Asking Intelligent Questions — The Statistical Mechanics of Query Learning

Peter Sollich

Doctor of Philosophy
University of Edinburgh
1995

Declaration

This thesis has been composed by myself and it has not been submitted in any previous application for a degree. The work reported within was executed by me, unless otherwise stated.

Acknowledgments

I would first of all like to thank my supervisors: David Wallace got me off to a good start in the first year, squeezing enormously productive supervision meetings into the tiniest gaps in his schedule, and has since supported me from afar. David Saad took over his duties during the second year; with lots of encouragement and advice, he helped me to pull it all through, reading innumerable drafts of articles along the way (and more recently even foregoing the pleasures of an afternoon on the beaches of Corsica in fulfillment of more than his supervisory duties). David Willshaw, finally, provided helpful comments from a Cognitive Science perspective and gave me the opportunity to present my work at the Parallel Distributed Processing workshops; access to his personal library also made last-minute literature searches a lot easier. I am very grateful to all three of them for their unceasing help, support and advice.

My work has benefited greatly from discussions with Chris Bishop, Alastair Bruce, David Cohn, Ton Coolen, Michael Kearns, Ronny Meir, Gerhard Paass, Mark Plutowski, Jonathan Shapiro, and Sara Solla,; I would like to thank all of them. Special thanks are due to Anders Krogh for an exciting and fun collaboration at NORDITA, Copenhagen. I would also like to acknowledge financial support by the Faculty of Science and Engineering through a Ph.D. studentship, and by the European Union through travel support under grant no. ERB CHRX-CT92-0063.

Thanks are due to everyone in the Neural Networks group; I very much enjoyed the collaborative and productive yet relaxed work environment. In particular, I would like to thank my office-and-next-door-mates and friends Ansgar, Dave and Glenn, who accompanied me through the ups and downs of doing a Ph.D., providing a reassuring atmosphere of 'we're all in it together' (and never letting the opportunity for a good pun slip). Without Dave, who never declined to read, reread or proofread anything from 'stream of consciousness' drafts to research reports to conference posters, who knows how many more out-of-place commas and germanically drawn out sentences this thesis would contain. Thanks a lot, Dave!

Finally, I am very grateful to my family for unflagging moral (and financial) support and encouragement. And last but by no means least, I would like to thank all my friends, be they in Edinburgh or elsewhere, for making the last three years a thoroughly enjoyable experience, and Jennie for keeping me happy and simply being my Bold.

Publications

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

- P Sollich. Query construction, entropy, and generalization in neural network models. *Physical Review E*, 49:4637–4651, 1994.
- P Sollich. Finite-size effects in learning and generalization in linear perceptrons. *Journal of Physics A*, 27:7771–7784, 1994.
- P Sollich. Learning unrealizable tasks from minimum entropy queries. *Journal of Physics A*. In press.
- P Sollich. Learning from minimum entropy queries in a large committee machine. Submitted to *Physical Review Letters*.
- P Sollich. Minimum entropy queries for linear students learning nonlinear rules. In Verleysen M, editor, *Third European Symposium on Artificial Neural Networks (ESANN'95), Proceedings*, pages 217–222, Brussels, 1995. D facto.
- P Sollich and D Saad. Learning from queries for maximum information gain in imperfectly learnable problems. In G Tesauro, D S Touretzky, and T K Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 287–294, Cambridge, MA, 1995. MIT Press.
- P Sollich. Learning in large linear perceptrons and why the thermodynamic limit is relevant to the real world. In G Tesauro, D S Touretzky, and T K Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 207–214, Cambridge, MA, 1995. MIT Press.
- P Sollich. Query learning for maximum information gain in a multi-layer neural network. *Annals of Mathematics and Artificial Intelligence*. In press.

Abstract

This thesis analyses the capabilities and limitations of query learning by using the tools of statistical mechanics to study learning in feed-forward neural networks.

In supervised learning, one of the central questions is the issue of generalization: Given a set of training examples in the form of input-output pairs produced by an unknown *teacher* rule, how can one generate a *student* which *generalizes*, i.e., which correctly predicts the outputs corresponding to inputs not contained in the training set? The traditional paradigm has been to study learning from *random examples*, where training inputs are sampled randomly from some given distribution. However, random examples contain redundant information, and generalization performance can thus be improved by *query learning*, where training inputs are chosen such that each new training example will be maximally 'useful' as measured by a given *objective function*.

We examine two common kinds of queries, chosen to optimize the objective functions, generalization error and entropy (or information), respectively. Within an extended Bayesian framework, we use the techniques of statistical mechanics to analyse the average case generalization performance achieved by such queries in a range of learning scenarios, in which the functional forms of student and teacher are inspired by models of neural networks. In particular, we study how the efficacy of query learning depends on the form of teacher and student, on the training algorithm used to generate students, and on the objective function used to select queries. The learning scenarios considered are simple but sufficiently generic to allow general conclusions to be drawn.

We first study perfectly learnable problems, where the student can reproduce the teacher exactly. From an analysis of two simple model systems, the high-low game and the linear perceptron, we conclude that query learning is much less effective for rules with continuous outputs – provided they are 'invertible' in the sense that they can essentially be learned from a finite number of training examples – than for rules with discrete outputs. Queries chosen to minimize the entropy generally achieve generalization performance close to the theoretical optimum afforded by minimum generalization error queries, but can perform worse than random examples in scenarios where the

training algorithm is under-regularized, i.e., has too much 'confidence' in corrupted training data.

For imperfectly learnable problems, we first consider linear students learning from nonlinear perceptron teachers and show that in this case the structure of the student space determines the efficacy of queries chosen to minimize the entropy in *student* space. Minimum *teacher* space queries, on the other hand, perform worse than random examples due to lack of feedback about the progress of the student. For students with discrete outputs, we find that in the absence of information about the teacher space, query learning can lead to self-confirming hypotheses far from the truth, misleading the student to such an extent that it will not approximate the teacher optimally even for an infinite number of training examples. We investigate how this problem depends on the nature of the noise process corrupting the training data, and demonstrate that it can be alleviated by combining query learning with Bayesian techniques of model selection. Finally, we assess which of our conclusions carry over to more realistic neural networks, by calculating finite size corrections to the thermodynamic limit results and by analysing query learning in a simple two-layer neural network. The results suggest that the statistical mechanics analysis is often relevant to real-world learning problems, and that the potentially significant improvements in generalization performance achieved by query learning can be made available, in a computationally cheap manner, for realistic multi-layer neural networks.

Contents

Chapter 1

Introduction

1.1 Motivation

This thesis deals with the general problem of learning a rule from examples. This is a task which humans normally perform with relative ease, but which is much more difficult to accomplish for conventional computers, robots and other non-human learners. Learning from examples is necessary when a rule is either unknown or too complicated to be specified explicitly and encoded into, for example, an expert system. A few examples of situations of this kind are: speech and image recognition, prediction of chaotic time series, forecasting, control, prediction of protein secondary structure, robot navigation; the list could be extended almost indefinitely. We can think of many of these scenarios in terms of a mapping from certain inputs (such as a speech signal, a digitized image, the previous values of a time series, the amino acid sequence of a protein etc.) to the corresponding outputs (for example a phoneme sequence, a string of characters, the next value of a time series, the secondary structure of a protein etc.), which one is trying to infer on the basis of a number of known input-output pairs. This kind of problem has been investigated in a number of different disciplines, each bringing its own emphasis and terminology to the subject, being called machine learning in computer science, regression, classification, interpolation and extrapolation in statistics, pattern recognition in engineering, etc. In the last decade or so, important contributions to the field have also been made by researchers interested in (artificial) neural networks. The distinguishing feature of the neural networks approach is its inspiration from biological networks of interconnected neurons. Artificial neural networks are extremely abstract version of such networks, consisting of a collection of neurons or elementary processors which perform some simple computation (such as thresholding, or application of a nonlinear transfer function) on the input they receive, and

feed the resulting output into the neurons to which they are connected. In this context, the problem of learning a rule became known as *supervised learning.* The desired input-output mappings were most frequently modelled by feed-forward networks with a layered structure, which can approximate a large class of input-output mappings (see, e.g., [HSW89]) and can indeed be viewed as a particular class of nonlinear statistical models (see, e.g., [Whi89]). Supervised learning in such feed-forward networks will be the focus of this thesis. However, the framework in which we investigate our topic will also draw on concepts from machine learning and in particular from statistics in order to provide a wider perspective, and we shall frequently relate our results to research in these disciplines.

In the language of supervised learning, the problem that we are concerned with can be stated as follows: One is given a set of *training examples* (input-output pairs) produced by a *teacher* according to some underlying but unknown *rule.* The aim is to generate, using a suitable *learning* or *training algorithm*, a *student* (e.g., a neural network) which *generalizes*, i.e., which can predict at least approximately the outputs corresponding to previously unseen inputs. The traditional approach has been to study generalization from *random examples*, where the input-output pairs which make up the training set are obtained by picking at random an input value according to some probability distribution, 'labelling' it with the corresponding output from the teacher and then possibly corrupting one or both of these values with some noise. It is clear, however, that the amount of novel information contained in random examples decreases towards zero as learning proceeds. This is due to the fact that as the size of the training set increases, the student becomes increasingly confident about its predictions in large regions of input space; random examples falling into these regions are essentially redundant and yield only a negligible improvement in generalization performance. The alternative approach of *query learning* (also referred to as active learning, active sampling, active data selection, experimental design etc.) has therefore attracted considerable interest. Here the inputs are not chosen at random, but rather by a *query selection algorithm* which, depending on the previously seen examples, selects an input value for the next input-output pair to be added to the training set. The motivation for query learning is that eliminating the redundancy contained in random examples should improve generalization performance or, equivalently, reduce the number of training examples needed to attain a certain level of generalization performance. In the context of human learning, one could loosely describe query learning as giving a student the possibility to ask non-redundant, 'intelligent' questions, rather than presenting her with randomly selected pieces of information.

Query learning makes most sense in situations where obtaining training outputs is

in some sense 'expensive'. For example, the teacher output could actually be the result of a complicated physical measurement (with control parameters given by the inputs). Or it might be that training outputs can only be obtained from a human expert, as is often the case in tasks like phoneme recognition, medical diagnosis of X-ray images etc. A further motivation for query learning is that the cost of training, i.e., producing a suitable student on the basis of the training set, can increase strongly with the number of training examples. Finally, it is often undesirable to store large amounts of training data, and one might like to restrict attention to the examples which are most relevant for learning a given rule.

The main question that we will try to answer in this thesis is: How useful is query learning? Quite apart from its independent theoretical interest, this question has significant implications for practical applications. The reason for this is that the improvements in generalization performance afforded by query learning do of course have their price in so far as a query selection algorithm can itself be computationally expensive, possibly off-setting the savings due to the reduced number of training examples that are needed. In order to decide whether to use query learning in a certain application or not, it is therefore of paramount importance to know what performance gains can be expected from query learning, in order to be able to assess the trade-off with the costs of query selection itself. Our approach to this problem will be explained in more detail below; first, however, we review some of the existing research in the area of query learning to put our work into its proper context.

1.2 Review of existing work

We will concentrate on the analysis of query learning in scenarios where training outputs are expensive. That is, we assume that the aim is to achieve a certain generalization performance (as measured by the generalization error, which is formally defined in Chapter 2) while keeping the number of training outputs that need to be obtained from the teacher at a minimum. This excludes situations where training data is in principle abundant and one is concerned with identifying a minimal subset of a given larger training set which will yield the desired generalization performance. For references on this approach, sometimes referred to as 'active (subset) selection', we refer to [PW93]; a more extensive list can be found in [Plu94]. We will also confine our analysis to training algorithms which generate a student on the basis of the whole training set ('batch' or 'off-line' learning). The alternative approach of 'on-line' learning, which modifies students only on the basis of the most recent training example, would be inappropriately wasteful when training data is expensive. Techniques closely related to

query learning have nevertheless been used to optimize the behaviour of on-line learning algorithms by selecting a particular order in which training examples are presented (see, e.g., [Cac94, Mun92] and [SR95] for a more extensive review from the point of view of cognitive science), but this is an issue that will not be addressed in this thesis. Finally, the use of active learning in unsupervised learning tasks such as clustering (see, for example, [HB95, CRL94]) is also beyond the scope of our work.

Having excluded the cases with which we will not be concerned, we define query learning as follows: The 'intelligent' selection of new training inputs (or 'queries') on the basis of the existing training set, *before* the corresponding new training outputs are known[1]. Note that we regard data gathering (i.e., query selection) and inference from the data (as defined by a learning or training algorithm) as distinct processes, which could be implemented separately[2]. In the context of human learning, for example, our point of view would allow queries to be selected by an independent 'adviser' different from the student and the teacher. The selection of queries is in general a sequential process in the sense that the choice of each query can be made only once the outputs corresponding to 'earlier' training inputs are known. However, we shall also discuss cases in which each new training input depends only on the previous inputs. A sequence of training inputs can then be determined before any outputs are received, and query selection becomes what we will term 'effectively non-sequential'.

A first classification of existing work on query learning is afforded by the criterion of whether queries are chosen heuristically or, alternatively, in a principled way, defined by the requirement of optimizing a given *objective function*. Another distinction can be made between approaches where queries are *constructed*, i.e., calculated (maybe stochastically) on the basis of the existing training set, or *filtered* from a stream of random input values until some criterion is met.

Heuristic query construction algorithms for two-layer feed-forward networks (with one layer of hidden units and one layer of output units) were proposed by, for example, Baum [Bau91, BL91] and Hwang *et al.* [HCOM91]. Baum considered a student network with binary threshold units (which output either 0 or 1) and suggested an algorithm for identifying hyperplanes in input space which separate examples with training output 0

[1] Formally, the condition that training inputs should be selected before their corresponding outputs are known ensures that our inferences are not biased by the way we gather training data; see Section 2.6. It is this condition that excludes most techniques for active selection of subsets training data from our discussion, since these normally base the selection of training examples on both the new input and output (see, e.g., [PW93, Plu94]). However, our approach can be straightforwardly extended to include methods for subset selection which choose new training examples only on the basis of their inputs.

[2] In the machine learning literature, data gathering and inference are normally viewed as forming a single 'learning algorithm'.

from examples with output 1; these hyperplanes determine the weights of connections between inputs and hidden units. Baum analysed the algorithm within the framework of probably approximately correct (PAC) learning and proved that if the teacher was defined by a network of the type considered with no more than four hidden units, then the desired mapping can be learned in polynomial time. This contrasts with the case of random examples, where the problem of learning such a network is NP complete [BR88]. Experiments on more complicated teacher rules also appeared to yield promising results. Extensions to situations with a noisy teacher or with student networks with continuous rather than threshold units were suggested, but not discussed in detail.

Hwang *et al.* [HCOM91] considered query learning starting from a partially trained two-layer network with sigmoidal transfer functions at the hidden and output units. The rule to be learned was a binary classification task, with outputs 0 and 1 corresponding to the two classes. Hwang *et al.* considered the heuristic of selecting training examples near the decision boundary of the student network, defined as the input region for which the predicted output was 0.5; this region was identified by an approximate inversion of the mapping realized by the student network. Their approach was critized by MacKay [Mac92c], who argued that there is no guarantee that inputs near the currently learned decision boundary will necessarily be very informative (in particular after a large number of queries have been selected), and pointed out that the algorithm runs the risk of not identifying decision boundaries that have not been found after training on the initial set of random examples.

For the simpler case of binary perceptron students and teachers (i.e., single layer networks with a threshold output unit), Kinzel and Ruján [KR90] and Watkin and Rau [WR92] analysed query construction based on an approach similar to the one in [HCOM91], using a statistical mechanics approach. The decision boundary of a binary perceptron is the hyperplane orthogonal to its weight vector, and thus queries were selected at each step orthogonally to the current weight vector. For batch learning, the average generalization error was found to decay exponentially with the number of training examples, which constitutes a significant improvement over the case of random examples. The improvement for a simple on-line learning algorithm was found to be less drastic, as expected. The above heuristic query construction algorithm was justified by Kinouchi and Caticha [KC92] through explicit minimization of the generalization error in an on-line learning scenario, and an exponential decay of the generalization error was found for a suitably optimized, though slightly unrealistic, learning algorithm. In the context of on-line learning of a time-varying rule, heuristic query construction for binary perceptron students was also studied by Biehl and Schwarze [BS93], and a

moderate performance improvement was found.

Cohn [Coh90, CAL90, CAL94] first proposed the method of *query filtering.* He studied concept learning (where there are only two different target outputs, 0 and 1; this is one of the most widely studied scenarios in machine learning) and argued that queries should be selected from what he called the 'region of uncertainty', where at least two students which are compatible with all the existing training examples make different predictions. He suggested ways of achieving this by training two students on the same training set but with different additional constraints which would make them maximally different (one 'most specific', the other 'most general'). He also demonstrated explicitly the drawbacks of the approach in [HCOM91] that were discussed above.

In an approach similar to Cohn's heuristic 'region of uncertainty', Seung *et al.* introduced the 'query by committee' algorithm for filtering queries [SOS92]. This algorithm relies on training a committee of several students, and queries are selected according to the principle of 'maximal disagreement' between the committee members. They justified their algorithm within an objective function approach to query selection by showing that as the number of committee members grows large, the query by committee algorithm maximizes the expected information gain from a new training example. A worst-case analysis of query by committee within the framework of PAC learning was given by Freund *et al.* [FSST93, Fre93]. For concept learning problems in which the student has exactly the same form as the teacher and the training examples are free of noise, they managed to relate the information gain to the decrease of the generalization error. In particular, they showed that an exponential decay of the generalization error is obtained if the information gain for queries selected by query by committee does not decay to zero as the number of training examples increases, and they identified an interesting class of learning scenarios for which this is the case.

At this point, let us mention some related research on learning from queries in the field of machine learning[3]. A lot of work in this area has focused on exact concept identification (see, e.g., [Val84, Ang88, MT92]), corresponding to perfect generalization, and on the *computational complexity* of query learning (see, e.g., [BEH89] and [AK95] and references therein). Of more interest to us are studies relating to the *sample complexity* of query learning [BEH89], which in our terminology is the number of training examples needed to achieve a certain generalization error. Eisenberg and Rivest [ER90]

[3]Note that our definition of query learning corresponds to the notion of 'membership queries' in machine learning. The term 'membership' arises because in concept learning the query "what is the output corresponding to input x ?" can be rephrased as "is x a member of class 1 ?". Other types of queries such as 'equivalence queries' – where the student can ask whether its guess for the teacher rule is entirely correct, and receives a counterexample if not – have also been studied, but will not be considered in the following.

investigated this issue within a worst-case, distribution-free model of learning, where nothing is known about the distribution of inputs. They found that a combination of random examples and queries selected by query construction does not yield a significantly lower sample complexity than learning from random examples alone. Similar results were derived by Kulkarni *et al.* [KMT93] for learning from general binary valued queries. The intuitive reason for this failure of query construction to be useful in distribution-free scenarios is that after any finite number of queries, a 'malicious adversary' can always pick an input distribution which has negligible weight in the input regions where the queries were chosen, thus rendering the information from the corresponding training outputs useless for the purpose of generalization. To prevent this problem, we shall assume in our discussion of query construction that the input distribution is known (either *a priori* or as an estimate based on a sample of random inputs) to the query selection algorithm; the results in [KMT93] show that in this case, queries *can* reduce the sample complexity. The alternative is to use query filtering instead of query construction, as advocated by Freund *et al.* [FSST93, Fre93] on the basis of the negative results for query construction mentioned above; this automatically incorporates knowledge about the input distribution through the sampling from a stream of random inputs. Note that for either query filtering or query construction, we implicitly assume that 'unlabelled' inputs (i.e., inputs without the corresponding outputs) are cheaply available, so that we only have to concern ourselves with reducing the number of training outputs needed. This assumption applies to a wide range of scenarios. Cohn *et al.* [CAL94] give speech recognition as an example: while speech recordings (inputs) can be obtained fairly easily and by a more or less automatic process, the phonetic classification (output) normally requires expensive human work.

The work of [SOS92, FSST93] referred to above provides an example of query selection for optimization of a given objective function. This general approach can be traced back to research in statistics on optimal experimental design. This field has been studied in great depth, and in this short survey we cannot hope to do justice to the vast body of existing literature. For reviews of earlier work, which concentrated on linear students (and teachers) and non-sequential query selection, we refer to, for example, [Fed72, AH78, Sil80, Atk82]; research in the closely related field of 'response surface methods' has been summarized in [BD87, KC87, MKC89]. Work on nonlinear learning problems is comparatively more recent, and has been reviewed in, e.g., [FTW85, FTK89, Pil91, AD91, CM93]. The general emphasis of work in optimal design has been the theoretical investigation of equivalences between different objective functions and the properties of the resulting predictors, such as unbiasedness and

asymptotic consistency. The question of how much query learning can improve generalization performance has been addressed only occasionally with the help of computer simulations; Cohn [Coh94, CGJ95], for example, used simulations to compare the performance obtained by query selection based on optimal experimental design procedures with learning from random examples.

Among the different approaches to optimal experimental design, the one that is most closely related to the framework that we shall use is 'optimal Bayesian sequential design' (see e.g., [Pil91]). MacKay [Mac92c] has proposed several objective functions for query selection in this context, all based on measures of information gain (i.e., entropy). Within a Gaussian approximation for all probability distributions involved in the analysis, he confirmed the intuitive notion that queries should be selected in regions where the uncertainty of the student's prediction is high. MacKay also pointed out and demonstrated the 'Achilles' heel' of the Bayesian approach [Mac92c, Mac92b], which consists in the implicit assumption that the training algorithm (or, more generally, the 'inference model') is correct, i.e., reproduces the true (posterior) distribution of teachers that could have generated the data. This point, which is discussed in more detail in Chapter 5, is our main motivation for using a framework which allows for an explicit distinction between student and teacher space (see Chapter 2).

Finally, we mention the work of Paass and Kindermann [PK95], who explored the practical implementation of Bayesian strategies for query learning using Monte Carlo methods. They used a decision theoretic framework and analysed, in several simple scenarios, the performance of queries selected to minimize the expected average loss. In our terminology, their approach corresponds to minimizing the generalization error, under the assumption that the inference model is correct. Sung and Niyogi [SN95] used a similar framework and found encouraging performance results for query learning in two toy learning problems with a one-dimensional input space.

1.3 Aims and approach

As explained above, the main question that we are interested in answering is: How useful is query learning? We already touched on the fact that an answer to this question has two components: On the one hand, one has to know by how much generalization performance can be improved by query learning; on the other hand, one should also take into account the computational cost of the query selection process itself, which can off-set the benefits of a reduced number of training examples. For simplicity, we shall largely ignore the second point in our work, focusing instead on an assessment of the performance gains afforded by query learning. The emphasis will be on exploring some

of the more basic capabilities and limitations of query learning, rather than necessarily finding practical query selection algorithms.

In our review of existing work, we have divided query selection algorithms into two categories: heuristic, and those derived from the optimization of some objective function. It has been pointed out before [Mac92c] that, while heuristic approaches can demonstrate the power of query learning in specific instances, they do not allow a systematic study of possible improvements of query selection algorithms, nor do their results generalize easily to learning problems other than those for which they have been designed. We shall therefore restrict our attention to query selection algorithms derived from optimization of objective functions, which we expect to yield more widely applicable insights. Existing work in this area as reviewed above leaves a lot of open questions, and the picture is far from complete. In particular, analytical evaluations of the generalization performance achievable by query learning [KR90, KC92, WR92, BS93, SOS92, FSST93] have been confined to extremely idealized situations where the student is of exactly the same form as the teacher, i.e., the rule is perfectly learnable, and the training examples are not corrupted by any form of noise. Noise is generally taken into account in work on optimal experimental design, but the analysis there is not normally concerned with the question of how much optimal design (i.e., query learning) improves generalization performance (with the exception of simulation work like [Coh94, CGJ95]). Bayesian optimal experimental design can be hampered by the built-in assumption of a correct inference model, which may lead to non-sensical query selection schemes [Mac92c, Mac92b]. In light of the above, the framework that we shall use will be probabilistic in order to allow a convenient representation of both noise processes corrupting the training data and non-deterministic training and query selection algorithms. As such, it will be similar to a 'traditional' Bayesian setup, but we will extend this to allow for an explicit distinction between student and teacher space, based on the work of Wolpert [Wol92]. Within this framework, we will try to elucidate the following questions: How does the efficacy of query learning depend on the nature of the rule to be learned (i.e., the functional form of the teacher)? How is it affected by noise, and by training algorithms which are not optimally matched to the learning problem at hand? What effect does the choice of the objective function for query selection have? What happens in scenarios where the student cannot reproduce the teacher perfectly, i.e., the problem is imperfectly learnable? What influence does the 'size' of the learning problem (given, for example, by the number of parameters needed to describe a student) have? And what are the differences between query learning in single-layer and multi-layer networks?

Our philosophy with regard to the above questions will be to study them in model

learning problems which are simple enough to be analysed analytically, but generic enough to allow conclusions of some generality to be drawn. The particular scenarios that we consider are inspired by models of feed-forward neural networks, and their simplicity will allow us to use the tools of statistical mechanics in analysing them. The power of these tools in the analysis of learning has been amply demonstrated in the past (see, e.g., [Ami89, HKP91, WRB93]). Of particular interest to us is their ability to predict average-case results – which can differ significantly from the worst-case bounds typically studied in machine learning and computational learning theory (see, e.g., [HKST94]) – and which are more representative of 'typical' behaviour. Furthermore, by considering an appropriate thermodynamic limit, statistical mechanics allows us to study situations with an effectively finite number of training examples, in contrast to the large sample size asymptotics conventionally used in statistics. Overall, we follow the theorist's paradigm that simple, analytically solvable models can go a long way towards improving our understanding of phenomena occurring in more realistic situations; although we will concentrate on theoretical analysis, computer simulations will occasionally be used to confirm and supplement the theoretical results.

1.4 Structure of thesis

The rest of this thesis is structured as follows. We begin in Chapter 2 with an exposition of the probabilistic framework within which we will discuss query learning, introducing also the objective functions for query selection which we shall investigate. In Chapter 3, this framework is then applied to simple learning scenarios in which student and teacher space are identical (perfectly learnable problems). The emphasis is on an elucidation of the dependence of the efficacy of query learning on the objective function used for query selection, and on the functional form of the rule to be learned. We also explore the effects of using training algorithms which are suboptimally matched to the learning problem. General issues such as the dependence of the effect of a single query on the 'learning history', i.e., on whether the previous examples were selected randomly or by querying, and the differences between 'globally' and 'locally' optimal query sequences are also investigated. Chapter 4, which focuses on learning with linear students, expands the focus out to imperfectly learnable problems. This gives us the opportunity to investigate the differences between objective functions for query selection defined in teacher and in student space, which can be applied depending on whether or not knowledge about the teacher space is available. In Chapter 5, we discuss why in the absence of such knowledge one is essentially forced back to a 'traditional' Bayesian framework for query learning. The consequences are explored in detail for learning

with binary perceptron students, in particular with regard to the effects of different kinds of noise on the training data. Several pitfalls are exposed which can limit the usefulness of query learning. A potential solution consists in combining query learning with methods for selecting or adapting the inference model in the light of the data. This idea is explored in Chapter 6, with encouraging results. In the final two chapters, we extend our analysis further, with the aim of making contact with neural networks that would be used in practical supervised learning problems: In Chapter 7, we investigate corrections to the thermodynamic limit of infinite system size on which most of the analysis in the preceding chapters is based; our findings demonstrate that the thermodynamic limit analysis is often directly applicable to learning systems of typical 'real-world' sizes. Chapter 8 constitutes a first step towards a better understanding of query learning in multi-layer networks, with results which point to the applicability of query learning in networks more complex than the ones we analyse explicitly. We conclude with a summary of our main results, a discussion of their implications, and an outline of open questions that remain or have been raised.

Chapter 2

A probabilistic framework for query selection

Abstract

A general probabilistic framework for deriving query selection algorithms from the requirement of optimizing a given objective function is described. We use an 'extended Bayesian' formalism, which has the advantage of distinguishing clearly between teacher and student space. Two important objective functions for query selection are introduced, which will be used frequently in later chapters: entropy (or information gain) and generalization error. Finally, we discuss the evaluation of the average generalization performance obtained by a query selection algorithm.

2.1 Introduction

In this chapter we introduce a general probabilistic framework for the derivation of query selection algorithms based on optimization of a given objective function. The structure is similar to Wolpert's 'extended Bayesian framework' [Wol92], but the notation we employ (explained in detail in Section 2.2 below) is closer to that normally used in the neural networks community, see e.g., [WRB93]. Two common objective functions for query selection, entropy and generalization error, are introduced in Section 2.3. In Sections 2.4 and 2.5, which form the core of this chapter, we explain how query selection algorithms should be derived from the requirement of optimizing a given objective function and how their generalization performance should be evaluated. General conventions and assumptions that will be used throughout this thesis are given in Appendix 2.6, together with some proofs.

2.2 Notation

In a discussion of supervised learning, the basic variables are the *teacher*, a *training set* of examples generated by the teacher, and the *student*, which is chosen on the basis of the training set and which is supposed to *generalize*, i.e., to make good predictions for outputs corresponding to new inputs. Our general approach will be to cast the problem in terms of probability distributions of these quantities. Although we shall use the terminology normally used by researchers in the field of neural networks, our framework is in fact applicable to more general forms of statistical inference (nonlinear regression, classification etc.).

We denote students by $\mathcal{N}$ (for 'Neural network') and teachers by $\mathcal{V}$ (for 'elements of the Version space', a term borrowed from Artificial Intelligence terminology [Mit82] which we will encounter more frequently in the following chapters). Each teacher and each student implement a mapping from inputs x (typically $\in \mathbb{R}^N$) to outputs y (often $\in \mathbb{R}$). Let $\Theta^{(p)}$ denote a (ordered) training set consisting of p examples (x^μ, y^μ), $\mu = 1, \ldots, p$. We define the following probability distributions:

- $P(y|x, \mathcal{V})$, the probability of, given input x, obtaining output y from teacher $\mathcal{V}$. This probability distribution specifies the input-output mapping implemented by the teacher $\mathcal{V}$, including possible corruption by noise. If $P(y|x, \mathcal{V})$ can be written in the form $\delta(y - f_{\mathcal{V}}(x))$, we call the teacher 'noise free', otherwise 'noisy'.

- $P(x)$, the probability distribution of inputs when these are randomly selected, i.e., not queried. As commonly assumed, this distribution also governs the selection of test examples, from which the generalization error is calculated.

- $P(\Theta^{(p)}|\mathcal{V})$, the probability of obtaining a specific training set from the teacher $\mathcal{V}$ (plus noise, possibly, which is always understood in the following). For randomly (and independently) selected examples, this can be written as

$$P(\Theta^{(p)}|\mathcal{V}) = \prod_{\mu=1}^{p} P(y^\mu|x^\mu, \mathcal{V})P(x^\mu). \tag{2.1}$$

- $P(\mathcal{V})$, the prior distribution of teachers[1].

[1] As a notational shorthand, we assume that in all probability distributions in which $\Theta^{(p)}$ appears, the number of examples p is held fixed, without writing this explicitly. Thus, for example, $P(\Theta^{(p)}|\mathcal{V})$ should strictly be written as $P(\Theta^{(p)}|\mathcal{V}, p)$; hence it is normalized to one when integrating over all possible training sets of size p. To make this convention consistent with the use of Bayes' Theorem as in (2.2), we also make the natural assumption that the number of training examples is independent of the teacher rule that we are trying to learn. Thus, $P(p|\mathcal{V}) = P(p)$ and hence $P(\mathcal{V}|p) = P(\mathcal{V})$, so that we only need one prior teacher distribution for all values of p.

- $P(\mathcal{V}|\Theta^{(p)})$, the posterior teacher distribution, which can be calculated from $P(\Theta^{(p)}|\mathcal{V})$ and $P(\mathcal{V})$ using Bayes' Theorem

$$P(\mathcal{V}|\Theta^{(p)}) = \frac{P(\Theta^{(p)}|\mathcal{V})\ P(\mathcal{V})}{\int d\mathcal{V} P(\Theta^{(p)}|\mathcal{V})\ P(\mathcal{V})} \ . \tag{2.2}$$

- $P(\mathcal{N}|\Theta^{(p)})$, the 'post-training' distribution of students which specifies the training or learning algorithm in terms of the probability that training on the given set of examples[2] $\Theta^{(p)}$ will yield the student $\mathcal{N}$. We shall assume that students are deterministic, i.e., that for each input x a student $\mathcal{N}$ provides an output which can be written in the form $y = f_{\mathcal{N}}(x)$.

We emphasize that in a real-world learning problem, normally only $P(\mathcal{N}|\Theta^{(p)})$ and possibly $P(x)$ will be known, whereas all the probability distributions concerning the teacher $\mathcal{V}$ will be unknown. The latter occur in two different roles below. Firstly, they are used in the definition of query selection algorithms with the aim of optimizing a given objective function (see Section 2.4). This is reasonable since knowledge about the teacher space and the relevant probability distributions, if available, should of course be used for selecting queries optimally. Scenarios of this type (i.e., with a *known* teacher space) are considered in detail in Chapters 3, 4 and 8. If the teacher space is *unknown*, the definition of query selection algorithms given below has to be modified by approximating unknown teacher space quantities by suitable student space quantities. This is explained in detail in Chapter 5; in such cases, the results obtained from analyses of learning scenarios with a known teacher space can only serve as a theoretical baseline for the optimal generalization performance that can be obtained by query learning.

The second role in which probabilities involving the teacher $\mathcal{V}$ will appear below is in the performance evaluation of query selection algorithms. This cannot be circumvented since quantitative analysis of a learning scenario is only possible if assumptions are made about the teacher space. The necessity of such assumptions follows from the intuitively obvious result that, in the absence of any knowledge about the functional form and complexity of the teacher, generalization (and hence also its improvement by query selection) is impossible [Wol92].

[2]The outcome of the learning algorithm can in principle depend on the order in which the training examples are presented. For brevity, however, we will continue to use the term 'training set' instead of the more technically correct alternative 'sequence of training examples'.

2.3 Candidate objective functions

There are a variety of objective functions that one might want to optimize by a query selection algorithm. We restrict our attention to two very common ones: Entropy (or information) and generalization error.

For a given training set $\Theta^{(p)}$, the entropy in teacher space can be defined as the entropy of the posterior distribution[3] $P(\mathcal{V}|\Theta^{(p)})$

$$S_{\mathcal{V}}(\Theta^{(p)}) = -\int d\mathcal{V}\, P(\mathcal{V}|\Theta^{(p)}) \ln P(\mathcal{V}|\Theta^{(p)}). \tag{2.3}$$

The entropy in student space is defined similarly as a functional of the post-training distribution $P(\mathcal{N}|\Theta^{(p)})$ (which depends on the learning algorithm that we are using). The information gain due to an additional example in either teacher or student space is defined as the decrease in the corresponding entropy.

We emphasize that student and teacher space entropy coincide *only* if $P(\mathcal{V}|\Theta^{(p)})$ and $P(\mathcal{N}|\Theta^{(p)})$ have exactly the same form. This is always the case in Bayesian analyses, where $\mathcal{V}$ and $\mathcal{N}$ are effectively identified (see, e.g., [Mac92c]). In recent research on query learning [SOS92, FSST93] where the distinction between $\mathcal{V}$ and $\mathcal{N}$ was taken into account, the learning algorithm was nevertheless always chosen such that $P(\mathcal{V}|\Theta^{(p)})$ and $P(\mathcal{N}|\Theta^{(p)})$ were still identical. In the applications of our framework in the following chapters, we shall see that new features can emerge if this is not the case.

The generalization error is probably the most commonly used measure of the performance of a student when trying to approximate a given teacher. It is defined starting from a specifically chosen error measure

$$e(y, x, \mathcal{N}) \tag{2.4}$$

which determines how much the output of the student $\mathcal{N}$ for input x is in error compared to the correct output y. Averaging this over all input/output pairs produced by a teacher $\mathcal{V}$, we obtain the generalization error, a measure of how closely $\mathcal{N}$ approximates $\mathcal{V}$:

$$\epsilon_{\mathrm{g}}(\mathcal{N}, \mathcal{V}) = \langle e(y, x, \mathcal{N}) \rangle_{P(y|x,\mathcal{V})P(x)}\,. \tag{2.5}$$

Regarding the relationship between generalization error and entropy, we note that a decrease in entropy need not necessarily be correlated with a decrease in generalization

[3] If there is a continuum of teachers $\mathcal{V}$, $P(\mathcal{V}|\Theta^{(p)})$ is a probability density which has the dimension of the inverse of $\mathcal{V}$. Strictly speaking, a dimensional normalizing constant is then necessary to make the argument of the logarithm in (2.3) dimensionless, but we shall not write this explicitly since it cancels from the entropy differences we will be concerned with.

error, cf. the discussion in [FSST93]. Note also that we have followed the bulk of the literature on the subject of supervised learning and statistical inference by defining the generalization error as an average of the error measure over all possible inputs. One could argue that this is incorrect since generalization should, strictly speaking, refer to making correct predictions for *new* inputs, and that the average should therefore only be taken over inputs not contained in the training set. For a detailed discussion of the properties of the resulting 'off-training-set' generalization error in scenarios where the total number of different inputs is finite, see [Wol92, Wol95]. In the cases that we will consider, there will always be a continuum of inputs (normally real vectors), so that the exclusion of a discrete set of training inputs will not make any difference to the definition of the generalization error.

Finally, for examples of a class of objective functions which have a somewhat intermediate character between entropy and generalization error, the 'prediction probabilities' and variants thereof, we refer the reader to [AM93, LTS90].

2.4 Derivation of query selection algorithms

We assume now that we are given an objective function, such as entropy or generalization error, which our query selection algorithm is supposed to optimize. We write this objective function in the generic form

$$\epsilon(\mathcal{N}, \mathcal{V}, \Theta^{(p)}).$$

We only consider query selection algorithms which are local in the sense that they work one example at a time, performing a greedy optimization of the given objective function at each query selection (see however Section 3.4, where we discuss what happens if this restriction is dropped). We also assume that after the new example is added to the training set, complete retraining takes place, i.e., the learning algorithm is re-applied to the enlarged training set $\Theta^{(p+1)}$. This excludes a dependence of query selection on the specific student (a representative of the distribution $P(\mathcal{N}|\Theta^{(p)})$) obtained after training on the existing data set, as considered in [KR90, WR92]. Of course, a dependence on the actual—unknown—teacher $\mathcal{V}$ that generated the training data is not possible either, and so query selection can only be based on the existing training set $\Theta^{(p)}$. We therefore need to derive a function $\epsilon(x, \Theta^{(p)})$ which depends only on this existing training set and the next input, x. A query construction algorithm is then defined as picking, each time it is invoked, as the next query the value of x at which $\epsilon(x, \Theta^{(p)})$ attains its global

optimum, or randomly one such value if there is more than one global optimum[4]. As pointed out in the introduction (Chapter 1), we shall not be concerned with the actual implementation or computational complexity of this optimization process. In order to prevent the construction of non-sensical queries, we restrict the range of input values from which the query construction algorithm is allowed to choose to the support of $P(x)$, i.e., to values of x which could also appear in a random training or test example. We remark in passing that the function $\epsilon(x, \Theta^{(p)})$ or an approximation to it can also be used to define a query filtering (as opposed to construction) algorithm which accepts a random input x with a probability which is a function of the corresponding value of $\epsilon(x, \Theta^{(p)})$. In the following chapters, we will not consider this possibility further, focusing instead on query construction. However, we shall return to the topic of query filtering in Chapters 5 and 8, where we discuss the 'query by committee' algorithm.

In order to obtain the function $\epsilon(x, \Theta^{(p)})$, which we do not want to depend on $\mathcal{N}$, we first average the given objective function over the post-training distribution:

$$\epsilon(\mathcal{V}, \Theta^{(p)}) = \left\langle \epsilon(\mathcal{N}, \mathcal{V}, \Theta^{(p)}) \right\rangle_{P(\mathcal{N}|\Theta^{(p)})}. \tag{2.6}$$

Averaging this result over the posterior teacher distribution, we obtain an average objective function which depends on the training data only:

$$\epsilon(\Theta^{(p)}) = \left\langle \epsilon(\mathcal{V}, \Theta^{(p)}) \right\rangle_{P(\mathcal{V}|\Theta^{(p)})}. \tag{2.7}$$

We can now calculate the function defining a query selection algorithm by averaging (2.7), evaluated for the training data set $\Theta^{(p)} + (x, y)$, over the possible outputs y that the teachers in the posterior distribution produce for the input x:

$$\epsilon(\Theta^{(p)}, x) = \left\langle \epsilon(\Theta^{(p)} + (x, y)) \right\rangle_{P(y|x,\Theta^{(p)})} \tag{2.8}$$

where $P(y|x, \Theta^{(p)})$ is given by

$$P(y|x, \Theta^{(p)}) = \int d\mathcal{V} P(y|x, \mathcal{V}) P(\mathcal{V}|\Theta^{(p)}). \tag{2.9}$$

We show in Appendix 2.6 that the same result can be obtained by first evaluating $\epsilon(\mathcal{V}, \Theta^{(p)})$ for the training set $\Theta^{(p)} + (x, y)$, averaging over the outputs that $\mathcal{V}$ produces

[4]This randomization is important to prevent certain pathologies that could otherwise occur; an explicit example of this, due to Freund [Fre93, FSST93], is described in Section 3.2.

for x, and then averaging this over the posterior teacher distribution:

$$\epsilon(\Theta^{(p)}, x) = \left\langle \left\langle \epsilon(\mathcal{V}, \Theta^{(p)} + (x, y)) \right\rangle_{P(y|x,\mathcal{V})} \right\rangle_{P(\mathcal{V}|\Theta^{(p)})} . \tag{2.10}$$

Equations (2.8) and (2.10) constitute the main result of this section and can be used interchangeably as definitions of the function $\epsilon(\Theta^{(p)}, x)$ which defines a query selection algorithm.

2.5 Performance evaluation of query selection algorithms

A query construction algorithm as defined in the previous section yields a probability of querying x if the existing training set is $\Theta^{(p)}$,

$$P_{\mathrm{Q}}(x|\Theta^{(p)}) \tag{2.11}$$

which is uniform over the set of all x for which $\epsilon(\Theta^{(p)}, x)$ attains its global optimum (among the x for which $P(x)$ is nonzero) and zero everywhere else. We shall evaluate the performance of this query construction algorithm when used to generate query sequences, using the generalization error as our performance measure. Starting from (2.5) we define first by analogy with (2.6) the average generalization error with respect to the post-training distribution:

$$\epsilon_{\mathrm{g}}(\mathcal{V}, \Theta^{(p)}) = \langle \epsilon_{\mathrm{g}}(\mathcal{N}, \mathcal{V}) \rangle_{P(\mathcal{N}|\Theta^{(p)})} . \tag{2.12}$$

We then define the average generalization error obtained after a sequence of p queries when the true teacher is $\mathcal{V}$ as

$$\epsilon_{\mathrm{g,Q}}(\mathcal{V}) = \left\langle \epsilon_{\mathrm{g}}(\mathcal{V}, \Theta^{(p)}) \right\rangle_{P_{\mathrm{Q}}(\Theta^{(p)}|\mathcal{V})} \tag{2.13}$$

where the training sets are now generated according to the distribution

$$P_{\mathrm{Q}}(\Theta^{(p)}|\mathcal{V}) = \prod_{\mu=1}^{p} P(y^{\mu}|x^{\mu}, \mathcal{V}) P_{\mathrm{Q}}(x^{\mu}|\Theta^{(\mu-1)}) \tag{2.14}$$

in analogy to (2.1) which applies to the case of random examples. By averaging over the prior teacher distribution, we obtain the generalization error for an average teacher:

$$\epsilon_{\mathrm{g,Q}} = \langle \epsilon_{\mathrm{g,Q}}(\mathcal{V}) \rangle_{P(\mathcal{V})} \tag{2.15}$$

In terms of $\epsilon_g(\Theta^{(p)}) = \left\langle \epsilon_g(\mathcal{V}, \Theta^{(p)}) \right\rangle_{P(\mathcal{V}|\Theta^{(p)})}$, this can be written as follows: As shown in Appendix 2.6, the posterior distribution of teachers $P(\mathcal{V}|\Theta^{(p)})$ is the same for random examples and for queries. This reflects the intuitively obvious fact that the way we generate data by querying or random sampling does not influence our inferences about the underlying rule [Mac92c]. Using Bayes' theorem, one can thus write

$$\begin{aligned} \epsilon_{g,Q} &= \left\langle \left\langle \epsilon_g(\mathcal{V}, \Theta^{(p)}) \right\rangle_{P_Q(\Theta^{(p)}|\mathcal{V})} \right\rangle_{P(\mathcal{V})} \\ &= \left\langle \left\langle \epsilon_g(\mathcal{V}, \Theta^{(p)}) \right\rangle_{P(\mathcal{V}|\Theta^{(p)})} \right\rangle_{P_Q(\Theta^{(p)})} \\ &= \left\langle \epsilon_g(\Theta^{(p)}) \right\rangle_{P_Q(\Theta^{(p)})} . \end{aligned} \tag{2.16}$$

Again applying Bayes' theorem, the distribution $P_Q(\Theta^{(p)})$ can be written as

$$P_Q(\Theta^{(p)}) = \prod_{\mu=1}^{p} P(y^\mu|x^\mu, \Theta^{(\mu-1)}) P_Q(x^\mu|\Theta^{(\mu-1)}) \; ; \tag{2.17}$$

$P(y^\mu|x^\mu, \Theta^{(\mu-1)})$ is given by (2.9). For the case of training sets which are generated by a mixture of queries and random examples, one can still use (2.16) as long as in (2.17) the terms $P_Q(x^\mu|\Theta^{(\mu-1)})$ are replaced by $P(x^\mu)$ for the examples (x^μ, y^μ) that were generated randomly.

We remark that if one wants to know the average generalization error obtained by adding a query x and the corresponding output y to an existing fixed training set $\Theta^{(p)}$, all one needs to do is drop the average over $\Theta^{(p)}$ in (2.16), with the result

$$\begin{aligned} \epsilon_g(\Theta^{(p)} + 1 \text{ query}) &= \left\langle \left\langle \epsilon_g(\Theta^{(p)} + (x, y)) \right\rangle_{P(y|x,\Theta^{(p)})} \right\rangle_{P_Q(x|\Theta^{(p)})} \\ &= \left\langle \epsilon_g(\Theta^{(p)}, x) \right\rangle_{P_Q(x|\Theta^{(p)})} . \end{aligned} \tag{2.18}$$

We have derived equations (2.13), (2.16) and (2.18) in order to show that for the evaluation of performance of a query selection algorithm, the same functions $\epsilon_g(\mathcal{V}, \Theta^{(p)})$, $\epsilon_g(\Theta^{(p)})$ and $\epsilon_g(\Theta^{(p)}, x)$ can be used that have to be calculated anyway for the derivation of minimum generalization error query selection. In the following chapters, however, we shall avoid formal use of these results whenever more direct and intuitive derivations are possible.

2.6 Appendix: Assumptions and proofs

Let us first explain some general conventions regarding our notation, which will be used throughout this thesis. In all the learning scenarios that we consider, we will assume that both students $\mathcal{N}$ and teachers $\mathcal{V}$ have a natural parameterization in terms of a vector of parameters. This vector will normally be referred to as a 'weight vector', in analogy with the term 'synaptic weights' which is often used in the neural networks literature. In our case, the weight vector components will normally be real, corresponding to a continuum of students or teachers. Integration over student and teacher space will be written symbolically as $\int d\mathcal{N}$ and $\int d\mathcal{V}$, respectively, and represents an integration over the weight space with the standard Lebesgue measure. The corresponding probability distributions such as $P(\mathcal{N}|\Theta^{(p)})$, $P(\mathcal{V}|\Theta^{(p)})$ etc. should be understood as probability densities with respect to this measure. As a general rule, we shall not distinguish in our notation between random variables and their realizations.

For inputs and outputs, we use similar conventions. While the inputs x will normally be real vectors in the cases that we consider, outputs y can be either real or discrete. In the latter case, an integration $\int dy$ over outputs should be understood as a summation over all the values that y can take, and quantities such as $P(y|x,\mathcal{V})$ should be read as probabilities rather than probability densities.

We now state some general assumptions which we make regarding the probability distributions that were introduced in Section 2.2. Since these assumptions will apply to both random examples and queries, we drop the subscript 'Q' used in Section 2.5 to denote probability distributions for the case of query learning. We start with the natural requirement that any training algorithm can only generate students on the basis of the training set, and does not have any other information about the unknown teacher. This is formalized as

$$P(\mathcal{N}|\Theta^{(p)},\mathcal{V}) = P(\mathcal{N}|\Theta^{(p)}) \tag{2.19}$$

(see, e.g., [Wol92]) and implies that the distributions of students and teachers (post-training and posterior distribution, respectively) are independent once a training set $\Theta^{(p)}$ is given:

$$P(\mathcal{N},\mathcal{V}|\Theta^{(p)}) = P(\mathcal{N}|\Theta^{(p)})\,P(\mathcal{V}|\Theta^{(p)}).$$

We have implicitly used this assumption in the definition of the average over students and teachers of the objective function for query selection (see eqs. (2.6, 2.7)).

Our next assumption formalizes the fact that the selection of a query x^{μ} is made *only* on the basis of the existing training set, and does not depend on either the unknown

teacher $\mathcal{V}$ or a particular student $\mathcal{N}$ obtained after training:

$$P(x^\mu|\Theta^{(\mu-1)},\mathcal{N},\mathcal{V}) = P(x^\mu|\Theta^{(\mu-1)}). \tag{2.20}$$

This is trivially verified for random examples (where, in addition, x^μ is independent of $\Theta^{(\mu-1)}$). Note that the relation (2.20) would not hold if any of the training inputs or outputs added to the training set *after* x^μ has been selected were added to the variables on which the distribution of x^μ is conditioned. This is due to the fact that the outputs of these 'later' training examples depend on their corresponding inputs, whose selection in turn depends on the output y^μ; this would destroy the independence of the teacher $\mathcal{V}$ since y^μ and x^μ are coupled through $\mathcal{V}$.

The last assumption that we make concerns the generation of training outputs, whose probability distribution is fully determined once the corresponding input and the teacher are known:

$$P(y^\mu|x^\mu,\Theta^{(\mu-1)},\mathcal{V}) = P(y^\mu|x^\mu,\mathcal{V}). \tag{2.21}$$

We now derive some general relations which follow from the above assumptions. First of all, by applying Bayes' theorem and using (2.20, 2.21) one has

$$\begin{aligned} P(\Theta^{(p)}|\mathcal{V}) &= \prod_{\mu=1}^{p} P(y^\mu|x^\mu,\Theta^{(\mu-1)},\mathcal{V})\, P(x^\mu|\Theta^{(\mu-1)},\mathcal{V}) \\ &= \prod_{\mu=1}^{p} P(y^\mu|x^\mu,\mathcal{V})\, P(x^\mu|\Theta^{(\mu-1)}). \end{aligned} \tag{2.22}$$

as stated in (2.14). It follows that

$$\begin{aligned} P(\Theta^{(p)}) &= \int d\mathcal{V}\, P(\Theta^{(p)}|\mathcal{V})\, P(\mathcal{V}) \\ &= \left(\prod_{\mu=1}^{p} P(x^\mu|\Theta^{(\mu-1)})\right) \int d\mathcal{V}\; P(\mathcal{V}) \prod_{\mu=1}^{p} P(y^\mu|x^\mu,\mathcal{V}) \end{aligned}$$

and this gives the posterior teacher distribution

$$P(\mathcal{V}|\Theta^{(p)}) = \frac{P(\Theta^{(p)}|\mathcal{V})P(\mathcal{V})}{P(\Theta^{(p)})} = \frac{P(\mathcal{V})\prod_{\mu=1}^{p}P(y^\mu|x^\mu,\mathcal{V})}{\int d\mathcal{V}\, P(\mathcal{V})\prod_{\mu=1}^{p}P(y^\mu|x^\mu,\mathcal{V})}. \tag{2.23}$$

The independence of this result of the distributions $P(x^\mu|\Theta^{(\mu-1)})$, which describe how training examples are selected, proves that the posterior teacher distribution is the same for random examples and queries [McC81]. This can be rephrased by saying that

our inferences about the teacher, as represented by the posterior distribution, are not biased by query learning [Mac92c]. Note that this no longer holds if, as is commonly done in procedures for selecting informative subsets of a given larger training set, the selection of a training input would be allowed to depend on the corresponding output (M. Plutowski, private communication; see also [Ber85, Mac92c]).

Our assumptions also imply the general result (2.9). From Bayes' theorem, one has

$$P(y^\mu|x^\mu,\Theta^{(\mu-1)}) = \int d\mathcal{V}\, P(y^\mu|x^\mu,\Theta^{(\mu-1)},\mathcal{V})\, P(\mathcal{V}|x^\mu,\Theta^{(\mu-1)}) \tag{2.24}$$

The last term on the right hand side can be simplified by using the following consequence of (2.20):

$$P(\mathcal{V}|x^\mu,\Theta^{(\mu-1)}) = P(\mathcal{V}|\Theta^{(\mu-1)}). \tag{2.25}$$

Eq. (2.9) then follows by inserting (2.21) into (2.24).

Let us now derive from our assumptions the equivalence of the two definitions given in the text, eqs. (2.8, 2.10), of the function $\epsilon(\Theta^{(p)},x)$ defining a query selection algorithm. If we denote the training set with the new training example added by $\Theta^{(p+1)} = \Theta^{(p)} + (x,y)$ (remember that we set $(x,y) \equiv (x^{p+1},y^{p+1})$ in Section 2.4 in order to simplify the notation), the definition (2.8) takes the form

$$\begin{aligned}\epsilon_1(\Theta^{(p)},x) &= \left\langle \epsilon(\Theta^{(p+1)})\right\rangle_{P(y|x,\Theta^{(p)})} \\ &= \left\langle\left\langle \epsilon(\mathcal{V},\Theta^{(p+1)})\right\rangle_{P(\mathcal{V}|\Theta^{(p+1)})}\right\rangle_{P(y|x,\Theta^{(p)})}\end{aligned}$$

while the second definition (2.10) reads

$$\epsilon_2(\Theta^{(p)},x) = \left\langle\left\langle \epsilon(\mathcal{V},\Theta^{(p+1)})\right\rangle_{P(y|x,\mathcal{V})}\right\rangle_{P(\mathcal{V}|\Theta^{(p)})}.$$

To prove that $\epsilon_1(\Theta^{(p)},x)$ is identical to $\epsilon_2(\Theta^{(p)},x)$, it therefore suffices to show that

$$P(\mathcal{V}|\Theta^{(p+1)})\, P(y|x,\Theta^{(p)}) = P(y|x,\mathcal{V})\, P(\mathcal{V}|\Theta^{(p)}) \tag{2.26}$$

Using Bayes' theorem, the left hand side can be written as

$$P(\mathcal{V},y|x,\Theta^{(p)}) = P(y|x,\Theta^{(p)},\mathcal{V})\, P(\mathcal{V}|x,\Theta^{(p)})$$

and equality with the right hand side of (2.26) follows from assumption (2.21) and the relation (2.25) derived from assumption (2.20).

We conclude by considering the case of effectively non-sequential queries (see, e.g.,

Section 3.3), where each training input depends only on the previous training inputs and not on the corresponding outputs. One then has the stronger version of (2.20)

$$P(x^\mu|\Theta^{(\mu-1)},\mathcal{N},\mathcal{V}) = P(x^\mu|x^{(\mu-1)})$$

where $x^{(\mu-1)}$ denotes the (ordered) set of training inputs $x^1, x^2 \ldots x^{\mu-1}$. This leads to the following simpler form of eq. (2.22)

$$P(\Theta^{(p)}|\mathcal{V}) = \left(\prod_{\mu=1}^{p} P(y^\mu|x^\mu,\mathcal{V})\right) P(x^{(p)}) \tag{2.27}$$

which we will use in Chapters 3 and 4 in order to split the average over training sets into an average over training outputs followed by an average over training inputs.

Chapter 3

Perfectly learnable problems: Two simple examples

Abstract

We study query construction in learning problems where the student can learn the teacher perfectly. For two simple scenarios, the high-low game and the linear perceptron, the generalization performance obtained by queries for minimum entropy and minimum generalization error is evaluated and compared to learning from random examples. We find qualitative differences between the two scenarios due to the different structure of the underlying rules (nonlinear and 'non-invertible' vs. linear); in particular, for the linear perceptron, random examples lead to the same generalization ability as a sequence of queries in the limit of an infinite number of examples. We also investigate the case of training algorithms which are ill-matched to the learning environment and find that in this case, minimum entropy queries can in fact yield a lower generalization ability than random examples. Finally, we study the efficacy of single queries and its dependence on the learning history, i.e., on whether the previous training examples were generated randomly or by querying, and the difference between globally and locally optimal query construction.

3.1 Introduction

In this chapter we apply the framework set out in the preceding chapter to two specific learning scenarios. We assume in both cases that the problem is perfectly learnable, i.e., that students and teachers have the same form. The case of imperfectly learnable rules, which must occur frequently in real-world problems, will be treated in the

next chapter. For a first pass at the problem of how the choice of objective function affects the performance of the corresponding query selection algorithms, we consider query construction based on optimization of the two objective functions *entropy* (or *information gain*) and *generalization error*. The specific learning scenarios considered are the 'high-low game' [FSST93] and the 'linear perceptron'. These two examples will allow us to gain some insight into the differences between query learning in linear and nonlinear systems. Since query selection is most effective when applied to all examples in the training set, i.e., when one allows the input of every new training example to be determined by the query construction algorithm, we consider the performance of the respective query construction algorithms when applied to generate *query sequences*, and compare the results to training on random examples. As performance measure we choose the generalization error because generalization is after all what we want to improve by query selection. For the linear perceptron, we also investigate the influence of a non-optimal training algorithm which is poorly matched to the posterior teacher distribution. In Section 3.4 we discuss some related issues: the efficacy of a single query and its dependence on the learning history, i.e., on whether the query is part of a query sequence or whether it is an *isolated query* after random examples; and the difference between *locally optimal* query selection, which builds up the training set step by step in a 'greedy' procedure, optimizing the given objective function at every step, and *globally optimal* query selection, which optimizes the whole query sequence for a given number of examples. We conclude in Section 3.5 with a summary and discussion of our results.

3.2 High-low

In the present section we consider the 'high-low game' [SOS92, FSST93], which is an extremely simple example of a nonlinear rule with real input x and binary output $y \in \{0, 1\}$. The output is simply 1 or 0 depending on whether the input x is greater or less than a certain preset threshold. Thus, for one-dimensional high-low, a noise free teacher is specified by a 'weight' $w_\mathcal{V}$ such that

$$P(y|x, \mathcal{V}) = \delta_{y, f_\mathcal{V}(x)}, \quad f_\mathcal{V}(x) = \Theta(x - w_\mathcal{V}) \tag{3.1}$$

where the Kronecker delta $\delta_{i,j}$ is equal to 1 if $i = j$ and 0 otherwise, and the step function $\Theta(x)$ is defined to be 1 if $x \geq 0$ and 0 otherwise. We assume that both inputs and teacher weights are taken from the unit interval $[0, 1]$. An N-dimensional generalization of this can be defined as follows [FSST93]: Inputs are now ordered pairs (i, x), where $i \in \{1, 2, \ldots, N\}$, $x \in [0, 1]$, and a teacher $\mathcal{V}$ is defined in terms of an

N-component vector $\mathbf{w}_\nu = (w_{\nu,i})_{i=1,2,...,N}$ and gives the output

$$f_\nu(i,x) = \Theta(x - w_{\nu,i}). \tag{3.2}$$

As explained in [FSST93], N-dimensional high-low is basically equivalent to N concurrent one-dimensional high-low games.

