1}{\sqrt{\mathcal{C}_0}}\right]\right\} \tag{6.B.2b}$$

$$\frac{dQ}{d\alpha} = \frac{4}{\pi}\frac{\eta}{K}\frac{1}{\gamma_1}\left\{\frac{R}{\sqrt{\mathcal{R}_0}} - \frac{Q}{\sqrt{\mathcal{Q}_0}} + (K-1)\left[\frac{S}{\sqrt{\mathcal{S}_0}} - \frac{C}{\sqrt{\mathcal{C}_0}}\right]\right\}$$
$$+ \frac{4}{\pi^2}\frac{\eta^2}{K^2}\frac{1}{\gamma_2}\left\{\arcsin\left(\frac{\mathcal{R}_1-\gamma_2}{\mathcal{R}_1}\right) - 2\arcsin\left(\frac{R}{\sqrt{\mathcal{Q}_1\mathcal{R}_1}}\right) + \arcsin\left(\frac{Q}{\mathcal{Q}_1}\right)\right.$$
$$+ (K-1)\left[2\arcsin\left(\frac{C}{\sqrt{\mathcal{Q}_1\mathcal{C}_1}}\right) - 2\arcsin\left(\frac{S}{\sqrt{\mathcal{Q}_1\mathcal{S}_1}}\right) - 2\arcsin\left(\frac{2\beta RS}{\sqrt{\mathcal{R}_1\mathcal{S}_1}}\right)\right.$$
$$- 2\arcsin\left(\frac{R\gamma_2-2\beta SC}{\sqrt{\mathcal{S}_1\mathcal{C}_1}}\right) - 2\arcsin\left(\frac{S\gamma_2-2\beta RC}{\sqrt{\mathcal{R}_1\mathcal{C}_1}}\right) + \arcsin\left(\frac{\mathcal{C}_1-\gamma_2}{\mathcal{C}_1}\right)$$
$$\left.\left. + \arcsin\left(\frac{\mathcal{S}_1-\gamma_2}{\mathcal{S}_1}\right)\right] + (K-1)(K-2)\left[\arcsin\left(\frac{C(\gamma_2-2\beta C)}{\mathcal{C}_1}\right)\right.\right.$$
$$\left.\left. - 2\arcsin\left(\frac{S(\gamma_2-2\beta C)}{\sqrt{\mathcal{S}_1\mathcal{C}_1}}\right) - \arcsin\left(\frac{2\beta S^2}{\mathcal{S}_1}\right)\right]\right\}, \tag{6.B.2c}$$

$$\frac{dC}{d\alpha} = \frac{4}{\pi}\frac{\eta}{K}\frac{1}{\gamma_1}\left\{\frac{R\gamma_1-\beta SC}{\sqrt{\mathcal{S}_0}} - \frac{Q\gamma_1-\beta C^2}{\sqrt{\mathcal{C}_0}} + \frac{S\gamma_1-\beta RC}{\sqrt{\mathcal{R}_0}} - \frac{C}{\sqrt{\mathcal{Q}_0}}\right.$$
$$\left. + (K-2)\left[\frac{S\gamma_3}{\sqrt{\mathcal{S}_0}} + \frac{C\gamma_3}{\sqrt{\mathcal{C}_0}}\right]\right\} + \frac{4}{\pi^2}\frac{\eta^2}{K^2}\frac{1}{\sqrt{\gamma_3\gamma_4}}\left\{2\arcsin\left(\frac{\mathcal{Q}_2-\gamma_3\gamma_4}{\mathcal{Q}_2}\right)\right.$$
$$+ 2\arcsin\left(\frac{\mathcal{R}_2-\gamma_3\gamma_4}{\mathcal{R}_2}\right) + 2\arcsin\left(\frac{\beta^2(R^2+S^2)-2\gamma_1\beta RS}{\mathcal{R}_2}\right) + 2\arcsin\left(\frac{C}{\mathcal{Q}_2}\right)$$
$$- 4\arcsin\left(\frac{R\gamma_1-\beta SC}{\sqrt{\mathcal{Q}_2\mathcal{R}_2}}\right) - 4\arcsin\left(\frac{S\gamma_1-\beta RC}{\sqrt{\mathcal{Q}_2\mathcal{R}_2}}\right) + (K-2)\left[4\arcsin\left(\frac{C\sqrt{\gamma_3}}{\sqrt{\mathcal{Q}_2\mathcal{C}_2}}\right)\right.$$
$$- 4\arcsin\left(\frac{(S\gamma_1-\beta RC)\sqrt{\gamma_3}}{\sqrt{\mathcal{R}_2\mathcal{C}_2}}\right) - 4\arcsin\left(\frac{\beta S(S+R)\sqrt{\gamma_3}}{\sqrt{\mathcal{R}_2\mathcal{S}_2}}\right) - 4\arcsin\left(\frac{S\sqrt{\gamma_3}}{\sqrt{\mathcal{Q}_2\mathcal{S}_2}}\right)$$
$$\left. + \arcsin\left(\frac{\mathcal{C}_2-\gamma_4}{\mathcal{C}_2}\right) - 2\arcsin\left(\frac{R\gamma_4-2\beta SC}{\sqrt{\mathcal{S}_2\mathcal{C}_2}}\right) + \arcsin\left(\frac{\mathcal{S}_2-\gamma_4}{\mathcal{S}_2}\right)\right]$$
$$\left. + (K-2)(K-3)\left[\arcsin\left(\frac{C\gamma_3}{\mathcal{C}_2}\right) - 2\arcsin\left(\frac{S\gamma_3}{\sqrt{\mathcal{S}_2\mathcal{C}_2}}\right) - \arcsin\left(\frac{2\beta S^2}{\mathcal{S}_2}\right)\right]\right\}, \tag{6.B.2d}$$

where we have for convenience defined

$$\gamma_1 = 1+\beta Q, \qquad \gamma_2 = 1+2\beta Q,$$

$$\gamma_3 = 1+\beta(Q-C), \qquad \gamma_4 = 1+\beta(Q+C),$$

$$\mathcal{Q}_0 = \gamma_1 + Q, \qquad \mathcal{Q}_1 = \gamma_2 + Q, \qquad \mathcal{Q}_2 = \gamma_3\gamma_4 + Q\gamma_1 - \beta C^2,$$

$$\mathcal{C}_0 = (1+Q)\gamma_1 - \beta C^2, \qquad \mathcal{C}_1 = (1+Q)\gamma_2 - 2\beta C^2, \qquad \mathcal{C}_2 = (1+Q)\gamma_4 - 2\beta C^2,$$

$$\mathcal{S}_0 = (1+T)\gamma_1 - \beta S^2, \qquad \mathcal{S}_1 = (1+T)\gamma_2 - 2\beta S^2, \qquad \mathcal{S}_2 = (1+T)\gamma_4 - 2\beta S^2,$$

$$\mathcal{R}_0 = (1+T)\gamma_1 - \beta R^2, \qquad \mathcal{R}_1 = (1+T)\gamma_2 - 2\beta R^2,$$

$$\mathcal{R}_2 = (1+T)\gamma_3\gamma_4 - \beta\gamma_1(R^2+S^2) + 2\beta^2 RSC.$$

6.C Symmetric fixed-point dynamics

Following Appendix 5.B, For a linear theory of the dynamics around their fixed point, we expand the differential equations (6.B.2) in a Taylor series to first order following Appendix 5.B

$$\frac{dp_i}{d\alpha} = m^{\mathrm{g}}_{i0} + \sum_{j=1}^{4} m^{\mathrm{g}}_{ij} p_j,$$

where $p_i = P_i - P_i^*$ and P_i are generic order parameters and m^{g} is a generic matrix element for either symmetric or convergence phase[11]. For a fixed point the zeroth-order terms vanish and the eigenvalues and eigenvectors of the Jacobian matrix $\mathbf{M}^{\mathrm{g}}$ of first derivatives determine the solution of the linearized differential equation.

Under the constraints $Q = C$ and $R = S$, which are characteristic for the symmetric fixed points studied analytically, one finds that the zeroth-order terms and the entries of the Jacobian matrix $\mathbf{M}^{\mathrm{s}}$ obey the relations (here $P_1 = R$, $P_2 = S$, $P_3 = Q$, and

[11] We will use the convention to superscript $\mathbf{M}$ and m by "s" and "c" for the symmetric and convergence phases, respectively.

$P_4 = C$)

$$\begin{aligned} m^s_{22} &= m^s_{11} + (K-2)m^s_{21}, & m^s_{10} &= m^s_{20}, & m^s_{30} &= m^s_{40}, \\ m^s_{32} &= m^s_{42} = (K-1)m^s_{31}, & m^s_{12} &= (K-1)m^s_{21}, & m^s_{23} &= m^s_{13}, \\ m^s_{44} &= m^s_{33} + m^s_{34} - m^s_{43}, & m^s_{41} &= m^s_{31}, & m^s_{24} &= m^s_{14}. \end{aligned} \tag{6.C.1}$$

We omit the exact form of the remaining free parameters of the matrix as they are extremely tedious but easily derivable from (6.B.2). The eigenvalues of such a Jacobian matrix are given by

$$\lambda_1 = m^s_{11} - m^s_{21}, \qquad \lambda_2 = m^s_{33} - m^s_{43}, \tag{6.C.2a}$$

$$\lambda_{3,4} = \frac{1}{2}\left[A_0 + B_0 \pm \sqrt{(A_0 - B_0)^2 + 4Km^s_{31}C_0}\right], \tag{6.C.2b}$$

with $A_0 = m^s_{11} + (K-1)m^s_{21}$, $B_0 = m^s_{33} + m^s_{34}$, and $C_0 = m^s_{13} + m^s_{14}$. The corresponding (unnormalized) eigenvectors $\boldsymbol{v}_i$ are given by

$$\begin{aligned} \boldsymbol{v}_1 &= \begin{pmatrix} (K-1) & -1 & 0 & 0 \end{pmatrix}, \\ \boldsymbol{v}_2 &= \begin{pmatrix} 1 & 1 & v_{23} & v_{24} \end{pmatrix}, \\ \boldsymbol{v}_{3,4} &= \begin{pmatrix} v_{(3,4);(1/2)} & v_{(3,4);(1/2)} & 1 & 1 \end{pmatrix}, \end{aligned} \tag{6.C.3a}$$

with

$$\begin{aligned} v_{23} &= \frac{m^s_{34}(m^s_{33} - m^s_{43} - A_0) + Km^s_{14}m^s_{31}}{m^s_{13}m^s_{34} - m^s_{14}m^s_{43}}, \\ v_{24} &= \frac{m^s_{43}(A_0 + m^s_{43} - m^s_{33}) - Km^s_{13}m^s_{31}}{m^s_{13}m^s_{34} - m^s_{14}m^s_{43}}, \\ v_{(3,4);(1/2)} &= \frac{\lambda_{3,4} - B_0}{Km^s_{31}}, \end{aligned} \tag{6.C.3b}$$

where the first digit indicates the eigenvalue number and the second indicates the component index.