As pointed out above we assume that the rule is perfectly learnable, i.e., that our students have the same functional form as the teachers, a student $\mathcal{N}$ being specified by an N-dimensional weight vector $\mathbf{w}_\mathcal{N}$. We assume the distribution of inputs to be $P(i,x) = P(i)P(x)$ with $P(i) = 1/N$, and $P(x)$ uniform on $[0,1]$ and zero everywhere else. We also assume the prior teacher distribution $P(w_\nu)$ to be uniform on $[0,1]^N$. Under these assumptions, the posterior teacher distribution can easily be derived to be constant over the 'version space', i.e., the set of all teacher weight vectors which could have generated the training data, which is here simply a hypercube:

$$P(\mathcal{V}|\Theta^{(p)}) \propto \prod_{i=1}^{N} \Theta(w_{\nu,i} - x_{L,i})\Theta(x_{R,i} - w_{\nu,i}) \tag{3.3}$$

where we have denoted by $x_{L,i}$ and $x_{R,i}$ ('L' for left and 'R' for right boundary of the version space) the largest and smallest x-value of inputs from the training set $\Theta^{(p)}$ with a given value of i and output 0 and 1, respectively. The entropy in teacher space then follows from the definition (2.3) as

$$S_\nu(\Theta^{(p)}) = \sum_{i=1}^{N} \ln\,(x_{R,i} - x_{L,i}). \tag{3.4}$$

For the calculation of the generalization error, an obvious error measure is

$$e(y,(i,x),\mathcal{N}) = |y - f_\mathcal{N}(i,x)| \tag{3.5}$$

which is 0 if y and $f_\mathcal{N}(i,x)$ agree and 1 otherwise, yielding

$$\epsilon_g(\mathcal{N},\mathcal{V}) = \frac{1}{N}\sum_{i=1}^{N} |w_{\nu,i} - w_{\mathcal{N},i}|. \tag{3.6}$$

We consider two training algorithms: Zero temperature Gibbs learning, which is just given by

$$P_{\text{Gibbs}}(\mathcal{N}|\Theta^{(p)}) = P(\mathcal{V}|\Theta^{(p)})\Big|_{\mathcal{V}=\mathcal{N}} \tag{3.7}$$

and optimal learning in the sense of [Wat93] for which

$$P_{\rm opt}(\mathcal{N}|\Theta^{(p)}) = \prod_{i=1}^{N} \delta(w_{\mathcal{N},i} - (x_{L,i} + x_{R,i})/2). \tag{3.8}$$

We remark that whereas for Gibbs learning the entropy in student space is identical to that in teacher space, the former is undefined for optimal learning, as is generally the case for deterministic training algorithms.

For the generalization error averaged over the post-training distribution according to (2.6) and then over the posterior teacher distribution as in (2.7) one obtains

$$\epsilon_{\rm g,opt}(\Theta^{(p)}) = \frac{3}{4}\,\epsilon_{\rm g,Gibbs}(\Theta^{(p)}) = \frac{1}{4N}\sum_{i=1}^{N}(x_{R,i} - x_{L,i}). \tag{3.9}$$

Due to the proportionality between the two results we can restrict our attention to optimal learning in the following, dropping the subscript 'opt'.

Using (2.8), it is straightforward to calculate from (3.4) and (3.9) the defining functions for query construction for minimal teacher space entropy and minimal generalization error, respectively:

$$S_\nu(\Theta^{(p)}, (i,x)) = S_\nu(\Theta^{(p)}) + q_i \ln q_i + (1-q_i)\ln(1-q_i) \tag{3.10}$$

$$\epsilon_{\rm g}(\Theta^{(p)}, (i,x)) = \epsilon_{\rm g}(\Theta^{(p)}) - \frac{x_{R,i} - x_{L,i}}{2N}\, q_i(1-q_i) \tag{3.11}$$

where we have used the abbreviation

$$q_i = P(y=1|(i,x),\Theta^{(p)}) = \begin{cases} 0 & x \le x_{L,i} \\ (x - x_{L,i})/(x_{R,i} - x_{L,i}) & x_{L,i} < x < x_{R,i} \\ 1 & x \ge x_{R,i}. \end{cases} \tag{3.12}$$

Equations (3.10) and (3.11) are both minimized for $q_i = 1/2$, i.e., $x = (x_{L,i} + x_{R,i})/2$. This corresponds to the intuitively obvious method of bisecting a component of the version space. For $q_i = 1/2$ the value of $S_\nu(\Theta^{(p)}, (i,x))$ is independent of i, so that query construction for minimal teacher space entropy selects randomly any of the N possible values for i and then $x = (x_{L,i} + x_{R,i})/2$. By contrast, query construction for minimal generalization error can only select from those i-values for which $x_{R,i} - x_{L,i}$ is maximal since only then will $\epsilon_{\rm g}(\Theta^{(p)}, (i,x))$ be minimized. Thus, query construction for minimal generalization error specifies along which component the version space should be bisected, a piece of information which cannot be obtained from the requirement

of maximal information gain. In fact, as explained in [FSST93], one can, simply by always bisecting the same component of the version space, construct a sequence of queries which at each step achieves the maximal entropy reduction but for which the generalization error never drops below a finite threshold. In our framework for query construction, this kind of pathology is avoided by randomly selecting one of the inputs for which the expected information gain is maximized.

The difference between the two objective functions, entropy and generalization error, is reflected in the average performance of the two query construction algorithms when they are used to generate query sequences: Query construction for minimal generalization error yields, after a sequence of $p = \alpha N = (\lfloor\alpha\rfloor + \Delta\alpha)N$ queries (where $\lfloor\alpha\rfloor$ denotes the integer part of α and $\Delta\alpha = \alpha - \lfloor\alpha\rfloor$ its non-integer part) and the corresponding outputs, a version space with $N\Delta\alpha$ components of length $(1/2)^{\lfloor\alpha\rfloor+1}$ and $N(1-\Delta\alpha)$ components of length $(1/2)^{\lfloor\alpha\rfloor}$ and hence from (3.9) a generalization error of[1]

$$\epsilon_g(\text{min. gen. error queries}) = \frac{1}{4}\left(\frac{1}{2}\right)^{\lfloor\alpha\rfloor}\left(1-\frac{\Delta\alpha}{2}\right) \tag{3.13}$$

so that increasing α by one always reduces the generalization error by a factor of $1/2$. For minimal teacher space entropy, on the other hand, one obtains after a sequence of p queries a version space with components of length $(1/2)^{p_1}, (1/2)^{p_2}, \ldots, (1/2)^{p_N}$ where p_i is the number of times the i-th component of the version space has been bisected ($\sum_i p_i = p$); averaging over the distribution of the p_i one obtains

$$\epsilon_g(\text{min. entropy queries}) = \frac{1}{4N}\sum_{\{p_i\}}\frac{p!}{N^p p_1!\cdots p_N!}\sum_{i=1}^{N}\left(\frac{1}{2}\right)^{p_i} = \frac{1}{4}\left[\left(1-\frac{1}{2N}\right)^N\right]^\alpha . \tag{3.14}$$

Comparing (3.13) and (3.14), we see that for $N = 1$, teacher space entropy and generalization error perform equally well as objective functions for query sequence construction, whereas for $N \geq 2$ a query sequence constructed for minimization of teacher space entropy needs to contain more examples than one constructed for minimization of generalization error in order to obtain the same generalization performance. As $N \to \infty$, $(1-1/2N)^N \to \exp(-1/2)$ and thus $-\ln(1/2)/(1/2) = \ln 4 \approx 1.39$ as many examples are needed.

We have seen that query construction both for minimal teacher space entropy and minimal generalization error yields a generalization error which decays exponentially

[1] For the case $N = 1$, the result (3.13) has been rederived and reformulated in terms of probably approximately correct (PAC) learning in [SN95].

with the number of examples normalized by the number of parameters of the high-low rule, $\alpha = p/N$, which is a drastic improvement over the case of random examples where the generalization error only decays algebraically with α. The result for random examples has been given in [SOS92] for $N = 1$ as $\epsilon_g(\text{random examples}) = 1/2(p+2)$; for $N \geq 2$ it generalizes to

$$\begin{aligned}\epsilon_g(\text{random examples}) &= \sum_{\{p_i\}} \frac{p!}{N^p p_1! \cdots p_N!} \frac{1}{4N} \sum_{i=1}^{N} \frac{2}{p_i+2} \\ &= \frac{N}{2(p+1)} \left\{ 1 - \frac{N}{p+2} \left[1 - \left(1 - \frac{1}{N} \right)^{p+2} \right] \right\} \end{aligned} \tag{3.15}$$

which as $\alpha = p/N \to \infty$ gives a decay with $1/2\alpha + O(1/\alpha^2)$ from the inequalities

$$\frac{1}{2(\alpha+2)} \leq \epsilon_g(\text{random examples}) \leq \frac{1}{2\alpha} \left[1 - \frac{1}{\alpha} \left(1 - e^{-\alpha} \right) \right]. \tag{3.16}$$

Our results in this section show that in the learning scenario considered, the teacher space entropy (or for the case of zero temperature Gibbs learning, the equivalent student space entropy) can serve as a useful guideline for query construction and does provide a large increase in generalization performance over random examples, but does not achieve quite as good a performance as query construction for minimum generalization error.

3.3 Linear perceptron

As a second application of the query learning framework set out in Chapter 2 we now consider the linear perceptron. A teacher is specified by a vector $\mathbf{w}_\nu \in \mathbb{R}^N$ such that it yields (in the absence of noise) the output

$$f_\nu(\mathbf{x}) = \frac{1}{\sqrt{N}} \mathbf{w}_\nu^{\mathrm{T}} \mathbf{x} \tag{3.17}$$

for the input $\mathbf{x}$ which is also an N-dimensional vector. Here $\mathbf{w}_\nu^{\mathrm{T}}$ denotes the transpose of $\mathbf{w}_\nu$, so that $\mathbf{w}_\nu^{\mathrm{T}}\mathbf{x}$ is simply the scalar product of $\mathbf{w}_\nu$ and $\mathbf{x}$. Again, we take the problem to be perfectly learnable, and thus assume students to be of the same functional form, with weight vectors $\mathbf{w}_\mathcal{N}$. We will mainly be interested in the thermodynamic limit $N \to \infty$, $p \to \infty$ at constant $\alpha = p/N$; the effects of finite system size N will be discussed in Chapter 7.

For convenience, we consider inputs $\mathbf{x}$ from a spherical distribution,

$$P(\mathbf{x}) \propto \delta(\mathbf{x}^2 - N\sigma_x^2) \tag{3.18}$$

and a Gaussian prior on teacher space

$$P(\mathcal{V}) = \mathcal{G}\Big(\mathbf{0}, \sigma_\nu^2 \mathbf{1}\Big) \propto \exp(-\mathbf{w}_\nu^2/2\sigma_\nu^2). \tag{3.19}$$

Here we have used the notation $\mathcal{G}(\mu, \Sigma)$ for a multivariate Gaussian distribution with mean μ and covariance matrix Σ, and denoted by $\mathbf{1}$ the N-dimensional unit matrix.

In order to fix $P(y|\mathbf{x}, \mathcal{V})$, we consider two forms of noise: Gaussian noise on the output of variance $1/\beta_\nu$, i.e.,

$$P(y|\mathbf{x}, \mathcal{V}) = \mathcal{G}(f_\nu(\mathbf{x}), 1/\beta_\nu) \tag{3.20}$$

and Gaussian noise on the teacher weights, yielding the output corresponding to a perturbed weight vector $\mathbf{w}'_\nu$ distributed as $\mathcal{G}(\mathbf{w}_\nu, \tilde{\sigma}_\nu^2 \mathbf{1})$:

$$P(y|\mathbf{x}, \mathcal{V}) = \langle \delta(y - f_{\nu'}(\mathbf{x})) \rangle_{\mathbf{w}'_\nu} = \mathcal{G}\Big(f_\nu(\mathbf{x}), \tilde{\sigma}_\nu^2 \mathbf{x}^2/N\Big)\,. \tag{3.21}$$

Under the spherical constraint for the inputs, (3.18), this is of the same functional form as (3.20) and need not be considered separately in what follows; all results for noise on the output also hold for noise on the weights with the replacement $\beta_\nu \to 1/\sigma_x^2\tilde{\sigma}_\nu^2$.

Combining (3.19) and (3.20) and using Bayes' formula one obtains that the posterior teacher distribution $P(\mathcal{V}|\Theta^{(p)})$ is a Gaussian distribution $\mathcal{G}(\mathbf{M}_\nu^{-1}\mathbf{a},\ (\beta_\nu \mathbf{M}_\nu)^{-1})$ where we have set

$$\mathbf{M}_\nu = \frac{1}{\beta_\nu \sigma_\nu^2}\mathbf{1} + \frac{1}{N}\sum_{\mu=1}^{p} \mathbf{x}^\mu (\mathbf{x}^\mu)^{\mathrm{T}} \tag{3.22}$$

and

$$\mathbf{a} = \frac{1}{\sqrt{N}}\sum_{\mu=1}^{p} y^\mu \mathbf{x}^\mu. \tag{3.23}$$

The entropy in teacher space is thus simply

$$S_\nu(\Theta^{(p)}) = -\frac{N}{2}\ln\beta_\nu - \frac{1}{2}\ln|\mathbf{M}_\nu| + \text{constant}. \tag{3.24}$$

Its independence of the outputs y^μ in the training set reflects the well known fact that in linear models information-based objective functions always lead to query selection algorithms or 'experimental designs' which can be expressed solely in terms of the input values of the training examples [Mac92c, Fed72].

For calculation of the generalization error we start from the commonly used quadratic error measure

$$e(y, \mathbf{x}, \mathcal{N}) = \frac{1}{2}(y - f_{\mathcal{N}}(\mathbf{x}))^2 \tag{3.25}$$

which yields according to (2.5) the generalization error between student $\mathcal{N}$ and teacher $\mathcal{V}$

$$\epsilon_g(\mathcal{N}, \mathcal{V}) = \frac{1}{2\beta_{\mathcal{V}}} + \frac{\sigma_x^2}{2N}(\mathbf{w}_{\mathcal{N}} - \mathbf{w}_{\mathcal{V}})^2. \tag{3.26}$$

The constant term $1/2\beta_{\mathcal{V}}$ which arises from the noise on the teacher alone will be omitted in the following.

For the training algorithm, we take Gibbs learning with weight decay (see, e.g., [DW93]), specified by a learning temperature $T = 1/\beta$ and weight decay parameter $\tilde{\lambda}$. The corresponding post-training student distribution

$$P(\mathcal{N}|\Theta^{(p)}) \propto \exp\left[-\beta\left(\sum_{\mu=1}^{p}\frac{1}{2}(y^\mu - f_{\mathcal{N}}(\mathbf{x}^\mu))^2 + \frac{\tilde{\lambda}}{2}\mathbf{w}_{\mathcal{N}}^2\right)\right] \tag{3.27}$$

can be thought of as the long time limit of stochastic gradient descent (see, e.g., [SST92]) on an 'energy function' which is the sum of the error on the training set $\sum_\mu \frac{1}{2}(y^\mu - f_{\mathcal{N}}(\mathbf{x}^\mu))^2$ and the weight decay term $\frac{1}{2}\tilde{\lambda}\mathbf{w}_{\mathcal{N}}^2$. The motivation for having a weight decay is to prevent the student from fitting noise in the training data, i.e., to *regularize* it, by penalizing large weight vectors. The size of the weight decay parameter $\tilde{\lambda}$ determines how strong this regularization effect is.

For our linear perceptron students, the Gibbs distribution (3.27) is simply a Gaussian,

$$P(\mathcal{N}|\Theta^{(p)}) = \mathcal{G}\left(\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{a}, (\beta\mathbf{M}_{\mathcal{N}})^{-1}\right). \tag{3.28}$$

Here we have introduced the matrix $\mathbf{M}_{\mathcal{N}}$, defined as

$$\mathbf{M}_{\mathcal{N}} = \tilde{\lambda}\mathbf{1} + \frac{1}{N}\sum_{\mu=1}^{p}\mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}} \tag{3.29}$$

which only differs from $\mathbf{M}_{\mathcal{V}}$ by a multiple of the unit matrix[2] . It follows from (3.29) that $\tilde{\lambda}/\sigma_x^2$ is a dimensionless quantity which we denote by

$$\lambda = \frac{\tilde{\lambda}}{\sigma_x^2} \tag{3.30}$$

[2]In statistics, learning with linear students in the presence of a weight decay is often referred to as 'ridge regression', see, e.g., [HK70]. This is due to the appearance of the diagonal 'ridge' proportional to $\tilde{\lambda}$ in the matrix $\mathbf{M}_{\mathcal{N}}$.

and also simply refer to as the weight decay parameter. The student space entropy is from (3.27)

$$S_{\mathcal{N}}(\Theta^{(p)}) = -\frac{N}{2}\ln\beta - \frac{1}{2}\ln|\mathbf{M}_{\mathcal{N}}| + \text{constant}. \tag{3.31}$$

Averaging over the post-training distribution according to (2.6) and then over the posterior teacher distribution as in (2.7) we get for the average generalization error as a function of the training set

$$\epsilon_g(\Theta^{(p)}) = \frac{\sigma_x^2}{2N}\left[\left(\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{a} - \mathbf{M}_{\mathcal{V}}^{-1}\mathbf{a}\right)^2 + \frac{1}{\beta}\text{tr }\mathbf{M}_{\mathcal{N}}^{-1} + \frac{1}{\beta_{\mathcal{V}}}\text{tr }\mathbf{M}_{\mathcal{V}}^{-1}\right] \tag{3.32}$$

Since a finite training temperature $T = 1/\beta$ only gives a positive definite additive contribution to the generalization error, we restrict ourselves to the case $T = 0$, i.e., $\beta \to \infty$ in the following[3]. We remark that optimal learning in the sense of [Wat93] is obtained as a special case of Gibbs learning (at $T = 0$) by setting the weight decay parameter λ to its optimal value

$$\lambda_{\text{opt}} = \frac{1}{\beta_{\mathcal{V}}\sigma_{\mathcal{V}}^2\sigma_x^2} = \frac{1}{s^2} \tag{3.33}$$

where

$$s = (\beta_{\mathcal{V}}\sigma_{\mathcal{V}}^2\sigma_x^2)^{1/2} = \left(\frac{\langle y^2\rangle_{P(y|\mathbf{x},\mathcal{V})P(\mathbf{x})P(\mathcal{V})} - 1/\beta_{\mathcal{V}}}{1/\beta_{\mathcal{V}}}\right)^{1/2} \tag{3.34}$$

is the root-mean-squared signal to noise ratio of the training examples. $\lambda_{\text{opt}} = 0$ thus corresponds to the limit of a noise free teacher, and a non-zero λ_{opt} measures the typical amount of corruption of noise relative to the average uncorrupted signal; for $\lambda_{\text{opt}} = 1$ noise and signal levels are equal on average. In the special case of optimal weight decay, one sees from (3.22, 3.29) that $\mathbf{M}_{\mathcal{V}} = \mathbf{M}_{\mathcal{N}}$ and hence the generalization error assumes the simple form

$$\epsilon_{g,\text{opt}}(\Theta^{(p)}) = \frac{\sigma_x^2}{2N}\frac{1}{\beta_{\mathcal{V}}}\text{tr }\mathbf{M}_{\mathcal{V}}^{-1}. \tag{3.35}$$

From (3.24) and (2.8) the defining function for query construction for minimal teacher space entropy follows immediately as

$$S_{\mathcal{V}}(\Theta^{(p)}, \mathbf{x}) = S_{\mathcal{V}}(\Theta^{(p)}) + \frac{1}{2}\ln|\mathbf{M}_{\mathcal{V}}| - \frac{1}{2}\ln|\mathbf{M}'_{\mathcal{V}}|, \tag{3.36}$$

where $\mathbf{M}'_{\mathcal{V}}$ is defined as the value of $\mathbf{M}_{\mathcal{V}}$ calculated for the training set $\Theta^{(p)}$ with the

[3]The divergence as $T \to 0$ of the term $(N/2)\ln T$ in the student space entropy (3.31) does not present a problem here since we will only be concerned with entropy differences for which this term is irrelevant.

new example $(\mathbf{x}, y)$ added:

$$\mathbf{M}'_\nu = \mathbf{M}_\nu + \frac{1}{N}\mathbf{x}\mathbf{x}^{\mathrm{T}}. \tag{3.37}$$

The analogous expression for the case of the student space entropy as objective function is obtained simply by replacing $\mathbf{M}_\nu$ by $\mathbf{M}_\mathcal{N}$, whereas the corresponding result for the generalization error, which can be straightforwardly derived from (3.32) and (2.8), is:

$$\begin{aligned}\epsilon_{\mathrm{g}}(\Theta^{(p)}, \mathbf{x}) &= \frac{\sigma_x^2}{2N}\left[\frac{1}{\beta_\nu}\mathrm{tr}\ \mathbf{M}_\nu'^{-1} + \left(\mathbf{M}_\mathcal{N}'^{-1}\mathbf{a}'_\nu - \mathbf{M}_\nu'^{-1}\mathbf{a}'_\nu\right)^2\right. \\ &+ \left.\frac{1}{N\beta_\nu}\left(1 + \frac{1}{N}\mathbf{x}^{\mathrm{T}}\mathbf{M}_\nu^{-1}\mathbf{x}\right)\left(\mathbf{M}_\mathcal{N}'^{-1}\mathbf{x} - \mathbf{M}_\nu'^{-1}\mathbf{x}\right)^2\right]\end{aligned} \tag{3.38}$$

where

$$\mathbf{M}'_\mathcal{N} = \mathbf{M}_\mathcal{N} + \frac{1}{N}\mathbf{x}\mathbf{x}^{\mathrm{T}} \qquad \mathbf{a}'_\nu = \mathbf{a} + \frac{1}{N}\mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{M}_\nu^{-1}\mathbf{a}. \tag{3.39}$$

For the case of optimal weight decay this simplifies to

$$\epsilon_{\mathrm{g,opt}}(\Theta^{(p)}, \mathbf{x}) = \frac{\sigma_x^2}{2N}\frac{1}{\beta_\nu}\mathrm{tr}\ \mathbf{M}_\nu'^{-1}. \tag{3.40}$$

It is to this simpler case that we now turn.

3.3.1 Optimal weight decay

In the case of optimal weight decay, it is straightforward to derive that under the spherical constraint (3.18) the defining functions for query construction for minimal teacher space entropy, (3.36), student space entropy (which can be derived analogously from (3.31)), and generalization error, (3.40), are *all* optimized (i.e., minimized) by choosing the query $\mathbf{x}$ along the direction of an eigenvector of $\mathbf{M}_\nu$ with minimal eigenvalue[4]. For $p < N$, i.e., $\alpha < 1$ this amounts to choosing $\mathbf{x}$ to be perpendicular to the subspace spanned by the previous training inputs $\mathbf{x}^\mu$, $\mu = 1, \ldots, p$, an intuitively obvious result.

Applying this query construction algorithm to generate a sequence of queries, one sees that with each new query the lowest eigenvalue of $\mathbf{M}_\nu$ is increased by σ_x^2. After $p = \alpha N$ queries $\mathbf{M}_\nu$ thus has a $(N\Delta\alpha)$-fold eigenvalue $(\lambda_{\mathrm{opt}} + \lfloor\alpha\rfloor + 1)\sigma_x^2$ and a $N(1 - \Delta\alpha)$-fold eigenvalue $(\lambda_{\mathrm{opt}} + \lfloor\alpha\rfloor)\sigma_x^2$ (we use the decomposition $\alpha = \lfloor\alpha\rfloor + \Delta\alpha$ introduced earlier). Thus from (3.35) one obtains

$$\epsilon_{\mathrm{g,opt}}(\mathrm{queries}) = \frac{1}{2\beta_\nu}G_{\mathrm{Q}}(\lambda_{\mathrm{opt}}) \tag{3.41}$$

[4] A similar result has been found for the more general class of 'additive models' (which include the linear perceptron) in [Pil91, HBH93]. A generalization to the case where more than one query is selected at the same time ('batch query learning') is discussed in [CS70].

with

$$G_{\mathrm{Q}}(\lambda_{\mathrm{opt}}) = \frac{\sigma_x^2}{N}\left\langle \mathrm{tr}\ \mathrm{M}_\nu^{-1}\right\rangle_{P_{\mathrm{Q}}(\Theta^{(p)})} = \frac{\Delta\alpha}{\lambda_{\mathrm{opt}} + \lfloor\alpha\rfloor + 1} + \frac{1-\Delta\alpha}{\lambda_{\mathrm{opt}} + \lfloor\alpha\rfloor}. \tag{3.42}$$

This result can now be compared to the generalization error achieved by training on random examples. We use the results of Krogh *et al.* [KH92a], who have calculated in the limit $N \to \infty$ the function[5]

$$\begin{aligned} G(\lambda_{\mathrm{opt}}) &= \frac{\sigma_x^2}{N}\left\langle \mathrm{tr}\ \mathrm{M}_\nu^{-1}\right\rangle_{P(\Theta^{(p)})} \\ &= \frac{1}{2\lambda_{\mathrm{opt}}}\left(1-\alpha-\lambda_{\mathrm{opt}} + \sqrt{(1-\alpha-\lambda_{\mathrm{opt}})^2 + 4\lambda_{\mathrm{opt}}}\right) \end{aligned} \tag{3.43}$$

which is the analogue of $G_{\mathrm{Q}}(\lambda_{\mathrm{opt}})$ for random examples. Thus, for the average generalization error after training on p random examples, one has

$$\epsilon_{\mathrm{g,opt}}(\text{random examples}) = \frac{1}{2\beta_\nu} G(\lambda_{\mathrm{opt}}). \tag{3.44}$$

The generalization error $\epsilon_{\mathrm{g,opt}}$ as a function of α is shown in figure 3.1 for various values of $\lambda_{\mathrm{opt}} = 1/s^2$, both for random examples and for query sequences. Also shown is the relative reduction in generalization error due to query selection, i.e., the ratio of (3.44) and (3.41) which we denote by

$$\kappa(\alpha) = \frac{\epsilon_{\mathrm{g}}(\text{random examples})}{\epsilon_{\mathrm{g}}(\text{queries})}\ . \tag{3.45}$$

For moderate noise levels (a numerical calculation yields $\lambda_{\mathrm{opt}} \leq 0.92$), the maximum of $\kappa(\alpha)$ is reached at $\alpha = 1$; its height

$$\kappa(\alpha=1) = \frac{1}{2}(1+\lambda_{\mathrm{opt}})\left[\left(1+\frac{4}{\lambda_{\mathrm{opt}}}\right)^{1/2} - 1\right] \tag{3.46}$$

decreases monotonically with λ_{opt}—hence increases with the signal-to-noise ratio s—and is simply given by $(\lambda_{\mathrm{opt}})^{-1/2} = s$ in the limit of small λ_{opt}. Query construction thus yields the greatest improvement of generalization performance for low noise levels. The fact that in the low noise regime the maximum of $\kappa(\alpha)$ is at $\alpha = 1$ can be understood as

[5] Strictly speaking Krogh *et al.* consider a Gaussian distribution for the inputs instead of the spherical distribution (3.18), but in the limit $N \to \infty$ these produce identical results, as can be checked by a direct calculation of the average eigenvalue spectrum of M_ν along the lines of [KO91]. Compare also the discussion in Chapter 7.

follows. For random examples, the average eigenvalue spectrum [KO91] of $\mathbf{M}_\nu$ extends down to $(\lambda_{\text{opt}} + (1 - \sqrt{\alpha})^2)\sigma_x^2$, and this lower spectral limit tends to zero as $\lambda_{\text{opt}} \to 0$ and $\alpha \to 1$. This makes tr $\mathbf{M}_\nu^{-1}$ much larger than for the case of query construction, where at $\alpha = 1$ all eigenvalues of $\mathbf{M}_\nu$ are $(\lambda_{\text{opt}} + 1)\sigma_x^2$. For larger noise levels, the maximum of $\kappa(\alpha)$ shifts to larger integer values of α and has a height which can be bounded by $1 + 4/\lambda_{\text{opt}}$ and which thus tends to 1 in the limit of large noise levels, $\lambda_{\text{opt}} \to \infty$.

The plots in figure 3.1 suggest that independently of the value of λ_{opt}, $\kappa(\alpha)$ tends to 1 as $\alpha \to \infty$, which means that for a sufficiently large number of examples, the relative improvement in generalization error that can be obtained from queries as compared to random examples tends to zero. This can be confirmed by an asymptotic expansion of $\kappa(\alpha)$ which yields

$$\kappa(\alpha) = 1 + \frac{1}{\alpha} + O\left(\frac{1}{\alpha^2}\right). \tag{3.47}$$

The above result is in stark contrast to the results for the high-low game obtained above and similar results for the binary perceptron [SOS92, FSST93], where the asymptotic behaviour of $\kappa(\alpha)$ for large α is

$$\kappa(\alpha) \propto \frac{1}{\alpha}(\exp(-c\alpha))^{-1} \tag{3.48}$$

for some positive constant c, which clearly tends to infinity as $\alpha \to \infty$. A plausible explanation for this qualitative difference might be that in the limit of a noise free teacher, N examples are actually enough to specify a teacher linear perceptron completely, so that beyond $\alpha = 1$ one is trying to reduce generalization error due to noise; by contrast, for high-low or the binary perceptron, the teacher cannot be uniquely specified by any finite set of examples even in the noise free limit. In this sense, the high-low game and the binary perceptron are 'non-invertible' for any finite α, and thus by querying the average amount of information about the teacher that can be gained from each new training example can be kept finite as $\alpha \to \infty$. This property was shown in [FSST93] to be a sufficient condition for exponentially decaying generalization error, at least for the specific query filtering algorithm considered there. For the linear perceptron, on the other hand, the information available about the teacher is, loosely speaking, 'exhausted' at $\alpha = 1$ and the information that can be gained from each new training example tends to zero as $\alpha \to \infty$.

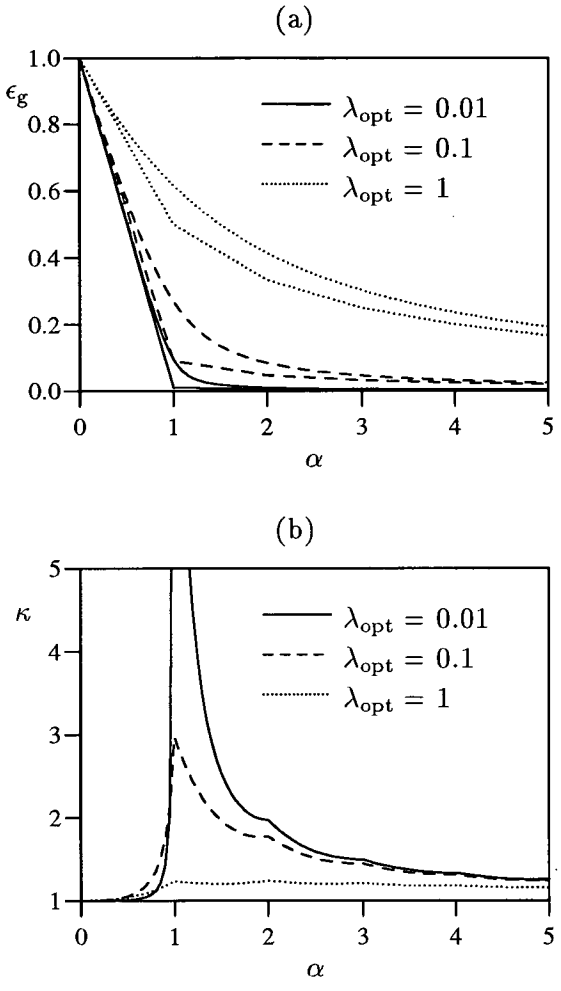

Figure 3.1. (a) Generalization error $\epsilon_g(\alpha)$ achieved by training a linear perceptron on αN *random examples* (higher, 'smooth' curves) and on the same number of examples generated by a sequence of *queries* (lower curves), in units of $\sigma_x^2\sigma_\nu^2/2$. The weight decay λ is assumed to be set to its optimal value λ_{opt}; hence minimum generalization error and minimum entropy queries are identical. The values of the (squared) teacher noise-to-signal ratio $\lambda_{opt} = 1/s^2$ are 0.01, 0.1 and 1. (b) Relative improvement in generalization error due to querying, $\kappa(\alpha)$, defined as the ratio of the values of ϵ_g for random examples and for query sequences.

3.3.2 Non-optimal weight decay

We now turn to the case of non-optimal weight decay, where the weight decay parameter λ in the training algorithm is not set to the optimal value determined by the signal-to-noise ratio of the teacher as in (3.33).

We consider first query construction for minimization of the entropy in teacher space (3.24). Since this quantity is independent of λ, the query construction algorithm remains the same as for optimal weight decay. (The same conclusion holds for the case of minimization of the student space entropy.) Therefore, as in the preceding section, query construction is effectively non-sequential in the sense that it only depends on the previous input $\mathbf{x}^\mu$ and not on the corresponding outputs. This simplifies the calculation of the average generalization error after a sequence of p minimum entropy queries, i.e., the average of (3.32) over the training set distribution obtained by querying, $P_{\mathrm{Q}}(\Theta^{(p)})$. Using the decomposition (2.27), one can first perform the average over the y^μ to derive

$$\epsilon_{\mathrm{g}}(\text{min. entropy queries}) = \frac{\sigma_x^2\sigma_\nu^2}{2}\left[\lambda_{\mathrm{opt}}G_{\mathrm{Q}}(\lambda) + \lambda(\lambda_{\mathrm{opt}} - \lambda)\frac{dG_{\mathrm{Q}}(\lambda)}{d\lambda}\right] \tag{3.49}$$

where the average over the $\mathbf{x}^\mu$ is taken care of in the definition of the function $G_{\mathrm{Q}}(\cdot)$ in (3.42). (A formal derivation of this result—in a more general scenario—can be found in Section 4.6.) The analogue of (3.49) for the case of random examples, as derived in [KH92a], is obtained simply by replacing $G_{\mathrm{Q}}(\cdot)$ with $G(\cdot)$. The resulting values of $\kappa(\alpha)$ are plotted in figure 3.2 for various values of λ and λ_{opt}. The most striking feature is that now κ can actually assume values smaller than 1, implying that minimal entropy query construction leads to a *higher* generalization error than random examples, a seemingly counter-intuitive result. It can be checked numerically, however, that $\kappa < 1$ occurs only when λ is smaller than the optimal value λ_{opt}, combined with high teacher noise levels $\lambda_{\mathrm{opt}} \geq 2$ and values of α for which the underlying rule is only just beginning to be learned, in that ϵ_{g} is still more than over 80% of its value at $\alpha = 0$, i.e., before any training examples were presented. In these cases the training algorithm is *over-confident* in that it underestimates the amount of noise in the training examples, making the entropy reduction or information gain a spurious indicator of an improvement in generalization ability. The correlation between reductions in entropy and generalization error is recovered as soon as α is large enough for the generalization error to be significantly smaller than at $\alpha = 0$; in the limit of an infinite number of training examples, one has

$$\kappa(\alpha) = 1 + \frac{1}{\alpha} + \frac{1}{\alpha^2}\left[1 - 2\lambda_{\mathrm{opt}} + 2\frac{(\lambda - \lambda_{\mathrm{opt}})^2}{\lambda_{\mathrm{opt}}} - \Delta\alpha(1 - \Delta\alpha)\right] + O\left(\frac{1}{\alpha^3}\right) \tag{3.50}$$

so that κ is again greater than one for large α. The last result also shows that for fixed, large α, κ increases with increasing $(\lambda - \lambda_{\rm opt})^2$, i.e., with the degree of mismatch between the training algorithm and the actual learning problem at hand.

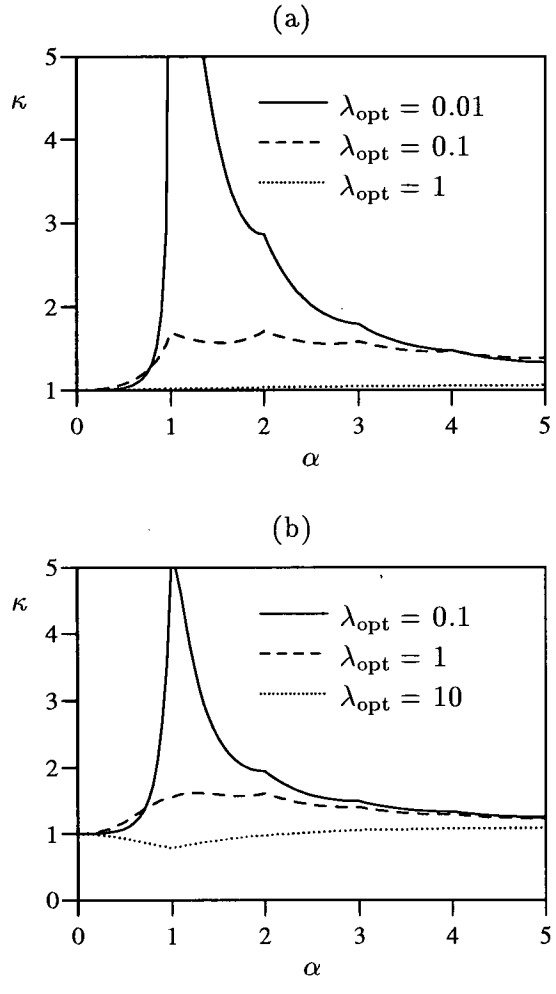

Figure 3.2. $\kappa(\alpha)$ for minimum entropy queries for the case of non-optimal weight decay $\lambda \neq \lambda_{\rm opt}$. (a) 'Under-confident', i.e., unnecessarily large weight decay $\lambda = 10\,\lambda_{\rm opt}$, for $\lambda_{\rm opt}$=0.01, 0.1 and 1. (b) 'Over-confident', i.e., inappropriately small weight decay $\lambda = \lambda_{\rm opt}/10$, for $\lambda_{\rm opt}$=0.1, 1 and 10. Notice that in the last case values of $\kappa(\alpha) < 1$ appear, which means that a sequence of minimum entropy queries can lead to a higher generalization error than the same number of random examples.

Now we consider for comparison the performance of query construction for minimal generalization error as defined by minimization of (3.38). Since we have not been able to perform this minimization analytically for the general case, we restrict our attention

to the special case in which $\mathbf{a}$ is an eigenvector of the matrix $\mathbf{M}_\nu$, and to the limit of a noise free teacher, $\lambda_{\text{opt}} \to 0$. If we also assume that $\mathbf{M}_\nu$ has full rank, i.e., that at least N training examples with linearly independent input vectors have been presented, then only the second term in (3.38) survives:

$$\epsilon_g(\Theta^{(p)}, \mathbf{x}) = \frac{\sigma_x^2}{2N} \left(\mathbf{M}_\mathcal{N}'^{-1} \mathbf{a}_\nu' - \mathbf{M}_\nu'^{-1} \mathbf{a}_\nu' \right)^2 \tag{3.51}$$

Setting

$$\boldsymbol{\Delta}' = \mathbf{M}_\mathcal{N}'^{-1} \mathbf{a}_\nu' - \mathbf{M}_\nu'^{-1} \mathbf{a}_\nu' \tag{3.52}$$

$$\boldsymbol{\Delta} = \mathbf{M}_\mathcal{N}^{-1} \mathbf{a} - \mathbf{M}_\nu^{-1} \mathbf{a} \tag{3.53}$$

one can derive that

$$\boldsymbol{\Delta}' = \boldsymbol{\Delta} - \mathbf{M}_\mathcal{N}'^{-1} \frac{1}{N} \mathbf{x} \mathbf{x}^T \boldsymbol{\Delta} \tag{3.54}$$

and under the above assumptions and the spherical constraint (3.18) one finds that $\epsilon_g(\Theta^{(p)}, \mathbf{x})$ is minimized by choosing $\mathbf{x}$ along $\boldsymbol{\Delta}$ and hence along $\mathbf{a}$. This makes intuitive sense: Under our assumption that $\mathbf{a}$ is an eigenvector of $\mathbf{M}_\nu$, $\mathbf{a}$ is proportional to $\mathbf{M}_\nu^{-1}\mathbf{a}$ which is in fact the true teacher, $\mathbf{w}_\nu$, due to the assumptions of full rank of $\mathbf{M}_\nu$ and $\lambda_{\text{opt}} \to 0$. Querying along $\mathbf{a}$ therefore yields the largest possible signal $y = \mathbf{w}_\nu^{\mathrm{T}} \mathbf{x}/\sqrt{N} = \sigma_x |\mathbf{w}_\nu|$ and hence reduces the generalization error (3.51) (which is due to the mismatch between $\lambda \neq 0$ and $\lambda_{\text{opt}} = 0$) most quickly. We remark that $\mathbf{x} \propto \mathbf{a}$ is a truly sequential query construction criterion since it involves, through $\mathbf{a}$, the previous outputs. This is in contrast to query construction for minimum entropy where the choice of each query is determined solely by the preceding inputs, as discussed above.

Let us now apply the query construction criterion $\mathbf{x} \propto \mathbf{a}$ to a simple case where the above assumptions are fulfilled. Namely, consider a noise free teacher $\mathbf{w}_\nu$ and a training set of N examples generated by minimum entropy query construction, i.e., containing N mutually orthogonal input vectors and thus having $\mathbf{M}_\nu = \sigma_x^2 \mathbf{1}$ and $\mathbf{a} = \sigma_x^2 \mathbf{w}_\nu$. Querying at $\mathbf{x} = \mathbf{a}(N\sigma_x^2/\mathbf{a}^2)^{1/2}$ then yields a new matrix $\mathbf{M}_\nu' = \mathbf{M}_\nu + \mathbf{a}\mathbf{a}^{\mathrm{T}}(\sigma_x^2/\mathbf{a}^2)$ and a new vector $\mathbf{a}' = \mathbf{a} + y\mathbf{x}/\sqrt{N} = \mathbf{a} + \sigma_x^2 \mathbf{w}_\nu = 2\mathbf{a}$. $\mathbf{a}'$ is again an eigenvector of $\mathbf{M}_\nu'$ and hence the next minimum generalization error query will have to be selected along $\mathbf{a}'$, i.e., again along $\mathbf{a}$. $\delta p = N\delta\alpha$ such queries in sequence generate a matrix $\mathbf{M}_\nu$ and a vector $\mathbf{a}$ with $\mathbf{M}_\nu \mathbf{a} = (\delta p + 1)\sigma_x^2 \mathbf{a}$ and $\mathbf{a} = \sigma_x^2 \mathbf{w}_\nu (1 + \delta p)$, thus leading to a generalization error (using (2.12) and (3.26))

$$\epsilon_g(\Theta^{(N+\delta p)}, \mathcal{V}) = \frac{\sigma_x^2}{2N} \left(\mathbf{M}_\mathcal{N}^{-1} \mathbf{a} - \mathbf{w}_\nu \right)^2 = \frac{\sigma_x^2 \sigma_\nu^2}{2} \frac{\mathbf{w}_\nu^2}{N\sigma_\nu^2} \frac{\lambda^2}{(\lambda + N\delta\alpha + 1)^2} \tag{3.55}$$

This result contains the size of the perceptron, N, and for fixed $\delta\alpha$ converges to zero in the thermodynamic limit $N \to \infty$, implying that in this limit ϵ_g expressed as a function of α has a step discontinuity at $\alpha = 1$. This result in itself, due to the limiting assumptions that we had to make, is probably less important than a more general conclusion which can be drawn: For query construction even in purely linear learning problems, maximizing information gain is not necessarily identical to minimizing generalization error, and to obtain the optimal generalization performance one will generally have to resort to truly sequential query selection.

3.4 General issues

In the preceding sections we have focussed our attention on the generalization performance obtained from query sequences. We now turn to two other interesting aspects of query learning: Single queries and locally vs. globally optimal query construction. We again investigate them for the two example learning scenarios considered above, confining ourselves to query construction for minimum entropy in the case of the linear perceptron in order to keep things analytically tractable.

3.4.1 Single queries

We refer to a single query which is constructed on the basis of an existing training set of random examples as 'isolated'. It is then natural to ask the question: How does the improvement in generalization capability due to an isolated query, i.e., the decrease in generalization error, compare with that due to a query in a query sequence and that due to a random example? The first comparison concerns the question of how the performance of a single query depends on the previous learning history, i.e., on the method by which the previous training examples have been generated (randomly or by querying). It is not entirely obvious whether for answering this question the relative or the absolute decrease in generalization error is the relevant quantity, and we shall consider both of these options below.

High-low

As derived in section 2.5, the average generalization error after a single query can simply be calculated by averaging the function $\epsilon_g(\Theta^{(p)}, x)$ over the respective query construction distribution $P_Q(x|\Theta^{(p)})$. For the high-low game, we thus find from (3.11) that a single query constructed for minimum generalization error and minimum teacher

space entropy, respectively, reduces the generalization error by

$$\Delta\epsilon_g(1 \text{ min. generalization error query}) = \frac{1}{8N} \max_i (x_{R,i} - x_{L,i}) \tag{3.56}$$

and

$$\Delta\epsilon_g(1 \text{ min. entropy query}) = \frac{1}{8N} \frac{1}{N} \sum_{i=1}^{N} (x_{R,i} - x_{L,i}) = \frac{1}{2N} \epsilon_g(\Theta^{(p)}). \tag{3.57}$$

Let us first consider the dependence of these results on the learning history. From (3.57), a minimum entropy query reduces the generalization error by an amount proportional to the generalization error before querying—which will therefore be large for previous training examples generated randomly and smaller if queries have been used—, making the relative improvement independent of the learning history. Comparing (3.56) and (3.57) one sees that a minimum generalization error query provides, as expected, a greater reduction (for $N \geq 2$; for $N = 1$ the two query construction algorithms are equivalent) in generalization error than a minimum entropy query, which is also more strongly dependent on the learning history. For previous training examples generated using minimum generalization error queries, the maximum in (3.56) is $(1/2)^{\lfloor \alpha \rfloor}$ as follows from the discussion before equation (3.13), giving an absolute decrease in generalization error decaying exponentially with the number of examples; from (3.13), the corresponding relative decrease is $(1 - \frac{\Delta\alpha}{2})^{-1}/2N$ and thus between $1/2N$ (the value for a minimum entropy query) and $1/N$. The difference to the case of previous random training examples is most clearly exhibited for $N \to \infty$, because in this limit it follows from the well-known combinatorial 'collector's problem' (see, e.g., [Fel70]) that for any α there is with probability one at least one component of the version space for which no training examples exist at all, making the maximum in (3.56) equal to 1 and yielding an absolute decrease in generalization error of $1/8N$, independently of α. From (3.16) the corresponding relative decrease[6] is $(\alpha + O(1))/4N$, greater by a factor of $O(\alpha)$ than for previous training examples generated by queries.

We now compare isolated queries to random examples. From (3.15), one finds that the absolute decrease in generalization error due to a random example after previous random training examples is given by $1/(2N\alpha^2) + O(1/N\alpha^3)$, yielding a relative decrease of $1/N\alpha + O(1/N\alpha^2)$. As $\alpha \to \infty$, this tends to zero, reflecting the fact

[6] For finite but large N, this expression can be estimated to be valid for values of α much smaller than $\ln N$, from results for the mean waiting time in the 'collector's problem' (see e.g., [Fel70]); this ensures that the relative decrease $(\alpha + O(1))/4N$ is always smaller than one as it has to be.

that the information carried by new random examples becomes more and more redundant. By comparison, for an isolated minimum entropy query we found above that the relative decrease in generalization error is $1/2N$, showing that minimum entropy query construction successfully avoids this redundancy. For a minimum generalization error query and in the limit $N \to \infty$, the relative generalization error decrease of $(\alpha + O(1))/4N$ is still larger, by a factor of $\alpha/2 + O(1)$, than the relative decrease achieved by a minimum entropy query, implying that minimum generalization error query construction selects among all queries providing non-redundant information the one with the greatest potential for improving generalization.

Linear perceptron

We now turn to the case of the linear perceptron. As pointed out before, we consider only the case of query construction for minimum (teacher or student space) entropy. In this case the reduction in generalization error due to a single query is particularly easy to calculate, since only the change in $G(\lambda)$ and $dG(\lambda)/d\lambda$ (or $G_Q(\lambda)$ and $dG_Q(\lambda)/d\lambda$, respectively) needs to be worked out. One obtains a result which in general depends on the learning history through the minimal eigenvalue of $\mathbf{M}_N$, which we write as $(\lambda + \lambda_{\min})\sigma_x^2$. For $\alpha < 1$, however, there is no such dependence since one always has $\lambda_{\min} = 0$ because the correlation matrix $\sum_\mu \mathbf{x}^\mu(\mathbf{x}^\mu)^T$ does not have full rank. In the case of previous random examples [KO91] one obtains, using the fact that in the thermodynamic limit $N \to \infty$ the eigenspectrum of $\mathbf{M}_N$ is self-averaging,

$$\lambda_{\min} = \begin{cases} 0 & \text{for } \alpha \leq 1 \\ (\sqrt{\alpha} - 1)^2 & \text{for } \alpha > 1. \end{cases} \tag{3.58}$$

For a query in a query sequence, one simply has $\lambda_{\min} = \lfloor \alpha \rfloor$ as discussed in section 3.3. Using these values of $\lambda_{\min}$, one finds that a query in a sequence generally leads to an absolute reduction in generalization error less or equal to that due to an isolated query. The exception is the case of over-confidence and high noise level, where at finite α a query in a sequence can reduce the generalization error by a larger amount than an isolated query. Asymptotically, a query in a query sequence reduces the generalization error by $(1/2\beta_\nu N)\alpha^{-2}(1+O(\alpha^{-1}))$, which corresponds to a relative decrease of $1/N\alpha + O(1/N\alpha^2))$, whereas for an isolated query both the absolute and relative reductions are bigger by a factor of $1 + 4\alpha^{-1/2} + O(\alpha^{-1})$.

Now let us compare isolated queries to random examples. The reduction in generalization error due to a single random example can be straightforwardly obtained by differentiating the analogue of (3.49) for random examples with respect to $N\alpha$, and is

shown in figure 3.3 along with the corresponding results for isolated queries[7]. It can be seen that the trend of the comparison between query sequences and sequences of random examples discussed in section 3.3 is mirrored in the result for isolated queries and single random examples: For optimal weight decay, an isolated query always performs better than a random example (maximally, it reduces the generalization error by 5 times as much as a random example, which is achieved at $\alpha = 9/4$ in the limit $\lambda_{\text{opt}} \to 0$) whereas for non-optimal, over-confident weight decay and small α it can perform worse. Asymptotically, the reduction due to an isolated query is greater by a factor of $1 + 4\alpha^{-1/2} + O(\alpha^{-1})$ than that due to an additional random example.

To summarize our discussion of single minimum entropy queries for the linear perceptron, we have found quite a different behaviour than for the high-low game, as would have been expected from the significant differences between the two systems regarding the efficacy of query sequences. Whereas minimum entropy queries in the high-low game—whether isolated or in a query sequence—lead to a relative improvement in generalization error which remains finite as $\alpha \to \infty$, the relative improvements in the linear perceptron decay towards zero roughly as $1/\alpha$ and to lowest order in $1/\sqrt{\alpha}$ are identical to those obtained from random examples. Again, we argue that the reason for this qualitative difference is that for large α learning in the linear perceptron is mainly learning against noise, for which queries are not significantly more useful than random examples.

We found for both high-low and the linear perceptron with optimal weight decay that the *absolute* reduction in generalization error is always larger for an isolated query than for one in a query sequence, whether we consider minimum generalization error or minimum entropy queries. This result makes intuitive sense because, if the previous training examples have already been generated by queries, one expects there to be less scope for reducing the generalization error by another query. We speculate that this might be more generally valid in learning problems where the training algorithm is well-matched to the learning environment, i.e., the posterior teacher distribution. For the linear perceptron with non-optimal weight decay, i.e., a poorly matched training algorithm, we find that the above does hold at least asymptotically (as $\alpha \to \infty$) for minimum entropy queries, but not necessarily for finite α. In terms of the *relative* reduction in generalization error, we observe that for large α an isolated query still

[7] Note in figure 3.3(c) that for over-confident weight decay and small α, negative values of $\Delta\epsilon_g$ can occur, corresponding to an increase of the generalization error as more training examples are received. This seemingly pathological behaviour is due to the fact that in this regime the student learns mainly the noise in the training data. An increase in the number of training examples thus effectively corresponds to 'more noise' and hence worse generalization.

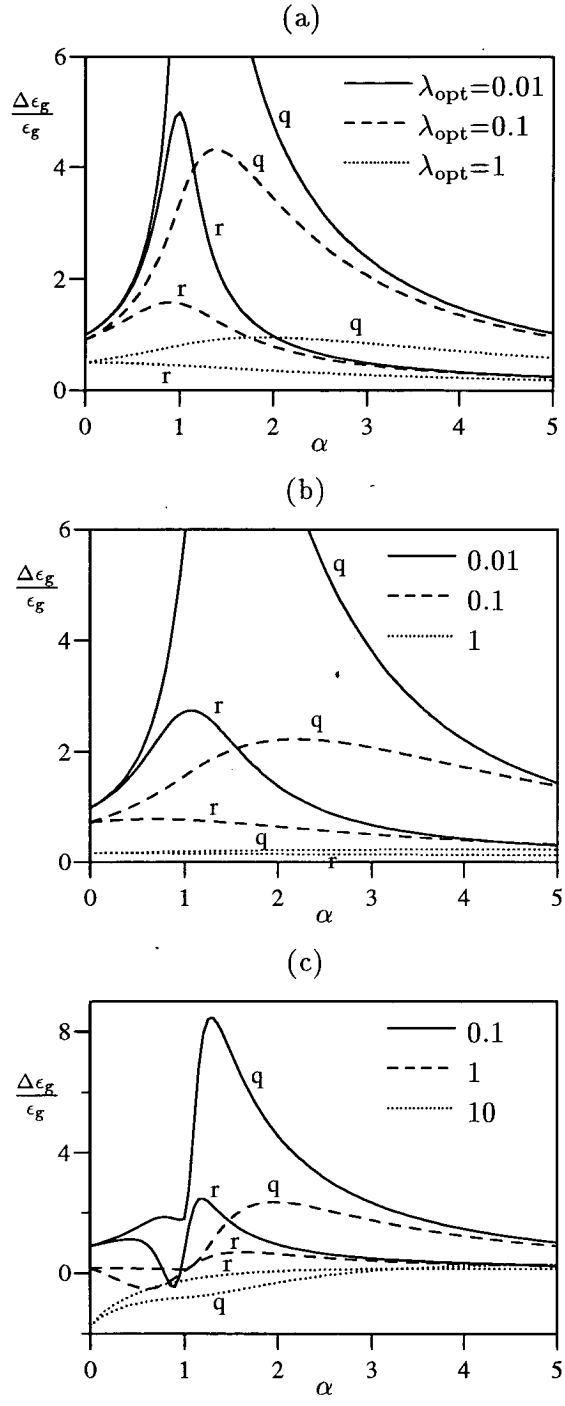

Figure 3.3. Relative reduction in generalization error (in units of $1/N$) due to an isolated minimum entropy query and a random example, respectively. (a) Optimal weight decay $\lambda = \lambda_{opt}$; λ_{opt}=0.01, 0.1, 1. (b) 'Under-confident' weight decay $\lambda = 10\,\lambda_{opt}$; λ_{opt}=0.01, 0.1, 1. (c) 'Over-confident' weight decay $\lambda = \lambda_{opt}/10$; λ_{opt}=0.1, 1, 10. Notice that in the last case for high teacher noise level (λ_{opt}=10) a minimum entropy query can reduce the generalization error less than an additional random example.

performs better than one in a query sequence, whereas for small α it can be shown that one can also have the reverse relationship between the two.

3.4.2 Locally vs. globally optimal query construction

All our considerations so far have been based on the assumption that query construction can be viewed as a 'greedy' optimization of some appropriate objective function. If one is looking for query construction algorithms which are applicable independently of the total number of queries that will eventually be used in the learning process, this approach, which we shall call 'locally optimal query construction', is perfectly reasonable. If the total number of allowed queries were known, one might want to optimize the query construction algorithm 'globally' in order to achieve the optimum of the relevant objective function after learning from the specified number of queries and the corresponding outputs (for a formal definition see, e.g., [DeG62]). It is the aim of the present section to compare the performance of globally and locally optimal query construction, with the goal of assessing the loss in performance that one incurs if one restricts oneself to locally optimal query construction. We emphasize that what we mean by globally optimal query construction is not identical to what is normally referred to as 'statistical' (or 'exact') design in the statistics literature, where all queries are chosen before any outputs are received; globally optimal query construction shares with this approach the fact that the total number of training examples is fixed, but sequentially selects each new query on the basis of all preceding training examples, inputs and outputs alike. We also stress that the major disadvantage of globally optimal query construction is that it is tied to the specific number p of queries that is considered; in fact, one must expect that a globally optimal sequence of p queries cannot be augmented by more queries later without leading to suboptimal generalization performance.

We shall first consider the question of possible equivalence of locally and globally optimal query construction in terms of the final value of the relevant objective function that they achieve. Intuitively, one expects that if a globally optimal sequence of p queries can always be augmented by another query to give a globally optimal sequence of $p+1$ queries, then globally optimal query sequences can be constructed using a local, i.e., step-by-step approach. This criterion can be formalized and one can check that it does indeed hold for the high-low game, whether generalization error or entropy is used as the objective function for query construction; thus locally and globally optimal query construction perform equally well. For the linear perceptron, however, the situation is different, as we now show. Consider the case of optimal weight decay, where the

generalization error is given by (3.35). From the convexity inequality

$$\frac{1}{N}\mathrm{tr}\ \mathbf{M}_\nu^{-1} \geq \left(\frac{1}{N}\mathrm{tr}\ \mathbf{M}_\nu\right)^{-1} = (\sigma_x^2(\lambda_{\mathrm{opt}}+\alpha))^{-1} \tag{3.59}$$

one has the bound

$$\epsilon_{\mathrm{g,opt}}(\Theta^{(p)}) \geq \frac{1}{2\beta_\nu}\frac{1}{\lambda_{\mathrm{opt}}+\alpha}. \tag{3.60}$$

For $\alpha < 1$, this bound can be tightened using the fact that $\mathbf{M}_\nu$ must have at least $N(1-\alpha)$ eigenvalues of size $\lambda_{\mathrm{opt}}\sigma_x^2$:

$$\epsilon_{\mathrm{g,opt}}(\Theta^{(p)}) \geq \frac{1}{2\beta_\nu}\left(\frac{\alpha}{\lambda_{\mathrm{opt}}+1}+\frac{1-\alpha}{\lambda_{\mathrm{opt}}}\right) \tag{3.61}$$

A result from [GP82] shows that the above inequalities can be made into equalities by appropriate choice of the $\mathbf{x}^\mu$, so that globally optimal query construction for minimum generalization error saturates the bounds (3.60),(3.61). Comparing this with the result (3.41),(3.42) for locally optimal query construction, one sees that the two achieve identical performance for $\alpha \leq 1$ and for the integer values $\alpha = 2, 3, \ldots$, but that for all other values of α locally optimal query construction performs worse. This can also be read off from figure 3.4 which shows the ratio ρ of the generalization error achieved by globally and locally optimal query selection as a function of α, for different values of λ_{opt}. This ratio attains its minimum of $8/9$ at $\alpha = 3/2$ for $\lambda_{\mathrm{opt}} \to 0$, and is for large α given by $1-\Delta\alpha(1-\Delta\alpha)/\alpha^2+O(\alpha^{-3})$, showing that although locally optimal query construction in general performs worse for finite α, it 'catches up' again with globally optimal query construction asymptotically.