6.C.1 Truncated equations

For the truncated differential equations, where η^2 are neglected, the onset of specialization is characterized by the eigenvalues

$$\lambda_1^0 = \frac{2}{\pi} \frac{\eta\beta T^2}{\sqrt{K(1+T)-T}\,[K(1+T)+\beta T]^{\frac{3}{2}}}, \tag{6.C.4a}$$

$$\lambda_2^0 = 0, \tag{6.C.4b}$$

$$\lambda_3^0 = -\frac{2}{\pi}\eta \left[\frac{K(1+T)-T}{K(1+T)+\beta T}\right]^{\frac{3}{2}}, \tag{6.C.4c}$$

$$\lambda_4^0 = -\frac{4}{\pi}\eta \sqrt{\frac{K(1+T)-T}{K(1+T)+\beta T}}, \tag{6.C.4d}$$

i.e., one finds only one relevant eigenvalue λ_1^0 (and one marginal eigenvalue λ_2^0). If one takes a closer look at the eigenvectors, whose non-constant terms take the form

$$v_{23}^0 = \frac{2K^{\frac{3}{2}}(1+T)}{T\sqrt{K(1+T)-T}}, \tag{6.C.5a}$$

$$v_{24}^0 = -\frac{2K^{\frac{3}{2}}}{(K-1)T\sqrt{K(1+T)-T}}, \tag{6.C.5b}$$

$$v_{3;(1/2)}^0 = \frac{2\sqrt{K}}{\sqrt{K(1+T)-T}}, \tag{6.C.5c}$$

$$v_{4;(1/2)}^0 = -\frac{2K^{\frac{3}{2}}(1+T)}{T(1+2\beta)\sqrt{K(1+T)-T}}, \tag{6.C.5d}$$

one can see that the positive eigenvalue λ_1^0 acts solely in the student-teacher overlap space. This eigenvalue is associated with a pure rotation of the weight vectors towards the teacher unit they will specialize on. The marginal eigenvalue λ_2^0 (which will be important in the case where η^2 terms are not neglected) shows an increase in the squared norm Q of the student weight vectors of $\mathcal{O}(K)$, but a decrease in their correlations C of $\mathcal{O}(1)$, which corresponds primarily to a growth of the student weight vectors outside the subspace spanned by the teacher weight vectors.

6.C.2 Small-η fixed point

To calculate the first-order correction in η to the fixed point of the truncated equations (6.8), we expand the full differential equations (6.B.2) to first order around Eqs. (6.8), and find the zeros of the resulting set of linear equations in (r, s, q, c). Examining the relations (6.C.1) more closely, one can see that the solution is characterized by $r = s$

and $q = c$, and we find for the new symmetric fixed point $Q^* = C^* = Q_0^* + Q_1^*$ and $R^* = S^* = R_0^* + R_1^*$ ignoring terms of $\mathcal{O}(\eta^2)$

$$Q_1^* = \frac{1}{\pi}\frac{[K(1+T)+2\beta T]}{[K(1+T)-T]}\mathcal{G}\mathcal{F}\frac{\eta}{K}, \tag{6.C.6a}$$

$$R_1^* = \frac{1}{2\pi}\frac{T(1+2\beta)}{\sqrt{K(1+T)-T}}\mathcal{G}\mathcal{F}\frac{\eta}{K^{\frac{3}{2}}}, \tag{6.C.6b}$$

with

$$\mathcal{G} = \frac{\sqrt{K(1+T)+\beta T}}{\sqrt{K(1+T)+(2\beta-1)T}}, \tag{6.C.6c}$$

$$\mathcal{F} = \arcsin\left(\frac{T\{K[K(1+T)-T]+(K-1)2\beta T\}}{[K(1+T)-T][K(1+T)+2\beta T]}\right) - K\arcsin\left(\frac{T}{K(1+T)K+2\beta T}\right) - (K-1)\arcsin\left(\frac{2\beta T^2}{[K(1+T)-T][K(1+T)+2\beta T]}\right). \tag{6.C.6d}$$

For the expansion to be valid, η has to be chosen to ensure $Q_1^* \ll Q_0^*$ and $R_1^* \ll R_0^*$. For large K, this implies $\eta \leq \mathcal{O}(K^{-1})$. We further note that the new fixed point is no longer confined to the subspace spanned by the teacher weight vectors as $R^* < \sqrt{Q^*T/K}$. However, the symmetries $Q = C$ and $R = S$ are not broken to first order. This is in contrast to the numerical results from integrating the full dynamics (6.A.2), where it is observed that the symmetric phase for finite learning rates is characterized by $Q > C$ (and $R = S$).

6.C.3 Small-η dynamics

To study the onset of specialization, the differential equations (6.B.2) are expanded around the new fixed point, which is again characterized by $Q = C$ and $R = S$, and the matrix relations (6.C.1) hold. Ignoring terms of $\mathcal{O}(\eta^3)$, we find that the eigenvalues (eigenvectors) of the Jacobian have acquired $\mathcal{O}(\eta^2)$ [$\mathcal{O}(\eta)$] corrections to their values in Eq. (6.C.4) [Eq. (6.C.5)]. In particular

$$\lambda_2^1 = \frac{4}{\pi^2}\frac{\sqrt{K(1+T)-T}}{K(1+T)+(2\beta-1)T}\beta\eta^2\left\{\frac{K(1+T)+(3\beta-1)T}{K\sqrt{K(1+T)+(2\beta-1)T}}\mathcal{F} + \frac{2\beta T^2}{\sqrt{K}[K(1+T)+2\beta T]}\left[\frac{1}{\sqrt{K^2(1+T)(1+2T)+K(1+2T)T(2\beta-1)-4\beta T^2}} + \frac{(K-1)\sqrt{1+T}}{\sqrt{K^2(1+T)^2+K(1+T)T(2\beta-1)-4\beta T^2}} - \frac{\sqrt{K}}{\sqrt{K(1+T)+(2\beta+1)T}}\right]\right\}, \tag{6.C.7}$$

which is, in general, positive and dominated by the $\mathcal{F}$ term, i.e., the marginal eigenvalue now becomes relevant to the dynamics. As mentioned in Appendix 6.C, the associated eigenvector (whose η dependence can be ignored as it constitutes only a minor correction) shows an increase in Q of $\mathcal{O}(K)$ and a decrease in C of $\mathcal{O}(1)$. As the increases in R and S are equal, this mode does not contribute to the specialization process but corresponds primarily to a growth of the student weight vectors outside the subspace spanned by the teacher weight vectors. Since the initial differences between Q and C are typically large, this eigenvalue will actually dominate the dynamics and quickly drive the student away from this particular fixed point. We therefore conclude that the fixed point associated with $Q = C$ is relevant only for $\eta = 0$ and that a fixed point characterized by $Q > C$ leads to the long symmetric phase for $\eta > 0$, which is not accessible by first-order correction to the fixed point studied in Appendix 6.C.2. An analytic study of that fixed point necessitates an expansion to second order and the subsequent solution of a set of quadratic equations, which we have found to be infeasible.

6.D Convergence fixed point dynamics

Note, that what follows is in some respect similar to the convergence analysis in Chapter 5, but for arbitrary β and fixed zero biases. In order to omit excessive crossreferencing and make comparisons between GD and ABP easier, we provide no pointers to previous equations and also show all results for GD, although some of them can be deduced as special cases from results derived in Appendix 5.B.

As for the symmetric fixed point, the differential equations (6.B.2) are expanded to first order around the zero generalization error fixed point $Q^* = R^* = T$ and $C^* = S^* = 0$, where the ordering $P_1 = R$, $P_2 = Q$, $P_3 = S$, and $P_4 = C$ was used for the convergence phase [again following the convention of earlier work (Saad and Solla 1995b)]. Similarly, the generalization error (6.B.1) is expanded also to second order. Explicitly, one finds for the generalization error

$$\epsilon_g = \frac{1}{\pi}\left\{\frac{1}{\sqrt{1+2T}}\left[(2r-q) - \frac{1}{4}\frac{T(2r-q)^2}{1+2T} + \frac{q(r-q)}{1+2T}\right] - \frac{K-1}{1+T}\left[(2s-c) + \frac{q(s-c)}{1+T}\right]\right\}, \quad \text{(6.D.1)}$$

The elements of the Jacobian matrix $\mathbf{M}^c$ are given by

$$m_{11}^c = -\frac{2}{\pi}\frac{\eta}{K}\frac{1+(1+2\beta)T}{[1+(1+\beta)T]^{\frac{3}{2}}}, \quad \text{(6.D.2a)}$$

$$m_{12}^{c} = \frac{1}{\pi}\frac{\eta}{K}\frac{T(1+2\beta)}{[1+(1+\beta)T]^{\frac{3}{2}}}, \tag{6.D.2b}$$

$$m_{13}^{c} = \frac{2}{\pi}\frac{\eta}{K}\frac{(K-1)(1+2\beta T)}{\sqrt{1+T}(1+\beta T)^{\frac{3}{2}}}, \tag{6.D.2c}$$

$$m_{14}^{c} = -\frac{2}{\pi}\frac{\eta}{K}\frac{(K-1)\beta T}{\sqrt{1+T}(1+\beta T)^{\frac{3}{2}}}, \tag{6.D.2d}$$

$$m_{21}^{c} = \frac{4}{\pi}\frac{\eta}{K}\left\{\frac{1+T}{[1+(1+\beta)T]^{\frac{3}{2}}} - \frac{2}{\pi}\frac{\eta}{K}\left[\frac{1}{\sqrt{1+2(1+\beta)T}} + \frac{(K-1)}{\sqrt{(1+2\beta T)(1+2T)}}\right]\right\}, \tag{6.D.2e}$$

$$m_{23}^{c} = -\frac{4}{\pi}\frac{\eta}{K}\frac{(K-1)}{\sqrt{1+T}}\left\{\frac{1}{(1+\beta T)^{\frac{3}{2}}} - \frac{2}{\pi}\frac{\eta}{K}\left[\frac{2}{\sqrt{1+(1+2\beta)T}} + \frac{(K-2)}{\sqrt{(1+2\beta T)(1+T)}}\right]\right\}, \tag{6.D.2f}$$

$$m_{31}^{c} = \frac{2}{\pi}\frac{\eta}{K}\frac{1}{\sqrt{(1+\beta T)(1+T)}}, \tag{6.D.2g}$$

$$m_{32}^{c} = -\frac{1}{\pi}\frac{\eta}{K}\frac{T}{\sqrt{1+\beta T}(1+T)^{\frac{3}{2}}}, \tag{6.D.2h}$$

$$m_{33}^{c} = -\frac{2}{\pi}\frac{\eta}{K}\left[\frac{1}{\sqrt{1+(1+\beta)T}} + \frac{(K-2)}{\sqrt{(1+\beta T)(1+T)}}\right], \tag{6.D.2i}$$

$$m_{34}^{c} = 0, \tag{6.D.2j}$$

$$m_{41}^{c} = -\frac{4}{\pi}\frac{\eta}{K}\frac{1}{\sqrt{1+\beta T}}\left\{\frac{1}{\sqrt{1+T}} - \frac{2}{\pi}\frac{\eta}{K}\left[\frac{2}{\sqrt{1+(2+\beta)T}} + \frac{(K-2)}{\sqrt{(1+\beta T)(1+2T)}}\right]\right\}, \tag{6.D.2k}$$

$$m_{43}^{c} = \frac{4}{\pi}\frac{\eta}{K}\left\{\frac{1}{\sqrt{1+(1+\beta)T}} + \frac{(K-2)}{\sqrt{(1+\beta T)(1+T)}} - \frac{2}{\pi}\frac{\eta}{K}\left[\frac{2}{1+(1+\beta)T} + \frac{(K-2)}{(1+\beta T)(1+T)}\left(4\frac{\sqrt{(1+\beta T)(1+T)}}{\sqrt{1+(1+\beta)T}} + (K-3)\right)\right]\right\}. \tag{6.D.2l}$$

The remaining elements can be deduced by the matrix relations (Riegler 1997)

$$\begin{aligned} m_{11}^{c} - \tfrac{1}{2}m_{21}^{c} &= m_{22}^{c} - 2m_{12}^{c}, & m_{33}^{c} - \tfrac{1}{2}m_{43}^{c} &= m_{44}^{c} - 2m_{34}^{c}, \\ m_{13}^{c} - \tfrac{1}{2}m_{23}^{c} &= m_{24}^{c} - 2m_{14}^{c}, & m_{31}^{c} - \tfrac{1}{2}m_{41}^{c} &= m_{42}^{c} - 2m_{32}^{c}. \end{aligned} \tag{6.D.3}$$

The eigenvalues of such a Jacobian matrix are given by the solutions to two quadratic equations

$$\lambda_{1,2} = \frac{1}{2}\left[A_1 + B_1 \pm \sqrt{(A_1 - B_1)^2 + 4C_1 D_1}\right] \tag{6.D.4a}$$

$$\lambda_{3,4} = \frac{1}{2}\left[A_2 + B_2 \pm \sqrt{(A_2 - B_2)^2 + 4C_2 D_2}\right], \tag{6.D.4b}$$

with

$$\begin{aligned} A_1 &= m_{11}^c - \tfrac{1}{2}m_{21}^c, & B_1 &= m_{44}^c - 2m_{34}^c, \\ C_1 &= m_{31}^c - \tfrac{1}{2}m_{41}^c, & D_1 &= m_{24}^c - 2m_{14}^c, \\ A_2 &= m_{11}^c + 2m_{12}^c, & B_2 &= m_{44}^c + \tfrac{1}{2}m_{43}^c, \\ C_2 &= m_{31}^c + 2m_{32}^c, & D_2 &= m_{24}^c + \tfrac{1}{2}m_{23}^c. \end{aligned} \tag{6.D.4c}$$

The corresponding (unnormalized) eigenvectors $\boldsymbol{v}_i$ are given by

$$\boldsymbol{v}_{1,2} = \begin{pmatrix} v_{(1,2);1} & v_{(1,2);2} & v_{(1,2);3} & v_{(1,2);4} \end{pmatrix}, \tag{6.D.5a}$$

$$\boldsymbol{v}_{3,4} = \begin{pmatrix} 1 & 2 & v_{(3,4);(3/4)} & 2v_{(3,4);(3/4)} \end{pmatrix}, \tag{6.D.5b}$$

with (using $m_{34}^c = 0$)

$$v_{(3,4);(3/4)} = \frac{\lambda_{3,4} - A_2}{D_2} \tag{6.D.5c}$$

$$v_{(1,2);1} = -\left\{2D_1\left[m_{14}^c C_1 + m_{12}^c\left(B_2 - \lambda_{1,2}\right) + m_{32}^c D_2\right] + m_{43}^c m_{14}^c\left(A_1 - \lambda_{1,2}\right)\right\}, \tag{6.D.5d}$$

$$\begin{aligned} v_{(1,2);2} = {} & m_{21}^c D_1(\lambda_{1,2} - m_{44}^c) + m_{43}^c m_{24}^c(\lambda_{1,2} - m_{11}^c) \\ & + D_1(m_{31}^c m_{23}^c + m_{41}^c m_{24}^c) + m_{43}^c m_{21}^c m_{14}^c, \end{aligned} \tag{6.D.5e}$$

$$\begin{aligned} v_{(1,2);3} = {} & 2m_{31}^c m_{14}^c(A_2 - \lambda_{1,2}) + 2m_{32}^c m_{24}^c(m_{11}^c - \lambda_{1,2}) \\ & - m_{14}^c m_{21}^c C_2 - 2m_{24}^c m_{12}^c m_{31}^c, \end{aligned} \tag{6.D.5f}$$

$$\begin{aligned} v_{(1,2);4} = {} & \frac{1}{C_1}(\lambda_{1,2} - A_1)\left\{2\left(m_{21}^c m_{32}^c - m_{12}^c m_{41}^c\right)(\lambda_{1,2} - m_{44}^c)\right. \\ & \left. + C_1\left[m_{21}^c(\lambda_{1,2} - m_{44}^c) + m_{43}^c(\lambda_{1,2} - A_2) + m_{41}^c D_1 + m_{23}^c C_2\right]\right\}. \end{aligned} \tag{6.D.5g}$$

Comparing the eigenvectors (6.D.5) with the expansion of the generalization error (6.D.1), one finds that the modes $\boldsymbol{v}_{3,4}$ are orthogonal to the first-order terms in the generalization error and therefore cannot contribute to their decay. These modes are therefore only relevant for second-order terms in the generalization error with a de-

cay rate of $2\lambda_{3,4}$. As discussed in Section 6.4.2, the fastest convergence is given by Eq. (6.12). This is achieved either for $\eta_{\mathrm{r}}^{\mathrm{opt}}$, where $2\lambda_3 = \lambda_1$, or for $\eta_{\mathrm{m}}^{\mathrm{opt}}$, which is defined by the minimum of λ_1. The critical (maximal) learning rates are defined by the zeros of the determinant in η

$$A_1 B_1 - C_1 D_1 = 0 \tag{6.D.6a}$$
$$A_2 B_2 - C_2 D_2 = 0, \tag{6.D.6b}$$

where only one nonzero learning rate solution exists in Eq. (6.D.6a), coinciding with $\lambda_1 = 0$.

Unfortunately, it is in general infeasible to optimize the eigenvalues with respect to the learning parameters η and β analytically for arbitrary K and T. However, one can make some progress in certain limits of K and T, which will be investigate below.

6.D.1 Large-K limit

The dominant terms for a large number of hidden units for all relevant quantities can be extracted by an asymptotic series expansion under the self-consistent ansatz $\eta = \mathcal{O}(1)$ and $\beta = \mathcal{O}(1)$. For the two relevant eigenvalues one makes the ansatz $\lambda_i = \mathcal{O}(K^{-1})$ and finds to leading order

$$\lambda_1(\beta) = -\frac{4}{\pi}\frac{\eta}{K}\frac{\pi\chi_1 - \eta\chi_2}{\mathcal{E}_1\mathcal{E}_2\mathcal{E}_3(\pi\mathcal{E}_1 - \eta)}, \tag{6.D.7a}$$
$$\lambda_3(\beta) = -\frac{2}{\pi}\frac{\eta}{K}\left(\mathcal{E}_3^{-3} - \mathcal{E}_1^{-3}\right), \tag{6.D.7b}$$

with the auxiliary variables

$$\chi_1 = \mathcal{E}_1\mathcal{E}_2\left(\mathcal{E}_1 - \mathcal{E}_3\right), \tag{6.D.7c}$$
$$\chi_2 = \mathcal{E}_1\mathcal{E}_2 - \mathcal{E}_3\left[\sqrt{1+2\beta T}(1+T) + \sqrt{1+2T}(1+\beta T) - \mathcal{E}_1^2\right], \tag{6.D.7d}$$
$$\mathcal{E}_1 = \sqrt{(1+T)(1+\beta T)}, \tag{6.D.7e}$$
$$\mathcal{E}_2 = \sqrt{(1+2T)(1+2\beta T)}, \tag{6.D.7f}$$
$$\mathcal{E}_3 = \sqrt{1+(1+\beta)T}. \tag{6.D.7g}$$

These define two critical learning rates

$$\eta_{\mathrm{crit}}^0(\beta) = \pi\frac{\chi_1}{\chi_2}, \tag{6.D.8a}$$
$$\eta_{\mathrm{crit}}^\infty(\beta) = \pi\mathcal{E}_1 > \eta_{\mathrm{crit}}^0, \tag{6.D.8b}$$

where λ_1 is identical to zero ($\eta^0_{\rm crit}$) and diverges ($\eta^\infty_{\rm crit}$), respectively. Solving Eq. (6.D.6a), one finds $\eta_{\rm max} = \eta^0_{\rm crit}$, as expected. It is important to realize that Eq. (6.D.7a) is only a valid expansion for λ_1 for $\eta < \eta^\infty_{\rm crit}$, beyond which the ansatz $\lambda_1 = \mathcal{O}(K^{-1})$ breaks down as $\lambda_1 = \mathcal{O}(1)$. In fact, the order of the two eigenvalues λ_1 and λ_2 changes at $\eta^\infty_{\rm crit}$ and Eq. (6.D.7a) is the correct asymptotic expansion of λ_2 for $\eta > \eta^\infty_{\rm crit}$. This change in the order of eigenvalues can be seen quite well in Figure 6.6(a), as the natural continuation for λ_1 for large η follows the curve representing λ_2 and vice versa. As mentioned above, one has to calculate, in general, both $\eta_{\rm r}^{\rm opt}$ and $\eta_{\rm m}^{\rm opt}$ by solving $2\lambda_3 = \lambda_1$ and $d\lambda_1/d\eta = 0$, respectively. Due to the breakdown of the ansatz for λ_1 above $\eta^\infty_{\rm crit}$, solutions with $\eta^{\rm opt} > \eta^\infty_{\rm crit}$ are spurious.

For GD the eigenvalues and the critical learning rates simplify to

$$\lambda_1(1) = -\frac{4}{\pi}\frac{\eta}{K}[(1+T) - \sqrt{1+2T}]\frac{\pi\sqrt{1+2T} - \eta}{(1+2T)\,[\pi(1+T) - \eta]}, \tag{6.D.9a}$$

$$\lambda_3(1) = -\frac{2}{\pi}\frac{\eta}{K}\left[(1+2T)^{-\frac{3}{2}} - (1+T)^{-3}\right], \tag{6.D.9b}$$

$$\eta^0_{\rm crit}(1) = \pi\sqrt{1+2T}, \tag{6.D.9c}$$

$$\eta^\infty_{\rm crit}(1) = \pi(1+T), \tag{6.D.9d}$$

resulting in the two candidates for the optimal learning rate taking the form

$$\eta_{\rm r}^{\rm opt}(1) = \frac{\eta^\infty_{\rm crit} T\left[2(1+T)^3 - (2+T)(1+2T)^{\frac{3}{2}}\right]}{(1+T)^4(\sqrt{1+2T} - 2) + (1+2T)^{\frac{3}{2}}}, \tag{6.D.10a}$$

$$\eta_{\rm m}^{\rm opt}(1) = \eta^\infty_{\rm crit} - \pi\sqrt{1+T}\left[(1+T) - \sqrt{1+2T}\right]^{\frac{1}{2}}. \tag{6.D.10b}$$

To decide on the correct learning rate for given T, one has to evaluate whether $\eta_{\rm r}^{\rm opt}(1) < \eta^\infty_{\rm crit}(1)$ and then calculate the convergence rates for the two learning rates. We find that $\eta^{\rm opt}(1) = \eta_{\rm r}^{\rm opt}(1)$ for $T > T^{\rm crit}$ and $\eta^{\rm opt}(1) = \eta_{\rm m}^{\rm opt}(1)$ for $T < T^{\rm crit}$, where $T^{\rm crit} = 1.2780$ is defined by $\eta_{\rm r}^{\rm opt}(1) = \eta_{\rm m}^{\rm opt}(1)$.