To illustrate the reason for the difference between locally and globally optimal query selection, we consider briefly the case $N = 2$, $p = \alpha N = 3$. The locally optimal query construction algorithm selects the first two queries $\mathbf{x}^1$ and $\mathbf{x}^2$ orthogonal to each other and the third one randomly, leading to a (2×2) correlation matrix $(1/N)\sum_\mu \mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}}$ with eigenvalues σ_x^2 and $2\sigma_x^2$. Globally optimal query selection selects the three queries $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$ at angles of $120°$ to each other, making the eigenvalues of the correlation matrix both equal to $3/2\sigma_x^2$ and thus saturating the bound (3.60). This example also illustrates another point which was mentioned above: for an unknown total number of training examples globally optimal query construction is not normally a good idea. If, after having chosen the globally optimal queries for $p = 3$, we were allowed an additional query, we would end up with a correlation matrix with eigenvalues $3/2\sigma_x^2$, $5/2\sigma_x^2$ which does not saturate the bound (3.60), whereas the locally optimal query construction algorithm would select the fourth query orthogonal to $\mathbf{x}^3$, yielding the

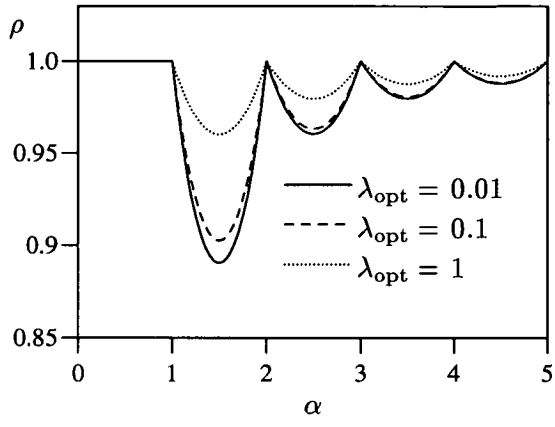

Figure 3.4. Ratio ρ of generalization error achieved by globally vs. locally optimal query sequences, for the linear perceptron with optimal weight decay $\lambda = \lambda_{\mathrm{opt}}$. Values of λ_{opt} are 0.01, 0.1, 1. The globally optimal query sequence leads to a generalization error which is at most smaller by a factor of 8/9 (at $\alpha = 3/2$ and for $\lambda_{\mathrm{opt}} \to 0$) than that of the locally optimal query sequence.

optimal correlation matrix with two eigenvalues of $2\sigma_x^2$.

Summarizing, we have found that in general locally optimal query construction will perform worse than its globally optimal equivalent, but that, at least for the two learning problems we have considered, the differences in performance, if they exist, become negligibly small for large values of α. Overall, the advantage of locally optimal query construction algorithms, namely their applicability whatever the total number of training examples is, thus seems to compensate well for the loss in performance compared to globally optimal query construction. It remains a matter of further research to establish how general this result is.

3.5 Conclusion

In the present chapter, we have explored the differences between the objective functions entropy and generalization error in two learning scenarios, the high-low game and the linear perceptron. Evaluating the average generalization ability obtained after training on examples generated by a sequence of queries and comparing to learning from random examples, we have found strong qualitative differences in the efficacy of query learning. These differences are due to the different structure of the underlying rule in the two scenarios: In the high-low game with its nonlinear and 'non-invertible' rule, the

generalization error decays exponentially with α, the number of examples normalized by the number of parameters in the system, which is a dramatic improvement over the asymptotic decay with $1/\alpha$ for random examples. For the linear perceptron with its purely linear rule, on the other hand, we have found that the relative reduction in generalization error due to querying is much less pronounced and indeed is given by a reduction factor $\kappa(\alpha)$ as small as $1 + 1/\alpha$ for large α. We have related this qualitative difference to the fact that in the high-low game query construction can realize a finite information gain per training example as $\alpha \to \infty$, whereas for the linear perceptron the maximal information gain per example tends to zero in this limit, the available information essentially being 'exhausted' at $\alpha = 1$.

As to the difference between entropy and generalization error as objective functions for query construction we have found that most of the time the entropy can serve as a useful guideline for query construction, but does not achieve the optimal performance obtained by query construction for minimum generalization error. For the case of the linear perceptron, we have observed that if the training algorithm is ill-matched to the details of the learning problem at hand (although the rule was still assumed to be perfectly learnable), minimum entropy queries can actually lead to a higher generalization error than random examples, but only if the teacher is very noisy, the training algorithm is over-confident (i.e., under-estimates the noise level) and the number of training examples is so low that the rule is only just beginning to be learned.

In section 3.4, we have considered the performance of isolated queries, i.e., queries which follow a training set of random examples, and compared them to single queries in a query sequence and single random examples. We have observed in our two example learning scenarios that for large α an isolated query leads to a greater (absolute) reduction in generalization error than a query in a query sequence and speculate that this result, as well as its analogue for the relative reduction in generalization error, might hold more generally. We have also investigated how much one could improve on the approach we have adopted in this paper, namely locally optimal query construction, i.e., 'greedy' optimization of the objective function at each step, by allowing global optimization of the query construction algorithm for a fixed total number of queries. We have found that the two methods will not in general be equivalent, but we expect from the results for our two example systems that the difference in performance will be small for many learning problems, especially for large numbers of training examples.

It should be clear from the above that much remains to be done in the field of query learning. In particular, more complicated rules need to be analysed and scenarios with imperfectly learnable rules considered. Some of these issues will be explored in the following chapters.

Chapter 4

Imperfectly learnable problems: Linear students

Abstract

We study the generalization performance achieved by query learning in situations where the student cannot learn the teacher perfectly. As a simple model scenario of this kind, we consider a linear perceptron student learning a general nonlinear perceptron teacher. Two kinds of queries for minimum entropy are investigated: Minimum *student space* entropy (MSSE) queries, which are appropriate if the teacher space is unknown, and minimum *teacher space* entropy (MTSE) queries, which can be used if the teacher space is assumed to be known, but a student of a simpler form has deliberately been chosen. We find that for MSSE queries, the structure of the student space determines the efficacy of query learning. MTSE queries, on the other hand, which we investigate for the extreme case of a binary perceptron teacher, lead to a higher generalization error than random examples, due to a lack of feedback about the progress of the student in the way queries are selected.

4.1 Introduction

In the previous chapter, query learning has been investigated for perfectly learnable problems, where student and teacher space are identical. In particular, we considered queries selected to minimize the entropy (i.e., maximize the information gain) in the parameter space of the student or teacher. Their effect on generalization performance was found to depend qualitatively on the structure of the input-output mapping to be learned. For the linear perceptron, for example, we obtained a relative reduction in generalization error compared to learning from random examples which becomes

insignificant as the number of training examples, p, tends to infinity. For high-low, on the other hand, minimum entropy queries result in a generalization error which decays exponentially as p increases—a marked improvement over the much slower algebraic decay with p in the case of random examples. Similar results have also been obtained for the binary perceptron [SOS92, FSST93].

We now extend our investigation to *imperfectly learnable* problems, where the student can only approximate the teacher, but not learn it perfectly. This implies that student and teacher space are different, and we therefore now have to distinguish between minimum *student space* entropy and minimum *teacher space* entropy (MSSE/MTSE) queries. In practical situations, imperfectly learnable problems can arise for two reasons: Firstly, the teacher space (i.e., the space of models generating the data) might be unknown. Because the teacher space entropy is then also unknown, MSSE (and not MTSE) queries have to be used for query learning. Secondly, the teacher space may be known, but a student of a simpler structure might have deliberately been chosen in order to facilitate or speed up training, for example. In this case, MTSE queries could be employed as an alternative to MSSE queries. The motivation for doing this would be strongest if, as in the learning scenario that we consider below, it is known from analyses of perfectly learnable tasks that the structure of the teacher space allows more significant improvements in generalization performance from query learning than the structure of the student space.

With the above motivation in mind, we investigate in this chapter the performance of both MSSE and MTSE queries for a simple imperfectly learnable problem, in which a linear perceptron student is trained on data generated by a general nonlinear perceptron teacher. Both student and teacher are specified by an N-dimensional weight vector with real components, and we will consider the thermodynamic limit $N \to \infty$, $p \to \infty$, with the normalized number of training examples, $\alpha = p/N = \text{const}$.

Let us comment briefly on the practical relevance of the analysis of a learning scenario with a linear student. While it is true that in most applications of neural networks, for example, nonlinearities play an important role, many fundamental insights into supervised learning have been obtained from analyses of linear model systems, where analytical solutions can be obtained (see, e.g., [KH92a, DW93, BSS95, LTS90, BS94, BH95]). Furthermore, it has been argued that the properties of networks with smooth nonlinearities can often be related to those of linear models by means of a local linearization procedure [SST92, KH92b, BKO93]. It is therefore reasonable to expect that at least qualitatively, the results of our analysis will to some extent carry over to learning with more realistic feedforward neural networks.

The remainder of this chapter is structured as follows: In Section 4.2, we formally

define the learning scenario to be investigated. The generalization error for learning from random examples and from MSSE queries is calculated in Section 4.3; MTSE queries are considered in Section 4.4 for a binary perceptron teacher, which is in some way the most extreme case as explained below. We conclude in Section 4.5 with a summary and discussion of our results.

4.2 The model

We denote students by $\mathcal{N}$ (for 'Neural network') and teachers by $\mathcal{V}$ (for 'elements of the Version space', see Section 4.4). A student $\mathcal{N}$ is specified by an N-dimensional real weight vector $\mathbf{w}_{\mathcal{N}} \in \mathbb{R}^N$ and calculates its output $y_{\mathcal{N}}$ for an input vector $\mathbf{x} \in \mathbb{R}^N$ according to

$$y_{\mathcal{N}} = f_{\mathcal{N}}(\mathbf{x}) = \frac{1}{\sqrt{N}} \mathbf{x}^{\mathrm{T}} \mathbf{w}_{\mathcal{N}}.$$

Teachers are similarly parameterized in terms of a weight vector $\mathbf{w}_{\mathcal{V}} \in \mathbb{R}^N$, but calculate their output $y_{\mathcal{V}}$ by passing the (scaled) scalar product of $\mathbf{x}$ with this weight vector through a general nonlinear output function. Since we allow the teacher outputs to be corrupted by noise, we only specify the average output for a given input and assume that it can be written in the form

$$\langle y_{\mathcal{V}} \rangle_{P(y_{\mathcal{V}}|\mathbf{x},\mathcal{V})} = \bar{g}\left(\frac{1}{\sqrt{N}} \mathbf{x}^{\mathrm{T}} \mathbf{w}_{\mathcal{V}}\right) \tag{4.1}$$

where $\bar{g}(\cdot)$ is a 'noise-averaged' output function. Implicit in eq. (4.1) is the assumption that the noise process preserves, on average, the perceptron structure of the teacher. Similarly, we assume that the variance of the fluctuations $\Delta y_{\mathcal{V}}$ of the teacher outputs $y_{\mathcal{V}}$ around their average values (4.1) can be written as a function $\Delta^2(\cdot)$ of $\frac{1}{\sqrt{N}} \mathbf{x}^{\mathrm{T}} \mathbf{w}_{\mathcal{V}}$ alone:

$$\left\langle (\Delta y_{\mathcal{V}})^2 \right\rangle_{P(y_{\mathcal{V}}|\mathbf{x},\mathcal{V})} = \Delta^2\left(\frac{1}{\sqrt{N}} \mathbf{x}^{\mathrm{T}} \mathbf{w}_{\mathcal{V}}\right). \tag{4.2}$$

This condition is fulfilled, for example, for additive noise on the outputs with finite, input-independent variance or (for inputs obeying a spherical constraint as considered below) when the components of the teacher weight vector are corrupted by additive Gaussian noise with identical variance for each of the components. Noise on the inputs, which has previously been studied with the aim of improving generalization performance (see, e.g., [HK92, GSW89, WS93]), can be treated similarly: For additive Gaussian noise on the input vector $\mathbf{x}$ (again with identical variance for each component), eq. (4.2) holds as long as the length of the teacher weight vector $\mathbf{w}_{\mathcal{V}}$ is fixed; this condition is enforced with probability one in the thermodynamic limit for the Gaussian

teacher prior considered below.

We assume that the inputs are drawn from a uniform spherical distribution, $P(\mathbf{x}) \propto \delta(\mathbf{x}^2 - N\sigma_x^2)$. Using as our error measure the standard squared output deviation, $\frac{1}{2}(y_{\mathcal{N}} - y_{\mathcal{V}})^2$, we obtain for the generalization error, i.e., the average error that a student $\mathcal{N}$ makes on a random test input when trying to approximate teacher $\mathcal{V}$,

$$\begin{aligned} \epsilon_g(\mathcal{N}, \mathcal{V}) &= \frac{1}{2}\left\langle (y_{\mathcal{N}} - y_{\mathcal{V}})^2 \right\rangle_{P(y_{\mathcal{V}}|\mathbf{x},\mathcal{V})P(\mathbf{x})} \\ &= \frac{1}{2}\left[Q_{\mathcal{N}}\sigma_x^2 + \left\langle \bar{g}^2(h) \right\rangle_h - 2\frac{R}{Q_{\mathcal{V}}} \left\langle h\bar{g}(h) \right\rangle_h \right] + \frac{1}{2}\left\langle \Delta^2(h) \right\rangle_h \end{aligned} \tag{4.3}$$

where

$$R = \frac{1}{N}\mathbf{w}_{\mathcal{N}}^{\mathrm{T}}\mathbf{w}_{\mathcal{V}} \qquad Q_{\mathcal{N}} = \frac{1}{N}\mathbf{w}_{\mathcal{N}}^2 \qquad Q_{\mathcal{V}} = \frac{1}{N}\mathbf{w}_{\mathcal{V}}^2 . \tag{4.4}$$

Here $\langle\cdot\rangle_h$ denotes an average over a Gaussian random variable h with zero mean and variance $Q_{\mathcal{V}}\sigma_x^2$, and we have assumed the thermodynamic limit, $N \to \infty$, of a perceptron with a very large number of input components. We have kept the last term in (4.3), which arises from the noise on the teacher outputs alone and could in principle be discarded, in order to make the comparison of linear and nonlinear teachers more transparent.

As our training algorithm we take stochastic gradient descent on the training error E_t which, for a training set $\Theta^{(p)} = \{(\mathbf{x}^\mu, y^\mu), \mu = 1 \ldots p\}$, is

$$E_t = \frac{1}{2}\sum_\mu (y^\mu - f_{\mathcal{N}}(\mathbf{x}^\mu))^2. \tag{4.5}$$

A weight decay term $\frac{1}{2}\lambda\sigma_x^2\mathbf{w}_{\mathcal{N}}^2$ is added for regularization, i.e., to prevent overfitting of noise in the training data, parameterized in terms of a dimensionless weight decay parameter λ. Stochastic gradient descent on the resulting energy function

$$E = E_t + \frac{1}{2}\lambda\sigma_x^2\mathbf{w}_{\mathcal{N}}^2 \tag{4.6}$$

yields a Gibbs post-training distribution of students

$$P(\mathcal{N}|\Theta^{(p)}) \propto \exp(-E/T) \tag{4.7}$$

where the training temperature T measures the amount of stochasticity in the training algorithm. For the linear perceptron students considered here, this post-training distribution of students is a Gaussian distribution with mean $\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{a}$ and covariance matrix

$T\mathbf{M}_N^{-1}$, where (see eq. (3.27))

$$\mathbf{M}_N = \lambda\sigma_x^2\mathbf{1} + \mathbf{A} \qquad \mathbf{A} = \frac{1}{N}\sum_\mu \mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}} \qquad \mathbf{a} = \frac{1}{\sqrt{N}}\sum_\mu y^\mu\mathbf{x}^\mu \tag{4.8}$$

with $\mathbf{1}$ denoting the $N \times N$ unit matrix. To have a well defined thermodynamic limit, we assume, as usual, that $p = \alpha N$, i.e., that the number of training examples is proportional to the size of the perceptrons. We will concentrate our analysis on the average generalization error, which is obtained by successively averaging eq. (4.3) over the post-training distribution of students, over the distribution of training sets $\Theta^{(p)}$ produced by a given teacher $\mathcal{V}$, and finally over the prior distribution of teachers, which we assume to be Gaussian, $P(\mathcal{V}) \propto \exp(-\frac{1}{2}\mathrm{w}_\mathcal{V}^2/\sigma_\mathcal{V}^2)$. Under this prior, $Q_\mathcal{V} = \sigma_\mathcal{V}^2 + O(1/\sqrt{N})$, so that in the thermodynamic limit $Q_\mathcal{V}$ can be replaced by $\sigma_\mathcal{V}^2$ in (4.3). Hence the only nontrivial averages in the calculation of the average generalization error are the averages of the overlap parameters R and $Q_\mathcal{N}$ defined in (4.4). Note that typical deviations of the generalization error from its average value are $O(1/\sqrt{N})$ and are therefore vanishingly small in the thermodynamic limit. This property is normally referred to as 'self-averaging' and holds quite generally for a wide range of quantities which are 'intensive', i.e., do not scale with the system size N (see, e.g., [BY86]); we will use it frequently throughout this and the following chapters.

The main aim of the present chapter is to calculate for the learning scenario defined above the average generalization error as a function of the normalized number of training examples, $\alpha = p/N$, for learning from MSSE and MTSE queries. By comparing the results to learning from random examples, we will be able to draw conclusions about the efficacy of query learning in imperfectly learnable problems.

4.3 Random examples and minimum student space entropy (MSSE) queries

We now calculate the generalization performance resulting from random examples and from MSSE queries. For learning from random examples, each input in the training set is drawn randomly and independently from the assumed uniform spherical input distribution. By contrast, for MSSE queries each new training input is chosen such that the entropy of the post-training distribution of students is minimized. The properties of MSSE queries for linear students have been discussed in detail in Section 3.3, and we only review the salient features here.

For the stochastic gradient descent learning algorithm described above and the

resulting Gaussian post-training distribution, the student space entropy (normalized by N) is given by

$$S_{\mathcal{N}} = -\frac{1}{2N} \ln |\mathbf{M}_{\mathcal{N}}| \tag{4.9}$$

up to an unimportant constant which depends on the learning temperature T only. The student space entropy is independent of the training outputs y^μ, which is characteristic of linear students (see, e.g., [Mac92c, Sil80]). The entropy (4.9) is minimized by choosing each new training input along an eigendirection of the existing $\mathbf{M}_{\mathcal{N}}$ with minimal eigenvalue. If we apply such minimum entropy queries in sequence, we find that the first N training inputs are pairwise orthogonal but otherwise random (on the sphere $\mathbf{x}^2 = N\sigma_x^2$), followed by another block of N such examples, and so on. The overlap $\frac{1}{N}(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{x}^\nu$ of two different inputs in a training set generated by MSSE queries is thus either 0 (if they belong to the same block) or of the size typical for random inputs, which is $O(1/\sqrt{N})$. These 'pseudo-random' overlaps simplify the calculation of the average generalization error, which is outlined in Appendix 4.6.

We obtain the following result for the average generalization error for learning from random examples and MSSE queries (primes denote derivatives):

$$\epsilon_{\mathrm{g}} = \frac{1}{2}\gamma_{\mathrm{eff}}^2 \sigma_\nu^2 \sigma_x^2 \left[\lambda_{\mathrm{opt}} G(\lambda) + \lambda(\lambda_{\mathrm{opt}} - \lambda) G'(\lambda)\right] + \epsilon_{\mathrm{g,min}}. \tag{4.10}$$

Here we have introduced the constants

$$\gamma_{\mathrm{eff}} = \frac{1}{\sigma_\nu^2 \sigma_x^2} \langle h\bar{g}(h)\rangle_h = \langle \bar{g}'(h)\rangle_h \tag{4.11}$$

$$\sigma_{\mathrm{eff}}^2 = \sigma_{\mathrm{act}}^2 + \left[\left\langle \bar{g}^2(h)\right\rangle_h - \frac{1}{\sigma_\nu^2\sigma_x^2}\langle h\bar{g}(h)\rangle_h^2\right] \qquad \sigma_{\mathrm{act}}^2 = \left\langle \Delta^2(h)\right\rangle_h \tag{4.12}$$

$$\lambda_{\mathrm{opt}} = \frac{\sigma_{\mathrm{eff}}^2}{\gamma_{\mathrm{eff}}^2 \sigma_\nu^2 \sigma_x^2} \tag{4.13}$$

$$\epsilon_{\mathrm{g,min}} = \frac{1}{2}\sigma_{\mathrm{eff}}^2 \tag{4.14}$$

where $\langle \ldots \rangle_h$ denotes an average over a zero mean Gaussian random variable with variance $\sigma_\nu^2\sigma_x^2$. The function G is the average of $\frac{\sigma_x^2}{N}\mathrm{tr}\,\mathbf{M}_{\mathcal{N}}^{-1}$ over the training inputs and is given by

$$G(\lambda) = \frac{1}{2\lambda}\left(1 - \alpha - \lambda + \sqrt{(1-\alpha-\lambda)^2 + 4\lambda}\right) \tag{4.15}$$

for random examples [KH92a], whereas for MSSE queries its value is (see eq. (3.42))

$$G(\lambda) = \frac{\Delta\alpha}{\lambda + \lfloor\alpha\rfloor + 1} + \frac{1 - \Delta\alpha}{\lambda + \lfloor\alpha\rfloor} \tag{4.16}$$

where $\lfloor\alpha\rfloor$ is the greatest integer less than or equal to α and $\Delta\alpha = \alpha - \lfloor\alpha\rfloor$. In eq. (4.10) we have restricted ourselves to the case of zero learning temperature T as nonzero T gives only an additional positive definite contribution $\frac{1}{2}TG(\lambda)$ to the average generalization error. For finite α, ϵ_g is minimized when the weight decay parameter λ is set to its optimal value, $\lambda_{\rm opt}$, which is related to the effective signal-to-noise ratio of the teacher as explained below. As $\alpha \to \infty$, the generalization error tends to the minimum achievable value, $\epsilon_{\rm g,min}$, which is independent of λ as expected for the limit of an infinitely large training set.

We now explain the remaining constants introduced in eqs. (4.11-4.14). First note that, in all of the averages involved, $\sigma_\nu\sigma_x$ sets the scale of the arguments of $\bar{g}(\cdot)$ and $\Delta^2(\cdot)$. This was to be expected since, under the assumed input distribution and teacher space prior, $\frac{1}{\sqrt{N}}\mathbf{x}^{\rm T}\mathbf{w}_\nu$ has zero mean and variance $\sigma_\nu^2\sigma_x^2$. In eq. (4.12), $\sigma_{\rm act}^2$ is the average variance of the fluctuations of the teacher outputs around their average, i.e., the actual noise level. In order to clarify the meaning of $\gamma_{\rm eff}$ and $\sigma_{\rm eff}^2$, consider the special case of a linear teacher with 'gain constant' γ, which is given by $\bar{g}(h) = \gamma h$, and let the teacher outputs be corrupted by zero mean additive noise. It then follows that $\gamma_{\rm eff} = \gamma$ and $\sigma_{\rm eff}^2 = \sigma_{\rm act}^2$, and the minimum generalization error becomes $\epsilon_{\rm g,min} = \frac{1}{2}\sigma_{\rm act}^2$, which is simply the contribution from the noise on the teacher output. The optimal weight decay is $\lambda_{\rm opt} = \sigma_{\rm act}^2/\gamma^2\sigma_\nu^2\sigma_x^2$, which can be shown to be the inverse of the mean-square signal-to-noise ratio of the teacher (see eq. (3.33)). For a general nonlinear teacher and noise model, we can interpret eqs. (4.11, 4.12) as definitions of an appropriate effective gain constant and noise level, from which $\lambda_{\rm opt}$ and $\epsilon_{\rm g,min}$ are calculated just like for a linear teacher with additive output noise. The difference $\sigma_{\rm eff}^2 - \sigma_{\rm act}^2$ is greater than zero for nonlinear $\bar{g}(\cdot)$, and can be interpreted as effective noise arising from the fact that the linear student cannot reproduce the teacher perfectly. Note from eq. (4.12) that this contribution to the effective noise can be very large for noise-averaged teacher output functions $\bar{g}(\cdot)$ containing a large part which is even in h. Since the effective gain $\gamma_{\rm eff}$ only depends on the odd part of $\bar{g}(\cdot)$, it follows from (4.13) that $\lambda_{\rm opt}$ can be arbitrarily large even if there is no actual noise on the teacher outputs.

By way of example, we show in Figure 4.1 plots of $\sigma_{\rm eff}^2$ vs. $\sigma_{\rm act}^2$ for a teacher with a $\tanh(\cdot)$ output function, for additive output noise (Fig. 4.1a), and for additive Gaussian noise with zero mean and identical variance on each of the N components of the teacher weight vector (Fig. 4.1b). In the latter case we have, denoting the noise variance by $\tilde{\sigma}_\nu^2$, $\bar{g}(h) = \left\langle \tanh(h+\tilde{h})\right\rangle_{\tilde{h}}$ and $\Delta^2(h) = \left\langle \tanh^2(h+\tilde{h})\right\rangle_{\tilde{h}} - \bar{g}^2(h)$, where $\tilde{h}$ is Gaussian with mean zero and variance $\tilde{\sigma}_\nu^2\sigma_x^2$. Applying eq. (4.12) we obtain $\sigma_{\rm eff}^2$ and $\sigma_{\rm act}^2$ as functions of $\tilde{\sigma}_\nu$; eliminating $\tilde{\sigma}_\nu$ yields $\sigma_{\rm eff}^2$ as a function of $\sigma_{\rm act}^2$ as shown in Fig. 4.1b. As $\sigma_{\rm act}^2 \to 1$ (which corresponds to $\tilde{\sigma}_\nu \to \infty$), the difference $\sigma_{\rm eff}^2 - \sigma_{\rm act}^2$ decreases towards

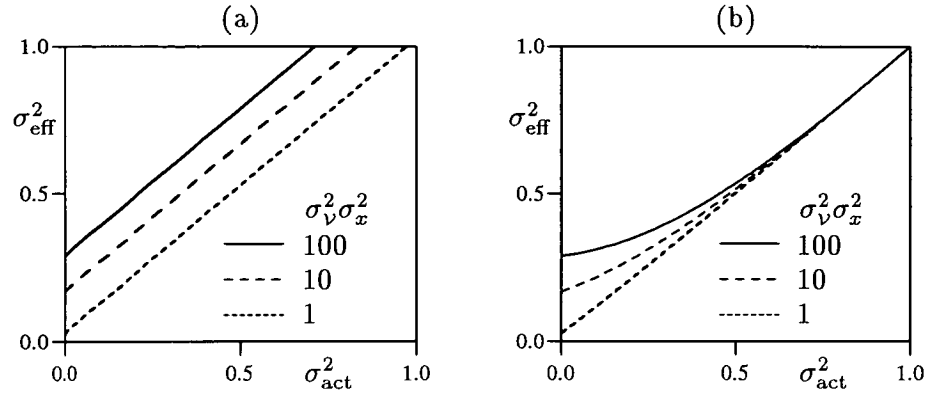

Figure 4.1. Effective noise level vs. actual noise level for a teacher with tanh output function, for additive Gaussian noise on (a) the outputs and (b) the components of the teacher weight vector. The curves are labelled by the values of $\sigma_\nu^2\sigma_x^2$.

zero because, with increasing $\tilde{\sigma}_\nu$, $\bar{g}(\cdot)$ becomes approximately linear over an increasingly large range. Note that due to the nonlinearity of the teacher $\tanh(\cdot)$ output function, σ^2_{eff} remains nonzero in all cases even for $\sigma^2_{\text{act}} = 0$.

We have seen that the average generalization error obtained when learning to approximate a nonlinear teacher with a linear student is exactly the same as for a noisy linear teacher with an effective gain and noise level given by eqs. (4.11, 4.12). Consequently, the efficacy of query learning for a nonlinear teacher is identical to that for a noisy linear teacher. Specifically, if we define the relative improvement in generalization performance due to querying, κ, as[1]

$$\kappa(\alpha) = \frac{\epsilon_g(\text{random examples}) - \epsilon_{g,\min}}{\epsilon_g(\text{queries}) - \epsilon_{g,\min}}$$

then the teacher nonlinearity enters the result only through the value of λ_{opt}. Furthermore, the functional dependence on λ and λ_{opt} is the same as for a noisy linear teacher. Figure 4.2 shows plots of $\kappa(\alpha)$ for some representative values of λ and λ_{opt}. For large α, κ has the asymptotic expansion $\kappa = 1 + 1/\alpha + O(1/\alpha^2)$, which means that for $\alpha \to \infty$, random examples and queries yield the same generalization performance. This can be interpreted in the sense that for large α, learning is essentially hampered by (effective) noise in the data, for which queries are not much more effective than random

[1] Note that this definition, although apparently different, agrees with the one in Chapter 3, eq. (3.45). This is due to the fact that for perfectly learnable problems, $\epsilon_{g,\min}$ is just the contribution to the generalization error arising from the teacher noise alone, which we disregarded in Chapter 3 (see after eq. (3.26)).

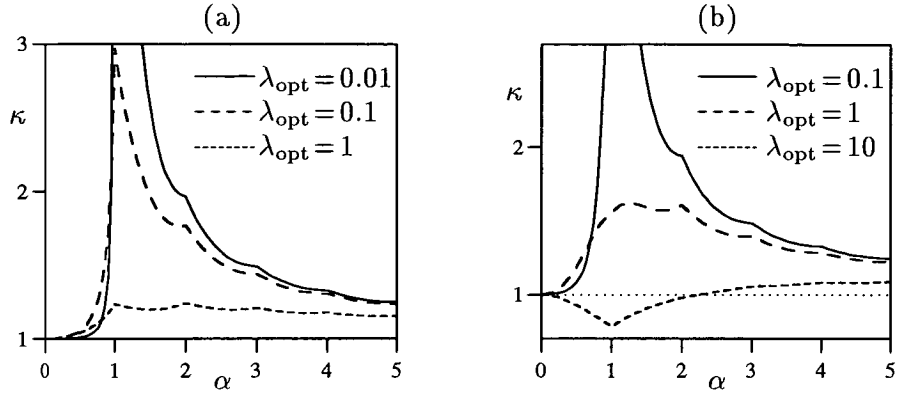

Figure 4.2. Relative improvement κ in generalization error due to MSSE queries, for (a) optimal weight decay, $\lambda = \lambda_{\rm opt}$, and (b) $\lambda = \lambda_{\rm opt}/10$.

examples (cf. the discussion in Section 3.3). For finite α, the behaviour of κ depends on λ and $\lambda_{\rm opt}$. For optimal weight decay $\lambda = \lambda_{\rm opt}$ (Fig. 4.2a), κ has a maximum at $\alpha = 1$ the height of which diverges as $\lambda_{\rm opt}^{-1/2}$ for $\lambda_{\rm opt} \to 0$. For $\lambda > \lambda_{\rm opt}$, the results are qualitatively similar but, for identical values of $\lambda_{\rm opt}$, κ is generally larger than for optimal weight decay $\lambda = \lambda_{\rm opt}$. For $\lambda < \lambda_{\rm opt}$ (Fig. 4.2b), κ tends to be smaller than for optimal weight decay; in fact, for $\lambda_{\rm opt} > 2$, values of $\kappa < 1$ can occur which means that queries do *worse* than random examples. As discussed in Section 3.3, this can be interpreted in the sense that for $\lambda < \lambda_{\rm opt}$, the weight decay 'underestimates' the effective teacher noise level, leading to spurious information gain in student space and thus making the student space entropy an unreliable indicator of generalization performance improvement. This case is particularly relevant for nonlinear teachers where $\lambda_{\rm opt}$ can be very large even if there is no actual noise on the teacher outputs. Nevertheless, even for $\lambda < \lambda_{\rm opt}$ the asymptotic expansion of $\kappa = 1 + 1/\alpha + O(1/\alpha^2)$ given above remains valid, and hence κ necessarily increases above one for large enough α.

The fact that κ tends to unity for $\alpha \to \infty$ implies that the relative improvement in generalization error over random examples due to MSSE querying tends to zero in this limit. We shall explore in the next section whether it is possible to improve generalization performance more significantly by using MTSE queries. Before doing so, however, we briefly mention the analogue of the result (4.10) for the average *training error*, in order to show that the training error is affected by the teacher nonlinearity in qualitatively the same way as the generalization error. To remove the trivial scaling of the training error $E_{\rm t}$ defined in (4.5) with the number of training examples, we consider the quantity $\epsilon_{\rm t} = E_{\rm t}/p$. Performing an average over students, training sets and teachers

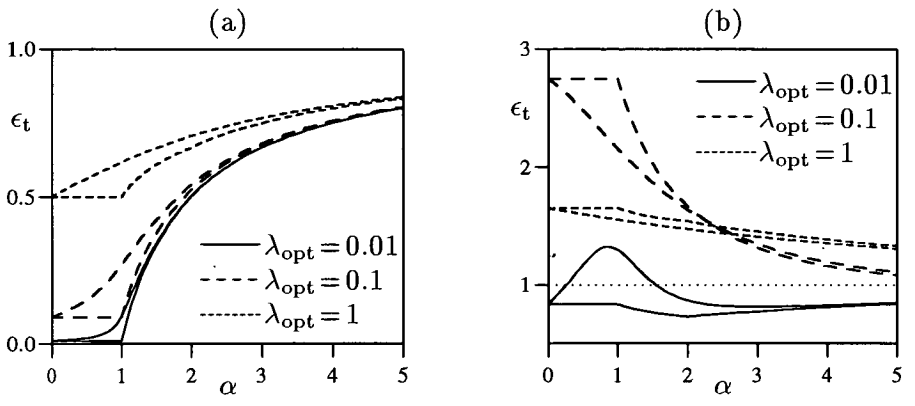

Figure 4.3. Average training error ϵ_t, in units of $\epsilon_{g,min}$, for MSSE queries (curves which are constant for $\alpha \in [0,1]$) and random examples. The weight decay parameter is (a) set to its optimal value, $\lambda = \lambda_{opt}$, and (b) $\lambda = 10\,\lambda_{opt}$.

as for the generalization error, we find

$$\epsilon_t = \epsilon_{g,min}\left[1 - \frac{1}{\alpha} + \frac{\lambda^2}{\alpha\lambda_{opt}}\Big(G(\lambda) + (\lambda - \lambda_{opt})G'(\lambda)\Big)\right]. \tag{4.17}$$

For a linear teacher and random training examples, this result agrees with the one derived in [KH92a, DW93]. As above, we have restricted ourselves to the case of zero training temperature T; nonzero T would give an additional positive contribution $T(1 - \lambda G(\lambda))/2\alpha$ to the average training error. The function $G(\lambda)$ is again given by (4.15) for random training examples and by (4.16) for MSSE queries. In eq. (4.17) the teacher nonlinearity only enters through $\epsilon_{g,min}$ and λ_{opt}, and hence we find again the analogy between nonlinear and noisy linear teachers discussed above. Figure 4.3 shows plots of $\epsilon_t(\alpha)$ for selected values of λ and λ_{opt}. Interestingly, it can be shown that the training error is always smaller for MSSE queries than for random examples for $\lambda \leq \lambda_{opt}$, whereas for $\lambda > \lambda_{opt}$ it can also be greater. In comparison with the analogous relationships for the generalization error discussed above, the roles of the two λ-regimes are thus reversed here. For large α, the ratio of the training error for random examples to that for queries is $1 + \lambda^2/(\lambda_{opt}\alpha^3) + O(1/\alpha^4)$, which is always larger than one for sufficiently large α.

Note that for $\alpha \to \infty$, ϵ_t tends to $\epsilon_{g,min}$, as does the average generalization error ϵ_g. For random training examples, this is necessarily the case as the training error becomes an unbiased estimate of the generalization error for an infinite number of training examples. The fact that the result also holds for MSSE queries shows that

they 'cover' the input space as well as random examples in the limit $\alpha \to \infty$. This is not necessarily the case for queries chosen to optimize an objective function other than the student space entropy. An example of this are the MTSE queries discussed in the next section, for which the generalization error tends to a limiting value for $\alpha \to \infty$ which depends on the weight decay λ, whereas the training error converges to $1/2$ in this limit, independently of λ, as shown in Appendix 4.7. In this case, therefore, the training error does not give an unbiased estimate of the generalization error, even for an infinite number of training examples.

4.4 Minimum teacher space entropy (MTSE) queries

We now consider the generalization performance achieved by MTSE queries. We remind readers that such queries could be employed if the teacher space is known, but a student of a simpler functional form has deliberately been chosen. As an example, consider a classification task, for which the teacher outputs are discrete class labels. In order to be able to use a training algorithm of gradient descent type, one might then choose to consider students with continuous outputs, for which the training error is a differentiable function of the student parameters. The scenario considered below, with a binary perceptron teacher and a linear perceptron student, can in fact be thought of as a simple model for situations of this kind. In general, the aim in using MTSE rather than MSSE queries would be to exploit the structure of the teacher space if this is known (for perfectly learnable problems) to make query learning very efficient compared to random examples. In the binary teacher/linear student scenario, this is indeed the case: as mentioned in the introduction, the efficacy of minimum entropy query learning is high for a perfectly learnable task with binary perceptron student and teacher, whereas it is comparatively low when both student and teacher are linear perceptrons. In the imperfectly learnable case, one would thus hope, by using MTSE queries, to 'transfer' the benefits for query learning of the binary perceptron structure of the teacher space into the student space.

The generalization performance achieved by MTSE and MSSE queries will differ most when the post-training student distribution and the posterior teacher distribution are maximally different. For continuous, invertible teacher output functions $\bar{g}(h)$, the posterior teacher distribution will be approximately Gaussian once the number of training examples is sufficiently large, and thus similar to the post-training student distribution (which, as explained above, is Gaussian for the linear students we are considering). This motivates our choice of considering a non-invertible teacher output function in our analysis of MTSE queries; specifically, we study the extreme case of an

output function which only takes on the two different values ± 1, $\bar{g}(h) = \mathrm{sgn}(h)$, corresponding to a binary perceptron teacher. Since in this case the length of the teacher weight vector has no influence on the teacher's input-output mapping, we set $\sigma_{\mathcal{V}}^2 = 1$ without loss of generality. Similarly, the value of σ_x^2 only scales the student overlap parameters R and $Q_{\mathcal{N}}$ and cancels from the average generalization error, and hence we also set $\sigma_x^2 = 1$.

For simplicity, we assume that the training data generated by the binary perceptron teacher is noise free (corresponding to $\Delta^2(\cdot) \equiv 0$). The posterior probability distribution in teacher space given a certain training set is then proportional to the prior distribution on the *version space* (the set of all teachers that could have produced the training set without error) and zero everywhere else. From this the teacher space entropy (normalized by N) can be derived to be, up to an additive constant,

$$S_{\mathcal{V}} = \frac{1}{N} \ln V$$

where the version space volume V is given by ($\Theta(z) = 1$ for $z > 0$ and 0 otherwise)

$$V = \int d\mathbf{w}_{\mathcal{V}}\, P(\mathbf{w}_{\mathcal{V}}) \prod_{\mu=1}^{p} \Theta\left(y^{\mu} \frac{1}{\sqrt{N}} \mathbf{w}_{\mathcal{V}}^{\mathrm{T}} \mathbf{x}^{\mu} \right).$$

It can easily be verified that this entropy is minimized[2] by choosing queries $\mathbf{x}$ which 'bisect' the existing version space, i.e., for which the hyperplane perpendicular to $\mathbf{x}$ splits the version space into two equal halves [SOS92, FSST93]. Such queries lead to an exponentially shrinking version space, $V(p) = 2^{-p}$, and hence a linear decrease of the entropy, $S_{\mathcal{V}} = -\alpha \ln 2$. We consider instead queries which achieve qualitatively the same effect, but permit a much simpler analysis of the resulting student performance. They are similar to those studied in the context of a perfectly learnable problem in Ref. [WR92], and are defined as follows. The $(p+1)$th query, $\mathbf{x}^{p+1}$, is obtained by first picking a random teacher vector $\tilde{\mathbf{w}}_{\mathcal{V}}$ from the version space defined by the existing p training examples, and then picking the new training input $\mathbf{x}^{p+1}$ from the distribution of random inputs but under the constraint that $\tilde{\mathbf{w}}_{\mathcal{V}}^{\mathrm{T}} \mathbf{x}^{p+1} = 0$.

For the calculation of the student performance, i.e., the average generalization error, achieved by the approximate MTSE queries described above, we use an approximation based on the following observation. As the number of training examples, p, increases,

[2] More precisely, what is minimized is the value of the entropy after a new training example $(\mathbf{x}, y)$ is added, averaged over the distribution of the unknown new training output y given the existing training set and the new training input $\mathbf{x}$. See Chapter 2 for a formal definition, and Section 5.4 for a more general discussion of minimum entropy query learning in binary output systems.

the teacher vectors $\tilde{\mathbf{w}}_\nu$ from the version space will align themselves with the true teacher $\mathbf{w}_\nu^0$; their components along the direction of $\mathbf{w}_\nu^0$ will increase, whereas their components perpendicular to $\mathbf{w}_\nu^0$ will decrease, varying widely across the $N-1$ dimensional hyperplane perpendicular to $\mathbf{w}_\nu^0$. Following Ref. [WR92], we therefore assume that the only significant effect of choosing queries $\mathbf{x}^{p+1}$ with $\tilde{\mathbf{w}}_\nu^{\mathrm{T}}\mathbf{x}^{p+1} = 0$ is on the distribution of the component of $\mathbf{x}^{p+1}$ along $\mathbf{w}_\nu^0$. Writing this component as $x_0^{p+1} = (\mathbf{x}^{p+1})^{\mathrm{T}}\mathbf{w}_\nu^0/|\mathbf{w}_\nu^0|$, its probability distribution can readily be shown to be

$$P(x_0^{p+1}) \propto \exp\left(-\tfrac{1}{2}(x_0^{p+1}/\sigma_x s_p)^2\right) \tag{4.18}$$

where s_p is the sine of the angle between $\tilde{\mathbf{w}}_\nu$ and $\mathbf{w}_\nu^0$. For finite N, the value of s_p is dependent on the p previous training examples that define the existing version space and on the teacher vector $\tilde{\mathbf{w}}_\nu$ sampled randomly from this version space. In the thermodynamic limit, however, the variations of s_p become vanishingly small. We can thus replace s_p by its average value, which is a function of p alone. As $N \to \infty$, this average value becomes a continuous function of $\alpha = p/N$, the number of training examples per weight, which we denote simply by $s(\alpha)$. The calculation can then be split into two parts: First, the function $s(\alpha)$ is obtained from a calculation of the teacher space entropy using the replica method, generalizing the results of Ref. [GT90]. The average generalization error can then be calculated by using an extension of the response function method described in Chapter 7 or by another replica calculation (now in student space) as in Ref. [DW93]. Below, we only give the results of these calculations, deferring details to Appendix 4.7.

The first part of the calculation yields the teacher space entropy S_ν as the saddle point of

$$\frac{1}{2}\left(\frac{q-r^2}{1-q} + \ln(1-q)\right) + 2\int_0^\alpha d\alpha' \int_0^\infty Dy \int_{-\infty}^\infty Dt \, \ln H\left(\frac{t\sqrt{q-r^2} - y\, r\, s(\alpha')}{\sqrt{1-q}}\right) \tag{4.19}$$

with respect to q and r, which are respectively the average scalar product (normalized by N) of two teachers from the version space, and of the true teacher and a teacher from the version space. Here we have used the abbreviations $Dz = \exp(-\frac{1}{2}z^2)\, dz/\sqrt{2\pi}$ and $H(z) = \int_z^\infty Dz'$. The value of $s(\alpha)$ can be expressed in terms of the saddle point value of r, which we denote by $r(\alpha)$, as $s^2(\alpha) = 1 - r^2(\alpha)$. The saddle point equations derived from (4.19) yield $r(\alpha)$ and hence $s(\alpha)$ as a function of the values of $s(\alpha')$ for $0 \le \alpha' < \alpha$. This determines the function $s(\alpha)$ recursively, starting from the initial condition $s(0) = 1$. Evaluating this recursion numerically, we obtain the results plotted in Figure 4.4. For large α values, the teacher space entropy decreases linearly

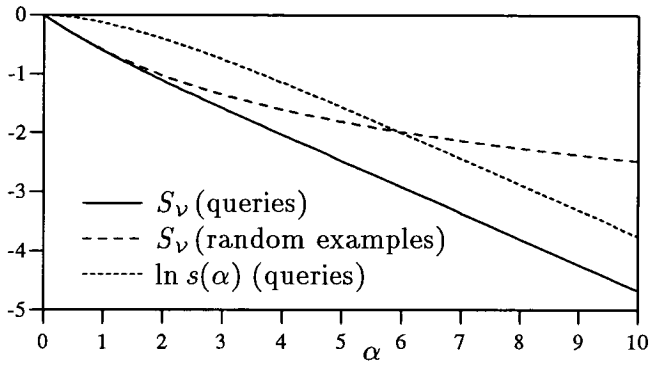

Figure 4.4. MTSE queries: Teacher space entropy, S_V (with value for random examples plotted for comparison), and $\ln s$, the log of the sine of the angle between the true teacher and a random teacher from the version space.

with α, with gradient $c \approx 0.44$, whereas the entropy for random examples, also shown for comparison, decreases much more slowly (asymptotically like $-\ln \alpha$ [GT90]). The linear α-dependence of the entropy for queries corresponds to an average reduction of the version space volume with each new training example by a factor of $\exp(-c) \approx 0.64$, which is reasonably close to the factor $\frac{1}{2}$ for proper bisection of the version space. This shows that our approximate MTSE queries achieve qualitatively the same results as true MTSE queries, and thus justifies our choice of analysing the former rather than the latter.

Before discussing the student performance achieved by (approximate) MTSE queries, we note from figure 4.4 that $\ln s(\alpha)$ decreases linearly with α for large α, with the same gradient as the teacher space entropy. Hence $s(\alpha) \propto \exp(-c\alpha)$ for large α, and MTSE queries force the teacher weight vectors from the version space to approach the true teacher exponentially quickly. It can easily be shown that if we were learning with a binary perceptron student, i.e., if the problem were perfectly learnable, then this would result in an exponentially decaying generalization error, $\epsilon_g \propto \exp(-c\alpha)$. MTSE queries would thus lead to a marked improvement in generalization performance over random examples (for which $\epsilon_g \propto 1/\alpha$ [GT90]). It is this significant benefit (in teacher space) of query learning that provides the motivation for using MTSE queries in imperfectly learnable problems such as the one considered here.

From the numerical values of $s(\alpha)$, the average generalization error achieved by the linear student when learning from our approximate MTSE queries can be calculated as outlined in Appendix 4.7. The results plotted in Figure 4.5 show that MTSE queries do not have the desired effect of translating benefits in teacher space into improvements in generalization performance for the linear student. In fact, they actually lead to

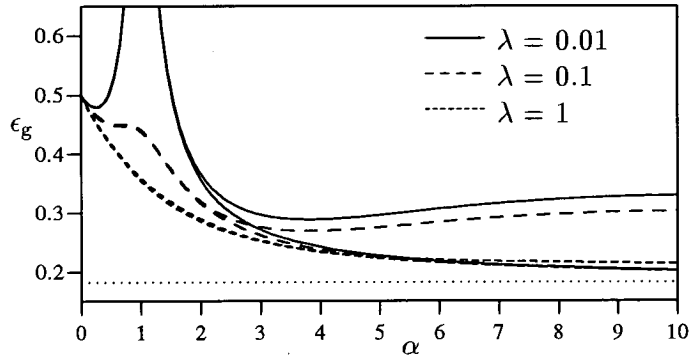

Figure 4.5. Generalization error for MTSE queries (higher curves of each pair) and random examples (lower curves), for weight decay $\lambda = 0.01, 0.1, 1$. The curves for random examples (which are virtually indistinguishable from one another already at $\alpha = 10$) converge to the minimum achievable generalization error $\epsilon_{g,\min}$ (dotted line) as $\alpha \to \infty$.

a deterioration of generalization performance, i.e., a larger generalization error than that obtained for random examples. Worse still, they 'mislead' the student to such an extent that the minimum achievable generalization error is not reached even for an infinite number of training examples, $\alpha \to \infty$. How does this happen? It can be verified from (4.30, 4.31) that the angle between the student and teacher weight vectors tends to zero for $\alpha \to \infty$ as expected, while Q_N, the normalized squared length of the student weight vector, approaches

$$Q_N(\alpha \to \infty) = \frac{2}{\pi}\left(\frac{\overline{s}(\infty)}{\lambda + \overline{s^2}(\infty)}\right)^2 \tag{4.20}$$

where $\overline{s}(\infty) = \int_0^\infty d\alpha\, s(\alpha)$, $\overline{s^2}(\infty) = \int_0^\infty d\alpha\, s^2(\alpha)$ as defined in (4.28). Unless the weight decay parameter λ happens to be equal to $\overline{s}(\infty) - \overline{s^2}(\infty)$, this is different from the optimal asymptotic value, which is $2/\pi$. This is the reason why in general the linear student does not reach the minimum possible generalization error even as $\alpha \to \infty$. The approach of Q_N to its non-optimal asymptotic value can cause an increase in the generalization error for large α and a corresponding minimum of the generalization error at some finite α, as can be seen in the plots for $\lambda = 0.01$ and 0.1 in Figure 4.5. For $\lambda = 0$, eq. (4.20) has the following intuitive interpretation: As α increases, the version space shrinks around the true teacher $\mathbf{w}_\nu^0$, and hence MTSE queries become 'more and more orthogonal' to $\mathbf{w}_\nu^0$. As a consequence, the distribution of training inputs along the direction of $\mathbf{w}_\nu^0$ is narrowed down progressively (compare eq. (4.18)). Trying to find a best fit to the teacher's binary output function over this narrower range of inputs,

the linear student learns a function which is steeper than the best fit over the range of random inputs (which would give minimum generalization error). This corresponds to a suboptimally large length of the student weight vector, in agreement with eq. (4.20): $Q_{\mathcal{N}}(\alpha \to \infty) > 2/\pi$ for $\lambda = 0$ because $\overline{s^2}(\infty) < \overline{s}(\infty)$.

Summarizing the results of this section, we have found that although MTSE queries are very beneficial in teacher space, they are entirely misleading for the linear student, to the extent that the student does not learn to approximate the teacher optimally even for an infinite number of training examples. With the benefit of hindsight, we note that this makes intuitive sense since the teacher space entropy, according to which MTSE queries are selected, contains no feedback about the progress of the student in learning the required generalization task, and thus MTSE queries cannot be guaranteed to have a positive effect.

It is tempting to think that sufficient feedback might be restored by selecting queries orthogonal to the weight vector of a random *student* from the post-training distribution, rather than the weight vector of a random *teacher* from the version space, i.e., the posterior teacher distribution. In this case, $s(\alpha)$, R and $Q_{\mathcal{N}}$ are obtained by solving eqs. (4.30, 4.31) together with the relation $s(\alpha) = [1 - R^2/Q_{\mathcal{N}}]^{1/2}$ in a self-consistent manner. The result is a power law decay $s(\alpha) \propto \alpha^{-3/4}$ for large α, and a diverging length of the student weight vector, $Q_{\mathcal{N}} \propto \alpha^{1/2}$. From (4.3), this leads to a similar divergence of the average generalization error, and the generalization performance achieved by such 'heuristic feedback queries' is thus even worse than for MTSE queries. Again, an intuitive explanation of this result can be found by considering the narrowing down of the input distribution along the direction of the true teacher $\mathbf{w}_{\mathcal{V}}^0$ that is generated by querying: For MTSE queries, this narrowing down is exponentially fast, effectively 'freezing' the length of the student weight vector to a suboptimal value for sufficiently large α, whereas for the heuristic feedback queries considered above the narrowing down is sufficiently slow to allow the length of the student weight vector to adapt steadily and thus to grow arbitrarily large as the width of the input distribution shrinks to zero.

4.5 Summary and discussion

We have found in our study of an imperfectly learnable problem with a linear perceptron student and a general nonlinear perceptron teacher that queries for minimum student and teacher space entropy, respectively, have very different effects on generalization performance. Minimum student space entropy (MSSE) queries essentially have the same effect as for a linear student learning a noisy linear teacher, with the effective

noise level given by the sum of the actual noise level and an additional contribution due to the fact that the student cannot learn the teacher perfectly. Hence the structure of the student space is the dominant influence on the efficacy of query learning. Minimum teacher space entropy queries (MTSE) on the other hand, which we have investigated for the case of a binary perceptron teacher, perform worse than random examples, leading to a higher generalization error even for an infinite number of training examples. This result is intuitively reasonable since the teacher space entropy contains no feedback about the progress of the student in learning the required generalization task. We have also found that such feedback cannot easily be restored by more heuristic methods of query selection similar to MTSE queries.

Our results, then, are a mixture of good and bad news for query learning for minimum entropy (i.e., maximum information gain) in imperfectly learnable problems: The bad news is that MTSE queries, due to a lack of feedback information about student progress, are not enough to translate significant benefits in teacher space into similar improvements of student performance and may in fact yield worse performance than random examples. The good news is that for MSSE queries, we have found evidence that the structure of the student space is the key factor in determining the efficacy of query learning. If this result holds more generally, then statements about the benefits of query learning can be made on the basis of *how one is trying to learn* only, independently of *what one is trying to learn*—a result of obvious practical significance.

4.6 Appendix: Calculation for random examples and MSSE queries

In this appendix, we outline the calculation of the average generalization error for random examples and MSSE queries. For this purpose, as pointed out in Section 4.2, it is sufficient to obtain the averages of the overlap parameters R and $Q_{\mathcal{N}}$. The averages over the Gaussian post-training distribution are straightforward and yield

$$\langle R\rangle_{P(\mathcal{N}|\Theta^{(p)})} = \frac{1}{N}\mathbf{w}_{\mathcal{V}}^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{a} \qquad \langle Q_{\mathcal{N}}\rangle_{P(\mathcal{N}|\Theta^{(p)})} = \frac{1}{N}\mathbf{a}^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-2}\mathbf{a} + T\frac{1}{N}\mathrm{tr}\ \mathbf{M}_{\mathcal{N}}^{-1}. \quad (4.21)$$

Since both for random examples and for MSSE queries, each new training input depends at most on the previous training inputs, we can use Bayes' theorem to decompose the remaining average over training sets and teachers into one over training outputs, teachers and training inputs. Formally, one has (see eq. (2.27))

$$P(\Theta^{(p)}|\mathcal{V})P(\mathcal{V}) = P(\{y^{\mu}\}|\{\mathbf{x}^{\mu}\},\mathcal{V})P(\mathcal{V})P(\{\mathbf{x}^{\mu}\}) \quad (4.22)$$

where

$$P(\{y^\mu\}|\{\mathbf{x}^\mu\},\mathcal{V}) = \prod_{\mu=1}^{p} P(y^\mu|\mathbf{x}^\mu,\mathcal{V})$$

and we will perform the averages on the r.h.s. of (4.22) in the order from left to right. The average over the y^μ-dependent terms in (4.21) yields, from the assumptions (4.1, 4.2),

$$\begin{aligned}\left\langle \mathbf{a}\mathbf{w}_\nu^{\mathrm{T}} \right\rangle_{P(\{y^\mu\}|\{\mathbf{x}^\mu\},\mathcal{V})} &= \frac{1}{\sqrt{N}} \sum_\mu \mathbf{x}^\mu \mathbf{w}_\nu^{\mathrm{T}} \bar{g}(h^\mu) \\ \left\langle \mathbf{a}\mathbf{a}^{\mathrm{T}} \right\rangle_{P(\{y^\mu\}|\{\mathbf{x}^\mu\},\mathcal{V})} &= \frac{1}{N} \sum_{\mu\neq\nu} \mathbf{x}^\mu (\mathbf{x}^\nu)^{\mathrm{T}} \bar{g}(h^\mu)\bar{g}(h^\nu) + \frac{1}{N}\sum_\mu \mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}}(\bar{g}^2(h^\mu)+\Delta^2(h^\mu))\end{aligned}$$

where we have set $h^\mu = \frac{1}{\sqrt{N}}\mathbf{w}_\nu^{\mathrm{T}}\mathbf{x}^\mu$. Performing the average over the prior teacher distribution $P(\mathcal{V}) \propto \exp(-\frac{1}{2}\mathbf{w}_\nu^2/\sigma_\nu^2)$ for *fixed* $\{\mathbf{x}^\mu\}$, the h^μ become Gaussian random variables with zero means and (co-)variances

$$\langle h^\mu h^\nu \rangle_{P(\mathcal{V})} = \frac{\sigma_\nu^2}{N}(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{x}^\nu.$$

For the assumed spherical input distribution, $\frac{1}{N}(\mathbf{x}^\mu)^2 = \sigma_x^2$, and the variance of each of the h^μ is thus identical to $\sigma_\nu^2\sigma_x^2$. The covariance between h^μ and h^ν for $\mu \neq \nu$ is much smaller since, for random examples, $\frac{1}{N}(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{x}^\nu$ is $O(1/\sqrt{N})$. The same holds for MSSE queries, due to the pseudo-random overlaps between training inputs that they produce. The resulting weak correlation of h^μ and h^ν can be used to expand the average of $\bar{g}(h^\mu)\bar{g}(h^\nu)$. To this end, one writes h^ν as $h^\nu = \epsilon h^\mu + (1-\epsilon^2)^{1/2}\tilde{h}$, where $\epsilon = \langle h^\mu h^\nu\rangle \,/\, \langle (h^\mu)^2\rangle = (\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{x}^\nu/(N\sigma_x^2)$ and $\tilde{h}$ is a zero mean Gaussian variable uncorrelated with h^μ which has variance $\left\langle \tilde{h}^2 \right\rangle = \langle (h^\mu)^2\rangle = \langle (h^\nu)^2\rangle = \sigma_\nu^2\sigma_x^2$. Expanding in the small parameter $\epsilon = O(1/\sqrt{N}) \ll 1$, one obtains ($\bar{g}' \equiv d\bar{g}/dh$)

$$\langle \bar{g}(h^\mu)\bar{g}(h^\nu)\rangle_{P(\mathcal{V})} = \langle \bar{g}(h)\rangle_h^2 + \frac{1}{N\sigma_x^2}(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{x}^\nu \,\langle h\bar{g}(h)\rangle_h \,\langle \bar{g}'(h)\rangle_h + O(1/N)$$

where h is a zero mean Gaussian random variable with variance $\sigma_\nu^2\sigma_x^2$. The remaining averages over the teacher prior $P(\mathcal{V})$ are straightforward:

$$\left\langle \bar{g}^2(h^\mu) + \Delta^2(h^\mu) \right\rangle_{P(\mathcal{V})} = \left\langle \bar{g}^2(h) + \Delta^2(h) \right\rangle_h \qquad \langle \mathbf{w}_\nu \bar{g}(h^\mu)\rangle_{P(\mathcal{V})} = \frac{\mathbf{x}^\mu}{\sqrt{N}\sigma_x^2}\,\langle h\bar{g}(h)\rangle_h$$

where the second equality follows from the fact that due to the isotropy of the teacher prior, the contribution from the components of $\mathbf{w}_\nu$ orthogonal to $\mathbf{x}^\mu$ vanishes.