When optimizing β, one always finds that the fastest convergence is achieved for $2\lambda_3 = \lambda_1$ and the optimal learning rate is determined by

$$\eta^{\rm opt}(\beta) = \pi\mathcal{E}_2 T\left\{\mathcal{E}_1^4(1+\beta) + \mathcal{E}_1\mathcal{E}_3^3\left[1+\beta(1+T)\right]\right\} \times \Big\{\mathcal{E}_1^3\mathcal{E}_2(1+\beta)T$$
$$-\mathcal{E}_3^3\left[\sqrt{1+2T}(1+\beta T)\mathcal{E}_1^2 + \sqrt{1+2\beta T}(1+T)\mathcal{E}_1^2 - \mathcal{E}_1^4 - \mathcal{E}_2\right]\Big\}^{-1}. \tag{6.D.11}$$

The optimal convergence rate, which is just given as $2\lambda_3$ at $\eta^{\rm opt}$, however, cannot be further optimized analytically with respect to β and this optimization has to be done

numerically. The results for β^{opt} and all other interesting quantities in this limit can be seen in Figures 6.7 and 6.8.

To make further progress in the $K \to \infty$ limit, one can look at the limits $T \to \infty$ and $T \to 0$. These results turn out to be equivalent, to leading order in K and T, to results where both T and K go to their limits simultaneously, i.e., taking the limit $K \to \infty$ with $T = T_\infty K$ and $T = T_0/K$, respectively. T_0 and T_∞ are prefactors controlling the significance between T and K. Therefore, the more general expansion in both variables has been used below for higher-order terms. Unfortunately, this was infeasible for higher-order terms for optimized ABP in the small-T limit, where results are presented that were obtained by taking the large-K limit first.

6.D.2 Small-T limit ($T = T_0/K$)

For GD the leading terms of the relevant quantities in this limit are

$$\eta_{\max} = \pi \left[1 + T - \frac{1}{2}T^2 + \frac{1}{2}\frac{T^2}{K}(TK - 4)\right], \tag{6.D.12a}$$

$$\eta^{\text{opt}} = \pi \left[1 + \frac{1}{2}\left(2 - \sqrt{2}\right)T - \frac{\sqrt{2}}{4}\frac{T}{K}\right], \tag{6.D.12b}$$

$$\lambda^{\text{opt}} = -2\frac{T^2}{K}\left[1 - \left(2 + \sqrt{2}\right)T + \frac{19 + 12\sqrt{2}}{4}T^2 + \frac{\sqrt{2}}{2}\frac{T}{K}\right], \tag{6.D.12c}$$

with $TK = T_0 = \mathcal{O}(1)$. The optimization for ABP yields, for $K \to \infty$ preceding $T \to 0$,

$$\beta^{\text{opt}} = \frac{2}{T} + \frac{3}{10}\frac{5^{\frac{3}{4}}\sqrt{6}(\sqrt{5} - 1)}{\sqrt{T}}, \tag{6.D.13a}$$

$$\eta_{\max} = \pi\sqrt{3}\left[1 + \frac{5^{\frac{3}{4}}\sqrt{6}(\sqrt{5} - 1)}{20}\sqrt{T} - \frac{9\sqrt{5} - 19}{8}T\right], \tag{6.D.13b}$$

$$\eta^{\text{opt}} = \pi\sqrt{3}\left[1 - \frac{1519\sqrt{5} - 3315}{300(3 - \sqrt{5})}T\right], \tag{6.D.13c}$$

$$\lambda^{\text{opt}} = -\frac{4}{3}\frac{T}{K}\left[1 - \frac{5^{\frac{3}{4}}\sqrt{6}(3 - \sqrt{5})}{5(\sqrt{5} - 1)}\sqrt{T}\right]. \tag{6.D.13d}$$

In this limit ABP yields in leading order a factor of $\frac{2}{3}T^{-1}$ in reduction of training time due to the increase of $\beta^{\text{opt}} \propto T^{-1}$. Furthermore, the decrease in the normalized gap between $\eta_{\max}$ and η^{opt} is slowed down proportional to $1/\sqrt{T}$.

6.D.3 Large-T limit ($T = T_\infty K$)

For GD the leading terms of the relevant quantities in this limit are

$$\eta_{\max} = \pi\sqrt{2}\sqrt{T}\left[1 - \frac{\sqrt{T}}{K} + \frac{(1+2T_\infty)^2}{4T}\right], \tag{6.D.14a}$$

$$\eta^{\text{opt}} = \eta_{\max} - \frac{\pi\sqrt{2}}{2\sqrt{T}}, \tag{6.D.14b}$$

$$\lambda^{\text{opt}} = -\frac{2}{KT}\left[1 - \frac{\sqrt{T}}{K} + \frac{T_\infty^2 + T_\infty - 1}{T}\right], \tag{6.D.14c}$$

whereas the optimization for ABP gives

$$\beta^{\text{opt}} = \frac{1}{3} - \frac{1}{18}\frac{3\sqrt{2}T_\infty + 8\sqrt{6} - 12 - 2\sqrt{3}}{\sqrt{T}}, \tag{6.D.15a}$$

$$\eta_{\max} = \pi\sqrt{T} - \frac{\pi}{16}\left[11\sqrt{2}T_\infty + 20 + 14\sqrt{3} - 8\sqrt{2}\left(2+\sqrt{3}\right)\right], \tag{6.D.15b}$$

$$\eta^{\text{opt}} = \eta_{\max} - \frac{3}{4}\frac{\pi}{\sqrt{T}}, \tag{6.D.15c}$$

$$\lambda^{\text{opt}} = -\frac{3}{2}\frac{\sqrt{3}}{KT}\left[1 - \frac{T_\infty - (2-\sqrt{2})(\sqrt{3}-\sqrt{2})}{\sqrt{2}\sqrt{T}}\right]. \tag{6.D.15d}$$

In this limit ABP yields only a constant factor of $3\sqrt{3}/4 \approx 1.2990$ in reduction of training time and an increase in the learning rate gap by a factor 3/2. This should be contrasted to the increase in training time for both algorithms by a factor T and a decrease in the normalized learning rate gap of T^{-1}. Two logical further extensions are to look at the limits $T \to 0$ and $T \to \infty$ for K finite, especially as the numerical solutions indicate [see Figure 6.7(b)] that there are qualitative changes in the learning behaviour at least for $T \to \infty$.

6.D.4 Small-T limit

For small T, where the network becomes nearly linear, one should only expect minor changes to the limits studied previously since the network behaves smoothly. In particular, one finds for GD

$$\eta_{\max} = \pi\left[1 + T - \frac{K+4}{2K}T^2\right], \tag{6.D.16a}$$

$$\eta^{\text{opt}} = \pi\left[1 + \left(1 - \sqrt{\frac{K-1}{2K}}\right)T(1+T)\right], \tag{6.D.16b}$$

$$\lambda^{\text{opt}} = -2\frac{T^2}{K}\left[1 - 2\left(1 + \sqrt{\frac{K-1}{2K}}\right)T\right]. \tag{6.D.16c}$$

For ABP only the leading term is feasible to calculate, resulting in

$$\beta^{\text{opt}} = \frac{2}{T}, \tag{6.D.17a}$$

$$\eta_{\max} = \pi\sqrt{3}\frac{5K}{5(K-1)+3\sqrt{5}}, \tag{6.D.17b}$$

$$\eta^{\text{opt}} = \eta_{\max}, \tag{6.D.17c}$$

$$\lambda^{\text{opt}} = -\frac{4}{3}\frac{5T}{5(K-1)+3\sqrt{5}}, \tag{6.D.17d}$$

which explains the very weak influence of K on the previous results (besides the natural rescaling of λ^{opt} with K^{-1}).

6.D.5 Large-T limit

Unlike for small T, one finds significant changes in the learning behaviour of both algorithms in the large-T limit. For GD one finds for the leading orders

$$\eta_{\max} = \pi\sqrt{2}K\left[1 - \frac{K-1}{\sqrt{T}}\right], \tag{6.D.18a}$$

$$\eta^{\text{opt}} = \eta_{\max} - \frac{\pi\sqrt{2}K}{2T}, \tag{6.D.18b}$$

$$\lambda^{\text{opt}} = -\frac{2}{T^{\frac{3}{2}}}\left[1 - \frac{K-1}{\sqrt{T}}\right]. \tag{6.D.18c}$$

For ABP the numerical solutions suggest the self-consistent ansatz $\beta^{\text{opt}} \propto T^{-\frac{1}{3}}$ and the leading terms are

$$\beta^{\text{opt}} = \frac{1}{6}\left[\frac{12(K-1)^2}{T}\right]^{\frac{1}{3}} - \frac{5K+19}{54}\left[\frac{18(K-1)}{T^2}\right]^{\frac{1}{3}}, \tag{6.D.19a}$$

$$\eta_{\max} = \pi K\left\{\sqrt{2} - \left[\frac{3\sqrt{2}(K-1)^2}{T}\right]^{\frac{1}{3}} - \frac{3K+1}{18}\left[\frac{36\sqrt{2}(K-1)}{T^2}\right]^{\frac{1}{3}}\right\}, \tag{6.D.19b}$$

$$\eta^{\text{opt}} = \eta_{\max} - \frac{\pi\sqrt{2}K}{T}, \tag{6.D.19c}$$

$$\lambda^{\text{opt}} = -\frac{1}{T^{\frac{3}{2}}}\left\{4\sqrt{2} - 6\left[\frac{3\sqrt{2}(K-1)^2}{T}\right]^{\frac{1}{3}} + \frac{37K+11}{12}\left[\frac{36\sqrt{2}(K-1)}{T^2}\right]^{\frac{1}{3}}\right\}. \tag{6.D.19d}$$

In this limit ABP yields a larger constant factor of $2\sqrt{2} \approx 2.828$ in reduction of training time and an increase in the learning rate gap by a factor 2, which is somewhat better than for the infinite-K case.

Chapter 7

Conclusions

7.1 Summary

In this thesis I have conducted a variety of research in two of the main problem fields of neural network research from the statistical physics viewpoint: the memorization capability (Chapters 3 and 4) and the generalization ability (Chapters 5 and 6) of simple network models. The two main aims of this work have been the study of the rôle of biases in neural network models (Chapters 3 and 5) and the investigation of the capabilities of multilayer networks (Chapters 4–6). Due to the diversity of calculations involved, most conclusions have been drawn in the individual chapters, and here we will summarize only the main results, concentrating on the limitations and possible extensions.