Collecting the results obtained so far we have for the averages of R and $Q_{\mathcal{N}}$ at fixed

$\{\mathbf{x}^\mu\}$:

$$\langle R\rangle\,|_{\{\mathbf{x}^\mu\}} = \langle h\bar{g}(h)\rangle_h \frac{1}{N^2\sigma_x^2}\sum_\mu (\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{x}^\mu \tag{4.23}$$

$$\begin{aligned}\langle Q_{\mathcal{N}}\rangle\,|_{\{\mathbf{x}^\mu\}} &= T\frac{1}{N}\mathrm{tr}\;\mathbf{M}_{\mathcal{N}}^{-1} + \frac{1}{N^2}\left\langle \bar{g}^2(h)+\Delta^2(h)\right\rangle_h \sum_\mu (\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-2}\mathbf{x}^\mu \\ &+ \frac{1}{N^3\sigma_x^2}\langle h\bar{g}(h)\rangle_h \langle \bar{g}'(h)\rangle_h \sum_{\mu\neq\nu}(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-2}\mathbf{x}^\nu(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{x}^\nu \\ &+ \frac{1}{N^2}\langle \bar{g}(h)\rangle_h^2 \sum_{\mu\neq\nu}(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-2}\mathbf{x}^\nu \end{aligned} \tag{4.24}$$

The last term in (4.24) can be shown to vanish upon averaging over the training inputs, due to the fact that both for random examples and for MSSE queries the distribution of each individual training input $\mathbf{x}^\mu$ is invariant under the reflection $\mathbf{x}^\mu \to -\mathbf{x}^\mu$, whatever the values of the other training inputs. The summations over μ and ν in (4.23, 4.24) can be written more succinctly by exploiting the definitions (4.8):

$$\begin{aligned}\frac{1}{N^2}\sum_\mu (\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-k}\mathbf{x}^\mu &= \frac{1}{N}\mathrm{tr}\;\mathbf{M}_{\mathcal{N}}^{-k}\mathbf{A} \qquad (k=1,2) \\ \frac{1}{N^3}\sum_{\mu\neq\nu}(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-2}\mathbf{x}^\nu(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{x}^\nu &= \frac{1}{N^2}\sum_\mu \mathrm{tr}\;\mathbf{M}_{\mathcal{N}}^{-2}\left[\mathbf{A}-\frac{1}{N}\mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}}\right]\mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}} \\ &= \frac{1}{N}\mathrm{tr}\;\mathbf{M}_{\mathcal{N}}^{-2}\mathbf{A}^2 - \frac{\sigma_x^2}{N}\mathrm{tr}\;\mathbf{M}_{\mathcal{N}}^{-2}\mathbf{A}\end{aligned}$$

If we now introduce the function $G(\lambda) = \frac{\sigma_x^2}{N}\left\langle \mathrm{tr}\;\mathbf{M}_{\mathcal{N}}^{-1}\right\rangle_{P(\{\mathbf{x}^\mu\})}$ and the constants γ_{eff}, σ_{eff}^2 defined in (4.11, 4.12), we can use the relations $\mathbf{A} = \mathbf{M}_{\mathcal{N}} - \lambda\sigma_x^2\mathbf{1}$ and $\partial G/\partial\lambda \equiv G' = -\frac{\sigma_x^4}{N}\left\langle \mathrm{tr}\;\mathbf{M}_{\mathcal{N}}^{-2}\right\rangle_{P(\{\mathbf{x}^\mu\})}$ to write the final averages of R and $Q_{\mathcal{N}}$ as

$$\begin{aligned}\langle R\rangle &= \sigma_\nu^2\gamma_{\mathrm{eff}}(1-\lambda G) \\ \langle Q_{\mathcal{N}}\rangle &= \sigma_\nu^2\left[\frac{T}{\sigma_x^2\sigma_\nu^2}G + \frac{\sigma_{\mathrm{eff}}^2}{\sigma_x^2\sigma_\nu^2}(G+\lambda G') + \gamma_{\mathrm{eff}}(1-2\lambda G-\lambda^2 G')\right].\end{aligned}$$

Inserting these results into (4.3), we finally obtain the expression (4.10) for the average generalization error. Parenthetically, we note that for random examples, the result (4.10) can also be obtained from a replica calculation [Dun].

4.7 Appendix: Calculation for MTSE queries

In this appendix, we sketch the calculation of the average generalization error achieved by our linear perceptron student when learning to approximate a noise free binary perceptron teacher from MTSE queries. We use the approximation explained before eq. (4.18) in order to carry out the average over training inputs. Specifically, we assume that the effect of MTSE queries on the distribution of training inputs is non-negligible only for the input components along the direction of the true teacher $\mathbf{w}_\nu^0$, which are distributed according to eq. (4.18). The other input components, i.e., the ones orthogonal to the true teacher, which for the $(p+1)$th query $\mathbf{x}^{p+1}$ are given by $\mathbf{x}_\perp^{p+1} = \mathbf{x}^{p+1} - x_0^{p+1}\mathbf{w}_\nu^0/|\mathbf{w}_\nu^0|$, are therefore distributed as for random examples, obeying the spherical constraint $\mathbf{x}^2 = N$ (remember that we set $\sigma_\nu^2 = \sigma_x^2 = 1$):

$$P\left(\mathbf{x}_\perp^{p+1}\left|x_0^{p+1}\right.\right) \propto \delta\left((\mathbf{x}_\perp^{p+1})^2 + (x_0^{p+1})^2 - N\right)$$

In the thermodynamic limit, this spherical distribution can be replaced by a Gaussian distribution yielding the same average value of $(\mathbf{x}_\perp^{p+1})^2$, and the term $(x_0^{p+1})^2$, which is of order unity, can be neglected compared to N. Combining this with eq. (4.18), the distribution of $\mathbf{x}^{p+1}$ can be written as a Gaussian with reduced covariance along the direction of the true teacher $\mathbf{w}_\nu^0$

$$P(\mathbf{x}^{p+1}) \propto \exp\left\{-\frac{1}{2}(\mathbf{x}^{p+1})^{\mathrm{T}}\left[1 + (s_p^2 - 1)\frac{\mathbf{w}_\nu^0(\mathbf{w}_\nu^0)^{\mathrm{T}}}{(\mathbf{w}_\nu^0)^2}\right]^{-1}\mathbf{x}^{p+1}\right\}. \tag{4.25}$$

As explained in the text, s_p, the sine of the angle between the true teacher and a random teacher from the version space defined by the first p training examples, is self-averaging in the thermodynamic limit and can therefore be regarded as a fixed constant whose value will be calculated later.

In the first part of the calculation, the average[3] of the teacher space entropy over all training sets generated by MTSE queries is determined, and this is then used to obtain the actual values of the s_p as explained after eq. (4.19). One uses the replica trick (see, e.g., [MPV87, Gar88, SST92])

$$\langle \ln V\rangle_{P(\Theta^{(p)})} = \lim_{n\to 0}\frac{1}{n}\ln\langle V^n\rangle_{P(\Theta^{(p)})}$$

[3]The teacher space entropy is, like the generalization error, self-averaging, which means that its value for a typical training set becomes arbitrarily close to its average over all training sets in the thermodynamic limit.

calculating the r.h.s. for positive integer values of n and continuing analytically to $n = 0$. By introducing n replicas of the teacher space, the n-th moment of the version space volume is expressed as

$$V^n = \int \prod_{a=1}^{n} (d\mathbf{w}_\nu^a \, P(\mathbf{w}_\nu^a)) \prod_{\mu=1}^{p} \prod_{a=1}^{n} \Theta\left(y^\mu \frac{1}{\sqrt{N}} (\mathbf{w}_\nu^a)^{\mathrm{T}} \mathbf{x}^\mu\right).$$

Following Ref. [GT90], one can use the fact that for the noise free binary perceptron teacher $y^\mu = \mathrm{sgn}(\frac{1}{\sqrt{N}}(\mathbf{w}_\nu^0)^{\mathrm{T}}\mathbf{x}^\mu)$ to decompose the product of Θ-functions for fixed training example index μ as

$$\prod_{a=1}^{n} \Theta\left(y^\mu \frac{1}{\sqrt{N}} (\mathbf{w}_\nu^a)^{\mathrm{T}} \mathbf{x}^\mu\right) = \prod_{a=0}^{n} \Theta\left(\frac{1}{\sqrt{N}} (\mathbf{w}_\nu^a)^{\mathrm{T}} \mathbf{x}^\mu\right) + \prod_{a=0}^{n} \Theta\left(-\frac{1}{\sqrt{N}} (\mathbf{w}_\nu^a)^{\mathrm{T}} \mathbf{x}^\mu\right).$$

(Note that the products on the r.h.s. include the value $a = 0$, which represents the contribution from the true teacher $\mathbf{w}_\nu^0$.) Introducing Gardner representations for the Θ-functions one can rewrite this as

$$\prod_{a=0}^{n} \left(\int \frac{d\hat{h}^a}{2\pi} \int_0^\infty dh^a\right) \exp\left(i \sum_{a=0}^{n} \hat{h}^a h^a\right) \left[\exp\left(\frac{i}{\sqrt{N}} \sum_{a=0}^{n} \hat{h}^a (\mathbf{w}_\nu^a)^{\mathrm{T}} \mathbf{x}^\mu\right) + \text{c.c.}\right]. \quad (4.26)$$

For a fixed true teacher $\mathbf{w}_\nu^0$, this expression can now easily be averaged over the distribution of $\mathbf{x}^\mu$ as given by (4.25). In principle, an average over the distribution of true teachers, $P(\mathbf{w}_\nu^0) \propto \exp(-\frac{1}{2}\mathbf{w}_\nu^2)$ also has to be carried out. However, this average can be dropped due to the isotropy of the problem both in input space and in weight vector space: The result for fixed $\mathbf{w}_\nu^0$ can only depend on $(\mathbf{w}_\nu^0)^2$, which for the chosen Gaussian teacher space prior equals N up to corrections which can be neglected in the thermodynamic limit. Using this, the average of (4.26) over $\mathbf{x}^\mu$ becomes

$$2\exp\left\{-\frac{1}{2}\left[s_{\mu-1}^2 + 2s_{\mu-1}^2 \sum_{a=1}^{n} r^a + \sum_{a,b=1}^{n} \left(q^{ab} + (s_{\mu-1}^2 - 1) r^a r^b\right)\right]\right\}$$

where we have introduced the order parameters

$$r^a = \frac{1}{N}(\mathbf{w}_\nu^a)^{\mathrm{T}} \mathbf{w}_\nu^0 \qquad q^{ab} = \frac{1}{N}(\mathbf{w}_\nu^a)^{\mathrm{T}} \mathbf{w}_\nu^b.$$

The calculation from this point onwards proceeds exactly as in Ref. [GT90], yielding a saddle point integral over r^a, q^{ab} and the corresponding conjugate order parameters. Assuming a replica symmetric saddle point, $r^a = r$ and $q^{ab} = q + (1 - q)\delta_{ab}$, and replacing the s_p by a continuous function $s(\alpha)$ of $\alpha = p/N$, one obtains the average

teacher space entropy in the form (4.19) given in the text[4].

In the second part of the calculation, the average generalization error achieved by the linear perceptron student when learning from MTSE queries is calculated. The necessary averages of the overlap parameters R and $Q_{\mathcal{N}}$ can again be obtained from a replica calculation. One starts from the free energy corresponding to the Gibbs post-training distribution of students (4.7)

$$f = -\frac{T}{N} \ln Z \qquad Z = \int d\mathbf{w}_{\mathcal{N}} \exp(-E/T) \tag{4.27}$$

which can be regarded as a generating function for the averages of the overlap parameters. The free energy is self-averaging and its value in the thermodynamic limit can hence be obtained by averaging over all training sets, again using the replica method. The calculation follows closely the standard method [SST92], with appropriate modifications taking into account the presence of a weight decay [DW93] and the nonlinearity of the teacher output function [BKO93]. The only major difference from the calculation for learning from random examples is the modified input distribution (4.25). Introducing the averages

$$\overline{s}(\alpha) = \int_0^\alpha d\alpha'\, s(\alpha') \qquad \overline{s^2}(\alpha) = \int_0^\alpha d\alpha'\, s^2(\alpha') \tag{4.28}$$

one obtains the average free energy as the saddle point of

$$\begin{aligned} \frac{1}{2}\Bigg\{ \lambda Q_{\mathcal{N}} \; - \; & T\frac{Q_{\mathcal{N}} - R^2}{Q_{\mathcal{N}} - Q} - T\ln[2\pi(Q_{\mathcal{N}} - Q)] + \alpha T \ln[1 + (Q_{\mathcal{N}} - Q)/T] + \\ & + \; \frac{\alpha(Q - R^2 + 1) - 2(2/\pi)^{1/2}\overline{s}(\alpha)R + \overline{s^2}(\alpha)R^2}{1 + (Q_{\mathcal{N}} - Q)/T} \Bigg\} \end{aligned} \tag{4.29}$$

with respect to R, $Q_{\mathcal{N}}$ and $Q = \frac{1}{N}\langle \mathbf{w}_{\mathcal{N}} \rangle^2_{P(\mathcal{N}|\Theta(P))}$. The saddle point values of R and $Q_{\mathcal{N}}$ are, in the thermodynamic limit, identical to their averages. Solving the saddle point equations and restricting attention to the limit $T \to 0$, one thus finds:

$$\langle R \rangle \;=\; \left(\frac{2}{\pi}\right)^{1/2} \overline{s}(\alpha)\, \frac{F}{1+G} \tag{4.30}$$

[4]Note that within an exact treatment not relying on the approximation explained before eq. (4.18), it can be shown that the exact symmetry $q = r$ must hold at the saddle point (see, e.g., Section 8.6.1). In our approximation, this q-r symmetry is violated. However, the violations are relatively small, in the sense that the relative deviation between q and r (and $1-q$ and $1-r$, which are the more relevant quantities for large α, when both q and r tend to unity) is never larger than 10%.

$$\langle Q_{\mathcal{N}} \rangle = G + \lambda \frac{\partial G}{\partial \lambda} + \frac{2}{\pi} \left(\frac{\overline{s}(\alpha)}{1+G} \right)^2 \left[\frac{2F}{1+G} \frac{\partial G}{\partial \lambda} - \frac{\partial F}{\partial \lambda} \right]. \tag{4.31}$$

Here the functions G and F are given respectively by (4.15) and

$$\frac{1}{F} = \lambda + \frac{\overline{s^2}(\alpha)}{1+G}.$$

The average generalization error achieved by the linear student as shown in figure 4.5 is obtained by inserting the results (4.30, 4.31) into eq. (4.3) (with the substitutions $\bar{g}(h) = \mathrm{sgn}(h)$ and $\Delta^2(h) \equiv 0$ appropriate for a noise free binary perceptron teacher) and using the numerical results for $s(\alpha)$ obtained from the calculation of the teacher space entropy. Note that eqs. (4.30, 4.31) can also be obtained within the response function formalism of Chapter 7. The function F then emerges as a generalization of the standard response function G in the form $F = \frac{1}{N} \langle \mathrm{tr}\, \mathbf{M}_s^{-1} \mathbf{M}_{\mathcal{N}}^{-1} \rangle_{P(\{\mathbf{x}^\mu\})}$. The matrix $\mathbf{M}_s = \lambda_s \mathbf{1} + \frac{1}{N} \sum_\mu (1/s_\mu^2 - 1) \mathbf{x}^\mu (\mathbf{x}^\mu)^{\mathrm{T}}$, with λ_s determined by the condition $\frac{1}{N} \mathrm{tr}\, \mathbf{M}_s^{-1} = 1$, occurs in the correlations of the variables $z^\mu = (\mathbf{x}^\mu)^{\mathrm{T}} \mathbf{w}_{\mathcal{V}}^0 / |\mathbf{w}_{\mathcal{V}}^0|$ in the form $\langle z^\mu z^\nu \rangle_{P(\mathcal{V}|\{\mathbf{x}^\mu\})} = \frac{1}{N} \mathbf{x}^\mu \mathbf{M}_s^{-1} \mathbf{x}^\nu$.

Finally, the replica formalism can also be used to obtain the average training error achieved by MTSE queries. From the definitions (4.6, 4.27), one has

$$\epsilon_{\mathrm{t}} = \frac{1}{p} \langle E_{\mathrm{t}} \rangle = \frac{1}{p} \left\langle E - \frac{1}{2} \lambda \mathbf{w}_{\mathcal{N}}^2 \right\rangle = \frac{1}{\alpha} \left[\frac{\partial (\langle f \rangle / T)}{\partial (1/T)} - \frac{1}{2} \lambda \langle Q_{\mathcal{N}} \rangle \right].$$

By differentiating (4.29) and inserting the saddle point value of Q, given by $Q = Q_{\mathcal{N}} - TG$, one obtains in the limit $T \to 0$

$$\epsilon_{\mathrm{t}} = \frac{1}{2(1+G)} - \frac{\lambda}{2\alpha} (Q_{\mathcal{N}} - R^2) + \frac{\overline{s^2}(\alpha) R^2 - 2\sqrt{2/\pi}\, \overline{s}(\alpha) R}{2\alpha (1+G)}.$$

In the limit $\alpha \to \infty$, only the first term survives and converges to $1/2$ since $G \to 0$; this proves the λ-independence of the asymptotic value of the average training error referred to in Section 4.3.

Chapter 5

Query learning assuming the inference model is correct

Abstract

We investigate the definition and performance of query learning in situations where there is no knowledge about the rule to be learned (the teacher space). The extended Bayesian framework used so far is modified to allow for the approximation of distributions over teacher space by corresponding distributions over student space, leading to the definition of query learning *assuming the inference model is correct.* Several drawbacks of the corresponding query selection procedures are exposed: Using the case of linear perceptron students as an example, we demonstrate that queries selected to optimize a given objective function can actually lead to values of this objective function which are worse than for learning from random examples. For binary output students, we find the problem of self-confirming hypotheses far from the truth, which means that even an infinite number of training examples does not lead to minimal generalization error. A potential solution to these dangers of query learning assuming the inference model is correct is discussed in the next chapter.

5.1 Introduction

We have argued in previous chapters that in most real-world learning scenarios, one is faced with imperfectly learnable problems, where the student cannot reproduce the teacher perfectly. We have also touched on the question of how query learning performs when the problem is imperfectly learnable *and* there is no information available about the type of rule that one is trying to learn, i.e., the teacher space (see Section 4.3).

We now proceed to investigate this issue in more detail. In Section 5.2, we describe how the extended Bayesian framework for query selection has to be modified in the absence of knowledge about the teacher space: Unknown quantities describing the teacher space have to be approximated, with the natural candidates being the corresponding quantities in student space. This leads us to the definition of query learning assuming the inference model is correct and effectively brings us back to a 'traditional' Bayesian framework; the connection to the extended Bayesian framework, however, allows us to see clearly where the approximations that we make enter the formalism. In the following sections, we then explore the efficacy of queries selected assuming the inference model is correct. For the case of linear perceptron students, we show in Section 5.3 that minimum generalization error and minimum entropy queries now become equivalent. Using the results of Chapters 3 and 4, we conclude that query learning assuming the inference model is correct runs the risk of achieving suboptimal values of the objective function (in this case the generalization error) that one wants to optimize by query learning. In Sections 5.4 and 5.5, we then focus on binary output systems. The general features of query learning for minimum entropy in such systems are discussed in Section 5.4, and a generalized bisection criterion is derived. In Section 5.5, we then proceed to analyse the effects of a mild form of inference model misspecification (incorrect noise model) on such bisection queries, for the case of binary perceptron students learning from binary perceptron teachers. Studying both small and large systems analytically and by computer simulations, we find the problem of self-confirming hypotheses far from the truth: When it is assumed that the inference model is correct, query learning may not yield optimal generalization performance even for an infinite number of training examples. In the space of 'hyperparameters' determining how strongly misspecified the inference model is, phase transitions are possible from such self-confirming hypotheses to a regime where query learning is extremely beneficial, yielding an exponential decay of the average generalization error with the number of training examples. We also discuss the phenomenon of learning in 'bursts' of successive refinement and rejection of hypotheses. We conclude in Section 5.6 with a brief summary and discussion of our results.

5.2 Assuming the inference model is correct or: Back to Bayes

In the present chapter, we continue our investigation of the performance of query learning in situations where knowledge about the structure of the teacher space is not

available. Let us briefly recapitulate how the lack of such knowledge affects the derivation of query selection algorithms. In the general framework set out in Chapter 2, we started from the requirement that queries should optimize a certain objective function $\epsilon(\mathcal{N}, \mathcal{V}, \Theta^{(p)})$ which, in its most general form, can depend on the student $\mathcal{N}$, the teacher $\mathcal{V}$, and the training set $\Theta^{(p)}$ consisting of p input-output pairs (x^μ, y^μ). Assuming that complete retraining takes place after each new training example is added, the dependence of the objective function on the particular student obtained after training is eliminated by averaging over the post-training student distribution $P(\mathcal{N}|\Theta^{(p)})$; likewise, the dependence on the unknown teacher $\mathcal{V}$ is removed by averaging over the posterior teacher distribution $P(\mathcal{V}|\Theta^{(p)})$. This yields an objective function which depends on the training set only:

$$\epsilon(\Theta^{(p)}) = \left\langle \left\langle \epsilon(\mathcal{N}, \mathcal{V}, \Theta^{(p)}) \right\rangle_{P(\mathcal{N}|\Theta^{(p)})} \right\rangle_{P(\mathcal{V}|\Theta^{(p)})}. \tag{5.1}$$

Given the training set $\Theta^{(p)}$, the next query x should be selected such as to optimize the value of this objective function for the enlarged training set $\Theta^{(p)} + (x, y)$. Since the new training output y is unknown, however, only the average of the objective function over the distribution of y is available for optimization by query selection:

$$\epsilon(\Theta^{(p)}, x) = \left\langle \epsilon(\Theta^{(p)} + (x, y)) \right\rangle_{P(y|x,\Theta^{(p)})} \tag{5.2}$$

where $P(y|x, \Theta^{(p)})$ is given by

$$P(y|x, \Theta^{(p)}) = \int d\mathcal{V} P(y|x, \mathcal{V}) P(\mathcal{V}|\Theta^{(p)}). \tag{5.3}$$

Looking back at this derivation of the function $\epsilon(\Theta^{(p)}, x)$ according to which queries should be chosen, we see that knowledge of the teacher space enters (through the posterior distribution $P(\mathcal{V}|\Theta^{(p)})$) in the teacher-average over the objective function (eq.(5.1)) and in the distribution of the unknown new training output y (eq.(5.3)). How, then, are we to select queries in the absence of knowledge about the teacher space? In (5.1), the requirement of having to know the posterior $P(\mathcal{V}|\Theta^{(p)})$ can be avoided by considering an objective function which does not depend on the teacher $\mathcal{V}$ such as, for example, the student space entropy. A scenario of this kind has in fact been investigated in Chapter 4 for linear students. There, the fact that the student space entropy does not depend on the training outputs made the average over the unknown new training output in eq. (5.2) trivial, and the objective function for query selection was therefore automatically well defined without knowledge of the teacher space. This

feature of query learning for minimum student space entropy with linear students was already hinted at in one of the earliest papers on optimal experimental design in the statistics literature, Ref. [Lin56].

In general, however, merely using a teacher-independent objective function for query selection will still leave us with the average (5.2), for which from (5.3) we need to know the teacher posterior $P(\mathcal{V}|\Theta^{(p)})$ and the probability $P(y|x,\mathcal{V})$ of a given teacher $\mathcal{V}$ producing output y for input x. In the absence of information about the teacher space, the only solution is to approximate these distributions in some way. An obvious replacement for the teacher posterior is of course the post-training student distribution $P(\mathcal{N}|\Theta^{(p)})$. This suggests that $P(y|x,\mathcal{V})$ should be approximated by a corresponding quantity $\tilde{P}(y|x,\mathcal{N})$, where the tilde indicates that this quantity is not a true, but only an *assumed* probability (since it is the teacher and not the student that produces the training examples (x^μ, y^μ)). To keep things consistent, one is then naturally led to the consideration of post-training distributions which can be written in a posterior-like form:

$$P(\mathcal{N}|\Theta^{(p)}) \propto \tilde{P}(\mathcal{N}) \prod_{\mu=1}^{p} \tilde{P}(y^\mu|x^\mu,\mathcal{N}) \tag{5.4}$$

Note that just as the $\tilde{P}(y^\mu|x^\mu,\mathcal{N})$ are not true probabilities, the 'pseudo-prior' $\tilde{P}(\mathcal{N})$ is not identical to the true marginal probability distribution of students[1], $P(\mathcal{N})$ (which would be obtained by averaging the post-training distribution $P(\mathcal{N}|\Theta^{(p)})$ over all training sets, cf. the discussion in [WL92]). Eq. (5.4) defines not only a post-training distribution of students but also a 'data generation model': Once $\tilde{P}(y|x,\mathcal{N})$ is specified, it can be used to approximate the true distribution (5.3), which specifies how new training outputs are generated, by

$$\tilde{P}(y|x,\Theta^{(p)}) = \int d\mathcal{N}\tilde{P}(y|x,\mathcal{N})P(\mathcal{N}|\Theta^{(p)}). \tag{5.5}$$

We can therefore refer to (5.4) as the definition of an 'inference model' which specifies both how we learn from the data and how the data generation process is modelled. The discussion so far can then be summed up by saying that in the absence of knowledge about the teacher space, one can still define a query selection algorithm by *assuming that the inference model is correct*: One substitutes in eqs. (5.1-5.3) all occurrences of teachers $\mathcal{V}$ by students[2] $\mathcal{N}$, correspondingly replacing the probabilities $P(\mathcal{V}|\Theta^{(p)})$

[1] Although neither $\tilde{P}(y|x,\mathcal{N})$ nor $\tilde{P}(\mathcal{N})$ are true probabilities, we do of course assume that they obey the usual normalization conditions $\int dy\, \tilde{P}(y|x,\mathcal{N}) = 1$ and $\int d\mathcal{N}\, \tilde{P}(\mathcal{N}) = 1$.

[2] Note that if an objective function $\epsilon(\mathcal{N},\mathcal{V},\Theta^{(p)})$ which depends on both teacher $\mathcal{V}$ and student $\mathcal{N}$ is used, then one has to replace the teacher by an *independent* copy of the student in order to get sensible

by $P(\mathcal{N}|\Theta^{(p)})$ and $P(y|x,\mathcal{V})$ by $\tilde{P}(y|x,\mathcal{N})$. This procedure relies on the assumption that these replacements constitute sufficiently good approximations to yield sensible results. It has previously been pointed out that this can be the Achilles' heel of query learning [Mac92c, Mac92b], and the results in this chapter support this statement.

Note that the above definition of query learning assuming the inference model is correct effectively brings us back to a 'traditional' Bayesian framework for query learning (see, e.g., [Ber85, El-91, Mac92c, PK95]): all reference to the teacher space has disappeared, and one is left with only the student space, which can now be viewed as a space of *hypotheses* (see, e.g., [BS94, Wol92]) about the probabilistic relationship between inputs and outputs. This implies a slight conceptual shift from the framework we have used so far, in which students were defined as implementing a deterministic input-output mapping $y = f_{\mathcal{N}}(x)$. The connection is that the assumed probabilistic input-output relation $\tilde{P}(y|x,\mathcal{N})$ is normally simply a noisy version of the deterministic mapping $y = f_{\mathcal{N}}(x)$.

The dual picture of students as either deterministic predictors or stochastic hypotheses also entails a choice of definitions for the generalization error ϵ_g. So far, we have defined ϵ_g as the average error between the deterministic student output and the noisy or noise-free teacher output. With the motivation of real-world neural network learning in mind, where the goal is normally to produce a single network for predicting the outputs corresponding to previously unseen inputs (although the uncertainty of these predictions may have to be quantified using the complete post-training student distribution), we shall retain this definition of the generalization error in the following. One possible alternative would be to compare the teacher output to the 'noisy' student output distributed according to $\tilde{P}(y|x,\mathcal{N})$; this normally yields a higher value of the generalization error than our previous definition. Secondly, and more in the spirit of traditional Bayesian inference, one could consider the generalization error of the Bayes optimal predictor. For each input x, the Bayes optimal prediction for the output is defined as the value $\hat{y}$ that minimizes the average error between $\hat{y}$ and the true output y, assuming that y is distributed according to $\tilde{P}(y|x,\Theta^{(p)})$ as defined by (5.5). However, with the exception of very simple cases (such as the linear perceptron discussed in the next section), the Bayes optimal predictor cannot normally be represented by a single student $\mathcal{N}$ from the assumed student space; for binary perceptron students, for example, the Bayes optimal predictor would be a committee machine with a large number of hidden units [OH91].

---

results. This means that instead of $\epsilon(\mathcal{N},\mathcal{V},\Theta^{(p)})$ one has to consider $\epsilon(\mathcal{N}_1,\mathcal{N}_2,\Theta^{(p)})$ with $\mathcal{N}_1$ and $\mathcal{N}_2$ independently drawn from the post-training distribution $P(\mathcal{N}|\Theta^{(p)})$.

5.3 Linear perceptron revisited

To illustrate the formalism of query learning assuming the inference model is correct with a simple example, we return in this section to the simple scenario of learning with linear perceptron students. A linear perceptron maps inputs $\mathbf{x} \in \mathbb{R}^N$ to real outputs via the linear mapping $y = f_{\mathcal{N}}(\mathbf{x}) = \mathbf{w}_{\mathcal{N}}^{\mathrm{T}}\mathbf{x}/\sqrt{N}$, and is specified by a weight vector $\mathbf{w}_{\mathcal{N}} \in \mathbb{R}^N$. In Chapters 3 and 4, we considered the Gibbs post-training distribution of students,

$$P(\mathcal{N}|\Theta^{(p)}) \propto \exp\left[-\beta\left(\sum_{\mu=1}^{p}\frac{1}{2}(y^\mu - f_{\mathcal{N}}(\mathbf{x}^\mu))^2 + \frac{\tilde{\lambda}}{2}\mathbf{w}_{\mathcal{N}}^2\right)\right] \tag{5.6}$$

which is parameterized by a temperature parameter β and a weight decay parameter $\tilde{\lambda}$ [DW93]. This has the posterior form (5.4) if we define a probabilistic input-output relation

$$\tilde{P}(y|\mathbf{x},\mathcal{N}) = \frac{1}{\sqrt{2\pi/\beta}}\exp\left(-\frac{\beta}{2}(y - f_{\mathcal{N}}(\mathbf{x}))^2\right) \tag{5.7}$$

and the pseudo-prior

$$\tilde{P}(\mathcal{N}) = (2\pi/\beta\tilde{\lambda})^{-N/2}\exp\left(-\frac{\beta\tilde{\lambda}}{2}\mathbf{w}_{\mathcal{N}}^2\right). \tag{5.8}$$

One can therefore now re-interpret $1/\beta$ as the variance of Gaussian noise added to the 'clean' student outputs $f_{\mathcal{N}}(\mathbf{x})$, while $1/\beta\tilde{\lambda}$ determines the width of the pseudo-prior in student space (see, e.g., [BW91, Mac92d]).

We now consider the various objective functions that can be used to select queries. As already pointed out above, queries for minimum student space entropy can be selected without knowledge of the teacher space, and hence are not affected by the assumption that the inference model is correct. Therefore, all the results for linear perceptron students learning from linear and nonlinear perceptron teachers obtained in Chapters 3 and 4 remain valid[3]. More interesting is the case of queries for minimum generalization error. In the scenario considered in Chapter 3 it was shown (eq. (3.26)) that the generalization error between a linear perceptron student and teacher with weight vectors $\mathbf{w}_{\mathcal{N}}$ and $\mathbf{w}_{\mathcal{V}}$, respectively, is

$$\epsilon_{\mathrm{g}}(\mathcal{N},\mathcal{V}) = \frac{\sigma_x^2}{2N}(\mathbf{w}_{\mathcal{N}} - \mathbf{w}_{\mathcal{V}})^2 \tag{5.9}$$

[3]Trivially, the results obtained for minimum student space entropy queries also hold for minimum teacher space entropy queries if we assume that the inference model is correct, since the teacher space entropy becomes identical to the student space entropy if teachers are replaced by students in its definition.

up to an irrelevant additive constant[4]. Here σ_x^2 determines the radius of the hypersphere $\mathbf{x}^2 = N\sigma_x^2$ from which random inputs are assumed to be sampled uniformly. Replacing the teacher weight vector $\mathbf{w}_\nu$ in this expression by an independent copy of the student weight vector and performing the average over the post-training student distribution, one obtains essentially the overall variance of this distribution as given in (3.27),

$$\epsilon_g(\Theta^{(p)}) = \frac{1}{\beta}\frac{\sigma_x^2}{N}\mathrm{tr}\ \mathbf{M}_\mathcal{N}^{-1} \tag{5.10}$$

where the matrix $\beta\mathbf{M}_\mathcal{N}$ determines the curvature of the post-training student probability distribution (5.6) at its maximum and is defined in terms of the correlation matrix of the training inputs as

$$\mathbf{M}_\mathcal{N} = \tilde{\lambda}\mathbf{1} + \mathbf{A} \qquad \mathbf{A} = \frac{1}{N}\sum_\mu \mathbf{x}^\mu(\mathbf{x}^\mu)^\mathrm{T}.$$

As pointed out in Section 3.3.1, under the assumed spherical constraint on the input vectors, $\mathbf{x}^2 = N\sigma_x^2$, minimizing the objective function (5.10) leads to exactly the same query selection algorithm as minimizing (student or teacher space) entropy. Assuming the inference model is correct therefore has the effect of making minimum generalization error and minimum entropy queries identical, independently of the teacher space. In light of the results obtained in Chapters 3 and 4, this means that queries for minimum generalization error can actually yield a higher generalization error than random examples (namely, when the weight decay $\tilde{\lambda}$ is too small to prevent over-fitting of noise in the training data). This constitutes a first example of the dangers of assuming the inference model is correct: queries can yield sub-optimal values of the objective function which they are selected to optimize. In the present context of learning with linear students, however, the consequences as discussed in Sections 3.3.2 and 4.3 remain rather benign: Queries still yield better generalization performance than random examples for a sufficiently large number of training examples or when the teacher noise level is not too high. For students with binary outputs, the effects of assuming the inference model is correct can be much more significant, as the results in the following sections show.

[4]Note that the precise value of this additive constant depends on whether we interpret students as deterministic predictors or as stochastic hypotheses: In the first case, it would be zero since students with equal weight vectors always agree in their deterministic outputs (and since the teacher weight vector is replaced by an independent copy of the student weight vector if we assume that the inference model is correct). In the second case, the constant would be the sum of the variances of the stochastic student outputs times the factor $\frac{1}{2}$ included in the definition of the error measure $\frac{1}{2}(y_\mathcal{N} - y_\nu)^2$, i.e., $2(1/2\beta) = 1/\beta$.

5.4 Minimum entropy queries for binary output students

In the following sections we will (as already in the previous chapter) focus on queries for minimum student space entropy. It will always be implicit in the following that queries are selected assuming that the inference model is correct. Using this assumption, let us first derive a general expression for the expected (student space) entropy decrease due to the addition of a new training example (x, y) to an existing training set $\Theta^{(p)}$, yielding the enlarged training set $\Theta^{(p+1)}$. The new post-training student distribution follows from the assumed posterior form (5.4) as

$$P(\mathcal{N}|\Theta^{(p+1)}) \propto \tilde{P}(y|x,\mathcal{N})P(\mathcal{N}|\Theta^{(p)}) \tag{5.11}$$

with the normalizing factor

$$\int d\mathcal{N}\tilde{P}(y|x,\mathcal{N})P(\mathcal{N}|\Theta^{(p)}) = \tilde{P}(y|x,\Theta^{(p)}) \tag{5.12}$$

from (5.5). Using the definition of the student space entropy,

$$S(\Theta^{(p)}) = -\int d\mathcal{N}P(\mathcal{N}|\Theta^{(p)}) \ln P(\mathcal{N}|\Theta^{(p)})$$

it only takes a few lines of algebra to show that the expected entropy decrease can be written as

$$\begin{aligned} S(\Theta^{(p)}) - \left\langle S(\Theta^{(p+1)})\right\rangle_{\tilde{P}(y|x,\Theta^{(p)})} &= -\int dy\ \tilde{P}(y|x,\Theta^{(p)}) \ln \tilde{P}(y|x,\Theta^{(p)}) \\ &+ \left\langle \int dy\ \tilde{P}(y|x,\mathcal{N}) \ln \tilde{P}(y|x,\mathcal{N}) \right\rangle_{P(\mathcal{N}|\Theta^{(p)})} \end{aligned} \tag{5.13}$$

The right hand side is the difference between the entropy of the (assumed) distribution of the unknown new training output y given the training set $\Theta^{(p)}$ and the entropy of the (assumed) distribution of y given a student $\mathcal{N}$, averaged over the post-training student distribution. The selection of a query x which maximizes the expected entropy decrease is in general determined by a competition between these two terms. Intuitively, the reason for this is that on the one hand, it is desirable to choose an input x with a corresponding output about which we are maximally uncertain given only the training examples $\Theta^{(p)}$ that we have already seen (corresponding to a large value of the first term on the right hand side of (5.13)). On the other hand, such an input will only decrease the entropy of the post-training student distribution significantly if the corresponding output can rule out some students in this distribution, i.e., if some students make rather definite predictions about this output (this corresponds to a small entropy of

the distribution $\tilde{P}(y|x,\mathcal{N})$ and hence to a large value of the second term on the right hand side of (5.13)).

We now specialize (5.13) to the case of binary outputs, $y = \pm 1$. Furthermore, we restrict attention to post-training distributions which can be written as Gibbs distributions

$$P(\mathcal{N}|\Theta^{(p)}) \propto \exp(-\beta E_t(\mathcal{N}, \Theta^{(p)}))\tilde{P}(\mathcal{N}) \tag{5.14}$$

with the training error E_t as the 'Hamiltonian' and a temperature parameter β. An intuitive motivation for this derives from the fact that most learning algorithms are based on the principle of minimization of the training error in some form or other; eq (5.14) constitutes a natural generalization to stochastic training error minimization. For a more elaborate discussion and theoretical justification of the use of Gibbs post-training distributions see, for example, Refs. [LTS90, BV92, SST92]. For binary output systems, the output of a student can either agree or disagree with a given training output, and any error measure is determined by the two values of the error it assigns these two cases. The standard choice, onto which all others can be mapped by a rescaling of the temperature parameter β and a shift of the origin of the error scale, is the 0/1 error measure, yielding the simple 'error count' training error

$$E_t(\mathcal{N}, \Theta^{(p)}) = \sum_{\mu=1}^{p} \Theta(-y^\mu f_{\mathcal{N}}(x^\mu)). \tag{5.15}$$

(Here the Heaviside step function is defined as usual by $\Theta(x) = 1$ for $x \geq 0$ and $\Theta(x) = 0$ otherwise.) The post-training distribution (5.14) therefore has the posterior form (5.4), with

$$\tilde{P}(y|x,\mathcal{N}) = \frac{\exp(-\beta\Theta(-yf_{\mathcal{N}}(x)))}{1+\exp(-\beta)}. \tag{5.16}$$

In the stochastic interpretation of students, this can be interpreted by saying that the 'true' output $f_{\mathcal{N}}(x)$ of student $\mathcal{N}$ is corrupted by reversing its sign with probability

$$p_{\mathcal{N}} = \frac{e^{-\beta}}{1+e^{-\beta}} = \frac{1}{1+e^{\beta}}. \tag{5.17}$$

This sign-flip probability $p_{\mathcal{N}}$ will in the following also be referred to as the student noise level. Inserting (5.16) into the general expression (5.13) for the expected entropy decrease, we obtain

$$S(\Theta^{(p)}) - \left\langle S(\Theta^{(p+1)}) \right\rangle_{\tilde{P}(y|x,\Theta^{(p)})} = h\left(\tilde{P}(y=+1|x,\Theta^{(p)})\right) - h(p_{\mathcal{N}}) \tag{5.18}$$

where

$$h(p) = -p \ln p - (1-p) \ln(1-p)$$

is the entropy of a binomial distribution over two events occurring with probability p and $1-p$, respectively. Since the second term in (5.18) is now independent of the new input x, the entropy decrease (or information gain) achieves its maximum $\ln 2 - h(p_{\mathcal{N}})$ when $\tilde{P}(y|x, \Theta^{(p)}) = \frac{1}{2}$. In the absence of (student) noise, $p_{\mathcal{N}} = 0$, this yields, as expected, a maximum expected information gain of $\ln 2 = 1$ bit from a binary output. Using the definition (5.5) of $\tilde{P}(y|x, \Theta^{(p)})$ and eqs. (5.16, 5.17), one finds that the condition $\tilde{P}(y|x, \Theta^{(p)}) = \frac{1}{2}$ for minimum entropy queries is equivalent to the requirement that

$$\int d\mathcal{N}\ \Theta(f_{\mathcal{N}}(x)) P(\mathcal{N}|\Theta^{(p)}) = \int d\mathcal{N}\ \Theta(-f_{\mathcal{N}}(x)) P(\mathcal{N}|\Theta^{(p)}) = \frac{1}{2}. \tag{5.19}$$

This means that exactly half the students from $P(\mathcal{N}|\Theta^{(p)})$ predict output $+1$ and the other half output -1 for the query x, i.e., that the query x has to *bisect* the post-training distribution. Note that this result only relies on the Gibbs form (5.14) of the post-training distribution, and requires no further assumptions about, for example, the actual form of the binary functions $f_{\mathcal{N}}(x)$ computed by the class of students considered. For the high-low game with a correct inference model (where the post-training student distribution and the posterior teacher distribution are identical), we have already found the bisection criterion (5.19) in Section 3.2.

5.5 Binary perceptron

Having established the bisection criterion for minimum entropy queries (for binary output students with Gibbs post-training distributions), we now explore the performance achieved by such queries in the special case of binary perceptron students. A binary perceptron is specified by an N-dimensional weight vector $\mathbf{w}_{\mathcal{N}}$ and, in the absence of noise, maps inputs $\mathbf{x} \in \mathbb{R}^N$ to outputs

$$y = f_{\mathcal{N}}(\mathbf{x}) = \operatorname{sgn}\left(\frac{1}{\sqrt{N}} \mathbf{w}_{\mathcal{N}}^{\mathrm{T}} \mathbf{x}\right) \tag{5.20}$$

Due to the invariance of the output under a rescaling of either the input or the weight vector, we assume without loss of generality the normalizations (or 'spherical constraints') $\mathbf{w}_{\mathcal{N}}^2 = N$ and $\mathbf{x}^2 = N$. A weight vector is therefore effectively specified by $N-1$ (rather than N) free parameters, leading to the definition of the number of examples per weight parameter as $\alpha = p/(N-1)$.

In order to facilitate the analysis and to control the number of parameters in the problem, we focus on a relatively mild case of inference model misspecification: The teacher space is assumed to be identical to the student space, consisting of all binary perceptrons operating on the input space of dimension N. The pseudo-prior $\tilde{P}(\mathcal{N})$ is also assumed to be identical to the true prior $P(\mathcal{V})$; we take both to be uniform on the hypersphere of weight vectors obeying the spherical constraint $\mathbf{w}_{\mathcal{N}}^2 = N$ (or $\mathbf{w}_{\mathcal{V}}^2 = N$, respectively) and zero otherwise. The only inference model misspecification arises from the noise model: whereas the student *always* assumes sign-flip noise with probability $p_{\mathcal{N}}$, eqs. (5.16, 5.17), the true noise process is either sign-flip noise with a probability $p_{\mathcal{V}} \neq p_{\mathcal{N}}$, or weight noise, where (independent) Gaussian noise of variance σ^2 is added to each of the components of the teacher weight vector, independently for each new training example. We will find below that these two noise processes yield significantly different results. Note that for (true) sign-flip noise, setting the assumed noise level $p_{\mathcal{N}}$ to be equal to the true noise level $p_{\mathcal{V}}$ brings us back to a scenario with a correctly specified inference model; for weight noise, the inference model is more strongly misspecified since the correct model cannot be obtained for any value of $p_{\mathcal{N}}$. The two different noise models will be considered separately below, for the two limits of very small ($N = 2$) and very large ($N \to \infty$, thermodynamic limit) systems. The main quantity of interest in our analysis will be the generalization error averaged over all training sets and teachers. As pointed out above, several definitions of the generalization error are possible; we confine ourselves to the version defined by comparing the noise-free student and teacher outputs. Using the 0/1 error measure (for agreement/disagreement between the binary student and teacher outputs), the generalization error, i.e., the average error on a random test input (assumed to be sampled uniformly from the sphere $\mathbf{x}^2 = N$), is then (for a formal derivation, see Section 8.6.3 or [OKKN90])

$$\epsilon_{\mathrm{g}}(\mathcal{N}, \mathcal{V}) = \frac{1}{\pi} \arccos R \tag{5.21}$$

where

$$R = \frac{1}{N} \mathbf{w}_{\mathcal{N}}^{\mathrm{T}} \mathbf{w}_{\mathcal{V}} \tag{5.22}$$

is the overlap between weight vectors of student and teacher. The result (5.21) has a natural geometrical interpretation, which is shown in figure 5.1. In the thermodynamic limit $N \to \infty$, the overlap R is self-averaging, i.e., equal to its average value with probability one. The average generalization error, which we simply denote by ϵ_{g} to avoid clutter, can therefore be evaluated by inserting the average of R into (5.21). For finite N, the fluctuations of R are non-negligible and one has to average $\epsilon_{\mathrm{g}}(\mathcal{N}, \mathcal{V})$

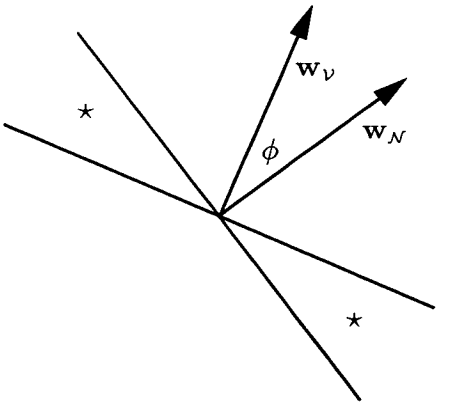

Figure 5.1. Geometrical interpretation of the generalization error between a binary perceptron student and teacher with weight vectors $\mathbf{w}_\mathcal{N}$ and $\mathbf{w}_\nu$, respectively. Shown is the projection of the input space onto the plane spanned by $\mathbf{w}_\mathcal{N}$ and $\mathbf{w}_\nu$; the input regions for which the outputs of student and teacher disagree are marked by asterisks. The generalization error ϵ_g is equal to the probability with which a random input vector will 'land' in one of these regions. For isotropically distributed inputs, this probability is simply ϕ/π where ϕ, the angle between $\mathbf{w}_\mathcal{N}$ and $\mathbf{w}_\nu$, is given by $\phi = \arccos(\mathbf{w}_\mathcal{N}^{\mathrm{T}}\mathbf{w}_\nu/N)$ due to the normalization $\mathbf{w}_\mathcal{N}^2 = \mathbf{w}_\nu^2 = N$.

directly, as we do in the next section.

5.5.1 Small system size limit, $N = 2$

Consider now the case of an extremely small system, $N = 2$. Due to the assumed spherical constraints, weight vectors and inputs can then be represented by points on a circle. For simplicity of presentation, we map this circle onto the interval[5] $[-1, 1]$, where the end points -1 and 1 represent the same weight vector or input vector (corresponding to a 'wrap-around' geometry). This emphasizes the similarity between the $N = 2$ binary perceptron and the high-low game considered in Chapter 3. The binary perceptron is, however, more suitable for our present purposes due to the absence of 'edge effects' in weight space which would occur for high-low (for weight noise, for example, the perturbed weight could end up outside the unit interval on which the clean weights are defined, and would have to be redefined in a suitable way).

[5] A mapping onto the interval $[-\pi, \pi]$, i.e., a representation in terms of angles, might be considered more natural but would make the notation more cumbersome due to the introduction of factors of π in most intermediate results.

If we denote the representatives of the 2-dimensional vectors $\mathbf{w}_{\mathcal{N}}$, $\mathbf{w}_{\mathcal{V}}$, $\mathbf{x}$, on the interval $[-1,1]$ by $w_{\mathcal{N}}$, $w_{\mathcal{V}}$, x, the input output mapping (5.20) implemented by a student (and, analogously, by a teacher) becomes

$$y = f_{\mathcal{N}}(x) = \begin{cases} +1 & \text{for } |x \ominus w_{\mathcal{N}}| < \frac{1}{2} \\ -1 & \text{otherwise} \end{cases}$$

where $\ominus$ denotes subtraction modulo 2, with the result always in the interval $[-1,1]$. Correspondingly, the generalization error (5.21) can be written as

$$\epsilon_g(\mathcal{N},\mathcal{V}) = |w_{\mathcal{V}} \ominus w_{\mathcal{N}}|. \tag{5.23}$$

Sign-flip noise

Consider now the generalization performance achieved by minimum entropy queries when the true noise process is sign-flip noise with probability $p_{\mathcal{V}}$. A simple case which can be solved analytically is the one in which the student assumes that there is no noise, corresponding to $p_{\mathcal{N}} = 0$. As shown in Appendix 5.7, the average generalization error as a function of the number of examples $\alpha = p/(N-1) = p$ is then

$$\begin{aligned} \epsilon_g(\alpha) &= \frac{1}{2} - (1-2p_{\mathcal{V}})\left[1 - \frac{1}{(1+p_{\mathcal{V}})(2-p_{\mathcal{V}})}\right] \\ &+ (1-2p_{\mathcal{V}})\left[\frac{1}{2^\alpha} - \frac{1}{3(1+p_{\mathcal{V}})}\left(\frac{1-p_{\mathcal{V}}}{2}\right)^\alpha - \frac{1}{3(2-p_{\mathcal{V}})}\left(\frac{p_{\mathcal{V}}}{2}\right)^\alpha\right]. \end{aligned} \tag{5.24}$$

At $\alpha = 0$, the average generalization error equals $1/2$ as expected, since the student can only guess randomly as long as no training examples have yet been presented. When the student's noise level estimate is correct, i.e., when the teacher is noise free, $p_{\mathcal{V}} = p_{\mathcal{N}} = 0$, one obtains an exponentially decaying generalization error $\epsilon_g(\alpha) \propto 2^{-\alpha}$ in close analogy with the results for the noise free high-low game obtained in Chapter 3 (see eq. (3.13)). For $p_{\mathcal{V}} > 0$, the generalization error still exhibits an exponential decay with α, but to a nonzero asymptotic value

$$\epsilon_g(\alpha \to \infty) = \frac{1}{2} - (1-2p_{\mathcal{V}})\left[1 - \frac{1}{(1+p_{\mathcal{V}})(2-p_{\mathcal{V}})}\right]$$

(which increases smoothly from 0 to $1/2$ as $p_{\mathcal{V}}$ increases from 0 to $1/2$, the latter limit corresponding to completely random training outputs). This conclusion is borne out by the results of simulations (described in more detail below) shown in figure 5.2. The nonzero asymptotic value of the generalization error is not due an inability of the students to reproduce the correct teacher—remember that in the scenario considered,

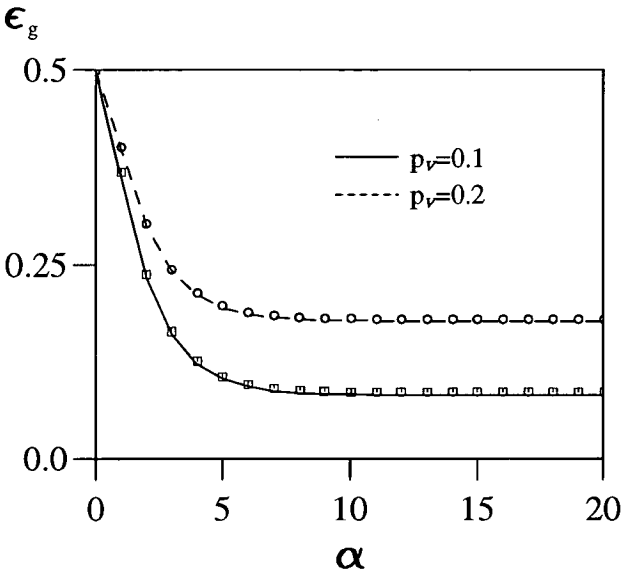

Figure 5.2. Average generalization error ϵ_g vs. number of training examples α in the binary perceptron of size $N = 2$, for query learning based on an assumed noise level $p_N = 0$, i.e., assuming noise free training outputs. The true sign-flip noise level is $p_V = 0.1, 0.2$. The lines give the analytical result (5.24); symbols show simulation results, with statistical errors smaller than the symbol size. Note the nonzero asymptotic value of the generalization error, corresponding to the occurrence of self-confirming hypotheses far from the truth.

student and teacher space are identical—but rather due to the fact that query learning assuming the inference model is correct can produce what one could call *self-confirming hypotheses far from the truth*: The queries 'home in' on a student which can be far from the teacher; and they do it in such a way that even an infinite number of training examples will not correct this wrong hypothesis. An example of how this happens is shown in figure 5.3: For $p_N = 0$, the post-training student distribution is simply constant over the version space (the set of all students which predict all training outputs correctly), and zero otherwise. The bisection criterion for minimum entropy queries then tell us that this version space is simply halved with each new training example. Once a single wrong training output is received, only that half of the version space which does not contain the teacher is retained, and any further training examples will only be able to refine this 'wrong' version space, but never lead to an escape from it. Note that at all times, the training set produced by bisection queries and the corresponding teacher outputs is entirely consistent with the assumption of a noise free teacher (see figure 5.3) and in this sense not only the students from the 'wrong' version space but also the assumption $p_N = 0$ constitute self-confirming hypotheses far from the truth.

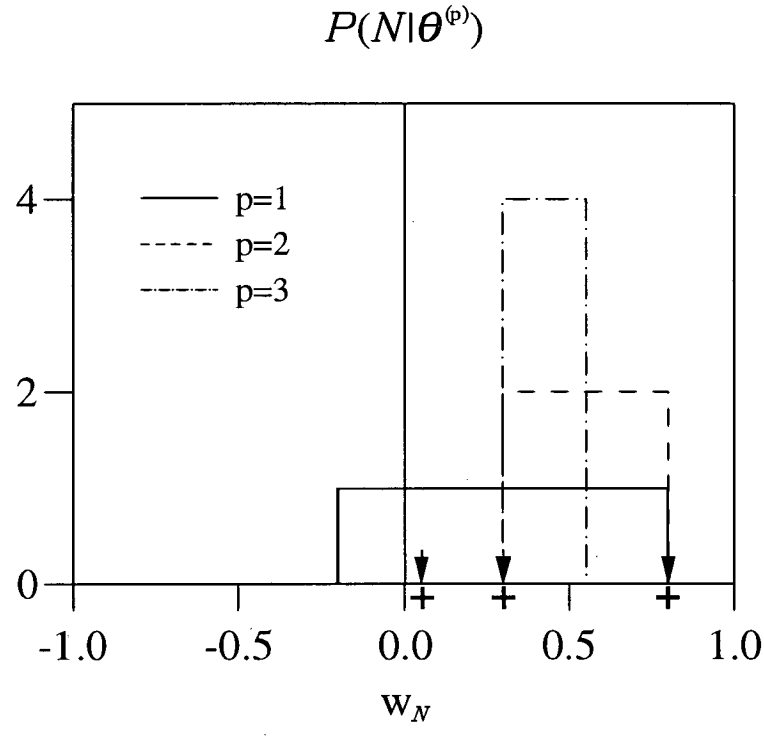

Figure 5.3. How self-confirming hypotheses occur for query learning with assumed noise level $p_N = 0$. Shown is the post-training student distribution $P(\mathcal{N}|\Theta^{(p)})$ ($p = 1, 2, 3$) for an exemplary sequence of training examples. Training inputs are marked by arrows along the x-axis, with the corresponding training outputs shown underneath. The first input $x^1 = 0.3$ is selected randomly; the training output $y^1 = +1$ provided by the teacher (with weight 'vector' $w_\nu = 0$) is uncorrupted. The version space (i.e., the region where $P(\mathcal{N}|\Theta^{(1)})$ is nonzero) therefore includes the teacher. The next output y^2 corresponding to the bisection query $x^2 = 0.8$ is corrupted (due to teacher noise), so that the 'wrong' half of the version space is discarded and cannot be recovered even if the following training outputs are all uncorrupted (as an example, $y^3 = +1$ for the query $x^3 = 0.05$ is shown). Note that at all times, the students from $P(\mathcal{N}|\Theta^{(p)})$ predict all training outputs correctly; it is therefore impossible to conclude from the training data that the teacher is noisy.

At this point the reader may well expect that the behaviour discussed above relies on a pathology of the case $p_{\mathcal{N}} = 0$. If we are not sure what the true noise model is, then we would almost certainly not start off with the guess that there is absolutely no noise. Surely any nonzero value of $p_{\mathcal{N}}$ would get around the problem of these self-confirming hypotheses far from the truth? We have not been able to obtain analytical results for the case $p_{\mathcal{N}} > 0$, and have therefore carried out computer simulations to verify whether this expectation is correct. The student distribution $P(\mathcal{N}|\Theta^{(p)})$ is piecewise constant on the interval $w_{\mathcal{N}} \in [-1, 1]$ as follows from eqs. (5.14, 5.15), and can therefore easily be stored and manipulated on a computer. The determination of minimum entropy queries, which must obey the bisection criterion (5.19), is then straightforward; the corresponding training outputs were produced by reversing the sign of the 'clean' teacher output with probability $p_{\mathcal{V}}$. Averaging over 10,000 training sets generated in this way[6] (for each value of α), we obtained the average generalization error $\epsilon_g(\alpha)$ as shown in figure 5.4 for teacher noise level $p_{\mathcal{V}} = 0.2$ and a range of values of $p_{\mathcal{N}}$. It can clearly be seen that the asymptotic generalization error remains nonzero for a range of nonzero $p_{\mathcal{N}}$ (in the case $p_{\mathcal{V}} = 0.2$ shown in the figure, this range extends at least up to $p_{\mathcal{N}} = 0.01$), implying that the problem of self-confirming hypotheses far from the truth is not confined to the 'pathological' case $p_{\mathcal{N}} = 0$. For larger $p_{\mathcal{N}}$, the average generalization error decays exponentially with α, with the fastest decay rate when $p_{\mathcal{N}} = p_{\mathcal{V}}$. This implies that there exist two substantially different regimes or 'phases' for the α dependence of ϵ_g, with a phase transition[7] taking place at some $p_{\mathcal{N}}$ in the range $0 < p_{\mathcal{N}} < p_{\mathcal{V}}$. The difference between the two regimes is emphasized by comparing the results for query learning with the generalization error for random examples, which has the asymptotic behaviour $\epsilon_g \propto 1/\alpha$ as shown in figure 5.4. If we define the improvement factor κ due to querying as in Chapter 3,

$$\kappa(\alpha) = \frac{\epsilon_g(\text{random examples})}{\epsilon_g(\text{minimum entropy queries})}$$

then we obtain for small $p_{\mathcal{N}}$ – in the self-confirming hypotheses regime – a value of κ which tends to zero as $\alpha \to \infty$, due to the fact that random examples reach zero (average) generalization error whereas queries do not. For high enough $p_{\mathcal{N}}$, on the other

[6] Once the average over training sets has been taken, the result does not depend on the direction of the teacher weight vector any more. It is therefore unnecessary to perform an explicit average over the teacher space prior.

[7] As the system we are considering is extremely small, this phase transition will naturally not be sharply defined, but rather 'smeared out', with a cross-over regime in between the two phases. This can be seen in figure 5.4 for $p_{\mathcal{N}} = 0.1$, where $\epsilon_g(\alpha)$ seems to decay to zero for $\alpha \to \infty$ but does so more slowly than exponentially with α.

hand, κ tends to infinity for $\alpha \to \infty$ since the exponential decay of ϵ_g with α produced by query learning is much faster than the algebraic $1/\alpha$ decay for random examples (see figure 5.4). These two regimes can clearly be distinguished in figure 5.5.

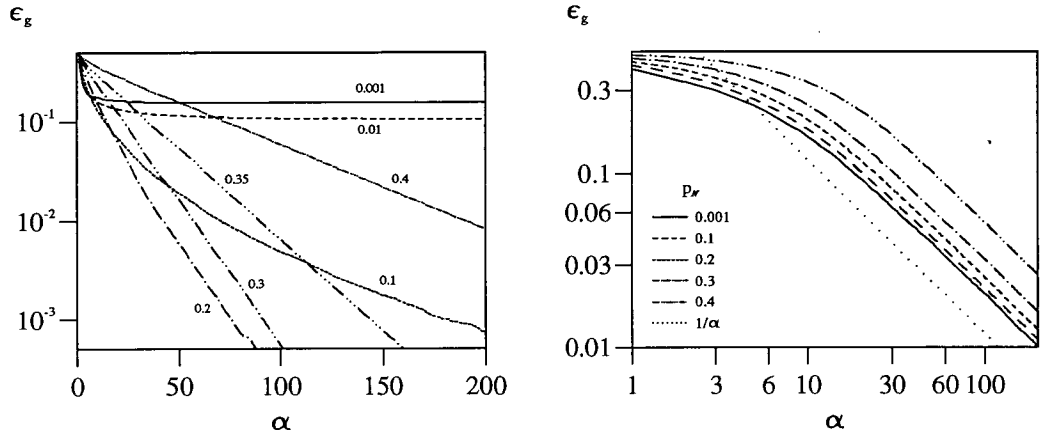

Figure 5.4. Left: Average generalization error ϵ_g achieved by query learning with various values of $p_{\mathcal{N}}$ (shown next to curves), for teacher sign-flip noise level $p_{\mathcal{V}} = 0.2$, vs. number of training examples α. A nonzero asymptotic value of ϵ_g persists for a range of nonzero $p_{\mathcal{N}}$, while for $p_{\mathcal{N}}$ of the order of $p_{\mathcal{V}}$ or larger, ϵ_g decays to zero exponentially with α. Right: Results for random examples; asymptotically, $\epsilon_g \propto 1/\alpha$ independently of $p_{\mathcal{N}}$. Statistical errors from the simulations are appreciable only for very small values of ϵ_g (one standard deviation less than 4% for $\epsilon_g > 0.01$; up to 17% for $\epsilon_g \approx 5 \cdot 10^{-4}$).

To get an intuitive understanding of how self-confirming hypotheses far from the truth can persist for nonzero $p_{\mathcal{N}}$, we show in figure 5.6 an example of how the post-training student distribution $P(\mathcal{N}|\Theta^{(p)})$ changes as more and more training examples (with inputs selected by minimum entropy queries) are received. One observes that every corrupted training output creates a kind of barrier, skewing the post-training student distribution away from the teacher. The next queries are then selected to bisect this incorrectly skewed distribution on the 'wrong' side of the barrier. If the corresponding training outputs are uncorrupted, the student distribution will be bisected increasingly close to the barrier by successive queries, until finally the probability mass on the 'right' side of the barrier becomes large enough for bisection to continue there. At this stage, the barrier has lost its function as a 'trap' for the student distribution. The 'time' (i.e., number of uncorrupted training outputs) it takes to cross the barrier erected by a corrupted training output increases with decreasing $p_{\mathcal{N}}$, as one can see by

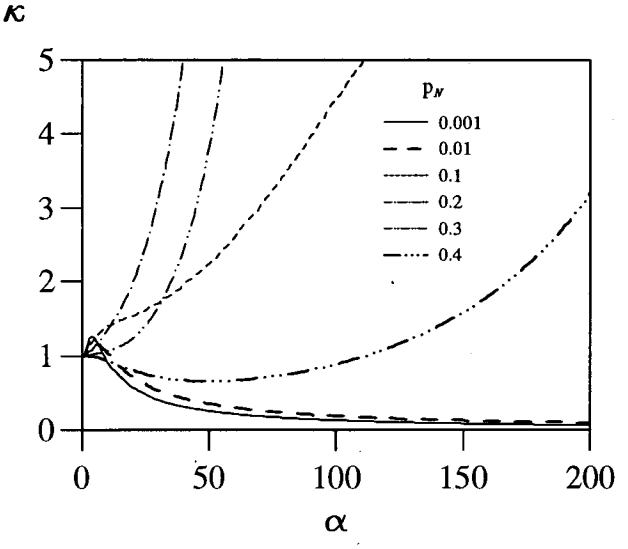

Figure 5.5. Improvement κ in generalization error due to query learning, for teacher sign-flip noise with probability $p_\mathcal{V} = 0.2$. See text for discussion.

considering the following two extreme cases: For $p_\mathcal{N}$ close to $\frac{1}{2}$, each new training example can only skew the post-training student distribution very slightly, and the barriers built up by corrupted training outputs are therefore low and can be crossed quickly. For $p_\mathcal{N} = 0$, on the other hand, barriers can never be crossed as discussed above. For intermediate values of $p_\mathcal{N}$, the 'barrier crossing time' must therefore increase with decreasing $p_\mathcal{N}$ and diverge as $p_\mathcal{N} \to 0$. For finite *teacher* noise level $p_\mathcal{V}$, new barriers will appear due to corruption of training outputs before existing barriers have been crossed, provided the barrier crossing time is sufficiently long, i.e., $p_\mathcal{N}$ is sufficiently small. If the number of new barriers thus created before old ones are crossed is large enough, it appears reasonable that the student distribution will get permanently trapped, making the asymptotic generalization error nonzero. This corresponds to the appearance of self-confirming hypotheses far from the truth observed above for $p_\mathcal{N}$ significantly smaller than $p_\mathcal{V}$.

Parenthetically, we note that for a learning scenario similar to the one considered here, it had previously been argued that the fastest decay of the generalization error achievable by query learning is $\epsilon_g \propto 1/\alpha$ when learning from a noisy teacher [KS95]. The apparent contradiction with the exponential decay we found for $p_\mathcal{N} \approx p_\mathcal{V}$ is resolved by noting that this statement was based on the Cramér-Rao 'information inequality' [Cra46] which only applies when the output probability distribution $P(y=+1|x,\mathcal{V})$

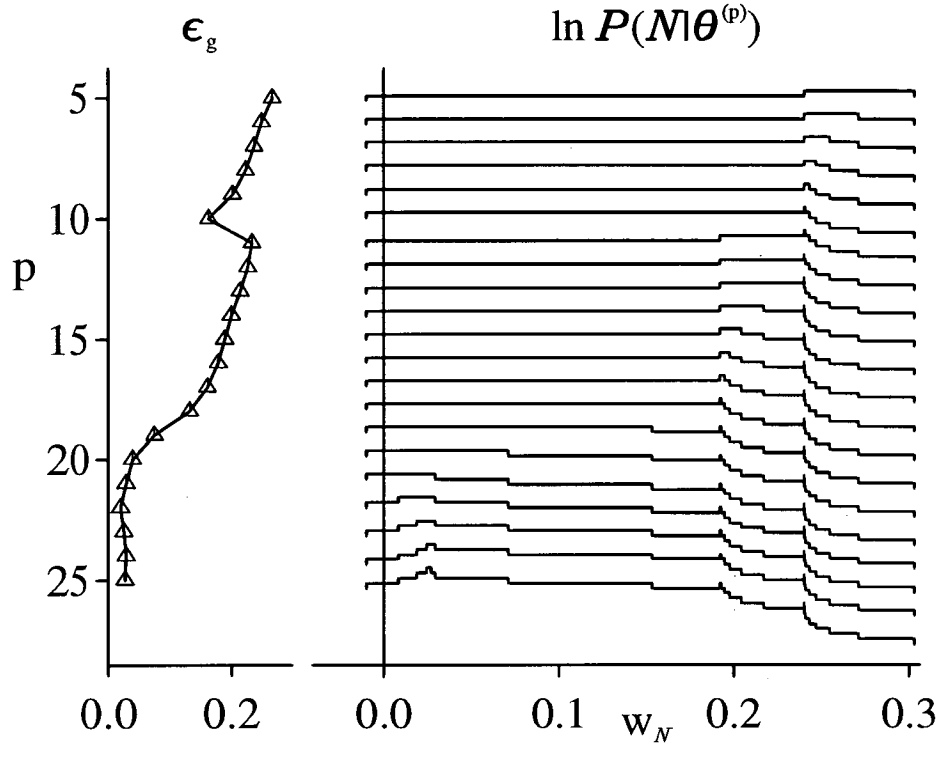

Figure 5.6. Sample evolution of the student distribution $P(\mathcal{N}|\Theta^{(p)})$, for query learning with assumed sign-flip noise level $p_{\mathcal{N}} = 0.005$ from a noisy teacher $w_{\mathcal{V}} = 0$. Right: $\ln P(\mathcal{N}|\Theta^{(p)})$ (shifted by additive constants for better visualization) on a subsection of the weight space $w_{\mathcal{N}} \in [-1, 1]$. Left: Corresponding generalization error (averaged over $P(\mathcal{N}|\Theta^{(p)})$). At $p = 5$, a corrupted training output has skewed the student distribution away from the teacher. The following queries (for which uncorrupted outputs are received) zoom in on the 'barrier' formed by the corrupted output until at $p = 11$, the barrier is crossed because enough probability mass has been shifted from the 'wrong' (right hand) side of the barrier to the 'right' (left hand) side. Training output y^{11} is corrupted and erects another barrier, which is crossed at $p = 19$. While the student distribution is 'trapped' on the wrong side of a barrier, the generalization error is dominated by the error of students near the barrier; when a barrier is crossed, larger changes in ϵ_g occur since the student distribution can respond more strongly to the new training examples.

depends smoothly on the teacher weights. This condition is not fulfilled in the case of sign-flip noise, where there is a discontinuity in $P(y=+1|x,\mathcal{V})$ at the decision boundary $x \ominus w_\nu = \pm\frac{1}{2}$ ($\mathbf{x}^{\mathrm{T}}\mathbf{w}_\nu = 0$ in vector notation).

Summarizing the results for teacher sign-flip noise, we have found that minimum entropy queries selected assuming the inference model is correct can perform both much worse and much better than random examples. They achieve worse generalization performance if the assumed noise level $p_{\mathcal{N}}$ is significantly smaller than the true noise level p_ν, due to the occurrence of self-confirming hypotheses far from the truth. If, on the other hand, $p_{\mathcal{N}}$ is approximately equal to p_ν (so that the inference model is 'approximately correct'), the performance of query learning is vastly superior to that of random examples. The fact that the inference model can be approximately correct or grossly incorrect, depending on the value of $p_{\mathcal{N}}$, is crucial for the occurrence of these two different regimes or phases. In the next section, we consider weight noise, for which the inference model is always incorrect, whatever the value of $p_{\mathcal{N}}$, and we shall confirm the expectation that in this case there is only one 'phase' in $p_{\mathcal{N}}$-space as far as the efficacy of query learning is concerned.

Weight noise

Consider now the case where the actual teacher noise is weight noise, while the student's inference model is based – as before – on the assumption that there is only sign-reversal noise (with probability $p_{\mathcal{N}}$). The training outputs are generated as the output of a perceptron with a perturbed weight vector $\mathbf{w}'_\nu = \mathbf{w}_\nu + \Delta\mathbf{w}_\nu$, where the components of the random vector $\Delta\mathbf{w}_\nu$ (sampled independently for each new training output) are Gaussian with mean zero and variance σ^2. Following Ref. [GT90], we parameterize the noise variance σ^2 in terms of the typical ratio of the lengths of the true and the perturbed weight vector,

$$\gamma = \sqrt{\frac{\langle \mathbf{w}_\nu^2 \rangle}{\langle (\mathbf{w}_\nu + \Delta\mathbf{w}_\nu)^2 \rangle}} = (1+\sigma^2)^{-1/2}. \tag{5.25}$$

The limits $\gamma = 1$ and $\gamma \to 0$ correspond to noise free and completely random training outputs, respectively. The effect of weight noise is illustrated in figure 5.7 in terms of the probability $P(y=+1|x,\mathcal{V})$ vs. x for a given teacher $\mathcal{V}$. Comparison with the case of sign-flip noise, which is also shown, demonstrates that the outputs y corresponding to inputs near the 'decision boundary' of the teacher (where the clean output changes from -1 to $+1$ and vice versa) are much noisier for weight noise than for sign-flip noise. In fact, weight noise can be thought of as producing sign-flip noise with an input dependent

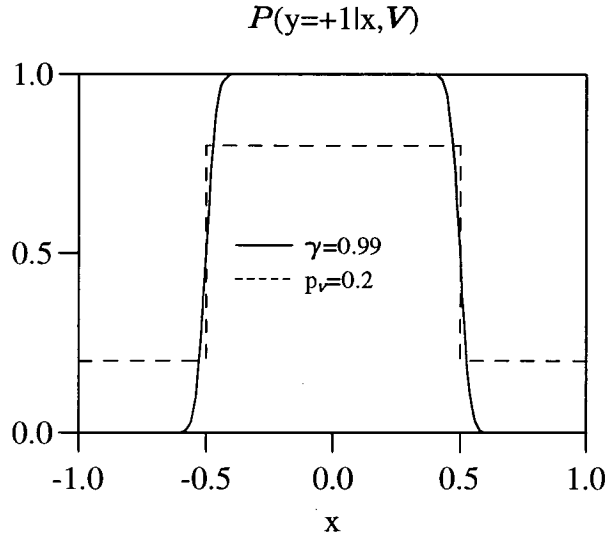

Figure 5.7. The effect of weight noise on teacher outputs. Shown is the probability of obtaining output $+1$, $P(y = +1|x, \mathcal{V})$ as a function of the input x, for the teacher $w_{\mathcal{V}} = 0$ and weight noise $\gamma = 0.99$. Approaching the decision boundary of the teacher ($x = \pm\frac{1}{2}$), the outputs become more and more random. This is not the case for sign-flip noise (the case $p_{\mathcal{V}} = 0.2$ is shown as an example), where all outputs are 'equally random'.

noise level tending to $\frac{1}{2}$ as x approaches the teacher's decision boundary. Since only training examples close to the decision boundary will decrease the generalization error ϵ_g significantly once a large enough number of training examples α have been received, this corresponds to increasing effective sign-flip noise as α increases. The generalization error $\epsilon_g(\alpha)$ must therefore be expected to decay much more slowly for weight noise than for sign-flip noise.

This expectation is confirmed by the simulation results in figure 5.8 for minimum entropy queries and random examples. Random examples yield a very slowly decaying generalization error; the results are compatible with a power law $\epsilon_g \propto \alpha^{-1/3}$ for large α (independently of $p_{\mathcal{N}}$) in agreement with the results of Ref. [KS92]. For query learning, we observe a faster power law decay ($\epsilon_g \propto \alpha^{-\delta}$ with δ approaching 2 for the largest values of $p_{\mathcal{N}}$ that we tested) for intermediate α which tails off into a nonzero asymptotic value of the generalization error for $\alpha \to \infty$. As expected from the discussion at the end of the previous section, this qualitative picture is independent of the value of the assumed noise level $p_{\mathcal{N}}$, due to the fact that in the presence of weight noise, the inference model remains incorrect for any value of $p_{\mathcal{N}}$. Indeed, the simulations

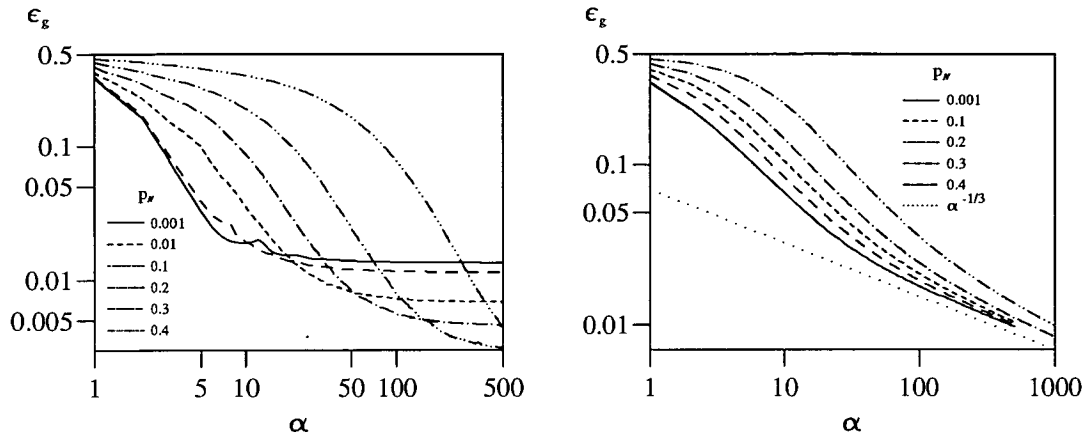

Figure 5.8. Left: Average generalization error ϵ_g achieved by query learning with various values of $p_{\mathcal{N}}$ (shown in the legend), for teacher weight noise $\gamma = 0.99$, vs. number of training examples α (note the logarithmic scales). For intermediate α, a power law decay is observed, tailing off to a nonzero asymptotic value of ϵ_g for larger α. See text for a discussion of the 'bumps' occurring for small $p_{\mathcal{N}}$. Right: Results for random examples, compatible with an asymptotic power law $\epsilon_g \propto \alpha^{-1/3}$ independently of $p_{\mathcal{N}}$. Statistical errors from the simulations (one standard deviation) are smaller than 2%.

results strongly suggest that the asymptotic generalization error $\epsilon_g(\alpha \to \infty)$ achieved by minimum entropy queries is nonzero for any $p_{\mathcal{N}} < \frac{1}{2}$, although it decreases towards zero with increasing $p_{\mathcal{N}}$. This means that query learning will always get trapped (on average) in self-confirming hypotheses far from the truth. Correspondingly, queries always perform worse than random examples for $\alpha \to \infty$, and the improvement factor κ shown in figure 5.9 tends to zero in this limit. For intermediate values of α, queries do achieve better generalization performance than random examples ($\kappa > 1$) due to the faster initial decay of the generalization error, but the improvement is not as significant as for the case of sign-flip noise discussed previously.

In summary, we have found that for weight noise, the efficacy of query learning based on the assumption of sign-flip noise is significantly reduced compared to the case where the true noise process is actually sign-flip noise. For any assumed noise level, queries perform worse than random examples when the number of training examples α becomes sufficiently large, because they eventually get stuck in self-confirming hypotheses far from the truth. Even for intermediate α, the fastest generalization error decay that we found was $\epsilon_g \propto \alpha^{-2}$, much slower than the exponential decay observed for sign-flip noise. We will explore in the next chapter whether these limitations of query learning

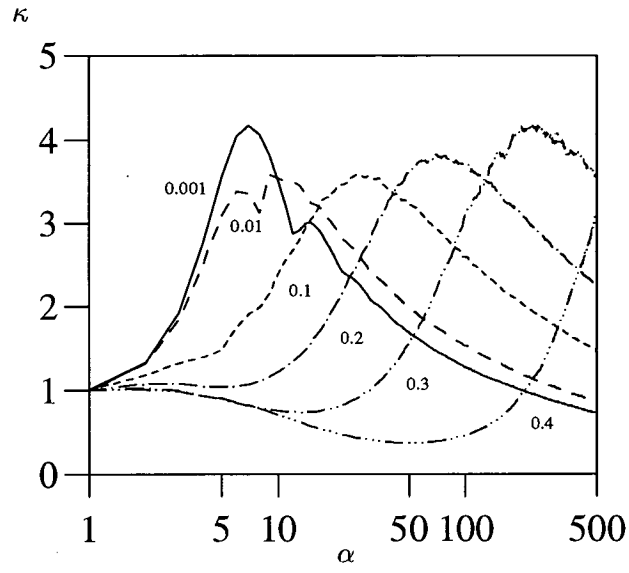

Figure 5.9. Improvement κ in generalization error due to query learning, for teacher weight noise $\gamma = 0.99$. Values of $p_{\mathcal{N}}$ are shown next to the curves.

can be overcome by *adapting* the assumed noise level $p_{\mathcal{N}}$ (or, more generally, any *hyperparameters* of the inference model) as more training examples are accumulated.

So far, we have not yet discussed the 'bumps' that appear in the learning curves $\epsilon_g(\alpha)$ for small $p_{\mathcal{N}}$ (see figures 5.8 and 5.9). They suggest that learning proceeds in 'bursts' when using minimum entropy queries. An intuitive understanding of this phenomenon can be gained if we look back to figure 5.6. We discussed above how corrupted training outputs act as barriers for the post-training student distribution, and explained that the number of uncorrupted training examples needed to overcome such barriers diverges as $p_{\mathcal{N}} \to 0$. The effect of this behaviour on the generalization error is shown on the left hand side of figure 5.6. The error varies slowly before a barrier is crossed, being dominated by contributions of students near the barrier. After the barrier has been crossed, the student distribution becomes once again more sensitive to the new training examples, leading to much larger variations of the generalization error (which can be either positive or negative, depending on whether the new training outputs are corrupted or not). The corresponding bumps in the variation of the generalization error with α are most pronounced for small $p_{\mathcal{N}}$, where the barrier crossing time is long enough for single barrier crossings to be distinguished clearly.

The discussion so far explains why bumps in the learning curves $\epsilon_g(\alpha)$ can occur, and why they must be expected to be most pronounced for small values of $p_{\mathcal{N}}$. It only

remains to be discussed why these bumps do not disappear when an average over all training sets is taken, as they do for sign-flip noise (see figure 5.4). For sign-flip noise, every training output is by definition corrupted with equal probability. The values of α at which bumps in $\epsilon_g(\alpha)$ occur for a particular sequence of training examples are therefore randomly distributed, leading to a smooth average generalization error. Weight noise, on the other hand, can be thought of as effective sign-flip noise which becomes increasingly strong as α increases and ϵ_g decreases. It seems plausible that this should lead to preferred α values at which bumps would occur, preventing them from being washed out by a training set average.

At this point, one might be tempted to rephrase some of the above results in terms normally reserved for learning by humans: Incorrect information received by a learner can lead to unjustified 'false beliefs' or prejudice (corresponding to the barriers referred to above) which can slow down learning significantly; only when enough evidence has been received to demonstrate that the prejudice is founded on false beliefs can new information be used beneficially. The strength of the prejudice and the amount of evidence needed to overcome it depends on how strongly the learner believes in the correctness of the information she receives. Learning can therefore proceed in 'bursts' of refinement and rejection of hypotheses which approximate the underlying 'truth' more and more closely. Such a rephrasing of our findings is of course purely speculative, and we do not mean to imply a connection between the results obtained above and real-world human learning. However, the question of whether such a connection could exist appears to be an interesting topic for future research.

5.5.2 Thermodynamic limit, $N \to \infty$

We now consider the thermodynamic limit $N \to \infty$ of a binary perceptron student and teacher with high-dimensional input (and weight) space, in order to see which of the conclusions obtained above for an extremely small system ($N = 2$) remain valid. The general derivation of the bisection criterion (5.19) for minimum entropy queries assuming the inference model is correct is independent of the system size N and therefore remains unchanged. However, the search for queries which fulfill this criterion becomes non-trivial for large N, when both the input space – in which the search for queries takes place – and the weight space – on which the post-training student distribution $P(\mathcal{N}|\Theta^{(p)})$, which queries should bisect, is defined – are high-dimensional. The 'query by committee algorithm' proposed in Ref. [SOS92] provides a solution to this problem: For the selection of each new query, one first samples $2k$ students $\mathcal{N}_\gamma$, $\gamma = 1 \ldots 2k$, from the post-training distribution $P(\mathcal{N}|\Theta^{(p)})$. With these

students, a finite sample approximation to the weight space integral in the bisection criterion (5.19) is then constructed as

$$\int d\mathcal{N}\ \Theta(f_{\mathcal{N}}(\mathbf{x}))P(\mathcal{N}|\Theta^{(p)}) \approx \frac{1}{2k}\sum_{\gamma=1}^{2k}\Theta(f_{\mathcal{N}_\gamma}(\mathbf{x})).$$

The requirement that this average be equal to $\frac{1}{2}$ for a bisection query $\mathbf{x}$ then simply translates into the condition that exactly k of the $2k$ students predict output $+1$ for the input $\mathbf{x}$, while the other k predict -1. In other words, the 'committee' of students has to be in 'maximal disagreement' over the query $\mathbf{x}$. Approximate bisection queries can be found by filtering inputs which meet this requirement from a stream of random inputs. By construction, the query by committee algorithm yields exact bisection in the limit[8] $k \to \infty$. We focus on this limit in the following; however, previous analyses [SOS92] suggest that the results would remain qualitatively unchanged for finite k.

As pointed out in Section 5.5, the average generalization as a function of the normalized number of training examples α can be calculated from the average overlap[9] R of the teacher and student weight vectors as defined in (5.22). This overlap is conveniently obtained from a replica calculation of the average free energy of the Gibbs post-training student distribution (5.14)

$$-\beta f = \frac{1}{N}\langle \ln Z\rangle_{P(\Theta^{(p)}|\mathcal{V})P(\mathcal{V})} \qquad Z = \int d\mathcal{N}\tilde{P}(\mathcal{N})\exp(-\beta E_{\mathrm{t}}(\mathcal{N},\Theta^{(p)})) \tag{5.26}$$

As pointed out above, the average over the teacher prior $P(\mathcal{V})$ can actually be dropped, since the result for a particular teacher does not depend on the orientation of its weight vector once the average over all training sets is carried out.

An outline of the replica calculation of the free energy is given in Appendix 5.8. One obtains $-\beta f$ in the saddle point form

$$-\beta f = \operatorname{extr}_{q,R}\left\{\frac{1}{2}\left[\frac{q-R^2}{1-q}+\ln(1-q)\right]+\frac{1}{N}\sum_{\mu=0}^{p-1} 2\int_0^\infty Dh\int_{-\infty}^\infty Dt\right.$$
$$\left.\left[(1-p_\nu)\ln\Big(H(u)+e^{-\beta}(1-H(u))\Big)+p_\nu\ln\Big(e^{-\beta}H(u)+1-H(u)\Big)\right]\right\} \tag{5.27}$$

[8] This statement is true for any fixed system size N, i.e., if the limit $k \to \infty$ is taken *before* the thermodynamic limit $N \to \infty$. In the analytical calculations, we actually reverse the order of these limits and verify *a posteriori* that the results are correct.

[9] We use the same notation for R and its average over all training sets and teachers since the two are identical with probability one in the thermodynamic limit.

where we have used the shorthand $Dz = \exp(-\frac{1}{2}z^2)\,dz/\sqrt{2\pi}$ and $H(z) = \int_z^\infty Dz'$. In (5.27), q is the (average) overlap between two students sampled from the Gibbs distribution $P(\mathcal{N}|\Theta^{(p)})$, which can also be interpreted as the normalized length of the *average* weight vector from $P(\mathcal{N}|\Theta^{(p)})$:

$$q = \frac{1}{N}(\overline{\mathbf{w}}_{\mathcal{N}}^{p})^{\mathrm{T}}\overline{\mathbf{w}}_{\mathcal{N}}^{p} = \frac{1}{N}(\overline{\mathbf{w}}_{\mathcal{N}}^{p})^2 \qquad \overline{\mathbf{w}}_{\mathcal{N}}^{p} = \langle \mathbf{w}_{\mathcal{N}} \rangle_{P(\mathcal{N}|\Theta^{(p)})} \tag{5.28}$$

Like R, q is self-averaging in the thermodynamic limit and therefore identical to its average over training sets with probability one. The variable u appearing in (5.27) is defined as

$$u = \rho h + \tau t \qquad \rho = -\frac{\tilde{R}^{\mu}}{\sqrt{1-q}} \qquad \tau = \left(\frac{\tilde{q}^{\mu} - (\tilde{R}^{\mu})^2}{1-q}\right)^{1/2} \tag{5.29}$$

where

$$\tilde{R}^{\mu} = \frac{R - R^{\mu}q^{\mu p}/q^{\mu}}{(1/\gamma^2 - (R^{\mu})^2/q^{\mu})^{1/2}} \qquad \tilde{q}^{\mu} = q - (q^{\mu p})^2/q^{\mu}. \tag{5.30}$$

The new order parameters appearing in these definitions are q^{μ} and R^{μ}, which are the equivalents of q and R for training sets $\Theta^{(\mu)}$ of size $\mu < p$, and $q^{\mu p}$, the overlap between two students trained on μ and p training examples, respectively:

$$q^{\mu p} = \frac{1}{N}(\overline{\mathbf{w}}_{\mathcal{N}}^{\mu})^{\mathrm{T}}\overline{\mathbf{w}}_{\mathcal{N}}^{p} \qquad \overline{\mathbf{w}}_{\mathcal{N}}^{\mu} = \langle \mathbf{w}_{\mathcal{N}} \rangle_{P(\mathcal{N}|\Theta^{(\mu)})} \tag{5.31}$$

The latter arise as the overlaps of the committee members defining the selection of the $(\mu+1)$-th query and the students produced after training on the complete training set $\Theta^{(p)}$. In the thermodynamic limit, all overlap parameters can be replaced by functions of the continuous variables[10] $\alpha = p/N$ and $\alpha' = \mu/N < \alpha$ ($q = q^p \to q(\alpha)$, $q^{\mu} \to q(\alpha')$, similarly for R and R^{μ}, and $q^{\mu p} \to q(\alpha', \alpha)$). This turns the saddle point equations resulting from the extremum condition in (5.27) into integral equations for $q(\alpha)$ and $R(\alpha)$. The main hurdle is then the determination of $q(\alpha', \alpha)$, for which no independent saddle point equation exists.

In the relatively simple case where the inference model *is* correct, i.e., $p_{\mathcal{N}} = p_{\mathcal{V}}$ and $\gamma = 1$ (no weight noise), we shall now prove that in fact $q(\alpha', \alpha) = q(\alpha')$ for all $\alpha > \alpha'$. This relation has previously been assumed without proof in an analysis of the query by committee algorithm [SOS92]. For the proof, we temporarily revert to the notation in terms of μ and p rather than α' and α. We assume as usual that all overlap parameters

[10] The definition of α for finite N, $\alpha = p/(N-1)$, becomes identical to the simpler form $\alpha = p/N$ in the thermodynamic limit.

are self-averaging. It then suffices to show that

$$\langle q^{\mu p} \rangle_{P(\Theta^{(p)})} = \langle q^{\mu} \rangle_{P(\Theta^{(p)})} \tag{5.32}$$

in order to prove the desired relation $q(\alpha', \alpha) = q(\alpha')$. Only the left hand side depends on the training examples $\mu + 1, \mu + 2 \ldots p$, which we collectively denote by $\Theta^{(p)} \setminus \Theta^{(\mu)}$. Inserting the definitions of $q^{\mu p}$ and q^{μ}, eqs. (5.28, 5.31), we rewrite the desired relation as

$$\left\langle \langle \overline{\mathbf{w}}_{\mathcal{N}}^{p} \rangle^{\mathrm{T}}_{P(\Theta^{(p)} \setminus \Theta^{(\mu)} | \Theta^{(\mu)})} \overline{\mathbf{w}}_{\mathcal{N}}^{\mu} \right\rangle_{P(\Theta^{(\mu)})} = \left\langle (\overline{\mathbf{w}}_{\mathcal{N}}^{\mu})^{\mathrm{T}} \overline{\mathbf{w}}_{\mathcal{N}}^{\mu} \right\rangle_{P(\Theta^{(\mu)})}.$$

This in turn follows from

$$\langle \overline{\mathbf{w}}_{\mathcal{N}}^{p} \rangle_{P(\Theta^{(p)} \setminus \Theta^{(\mu)} | \Theta^{(\mu)})} = \overline{\mathbf{w}}_{\mathcal{N}}^{\mu}. \tag{5.33}$$

Writing out the averages in the definition of $\overline{\mathbf{w}}_{\mathcal{N}}^{p}$ and $\overline{\mathbf{w}}_{\mathcal{N}}^{\mu}$, one sees that it is sufficient to show that

$$\left\langle P(\mathcal{N} | \Theta^{(p)}) \right\rangle_{P(\Theta^{(p)} \setminus \Theta^{(\mu)} | \Theta^{(\mu)})} = P(\mathcal{N} | \Theta^{(\mu)})$$

which by induction holds if

$$\left\langle P(\mathcal{N} | \Theta^{(\mu+1)}) \right\rangle_{P(y^{\mu+1}, \mathbf{x}^{\mu+1} | \Theta^{(\mu)})} = P(\mathcal{N} | \Theta^{(\mu)}) \tag{5.34}$$

for all μ. This finally, can be derived by averaging the relation (see eqs. (5.11, 5.12))

$$P(\mathcal{N} | \Theta^{(\mu+1)}) = \frac{\tilde{P}(y^{\mu+1} | \mathbf{x}^{\mu+1}, \mathcal{N}) P(\mathcal{N} | \Theta^{(\mu)})}{\tilde{P}(y^{\mu+1} | \mathbf{x}^{\mu+1}, \Theta^{(\mu)})} \tag{5.35}$$

over $y^{\mu+1}$ for fixed $\mathbf{x}^{\mu+1}$ and $\Theta^{(\mu)}$, using the fact that for a correctly specified inference model the true distribution of $y^{\mu+1}$ equals the assumed distribution, i.e.,

$$P(y^{\mu+1} | \mathbf{x}^{\mu+1}, \Theta^{(\mu)}) = \tilde{P}(y^{\mu+1} | \mathbf{x}^{\mu+1}, \Theta^{(\mu)})$$

so that the denominator in (5.35) is cancelled exactly. Eq. (5.34) then follows from the normalization condition

$$\int dy^{\mu+1} \tilde{P}(y^{\mu+1} | \mathbf{x}^{\mu+1}, \mathcal{N}) = 1.$$

Note that the above proof does not rely on any specific properties of the query selection scheme as long as it is sequential in the sense that the probability distribution of each

query only depends on the existing training set[11] (and not on any outputs or inputs that are received later).

The relations (5.32, 5.33) which we have proved may seem counter-intuitive at first, suggesting that since the average student weight vector remains unchanged, no learning can take place as more and more training examples are received. This interpretation would only be correct, however, if the average over the 'future' training examples $\Theta^{(p)} \setminus \Theta^{(\mu)}$ in (5.33) was absent. As it stands, eq. (5.33) expresses the fact that if we only know that the training examples $\mu+1, \mu+2 \ldots p$ have been received, but not what their actual values are, then we have not gained any additional knowledge about the student distribution. This reasoning depends on the correctness of the inference model because an incorrect inference model introduces a bias into the post-training student distribution $P(\mathcal{N}|\Theta^{(\mu)})$ which will cause it to be skewed away from the teacher posterior distribution $P(\mathcal{V}|\Theta^{(\mu)})$. As more training examples are learned, this bias is expected to be reduced even when an average over future training examples is taken.

Sign-flip noise

Let us now explore the performance of minimum entropy queries – selected assuming the inference model is correct – in the case when the training outputs are corrupted by sign-flip noise. Specializing further to the case where the assumed student noise level $p_{\mathcal{N}}$ equals the true noise level $p_{\mathcal{V}}$, which means that the inference model *is* correct, we can use the relation $q(\alpha', \alpha) = q(\alpha')$ derived above to 'close' the system of saddle point equations for $q(\alpha)$ and $R(\alpha)$ arising from (5.27). The equations can then be solved numerically, and one obtains the results shown in figure 5.10. It can clearly be seen that for large α, the generalization error decays exponentially fast as α increases. The decay constant can be obtained analytically, as explained in Appendix 5.8, with the result

$$\epsilon_g(\alpha) \propto \exp[-(\ln 2 + p_{\mathcal{V}} \ln p_{\mathcal{V}} + (1-p_{\mathcal{V}}) \ln(1-p_{\mathcal{V}}))\alpha]. \tag{5.36}$$

For the noise free case $p_{\mathcal{V}} = 0$, one has $\epsilon_g \propto \exp(-\alpha \ln 2)$ in agreement with the analysis in Ref. [SOS92]. For almost random training outputs ($\frac{1}{2} - p_{\mathcal{V}} \ll 1$), on the other hand, one reads off $\epsilon_g \propto \exp(-2(\frac{1}{2} - p_{\mathcal{V}})^2 \alpha)$, with the decay constant decreasing to zero for $p_{\mathcal{V}} \to \frac{1}{2}$ as expected. We can conclude that the main qualitative effect of query learning, namely, the exponential decay of the generalization error, is the same in the limit of a very large system ($N \to \infty$) as for the very small system ($N = 2$) studied in the previous section, as long as the inference model *is* correct.

[11] This implies, of course, that the proof is also valid for learning from random examples, where training inputs are sampled independently from some fixed distribution.

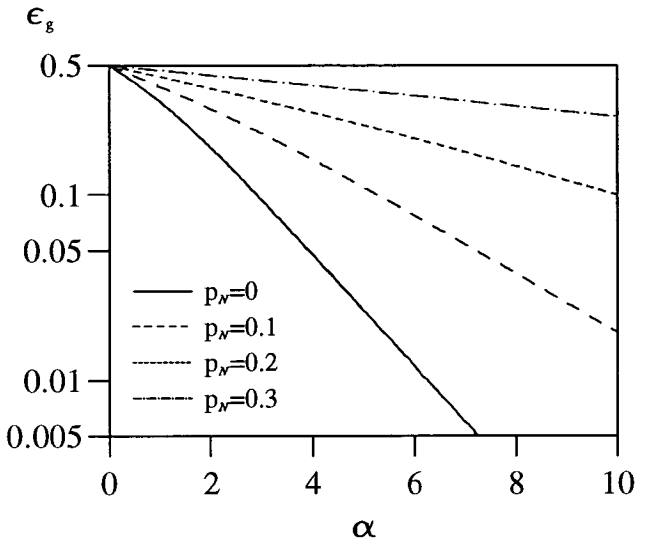

Figure 5.10. Generalization error ϵ_g vs. α for query learning in the $N \to \infty$ binary perceptron with a correct inference model ($p_N = p_V$), as predicted by the replica calculation. For large α, the decay of ϵ_g is exponential (note the logarithmic ϵ_g-axis), with decay constant given by (5.36).

In the case where the inference model is incorrect, i.e., $p_N \neq p_V$, an analytical treatment is much more involved due to the lack of a simple relation between $q(\alpha', \alpha)$ to $q(\alpha)$. In principle, it should be possible to obtain the $q(\alpha', \alpha) = q^{\mu p}$ from a (double) replica calculation of the correlations between the free energies for training sets of size μ and p, as outlined in Appendix 5.8. However, conceptual and computational difficulties with this approach remain, and we have therefore resorted to Monte Carlo simulations (see, e.g., [AT87]) in order to gain a qualitative understanding of the large system behaviour. The simulations where carried out as follows: Each time a new training example was added to the training set, 5000 Monte Carlo steps in the student weight vector space (with the training error (5.15) as energy function) were taken for equilibration towards the post-training Gibbs distribution (5.14). Of the next 5000 Monte Carlo steps, 20% were sampled to calculate the 'thermal' averages over the post-training distribution. Throughout, the acceptance ratio for Monte Carlo steps was kept around 0.2 by adaptation of the stepsize for trial weight vector changes; the value 0.2 was obtained from a rough empirical minimization of the correlations between successive steps. The next query was then selected to be orthogonal to the thermally averaged student weight vector, but otherwise random; this procedure for selecting queries is computationally cheaper than the query by committee approach and yields

the same results in the thermodynamic limit (see Appendix 5.8). The corresponding uncorrupted teacher output was calculated, its sign reversed with probability p_ν, and the whole procedure repeated. The final results were obtained by averaging over 50–100 training sets generated this way.

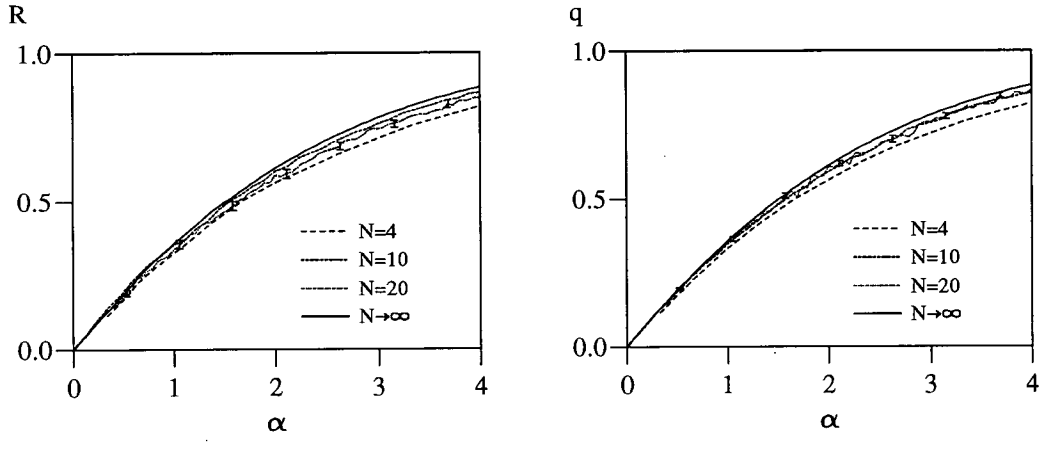

Figure 5.11. Left: Average student-teacher overlap R for query learning in the binary perceptron with a correct inference model, $p_\mathcal{N} = p_\nu = 0.1$. Shown are the results of Monte Carlo simulations for system size $N = 4, 10, 20$; the agreement with the theoretical prediction for $N \to \infty$ is already fairly good for $N = 10, 20$. Some exemplary error bars (± 1 standard deviation) for the simulation results are shown for $N = 20$, where they are largest since only 50 training sets were sampled. Right: Average student-student overlap q. The similarity to the graph on the left is not accidental: For the case of a correct inference model, the averages of q and R must be identical (see Appendix 5.8).

To verify the correctness of the Monte Carlo simulations, we first compared them to the analytical results for the case $p_\mathcal{N} = p_\nu$ discussed above. Figure 5.11 shows the α dependence of the student-teacher overlap R obtained from Monte Carlo simulations for system sizes $N = 4, 10, 20$; note that even for such moderate system sizes, the agreement with the analytical results for $N \to \infty$ is good. Moving on the case of an incorrect inference model, $p_\mathcal{N} \neq p_\nu$, we would like to know whether the problem of self-confirming hypotheses far from the truth persists for large N. The simulation results shown in figures 5.12 and 5.13 answer this question in the affirmative; in fact, they suggest that the average generalization error ϵ_g for fixed and sufficiently large $\alpha = p/(N-1)$ increases with the system size N, so that the problem of self-confirming hypotheses is, if anything, exacerbated for larger systems. This also implies that the value of $p_\mathcal{N}$ at which, for given p_ν, the phase transition from self-confirming hypotheses

to exponentially decaying generalization error occurs, must be expected to increase with N. Given the computational resources that would be required for Monte Carlo simulations of larger systems, we are not at present in a position to obtain more precise information about the location of the phase transition in the $N \to \infty$ limit. We therefore leave this topic (and, in particular, the question of whether the phase transition for an infinitely large system occurs at the largest allowed value, $p_{\mathcal{N}} = p_{\mathcal{V}}$, or below) as an interesting avenue for further research.

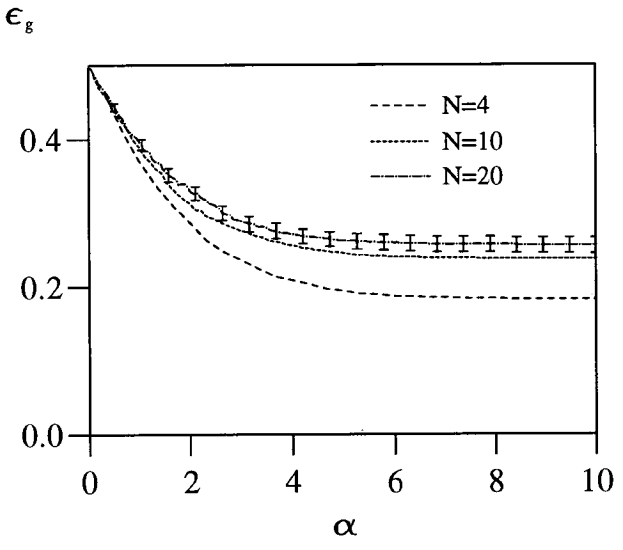

Figure 5.12. Average generalization error ϵ_g vs. α for query learning with $p_{\mathcal{N}} = 0.0$ from a teacher with sign-flip noise level $p_{\mathcal{V}} = 0.2$, as obtained from Monte Carlo simulations. The results suggest that for large α, ϵ_g is an increasing function of system size N. Exemplary error bars (± 1 standard deviation) are shown for $N = 20$.

Weight noise

Consider now the case of where training outputs are corrupted by weight noise. As emphasized above, the student's inference model is then incorrect for any value of $p_{\mathcal{N}}$. Consequently, we find ourselves again in a situation where no obvious relationship between the order parameters $q(\alpha', \alpha)$ and $q(\alpha')$ exists, rendering an analytical treatment distinctly non-trivial. As above, we therefore rely on clues from Monte Carlo simulations to analyse the large system behaviour. Turning first to the question of the occurrence of self-confirming hypotheses, we see from the results shown in figures 5.14 that as in the case of sign-flip noise, the average generalization error normally increases

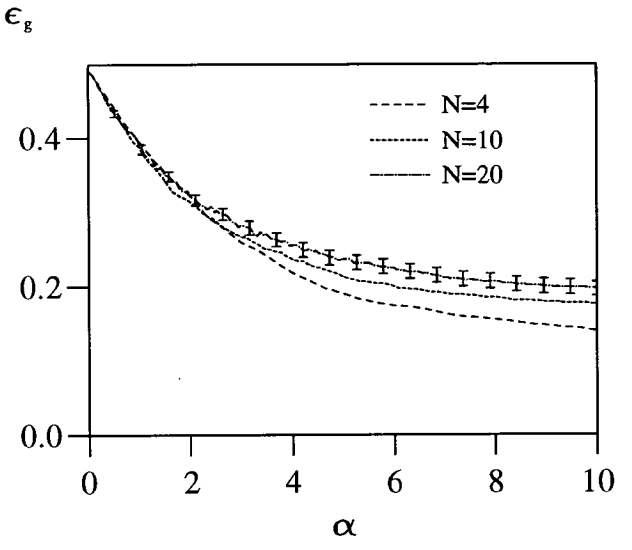

Figure 5.13. As figure 5.12, but for assumed noise level $p_{\mathcal{N}} = 0.01$.

with N, making self-confirming hypotheses more likely for large systems. Based on the results for $N = 2$ obtained in Section 5.5.1, we can therefore conclude that for large system size N, the student distribution will eventually get trapped in self-confirming hypotheses far from the truth as the number of training examples α increases, whatever the value of the assumed (sign-flip) noise level $p_{\mathcal{N}}$. For a qualitative discussion of the behaviour of the generalization error for intermediate values of α, we show in figure 5.15 a case with relatively weak weight noise, $\gamma = 0.99$, and moderate student noise level, $p_{\mathcal{N}} = 0.1$, where self-confirming hypotheses only manifest themselves for rather large α. A power-law behaviour $\epsilon_g \propto \alpha^{-\delta}$ can be observed, with an exponent δ of order unity. This is in qualitative agreement with the high temperature analysis ($\beta \to 0$ at $\tilde{\alpha} = \beta\alpha = \text{const}$) carried out in Appendix 5.8, which gives the asymptotic behaviour $\epsilon_g \propto 1/\tilde{\alpha}$. Note, however, that the high temperature limit cannot in general be relied upon to give the correct asymptotic behaviour at finite temperature, i.e., $p_{\mathcal{N}} < 1/2$ (see, e.g., [SST92, Spo95]). The above results for query learning should be compared with the corresponding power laws for learning from random examples in the presence of weight noise: the high temperature analysis yields $\epsilon_g \propto \tilde{\alpha}^{-1/2}$, whereas in Ref. [GT90], $\epsilon_g \propto \alpha^{-1/4}$ was derived within the replica symmetric approximation. Qualitatively, we therefore obtain the same picture regarding the efficacy of query learning for large N as for the small system limit $N = 2$: Before the onset of self-confirming hypotheses, queries yield a power law decay of the generalization error which is faster than that for

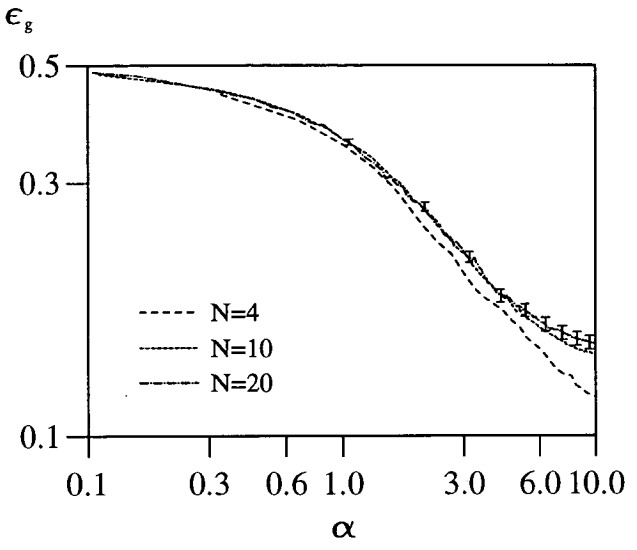

Figure 5.14. Average generalization error ϵ_g vs. α for query learning with $p_N = 0.01$ from a teacher with weight noise $\gamma = 0.8$, as obtained from Monte Carlo simulations. The results suggest that for large α, ϵ_g is an increasing function of system size N. Exemplary error bars (± 1 standard deviation) are shown for $N = 20$.

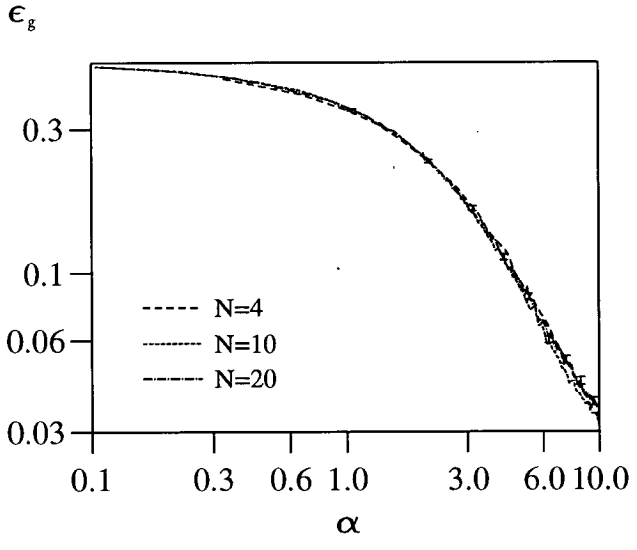

Figure 5.15. As figure 5.14, but for assumed noise level $p_N = 0.1$ and teacher weight noise $\gamma = 0.99$. This case shows more clearly the power-law decay of the generalization error ϵ_g in the regime before it reaches its nonzero asymptotic value.

random examples. This implies that, as for $N = 2$, the improvement factor κ will be larger than one for intermediate α, implying improved generalization performance from query learning, but must decay to zero for large α when self-confirming hypotheses far from the truth start to degrade the performance achieved by queries.

5.6 Summary and conclusion

In this chapter, we have studied query learning in the absence of prior knowledge about the teacher space. The introduction of query learning assuming the inference model is correct has, in effect, lead us back to a traditional Bayesian framework for query selection. Applying this to learning with linear and binary perceptron students, we have found two potential pitfalls of assuming the inference model is correct. Firstly, queries selected to optimize a given objective function such as the generalization error may in fact lead to a higher value of this objective function than random examples. Secondly, query learning can generate self-confirming hypotheses far from the truth, which means that the optimal approximation to the teacher is not learned even in the limit of an infinitely large number of training examples. These problems obviously have to be overcome if query learning is to be useful and reliable in practical applications; a potential solution is proposed and investigated in the next chapter.

The results summarized above also add to our understanding of the effect of noise on query learning in binary output systems. In Chapter 3, it was suggested that queries generally yield drastic reductions in generalization error—compared to random examples—for binary or discrete output systems. In light of the findings in the present chapter, this statement has to be qualified slightly: The efficacy of query learning (as captured in the improvement factor κ) depends significantly on the noise process corrupting the training data. Loosely speaking, the two noise models that we have considered for the binary perceptron can be distinguished according to whether they preserve the discontinuous nature of the underlying rule or not. For sign-flip noise, the probability of obtaining output $+1$, say, is still (as in the absence of noise) a discontinuous function of the input $\mathbf{x}$ for a fixed teacher weight vector $\mathbf{w}_\nu$ (or vice versa), while for weight noise, this discontinuity is 'smoothed out'. Correspondingly, we found in the first case that queries still yield an exponential decay of the generalization error resembling the behaviour for a noise free binary perceptron teacher ($p_\nu = p_\mathcal{N} = 0$, see also [SOS92]), while in the second case, a power law decay was obtained at best and the improvement in generalization performance over random examples remained bounded, reminiscent of the results for students with continuous outputs (see Chapters 3 and 4). While it seems likely that the crucial difference between the two cases

is indeed the discontinuity in the probabilistic input-output relation, further work is clearly needed in this direction, in particular with the aim of identifying more generally the classes of learning scenarios for which query learning will achieve the most significant improvements in generalization performance. A closer investigation of the typical noise processes occurring in practical supervised learning problems with discrete outputs would be interesting for the same reason.

Finally, note that as emphasized above, the scenarios considered in this chapter contain only fairly mild forms of inference model misspecification. An extension of the analysis to problems with mismatched student and teacher space or mismatched priors would certainly be desirable. This would appear to be a fruitful avenue for future research, which could clarify the extent to which our conclusions are general rather than problem specific.

5.7 Appendix: Analytical results for the binary perceptron, $N = 2$

In this appendix, we outline the calculation of the average generalization error achieved by minimum entropy queries (selected assuming the inference model is correct) for binary perceptrons of size $N = 2$. In particular, we derive the result (5.24) for assumed noise free outputs ($p_{\mathcal{N}} = 0$), when the training outputs are in fact corrupted by sign-flip noise with probability $p_{\mathcal{V}}$.

The quantity that we want to calculate is the average generalization error, defined by

$$\epsilon_g = \langle \epsilon_g(\mathcal{N}, \mathcal{V}) \rangle_{P(\mathcal{N}|\Theta^{(p)})P(\Theta^{(p)}|\mathcal{V})P(\mathcal{V})} \,. \tag{5.37}$$

As explained in the text, the average over students $\mathcal{N}$ and training sets $\Theta^{(p)}$ makes the result independent of the teacher, so we can drop the average over the teacher prior $P(\mathcal{V})$ and simply consider a fixed teacher. We use the mapping of weight vector space and input space onto the interval $[-1, 1]$ with wrap-around boundary conditions as explained in Section 5.5.1, and consider the teacher given by $w_{\mathcal{V}} = 0$ in this representation. To calculate the dependence of the average generalization error (5.37) on the number of training examples p, we analyse how the averaged post-training student distribution

$$P(\mathcal{N}|\mathcal{V}, p) = \left\langle P(\mathcal{N}|\Theta^{(p)}) \right\rangle_{P(\Theta^{(p)}|\mathcal{V})} \tag{5.38}$$

changes with p. An explicit example of $P(\mathcal{N}|\mathcal{V}, p)$ for $p = 0, 1, 2$ is shown figure 5.16.

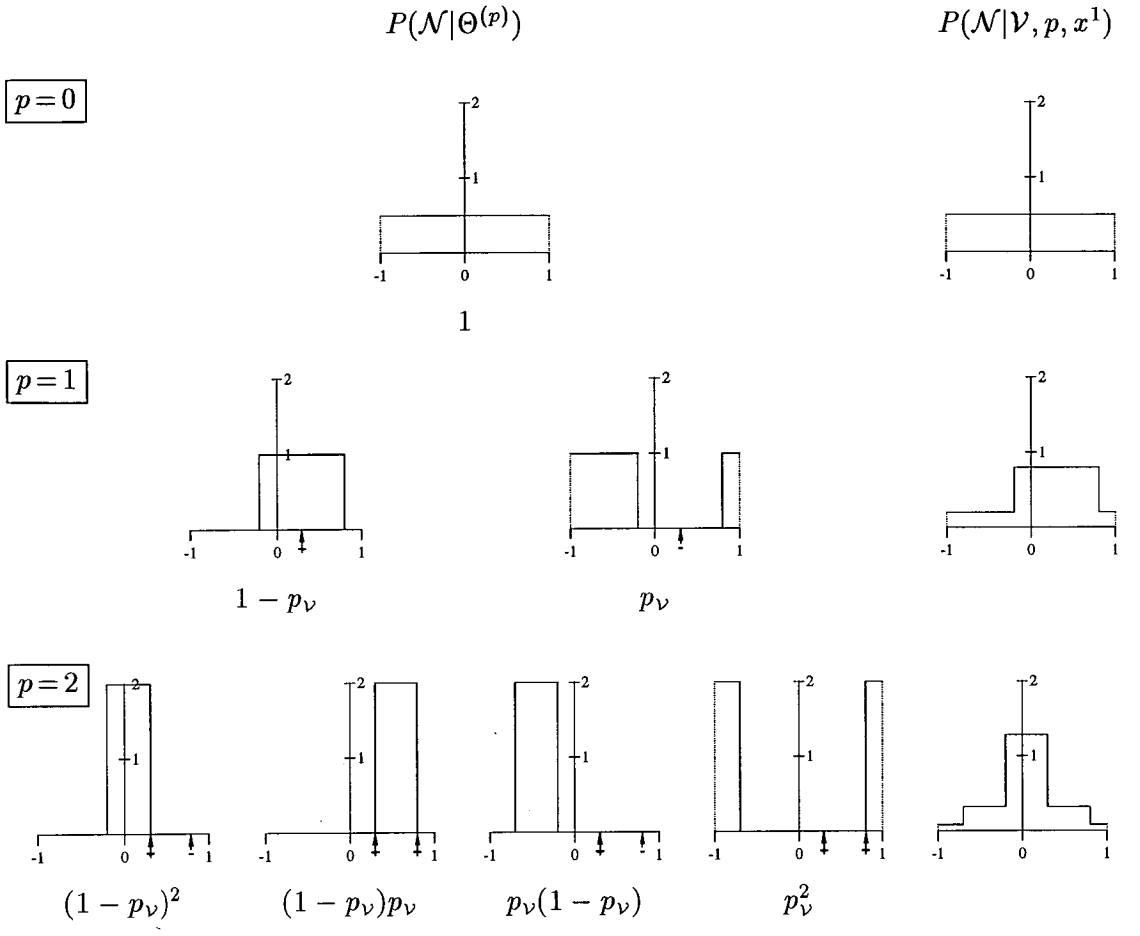

Figure 5.16. Student distributions for query learning with assumed noise level $p_{\mathcal{N}} = 0$, for $p = 0, 1, 2$ training examples (top to bottom). The teacher $w_\nu = 0$ generates training outputs with sign-flip noise $p_\nu = 0.2$; the first training input $x^1 = 0.3$ is chosen randomly. Training inputs are marked by arrows, with the corresponding outputs (± 1) shown underneath. $P(\mathcal{N}|\mathcal{V},p,x^1)$ is shown on the extreme right; the other distributions show its decomposition into the $P(\mathcal{N}|\Theta^{(p)})$ for the different possible training sets $\Theta^{(p)}$ (see eq. (5.38)). The weight of each of the $P(\mathcal{N}|\Theta^{(p)})$, shown underneath the diagrams, is given by the probability of obtaining the corresponding output sequence, each uncorrupted/corrupted output contributing a factor of $1-p_\nu$ or p_ν, respectively. We have only shown one of the two possible values of the second query x^2; see footnote on p.108. Note finally that the regime $p \leq 2$ is 'pathological' in the sense that query selection is independent of the training outputs received previously; for $p \geq 3$, this is no longer the case.

The first training input is selected randomly; in the figure, we chose $x_1 = 0.3$. Thereafter, the selection of minimum entropy (i.e., bisection) queries becomes a deterministic

process depending only on the teacher outputs[12]. For fixed x^1, there are therefore only 2^p different post-training distributions $P(\mathcal{N}|\Theta^{(p)})$ for training sets of size p. Each of them is non-zero only on an interval of length 2^{-p+1}, where it has the constant value 2^{p-1} derived from the normalization condition. The union of all these (disjoint) intervals covers the whole weight space $[-1,1]$ (see figure 5.16). In the average post-training distribution (5.38), each of the $P(\mathcal{N}|\Theta^{(p)})$ is simply multiplied by the probability of the corresponding sequence of training outputs, which is $p_\nu^k(1-p_\nu)^{p-k}$ if exactly k outputs are corrupted. The distribution $P(\mathcal{N}|\mathcal{V},p,x^1)$, where x^1 denotes the fixed first training input, is therefore piecewise constant over 2^p sub-intervals[13] of $[-1,1]$ of equal length 2^{-p+1}, which we shall call 'sectors'.

Consider now what happens to the sectors of $P(\mathcal{N}|\mathcal{V},p,x^1)$ when p is increased by one. The following explanation is probably most easily understood by looking at the graphs in the rightmost column of figure 5.16, remembering that p increases from top to bottom. As we increase p by one, each sector of $P(\mathcal{N}|\mathcal{V},p,x^1)$ is split into two equal halves by a bisection query for the corresponding $P(\mathcal{N}|\Theta^{(p)})$. The half sector closer to the true teacher corresponds to $P(\mathcal{N}|\Theta^{(p+1)})$ for the case where the new training output y^{p+1} is uncorrupted; hence, the value of $P(\mathcal{N}|\mathcal{V},p+1,x^1)$ there is obtained by multiplying $P(\mathcal{N}|\mathcal{V},p,x^1)$ by $2(1-p_\nu)$. The factor of 2 comes from the normalization of $P(\mathcal{N}|\Theta^{(p+1)})$, while the factor $(1-p_\nu)$ is the probability of receiving an uncorrupted output. The multiplication factor for the other half sector, corresponding to a corrupted training output, is $2p_\nu$. Together, the two new sectors contribute the same probability mass to $P(\mathcal{N}|\mathcal{V},p+1,x^1)$ as their 'parent' sector to $P(\mathcal{N}|\mathcal{V},p,x^1)$.