In Chapter 3, the performance of the Boolean perceptron in learning random dichotomies above its saturation limit was investigated. Although the Boolean perceptron is one of the simplest neural network models, the inclusion of an adjustable bias has a dramatic impact on its behaviour. The naive assumption that an unbiased output distribution should automatically lead to order-parameter solutions that mirror this symmetry, i.e., zero bias, holds only for zero stability. For any finite stability, the increase in error triggers a second-order phase transition in order-parameter space at a stability-dependent critical load, with spontaneous symmetry breaking in the space of biases. A simple analogy can be drawn: the load corresponds to the inverse temperature in a ferromagnet, which exhibits finite magnetization (corresponding to finite bias) above a critical inverse temperature. Similarly, the output distribution bias can be identified with an external magnetic field that breaks the symmetry and smoothes out the phase transition. Similar non-trivial effects for the Boolean perceptron have been reported in the case of generalization (Wendemuth 1994a), where the generalization

error shows non-monotonic behaviour. The non-invertiblility of the Boolean output function may be instrumental in explaining why such non-trivial behaviour exists in single-layer networks.

The results for the Boolean perceptron in Chapter 3 were then used to calculate the capacity of multilayer networks built by constructive algorithms in Chapter 4. The capacity of multilayer networks is of interest due to its relation to the VC dimension and hence generalization. The inherent difficulties of the replica framework has, however, hampered progress in the evaluation of the capacity for multilayer networks until very recently (Monasson and Zecchina 1996; Urbanczik 1997). The approach taken in this thesis avoids these difficulties yielding a good approximation of the capacity for network sizes realistic for practical considerations. It furthermore enabled us to compare the capacity performance of various constructive algorithms. In particular, the performance degradation to be expected from using constructive algorithms, in comparison to fully connected architectures, is surprisingly small, considering the fact that the optimization is local and only a much smaller space of internal representations is accessible. The capacity curves calculated suggest that the capacity of the constructed networks also behaves with a power-law in $\log K$. However, due to finite size effects reliable estimates of the exponent value could not be reported. Nevertheless, several interesting properties were found, such as that the simpler replica symmetric treatment violates theoretical upper bounds only slightly, whereas in fixed architecture cases power-laws in K have been reported (Barkai et al. 1992; Engel et al. 1992) and the failure is much more severe.

In Chapter 5, we studied the influence of biases on the generalization performance of a smooth multilayer network model within the on-line learning paradigm of supervised learning. The model considered, the soft-committee machine, is in principle a universal approximator and only a slightly simplified version of the MLP architecture most widely used in applications. As for the simple perceptron, we find that the inclusion of biases alters the behaviour of the model considerably. The dynamics of the original model with fixed biases suggests, that the breaking of hidden node symmetry is the major obstacle in training multilayer networks (since the symmetric learning phase, characterized by a lack of specialization of the student hidden units, dominates the learning dynamics for many teacher tasks). This picture is dramatically altered when one allows for dynamic biases, where asymmetric initial student biases can break the node symmetry almost immediately, provided that the biases of the teacher task are not symmetric. For teacher tasks with symmetric biases, i.e., tasks which are similar to the original model, the inclusion of bias dynamics can severly prolong the training process and can in many cases even trap the learning process indefinitely in attractive

fixed points — a behaviour which could not have been anticipated from studying the fixed-bias model. Training failures, where networks become stuck in local minima, are a well-known problem in practical application and the above behaviour may provide a theoretical explanation. This view can be corroborated, as both in the model studied and in real world applications, such behaviour is closely linked to the student parameter initialization, which, conversely, is of negligible importance in the fixed bias case. We believe that in multilayer networks, the existence of node symmetry (which is not a problem, for example, in multilayer networks with non-overlapping fields) is the explanation for the dramatic alteration of the behaviour by biases. In fact, once the hidden unit symmetry is broken, optimal convergence is achieved for an infinite bias learning rate[1], suggesting that the input-hidden weights dominate the learning behaviour in this phase.

The results in Chapter 5, in combination with previous results (Saad and Solla 1995b) showing that for relatively symmetric teacher tasks the student network takes a long time to break its internal node symmetry, prompted the investigation of its causes and exploration for possible improvements in Chapter 6. Here, we introduced a slight modification of the standard back-propagation algorithm which deforms search space by an adjustable parameter β, which can be smoothly tuned between favouring longitudinal or rotational changes in comparison to gradient descent. We find remarkable improvements in many areas, signified by an optimal choice of β far removed from $\beta = 1$, which corresponds to standard gradient descent.

7.2 Limitations and outlook

Some interesting questions have been answered in this thesis, but many more remain open and, indeed, new ones have arisen. There are several areas where I hope that the results of the thesis will help to instigate renewed efforts.

Let us begin with the more obvious omissions and extensions for the models studied. For the Boolean perceptron this may include the extension of further RSB breaking, as has been performed in the case without threshold (Whyte and Sherrington 1996; Györgyi and Reimann 1997), although only minor corrections are to be expected. Of arguably greater importance is a more thorough investigation into whether 1RSB in the macrocanonical approach is exact for the Ising perceptron. This could be achieved by either numerical investigation of higher RSB breaking equations or by a stability

[1]The bias learning rate remains, however, of $\mathcal{O}(1/N)$, suggesting that the correct scaling of the bias learning rate may change from $\mathcal{O}(1/N)$ to $\mathcal{O}(1)$ after the hidden unit symmetry is broken.

analysis. Furthermore, it would be interesting to extend the multi-fractal weight cell structure calculations (Weigt and Engel 1997) to perceptrons with threshold.

The capacity calculation of constructive algorithms has been unsatisfactory in several ways. Arguably, the most important is that finite size effects have hampered the calculation of good (and consistent) estimates for the suggested power-law behaviour of the capacity limit as a function of the number of hidden units. The obvious remedy of calculating the capacity for even larger networks may be infeasible due to the number of hidden units required to reach the asymptotic regime provided that the numerical burden cannot be reduced significantly by approaches such as the derivation of a recurrence relation. We have also only studied two constructive algorithms, and it would be interesting to identify the advantages and disadvantages of other algorithms investigable within this framework. Furthermore, it would be interesting to study the generalization behaviour of networks built by such algorithms and compare their memorization and generalization ability, in light of the suggestion that a high capacity limit is associated with poor generalization (Opper 1994).

In the case of generalization curves of multi-layer neural networks for on-line learning, the most glaring one is that we have restricted ourselves mainly to realizable cases. The brief study of the impact of structural unrealizability in Chapter 5 has shown that there remains much to be done and understood. This extension presents no technical difficulty, since the equations were derived for any number of hidden units for both teacher and student, but may require quite exhaustive numerical studies and very careful analysis. The extension to noisy rules also represents no major technical challenge and may bring further insights, especially on the influence of noise on the basin of attractions associated with fixed points. A more detailed and/or principled study of the impact of adiabatic elimination before the breaking of student node symmetry would also be desirable (see below). A further, minor, extension would be the study of the influence of biases in the case of non-overlapping hidden unit fields. We believe, however, that the biases may be eliminated adiabatically since the breaking of internal node symmetry plays no rôle in training such networks.

The study of the short-comings of gradient descent in Chapter 6 has brought the insight that a more general functional form of the learning rule than gradient descent should be employed in order to achieve fast training. The setting of globally optimal[2] separate learning rate parameters for individual hidden units has recently shown some success (Saad and Rattray 1997a; Rattray and Saad 1997a). Furthermore, the extension of this global variational approach to the functional form of the learning rule

[2] in the sense of generalization

(Rattray and Saad 1997b), along the lines of the ideas developed for local optimization (Kinouchi and Caticha 1992) seems a promising step in the right direction. However, this approach still suffers from three problems: The employed functional ansatz does not include the rule, which is known to be optimal asymptotically (Opper 1996; Amari 1997b; Amari 1997a); it does not include biases in the model; finally, the resulting equations are only solvable in closed form for rather trivial problems (Rattray and Saad 1997b). Nevertheless, further studies along these lines, including making justified approximations, may eventually lead to more sophisticated on-line learning algorithms that can both avoid long or even infinite trapping around fixed points and converge optimally asymptotically. Other theoretically interesting questions may be answered in this course, such as whether the existence of the symmetric phase is merely a dynamical problem caused by the inefficiency of gradient descent or whether it is attributed to a lack of information about the rule [supported by the possible sudden node symmetry breaking effect in batch-learning of a hard-committee machine (Schwarze 1993)]. From the viewpoint of this thesis, it would be of interest as to whether the inclusion of the biases in either the globally-optimal learning rate and/or rule framework can shed some light on the perceived failure of adiabatic elimination (which could be seen as locally optimal) for biases.

Let us now point out further omissions and extensions which go somewhat beyond the scope of this thesis. In the case of capacity calculations, it would be worthwhile to investigate whether it is possible to extend the analysis of the capacity of the hard-committee machine, which has been calculated recently (Monasson and Zecchina 1996; Urbanczik 1997) to finite stabilities and biased distributions, where the inclusion of biases may trigger similar phase transitions as found in the case of the simple Boolean perceptron in Chapter 3. This would further allow us to gain a better understanding of the influences of stability and bias on the capacity limit, as already achieved in Chapter 4 for constructive algorithms.

In the case of generalization, it would also be interesting (but also technically difficult) to include biases in the study of batch learning in hard-committee machines (Schwarze and Hertz 1992; Schwarze 1993) since highly non-trivial effect have already been reported for the Boolean perceptron (Wendemuth 1994a). An even more desirable extension would be to the soft-committee machine calculation for finite training sets. This seems technically infeasible in the case of equilibrium calculations. For on-line learning (or dynamics of batch learning), however, the framework described in (Mace and Coolen 1997) may provide a feasible alternative resulting in good approximations, although a simplified treatment of this framework can lead to both quantitatively and qualitatively incorrect results (Sollich and Barber 1997c).

A further undesirable restriction in the analysis of on-line learning has been the assumption that the number of hidden units K remains finite, whilst the input dimension N grows arbitrarily large. In practice, one often finds the reverse case, i.e., the number of hidden units is much larger than the input dimension. Indeed, theoretically, the number of hidden units is unbounded in universal approximation proofs. It would therefore be highly desirable to extend the theoretical framework used here to encompass the case where K is of the order of N. The associated problems are evident from this thesis: self-averaging is likely to break down and the number of order parameters grows with K^2, making it impossible to carry out the thermodynamic limit of infinite input dimension without the introduction of an order parameter distribution description with a finite number of parameters.

To be able to carry out some of the described programs may involve the simplification of the model by neglecting the influence of biases. However, the work presented in this thesis has hopefully made it clear that, once a model is solved, attempts should be made to either assess the influence of biases qualitatively or to extend the analysis to biases. We believe that this is of particular importance in cases where the transfer function is non-invertible or when node symmetry in the hidden unit space of multilayer networks exists.

Appendix A

Mathematical Identities

A.1 Integral representations

For the δ-funtion the following integral representations have been extensively use in Chapter 3 and Chapter 4

$$\delta(x-x_0)=\int_{-\infty}^{\infty}\frac{d\lambda}{2\pi}\exp\left[\mathrm{i}\lambda(x-x_0)\right]=\int_{-\mathrm{i}\infty}^{\mathrm{i}\infty}\frac{d\lambda}{2\pi\mathrm{i}}\exp\left[\lambda(x-x_0)\right]. \tag{A.1}$$

For the erf-function the integral representation below has been frequently applied in Chapter 5 and Chapter 6

$$\mathrm{erf}\left[\frac{1}{\sqrt{2}}(\sigma x-x_0)\right]=1-2\int_{-x_0}^{\infty}\frac{d\lambda}{\sqrt{2\pi}}\exp\left[-\frac{1}{2}(\lambda+\sigma x)^2\right]. \tag{A.2}$$

A.2 Asymptotic expansions

For the H-funtion, $[H(x)=\int_x^{\infty}Dt]$, an asymptotic expansion was frequently employed in the $\beta\to\infty$ limit in Chapter 3 and Chapter 4

$$\lim_{x\to\infty}H(x)=\exp\left(-\frac{1}{2}x^2\right)\frac{1}{\sqrt{2\pi}x}\left\{1+\sum_{k=1}^{\infty}\frac{(-1)^k}{x^{2k}}[1\times 3\times\cdots\times(2k-1)]\right\}. \tag{A.3}$$

A.3 General Gaussian integrals

Gaussian integrals have been calculated in either directions (for the linearization of quadratic terms) with the general formula

$$\int_{-\infty}^{\infty} \frac{d\boldsymbol{x}}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\mathbf{B}^{-1}\boldsymbol{x} + \boldsymbol{x}\cdot\boldsymbol{b} + b_0\right)\left(\boldsymbol{x}^{\mathrm{T}}\mathbf{A}\boldsymbol{x} + \boldsymbol{x}\cdot\boldsymbol{a} + a_0\right) =$$

$$|\mathbf{B}|^{\frac{1}{2}} \exp\left(\frac{1}{2}\boldsymbol{b}^{\mathrm{T}}\mathbf{B}\boldsymbol{b} + b_0\right)\left[\mathrm{Tr}(\mathbf{AB}) + \boldsymbol{b}^{\mathrm{T}}\mathbf{B}\boldsymbol{a} + \boldsymbol{b}^{\mathrm{T}}\mathbf{BAB}\boldsymbol{b} + a_0\right]. \quad \text{(A.4)}$$

The following two versions of a definite Gaussian integral were used in Chapter 6

$$\int_0^{\infty}\int_0^{\infty} \frac{dx\,dy}{2\pi\sqrt{1-\rho^2}} \exp\left[-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}\right] = \frac{1}{4}\left[1 + \frac{2}{\pi}\arcsin(\rho)\right], \quad \text{(A.5a)}$$

$$\int_{-\infty}^{\infty} Dz\ \mathrm{erf}\left(\frac{\alpha}{\sqrt{2}}z\right)\mathrm{erf}\left(\frac{\beta}{\sqrt{2}}z\right) = \frac{2}{\pi}\arcsin\left[\frac{\alpha\beta}{\sqrt{(1+\alpha^2)(1+\beta^2)}}\right]. \quad \text{(A.5b)}$$

Bibliography

Amari, S. (1997a). Natural gradient works efficiently in learning. RIKEN Preprint.

Amari, S. (1997b). Neural learning in structured parameter spaces - Natural Riemannian gadient. In M. C. Mozer, M. I. Jordan, and T. Petsche (Eds.), *Advances in Neural Information Processing Systems*, Volume 9, pp. 127–133. MIT Press.

Amari, S., N. Fujita, and S. Shinomoto (1992). Four types of learning curves. *Neural Computation 4*(4), 605–618.

Amari, S. and N. Murata (1993). Statistical theory of learning curves under entropic loss criterion. *Neural Computation 5*(1), 140–153.

Amit, D. J., M. R. Evans, H. Horner, and K. Y. M. Wong (1990). Retrieval phase diagrams for attractor neural networks with optimal interactions. *Journal of Physics A: Mathematical and General 23*, 3361–3381.

Anderson, J. A. and E. Rosenfeld (Eds.) (1988). *Neurocomputing: Foundations of Research*. Cambridge, MA: MIT Press.

Anlauf, J. K. and M. Biehl (1989). The AdaTron — an adaptive perceptron algorithm. *Europhysics Letters 10*(7), 687–692.

Ash, T. (1989). Dynamic node creation in backpropagation networks. *Connection Science 1*, 365–375.

Barber, D., D. Saad, and P. Sollich (1996). Finite size effects in on-line learning of multilayer neural networks. *Europhysics Letters 34*(2), 151–156.

Barkai, E. (1990). Correction. *Physical Review Letters 65*(25), 3210.

Barkai, E., D. Hansel, and I. Kanter (1990). Statistical mechanics of a multylayered neural network. *Physical Review Letters 65*(18), 2312–2315. See also (Barkai 1990).

Barkai, E., D. Hansel, and H. Sompolinsky (1992). Broken symmetries in multilayered perceptrons. *Physical Review A 45*, 4146–4161.

Barron, A. R. (1993). Universal approximation bounds for superposition of a sigmoidal function. *IEEE Transactions on Information Theory 39*, 930–945.

Bartlett, P. L. (1993). Vapnik-Chervonenkis dimension bounds for two-and three-layer networks. *Neural Computation 5*(3), 371–373.

Baum, E. B. and D. Haussler (1989). What size net gives valid generalization? *Neural Computation 1*(1), 151–160.

Biehl, M. and M. Opper (1991). Tilinglike learning in the parity machine. *Physical Review A 44*(10), 6888–6894.

Biehl, M. and M. Opper (1993). Construction algorithm for the parity-machine. *Physica A 193*(3–4), 307–313.

Biehl, M. and P. Riegler (1994). Online learning with a perceptron. *Europhysics Letters 28*(7), 525–530.

Biehl, M., P. Riegler, and C. Wöhler (1996). Transient dynamics of on-line learning in two-layered neural networks. *Journal of Physics A: Mathematical and General 29*(3), 4769–4780.

Biehl, M. and H. Schwarze (1992). Online learning of a time-dependent rule. *Europhysics Letters 20*(8), 733–738.

Biehl, M. and H. Schwarze (1995). Learning by online gradient descent. *Journal of Physics A: Mathematical and General 28*(3), 643–656.

Binder, K. and A. P. Young (1986). Spin glasses — experimental facts, theoretical concepts, and open questions. *Reviews of Modern Physics 58*(4), 801–976.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press.

Bishop, C. M., C. M. Roach, and M. G. von Hellermann (1993). Automatic analysis of JET charge exchange recombination spectra using neural networks. *Plasma Physics and Controlled Fusion 35*, 765–773.

Blumer, A., A. Ehrenfeucht, D. Haussler, and M. K. Warmuth (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery 36*(4), 929–965.

Bouten, M. (1994). Replica symmetry instability in perceptron models. *Journal of Physics A: Mathematical and General 27*, 6021–6023.

Bouten, M., J. Schietse, and C. Van den Broeck (1995). Gradient descent learning in perceptrons — a review of its possibilities. *Physical Review E 52*(2), 1958–1967.

Campbell, C. and C. J. Perez Vicente (1995). The Target Switch algorithm: A constructive learning procedure for feed-forward neural networks. *Neural Computation 7*(6), 1245–1264.

Chauvin, Y. (1989). A back-propagation algorithm with optimal use of hidden units. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems*, Volume 1, pp. 519–526. Morgan Kaufmann.

Cocco, S., R. Monasson, and R. Zecchina (1996). Analytical and numerical study of internal representations in multilayer neural networks with binary weights. *Physical Review E 54*(1), 717–725.

Coolen, A. C. C., S. N. Laughton, and D. Sherrington (1996). Dynamical replica theory for disordered spin systems. *Physical Review B 53*(13), 8184–8187.

Copelli, M. and N. Caticha (1995). On-line learning in the committee machine. *Journal of Physics A: Mathematical and General 28*, 1615–1625.

Copelli, M., O. Kinouchi, and N. Caticha (1996). Equivalence between learning in noisy perceptrons and tree commitee machines. *Physical Review E 53*, 6341–6352.

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers 14*, 326–334.

Cybenko, G. (1989). Approximation by superpositions of a sigmoid function. *Math. Control Signals and Systems 2*, 303–314. See also (Cybenko 1992).

Cybenko, G. (1992). Correction. *Math. Control Signals and Systems 5*, 455.

Derrida, B. (1994). Reply to the comment of M Bouten. *Journal of Physics A: Mathematical and General 27*, 6025.

Diederich, S. and M. Opper (1987). Learning of correlated patterns in spin-glass networks by local learning rules. *Physical Review Letters 58*(9), 949–952.

Drucker, H., C. Cortes, L. D. Jackel, Y. Le Cun, and V. Vapnik (1994). Boosting and other ensemble methods. *Neural Computation 6*(6), 1289–1301.

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

Engel, A., M. Bouten, A. Komoda, and R. Serneels (1990). Enlarged basin of attraction in neural networks with persistent stimuli. *Physical Review A 42*(8), 4998–5005.

Engel, A., H. English, and A. Schütte (1989). Improved retrieval in neural networks with external fields. *Europhysics Letters 8*(4), 393–397.

Engel, A., H. M. Köhler, F. Tschepke, H. Vollmayr, and A. Zippelius (1992). Storage capacity and learning algorithms for two-layer neural networks. *Physical Review A 45*, 7590–7609.

Engel, A. and M. Weigt (1996). Multifractal analysis of the coupling space of feedforward neural networks. *Physical Review E 53*(3), R2064–R2067.

Erichsen, J. R. and W. K. Thuemann (1993). Optimal storage of a neural network model: a replica symmetry-breaking solution. *Journal of Physics A: Mathematical and General 26*, L61–L68.

Fahlman, S. E. and C. Lebiere (1990). The cascade-correlation learning architecture. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems*, Volume 2, pp. 524–532. Morgan Kaufmann.

Fontanari, J. F. and R. Meir (1993). The statistical mechanics of the Ising perceptron. *Journal of Physics A: Mathematical and General 26*, 1077–1089.

Frean, M. (1990a). The Upstart algorithm: a method for constructing and training feed-forward neural networks. *Neural Computation 2*, 198–209.

Frean, M. (1992). A "thermal" perceptron learning rule. *Neural Computation 4*(6), 946–957.

Frean, P. (1990b). *Small nets and small Paths: Optimising neural computation.* Ph. D. thesis, Edinburgh University.

Freund, Y. (1995). Boosting a weak learining algorithm by majority. *Information and Computation 121*(2), 256–285.

Freund, Y. and R. E. Schapire (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Second European Conference, EuroCOLT '95*, Barcelona, pp. 23–27. Springer Verlag.

Fritzke, B. (1994). Growing cell structures — a self-organizing network for unsupervised and supervised learning. *Neural Networks 7*(9), 1441–1460.

Gallant, S. I. (1986a). Optimal linear discriminants. In *Proceedings of the Eighth IEEE International Conference on Pattern Recognition*, Volume 1, Washington, DC, pp. 849–852. IEEE Computer Society.

Gallant, S. I. (1986b). Three constructive algorithms for network learning. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Hillsdale, NJ, pp. 652–660. Lawrence Erlbaum.

Gallant, S. I. (1990). Perceptron-based learning algorithms. *IEEE Transactions on Neural Networks 1*, 179–191. See also (Gallant 1986b; Gallant 1986a).

Gardiner, C. W. (1983). *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, Volume 13 of *Springer series in Synergetics*. Berlin: Springer Verlag.