Let us now examine the effect on the average generalization error of these changes in the average student distribution. The generalization error (5.23) equals $\epsilon_g(\mathcal{N},\mathcal{V}) = |w_\mathcal{N}|$ for the teacher $w_\nu = 0$ considered here. We have to distinguish two kinds of sectors of $P(\mathcal{N}|\mathcal{V},p,x^1)$: Firstly, there are the $2^p - 2$ sectors for which the generalization error is either given by $w_\mathcal{N}$, or by $-w_\mathcal{N}$, in the whole sector. In the remaining two sectors, containing either the true teacher $w_\nu = 0$ or its 'opposite' on the circular weight space (corresponding to the boundary point $-1 = 1 \pmod 2$ of $[-1,1]$) the generalization error is given by $w_\mathcal{N}$ in part of the sector and by $-w_\mathcal{N}$ in the rest. For the first class of sectors, the effect of incrementing p on their contribution to the average

[12] In fact, the situation is slightly more complicated since there are always *two* possible bisection queries for $p \geq 1$, related by $\mathbf{x} \to -\mathbf{x}$ in the vector notation or $x \to x \oplus 1$ (addition modulo 2) in the interval representation. However, it is easy to verify that these two inputs, together with their corresponding corrupted/uncorrupted teacher outputs, have exactly the same effect on the post-training student distribution, so they need not be distinguished for our purposes.

[13] Due to the assumed wrap-around boundary conditions, the two sub-intervals containing the endpoints of $[-1,1]$ are considered as one.

generalization error can be derived as follows: Consider a sector of $P(\mathcal{N}|\mathcal{V}, p, x^1)$ with probability mass m and midpoint $w > 0$, contributing mw to the average generalization error. As explained above, this sector splits into two subsectors of equal length 2^{-p} in $P(\mathcal{N}|\mathcal{V}, p+1, x^1)$, with midpoints $w \mp 2^{-p-1}$ and probability masses $m[\frac{1}{2} \pm (\frac{1}{2} - p_\mathcal{V})]$; the one with the midpoint closer to the teacher $w_\mathcal{V} = 0$ carries the larger probability mass as pointed out above. These two subsectors together contribute

$$m(1-p_\mathcal{V})(w - 2^{-p-1}) + mp_\mathcal{V}(w + 2^{-p-1}) = mw - m(1-2p_\mathcal{V})2^{-p-1}$$

to the generalization error, giving a reduction of $m(1-2p_\mathcal{V})2^{-p-1}$ from the value for p training examples. This result also holds for sectors with midpoints $w < 0$, as can easily be checked. Summing over all sectors in the first class, we therefore obtain a reduction in average generalization error of

$$\Delta\epsilon_g(\text{sectors in first class}) = M(1-2p_\mathcal{V})2^{-p-1} \tag{5.39}$$

where M is the joint probability mass of all sectors in the first class. Due to the normalization of $P(\mathcal{N}|\mathcal{V}, p, x^1)$, M can be expressed in terms of the probability mass of the two sectors in the second class; since these correspond to training sets of size p with either all training outputs uncorrupted or all corrupted, it follows that

$$M = 1 - (1-p_\mathcal{V})^p - p_\mathcal{V}^p.$$

Now we only need to find the reduction in average generalization error contributed by the sectors in the second class. Consider first the sector of $P(\mathcal{N}|\mathcal{V}, p, x^1)$ containing the teacher, $w_\mathcal{V} = 0$, with probability mass $(1-p_\mathcal{V})^p$ and midpoint w. The calculation of the reduction in average generalization error contributed by this sector proceeds similarly to that outlined above for the sectors in the first class, apart from the fact that the modulus in $\epsilon_g(\mathcal{N}, \mathcal{V}) = |w_\mathcal{N}|$ has to be taken into account explicitly. One obtains

$$\Delta\epsilon_g(\text{sector containing teacher}) = (1-p_\mathcal{V})^p(1-2p_\mathcal{V})2^{-p-1}(1-4\delta^2). \tag{5.40}$$

The first three factors are in complete analogy with (5.39). The quantity δ measures the distance of the teacher from the closest boundary of the sector, as a fraction of the sector length 2^{-p+1}, and is given in terms of the midpoint w of the sector by $\delta = (2^{-p} - |w|)/2^{-p+1}$. Hence, the factor $1 - 4\delta^2$ in (5.40) equals one when the teacher is on one of the sector boundaries ($\delta = 0$, $|w| = 2^{-p}$) and zero when the teacher is in

the middle of the sector ($\delta = 1/2$, $w = 0$). This makes intuitive sense since in the first case, the sector could actually have been viewed as a sector of the first class; while in the second case, the generalization error is only affected by the sum of the probability masses of the two halves of the sector, which remains unchanged as p is increased by one. As explained above, the selection of bisection queries is a deterministic process for $p \geq 1$; the position of the sectors of $P(\mathcal{N}|\mathcal{V}, p, x^1)$ and hence the value of δ therefore only depends on the first training input x^1, which we have kept fixed so far. Since x^1 is randomly and uniformly chosen from the interval $[-1, 1]$, δ assumes all values in the interval $[0, \frac{1}{2}]$ with equal probability. Averaging the reduction in generalization (5.40) over δ, the factor $1 - 4\delta^2$ is therefore replaced by $2/3$.

The same line of reasoning as for the sector containing the teacher $w_\nu = 0$ can also be applied to the sector including its opposite $-1 = 1 \pmod 2$, and the result is exactly analogous, with the appropriate replacement for the probability mass $(1 - p_\nu)^p \to p_\nu^p$. Collecting everything, we therefore obtain for the reduction in average generalization error

$$\begin{aligned} \epsilon_g(p) - \epsilon_g(p+1) &= (1 - 2p_\nu)2^{-p-1}\left[1 - (1-p_\nu)^p - p_\nu^p + \frac{2}{3}(1-p_\nu)^p + \frac{2}{3}p_\nu^p\right] \\ &= (1 - 2p_\nu)2^{-p-1}\left[1 - \frac{1}{3}(1-p_\nu)^p - \frac{1}{3}p_\nu^p\right]. \end{aligned} \tag{5.41}$$

The derivation of this expression relies on the condition $p \geq 1$, since we have assumed that there are two sectors in the second class, which is not true for $p = 0$ where there is only one single sector. However, eq. (5.41) does actually hold for $p = 0$ as well, as can be checked by an explicit calculation. Starting from $\epsilon_g(p = 0) = \frac{1}{2}$, one obtains the desired quantity $\epsilon_g(p)$ by summing (5.41) over p, yielding a geometric series. The result (5.24) given in the text is obtained by separating the p-dependent parts from the constant terms, using that for $N = 2$, $\alpha = p/(N-1) = p$.

5.8 Appendix: Replica calculation for the binary perceptron, $N \to \infty$

5.8.1 Calculation of free energy

In this Appendix, we sketch the replica calculation of the free energy

$$-\beta f = \frac{1}{N}\langle \ln Z \rangle_{P(\Theta^{(p)}|\mathcal{V})} \qquad Z = \int d\mathcal{N}\tilde{P}(\mathcal{N}) \exp(-\beta E_t(\mathcal{N}, \Theta^{(p)})) \tag{5.42}$$

for binary perceptron students in the thermodynamic limit $N \to \infty$. We consider queries for minimum entropy, selected assuming the inference model is correct, i.e., assuming sign-flip noise on the training outputs with probability $p_{\mathcal{N}}$ (related to β by (5.17)). The teacher is a binary perceptron, with sign-reversal noise $p_{\mathcal{V}}$ and weight noise γ (see eq. (5.25)).

We use the replica trick

$$\langle \ln Z \rangle_{P(\Theta^{(p)}|\mathcal{V})} = \lim_{n\to 0} \frac{1}{n} \ln \langle Z^n \rangle_{P(\Theta^{(p)}|\mathcal{V})}$$

to average the log partition function over training sets, calculating the right hand side for integer values of n and continuing analytically to $n = 0$. By introducing n replicas of the student space and using the explicit expression for the training error (5.15), the n-th moment of the partition function is expressed as

$$Z^n = \int \prod_{a=1}^{n} \left(d\mathbf{w}_{\mathcal{N}}^a \tilde{P}(\mathbf{w}_{\mathcal{N}}^a) \right) \prod_{\mu=1}^{p} \prod_{a=1}^{n} \exp\left[-\beta\Theta\left(y^\mu \mathrm{sgn}\left(\frac{1}{\sqrt{N}} (\mathbf{w}_{\mathcal{N}}^a)^T \mathbf{x}^\mu \right) \right) \right]$$

where $\tilde{P}(\mathbf{w}_{\mathcal{N}})$ is the assumed uniform prior on the hypersphere $\mathbf{w}_{\mathcal{N}}^2 = N$, $\tilde{P}(\mathbf{w}_{\mathcal{N}}) \propto \delta(\mathbf{w}_{\mathcal{N}}^2 - N)$. Using the fact that the training outputs y^μ are binary, the n-fold product of exponentials for a given training example index μ (which we drop for now) can be decomposed as [GT90]

$$\begin{aligned} &\prod_{a=1}^{n} \exp\left[-\beta\Theta\left(y \ \mathrm{sgn}\left(\frac{1}{\sqrt{N}} (\mathbf{w}_{\mathcal{N}}^a)^T \mathbf{x} \right) \right) \right] \\ &= \Theta(y) \prod_{a=1}^{n} \phi\left(\frac{1}{\sqrt{N}} (\mathbf{w}_{\mathcal{N}}^a)^T \mathbf{x} \right) + \Theta(-y) \prod_{a=1}^{n} \phi\left(-\frac{1}{\sqrt{N}} (\mathbf{w}_{\mathcal{N}}^a)^T \mathbf{x} \right) \end{aligned} \tag{5.43}$$

where

$$\phi(z) = \Theta(z) + e^{-\beta}\Theta(-z).$$

From this representation, one sees that sign-flip noise $y \to -y$ merely interchanges the roles of $\phi(z)$ and $\phi(-z)$, and hence of the factors 1 and $e^{-\beta}$ in the definition of $\phi(\cdot)$. This translates directly into the corresponding terms in the final result for the free energy, eq. (5.27), and we therefore ignore sign-flip noise in the following and restrict attention to the case of weight noise. The training output y can then be written as $\mathrm{sgn}((\mathbf{w}_{\mathcal{V}} + \Delta\mathbf{w}_{\mathcal{V}})^{\mathrm{T}}\mathbf{x}/\sqrt{N})$, with $\Delta\mathbf{w}_{\mathcal{V}}$ the Gaussian perturbation of the teacher weight

vector. Introducing the usual Gardner integral representations of the form

$$1 = \int dh^a \; \delta\left(h^a - \frac{1}{\sqrt{N}}(\mathbf{w}_{\mathcal{N}}^a)^{\mathrm{T}}\mathbf{x}\right) = \int \frac{dh^a d\hat{h}^a}{2\pi} \exp\left(i\hat{h}^a h^a - \frac{i}{\sqrt{N}}\hat{h}^a(\mathbf{w}_{\mathcal{N}}^a)^{\mathrm{T}}\mathbf{x}\right)$$

in order to shift all terms containing the input $\mathbf{x}$ into an exponential, one can then rewrite (5.43) in the form

$$\int_0^\infty dh \int \frac{d\hat{h}}{2\pi} \prod_a \left(\int \frac{dh^a d\hat{h}^a}{2\pi}\phi(h^a)\right) \exp\left(i\hat{h}h + i\sum_a \hat{h}^a h^a\right)$$
$$\times \left[\exp\left(\frac{i}{\sqrt{N}}\hat{h}(\mathbf{w}_\nu + \Delta\mathbf{w}_\nu)^{\mathrm{T}}\mathbf{x} + \frac{i}{\sqrt{N}}\sum_a \hat{h}^a(\mathbf{w}_{\mathcal{N}}^a)^{\mathrm{T}}\mathbf{x}\right) + \text{c.c.}\right]. \qquad (5.44)$$

Here we have used the same symbols h and $\hat{h}$ for integration variables relating to the student replicas and the teacher, distinguished only by the additional replica index superscript for the students. This emphasizes the fact that all the h's denote products of weight vectors with the input vector, while the $\hat{h}$'s are their conjugate variables.

The average over the Gaussian weight noise is now trivial,

$$\left\langle \exp\left(\frac{i}{\sqrt{N}}\hat{h}\Delta\mathbf{w}_\nu^{\mathrm{T}}\mathbf{x}\right)\right\rangle_{\Delta\mathbf{w}_\nu} = \exp\left(-\frac{1}{2}\sigma^2\hat{h}^2\right) \qquad (5.45)$$

due to the spherical constraint on inputs, $\mathbf{x}^2 = N$. The main task consists in calculating the average of

$$\exp\left(\frac{i}{\sqrt{N}}\hat{h}\mathbf{w}_\nu^{\mathrm{T}}\mathbf{x} + \frac{i}{\sqrt{N}}\sum_a \hat{h}^a(\mathbf{w}_{\mathcal{N}}^a)^{\mathrm{T}}\mathbf{x}\right) \qquad (5.46)$$

over the distribution for bisection queries.

We do this by exploiting the query by committee approach described in the text, following Ref. [SOS92]. For committee size $2k$, the distribution of the $(\mu+1)$-th query $\mathbf{x}^{\mu+1}$ selected on the basis of a training set of μ examples, $\Theta^{(\mu)}$, is

$$P(\mathbf{x}^{\mu+1}|\Theta^{(\mu)}) \propto P(\mathbf{x}^{\mu+1}) \sum_{\{\sigma_\gamma\}} \prod_{\gamma=1}^{2k} \Theta\left(\sigma_\gamma \frac{1}{\sqrt{N}}(\mathbf{w}_{\mathcal{N}}^\gamma)^{\mathrm{T}}\mathbf{x}^{\mu+1}\right) \qquad (5.47)$$

where the $\mathbf{w}_{\mathcal{N}}^\gamma$ are the weight vectors of the $2k$ committee members $\mathcal{N}_\gamma$ randomly sampled from $P(\mathcal{N}|\Theta^{(\mu)})$, and the $\sigma_\gamma = \pm 1$ are their outputs[14]. According to the principle of maximal disagreement between committee members, the summation is

[14] In principle, an average over the $\mathbf{w}_{\mathcal{N}}^\gamma$ has to be taken on the right hand side of (5.47), but this is unnecessary in the thermodynamic limit due to the self-averaging of the overlaps (5.51,5.52).

over all combinations of the σ_γ for which exactly k of the σ_γ are $+1$ and the others -1. The distribution $P(\mathbf{x})$, finally, implements the spherical constraint on input vectors, $P(\mathbf{x}) \propto \delta(\mathbf{x}^2 - N)$. To calculate the average of (5.46), we have to multiply it by the right hand side of (5.47), integrate over $\mathbf{x} \equiv \mathbf{x}^{\mu+1}$, and divide by the corresponding normalization factor. Denoting the scalar products of $\mathbf{x}^{\mu+1}$ with the various weight vectors by[15]

$$z = \frac{1}{\sqrt{N}}\mathbf{w}_\nu^{\mathrm{T}}\mathbf{x}^{\mu+1} \qquad z^a = \frac{1}{\sqrt{N}}(\mathbf{w}_\mathcal{N}^a)^{\mathrm{T}}\mathbf{x}^{\mu+1} \qquad z^\gamma = \frac{1}{\sqrt{N}}(\mathbf{w}_\mathcal{N}^\gamma)^{\mathrm{T}}\mathbf{x}^{\mu+1}$$

the desired average of (5.46) can therefore be written as a ratio of two averages over z, $\{z^a\}$ and $\{z^\gamma\}$:

$$\left\langle \exp\left(\frac{i}{\sqrt{N}}\hat{h}\mathbf{w}_\nu^{\mathrm{T}}\mathbf{x}^{\mu+1} + \frac{i}{\sqrt{N}}\sum_a \hat{h}^a(\mathbf{w}_\mathcal{N}^a)^{\mathrm{T}}\mathbf{x}^{\mu+1}\right)\right\rangle_{P(\mathbf{x}^{\mu+1}|\Theta(\mu))} =$$

$$\frac{\left\langle \sum_{\{\sigma_\gamma\}} \prod \Theta(\sigma_\gamma z^\gamma) \exp(i\hat{h}z + i\sum_a \hat{h}^a z^a)\right\rangle}{\left\langle \sum_{\{\sigma_\gamma\}} \prod \Theta(\sigma_\gamma z^\gamma)\right\rangle} . \tag{5.48}$$

In the thermodynamic limit, z, $\{z^a\}$ and $\{z^\gamma\}$ become zero mean unit variance Gaussian variables with correlations

$$\langle z z^a\rangle = \frac{1}{N}\mathbf{w}_\nu^{\mathrm{T}}\mathbf{w}_\mathcal{N}^a = R$$

$$\langle z z^\gamma\rangle = \frac{1}{N}\mathbf{w}_\nu^{\mathrm{T}}\mathbf{w}_\mathcal{N}^\gamma = R^\mu \tag{5.49}$$

$$\left\langle z^a z^b\right\rangle = \frac{1}{N}(\mathbf{w}_\mathcal{N}^a)^{\mathrm{T}}\mathbf{w}_\mathcal{N}^b = q \tag{5.50}$$

$$\left\langle z^\gamma z^\delta\right\rangle = \frac{1}{N}(\mathbf{w}_\mathcal{N}^\gamma)^{\mathrm{T}}\mathbf{w}_\mathcal{N}^\delta = q^\mu \tag{5.51}$$

$$\langle z^a z^\gamma\rangle = \frac{1}{N}(\mathbf{w}_\mathcal{N}^a)^{\mathrm{T}}\mathbf{w}_\mathcal{N}^\gamma = q^{\mu p} \tag{5.52}$$

where we have used the self-averaging property of the overlaps between the various weight vectors to replace them by the corresponding order parameters[16]. The independence of the overlaps q and $q^{\mu p}$ in (5.50, 5.52) of the replica indices a, b embodies the assumption of replica symmetry. Since the committee members can be viewed as

[15] Again, we use the same symbol z for scalar products with the weight vectors of the teacher, the student replicas and the committee members, in order to emphasize functional similarities between them. Greek superscripts always refer to committee members in the following, while Roman superscripts denote student replicas.

[16] It is at this point that we have to assume that the committee size k is much smaller than the system size N, since otherwise the overlaps of the committee members could not be assumed to be self-averaging any more. For $k = O(N)$, for example, one cannot expect all the committee members to have identical overlaps (5.51,5.52).

student replicas sampled from the distribution $P(\mathcal{N}|\Theta^{(\mu)})$ (rather than $P(\mathcal{N}|\Theta^{(p)})$ as for the 'real' replicas), replica symmetry also implies the independence of the overlaps (5.51, 5.52) of the committee indices γ and δ. We have not explicitly analysed the stability of the replica symmetric solution against replica symmetry breaking (see, e.g., [MPV87]). However, for the case of perceptron students with continuous – as opposed to discrete – weights that we are dealing with, replica symmetry breaking is normally signalled by the order parameter q approaching one at some finite value of α (see, e.g., [GT90, MEZ93]), for which we have not found any evidence.

At this point, we would like to point out an important feature of eqs. (5.49, 5.51, 5.52): The average of (5.46) over the distribution $P(\mathbf{x}^{\mu+1}|\Theta^{(\mu)})$ of the $(\mu+1)$-th training input has now effectively become independent of the particular realization of the existing training set $\Theta^{(\mu)}$, being dependent solely on self-averaging overlap parameters. This justifies *a posteriori* our procedure of carrying out the averages over the different training inputs separately, which is crucial in making the problem analytically tractable. This a good demonstration of the power of the statistical mechanics approach, which allows one to treat complex correlations between successive training examples such as the ones generated by query learning.

To carry out the averages in (5.48), we represent z, $\{z^a\}$ and $\{z^\gamma\}$ in terms of uncorrelated zero mean unit variance Gaussian variables $\tilde{x}$, $\tilde{y}$, $\tilde{z}$, $\{\tilde{z}^a\}$ and $\{\tilde{z}^\gamma\}$:

$$\begin{aligned} z^\gamma &= \tilde{x}\sqrt{q^\mu} + \tilde{z}^\gamma\sqrt{1-q^\mu} \\ z^a &= \tilde{x}q^{\mu p}/\sqrt{q^\mu} + \tilde{y}c_1 + \tilde{z}^a c_2 \\ z &= \tilde{x}R^\mu/\sqrt{q^\mu} + \tilde{y}c_2 + \tilde{z}c_4 \end{aligned}$$

where

$$c_1 = (q-(q^{\mu p})^2/q^\mu)^{1/2} \qquad c_2 = \sqrt{1-q} \tag{5.53}$$

$$c_3 = c_1^{-1}(R - R^\mu q^{\mu p}/q^\mu) \qquad c_4 = (1-c_3^2-(R^\mu)^2/q^\mu)^{1/2}. \tag{5.54}$$

Averaging over the $\{\tilde{z}^\gamma\}$, one obtains

$$\left\langle \sum_{\{\sigma_\gamma\}} \prod_{l=1}^{2k} \Theta(\sigma_\gamma z^\gamma) \right\rangle_{\{\tilde{z}^\gamma\}} = \binom{2k}{k} H^k\left(\tilde{x}\sqrt{\frac{q^\mu}{(1-q^\mu)}}\right) H^k\left(-\tilde{x}\sqrt{\frac{q^\mu}{(1-q^\mu)}}\right) \tag{5.55}$$

where $H(z) = \int_z^\infty Dx$, with Dx the Gaussian measure $Dx = \exp(-\frac{1}{2}x^2)\,dx/\sqrt{2\pi}$. In the limit $k \to \infty$ of interest to us, the factor (5.55) in the numerator and denominator of (5.48) constrains $\tilde{x}$ to values arbitrarily close to zero, effectively approaching a delta

distribution with respect to $\tilde{x}$. Having set $\tilde{x}$ to zero, the remaining averages in (5.48) can easily be carried out. Using the values of the constants $c_1 \ldots c_4$ given in (5.53, 5.54), one obtains

$$\exp\left[-\frac{1}{2}\left(1-\frac{(R^\mu)^2}{q^\mu}\right)\hat{h}^2-\left(R-\frac{R^\mu q^{\mu p}}{q^\mu}\right)\hat{h}\sum_a \hat{h}^a \right. \\ \left. -\frac{1}{2}\left(q-\frac{(q^{\mu p})^2}{q^\mu}\right)\left(\sum_a \hat{h}^a\right)^2-\frac{1}{2}(1-q)\sum_a(\hat{h}^a)^2\right]. \tag{5.56}$$

This result together with the weight noise average can now be plugged into (5.44). We omit details of the manipulations leading from this stage of the calculation to the result (5.27), as they follow exactly the calculation for random training examples described in Ref. [GT90].

We note at this point that the above result (5.56) for query by committee in the $k \to \infty$ limit also results from a much simpler algorithm for query selection. Suppose each query $\mathbf{x}^{\mu+1}$ is selected to be orthogonal to the average weight vector $\overline{\mathbf{w}}_{\mathcal{N}}^{\mu}$ from the post-training student distribution $P(\mathcal{N}|\Theta^{(\mu)})$ generated by the existing training set $\Theta^{(\mu)}$, but otherwise random, i.e., according to $P(\mathbf{x})$. Then the average over $\mathbf{x}^{\mu+1}$ of a quantity of the form (5.46) is, in the thermodynamic limit,

$$\left\langle \exp\left(\frac{i}{\sqrt{N}}\mathbf{a}^{\mathrm{T}}\mathbf{x}^{\mu+1}\right)\right\rangle_{P(\mathbf{x}^{\mu+1}|\Theta^{(\mu)})} = \exp\left[-\frac{1}{2}\left(\frac{1}{N}\mathbf{a}^2-\frac{\left(\frac{1}{N}\mathbf{a}^{\mathrm{T}}\overline{\mathbf{w}}_{\mathcal{N}}^{\mu}\right)^2}{\frac{1}{N}(\overline{\mathbf{w}}_{\mathcal{N}}^{\mu})^2}\right)\right].$$

Inserting $\mathbf{a} = \hat{h}\mathbf{w}_{\mathcal{V}} + \sum_a \hat{h}^a \mathbf{w}_{\mathcal{N}}^a$, it can easily be verified from the definitions (5.22, 5.28, 5.31) that this expression is identical to the result (5.56). This implies that it is possible to *construct* queries which, in the thermodynamic limit, achieve the same effect as queries *filtered* by an infinitely large committee from a stream of random inputs. This is particularly interesting since it has been shown [FSST93] that the time for query filtering increases exponentially with α, due to the fact that the student distribution becomes more and more narrow as α increases, and is bisected only by a small fraction of all possible input vectors. In practical terms, having a constructive query algorithm at one's disposal which avoids this problem but achieves the same performance is therefore a clear advantage. We shall return to this point in our discussion of query learning in multi-layer neural networks in Chapter 8.

5.8.2 Asymptotic solution

Let us comment briefly on the large α behaviour of the solution of the saddle point equations derived from the free energy (5.27), for the case of a correctly specified inference model, where the correct sign-flip probability is assumed, $p_{\mathcal{N}} = p_{\mathcal{V}}$, and there is no weight noise, $\gamma = 1$. As shown in Section 5.5.2, one then has $q^{\mu p} = q^{\mu}$. Furthermore, $q = R$ in this case due to the symmetry between the posterior teacher distribution and the post-training student distribution (see, e.g., [SOS92, WRB93]); this can also be confirmed directly from the numerical solution of the saddle point equations derived from (5.27).

The numerical results show an exponential decay of the generalization error ϵ_g with α (see figure 5.10). To calculate the corresponding decay constant, one can proceed as follows. First, we note that for bisection queries in a binary output system, the log partition function $\ln Z$ is always a linear function of the number of training examples. This can be seen by using the definition of Z, eq. (5.42), to rewrite the ratio between two 'successive' partition functions as

$$\begin{aligned} \frac{Z(\Theta^{(p+1)})}{Z(\Theta^{(p)})} &= \int d\mathcal{N}\tilde{P}(\mathcal{N})\left(Z(\Theta^{(p)})\right)^{-1}\exp\left[-\beta E_t(\mathcal{N},\Theta^{(p+1)})\right] \\ &= \int d\mathcal{N}P(\mathcal{N}|\Theta^{(p)})\exp\left[-\beta\Theta\left(-y^{p+1}f_{\mathcal{N}}(\mathbf{x}^{p+1})\right)\right]. \end{aligned}$$

Using the decomposition $\exp(-\beta\Theta(-z)) = \Theta(z)+e^{-\beta}\Theta(-z)$, it follows that for queries $\mathbf{x}^{p+1}$ selected according to the bisection criterion (5.19),

$$\ln Z(\Theta^{(p+1)}) - \ln Z(\Theta^{(p)}) = \ln\left[\frac{1}{2}(1+e^{-\beta})\right] = -\ln[2(1-p_{\mathcal{N}})].$$

With $Z(\Theta^{(0)}) = 1$ from the normalization of the pseudo prior $\tilde{P}(\mathcal{N})$, this gives

$$\ln Z(\Theta^{(p)}) = -p\ln[2(1-p_{\mathcal{N}})]. \tag{5.57}$$

If the inference model is correct, this is in agreement with our replica calculation of the average free energy. To show this, let us first rewrite (5.27) in a simpler form by replacing $(1/N)\sum_{\mu}$ by an integral over α' (as is appropriate in the thermodynamic limit), inserting the relations $q(\alpha',\alpha) = q(\alpha')$ and $q = R$. Using the identity

$$\int_0^{\infty} Dh \int Dt\; f(\rho h+\tau t) = \int Dv\; f\left(v\sqrt{\rho^2+\tau^2}\right) H\left(-v\frac{\rho}{\tau}\right)$$

(which, incidentally, can also be employed to simplify the numerical solution of the

saddle point equations), one obtains

$$-\beta f = \frac{1}{N}\langle \ln Z\rangle = \mathrm{extr}_q \left\{ \frac{1}{2}(q + \ln(1-q)) + \int_0^\alpha d\alpha' F\left(\sqrt{\frac{q-q(\alpha')}{1-q}}\right)\right\} \tag{5.58}$$

with

$$F(x) = 2\int Dv\ H(xv) \times \left[(1-p_{\mathcal{N}})\ln\left(H(xv) + e^{-\beta}(1-H(xv))\right)\right.$$
$$\left. + \ p_{\mathcal{N}} \ln\left(e^{-\beta}H(xv) + 1 - H(xv)\right)\right].$$

This gives for the α derivative of the average log partition function

$$\frac{1}{N}\frac{d\langle \ln Z\rangle}{d\alpha} = \frac{1}{N}\frac{\partial\langle \ln Z\rangle}{\partial\alpha} = F(0) = -\ln[2(1-p_{\mathcal{N}})] \tag{5.59}$$

in agreement with the general result derived above. In (5.59), we have used the fact that the variation of q with α does not contribute to the variation of the log partition function due to the saddle point condition. The fact that the correct behaviour of the log partition function is predicted by the replica calculation supports our assumptions of replica symmetry and interchangeability of the limits $k \to \infty$ and $N \to \infty$.

Let us now use the result $\langle \ln Z\rangle = -N\alpha \ln[2(1-p_{\mathcal{N}})]$ to derive the decay constant in the exponential decay of the average generalization error ϵ_g with α. Using (5.21) and $q = R$, ϵ_g is expressed $\epsilon_g = (1/\pi)\arccos q$. As $\epsilon_g \to 0$, $q \to 1$, and one has $\epsilon_g \propto (\Delta q)^{1/2}$, where $\Delta q = 1 - q$. An asymptotic exponential decay $\epsilon_g \propto \exp(-c\alpha)$ therefore corresponds to an exponential decay of $\Delta q \propto \exp(-2c\alpha)$. Inserting this into (5.58), one derives for the large α behaviour of the log partition function

$$\frac{1}{N}\langle \ln Z\rangle = -\alpha\left[c + p_{\mathcal{N}}\ln\left(\frac{p_{\mathcal{N}}}{1-p_{\mathcal{N}}}\right)\right]$$

up to terms which remain bounded as $\alpha \to \infty$. Comparing this with the result (5.59), one finds the value of the decay constant

$$c = \ln 2 + p_{\mathcal{N}} \ln p_{\mathcal{N}} + (1-p_{\mathcal{N}})\ln(1-p_{\mathcal{N}})$$

given in the text (see eq. (5.36); note that $p_{\mathcal{N}} = p_{\mathcal{V}}$ in the case considered here).

5.8.3 Free energy correlations

We conclude this Appendix by some comments on how the case of an incorrect inference model could be treated analytically. As explained in the text, the main hurdle is

the determination of the $q^{\mu p} \equiv q(\alpha', \alpha)$, which are no longer simply related to the $q^{\mu} \equiv q(\alpha')$. A solution would be to calculate the correlations of the fluctuations of the log partition functions corresponding to training sets of size μ and p ($\mu < p$) around their averages, i.e.,

$$f^{\mu p} = \frac{1}{N}\left(\left\langle \ln Z(\Theta^{(\mu)}) \ln Z(\Theta^{(p)})\right\rangle - \left\langle \ln Z(\Theta^{(\mu)})\right\rangle \left\langle \ln Z(\Theta^{(p)})\right\rangle\right).$$

where the averages are over the training set distribution $P(\Theta^{(p)}|\mathcal{V})$. The logarithms inside the averages can be removed by a double replica trick

$$f^{\mu p} = \frac{1}{N} \frac{\partial^2}{\partial m \partial n} \ln \left\langle Z^m(\Theta^{(\mu)}) Z^n(\Theta^{(p)})\right\rangle_{P(\Theta^{(p)}|\mathcal{V})}\Big|_{m=n=0} \tag{5.60}$$

One now has m replicas of students from $P(\mathcal{N}|\Theta^{(\mu)})$ and n replicas of students from $P(\mathcal{N}|\Theta^{(p)})$, with the overlap of two students from the two replica groups being simply $q^{\mu p}$ from the definition (5.31). Since we are calculating correlations between $\ln Z(\Theta^{(\mu)})$ and $\ln Z(\Theta^{(p)})$, it is natural that $q^{\mu p}$ would appear as a saddle point parameter in the result for $f^{\mu p}$. Indeed, by a calculation similar to the one outlined in Section 5.8.1, one obtains

$$\begin{aligned} f^{\mu p} &= \text{extr}_{\,q^{\mu p}} \left\{ -\frac{1}{2}\frac{(q^{\mu p} - R^{\mu} R)^2}{(1-q^{\mu})(1-q)} \right. \\ &+ \left. \frac{1}{N}\sum_{\nu=0}^{\mu-1}\left[(1-p_\nu)\left(\langle l^{\mu} l\rangle - \langle l^{\mu}\rangle \langle l\rangle\right) + p_\nu\left(\langle \bar{l}^{\mu}\bar{l}\rangle - \langle \bar{l}^{\mu}\rangle \langle \bar{l}\rangle\right)\right]\right\}. \end{aligned} \tag{5.61}$$

where

$$\begin{aligned} l^{\mu} &= \ln\left(H(u^{\mu}) + e^{-\beta}(1 - H(u^{\mu}))\right) \\ l &= \ln\left(H(u) + e^{-\beta}(1 - H(u))\right) \end{aligned}$$

and $\bar{l}^{\mu}$ and $\bar{l}$ are obtained by reversing the roles of the factors $e^{-\beta}$ and 1 in the above definitions. The averages on the right hand side of (5.61) are taken over zero mean Gaussian variables h, t^{μ} which are related to u^{μ} and u by

$$u^{\mu} = \rho^{\mu}|h| + t^{\mu} \qquad u = \rho|h| + t.$$

The (co-)variances of t^{μ} and t are

$$\left\langle (t^{\mu})^2\right\rangle = (\tau^{\mu})^2 \qquad \left\langle t^2\right\rangle = \tau^2 \qquad \langle t^{\mu} t\rangle = \tau^{\mu p},$$

while h is uncorrelated with t^μ and t and has unit variance. The coefficient ρ is defined as in (5.29, 5.30) apart from the replacement of the subscript μ by ν everywhere, and ρ^μ is the corresponding quantity for μ training examples (i.e., with q and R replaced by q^μ and R^μ). Equivalent definitions hold for τ and τ^μ. As expected, it is only in the correlation coefficient

$$\tau^{\mu p} = \frac{q^{\mu p} - q^{\nu\mu} q^{\nu p}/q^\nu}{\sqrt{1-q^\mu}\sqrt{1-q}} - \rho^\mu \rho$$

that the overlap $q^{\mu p}$ appears. Note that the 'standard' overlap parameters q, R and q^μ, R^μ are not determined by extremizing (5.61), but rather the usual 'single replica' free energies (5.27) for p and μ training examples, respectively. This is because the single replica free energies appear as $O(m)$ and $O(n)$ contributions in $\ln \left\langle Z^m(\Theta^{(\mu)}) Z^n(\Theta^{(p)}) \right\rangle$ and, being much larger in the limit $m \to 0, n \to 0$ than the $O(mn)$ contributions which according to (5.60) determine $f^{\mu p}$, fix the limiting values of the single replica order parameters.

In principle, the $q^{\mu p}$ can be determined from the saddle point equation generated by (5.61). This is, however, a very computationally demanding task: The introduction of an additional auxiliary integration variable t^μ makes the explicit evaluation of the $q^{\mu p}$ derivative of (5.61) expensive; furthermore, for a given value of $\alpha = p/N$, $q^{\mu p} \equiv q(\alpha', \alpha)$ has to be determined for all $\alpha' \in [0, \alpha]$, with a new saddle point equation having to be solved for each value of α'. For small α, this can be circumvented by Taylor expanding all overlap parameters in α' and α and calculating the coefficients successively. This, however, reveals a more profound problem arising from (5.61): We find that the average log partition function is no longer linear in α, but contains a contribution of order α^3, whose coefficient *only* vanishes when the inference model is correct, i.e., $p_\mathcal{N} = p_\mathcal{V}$ and $\gamma = 1$. This contradicts the general result (5.57) derived above, which holds independently of whether the inference model is correct or not. At present, we can only speculate on the reasons for this disagreement: While it is in principle possible that replica symmetry breaking might play a role, this appears unlikely. A more probable cause is the exchange of the limits $k \to \infty$ and $N \to \infty$: it may be that the query by committee algorithm does not generate bisection queries in the limit $k \to \infty$ as long as k remains much smaller than the system size N. Monte Carlo simulations for a range of values of k and N should be able to shed some light on this question; we leave this as a topic for future research.

Notwithstanding the above problems associated with the double replica approach to determining the $q^{\mu p}$, we shall use eq. (5.61) to analyse the asymptotic behaviour of the generalization error achieved by (minimum entropy) query learning in the so-called high temperature limit (see, e.g., [SST92]). This limit is defined by $\beta \to 0$ at constant

$\tilde{\alpha} = \beta\alpha$ and can often give qualitatively, though not quantitatively, correct predictions for the behaviour at large but finite temperatures $1/\beta$. The saddle point equation for $q^{\mu p} \equiv q(\tilde{\alpha}', \tilde{\alpha})$ resulting from (5.61) simplifies considerably in the high-temperature limit and can be solved explicitly to give

$$q(\tilde{\alpha}', \tilde{\alpha}) = R(\tilde{\alpha}')R(\tilde{\alpha}) + O(\beta). \tag{5.62}$$

This relation has the following intuitive interpretation: One can think of learning as a directed random walk of the average weight vector $\overline{\mathbf{w}}_N^p$ from the student distribution $P(\mathcal{N}|\Theta^{(p)})$, with the number of training examples $\alpha = p/N$ acting as a kind of time coordinate. The overlap parameter R captures the directed component of this random walk along the direction of the teacher $\mathbf{w}_\nu$, while $q^{\mu p}$ determines the correlation between random walkers at times $\alpha' = \mu/N$ and $\alpha = p/N$. Eq. (5.62) then expresses the fact that in the high temperature limit, where the separation between the times $\alpha' = \tilde{\alpha}'/\beta$ and $\alpha = \tilde{\alpha}/\beta$ diverges as $1/\beta$, the correlations between the non-directed components (orthogonal to $\mathbf{w}_\nu$) of the random walk decay to zero.

Inserting the relation (5.62) into the high temperature form of the saddle point equation for R derived from the single replica free energy (5.27), one obtains in the case of weight noise on the training outputs ($\gamma < 1$, $p_\nu = 0$)

$$\frac{R}{1-R^2} = \frac{2}{\pi}\tilde{\alpha}\left[(1-R^2)(1/\gamma^2 - 1)\right]^{-1/2} + O(\beta).$$

In the high-temperature limit, the $O(\beta)$ term can be neglected, and one finds

$$R = \frac{c\tilde{\alpha}}{\sqrt{1+c^2\tilde{\alpha}^2}} \qquad c = \frac{2}{\pi}\frac{\gamma}{\sqrt{1-\gamma^2}}.$$

For large $\tilde{\alpha}$, this yields the power-law decay for the generalization error

$$\epsilon_g = \frac{1}{\pi}\arccos R \propto (1-R)^{1/2} \propto 1/\tilde{\alpha}$$

referred to in Section 5.5.2. The corresponding result for random examples can be derived similarly: The form of the free energy (5.27) remains unchanged up to the replacement of $\tilde{q}$ and $\tilde{R}$ by q and R [GT90]; in the high temperature limit, one obtains the saddle point equation

$$\frac{R}{1-R^2} = \frac{2}{\pi}\tilde{\alpha}\left(1/\gamma^2 - R^2\right)^{-1/2}$$

which asymptotically gives $1 - R \propto 1/\tilde{\alpha}$ and $\epsilon_g \propto \tilde{\alpha}^{-1/2}$.

Chapter 6

Combining query learning and model selection

Abstract

In the previous chapter, we exposed several problems of query learning assuming the inference model is correct. As a potential solution, we propose to combine query learning with inference model selection or adaptation. We first outline an appropriate theoretical framework and then analyse the consequences of combining query learning with a particular model selection technique, called the evidence procedure. The results that we find for the two scenarios considered in the previous chapter (linear and binary perceptron students) are encouraging: The problem of self-confirming hypotheses is avoided, and the resulting generalization performance is consistently and significantly better than for learning from random examples.

6.1 Introduction

We have seen in the previous chapter that query learning assuming the inference model is correct has several drawbacks when the assumed inference model is actually *incorrect*. For the linear perceptron, we saw that queries are no longer guaranteed to yield a lower generalization error than random examples, even if they are explicitly selected to minimize the generalization error. More seriously, for binary output students we found the problem of self-confirming hypotheses far from the truth, where query learning prevents the student from approximating the teacher optimally even after having been presented with an infinite number of training examples. These problems obviously need to be circumvented if we are to trust what we have learned from queries. Since they

stem from the fact that the inference model is incorrect, it is natural to expect that they will at least be alleviated by considering more than one inference model and selecting the most appropriate one in the light of the training data. We will explore this idea in the present chapter, focusing on learning with linear and binary perceptrons as in the previous chapter.

We begin in Section 6.2 by outlining a theoretical framework for combining query learning and inference model selection. In Sections 6.3 and 6.4, we then apply this framework by investigating the consequences of a particular model selection technique, called the evidence procedure, for query learning with linear and binary perceptron students. The different inference models between which we choose in these cases are parameterized by the 'hyperparameters' (see, e.g., [Ber85]) weight decay $\tilde{\lambda}$ and temperature parameter β for the linear perceptron, and assumed sign-reversal probability $p_{\mathcal{N}}$ for the binary perceptron[1]. We are therefore effectively considering a continuum of inference models, and the term 'inference model adaptation' might be more appropriate than 'model selection' in this context, in particular if, as considered below, the hyperparameters are adapted continuously as new training examples are received. Section 6.5 offers a brief summary of our results and some perspectives for future work.

6.2 Theoretical framework

Let us outline how inference model adaptation or selection can be incorporated into the framework for query learning assuming the inference model is correct as set out in Section 5.2. Assume that the inference models between which we can choose are parameterized by a certain hyperparameter. This hyperparameter can be a continuous parameter such as, for example, the assumed noise level $p_{\mathcal{N}}$ for the binary perceptron, it may be discrete, e.g., simply numbering different inference models, or it may in fact be a collection of several hyperparameters. In terms of notation, we shall not distinguish between these three cases, and simply use the generic symbol λ for the hyperparameter(s). In general, the objective function for query selection will depend on λ; for the binary perceptron, for example, the student space entropy is a function of the hyperparameter $p_{\mathcal{N}}$. Furthermore, the value of λ can affect both the assumed student prior $\tilde{P}(\mathcal{N})$ and the assumed probabilistic input-output relation $\tilde{P}(y|x,\mathcal{N})$. We therefore make the replacements

$$\tilde{P}(\mathcal{N}) \to \tilde{P}(\mathcal{N}|\lambda) \qquad \tilde{P}(y|x,\mathcal{N}) \to \tilde{P}(y|x,\mathcal{N},\lambda)$$

[1] Unless otherwise specified, the notation used in this chapter is the same as in Chapter 5.

and correspondingly for the post-training distribution defined by (5.4)

$$P(\mathcal{N}|\Theta^{(p)}) \to P(\mathcal{N}|\Theta^{(p)},\lambda) \propto \tilde{P}(\mathcal{N}|\lambda)\prod_{\mu=1}^{p}\tilde{P}(y^{\mu}|x^{\mu},\mathcal{N},\lambda). \tag{6.1}$$

We now want to determine the plausibility of the different inference models, given a training set $\Theta^{(p)}$. This can be done in a variety of ways, in the same way as there are (if we do not restrict ourselves to a posterior-like form) many different possible choices for the post-training distribution $P(\mathcal{N}|\Theta^{(p)})$ determining how we infer the plausibility of different students $\mathcal{N}$ from the training data. However, as noted in the previous chapter, our framework for query learning assuming the inference model is correct has essentially brought us back to a 'traditional' Bayesian framework, and it is therefore most natural to base our choice between different inference models on the 'posterior' distribution of λ:

$$P(\lambda|\Theta^{(p)}) \propto \tilde{P}(\Theta^{(p)}|\lambda)\tilde{P}(\lambda) = \tilde{P}(\lambda)\prod_{\mu=1}^{p}\tilde{P}(y^{\mu}|x^{\mu},\Theta^{(\mu-1)},\lambda)P(x^{\mu}|\Theta^{(\mu-1)},\lambda). \tag{6.2}$$

Here $\tilde{P}(\lambda)$ is an assumed prior on the hyperparameters, which will normally chosen to be 'flat', i.e., uninformative, if we have no specific preference for any of the inference models. In Chapter 2, we argued that query selection cannot depend on the unknown true teacher $\mathcal{V}$. Similarly, in the present context the probability of selecting query x^{μ} should depend only on the existing training set $\Theta^{(\mu-1)}$, not on any particular 'true' value of λ. In fact, it can be shown that without this assumption, nonsensical results would be obtained[2]. We can therefore write

$$P(x^{\mu}|\Theta^{(\mu-1)},\lambda) = P(x^{\mu}|\Theta^{(\mu-1)}). \tag{6.3}$$

We now exploit the analogue of (5.5),

$$\tilde{P}(y^{\mu}|x^{\mu},\Theta^{(\mu-1)},\lambda) = \int d\mathcal{N}\,\tilde{P}(y^{\mu}|x^{\mu},\mathcal{N},\lambda)\,P(\mathcal{N}|\Theta^{(\mu-1)},\lambda) \tag{6.4}$$

[2]Consider, for example, the binary perceptron with $N = 2$ (see Section 5.5.1). It is a simple matter to convince oneself that the selection of the first three training inputs by minimum entropy queries is independent of the assumed noise level $p_{\mathcal{N}}$, whereas the position in input space of the fourth query (with respect to the first three inputs), *if based on a particular value of* $p_{\mathcal{N}}$, is in one-to-one correspondence with the value of $p_{\mathcal{N}}$. This means that the probability $P(x^4|\Theta^{(3)},p_{\mathcal{N}})$ would be nonzero only for this particular value of $p_{\mathcal{N}}$. The same would then hold for the probability distribution $P(p_{\mathcal{N}}|\Theta^{(p)})$, which contains $P(x^4|\Theta^{(3)},p_{\mathcal{N}})$ as a factor for $p \geq 4$ (see eq. (6.2)), so that further adaptation of the value of $p_{\mathcal{N}}$ would be impossible.

which using (6.1) can be rewritten as

$$\tilde{P}(y^\mu|x^\mu,\Theta^{(\mu-1)},\lambda)=\frac{\int d\mathcal{N}\ \tilde{P}(\mathcal{N}|\lambda)\prod_{\nu=1}^{\mu}\tilde{P}(y^\nu|x^\nu,\mathcal{N},\lambda)}{\int d\mathcal{N}\ \tilde{P}(\mathcal{N}|\lambda)\prod_{\nu=1}^{\mu-1}\tilde{P}(y^\nu|x^\nu,\mathcal{N},\lambda)}.$$

Combining this with (6.3), it follows that the λ posterior (6.2) is given by $P(\lambda|\Theta^{(p)})\propto\tilde{P}(\lambda)\tilde{P}(\Theta^{(p)}|\lambda)$ with

$$\tilde{P}(\Theta^{(p)}|\lambda)\propto\int d\mathcal{N}\,\tilde{P}(\mathcal{N}|\lambda)\prod_{\mu=1}^{p}\tilde{P}(y^\mu|x^\mu,\mathcal{N},\lambda) \tag{6.5}$$

where the proportionality constants are independent of λ.

The choice of λ (and hence an inference model) affects both the post-training student distribution (6.1), i.e., the way we make inferences about students on the basis of the training data, and the query selection process, through a possible λ dependence of the objective function for query selection and through the assumed distribution of new training outputs, eq. (6.4). One possible way of using the posterior distribution of λ, $P(\lambda|\Theta^{(p)})$, as derived above, would be to average all quantities dependent on λ over this distribution, analogous to the averaging over the posterior teacher distribution $P(\mathcal{V}|\Theta^{(p)})$ in the original framework for query selection of Chapter 2 (see eq. (2.7)). However, the computational cost of this method, which corresponds to 'hierarchical Bayesian inference' (see, e.g., [Ber85]), would appear to be prohibitively high in all but the simplest scenarios. We therefore consider a somewhat simpler approach, where instead of *averaging* over λ, we simply fix its value to maximize the so-called *evidence* $\tilde{P}(\Theta^{(p)}|\lambda)$, eq. (6.5); this adaptation of λ is repeated every time a new training example is added to the training set. In the context of learning from random examples, it has previously been argued that this 'evidence procedure' can constitute a good approximation to the hierarchical Bayes scheme[3] [Mac92a, Mac92d, Gul89], although claims to the contrary have also been advanced [Wol93, WSW93, WS95]. Here, we will disregard this question and simply explore the consequences of the evidence procedure for inference model selection when it is combined with query learning.

[3] Note that in the case where the hyperparameters influence only the (pseudo-) prior, the evidence procedure corresponds to the well-known statistical technique of 'type II maximum likelihood' (ML-II) (see, e.g., [Ber85]).

6.3 Linear perceptron

We now investigate the effects of combining query learning with the evidence procedure for the case of linear perceptron students, continuing the discussion in Section 5.3. The inference models that we consider are parameterized by the weight decay $\tilde{\lambda}$ and the temperature parameter β which appear in the Gibbs post-training distribution of students (5.6),

$$P(\mathcal{N}|\Theta^{(p)}) \propto \exp\left[-\beta\left(\sum_{\mu=1}^{p}\frac{1}{2}(y^\mu - f_{\mathcal{N}}(\mathbf{x}^\mu))^2 + \frac{\tilde{\lambda}}{2}\mathbf{w}_{\mathcal{N}}^2\right)\right]. \tag{6.6}$$

As discussed in Section 5.3, queries selected for minimum entropy and for minimum generalization error (assuming the inference model is correct) are identical. Furthermore, query selection is not affected by the values of the hyperparameters $\tilde{\lambda}$ and β. The only effect of choosing $\tilde{\lambda}$ and β is therefore on the post-training student distribution (6.6).

Using the explicit forms of the probabilistic input-output relation (5.7) and the pseudo-prior (5.8) leading to the post-training distribution (6.6), the evidence (6.5) for $\tilde{\lambda}$ and β can easily be shown to be

$$\tilde{P}(\Theta^{(p)}|\tilde{\lambda},\beta) = \text{const}\cdot\frac{\tilde{\lambda}^{N/2}}{|\mathbf{M}_{\mathcal{N}}|^{1/2}}\left(\frac{\beta}{2\pi}\right)^{p/2}\exp\left[-\frac{\beta}{2}\left(\sum_{\mu=1}^{p}(y^\mu)^2 - \mathbf{a}^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{a}\right)\right]$$

where the multiplicative constant depends only on $\Theta^{(p)}$. Here we have, as usual, defined the matrix $\mathbf{M}_{\mathcal{N}}$ and vector $\mathbf{a}$ as

$$\mathbf{M}_{\mathcal{N}} = \tilde{\lambda}\mathbf{1} + \mathbf{A} \qquad \mathbf{A} = \frac{1}{N}\sum_{\mu}\mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}} \qquad \mathbf{a} = \frac{1}{\sqrt{N}}\sum_{\mu}y^\mu\mathbf{x}^\mu$$

with $\mathbf{1}$ denoting the $N\times N$ unit matrix. Choosing $\tilde{\lambda}$ and β to maximize the evidence is equivalent to maximizing the normalized log-evidence

$$\frac{1}{N}\ln\tilde{P}(\Theta^{(p)}|\tilde{\lambda},\beta) = -\frac{1}{2N}\mathrm{tr}\,\ln\left(\frac{1}{\tilde{\lambda}}\mathbf{M}_{\mathcal{N}}\right) + \frac{\alpha}{2}\ln\beta - \frac{\beta}{2N}\left(\sum_{\mu=1}^{p}(y^\mu)^2 - \mathbf{a}^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{a}\right) + \text{const.} \tag{6.7}$$

In general, the values of $\tilde{\lambda}$ and β obtained by maximizing (6.7) obviously depend on the training set $\Theta^{(p)}$. In the thermodynamic limit $N\to\infty$, however, their variances vanish as $O(1/N)$ [MS95]. With probability one, $\tilde{\lambda}$ and β therefore equal their average values, obtained by maximizing the *average* log-evidence $\frac{1}{N}\left\langle\ln\tilde{P}(\Theta^{(p)}|\tilde{\lambda},\beta)\right\rangle_{P(\Theta^{(p)})}$ [BS94, MS95]. To perform the average, an assumption about the functional form of

the teachers actually producing the training data has to be made. In the following, we restrict attention to the case of noisy nonlinear perceptron teachers considered in Chapter 4, writing the average output of teacher $\mathcal{V}$ for input $\mathbf{x}$ as

$$\bar{y} = \langle y \rangle_{P(y|\mathbf{x},\mathcal{V})} = \bar{g}\left(\frac{1}{\sqrt{N}}\mathbf{x}^{\mathrm{T}}\mathbf{w}_{\mathcal{V}}\right)$$

with a nonlinear noise-averaged output function $\bar{g}(\cdot)$, and its variance as

$$\left\langle (y-\bar{y})^2 \right\rangle_{P(y|\mathbf{x},\mathcal{V})} = \Delta^2\left(\frac{1}{\sqrt{N}}\mathbf{x}^{\mathrm{T}}\mathbf{w}_{\mathcal{V}}\right).$$

(compare eqs. (4.1, 4.2)). The teacher weight vector $\mathbf{w}_{\mathcal{V}}$ is assumed to be drawn from a Gaussian prior, $P(\mathcal{V}) \propto \exp(-\frac{1}{2}\mathbf{w}_{\mathcal{V}}^2/\sigma_{\mathcal{V}}^2)$, while input vectors $\mathbf{x}$ are taken to be normalized to $\mathbf{x}^2 = N\sigma_x^2$. Given these assumptions, the average of the log-evidence (6.7) can be carried out using the techniques described in Appendix 4.6. As in Chapters 3 and 4, it is convenient to express the result in terms of a rescaled version of the weight decay, $\lambda = \tilde{\lambda}/\sigma_x^2$:

$$\begin{aligned}\frac{1}{N}\left\langle \ln \tilde{P}(\Theta^{(p)}|\lambda,\beta)\right\rangle_{P(\Theta^{(p)})} &= \frac{1}{2}\int_{\lambda}^{\infty} d\lambda'\left(G(\lambda') - \frac{1}{\lambda'}\right) + \frac{\alpha}{2}\ln\beta \\ &+ \frac{\beta\sigma_{\mathrm{eff}}^2}{2}\left[(1-\lambda G(\lambda))\left(1-\frac{\lambda}{\lambda_{\mathrm{opt}}}\right) - \alpha\right] \end{aligned} \tag{6.8}$$

The notation employed here follows that introduced in Chapter 4: The 'response function' $G(\lambda)$ is the average of $\frac{\sigma_x^2}{N}\mathrm{tr}\,\mathbf{M}_{\mathcal{N}}^{-1}$ over the training inputs selected by queries and is given by

$$G(\lambda) = \frac{\Delta\alpha}{\lambda + \lfloor\alpha\rfloor + 1} + \frac{1-\Delta\alpha}{\lambda + \lfloor\alpha\rfloor} \tag{6.9}$$

where $\lfloor\alpha\rfloor$ is the greatest integer less than or equal to α and $\Delta\alpha = \alpha - \lfloor\alpha\rfloor$. The effective noise level of the teacher, given by

$$\sigma_{\mathrm{eff}}^2 = \sigma_{\mathrm{act}}^2 + \left[\left\langle \bar{g}^2(h)\right\rangle_h - \frac{1}{\sigma_{\mathcal{V}}^2\sigma_x^2}\left\langle h\bar{g}(h)\right\rangle_h^2\right] \qquad \sigma_{\mathrm{act}}^2 = \left\langle \Delta^2(h)\right\rangle_h$$

(where h is a Gaussian variable with zero mean and variance $\sigma_{\mathcal{V}}^2\sigma_x^2$) is the sum of the actual teacher noise and the effective noise arising from the fact that the linear student cannot reproduce a nonlinear teacher perfectly. The optimal weight decay λ_{opt}, finally, is the value of λ for which optimal generalization performance is achieved at zero learning temperature ($T = 1/\beta \to 0$); it is defined formally in eq. (4.13). Note that eq. (6.8) is also valid for learning from random examples if the appropriate value of the

response function,

$$G(\lambda) = \frac{1}{2\lambda}\left(1-\alpha-\lambda+\sqrt{(1-\alpha-\lambda)^2+4\lambda}\right)$$

is used; it therefore generalizes the result derived for linear teachers in Ref. [BS94]. By differentiating the average log-evidence (6.8) with respect to λ and β, one finds that it always has a maximum at

$$\lambda = \lambda_{\rm opt} \qquad \beta = 1/\sigma^2_{\rm eff}. \tag{6.10}$$

For learning from (minimum entropy or minimum generalization error) queries, it can be verified numerically that this maximum is unique for $\alpha > 1$; for random examples, the same holds true for *any* nonzero α. For a detailed discussion of the consequences of the hyperparameter assignments (6.10) resulting from the evidence procedure, we refer to Ref. [BS94]. For our purposes, it is sufficient to note that the resulting (average) generalization error is given by

$$\epsilon_{\rm g}(\alpha) - \frac{1}{2}\sigma^2_{\rm eff} = \frac{1}{2}\left(\sigma^2_{\rm eff}+\frac{1}{\beta}\right)G(\lambda_{\rm opt}) = \sigma^2_{\rm eff}G(\lambda_{\rm opt}) \tag{6.11}$$

where the contribution $\frac{1}{2}\sigma^2_{\rm eff}$ arising from the (effective) teacher noisė alone has been subtracted off explicitly on the left hand side. The result (6.11) follows from the generalization to finite $T = 1/\beta$ of the result (4.10) (cf. the discussion after eq. (4.16)). As was to be expected on the grounds that the evidence procedure actually assigns the value $\lambda_{\rm opt}$ to λ, eq. (6.11) is (up to the slightly modified prefactor[4]) identical to the result (3.41, 3.44) derived in Chapter 3 for the case of optimal weight decay. We found there that queries always lead to a lower generalization error than random examples. Our main conclusion at this point is therefore that the evidence procedure avoids the problem of query learning assuming the inference model is correct which we discussed in Section 5.3: Queries selected to minimize the generalization error can now no longer lead to a higher generalization error than random examples, but perform consistently better.

So far, we have not yet discussed the effects of combining the evidence procedure with query learning in the regime $\alpha < 1$. From the results (6.8, 6.9) obtained above,

[4] In Ref. [BS94], the generalization error is defined by the deviation of the *average* (over the post-training distribution) student output from the teacher output, which corresponds to an evaluation of the Bayes optimal predictor (see, e.g., [Ber85, Pil91]). This would simply remove the contribution proportional to $1/\beta$ from (6.11) and therefore yield exactly the optimal generalization error (3.41, 3.44).

one derives in this case

$$\frac{1}{N}\left\langle \ln \tilde{P}(\Theta^{(p)}|\tilde{\lambda},\beta)\right\rangle_{P(\Theta^{(p)})} = \frac{\alpha}{2}\left[\ln\left(\frac{\lambda}{\lambda+1}\right) + \ln\beta - \beta\sigma_{\rm eff}^2\frac{\lambda(\lambda_{\rm opt}+1)}{(\lambda+1)\lambda_{\rm opt}}\right] + \text{const.} \tag{6.12}$$

This means that the log-evidence becomes independent of α apart from an overall prefactor. For given λ, its maximum value with respect to β is

$$\frac{\alpha}{2}\left[\ln\left(\frac{\lambda_{\rm opt}}{\lambda_{\rm opt}+1}\right) - 1\right]$$

independently of λ. This means that the log-evidence does not have a unique maximum; instead, there is a line of degenerate maxima in β-λ space. Intuitively, the fact that for query learning in the regime $\alpha < 1$, the evidence does not contain sufficient information to determine λ and β uniquely may not be too surprising: We recall that queries for minimum entropy or minimum generalization error select input vectors which are mutually orthogonal (as long as this is possible, i.e., for $\alpha < 1$), because this yields most information about the student weight vector. In order to estimate both hyperparameters λ and β, however, some overlap between training inputs is needed to separate the signal and noise components of the training outputs. Queries can therefore be described as obtaining maximum information about the student $\mathcal{N}$, leaving only a limited amount of information about the hyperparameters. This suggests that, in particular when query learning is to be combined with inference model selection, it might be useful to select queries which maximize the *joint* information about student parameters *and* hyperparameters. We leave a more detailed investigation of this approach (which can be traced back to work on optimal experiment design in the statistics literature, see, e.g., [Bor75]), as a topic for future research.