Gardner, E. (1988). The space of interactions in neural network models. *Journal of Physics A: Mathematical and General 21*, 257–270.

Gardner, E. (1989). Optimal basins of attraction in randomly sparse neural network models. *Journal of Physics A: Mathematical and General 22*, 1969–1974.

Gardner, E. and B. Derrida (1988). Optimal storage properties of neural network models. *Journal of Physics A: Mathematical and General 21*, 271–284. See also (Bouten 1994; Derrida 1994).

Geman, S., E. Bienenstock, and R. Doursat (1992). Neural networks and the bias/variance dilema. *Neural Computation 4*(1), 1–58.

Gerl, F. and U. Krey (1994). Storage capacity and optimal learning of Potts-model perceptrons by a cavity method. *Journal of Physics A: Mathematical and General 27*(22), 7353–7372.

Gerl, F. and U. Krey (1995). A Kuhn-Tucker cavity method for generalization with applications to perceptrons with Ising and Potts neurons. *Journal of Physics A: Mathematical and General 28*(23), 6501–6516.

Gerl, F. and U. Krey (1997). Replica symmetry breaking and the Kuhn-Tucker cavity method in simple and multilayer perceptrons. *Journal de Physique (Paris) I 7*(2), 303–327.

Griniasty, M. (1993). "Cavity-approach" analysis of the neural-network learning problem. *Physical Review E 47*(6), 4496–4513.

Griniasty, M. and H. Gutfreund (1991). Learning and retrieval in attractor neural networks above saturation. *Journal of Physics A: Mathematical and General 24*, 715–734.

Gross, D. J. and M. Mézard (1984). The simplest spin glass. *Nuclear Physics B 240*, 3057–3066. Reprinted in (Mézard, Parisi, and Virasoro 1987).

Grossman, T., R. Meir, and E. Domany (1989). Learning by choice of internal representations. *Complex Systems 2*, 555–575.

Gutfreund, H. and Y. Stein (1990). Capacity of neural networks with discrete couplings. *Journal of Physics A: Mathematical and General 23*, 2613–2630.

Györgyi, G. and P. Reimann (1997). Parisi phase in a neuron. *Physical Review Letters 79*(14), 2746–2749.

Hansen, L. K., R. Pathria, and P. Salamon (1993). Stochastic dynamics of supervised learning. *Journal of Physics A: Mathematical and General 26*(1), 63–71.

Hassibi, B. and D. G. Stork (1993). Second order derivatives for network pruning: optimal brain surgeon. In S. J. Hanson, J. D. Cowan, and C. L. Giles (Eds.), *Advances in Neural Information Processing Systems*, Volume 5, pp. 164–171. Morgan Kaufmann.

Hebb, D. O. (1949). *The Organization of Behavior*. New York: Wiley. Partially reprinted in (Anderson and Rosenfeld 1988).

Hertz, J., A. Krogh, and R. G. Palmer (1991). *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley.

Heskes, T. (1994). On Fokker-Planck approximations of on-line learning processes. *Journal of Physics A: Mathematical and General 27*, 5145–5160.

Heskes, T. and B. Kappen (1991). Learning processes in neural networks. *Physical Review A 44*, 2718–2762.

Hinton, G. E. and T. J. Sejnowski (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart and J. L. McClelland (Eds.), *Parallel Distributed Processing*, Volume 1, Chapter 7, pp. 282–317. Cambridge, MA: MIT Press.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA 79*, 2554–2558. Reprinted in (Mézard, Parisi, and Virasoro 1987; Anderson and Rosenfeld 1988).

Hopfield, J. J. (1984). Neurons with graded responses have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences, USA 81*, 3088–3092. Reprinted in (Anderson and Rosenfeld 1988).

Horner, H. (1992a). Dynamics of learning and generalization in a binary perceptron model. *Zeitschrift für Physik B: Condensed Matters 87*(3), 371–376.

Horner, H. (1992b). Dynamics of learning for the binary perceptron problem. *Zeitschrift für Physik B: Condensed Matters 86*(2), 291–308.

Horner, H. (1993). Dynamics of learning and generalization in perceptrons with constraints. *Physica A 200*(1–4), 552–562.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks 4*, 251–257.

Kearns, M. J. and U. V. Vazirani (1994). *An Introduction to Computational Learning Theory*. Cambridge, MA: MIT Press.

Kepler, T. B. and L. F. Abbott (1988). Domains of attraction in neural networks. *Journal de Physique (Paris) 49*, 1657–1662.

Kim, J. W. and H. Sompolinsky (1996). Online Gibbs learning. *Physical Review Letters 76*, 3021–3024.

Kim, Y. K. and J. B. Ra (1991). Weight value initialization for improving training speed in the backpropagation network. In *International Joint Conference on Neural Networks*, pp. 2396–2401. IEEE Press.

Kinouchi, O. and N. Caticha (1992). Optimal generalization in perceptrons. *Journal of Physics A: Mathematical and General 25*, 6243–6250.

Kinouchi, O. and N. Caticha (1995). Online versus off-line learning in the linear perceptron — a comparative study. *Physical Review E 52*, 2878–2886.

Kinzel, W. and M. Opper (1991). Dynamics of learning. In E. Domany, J. L. van Hemmen, and K. Schulten (Eds.), *Models of Neural Networks*, Volume 1 of *Physics of Neural Networks*, Chapter 4, pp. 149–171. Berlin: Springer Verlag.

Krauth, W. and M. Mézard (1987). Learning algorithms with optimal stability in neural networks. *Journal of Physics A: Mathematical and General 20*(11), L745–L752.

Krauth, W. and M. Mézard (1989). Storage capacity of memory networks with binary couplings. *Journal de Physique (Paris) 20*, 3057–3066.

Krogh, A. and J. A. Hertz (1992). Generalization in a linear perceptron in the presence of noise. *Journal of Physics A: Mathematical and General 25*, 1135–1147.

Le Cun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation 1*(4), 541–551.

Le Cun, Y., J. Denker, and S. A. Solla (1990). Optimal brain damage. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems*, Volume 2, pp. 477–484. Morgan Kaufmann.

Le Cun, Y., I. Kanter, and S. A. Solla (1991). Second order properties of error surfaces: Learning time and generalization. In R. P. Lippann, J. E. Moody, and D. S. Touretzky (Eds.), *Advances in Neural Information Processing Systems*, Volume 3, pp. 918–924. Morgan Kaufmann.

Leen, T. K. and J. Moody (1993). Weight space probability densities in stochastic learning: I. Dynamics and equilibria. In S. J. Hanson, J. D. Cowan, and C. L. Giles (Eds.), *Advances in Neural Information Processing Systems*, Volume 5, pp. 451–458. Morgan Kaufmann.

Leen, T. K. and G. B. Orr (1994). Optimal stochastic search and adaptive momentum. In J. D. Cowan, G. Tesauro, and J. Alspector (Eds.), *Advances in Neural Information Processing Systems*, Volume 6, pp. 477–484. Morgan Kaufmann.

Lehtokangas, M., J. Saarinen, K. Kaski, and P. Huuhtanen (1995). Initializing weights of a multilayer perceptron network by using the orthogonal least squares algorithm. *Neural Computation 7*(5), 982–999.

Levin, A. U., T. K. Leen, and J. E. Moody (1994). Fast pruning using principal components. In J. D. Cowan, G. Tesauro, and J. Alspector (Eds.), *Advances in Neural Information Processing Systems*, Volume 6, pp. 35–42. Morgan Kaufmann.

Levin, E., N. Tishby, and S. A. Solla (1990). A statistical approach to learning and generalization in layered neural networks. *IEEE Transactions on Pattern Analysis and Machine In 78*(10), 1568–1574.

Little, W. A. (1974). The existence of persistent states in the brain. *Mathematical Biosciences 19*, 101–120.

Littmann, E. and H. Ritter (1996). Learning and generalization in cascade network archictectures. *Neural Computation 8*(7), 1521–1539.

Maass, W. (1994). Neural nets with superlinear VC-dimension. *Neural Computation 6*(5), 877–884.

Mace, C. W. H. and A. C. C. Coolen (1997). Statistical mechanical analysis of the dynamics of learning in perceptrons. Preprint KCL-MTH-97-36, King's College London, London. To appear in *Statistics and Computing*, available from the homepage http://www.mth.kcl.ac.uk/research/staff/acc_coolen.html.

Majer, P., A. Engel, and A. Zippelius (1993). Perceptrons above saturation. *Journal of Physics A: Mathematical and General 26*, 7405–7416.

Malzahn, D., A. Engel, and I. Kanter (1997). Storage capacity of correlated perceptrons. *Physical Review E 55*(6), 7369–7378.

Marchand, M. and M. Golea (1993). On learning simple neural concepts — from half-space intersections to neural decision lists. *Network 4*(1), 67–85.

Marchand, M., M. Golea, and P. Rujan (1990). A convergence theorem for sequential learning in two-layer perceptrons. *Europhysics Letters 11*(6), 487–492.

Martinez, D. and D. Estève (1992). The Offset algorithm — building and learning-method for multilayer neural networks. *Europhysics Letters 18*(2), 95–100.

McCulloch, W. S. and W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics 5*, 115–133. Reprinted in (Anderson and Rosenfeld 1988).

Mézard, M. (1989). The space of interactions in neural networks — Gardner computation with the cavity method. *Journal of Physics A: Mathematical and General 22*(12), 2181–2190.

Mézard, M. and J.-P. Nadal (1989). Learning in feed-forward layered networks: the tiling algorithm. *Journal of Physics A: Mathematical and General 22*, 2191–2203.

Mézard, M., G. Parisi, and M. G. Virasoro (1987). *Spin Glass Theory and Beyond.* Singapore: World Scientific.

Minsky, M. L. and S. A. Papert (1969). *Perceptrons.* Cambridge, MA: MIT Press. Expanded Edition 1990.

Mitchison, G. J. and R. M. Durbin (1989). Bounds on the learning capacity of some multi-layer networks. *Biological Cybernetics 60*, 345–356.

Monasson, R. and D. O'Kane (1994). Domains of solutions and replica symmetry-breaking in multilayer neural networks. *Europhysics Letters 27*(2), 85–90.

Monasson, R. and R. Zecchina (1995). Weight space structure and internal representations: a direct approach to learning and generalization in multilayer neural networks. *Physical Review Letters 75*, 2432–2435.

Monasson, R. and R. Zecchina (1996). Learning and generalization theories of large committee machines. *Modern Physics Letters B 9*, 1887–1897.

Mozer, M. C. and P. Smolensky (1989). Skeletonization: a technique for trimming the fat from a network via relevance assessment. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems*, Volume 1, pp. 107–115. Morgan Kaufmann.

Nadal, J.-P. (1989). Study of a growth algorithm for a feed-forward network. *International Journal of Neural Systems 1*, 55–59.

Neal, R. M. (1992). Bayesian training of backpropagation networks by the hybrid Monte Carlo method. Technical Report CRG-TR-92-1, Department of Computer Science, University of Toronto, Canada.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, Volume 118 of *Lecture Notes in Statistics*. New York: Springer Verlag.

Nguyen, D. and B. Widrow (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *International Joint Conference on Neural Networks*, pp. C21–C26. IEEE Press.

Nowlan, S. J. and G. E. Hinton (1992). Simplifying neural networks by soft weight sharing. *Neural Computation 4*(4), 473–493.

Opper, M. (1994). Learning and generalization in a two-layer neural network — the role of the Vapnik-Chervonenkis dimension. *Physical Review Letters 74*(13), 2113–2116.

Opper, M. (1996). Online versus off-line learning from random examples — general results. *Physical Review Letters 77*(22), 4671–4674.

Opper, M. and O. Winther (1996). Mean-field approach to Bayes learning in feed-forward neural networks. *Physical Review Letters 76*(11), 1964–1967.

Orr, G. B. and T. K. Leen (1993). Weight space probability densities in stochastic learning: II. Transients and basin hopping times. In S. J. Hanson, J. D. Cowan, and C. L. Giles (Eds.), *Advances in Neural Information Processing Systems*, Volume 5, pp. 507–514. Morgan Kaufmann.

Prügel-Bennett, A. (1996). On-line learning with momentum. Unpublished.

Radons, G. (1993). On stochastic dynamics of supervised learning. *Journal of Physics A: Mathematical and General 26*(14), 3455–3461.

Rattray, M. (1997). Private communication.

Rattray, M. and D. Saad (1997a). Globally optimal learning rates for multi-layer neural networks. Submitted.

Rattray, M. and D. Saad (1997b). Globally optimal on-line learning rules. In *Proceedings of the 10th Annual Conference of Neural Information Processing Systems, 1997*, Denver, CO. MIT Press. To be published.

Rau, A. and D. Sherrington (1990). Retrieval enhancement due to external stimuli in dilute neural networks. *Europhysics Letters 11*(6), 499–504.

Rau, A., D. Sherrington, and K. Y. M. Wong (1991). External fields in attractor neural networks with different learning rules. *Journal of Physics A: Mathematical and General 24*(1), 313–326.

Riegler, P. (1997). *Dynamics of On-line Learning in Neural Networks.* Ph. D. thesis, Bayerische Julius-Maximilians-Universität Würzburg.

Riegler, P. and M. Biehl (1995). Online backpropagation in 2-layered neural networks. *Journal of Physics A: Mathematical and General 28*, L507–L513.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge: Cambridge University Press.

Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Washington DC: Spartan.

Rumelhart, D. E., R. Durbin, R. Golden, and Y. Chauvin (1995). Backpropagation: the basic theory. In Y. Chauvin and D. E. Rumelhart (Eds.), *Backpropagation: Theory, Architectures, and Applications*, pp. 1–34. Hillsdale, NJ: Lawrence Erlbaum.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986b). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1: Foundations, pp. 318–362. Cambridge, MA: MIT Press. Reprinted in (Anderson and Rosenfeld 1988).

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986a). Learning representations by back-propagating errors. *Nature 323*, 533–536. See also (Rumelhart, Hinton, and Williams 1986b; Rumelhart, Durbin, Golden, and Chauvin 1995).

Saad, D. (1994). Explicit symmetries and the capacity of multilayer neural networks. *Journal of Physics A: Mathematical and General 27*, 2719–2734.

Saad, D. and M. Rattray (1997a). Globally optimal learning rates for multilayer neural networks. In *Proceedings of the Minerva Workshop on Mesoscopics, Fractals and Neural Networks*, Eilat, Israel.

Saad, D. and M. Rattray (1997b). Globally optimal parameters for on-line learning in multilayer neural networks. *Physical Review Letters 79*(13), 2578–2581.

Saad, D. and S. A. Solla (1995a). Exact solution for online learning in multilayer neural networks. *Physical Review Letters 74*, 4337–4340.

Saad, D. and S. A. Solla (1995b). Online learning in soft committee machines. *Physical Review E 52*(4), 4225–4243.

Saad, D. and S. A. Solla (1996). Dynamics of on-line gradient descent learning for multilayer neural networks. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems*, Volume 8, pp. 302–308. MIT Press.

Saad, D. and S. A. Solla (1997). Learning with noise and regularizers in multilayer neural networks. In M. C. Mozer, M. I. Jordan, and T. Petsche (Eds.), *Advances in Neural Information Processing Systems*, Volume 9, pp. 260–266. MIT Press.

Sakurai, A. (1995). On the VC-dimension of depth-4 threshold circuits and the complexity of boolean-valued functions. *Theoretical Computer Science 137*(1), 109–127.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning 5*, 197–227.

Schwarze, H. (1993). Learning a rule in a multilayer neural-network. *Journal of Physics A: Mathematical and General 26*(21), 5781–5794.

Schwarze, H. and J. Hertz (1992). Generalization in a large committee machine. *Europhysics Letters 20*(4), 375–380.

Setiono, R. (1997). A penalty-function approach for pruning feedforward neural networks. *Neural Computation 9*(1), 185–204.

Seung, H. S., H. Sompolinsky, and N. Tishby (1992). Statistical mechanics of learning from examples. *Physical Review A 45*, 6056–91.

Sherrington, D. and S. Kirkpatrick (1975). Solvable model of a spin-glass. *Physical Review Letters 35*(26), 1792–1796. Reprinted in (Mézard, Parisi, and Virasoro 1987).

Sollich, P. (1994). Finite size effects in learning and generalization in linear perceptrons. *Journal of Physics A: Mathematical and General 27*, 7771–7784.

Sollich, P. and D. Barber (1997b). On-line learning from finite training sets. *Europhysics Letters 38*, 477–482.

Sollich, P. and D. Barber (1997a). On-line learning from finite training sets: An analytical case study. In M. C. Mozer, M. I. Jordan, and T. Petsche (Eds.), *Advances in Neural Information Processing Systems*, Volume 9, pp. 274–280. MIT Press.

Sollich, P. and D. Barber (1997c). On-line learning from finite training sets in nonlinear networks. In *Proceedings of the 10th Annual Conference of Neural Information Processing Systems, 1997*, Denver, CO. MIT Press. To be published.

Sollich, P. and D. Barber (1997d). Online learning from finite training sets and robustness to input bias. Submitted to Neural Computation.

Tesauro, G. (1990). Neurogammon wins computer olympiad. *Neural Computation 1*, 321–323.

Thouless, D. J., P. W. Anderson, and R. G. Palmer (1977). Solution of 'solvable model of a spin glass'. *Philosophical Magazine 35*(3), 593–601. Reprinted in (Mézard, Parisi, and Virasoro 1987).

Tishby, N., E. Levin, and S. Solla (1989). Consistent inference of probabilities in layered networks: Predictions and generalization. In *International Joint Conference on Neural Networks*, Volume 2, Washington 1989, pp. 403–409. IEEE, New York.

Urbanczik, R. (1995). A fully connected committee machine learning unrealizable rules. *Journal of Physics A: Mathematical and General 28*(24), 7097–7104.

Urbanczik, R. (1997). Storage capacity of the fully-connected committee machine. *Journal of Physics A: Mathematical and General 30*, L387–L392.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM 27*, 1134–1142.

van Ooyen, A. and B. Nienhuis (1992). Improving the convergence of the back-propagation algorithm. *Neural Networks 5*(3), 465–471.

Vapnik, V. N. (1982). *Estimation of Dependences Based on Empirical Data.* New York: Springer Verlag.

Vapnik, V. N. and A. Y. Chervonenkis (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory Prob. Appl. 16*(2), 264–280.

Watkin, T. L. H., A. Rau, and M. Biehl (1993). The statistical mechanics of learning a rule. *Reviews of Modern Physics 65*, 499–556.

Weigend, A. S., D. E. Rumelhart, and B. A. Huberman (1990). Back-propagation, weight-elimination and time series prediction. In D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton (Eds.), *Proceedings of the 1990 Connectionist Models Summer School*, San Mateo, CA, pp. 105–116. Morgan Kaufmann.

Weigt, M. (1997, July). Multifractal analysis of perceptron learning with errors. Preprint cond-mat/9707178, Universität Magdeburg, Germany. Available via condensed matter repository or from the homepage http://rhea.nat.uni-magdeburg.de/~itpl/papers.html.

Weigt, M. and A. Engel (1997). Multifractality and percolation in the coupling space of perceptrons. *Physical Review E 55*(4), 4552–4561.

Wendemuth, A. (1994a). Generalization ability of optimal cluster separation networks. *Journal of Physics A: Mathematical and General 27*(7), 2325–2333.

Wendemuth, A. (1994b). Training of optimal cluster separation networks. *Journal of Physics A: Mathematical and General 27*(11), L387–L390.

Wendemuth, A. (1995a). Learning the unlearnable. *Journal of Physics A: Mathematical and General 28*(18), 5423–5436. See also (Wendemuth 1995b).

Wendemuth, A. (1995b). Performance of robust training algorithms for neural networks. *Journal of Physics A: Mathematical and General 28*(19), 5485–5493. See also (Wendemuth 1995a).

Wendemuth, A. (1995c). Stabilities in optimal cluster separation networks. *Neural Networks 8*(3), 387–390.

Wendemuth, A. (1995d). Storage capacity bounds in multilayer neural networks. *International Journal of Neural Systems 5*(3), 217–228.

Wendemuth, A., M. Opper, and W. Kinzel (1993). The effect of correlations in neural networks. *Journal of Physics A: Mathematical and General 26*, 3165–3185.

Werbos, P. J. (1974). *Beyond regression: new tools for prediction and analysis in the behavioural sciences.* Ph. D. thesis, Harvard University, Cambridge, MA.

West, A. H. L. and D. Saad (1997). The capacity of the Upstart algorithm. In S. W. Ellacott, J. C. Mason, and I. J. Anderson (Eds.), *Mathematics of Neural Networks: Models, Algorithms and Applications*, Volume 8 of *Operations Research/Computer Science Interfaces*, Chapter 65, pp. 372–377. Boston: Kluwer Academic Publishers.

West, A. H. L., D. Saad, and I. T. Nabney (1997). The learning dynamcis of a universal approximator. In M. C. Mozer, M. I. Jordan, and T. Petsche (Eds.), *Advances in Neural Information Processing Systems*, Volume 9, pp. 288–294. MIT Press.

Whyte, W. and D. Sherrington (1996). Replica-symmetry breaking in perceptrons. *Journal of Physics A: Mathematical and General 29*, 3063–3073.

Wolpert, D. H. (1995). Off-training set error and a priori distinctions between learning algorithms. Technical Report SFI TR 95-01-003, Santa Fe Institute, Santa Fe.

Wong, K. Y. M. (1995). Microscopic equations and stability conditions in optimal neural networks. *Europhysics Letters 30*(4), 245–250.

Wong, K. Y. M. (1997). Exact dynamics in feedforward neural networks. *Europhysics Letters 38*(8), 631–636.

Xiong, Y., J.-H. Oh, and C. Kwon (1997). Weight space structure and the storage capacity of a fully connected committee machine. *Physical Review E 56*(4), 4540–4544.

Yau, H. W. and D. J. Wallace (1991). Enlarging the attractor basins of neural networks with noisy external fields. *Journal of Physics A: Mathematical and General 24*(23), 5639–5650.

Zollner, R., H. J. Schmitz, F. Wunsch, and U. Krey (1992). Fast generating algorithm. *Neural Networks 5*(5), 771–777.