Finally, note that our above conclusion that the evidence procedure is ill-defined for query learning in the regime $\alpha < 1$ is not an artefact of averaging the log-evidence over training sets. This can be verified directly from the expression (6.7) for the unaveraged log-evidence, using the fact that for queries and $\alpha < 1$, all input vectors $\mathbf{x}^\mu$ are orthogonal to each other (and of equal length, due to the assumed spherical constraint on inputs vectors).

6.4 Binary perceptron, $N = 2$

Let us now explore the performance achieved by query learning for minimum entropy combined with the evidence procedure, when applied to binary perceptron students.

We use the same general setup and notation as in Section 5.5. The inference models that we consider are parameterized by the parameter $p_{\mathcal{N}}$ which determines the assumed probability of sign-reversals of the training outputs due to noise. In contrast to the case of linear perceptron students, the value of $p_{\mathcal{N}}$ affects both how inferences are made from the training data *and* how queries are selected. This makes an analytical treatment of this scenario extremely difficult, and we have therefore resorted to computer simulations. For simplicity, we restrict ourselves to the small system limit, $N = 2$, leaving the analysis of larger systems for further study.

The simulation results shown below were obtained using the techniques described in Section 5.5.1. After each new training example had been added to the training set, the evidence (which, as can be seen by inserting (5.16, 5.17) into (6.5), is simply a polynomial in $p_{\mathcal{N}}$), was calculated and $p_{\mathcal{N}}$ set to the value at which it attained its maximum in the interval $p_{\mathcal{N}} \in [0, 1/2]$. For $\alpha = p \leq 3$, where the evidence is independent of $p_{\mathcal{N}}$, the value of $p_{\mathcal{N}}$ was left unchanged from its initial value, chosen here to be[5] $p_{\mathcal{N}} = 0$.

Figure 6.1 shows the average generalization error achieved by minimum entropy queries combined with the evidence procedure, for the case where the true noise process is sign-flip noise. The average generalization error is seen to decay to zero exponentially quickly with α, as was the case for an almost correct, fixed noise level $p_{\mathcal{N}} \approx p_{\mathcal{V}}$ (see Section 5.5.1). As expected, the values of $p_{\mathcal{N}}$ chosen by the evidence procedure (see the histogram in Figure 6.2) approach the true noise level $p_{\mathcal{V}}$ for increasing α. In this case, combining query learning with the evidence procedure is therefore a definite advantage: the problem of self-confirming hypotheses far from the truth is avoided, and the resulting generalization performance is significantly better than for random examples (compare Figure 5.4). For the case where the true noise process is Gaussian weight noise, one sees from Figure 6.3 that the evidence procedure still prevents the occurrence of self-confirming hypotheses far from the truth. However, the decay of the generalization error to zero is much slower than for sign-flip noise, in agreement with our qualitative description in Section 5.5.1 of weight noise as sign-flip noise which becomes increasingly strong as α increases. The $p_{\mathcal{N}}$-histogram in Figure 6.4 also supports this picture, with the largest values of $p_{\mathcal{N}}$ selected by the evidence procedure steadily increasing as more and more training examples are collected. The simulation results suggest that minimum entropy queries combined with the evidence procedure actually yield the same asymptotic power law decay of the generalization error with α

[5] It is a straightforward exercise to verify that query selection is independent of $p_{\mathcal{N}}$ for $\alpha \leq 3$. The results for $\alpha > 3$ are therefore entirely independent of the initial value of $p_{\mathcal{N}}$ at $\alpha = 0$.

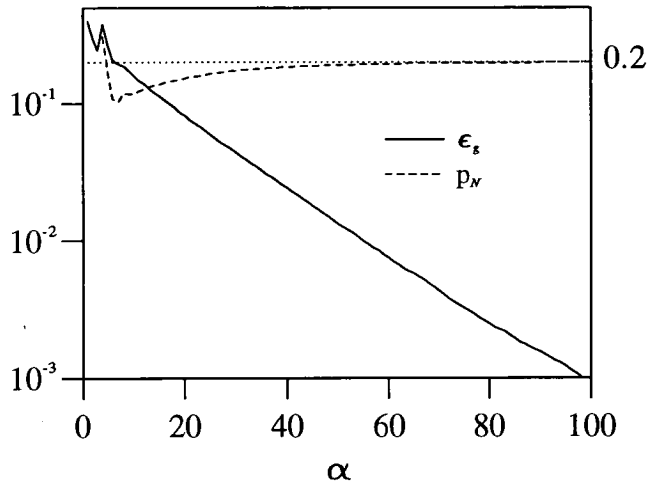

Figure 6.1. Query learning combined with the evidence procedure for the binary perceptron of size $N = 2$, when learning from a teacher with sign-flip noise level $p_\nu = 0.2$. The average generalization error (statistical error less than 4%) decays exponentially with the number of training examples α, while the average value of p_N selected by the evidence procedure tends to the true noise level $p_\nu = 0.2$.

as random examples ($\epsilon_g \propto \alpha^{-1/3}$), but with a reduced prefactor. This can be seen from the generalization performance improvement factor κ, which tends to a constant (≈ 2) for $\alpha \to \infty$ (see figure 6.3). Although much less drastic than in the case of sign-flip noise, where κ diverges for $\alpha \to \infty$, the improvement in generalization performance is therefore still appreciable: after all, a value of $\kappa \approx 2$, say, together with a power law decay $\epsilon_g \propto \alpha^{-1/3}$ implies that the number of training examples needed to achieve a certain generalization error is roughly eight times smaller for query learning (combined with the evidence procedure) than for random examples.

In summary of our results for the $N = 2$ binary perceptron, we have found that a combination of query learning with the evidence procedure avoids the problem of self-confirming hypotheses far from the truth. This result is particularly encouraging for the case of weight noise: Although the true inference model is not contained in the class of inference models from which the evidence procedure can select the most appropriate one (by choice of the hyperparameter p_N), the fact that the inference model is adapted at all seems to be sufficient to prevent self-confirming hypotheses. The evidence procedure also secures a generalization performance for queries which is consistently and significantly better than from random examples.

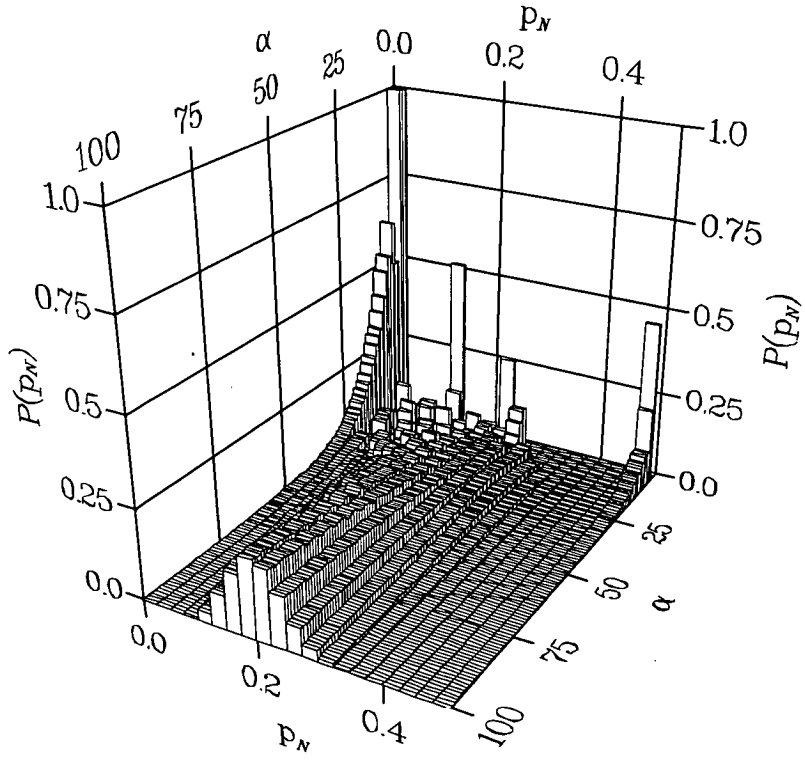

Figure 6.2. Histogram of p_N values selected by the evidence procedure for teacher sign-flip noise. As the number of training examples α increases, the distribution of p_N narrows down around the true noise level $p_N = 0.2$.

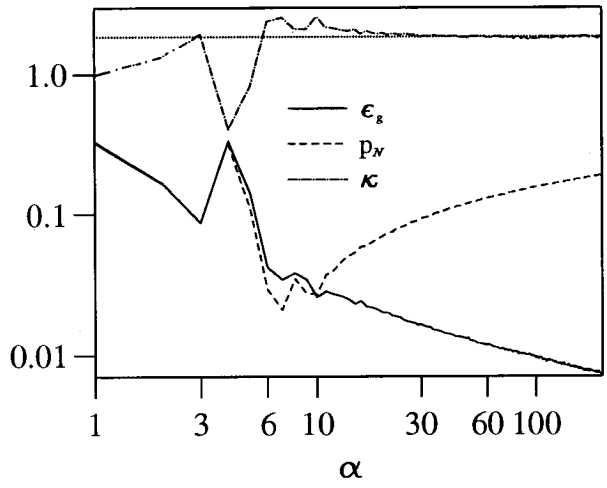

Figure 6.3. Query learning combined with the evidence procedure for the binary perceptron of size $N = 2$, when learning from a teacher with weight noise $\gamma = 0.99$. The average generalization error (statistical error less than 1.5%) decays as a slow power law with the number of training examples α, while the average value of p_N selected by the evidence procedure keeps increasing towards the maximum value $p_N = 0.5$. The improvement in generalization error κ over the best performance obtainable from random examples (corresponding to $p_N = 0.0$, see figure 5.8) tends to a constant, $\kappa \approx 2$, for $\alpha \to \infty$.

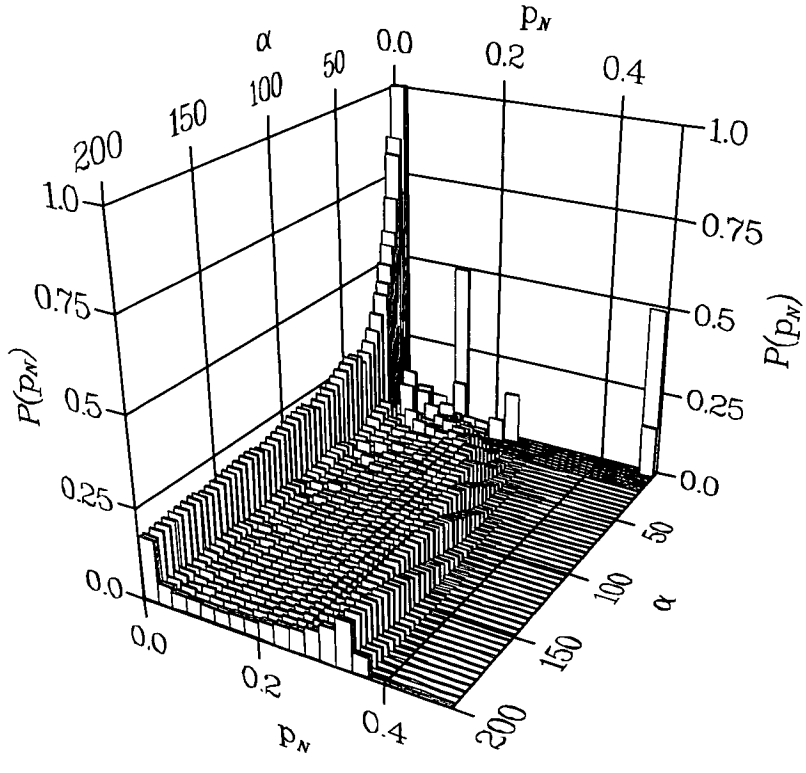

Figure 6.4. Histogram of $p_{\mathcal{N}}$ values selected by the evidence procedure for teacher weight noise $\gamma = 0.99$. As the number of training examples α increases, larger and larger values of $p_{\mathcal{N}}$ are selected.

6.5 Summary and conclusion

Combining query learning with an adaptation of the inference model using the evidence procedure, we have found for the scenarios considered that the problem of self-confirming hypotheses is avoided, and that the generalization error is consistently reduced compared to learning from random examples. Overall, the examples that we have studied suggest that combining query learning and model selection is a promising method for making query learning robust against inference model misspecification; it will of course be necessary to test this statement for more realistic learning scenarios before any claim to generality can be made.

Let us conclude by mentioning a few possible extensions of the work presented in this chapter. We have only explored one of a variety of conceivable methods for making query learning more robust against inference model misspecification. First of all, other procedures for selecting appropriate inference models based on, for example, cross-validation, bootstrap etc. (see, e.g., [Sto74, Sto77]) could be tried and compared to the evidence procedure, building on the analyses for learning from random examples [MS95]. Furthermore, it would be interesting to consider the possibility of enlarging the space of inference models dynamically as more training data are received. Finally, it might also be possible to tackle the problem of self-confirming hypotheses by relating it to the 'exploration-exploitation' dilemma studied by researchers in the field of reinforcement learning (see, e.g., [BSW90, BBS91, TM92]). This could prove useful because the problem of self-confirming hypotheses can be viewed as a case of 'exploiting' the teacher's knowledge to make the student distribution as narrow as possible, while not 'exploring' the input space sufficiently.

Chapter 7

Towards realistic neural networks I: Finite size effects

Abstract

Our previous analyses of the efficacy of query learning have focussed mainly on the thermodynamic limit $N \to \infty$ of infinite system size. We now remove this restriction by studying finite N corrections. In particular, we consider a learning scenario with a linear perceptron student learning from a noisy linear teacher, for which most properties of learning and generalization can be derived from the average response function G. A method for calculating G using only simple matrix identities and partial differential equations is presented. Using this method, we first rederive the known result for G in the thermodynamic limit of infinite perceptron size N, which has previously been calculated using replica and diagrammatic methods. We also show explicitly that the response function is self-averaging in the thermodynamic limit. Extensions of the method to more general learning scenarios with anisotropic teacher space priors, input distributions, and weight decay terms are discussed briefly. Finally, finite size effects are considered by calculating the $O(1/N)$ correction to G. We verify the result by computer simulations and discuss the consequences for the efficacy of query learning in linear perceptrons of finite size.

7.1 Introduction

In the present chapter, we extend our previous analyses of query learning by removing the restriction to the thermodynamic limit of large system size, $N \to \infty$. In particular, we study finite size effects for learning with linear perceptron students by calculating

$O(1/N)$ corrections to the average generalization error and related quantities. We use the results to bound the critical system size N_c above which the thermodynamic limit predictions are valid to a good approximation. Our main conclusion will be that except in a small region in parameter space around a phase transition in the learning behaviour, the critical system size is small enough for the thermodynamic limit to be directly relevant to real-world system sizes N of the order of several tens or hundreds.

Let us first review briefly the scenario of linear perceptron learning that we consider, described in more detail in Chapter 3. A linear perceptron student $\mathcal{N}$ is parameterized in terms of a weight vector $\mathbf{w}_{\mathcal{N}} \in \mathbb{R}^N$ and maps real input vectors $\mathbf{x} \in \mathbb{R}^N$ to real outputs $y \in \mathbb{R}$ according to

$$y = f_{\mathcal{N}}(\mathbf{x}) = \frac{1}{\sqrt{N}} \mathbf{w}_{\mathcal{N}}^{\mathrm{T}} \mathbf{x}.$$

A commonly used learning algorithm for the linear perceptron is minimization of the training error E_{t}, i.e., the error that the student $\mathcal{N}$ makes on the training set. Using as an error measure the usual squared output deviation, the training error for a given set of p training examples, $\Theta^{(p)} = \{(\mathbf{x}^\mu, y^\mu), \mu = 1 \ldots p\}$, is

$$E_{\mathrm{t}} = \sum_{\mu=1}^{p} \frac{1}{2} (y^\mu - f_{\mathcal{N}}(\mathbf{x}^\mu))^2 = \frac{1}{2} \sum_{\mu=1}^{p} \left(y^\mu - \frac{1}{\sqrt{N}} \mathbf{w}_{\mathcal{N}}^{\mathrm{T}} \mathbf{x}^\mu \right)^2 .$$

To prevent the student from fitting noise in the training data, a quadratic weight decay term $\frac{1}{2}\lambda \mathbf{w}_{\mathcal{N}}^2$ is normally added to the training error, with the value of the weight decay parameter λ determining how strongly large weight vectors are penalized. Thus, it is the function

$$E = E_{\mathrm{t}} + \frac{1}{2} \lambda \mathbf{w}_{\mathcal{N}}^2 \tag{7.1}$$

that is minimized. This minimization can be realized by stochastic gradient descent, for example, leading to the Gibbs distribution of students (3.27),

$$P(\mathcal{N}|\Theta^{(p)}) \propto \exp \left[-\beta \left(\sum_{\mu=1}^{p} \frac{1}{2} (y^\mu - f_{\mathcal{N}}(\mathbf{x}^\mu))^2 + \frac{\lambda}{2} \mathbf{w}_{\mathcal{N}}^2 \right) \right] \tag{7.2}$$

where the 'temperature' $T = 1/\beta$ measures 'how stochastic' the gradient descent is. This distribution has an associated entropy

$$S_{\mathcal{N}}(\Theta^{(p)}) = -\frac{N}{2} \ln \beta - \frac{1}{2} \ln |\mathbf{M}_{\mathcal{N}}| + \text{constant}$$

where the matrix $\mathbf{M}_{\mathcal{N}}$ is related to the correlation matrix of the training inputs by

$$\mathbf{M}_{\mathcal{N}} = \lambda \mathbf{1} + \mathbf{A} \qquad \mathbf{A} = \frac{1}{N} \sum_{\mu} \mathbf{x}^{\mu} (\mathbf{x}^{\mu})^{\mathrm{T}}. \tag{7.3}$$

We shall consider minimum (student space) entropy queries in the following, which are chosen to minimize $S_{\mathcal{N}}$. If we assume training inputs to be normalized to $\mathbf{x}^2 = N$, then this implies that each new training input has to be chosen along the direction of an eigenvector of $\mathbf{M}_{\mathcal{N}}$ with minimal eigenvalue, as explained in Section 3.3. Applying such queries in sequence, one obtains 'blocks' of size N of mutually orthogonal (but otherwise random) training inputs.

As usual, we have to specify what type of rule our linear perceptron student is trying to learn in order to examine the resulting generalization performance. In other words, we have to define the teacher space. The simplest assumption is that the problem is perfectly learnable, i.e., that the teacher, like the student, is a linear perceptron. A teacher $\mathcal{V}$ is then specified by a weight vector $\mathbf{w}_{\mathcal{V}}$ and maps a given input $\mathbf{x}$ to the output $y = f_{\mathcal{V}}(\mathbf{x}) = \mathbf{w}_{\mathcal{V}}^{\mathrm{T}} \mathbf{x} / \sqrt{N}$. We assume that the test inputs for which the student is asked to predict the corresponding outputs are drawn from an isotropic distribution over the hypersphere $\mathbf{x}^2 = N$, $P(\mathbf{x}) \propto \delta(\mathbf{x}^2 - N)$. The generalization error, i.e., the average error that a student $\mathcal{N}$ makes on a random input when compared to teacher $\mathcal{V}$, is then given by eq. (3.26):[1]

$$\epsilon_{\mathrm{g}} = \frac{1}{2} \left\langle (f_{\mathcal{N}}(\mathbf{x}) - f_{\mathcal{V}}(\mathbf{x}))^2 \right\rangle_{P(\mathbf{x})} = \frac{1}{2N} (\mathbf{w}_{\mathcal{N}} - \mathbf{w}_{\mathcal{V}})^2. \tag{7.4}$$

The main quantity of interest to us will be the average of the generalization error over all possible training sets and teachers; to avoid clutter, we write this average simply as ϵ_{g}. We assume that the training outputs generated by the teacher are corrupted by additive noise, $y^{\mu} = f_{\mathcal{V}}(\mathbf{x}^{\mu}) + \eta^{\mu}$, where the η^{μ} are independent random variables with zero mean and variance σ^2. To perform the average over teachers, we assume that teacher weight vectors are sampled randomly from an isotropic Gaussian prior, $P(\mathbf{w}_{\mathcal{V}}) \propto \exp(-\frac{1}{2}\mathbf{w}_{\mathcal{V}}^2)$. Specializing to the limit $T \to 0$ (corresponding to deterministic gradient descent), the resulting average generalization error is given by eq. (3.49),

$$\epsilon_{\mathrm{g}} = \frac{1}{2} \left[\sigma^2 G + \lambda (\sigma^2 - \lambda) \frac{\partial G}{\partial \lambda} \right] \tag{7.5}$$

[1] As in Section 3.3, we neglect the additive contribution arising from noise on the teacher outputs alone.

where G is the average of the *response function* over the training inputs:

$$G = \langle \mathcal{G} \rangle_{P(\{\mathbf{x}^\mu\})} \qquad \mathcal{G} = \frac{1}{N} \mathrm{tr}\, \mathbf{M}_N^{-1}. \tag{7.6}$$

In order to determine the effects of finite system size N on the improvement in generalization performance due to query learning,

$$\kappa = \frac{\epsilon_g(\text{random examples})}{\epsilon_g(\text{minimum entropy queries})} \tag{7.7}$$

we therefore need to calculate the average response function for random examples and minimum entropy queries. For *minimum entropy queries*, we denote the response function by $\mathcal{G}_Q$ and its average by G_Q; the two are actually identical and can be evaluated exactly for any system size N, with the result (3.42)

$$G_Q \equiv \mathcal{G}_Q = \frac{\Delta\alpha}{\lambda + \lfloor\alpha\rfloor + 1} + \frac{1 - \Delta\alpha}{\lambda + \lfloor\alpha\rfloor}$$

where $\lfloor\alpha\rfloor$ denotes the integer part of α and $\Delta\alpha = \alpha - \lfloor\alpha\rfloor$ its non-integer part. Our main task in this chapter is therefore the calculation of the average response function G for *random training inputs.*

We will sometimes find it useful to interpret our results in terms of the average eigenvalue spectrum $\rho(a)$ of the input correlation matrix $\mathbf{A}$. It is defined as

$$\rho(a) = \left\langle \frac{1}{N} \sum_{i=1}^{N} \delta(a - a_i) \right\rangle_{P(\{\mathbf{x}^\mu\})} \tag{7.8}$$

where we have denoted the eigenvalues of $\mathbf{A}$ by a_i $(i = 1 \ldots N)$. From this definition, it follows directly that

$$G = \int da\, \frac{\rho(a)}{\lambda + a}\,. \tag{7.9}$$

Conversely, $\rho(a)$ can be related to G by using the identity

$$\delta(x) = \frac{1}{\pi} \lim_{\epsilon \to 0^+} \mathrm{Im}\, \frac{1}{x - i\epsilon}$$

which yields [Kro92]

$$\rho(a) = \frac{1}{\pi} \lim_{\epsilon \to 0^+} \mathrm{Im}\, G|_{\lambda = -a - i\epsilon}\,. \tag{7.10}$$

As could have been expected, the singularities of the average response function G in the complex λ plane determine the average eigenvalue spectrum of the input correlation matrix $\mathbf{A}$.

Parenthetically, we note that the result (7.5) for the average generalization error can be viewed as the long time limit of gradient descent in weight space on the energy function E given by eq. (7.1). The average eigenvalue spectrum $\rho(a)$ can be used to determine the corresponding time evolution for finite times [Kro92]; for a discussion of finite size effects in this context, see Ref. [Sol94].

Equations (7.5, 7.10) show that the key quantity determining learning and generalization in the linear perceptron is the average response function G defined in (7.6). This function has previously been calculated in the thermodynamic limit, $N \to \infty$ at $\alpha = p/N = \text{const.}$, using a diagrammatic expansion [HKT89] and the replica method [Opp89, KO91]. In Section 7.2, we present what we believe to be a much simpler method for calculating G, based on simple matrix identities. We also show explicitly that the response function $\mathcal{G}$ is self-averaging in the thermodynamic limit, which means that the fluctuations of $\mathcal{G}$ around its average G become vanishingly small as $N \to \infty$. This implies, for example, that the generalization error is also self-averaging. In Section 7.3, the method is extended to more general cases such as anisotropic teacher space priors and input distributions, and general quadratic penalty terms. Finally, finite size effects are considered in Section 7.4, where we calculate the $O(1/N)$ correction to G, verify the result by computer simulations and examine the resulting effects on the generalization performance for learning from random examples and on the efficacy of query learning. We conclude in Section 7.5 with a brief summary and discussion of our results.

7.2 Calculating the response function

Our method for calculating the average response function G is based on a recursion relation relating the values of the (unaveraged) response function $\mathcal{G}$ for p and $p+1$ training examples. Assume that we are given a set of p training examples with corresponding matrix $\mathbf{M}_{\mathcal{N}}$. By adding a new training example with input $\mathbf{x}$, we obtain the matrix $\mathbf{M}_{\mathcal{N}}^{+} = \mathbf{M}_{\mathcal{N}} + \frac{1}{N}\mathbf{x}\mathbf{x}^{\mathrm{T}}$. It is straightforward to show that the inverse of $\mathbf{M}_{\mathcal{N}}^{+}$ can be expressed as

$$(\mathbf{M}_{\mathcal{N}}^{+})^{-1} = \mathbf{M}_{\mathcal{N}}^{-1} - \frac{\frac{1}{N}\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-1}}{1 + \frac{1}{N}\mathbf{x}^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{x}}. \tag{7.11}$$

(One way of proving this identity is to multiply both sides by $\mathbf{M}_{\mathcal{N}}^{+}$ and to exploit the fact that $\mathbf{M}_{\mathcal{N}}^{+}\mathbf{M}_{\mathcal{N}}^{-1} = \mathbf{1} + \frac{1}{N}\mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-1}$.) Taking the trace, we obtain the following

recursion relation for $\mathcal{G}$:

$$\mathcal{G}(p+1) = \mathcal{G}(p) - \frac{1}{N}\frac{\frac{1}{N}\mathbf{x}^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-2}\mathbf{x}}{1+\frac{1}{N}\mathbf{x}^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-1}\mathbf{x}}. \tag{7.12}$$

Now denote $z_i = \frac{1}{N}\mathbf{x}^{\mathrm{T}}\mathbf{M}_{\mathcal{N}}^{-i}\mathbf{x}$ $(i = 1, 2)$. With $\mathbf{x}$ drawn randomly from the assumed input distribution $P(\mathbf{x}) \propto \delta(\mathbf{x}^2 - N)$, the z_i can readily be shown to be random variables with means and (co-)variances

$$\langle z_i \rangle = \frac{1}{N}\mathrm{tr}\ \mathbf{M}_{\mathcal{N}}^{-i} \qquad \langle \Delta z_i \Delta z_j \rangle = \frac{2}{N+2}\left[\frac{1}{N}\mathrm{tr}\ \mathbf{M}_{\mathcal{N}}^{-i-j} - \left(\frac{1}{N}\mathrm{tr}\ \mathbf{M}_{\mathcal{N}}^{-i}\right)\left(\frac{1}{N}\mathrm{tr}\ \mathbf{M}_{\mathcal{N}}^{-j}\right)\right]$$

where we have used the notation $\Delta z_i = z_i - \langle z_i \rangle$. Combining this with the fact that for $k > 0$, $\mathrm{tr}\ \mathbf{M}_{\mathcal{N}}^{-k} \leq N\lambda^{-k} = O(N)$, we have that the fluctuations Δz_i of the z_i around their average values are $O(1/\sqrt{N})$; inserting this into (7.12), we obtain

$$\begin{aligned} \mathcal{G}(p+1) &= \mathcal{G}(p) - \frac{1}{N}\frac{\frac{1}{N}\mathrm{tr}\ \mathbf{M}_{\mathcal{N}}^{-2}}{1+\frac{1}{N}\mathrm{tr}\ \mathbf{M}_{\mathcal{N}}^{-1}} + O(N^{-3/2}) \\ &= \mathcal{G}(p) + \frac{1}{N}\frac{\partial \mathcal{G}(p)}{\partial \lambda}\frac{1}{1+\mathcal{G}(p)} + O(N^{-3/2}). \end{aligned} \tag{7.13}$$

Starting from $\mathcal{G}(0) = 1/\lambda$, we can apply this recursion p times to obtain $\mathcal{G}(p)$ up to terms which add up to at most $O(pN^{-3/2})$. This shows that in the thermodynamic limit, defined by $N \to \infty$, $\alpha = p/N = \text{const.}$, the response function $\mathcal{G}$ is self-averaging: whatever the training set, the value of $\mathcal{G}$ will always be the same up to fluctuations of $O(N^{-1/2})$. In fact, we shall show in Section 7.4 that the fluctuations of $\mathcal{G}$ are only $O(1/N)$. This means that the $O(N^{-3/2})$ fluctuations from each iteration of (7.13) are only weakly correlated, so that they add up like independent random variables to give a total fluctuation for $\mathcal{G}(p)$ of $O((p/N^3)^{1/2}) = O(1/N)$.

We have seen that, in the thermodynamic limit, $\mathcal{G}$ is identical to its average, G, because its fluctuations are vanishingly small. To calculate the value of G in this limit as a function of α and λ, we replace $\mathcal{G}$ by G in (7.13), insert the relation $G(p+1) - G(p) = \frac{1}{N}\partial G(\alpha)/\partial\alpha + O(1/N^2)$, and neglect all finite N corrections. This yields the partial differential equation

$$\frac{\partial G}{\partial \alpha} - \frac{\partial G}{\partial \lambda}\frac{1}{1+G} = 0 \tag{7.14}$$

which can readily be solved using the method of characteristic curves. A brief account of this method can be found in Appendix 7.6. Using the initial condition $G|_{\alpha=0} = 1/\lambda$, one

obtains $1/G = \lambda + \alpha/(1+G)$ which leads to the well-known result (see, e.g., [HKT89])

$$G = \frac{1}{2\lambda}\left(1 - \alpha - \lambda + \sqrt{(1-\alpha-\lambda)^2 + 4\lambda}\right). \tag{7.15}$$

In the complex λ plane, G has a pole at $\lambda = 0$ and a branch cut arising from the root; according to (7.10), these singularities determine the average eigenvalue spectrum $\rho(a)$ of $\mathbf{A}$, with the result [Kro92]

$$\rho(a) = (1-\alpha)\Theta(1-\alpha)\delta(a) + \frac{1}{2\pi a}\sqrt{(a_+ - a)(a - a_-)} \tag{7.16}$$

where $\Theta(x)$ is the Heaviside step function, $\Theta(x) = 1$ for $x > 0$ and 0 otherwise. The root in (7.16) only contributes when its argument is non-negative, i.e., for a between the 'spectral limits' a_- and a_+, which have the values $a_\pm = (1 \pm \sqrt{\alpha})^2$. Since $\mathcal{G}$ is self-averaging, the fluctuations of the true eigenvalue spectrum of $\mathbf{A}$ around its average $\rho(a)$ are also vanishingly small in the thermodynamic limit[2].

7.3 Extensions to more general learning scenarios

We now discuss some extensions of our method to more general learning scenarios. First, consider the case of an anisotropic teacher space prior, given by $P(\mathbf{w}_\nu) \propto \exp(-\frac{1}{2}\mathbf{w}_\nu^{\mathrm{T}}\mathbf{\Sigma}_\nu^{-1}\mathbf{w}_\nu)$ with a symmetric positive definite covariance matrix $\mathbf{\Sigma}_\nu$. This does not affect the definition of the response function, but (7.5) now has to be replaced by

$$\epsilon_{\mathrm{g}}(t \to \infty) = \frac{1}{2}\left(\frac{1}{N}\mathrm{tr}\ \mathbf{\Sigma}_\nu\right)\left[\tilde{\sigma}^2 G + \lambda(\tilde{\sigma}^2 - \lambda)\frac{\partial G}{\partial \lambda}\right]$$

with a renormalized noise level $\tilde{\sigma}^2 = \sigma^2/(\frac{1}{N}\mathrm{tr}\ \mathbf{\Sigma}_\nu)$. The factor $\frac{1}{N}\mathrm{tr}\ \mathbf{\Sigma}_\nu$ determines by how much the average squared length of the teacher weight vector is now larger than for the isotropic teacher space prior considered in the previous section. This factor also scales the size of the typical squared teacher output. Therefore, it appears as a multiplicative factor in the generalization error, and also determines the renormalized noise level (which is, effectively, a mean square noise-to-signal ratio).

As a second extension, assume that the inputs are drawn from an anisotropic distribution, $P(\mathbf{x}) \propto \delta(\mathbf{x}^{\mathrm{T}}\mathbf{\Sigma}^{-1}\mathbf{x} - N)$. It can then be shown that the asymptotic value of the average generalization error is still given by (7.5) if the average response function is redefined to be $G = \left\langle \frac{1}{N}\mathrm{tr}\ \mathbf{\Sigma}\mathbf{M}_N^{-1}\right\rangle$. This modified response function can be calculated

[2] More precisely, the fluctuations of linear functionals of the eigenvalue spectrum of $\mathbf{A}$ (which is, mathematically, a distribution) vanish as $N \to \infty$.

as follows: First we rewrite G as $\left\langle \frac{1}{N}\mathrm{tr}\,(\lambda\mathbf{\Sigma}^{-1}+\tilde{\mathbf{A}})^{-1}\right\rangle$, where $\tilde{\mathbf{A}} = \frac{1}{N}\sum_\mu (\tilde{\mathbf{x}}^\mu)^{\mathrm{T}}\tilde{\mathbf{x}}^\mu$ is the correlation matrix of the transformed input examples[3] $\tilde{\mathbf{x}}^\mu = \mathbf{\Sigma}^{-1/2}\mathbf{x}^\mu$. Since the $\tilde{\mathbf{x}}^\mu$ are distributed according to $P(\tilde{\mathbf{x}}^\mu) \propto \delta((\tilde{\mathbf{x}}^\mu)^2 - N)$, the problem is thus reduced to finding the average response function $G_L = \langle \mathcal{G}_L\rangle = \left\langle \frac{1}{N}\mathrm{tr}\,(\mathbf{L}+\mathbf{A})^{-1}\right\rangle$ for isotropically distributed inputs and $\mathbf{L} = \lambda\mathbf{\Sigma}^{-1}$. As explained in Appendix 7.6, a differential equation analogous to (7.14) holds for G_L. Together with the initial condition $G_L|_{\alpha=0} = \frac{1}{N}\mathrm{tr}\,\mathbf{L}^{-1}$, one obtains G_L as the solution of the implicit equation

$$G_L = \frac{1}{N}\mathrm{tr}\left(\mathbf{L} + \frac{\alpha}{1+G_L}\mathbf{1}\right)^{-1}. \tag{7.17}$$

As explained above, the modified response function $G = \left\langle \frac{1}{N}\mathrm{tr}\,\mathbf{\Sigma}\mathbf{M}_{\mathcal{N}}^{-1}\right\rangle$ for the case of an anisotropic input distribution considered here is given by the value of G_L which solves (7.17) for $\mathbf{L} = \lambda\mathbf{\Sigma}^{-1}$. If the eigenvalue spectrum of $\mathbf{\Sigma}$ has a particularly simple form, then the resulting dependence of G on α and λ can be expressed analytically, but in general (7.17) will have to be solved numerically.

Finally, one can also investigate the effect of a general quadratic weight decay term, $\frac{1}{2}\mathbf{w}_{\mathcal{N}}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{w}_{\mathcal{N}}$, in the energy function E, eq. (7.1), which defines the training algorithm. This modifies the definition (7.3) of the matrix $\mathbf{M}_{\mathcal{N}}$ to $\mathbf{M}_{\mathcal{N}} = \mathbf{\Lambda}+\mathbf{A}$, and the calculation of the average generalization error becomes more complicated in this case. In addition to the average response function $G = \left\langle \frac{1}{N}\mathrm{tr}\,\mathbf{M}_{\mathcal{N}}^{-1}\right\rangle$, which can be obtained as the solution of (7.17) for $\mathbf{L} = \mathbf{\Lambda}$, one now also needs to know the modified response functions $G_{\Lambda^n} = \left\langle \frac{1}{N}\mathrm{tr}\,\mathbf{\Lambda}^n\mathbf{M}_{\mathcal{N}}^{-1}\right\rangle$ for $n = 1, 2$. Fortunately, it is possible to calculate the general modified response function $G_{BL} = \left\langle \frac{1}{N}\mathrm{tr}\,\mathbf{B}(\mathbf{L}+\mathbf{A})^{-1}\right\rangle$ for positive definite symmetric $\mathbf{L}$ and a general matrix $\mathbf{B}$ by extending the methods of the previous section. As outlined in Appendix 7.6, one obtains in the thermodynamic limit a differential equation for G_{BL} similar but not exactly identical to (7.14), which can be solved to give

$$G_{BL} = \frac{1}{N}\mathrm{tr}\,\mathbf{B}\left(\mathbf{L} + \frac{\alpha}{1+G_L}\mathbf{1}\right)^{-1}. \tag{7.18}$$

Thus, G_{BL} can be calculated straight away once G_L is known. In the specific case of a general quadratic weight decay that we consider here, one has $\mathbf{L} = \mathbf{\Lambda}$ and $G_L = G$, and by setting $\mathbf{B} = \mathbf{\Lambda}$ and $\mathbf{\Lambda}^2$ in (7.18), one obtains $G_\Lambda = 1 - \alpha G/(1+G)$ and $G_{\Lambda^2} = \frac{1}{N}\mathrm{tr}\,\mathbf{\Lambda} - \alpha/(1+G) + \alpha^2 G/(1+G)^2$. Using these relations, the average generalization error can be written in terms of G alone, although the final expressions become rather more

[3] We write $\mathbf{\Sigma}^{-1/2}$ for the unique positive definite symmetric matrix which obeys $\mathbf{\Sigma}^{-1/2}\mathbf{\Sigma}^{-1/2} = \mathbf{\Sigma}^{-1}$.

cumbersome than (7.5). We note parenthetically that expressions (7.17) and (7.18) can also be obtained using diagrammatic methods [Saa].

7.4 Finite size effects

So far, we have considered the calculation of the response function for random training examples in the thermodynamic limit of perceptrons of infinite size N. The results are clearly only approximately valid for real, finite systems, and we therefore now turn to an investigation of the corrections for finite N, by calculating the $O(1/N)$ correction to G and $\rho(a)$. We will use the results to analyse the effects of finite system size on the average generalization error ϵ_g for random examples and the improvement in generalization performance κ due to query learning. Rather than directly calculating the desired quantities for the case of normalized inputs of interest ($P(\mathbf{x}) \propto \delta(\mathbf{x}^2 - N)$), we shall first study the case of *Gaussian* distributed inputs, $P(\mathbf{x}) \propto \exp(-\frac{1}{2}\mathbf{x}^2)$, for which both the analysis and the interpretation of the results is somewhat simpler. The superscipt 'G' will be used to denote quantities for Gaussian inputs, while 'n' refers to inputs normalized to $\mathbf{x}^2 = N$.

First note that, for $\lambda = 0$, the exact value of the average response function for Gaussian inputs is [Han93]

$$G^{\mathrm{G}}|_{\lambda=0} = \frac{1}{N}\left\langle \mathrm{tr}\ \mathbf{A}^{-1} \right\rangle = (\alpha - 1 - 1/N)^{-1} \tag{7.19}$$

for $\alpha > 1 + 1/N$. This result follows straightforwardly from the fact that the inverse input correlation matrix, $\mathbf{A}^{-1}$, obeys an 'inverted Wishart distribution' (see, e.g., [Eat83], Def. 8.1 and Exercise 8.7). Eq. (7.19) clearly admits a series expansion in powers of $1/N$. Assuming that a similar expansion also exists for nonzero λ, we write

$$G^{\mathrm{G}} = G_0 + G_1^{\mathrm{G}}/N + O(1/N^2). \tag{7.20}$$

Here G_0 is the value of G in the thermodynamic limit as given by (7.15), which is identical for Gaussian and for normalized inputs[4]. We calculate G_1^{G} below, and verify the analytical result by computer simulations. Note that there is no *a priori* guarantee that an expansion of the type (7.20) exists; compare for example the results of [DGPB91], which suggest that for the binary perceptron, finite size effects depend

[4]This follows directly from our derivation of the recursion relation (7.13), which depends only on the averages of the variables $z_i = \frac{1}{N}\mathbf{x}^{\mathrm{T}}\mathbf{M}_N^{-i}\mathbf{x}$ ($i = 1, 2$) and hence only on the correlation matrix $\left\langle \mathbf{x}\mathbf{x}^{\mathrm{T}} \right\rangle$ of the training inputs.

non-analytically on $1/N$. However, the simulation results presented below do provide compelling evidence for the existence of the expansion (7.20) of the average response function in powers of $1/N$.

For finite N, not only the corrections to the average response function G but also the fluctuations $\Delta\mathcal{G} = \mathcal{G} - G$ of $\mathcal{G}$ around its average value G become relevant. For $\lambda = 0$, the variance of these fluctuations is known to have a power series expansion in $1/N$ (see, e.g., [BSS95]), and again we assume a similar expansion for finite λ,

$$\left\langle (\Delta\mathcal{G})^2 \right\rangle^{\mathrm{G}} = (\Delta^{\mathrm{G}})^2/N + O(1/N^2).$$

Here the first term is $O(1/N)$ and not $O(1)$ because, as discussed in Section 7.2, the fluctuations of $\mathcal{G}$ for large N can be no greater than $O(N^{-1/2})$.

To calculate G_1^{G} and $(\Delta^{\mathrm{G}})^2$, we start again from the recursion relation (7.12). However, now we cannot neglect terms involving fluctuations of $\mathcal{G}$ and $z_i = \frac{1}{N}\mathbf{x}^{\mathrm{T}}\mathbf{M}_N^{-i}\mathbf{x}$ $(i = 1, 2)$, but have to expand everything up to second order in the fluctuation quantities $\Delta\mathcal{G}$ and Δz_i. To carry out the averages, the (co-)variances of z_1 and z_2 are needed, given by

$$\langle \Delta z_i \Delta z_j \rangle = \frac{2}{N^2} \operatorname{tr} \mathbf{M}_N^{-i-j}$$

for Gaussian inputs. Averaging over the training inputs and collecting orders of $1/N$ yields after some straightforward algebra the known equation (7.14) for G_0 and

$$\begin{aligned} \frac{\partial G_1^{\mathrm{G}}}{\partial\alpha} - \frac{\partial G_1^{\mathrm{G}}}{\partial\lambda}\frac{1}{1+G_0} &= \frac{\partial}{\partial\lambda}\left[\left(\frac{\partial G_0}{\partial\lambda} - \frac{1}{2}(\Delta^{\mathrm{G}})^2\right)\frac{1}{(1+G_0)^2}\right] \\ &\quad - \frac{G_1^{\mathrm{G}}}{(1+G_0)^2}\frac{\partial G_0}{\partial\lambda} - \frac{1}{2}\frac{\partial^2 G_0}{\partial\alpha^2}. \end{aligned} \tag{7.21}$$

By squaring the difference between (7.12) and its average over the training inputs, one can similarly derive an equation for $(\Delta^{\mathrm{G}})^2$:

$$\frac{\partial(\Delta^{\mathrm{G}})^2}{\partial\alpha} - \frac{\partial(\Delta^{\mathrm{G}})^2}{\partial\lambda}\frac{1}{1+G_0} = -2\frac{(\Delta^{\mathrm{G}})^2}{(1+G_0)^2}\frac{\partial G_0}{\partial\lambda}. \tag{7.22}$$

Details of the solution of these two partial differential equations are again relegated to Appendix 7.6. At $\alpha = 0$, one has $\mathcal{G} = G = G_0 = 1/\lambda$ (for both Gaussian and normalized inputs). It follows that $G_1^{\mathrm{G}} = (\Delta^{\mathrm{G}})^2 = 0$; using these initial conditions, one finds $(\Delta^{\mathrm{G}})^2 \equiv 0$ for all α and λ, and

$$G_1^{\mathrm{G}} = \frac{G_0^2(1-\lambda G_0)}{(1+\lambda G_0^2)^2}. \tag{7.23}$$

In the limit $\lambda \to 0$, $G_1^{\rm G} = 1/(\alpha - 1)^2$ consistent with (7.19); likewise, the result $(\Delta^{\rm G})^2 \equiv 0$ agrees with the exact series expansion of the variance of the fluctuations of $\mathcal{G}$ for $\lambda = 0$, which begins with an $O(1/N^2)$ term [BSS95].

Note from (7.23) that $G_1^{\rm G}$ is positive for all $\alpha > 0$, since $G_0 < 1/\lambda$. This means that at least to first order in $1/N$, the average response function is an increasing function of $1/N$. This can be related to the $1/N$ corrections to the average eigenvalue spectrum $\rho(a)$ of the input correlation matrix. Setting

$$\rho^{\rm G}(a) = \rho_0(a) + \rho_1^{\rm G}(a)/N + O(1/N^2) \tag{7.24}$$

where $\rho_0(a)$ is the $N \to \infty$ result given by (7.16), one derives from (7.10) and (7.23)

$$\rho_1^{\rm G}(a) = \frac{1}{4}\delta(a - a_+) + \frac{1}{4}\delta(a - a_-) - \frac{1}{2\pi}\frac{1}{\sqrt{(a_+ - a)(a - a_-)}} \,. \tag{7.25}$$

Figure 7.1 shows sketches of $\rho_0(a)$ and $\rho_1^{\rm G}(a)$. Note that $\int da\, \rho_1^{\rm G}(a) = 0$ as expected since from the definition (7.8) the normalization of $\rho(a)$, $\int da\, \rho(a) = 1$, is independent of N. Furthermore, there is no $O(1/N)$ correction to the δ-peak in $\rho_0(a)$ at $a = 0$, since this peak arises from the $N - p$ zero eigenvalues of $\mathbf{A}$ for $\alpha = p/N < 1$ and therefore has an exact height of $1 - \alpha$ for any N. The δ-peaks in $\rho_1^{\rm G}(a)$ at the spectral limits a_+ and a_- are an artefact of the truncated $1/N$ expansion: $\rho(a)$ is determined by the singularities of G as a function of λ, and the location of these singularities is only obtained correctly by resumming the full $1/N$ expansion. The δ-peaks in $\rho_1^{\rm G}(a)$ can be interpreted as 'precursors' of a broadening of the eigenvalue spectrum of $\mathbf{A}$ to values which, when the whole $1/N$ series is resummed, will lie outside the $N \to \infty$ spectral range $[a_-, a_+]$. The negative term in $\rho_1^{\rm G}(a)$ represents the corresponding 'flattening' of the eigenvalue spectrum between a_- and a_+. We can thus conclude that the average eigenvalue spectrum of $\mathbf{A}$ for finite N will be broader than for $N \to \infty$ for Gaussian inputs. Since the average response function is from (7.9) an average of the convex function $1/(\lambda + a)$ over $\rho(a)$, this broadening translates into a higher value of the response function for finite N than for $N \to \infty$, in agreement with the result $G_1^{\rm G} > 0$ found above.

Note that our prediction of a broadening of $\rho^{\rm G}(a)$ for finite N can also be confirmed by considering the extreme case $N = 1$: In this case, the 'matrix' $\mathbf{A}$ becomes the scalar sum of p Gaussian random variables with zero mean and unit variance. Hence, $\rho^{\rm G}(a)$ is just the probability density of a χ^2-distribution with p degrees of freedom, which is nonzero for all $a > 0$, i.e., over a much broader range than the spectrum $[a_-, a_+]$ predicted for $N \to \infty$.

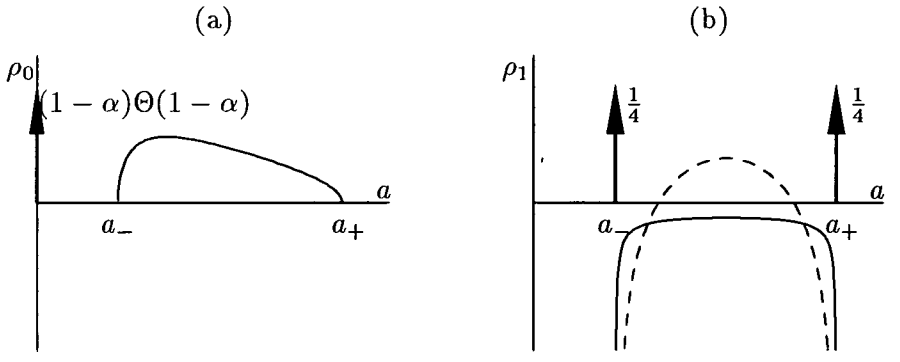

Figure 7.1. Schematic plot of the average eigenvalue spectrum $\rho(a)$ of the input correlation matrix $\mathbf{A}$. (a) Result for $N \to \infty$, $\rho_0(a)$. (b) $O(1/N)$ correction, $\rho_1(a)$, for Gaussian (solid line) and normalized (broken line) inputs. Arrows indicate δ-peaks and are labelled by the corresponding heights; in (b), the δ peaks are identical for Gaussian and for normalized inputs.

Before comparing the result (7.23) to the corresponding $O(1/N)$ correction to the average response function for *normalized* inputs, we present the results of computer simulations, performed to test our analytic predictions. For perceptron sizes between $N = 4$ and $N = 80$, we calculated the response function by direct matrix inversion, averaging over between 1200 (for $N = 80$) and 200 000 (for $N = 4$) randomly sampled sets of training inputs to obtain an 'experimental' value G^{G} of the average response function. In figure 7.2, we plot the results in the form $(G^{\mathrm{G}} - G_0)/G_1^{\mathrm{G}}$ versus $1/N$ for $\alpha = 0.5, 1, 2$ and $\lambda = 0.01, 0.1, 1$, using the results for G_0 and G_1^{G} from eqs. (7.15,7.23). The simulation results are seen to agree well with the theoretical prediction from (7.20), namely, $(G^{\mathrm{G}} - G_0)/G_1^{\mathrm{G}} = 1/N + O(1/N^2)$. The $O(1/N^2)$ terms, which correspond to corrections to G^{G} of second and higher order in $1/N$, appear as deviations from the straight line $(G^{\mathrm{G}} - G_0)/G_1^{\mathrm{G}} = 1/N$ in figure 7.2 for larger values of $1/N$. These higher-order corrections are expected to be negligible as long as $1/N \ll G_0/G_1^{\mathrm{G}}$, because this entails that the first-order correction G_1^{G}/N is already small compared to the zeroth order contribution G_0. Correspondingly, the strongest higher-order corrections in the plots in figure 7.2 are seen to occur for $\lambda = 0.01, \alpha = 1$, which has the smallest value of G_0/G_1^{G} amongst all the (λ, α) values used in the plots.

We also used the computer simulations to calculate directly the average eigenvalue spectrum of the input correlation matrix $\mathbf{A}$. Figure 7.3 shows the results for $\alpha = 10$ and $N = 4$, $N = 8$, which are based on 10^7 and 2×10^6 randomly sampled sets of

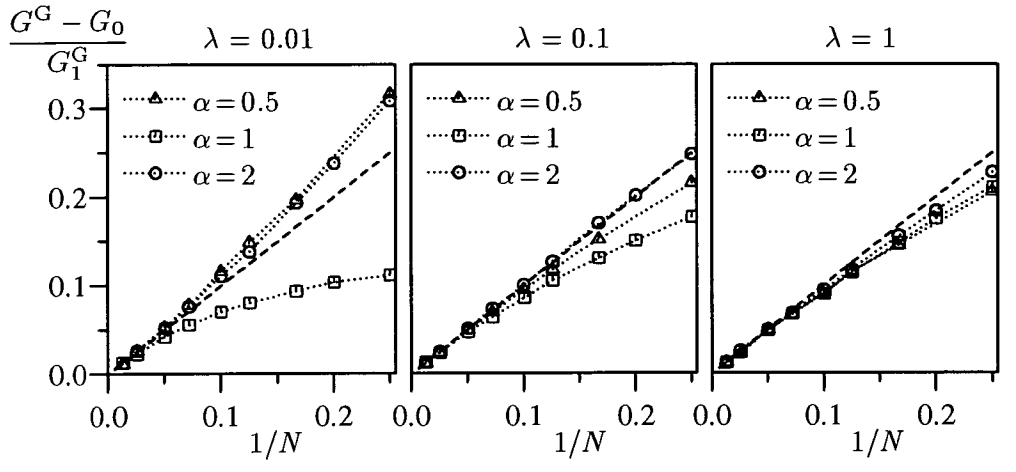

Figure 7.2. Simulation results for the average response function G^G for Gaussian inputs at finite perceptron size N, for different values of weight decay λ and (normalized) number of training examples α. The plots of $(G^G - G_0)/G_1^G$ versus $1/N$ show that as $1/N$ approaches zero, the results (symbols connected by dotted lines as a visual aid) are well approximated by $(G^G - G_0)/G_1^G = 1/N$ (broken line), in agreement with (7.20). Statistical errors due to the finite number of simulation samples are smaller than the symbol size.

training inputs, respectively. The average eigenvalue spectrum $\rho^G(a)$ was found by sorting the numerically determined eigenvalues of $\mathbf{A}$ into 100 histogram slots, evenly spaced across the spectral range shown in figure 7.3, and then applying a suitable normalization. Instead of displaying the resulting $\rho^G(a)$ directly, we plot in figure 7.3 the quantity $N(\rho^G(a) - \rho_0(a))$, which should approach $\rho_1^G(a)$ for large N from (7.24). This approach can already clearly be seen for the relatively small values of N used in our simulations. We note parenthetically that the results of more extensive computer simulations for perceptron sizes $N = 2 \ldots 10$ suggest that for any α, $N(\rho^G(a) - \rho_0(a))$ as a function of a has $2N$ turning points between a_- and a_+ (compare figure 7.3). This appears to be a signature of the 'level repulsion' in the joint probability density of the N eigenvalues of $\mathbf{A}$, which tends to zero proportionally to $|a_i - a_j|$ as two eigenvalues or 'energy levels' a_i and a_j approach each other (see, e.g., [Eat83]).

Let us now compare the above results for Gaussian inputs to the case of normalized inputs. Expanding

$$G^n = G_0 + G_1^n/N + O(1/N^2) \tag{7.26}$$

and

$$\left\langle (\Delta \mathcal{G})^2 \right\rangle^n = (\Delta^n)^2/N + O(1/N^2)$$

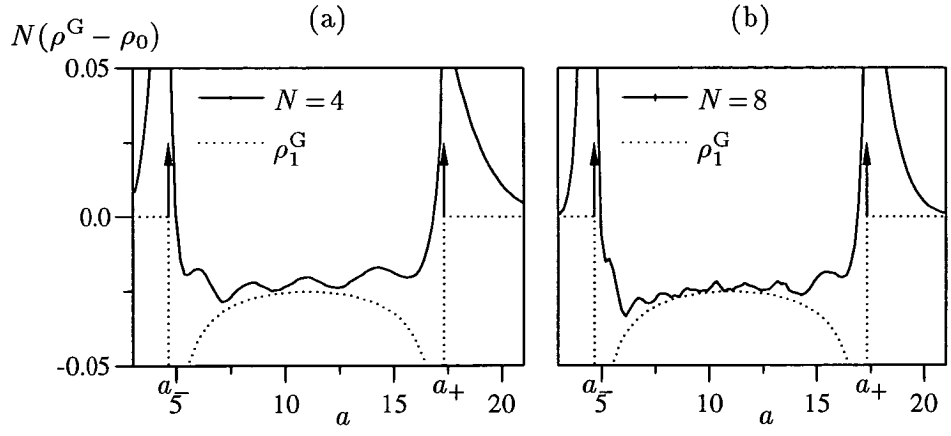

Figure 7.3. Simulation results for the average eigenvalue spectrum, $\rho^G(a)$, of the input correlation matrix $\mathbf{A}$, for the case of Gaussian inputs. The normalized number of training examples is $\alpha = 10$ and the system size is (a) $N = 4$, (b) $N = 8$. Shown is the scaled difference $N(\rho^G(a) - \rho_0(a))$ (full line), which should approach $\rho_1^G(a)$ (dotted line) for large N from (7.24). The arrows indicate the delta-peak contributions to ρ_1^G at the $N \to \infty$ spectral limits $a_\pm = (1 \pm \sqrt{\alpha})^2$ (compare equation (7.25) and figure 7.1); typical error bars for the simulation results are shown in the legend.

as before, differential equations for G_1^n and $(\Delta^n)^2$ can be obtained by the methods outlined above. The equation for $(\Delta^n)^2$ is identical to that for $(\Delta^G)^2$, yielding again the solution $(\Delta^n)^2 \equiv 0$, while G_1^n is determined by an equation analogous to (7.21), but with an extra term

$$2\frac{\partial G_0}{\partial \lambda}\frac{G_0}{(1+G_0)^2}$$

on the right hand side. The corresponding solution can be shown to be

$$G_1^n = \frac{G_0^2(1-\lambda G_0)}{1+\lambda G_0^2}\left(\frac{1}{1+\lambda G_0^2} - \frac{2}{1+G_0}\right). \tag{7.27}$$

The first term in brackets just yields G_1^G, the result for Gaussian inputs, and hence $G_1^n \leq G_1^G$ always. In fact, one can show that the ratio G_1^n/G_1^G is bounded by

$$-1 \leq \frac{G_1^n}{G_1^G} \leq 2\min\{\alpha, 1/\alpha\} - 1 \tag{7.28}$$

and decreases monotonically with λ, the lower and upper bound being attained for $\lambda \to \infty$ and $\lambda \to 0$, respectively. This behaviour is shown in figure 7.4, which also illustrates that G_1^n can be both positive and negative (since G_1^G is always positive). The reason for this can be understood by looking at the $O(1/N)$ correction to $\rho(a)$.

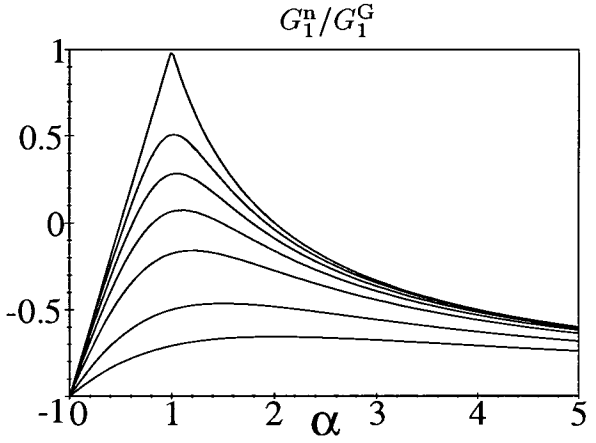

Figure 7.4. Ratio G_1^n/G_1^G of $O(1/N)$ corrections to average response function for normalized and Gaussian inputs, as a function of α for fixed weight decay λ. From top to bottom, the values of λ are: $\lambda \to 0$, $\lambda = 0.02, 0.05, 0.1, 0.2, 0.5, 1.0$.

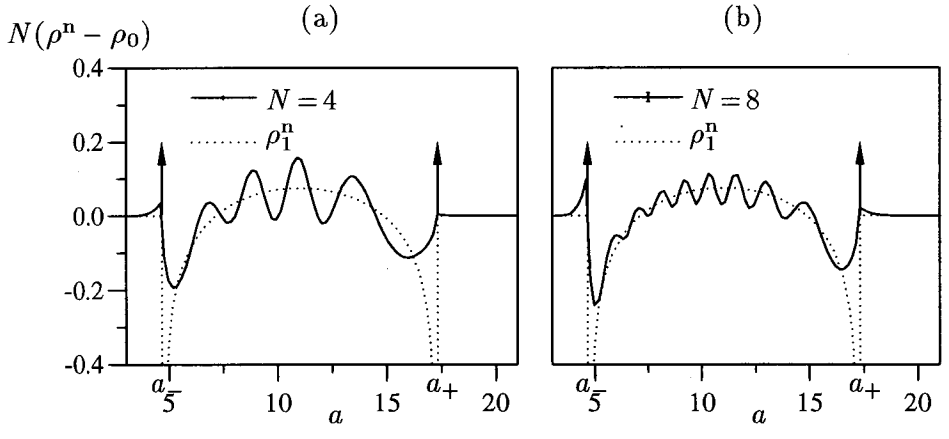

Figure 7.5. Analogue of figure 7.3 for the case of normalized inputs.

Writing

$$\rho^n(a) = \rho_0(a) + \rho_1^n(a)/N + O(1/N^2) \tag{7.29}$$

one finds from (7.10, 7.27) that

$$\rho_1^n(a) = \rho_1^G(a) + \frac{1}{\pi\alpha}\sqrt{(a_+ - a)(a - a_-)} - \frac{2}{\pi}\frac{1}{\sqrt{(a_+ - a)(a - a_-)}} \tag{7.30}$$

as illustrated in Figure 7.1 and in agreement with the simulation results shown in Figure 7.5. The difference $\rho_1^n - \rho_1^G$, which integrates to zero over the thermodynamic limit

spectrum $[a_-, a_+]$, is positive in the middle of this interval and negative at the edges, representing a shift of spectral weight towards the center of the spectrum. In $\rho_1^{\rm n}$, this counteracts the broadening of the spectrum contributed by $\rho_1^{\rm G}$. Depending on which of the two effects is stronger, $G_1^{\rm n}$ can therefore have either positive or negative sign, in agreement with our above results. This of course implies that $G_1^{\rm n}$ can become 'accidentally' zero at nonzero α (compare figure 7.4), in which case it cannot be expected to give a good estimate of the size of finite N corrections to the average response function since corrections of order $1/N^2$ and higher may become significant. This is illustrated in figure 7.6, where the linear extrapolations in $1/N$ from the thermodynamic limit result do not always give a good estimate of the deviation of $G^{\rm n}$ from G_0 for, say, $N = 4$ (see for example the case $\alpha = 1$, $\lambda = 0.1$). However, the result $|G_1^{\rm n}| \leq |G_1^{\rm G}|$ shown above suggests that the size of the finite size corrections for normalized inputs should be (at least approximately) bounded by the corrections for Gaussian inputs, and inspection of figure 7.6 shows that this is indeed the case. Below, when we estimate the critical system size for validity of the thermodynamic limit analysis for *normalized* inputs, we shall therefore in fact use the results for *Gaussian* inputs to obtain an upper bound.

First, however, we examine more closely the finite size corrections to the average generalization error for learning from random examples and the efficacy of query learning implied by the $O(1/N)$ corrections for the response function derived above. From the $1/N$ expansions (7.20, 7.26) of G we obtain a corresponding expansion of the average generalization error, which we write in the form

$$\epsilon_{\rm g}(\text{random examples}) = \epsilon_{\rm g,0} + \epsilon_{\rm g,1}/N + O(1/N^2). \tag{7.31}$$

From (7.5), the explicit expressions for $\epsilon_{\rm g,0}$ and $\epsilon_{\rm g,1}$ are

$$\epsilon_{\rm g,0} = \frac{1}{2}\left[\sigma^2 G_0 + \lambda(\sigma^2 - \lambda)\frac{\partial G_0}{\partial \lambda}\right], \quad \epsilon_{\rm g,1} = \frac{1}{2}\left[\sigma^2 G_1 + \lambda(\sigma^2 - \lambda)\frac{\partial G_1}{\partial \lambda}\right].$$

From (7.31) one has a corresponding expansion of the improvement in generalization performance due to query learning, κ, as defined in eq. (7.7):

$$\kappa = \kappa_0 + \kappa_1/N + O(1/N^2).$$

Since the average generalization error achieved by minimum entropy queries is independent of the system size N, the expansion coefficients are simply

$$\kappa_0 = \frac{\epsilon_{\rm g,0}}{\epsilon_{\rm g}(\text{queries})} \qquad \kappa_1 = \frac{\epsilon_{\rm g,1}}{\epsilon_{\rm g}(\text{queries})}.$$

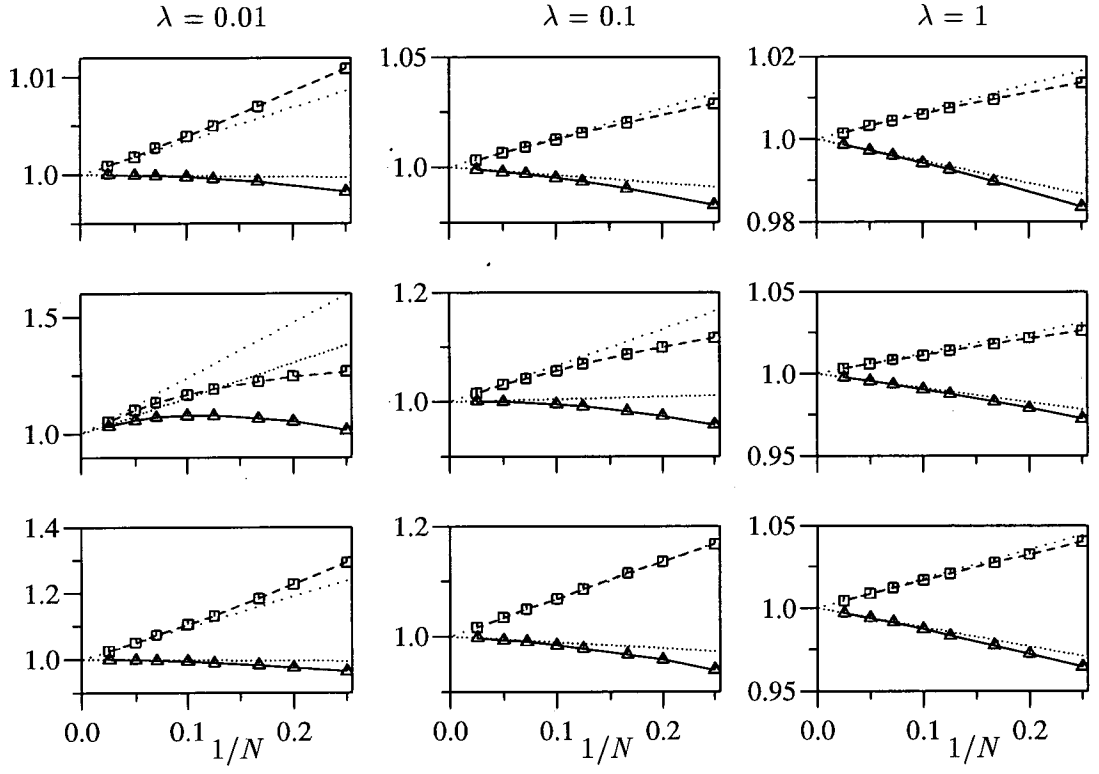

Figure 7.6. Comparison of finite size effects for G for Gaussian (squares) and normalized (triangles) inputs, from simulations. Shown is the ratio of the average response function G and the thermodynamic limit value G_0. The dotted lines indicate the theoretical result up to $O(1/N)$, $1 + G_1/NG_0$, with G_1 given by (7.23) and (7.27), respectively. Top to bottom: $\alpha = 0.5, 1, 2$; left to right: $\lambda = 0.01, 0.1, 1$. Error bars are smaller than the symbol size.

The relative size of the $O(1/N)$ correction is therefore the same for ϵ_g(random examples) and κ,

$$\frac{\kappa_1}{\kappa_0} = \frac{\epsilon_{g,1}}{\epsilon_{g,0}}.$$

The thermodynamic limit value $\epsilon_{g,0}$ is independent of whether Gaussian or normalized inputs are considered, while $\epsilon_{g,1}$ assumes different values in the two cases[5] arising from the corresponding response function corrections G_1^G and G_1^n. Figure 7.7 shows plots of $\epsilon_{g,0}$ together with the relative correction $\epsilon_{g,1}/\epsilon_{g,0}$ for the case of Gaussian and normalized inputs, for teacher noise level $\sigma^2 = 0$ and $\sigma^2 = 0.5$. The graphs suggest that the modulus of the relative correction is largest when λ is small and α is close to 1. This is in fact not surprising; as has been pointed out by several authors (see, e.g., [HKT89, LKS91, BS94, Kro92, KH92a]), a *phase transition* in α-λ space takes place at the point $\alpha = 1$, $\lambda = 0$, signalled by a critical slowing down of the learning dynamics [HKT89, LKS91], divergences of generalized susceptibilities [BS94, MS95] etc. It is therefore only reasonable to expect that finite size effects should be largest in this region.

This expectation can also confirmed by an estimation of the critical system size N_c. N_c should be defined such that the thermodynamic limit predictions for ϵ_g(random examples) and κ (for the case of normalized inputs) are valid to a good approximation for system sizes $N \gg N_c$. This suggests the definition

$$N_c = \max\left\{\frac{|\epsilon_{g,1}^G|}{\epsilon_{g,0}}, \frac{|\epsilon_{g,1}^n|}{\epsilon_{g,0}}\right\} \tag{7.32}$$

which ensures that $|\epsilon_{g,1}/N| \ll \epsilon_{g,0}$ and $|\kappa_1/N| \ll \kappa_0$ for $N \gg N_c$. As argued above, taking the maximum over the cases of Gaussian and normalized inputs ensures a reasonable estimate for N_c even in the regime where G_1^n is 'accidentally' close to zero. In principle, N_c depends on the number of training examples α, the weight decay λ and the noise level σ^2. We confine ourselves to bounding N_c from above by maximizing over σ^2 (which, in an experimental setting, is beyond our control anyway), replacing (7.32) by

$$N_c = \max_{\sigma^2} \max\left\{\frac{|\epsilon_{g,1}^G|}{\epsilon_{g,0}}, \frac{|\epsilon_{g,1}^n|}{\epsilon_{g,0}}\right\}.$$

The second bounding operation is easily performed since $|\epsilon_{g,1}/\epsilon_{g,0}|$ attains its maximum w.r.t. σ^2 either at $\sigma^2 = 0$ or for $\sigma^2 \to \infty$, due to the monotonicity of $\epsilon_{g,1}/\epsilon_{g,0}$ as a

[5] Note that unlike ϵ_g, κ is not defined for Gaussian inputs since we have only calculated ϵ_g(queries) for the case of normalized inputs; Gaussian inputs would lead to problems in the definition of query selection because the optimal queries would be input vectors of unbounded length.

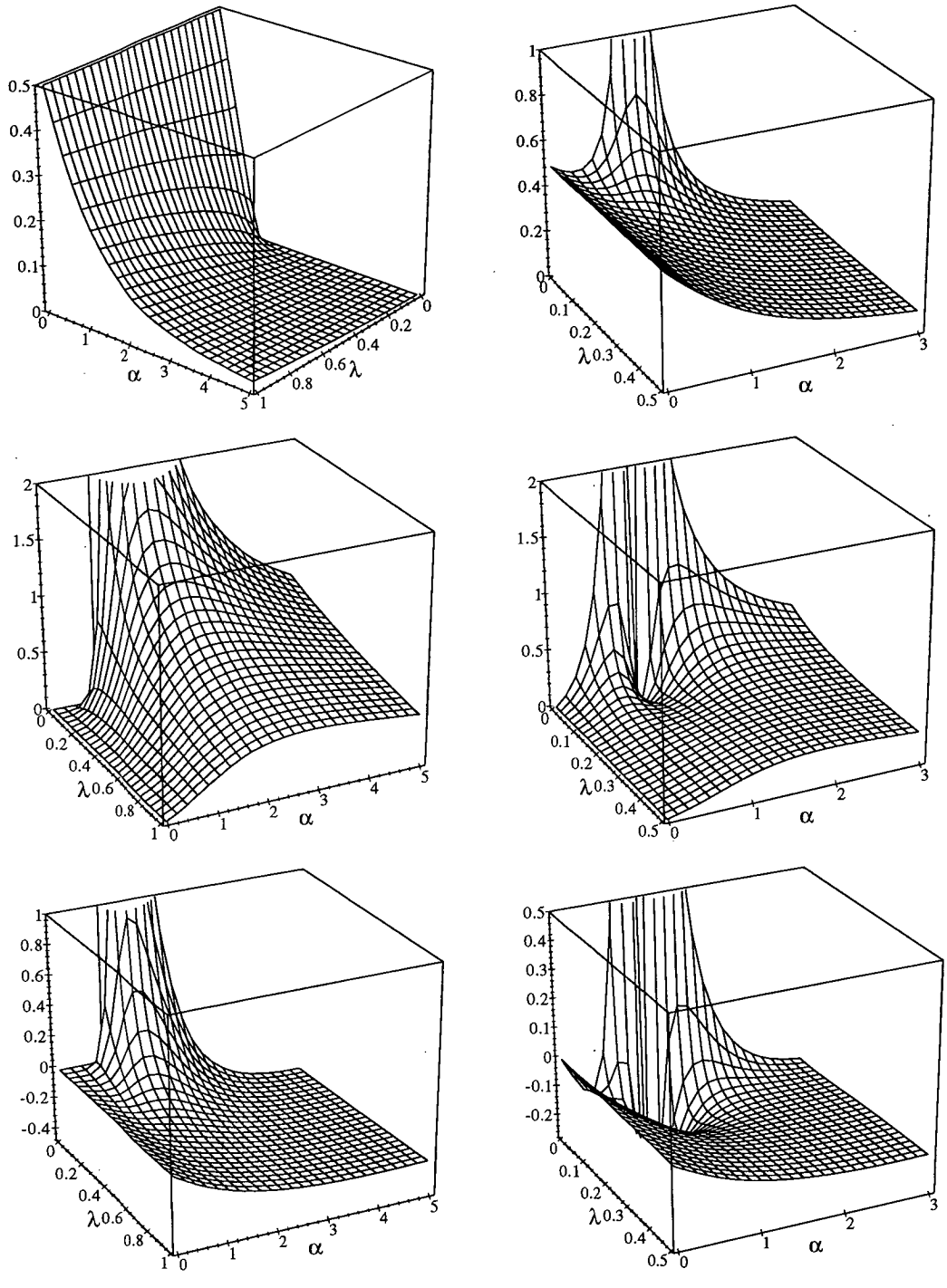

Figure 7.7. Average generalization error for random examples: Thermodynamic limit results and $O(1/N)$ corrections. Top row: Average generalization error $\epsilon_{g,0}$ for $N \to \infty$. Middle row: Relative size $\epsilon^{G}_{g,1}/\epsilon_{g,0}$ of $O(1/N)$ correction (with factor $1/N$ suppressed) for Gaussian inputs. Bottom row: $O(1/N)$ correction $\epsilon^{n}_{g,1}/\epsilon_{g,0}$ for normalized inputs. All quantities are shown as functions of λ and α, for teacher noise level $\sigma^2 = 0$ (left column) and $\sigma^2 = 0.5$ (right column); for visual purposes, the top left graph is slightly rotated compared to the others. Note that finite size corrections are generally largest near $\lambda = 0$, $\alpha = 1$.

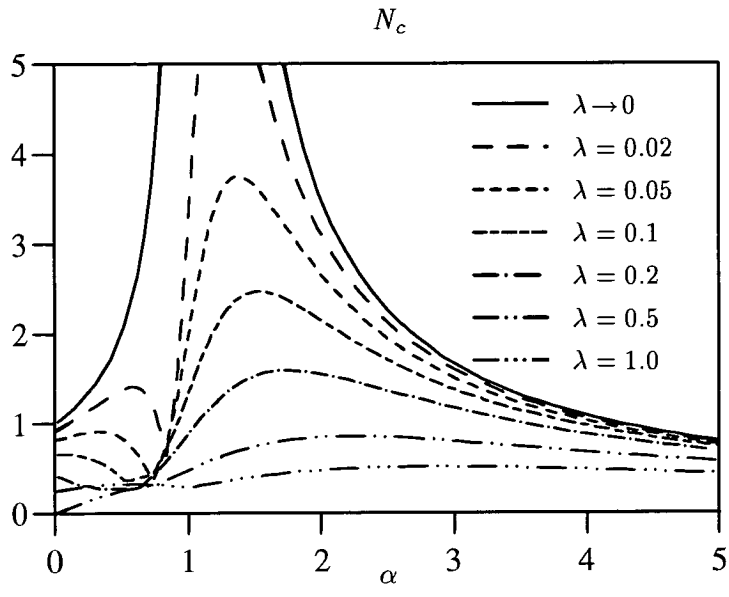

Figure 7.8. Critical perceptron size N_c: For $N \gg N_c$, the results for ϵ_g(random examples) and κ obtained in the thermodynamic limit are valid to a good approximation, for any noise level σ^2. Note that the maximum of N_c w.r.t. λ is obtained for $\lambda \to 0$.

function of σ^2. The above definition of N_c also a provides an additional safeguard against 'accidentally' small values of $|\epsilon_{g,1}|$ which can occur for particular combinations of α, λ and σ^2 (see figure 7.7). We plot the resulting α dependence of N_c in figure 7.8 for several values of λ. N_c is maximal for $\lambda \to 0$; evaluating this limit, we obtain[6]

$$N_c \leq N_c(\alpha) = \begin{cases} 1/(1-\alpha) & \text{for } 0 < \alpha < 1 \\ (3\alpha+1)/[\alpha(\alpha-1)] & \text{for } \alpha > 1. \end{cases} \tag{7.33}$$

Therefore, results for ϵ_g(random examples) and κ derived in the thermodynamic limit will be valid for any λ and σ^2 provided that $N \gg N_c(\alpha)$. For large α, $N_c(\alpha) = 3/\alpha + O(1/\alpha^2)$, and the condition $N \gg N_c(\alpha)$ will easily be fulfilled. For finite λ and near $\alpha = 1$, the bound (7.33) is unnecessarily pessimistic, as figure 7.8 shows. To remedy this, we have verified numerically that for $\lambda > 2$, N_c attains its maximum

[6] Note that N_c is discontinuous at $\alpha = 0$: its limit as $\alpha \to 0$ is in general nonzero, whereas at $\alpha = 0$, where $\epsilon_{g,1}$ vanishes, it is exactly zero.

w.r.t. α for $\alpha \to 0$. From this one can derive an alternative bound for N_c,

$$N_c \leq \frac{2\lambda - 1}{(\lambda + 1)^2} \qquad \text{for } \lambda > 2$$

which is independent of α and σ^2 and will be tighter than (7.33) near $\alpha = 1$ and for sufficiently large λ.

7.5 Summary and discussion

In this chapter, we have analysed finite size effects on the efficacy of query learning, in a learning scenario with linear students learning from a noisy linear teacher. In order to calculate finite size effects to first order in $1/N$, we have presented a new method, based on simple matrix identities, for calculating the average response function G. This function determines most of the properties of learning and generalization in linear perceptrons. In the thermodynamic limit, $N \to \infty$, we have recovered the known result for G and have also shown explicitly that the response function is self-averaging. The versatility of our method has been demonstrated by using it to extend the thermodynamic limit analysis to more general learning scenarios. We have calculated the $O(1/N)$ correction to G for the case of both Gaussian and normalized training inputs, and found good agreement with the results of computer simulations in both cases. The effect of finite system size on $\rho(a)$, the average eigenvalue spectrum of the input correlation matrix, has also been derived, confirmed by simulations, and used to interpret the response function results. Finally, the $O(1/N)$ corrections to the average generalization error for learning from random examples and the corresponding correction to the efficacy of query learning (for normalized inputs) has been obtained. We have used these results to estimate quantitatively how large the system size N has to be for the results obtained in the thermodynamic limit to be valid. The corresponding critical system size N_c was found to be of order unity except in the region around $\alpha = 1$, $\lambda = 0$, where a phase transition in the learning and generalization behaviour takes place. This is also the region where strong overfitting occurs if the teacher is noisy, leading to a large value of the generalization error. For other values of α and λ – where successful generalization takes place – we can conclude from the smallness of the critical system size that the thermodynamic limit results are directly relevant for real-world system sizes of the order of a few tens or hundreds. The misgivings that have often been expressed by non-physicists about the applicability of thermodynamic limit results to practical learning scenarios therefore seem to be unfounded; the system size certainly does not have to be as large as for the physical systems typically studied with the help

of statistical mechanics ($\approx 10^{23}$) in order for the thermodynamic limit to yield valid predictions.

Parenthetically, we remark that the $O(1/N)$ corrections that we have calculated in this chapter can also be used in different contexts. For example, the generalization error can be estimated by the test error, obtained by comparing the outputs of student and teacher on a finite number of randomly chosen test inputs. Using our results, test error fluctuations can be analysed, and an optimal test set size can be derived for the case where the total number of training and test examples is limited [BSS95]. Another application is in an analysis of the evidence procedure in Bayesian inference for finite N, where optimal values of 'hyperparameters' like the weight decay parameter λ are determined on the basis of the training data [MS95]. Finally, the methods presented in this chapter can also be used to analyse finite size effects in on-line learning [BSS]. We therefore hope that our results will be able to provide the basis for a systematic investigation of finite size effects in learning and generalization.

7.6 Appendix: The method of characteristic curves

In this appendix, we briefly describe the method of characteristic curves for the solution of partial differential equations, following the exposition in [Joh78]. We then apply the method to obtain the solutions of the differential equations for the various response functions introduced in this chapter.

Consider the following quasi-linear first-order partial differential equation for $f(x, y)$,

$$a \frac{\partial f}{\partial x} + b \frac{\partial f}{\partial y} - c = 0 \tag{7.34}$$

where a, b, and c are functions of x, y, and f. The solution $f = f(x, y)$ can be thought of as a surface in (x, y, f) space, which has normal vectors proportional to $(\partial f/\partial x, \partial f/\partial y, -1)$. Equation (7.34) can then be interpreted as defining a vector field (a, b, c) of 'characteristic directions', which are orthogonal to the normal vectors of the solution surface. This suggests that any curve starting at a point within the solution surface remains within that surface if it follows the characteristic direction at every point. Formally, such 'characteristic curves' are defined by the requirement $d(x, y, f)/dt = (a, b, c)$, where t parameterizes the points along the curve. It can be shown rigorously [Joh78] that the solution surface is indeed given by the union of all characteristic curves which pass through a one-parameter family of points defining the initial conditions for $f(x, y)$.

Consider now equation (7.14) for the average response function G in the thermodynamic limit. The characteristic curves are the solutions of

$$\frac{d\alpha}{dt} = 1, \quad \frac{d\lambda}{dt} = -\frac{1}{1+G}, \quad \frac{dG}{dt} = 0$$

which are given by

$$\alpha = \alpha_0 + t, \quad \lambda = \lambda_0 - \frac{t}{1+G_0}, \quad G = G_0. \tag{7.35}$$

The initial condition $G|_{\alpha=0} = 1/\lambda$ selects the characteristic curves with $\alpha_0 = 0$, λ_0 arbitrary, $G_0 = 1/\lambda_0$. Inserting this into (7.35), one can eliminate α_0, λ_0, G_0 and t to obtain $1/G = \lambda + \alpha/(1+G)$. This yields the solution (7.15) for $G(\alpha, \lambda)$.

We now turn to equation (7.17) for the modified response function $G_L = \langle \mathcal{G}_L \rangle = \left\langle \frac{1}{N} \mathrm{tr}\, (\mathbf{L}+\mathbf{A})^{-1} \right\rangle$. To obtain this result, one first replaces the matrix $\mathbf{L}$ by $\mathbf{L} + \lambda \mathbf{1}$. The recursion relation (7.12) between $\mathcal{G}(p+1)$ and $\mathcal{G}(p)$ remains valid for $\mathcal{G}_L$, and results, in the thermodynamic limit, in a differential equation for G_L exactly analogous to (7.14), with G replaced by G_L. The corresponding characteristic curves are the same as in (7.35), but the initial condition $G_L|_{\alpha=0} = \frac{1}{N} \mathrm{tr}\, (\mathbf{L} + \lambda \mathbf{1})^{-1}$ now selects a different subset of these characteristic curves. This leads to the equation $G_L = \frac{1}{N} \mathrm{tr}\, [\mathbf{L} + (\lambda + \alpha/(1+G_L))\mathbf{1}]^{-1}$; from which (7.17) is obtained by setting $\lambda = 0$.

The solution for the general modified response function $G_{BL} = \left\langle \frac{1}{N} \mathrm{tr}\, \mathbf{B}(\mathbf{L}+\mathbf{A})^{-1} \right\rangle$ given in (7.18) is obtained as follows: First, one again replaces the matrix $\mathbf{L}$ by $\mathbf{L} + \lambda \mathbf{1}$. Multiplying the matrix equation (7.11) by $\mathbf{B}$ and taking the trace, one can follow the procedure described in Section 7.2 to obtain, in the thermodynamic limit, the differential equation

$$\frac{\partial G_{BL}}{\partial \alpha} - \frac{\partial G_{BL}}{\partial \lambda} \frac{1}{1+G_L} = 0.$$

Since G_L is a fairly complicated function of α and λ, the corresponding characteristic equations

$$\frac{d\alpha}{dt} = 1, \quad \frac{d\lambda}{dt} = -\frac{1}{1+G_L(\alpha,\lambda)}, \quad \frac{dG_{BL}}{dt} = 0$$

might seem hard to solve. However, G_L is in fact constant along the characteristic curves: As pointed out above, G_L obeys equation (7.14) (with G replaced by G_L), and hence

$$\frac{dG_L}{dt} = \frac{d\alpha}{dt}\frac{\partial G_L}{\partial \alpha} + \frac{d\lambda}{dt}\frac{\partial G_L}{\partial \lambda} = \frac{\partial G_L}{\partial \alpha} - \frac{1}{1+G_L}\frac{\partial G_L}{\partial \lambda} = 0.$$

Therefore, the characteristic curves are

$$\alpha = \alpha_0 + t, \quad \lambda = \lambda_0 - \frac{t}{1+G_L}, \quad G_{BL} = \text{const.}$$

Together with the initial condition $G_{BL}|_{\alpha=0} = \frac{1}{N}\text{tr}\,\mathbf{B}(\mathbf{L}+\lambda\mathbf{1})^{-1}$, this yields $G_{BL}|_{\alpha=0} = \frac{1}{N}\text{tr}\,\mathbf{B}[\mathbf{L}+(\lambda+\alpha/(1+G_L))\mathbf{1}]^{-1}$. Equation (7.18) is recovered for $\lambda = 0$.

Finally, we consider the solution of (7.21) and (7.22), dropping the superscript 'G' for brevity. One first verifies that $\Delta^2 \equiv 0$ satisfies (7.22) and the corresponding initial conditions; of course, this solution can also be obtained using the method of characteristic curves. One can then simplify (7.21) by inserting $\Delta^2 = 0$ and by using the fact that G_0, the value of G in the thermodynamic limit, obeys (7.14) (with G replaced by G_0). After some algebra, one obtains

$$\frac{\partial G_1}{\partial \alpha} - \frac{\partial G_1}{\partial \lambda}\frac{1}{1+G_0} = \frac{1}{2}G_0'' - G_1\frac{G_0'}{1+G_0}. \tag{7.36}$$

Here we have introduced the abbreviations $G_0' = \partial G_0/\partial\alpha$, $G_0'' = \partial^2 G_0/\partial\alpha^2$. By the same reasoning as above, one can show that G_0 is constant along the characteristic curves of (7.36). The characteristic curves obeying the initial condition $G_1|_{\alpha=0} = 0$ are therefore given by

$$\alpha = t, \quad \lambda = \lambda_0 - \frac{t}{1+G_0}, \quad \frac{dG_1}{dt} = \frac{1}{2}G_0'' - G_1\frac{G_0'}{1+G_0}$$

with $G_1(t=0) = 0$. The constant value of G_0 along a characteristic curve is related to λ_0 by $G_0 = G_0(t=0) = G_0|_{\lambda=\lambda_0,\alpha=0} = 1/\lambda_0$. Using the explicit form of $G_0(\alpha,\lambda)$ given in eq. (7.15), both G_0' and G_0'' can be expressed as functions of G_0 and λ alone as follows:

$$G_0' = -\frac{1}{\lambda + 1/G_0^2}, \quad G_0'' = \frac{2/G_0^3}{(\lambda+1/G_0^2)^3}.$$

This finally leads to the following linear differential equation for G_1 as a function of λ along a characteristic curve with a given value of G_0:

$$\frac{dG_1}{d\lambda} = -(1+G_0)\frac{dG_1}{dt} = -\frac{(1+G_0)/G_0^3}{(\lambda+1/G_0^2)^3} - \frac{G_1}{\lambda+1/G_0^2}. \tag{7.37}$$

Since $\lambda = \lambda_0 = 1/G_0$ at $t = 0$, the initial condition is $G_1(\lambda = 1/G_0) = 0$. The integration of (7.37) is straightforward and yields directly the solution (7.23) given in the text.

Chapter 8

Towards realistic neural networks II: Multi-layer networks

Abstract

In this chapter, we provide an exact average case analysis of query learning for maximum information gain in a multi-layer network. In particular, we consider a large tree-committee machine (TCM) trained on noise free training data produced by a TCM of the same architecture. Our results show that the generalization error decreases exponentially with the number of training examples, providing a significant improvement over the slow algebraic decay for random examples. The results are compared to those obtained previously for query learning in the binary perceptron, and the resulting implications for the connection between information gain and generalization error in multi-layer networks are discussed. We conclude by suggesting a computationally cheap algorithm for constructing approximate maximum information gain queries, which can be extended to more complicated multi-layer networks and which in our analysis shows performance even slightly superior to exact maximum information gain queries.

8.1 Introduction

In this chapter, we continue our investigation of query learning for maximum information gain (or, equivalently, minimum entropy). The generalization performance achieved by maximum information gain queries has been analysed in previous chapters for single-layer neural networks such as linear and binary perceptrons (see also [SOS92, FSST93]). For multi-layer networks, which are much more widely used in practical applications, several heuristic algorithms for query learning have been proposed (see

e.g., [Bau91, HCOM91]). While such heuristic approaches can demonstrate the power of query learning, they are hard to generalize to situations other than the ones for which they have been designed. Furthermore, the existing analyses of such algorithms have been carried out within the framework of 'probably approximately correct' (PAC) learning, yielding worst case results which are not necessarily close to the potentially more relevant average case results.

In this chapter the powerful tools of statistical mechanics are used to analyse the average generalization performance achieved by query learning in a multi-layer network. This is the first quantitative analysis of its kind that we are aware of. In particular, we consider query learning in a large *tree-committee machine* (TCM), with noise free training data generated by a teacher network of the same architecture. The details of the model are explained in the next section. In Section 8.3, we then outline the calculation of the main quantity of interest, the average generalization error ϵ_g as a function of the (normalized) number of training examples, α. The results are compared to existing analyses of learning from random examples in a TCM and related to corresponding results for the binary perceptron. We also discuss the relationship between information gain and generalization error in the TCM. In Section 8.4, we analyse a computationally cheap algorithm for constructing approximate maximum information gain queries, and find that it achieves generalization performance even slightly superior to that of exact maximum information gain queries. In Section 8.5, we summarize and discuss our results and offer our conclusions regarding the potential for practical applications of query learning in multi-layer neural networks.

8.2 The model

8.2.1 Tree-committee machine

A tree-committee machine (TCM) is a two-layer neural network with N input units, K binary hidden units and one binary output unit. The 'receptive fields' of the individual hidden units do not overlap, and each hidden units calculates the sign of a linear combination (with real coefficients) of the N/K input components to which it is connected. The output unit then calculates the sign of the sum of all the hidden unit outputs. A TCM therefore effectively has all the weights from the hidden to the output layer fixed to one. Formally, the output y for a given input vector $\mathbf{x}$ is

$$y = f(\mathbf{x}) = \mathrm{sgn}\left(\frac{1}{\sqrt{K}}\sum_{i=1}^{K}\sigma_i\right) \qquad \sigma_i = \mathrm{sgn}\left(\sqrt{\frac{K}{N}}\mathbf{x}_i^{\mathrm{T}}\mathbf{w}_i\right) \tag{8.1}$$

where the σ_i are the outputs of the hidden units, the $\mathbf{w}_i \in \mathbb{R}^{N/K}$ are their weight vectors, and we have decomposed the input vector $\mathbf{x}^{\mathrm{T}} = (\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_K^{\mathrm{T}})$ into the vectors $\mathbf{x}_i \in \mathbb{R}^{N/K}$ containing the N/K inputs to which hidden unit i is connected[1]. The N components of the K hidden unit weight vectors $\mathbf{w}_i$, which we denote collectively by $\mathbf{w}$, form the adjustable parameters of a TCM. Without loss of generality, we assume the weight vectors to be normalized to $\mathbf{w}_i^2 = N/K$. This ensures that typical individual weights are roughly of order one and that the arguments of the sgn-functions in (8.1) are of order unity for typical hidden unit input vectors $\mathbf{x}_i$ of length $\mathbf{x}_i^2 = N/K$. We shall restrict our analysis to the case where both the input space dimension and the number of hidden units are large ($N \to \infty$, $K \to \infty$), assuming that each hidden unit is connected to a large number of inputs, i.e., $N/K \gg 1$. The $K \to \infty$ limit is chosen because it is most likely to yield qualitatively new results compared to the case of the binary perceptron ($K = 1$).

As our training algorithm we take (zero temperature) Gibbs learning, which generates at random any TCM (in the following referred to as a 'student', as usual) which predicts all the training outputs in a given set of training examples $\Theta^{(p)} = \{(\mathbf{x}^\mu, y^\mu), \mu = 1 \ldots p\}$ correctly[2]. As usual, the number of training examples p is taken to be proportional to N, $p = \alpha N$, in order to ensure the existence of a well-defined thermodynamic limit. We take the problem to be perfectly learnable, which means that the outputs y^μ corresponding to the inputs $\mathbf{x}^\mu$ are generated by a teacher TCM, $\mathcal{V}$, with the same architecture as the student but with different, unknown weights $\mathbf{w}_\mathcal{V}$. It is further assumed that there is no noise on the training examples. For learning from *random examples*, the training inputs $\mathbf{x}^\mu$ are sampled randomly from a distribution $P(\mathbf{x})$. Since the output (8.1) of a TCM is independent of the length of the hidden unit input vectors $\mathbf{x}_i$, we assume this distribution $P(\mathbf{x})$ to be uniform over all vectors $\mathbf{x}^{\mathrm{T}} = (\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_K^{\mathrm{T}})$ which obey the spherical constraints $\mathbf{x}_i^2 = N/K$. For *query learning*, the training inputs $\mathbf{x}^\mu$ are chosen to maximize the expected information gain of the student, as explained in the next section.

At this point we would like to remind the reader that in order to achieve optimal

[1] We assume that K is odd in order to avoid having to define the output y for the case when the hidden unit outputs σ_i sum to zero.

[2] In the terminology of Chapter 5, this corresponds to a post-training student distribution $P(\mathcal{N}|\Theta^{(p)})$ of posterior-like form (5.4) with an assumed spin-flip noise level $p_\mathcal{N} = 0$ (see eqs. (5.16,5.17)) and a pseudo-prior $\tilde{P}(\mathcal{N})$ which is uniform over all student TCMs satisfying the constraints $\mathbf{w}_i^2 = N/K$. In the following, we simply denote student weight vectors by $\mathbf{w}$, dropping the subscript '$\mathcal{N}$' for the sake of brevity. As in previous chapters, we also abuse the notation by identifying weight vectors with the students or teachers that they represent, so that for example $P(\mathbf{w}|\Theta^{(p)}) \equiv P(\mathcal{N}|\Theta^{(p)})$ and $P(\mathbf{w}_\mathcal{V}|\Theta^{(p)}) \equiv P(\mathcal{V}|\Theta^{(p)})$.

generalization performance, the theoretically optimal choice of queries would of course be based on direct minimization of the generalization error, and not on maximization of the information gain. However, the generalization error as an objective function for query selection is in general not easy to calculate, while the expected information gain of a query can often be determined fairly easily. Since information gain and decrease in generalization error are normally correlated (cf. the results in Chapter 3), maximizing information gain therefore provides a practical method for achieving near-optimal generalization performance by query learning.

8.2.2 Maximum information gain queries

The definition of maximum information gain (i.e., minimum entropy queries) for TCM students follows the general framework described in Section 5.4, the relevant features of which we briefly review below. We assume a teacher space prior $P(\mathcal{V})$ which is, like the student pseudo-prior, uniform over all TCMs with weight vectors satisfying $\mathbf{w}_i^2 = N/K$. Since the student correctly assumes that there is no noise on the training examples, we are thus dealing with the case of a correct inference model. Minimum entropy queries as defined in Chapter 5, selected assuming the inference model is correct, are therefore identical to true minimum entropy queries according to the general definition in Chapter 2.

Information gain is defined as the decrease in the entropy S in the parameter space of the student. The entropy for a given training set $\Theta^{(p)}$ is given by

$$S(\Theta^{(p)}) = -\int d\mathbf{w} P(\mathbf{w}|\Theta^{(p)}) \ln P(\mathbf{w}|\Theta^{(p)}). \tag{8.2}$$

For the Gibbs learning algorithm considered here, $P(\mathbf{w}|\Theta^{(p)})$ is uniform on the 'version space', the space of all students which predict all training outputs correctly (and satisfy $\mathbf{w}_i^2 = N/K$), and zero otherwise. Denoting the version space volume by $V(\Theta^{(p)})$, the entropy can thus simply be written as $S(\Theta^{(p)}) = \ln V(\Theta^{(p)})$. When a new training example $(\mathbf{x}^{p+1}, y^{p+1})$ is added to the existing training set, the information gain is $I = S(\Theta^{(p)}) - S(\Theta^{(p+1)})$. Since the new training output y^{p+1} is unknown, only the *expected* information gain, obtained by averaging over y^{p+1}, is available for selecting a maximally informative query $\mathbf{x}^{p+1}$. The probability of obtaining output $y^{p+1} = \pm 1$ given input $\mathbf{x}^{p+1}$ is simply $v^{\pm} = V(\Theta^{(p+1)})|_{y^{p+1}=\pm 1}/V(\Theta^{(p)})$, the fraction of the version space left over after the new example $(\mathbf{x}^{p+1}, y^{p+1} = \pm 1)$ has been added [SOS92]. This follows from the fact that the posterior teacher distribution $P(\mathcal{V}|\Theta^{(p)})$ is, like the student distribution $P(\mathcal{N}|\Theta^{(p)})$, uniform on the version space and zero elsewhere. The

expected information gain is therefore

$$\langle I \rangle_{P(y^{p+1}|\mathbf{x}^{p+1},\Theta^{(p)})} = -v^+ \ln v^+ - v^- \ln v^- \tag{8.3}$$

in agreement with the general result (5.18). It attains its maximum value $\ln 2$ ($\equiv 1$ bit) when $v^\pm = \frac{1}{2}$, i.e., when the new input $\mathbf{x}^{p+1}$ *bisects* the existing version space. This is intuitively reasonable, since $v^\pm = \frac{1}{2}$ corresponds to maximum uncertainty about the new output and hence to maximum information gain once this output is known.

Due to the complex geometry of the version space, the generation of queries which achieve exact bisection is in general computationally infeasible. The 'query by committee' algorithm [SOS92] provides a solution to this problem by first sampling a 'committee' of $2k$ students[3] from the Gibbs distribution $P(\mathbf{w}|\Theta^{(p)})$ and then using the fraction of committee members which predict $+1$ or -1 for the output y corresponding to an input $\mathbf{x}$ as an approximation to the true probability $P(y = \pm 1|\mathbf{x}, \Theta^{(p)}) = v^\pm$. The condition $v^\pm = \frac{1}{2}$ is then approximated by the requirement that exactly k of the committee members predict output $+1$ and the others -1 for the new training input $\mathbf{x}^{p+1}$. This corresponds to the requirement of 'maximal disagreement' between committee members. An approximate minimum entropy query can thus be found by sampling (or *filtering*) inputs from a stream of random inputs until this condition is met. The procedure is then repeated for each new query. As $k \to \infty$, this algorithm approaches exact bisection, and we focus on this limit in the following. Based on the results for the binary perceptron [SOS92], however, we would expect the results to be qualitatively similar for finite k.

8.3 Exact maximum information gain queries

The main quantity of interest in our analysis is the generalization error ϵ_g, defined as the probability that a given student TCM will predict the output of the teacher incorrectly for a random test input sampled from $P(\mathbf{x})$. In the thermodynamic limit $N \to \infty$, $K \to \infty$, $N/K \gg 1$ that we consider here it can be expressed in the form [SH92] (see also Appendix 8.6.3)

$$\epsilon_g = \frac{1}{\pi} \arccos R_{\text{eff}} \tag{8.4}$$

[3] Each of these students is, of course, itself a TCM - the number of committee members $2k$ should therefore not be confused with the number of hidden units K in the TCM architecture.

where R_{eff} is an effective overlap parameter given by

$$R_{\text{eff}} = \frac{1}{K}\sum_{i=1}^{K}\frac{2}{\pi}\arcsin R_i \qquad R_i = \frac{K}{N}\mathbf{w}_i^{\mathrm{T}}\mathbf{w}_{\nu,i} \tag{8.5}$$

in terms of the student and teacher hidden unit weight vectors $\mathbf{w}_i$ and $\mathbf{w}_{\nu,i}$. In the thermodynamic limit, the R_i are self-averaging, i.e., their values for a specific training set and student from the Gibbs distribution are identical to their averages with probability one. These averages can be obtained from a replica calculation of the average entropy S as a function of the normalized number of training examples, $\alpha = p/N$, following the calculations in Refs. [SOS92, SH92]. We use the assumption of replica symmetry, which is believed to be exact for the case of noise free training data [SH92]. The replica calculation involves, in addition to the R_i, the overlap parameters

$$q_i^p = \frac{K}{N}(\overline{\mathbf{w}}_i^p)^2 \qquad q_i^{\mu p} = \frac{K}{N}(\overline{\mathbf{w}}_i^p)^{\mathrm{T}}\overline{\mathbf{w}}_i^{\mu} \tag{8.6}$$

where $\overline{\mathbf{w}}_i^p = \langle \mathbf{w}_i \rangle_{P(\mathbf{w}|\Theta^{(p)})}$ and similarly $\overline{\mathbf{w}}_i^{\mu} = \langle \mathbf{w}_i \rangle_{P(\mathbf{w}|\Theta^{(\mu)})}$ $(\mu < p)$. The $q_i^{\mu p}$ arise as the overlaps of the students trained on p examples with the committee members which determine the selection of the $(\mu+1)$-th example. The q_i^p can be determined from saddle point equations, whereas for the $q_i^{\mu p}$, an independent ansatz has to be made. In Ref. [SOS92], it was assumed that $q_i^{\mu p} = q_i^{\mu}$, i.e., that the overlap of two students trained on μ and p examples, respectively, is the same as the overlap of two students both trained on μ examples. From the discussion in Section 5.5.2, it follows that this ansatz is in fact exact for the case of a correct inference model considered here.

Following [SH92, EKT+92], we assume symmetry between the hidden units, i.e., $q_i^p = q^p \equiv q$, $q_i^{\mu p} = q_i^{\mu} = q^{\mu} \equiv q(\alpha')$ $(\alpha' = \mu/N)$ and $R_i = R$. The calculation, details of which are relegated to Appendix 8.6.1, can be further simplified by exploiting the relation $R = q$, which expresses the symmetry between teacher and student. One then obtains the normalized entropy $s = S/N$ (apart from an irrelevant additive constant, which we fix such that $s = 0$ at $\alpha = 0$) in the saddle point form

$$s = \operatorname{extr}_q \left\{ \frac{1}{2}(q + \ln(1-q)) + 2\int_0^{\alpha} d\alpha' \int Dz\, H(\gamma z)\ln H(\gamma z) \right\} \tag{8.7}$$

where

$$\gamma = \sqrt{\frac{q_{\text{eff}} - q_{\text{eff}}(\alpha')}{1 - q_{\text{eff}}}} \qquad q_{\text{eff}} = \frac{2}{\pi}\arcsin q \qquad q_{\text{eff}}(\alpha') = \frac{2}{\pi}\arcsin q(\alpha') \tag{8.8}$$

and we have used the usual shorthand $Dz = dz\exp(-\frac{1}{2}z^2)/\sqrt{2\pi}$ and $H(z) = \int_z^{\infty} Dx$.

The effect of query learning enters in (8.7) only through $q_{\text{eff}}(\alpha')$, and by replacing $q_{\text{eff}}(\alpha') \to 0$, the known result for random examples [SH92] is recovered. Differentiating (8.7) with respect to α, one verifies that $ds/d\alpha = -\ln 2$ as expected for maximum information gain queries[4] (the large committee limit $k \to \infty$ has already been taken).

Solving the saddle point equation (which is an integral equation for $q(\alpha)$) numerically, we obtain the average generalization error as plotted in Figure 8.1. For large α, one finds analytically that $\epsilon_g \propto \exp(-c\alpha)$ with $c = \frac{1}{2}\ln 2$ as discussed below. This

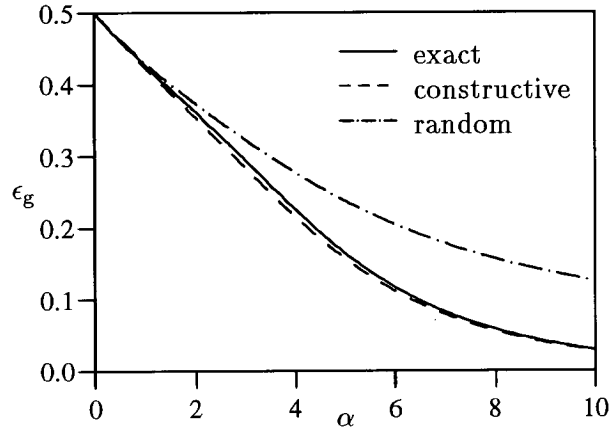

Figure 8.1. Generalization error ϵ_g as a function of the normalized number of examples, α. Full line: exact maximum information gain queries (Section 8.3); broken line: queries as selected by constructive algorithm (Section 8.4); dash-dotted line: random examples.

exponential decay of the generalization error ϵ_g with α provides a marked improvement over the $\epsilon_g \propto 1/\alpha$ decay achieved by random examples [SH92]. The effect of maximum information gain queries is thus similar to what is observed for a binary perceptron student learning from a binary perceptron teacher [SOS92], but the decay constant c in $\epsilon_g \propto \exp(-c\alpha)$ is only half as large in the TCM. This means that asymptotically, twice as many examples are needed for a TCM as for a binary perceptron (when learning from a teacher with the respective architecture) to achieve the same generalization performance, in agreement with the corresponding result for random examples [SH92]. Since in both networks, due to the binary nature of their outputs, maximum information gain queries lead to an entropy $s = -\alpha \ln 2$, we can also conclude that the relation $s \approx \ln \epsilon_g$ for the binary perceptron [SOS92] has to be replaced by $s \approx \ln \epsilon_g^2$ for the

[4]The fact that the entropy is predicted correctly lends further support to the claim that the assumption of replica symmetry is correct for the problem considered here.

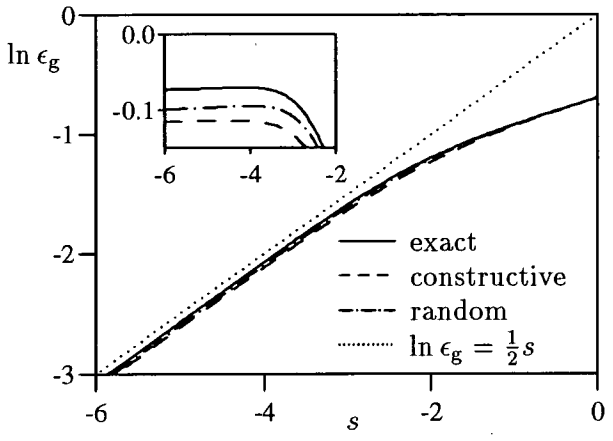

Figure 8.2. Log generalization error $\ln \epsilon_g$ vs. (normalized) entropy s, for queries (exact/constructive algorithm) and random examples. For both queries and random examples, $\ln \epsilon_g \approx \frac{1}{2}s$ for large negative values of s (corresponding to large α). The very small separation between the curves is more clearly seen in the inset, which shows $\ln \epsilon_g - \frac{1}{2}s$ vs. s.

tree committee machine. This relation should hold generally, independently of whether one is learning from queries or from random examples. We have confirmed this by calculating the entropy for learning from random examples and comparing with the corresponding generalization error [SH92], as shown in Figure 8.2.

The above results apply for any finite α in the limit $K \to \infty$. We now briefly discuss the large α asymptotics of the generalization error for large but fixed K (which corresponds to the limit $\alpha \to \infty$ being taken *before* $K \to \infty$). The relation $q = R$ arising from the symmetry between student and teacher holds for any value of K, and we therefore write $q = R = 1 - \Delta$, with $\Delta \ll 1$ in the large α regime. For $K \to \infty$, the asymptotic behaviour of $\Delta(\alpha)$ can be derived from the expression (8.7) for the entropy: Following the reasoning in [SOS92], it can be shown that if Δ decreases exponentially for large α, i.e., $\Delta \propto \exp(-c'\alpha)$, then the integral over α' in (8.7) approaches a constant for $\alpha \to \infty$. The entropy is therefore well approximated by $s = \frac{1}{2}\ln(1-q) + \text{const} = \frac{1}{2}\ln\Delta + \text{const}$ in this limit. Comparing this with the known α dependence of the entropy for maximum information gain query learning, $s = -\alpha \ln 2$, it follows that the ansatz $\Delta \propto \exp(-c'\alpha)$ is self-consistent, with $c' = 2\ln 2$. This result for $K \to \infty$ is identical to that obtained for the binary perceptron (corresponding to $K = 1$) in [SOS92]. Excluding the unlikely possibility that the asymptotic behaviour of $\Delta(\alpha)$ depends non-monotonically on K, we thus conclude that $\Delta \propto \exp(-\alpha\, 2\ln 2)$ asymptotically for *all* K. For the $K \to \infty$ case, where $\epsilon_g \propto \Delta^{1/4}$ for small Δ from (8.4), one then obtains the

large α behaviour $\epsilon_g \propto \exp(-\alpha\frac{1}{2}\ln 2)$ referred to above. For large but finite K, one has to consider the $O(1/K)$ correction to the expression for the generalization error (8.4), which as shown in Appendix 8.6.3 is

$$\epsilon_g = \frac{1}{\pi}\arccos R_{\rm eff} - \frac{1}{2\pi K}\frac{R_{\rm eff}}{\sqrt{1-R_{\rm eff}^2}} + O\left(\frac{1}{K^2}\right). \tag{8.9}$$

For small Δ, it follows from (8.5) that the first and the second term on the right hand side of (8.9) scale as $\Delta^{1/4}$ and $K^{-1}\Delta^{-1/4}$, respectively. This implies that as long as $\Delta^{1/2} \gg O(1/K)$ or, equivalently, $\epsilon_g \gg O(K^{-1/2})$, the first term dominates, yielding $\epsilon_g \propto \exp(-\alpha\frac{1}{2}\ln 2)$ as for $K \to \infty$. In the opposite regime $\epsilon_g \ll O(K^{-1/2})$, we show in Appendix 8.6.3 that the generalization error is given by

$$\epsilon_g = \sqrt{\frac{K}{2\pi}}(1 - R_{\rm eff}) = \sqrt{\frac{2K}{\pi}}\frac{1}{\pi}\arccos R \tag{8.10}$$

for large K. The functional dependence of ϵ_g on R is the same as for the binary perceptron, due to the fact that the dominant contribution to the generalization error arises from errors for which student and teacher only differ in the output of a single hidden unit. This yields $\epsilon_g \propto \Delta^{1/2}$ for small Δ and hence $\epsilon_g \propto \exp(-\alpha\ln 2)$ as for the binary perceptron [SOS92]. We have thus found that the large α behaviour of the generalization error obtained for $K \to \infty$ is valid until the generalization error reaches values of $O(K^{-1/2})$, while for larger α a crossover to behaviour typical of the binary perceptron occurs. This result is consistent with the corresponding observation for random examples in [SH92].

8.4 Constructive query selection algorithm

We now consider the practical realization of maximum information gain queries in the TCM. The query by committee approach, which in the limit $k \to \infty$ is an exact algorithm for selecting maximum information queries, filters queries from a stream of random inputs. This leads to an exponential increase of the query filtering time with the number of training examples that have already been gathered [FSST93]. As a cheap alternative we propose a simple algorithm for *constructing* queries, which is based on the assumption of an approximate decoupling of the entropies of the different hidden units, as follows. Each individual hidden unit of a TCM can be viewed as a binary perceptron. The distribution $P(\mathbf{w}_i|\Theta^{(p)})$ of its weight vector $\mathbf{w}_i$ given a set of training examples $\Theta^{(p)}$ has an entropy S_i associated with it, in analogy to the entropy (8.2)

of the full weight distribution $P(\mathbf{w}|\Theta^{(p)})$. Our 'constructive algorithm' for selecting queries then consists in choosing, for each new query $\mathbf{x}^{\mu+1}$, the inputs $\mathbf{x}_i^{\mu+1}$ to the individual hidden units in such a way as to maximize the decrease in their entropies S_i. As discussed in Appendix 5.8, this can be achieved simply by choosing each $\mathbf{x}_i^{\mu+1}$ to be orthogonal to $\overline{\mathbf{w}}_i^{\mu}$ (and otherwise random, i.e., according to $P(\mathbf{x})$), thus avoiding the cumbersome and time-consuming filtering from a random input stream. In practice, one would of course approximate $\overline{\mathbf{w}}_i^{\mu}$ by an average of $2k$ (say) samples from the Gibbs distribution $P(\mathbf{w}|\Theta^{(\mu)})$; these samples would have been needed anyway in the query by committee approach.

An analysis of the generalization performance achieved by this constructive algorithm proceeds along the same lines as the calculation in the last section; an outline is given in Appendix 8.6.2. Again restricting attention to the limit $k \to \infty$, we find that the saddle point expression (8.7) for the normalized entropy s still holds, but with γ now given by

$$\gamma = \sqrt{\frac{a}{1-a}} \qquad a = \frac{2}{\pi} \arcsin\left(\frac{q - q(\alpha')}{1 - q(\alpha')}\right) \tag{8.11}$$

Differentiating (8.7) with this replacement with respect to α, we find again that $ds/d\alpha = -\ln 2$, which means that in the thermodynamic limit that we consider, queries selected to minimize the individual hidden units' entropies also minimize the overall entropy of the TCM. This may seem surprising at first; heuristically, however, one can argue that for a large number of hidden units K, the correlations in the Gibbs distribution between the hidden unit weight vectors must be weak, and may indeed become negligible in the $K \to \infty$ limit considered here. The generalization performance achieved by the constructive query algorithm, shown in Figure 8.1, is actually slightly superior to that of exact maximum information gain queries as calculated in the previous section. This decrease in generalization error, although slight (about 4% for large α), exemplifies the fact that while information gain and decrease in generalization error are normally correlated, there is no exact one-to-one relationship between them (compare the discussion in Chapters 3 and 4). Query selection algorithms which achieve the same information gain can therefore lead to different generalization performance.

Parenthetically, we note that the two query selection algorithms that we have discussed are quite different in the way they achieve maximum information gain. This can be seen by considering, for a fixed training set $\Theta^{(p)}$ and fixed students chosen randomly from the Gibbs distribution $P(\mathbf{w}|\Theta^{(p)})$, the statistics of the hidden unit outputs induced by the distribution of the next query $\mathbf{x}^{p+1}$ as chosen by either of the two algorithms. The constructive algorithm does not introduce any correlations between the hidden unit outputs, since by definition the $\mathbf{x}_i^{p+1}$ are selected independently of each

other. Maximum information gain is achieved in this case by choosing the hidden unit inputs $\mathbf{x}_i$ such that the output of any given hidden unit is $+1$ or -1 with probability $\frac{1}{2}$, i.e., maximally uncertain. For the exact algorithm discussed in the last section, on the other hand, the distribution of the output of any given hidden unit turns out to be (up to $O(1/K)$ corrections) the same as for random examples (see eq. (8.39) in Appendix 8.6.1). The total output of the TCM is still made maximally uncertain by creating correlations between the outputs of different hidden units (of $O(1/K)$, compare eq. (8.45)), which cause the sum of all the hidden unit outputs to be positive or negative with equal probability. The two algorithms can therefore be seen as two opposite extremes in the way they realize maximum information gain, one based solely on the behaviour of individual hidden units, the other relying only on correlations between hidden units. This leads us to speculate that other algorithms for selecting maximum information gain queries, which would have to operate somewhere between these two extremes, would achieve a generalization performance in the range defined by the results for the two algorithms discussed above.

8.5 Conclusions

We have used the tools of statistical mechanics to analyse query learning for maximum information gain in large tree-committee machines (TCM). For the noise free, perfectly learnable scenario that we have considered, the generalization error ϵ_g decays exponentially with the normalized number of training examples α, which is a significant improvement over learning from random examples, for which $\epsilon_g \propto 1/\alpha$ for large α. Comparing with results for query learning in the binary perceptron, the decay constant c in $\epsilon_g \propto \exp(-c\alpha)$ turns out to be half as large in the TCM, and this implies that the relationship between entropy s and generalization error is $s \approx \ln \epsilon_g^2$ in the TCM, rather than $s \approx \ln \epsilon_g$ as in the binary perceptron. Modifications of the relationship between s and ϵ_g must also be expected for multi-layer networks with architectures more complicated than that of the TCM. This leads to a number of interesting open question regarding the dependence of the decay constant c in the exponential decay of ϵ_g with α on the number of hidden units K in general multi-layer networks. The bound in [FSST93], derived for the $k = 1$ query by committee algorithm, implies a lower bound on c which scales inversely with the VC-dimension [VC71] of the class of networks considered. Taking the storage capacity of a network as a coarse measure of its VC-dimension, one would then conclude from existing bounds [MD89] that c could be as small as $O(1/\ln K)$ for large K. However, the existing results for the capacity of particular networks like the TCM (which was conjectured to be finite for $K \to \infty$

in [Opp94] on the basis of the results of [SH92], but estimated to be infinite in the same limit in [BHS92]) are not unambiguous enough to decide whether realistic networks can saturate the Mitchison-Durbin bound [MD89]. Furthermore, it has been argued previously [Opp94] that both the input space dimension *and* the VC-dimension determine the α-dependence of the generalization error. It may therefore be possible to replace the VC-dimension in the bound in [FSST93] with the input space dimension, and this would yield a c of $O(1)$ independently of K. Further theoretical work is clearly needed to clarify these issues.

In Section 8.4, we have analysed a computationally cheap algorithm for constructing (rather than filtering) approximate maximum information gain queries, based on the assumption of a decoupling of the entropies of individual hidden units. We have found that this constructive algorithm actually achieves slightly better generalization performance than exact maximum information gain queries. This result is particularly encouraging considering the practical application of query learning in more complex multi-layer networks. For example, the proposed constructive algorithm can be modified for query learning in a fully-connected committee machine (where each hidden unit is connected to all the inputs), by simply choosing each new query to be orthogonal to the subspace spanned by the average weight vectors of *all* K hidden units. As long as K is much smaller than the input dimension N, and assuming that for large enough K the approximate decoupling of the hidden unit entropies still holds for fully connected networks, one would expect this algorithm to yield a good approximation to maximum information gain queries[5]. It is an open question whether the same conclusion would also hold for a *general* two-layer network with threshold units (where, in contrast to the committee machine, the hidden-to-output weights are free parameters), which can approximate a large class of input-output mappings. In summary, our results suggest that the drastic improvements in generalization performance achieved by maximum information gain queries can be made available, in a computationally cheap manner, for realistic neural network learning problems.

[5]To make the algorithm work once the permutation symmetry between hidden units is broken, one would of course have to restrict all weight space averages to one of the $K!$ 'ergodic' weight space sectors.

8.6 Appendix: Calculations

8.6.1 Exact maximum information gain queries

In this appendix, we outline the replica calculation of the average entropy (8.7) for exact maximum information gain queries. As usual, the replica trick is employed to remove the logarithm in the average of the (normalized) student space entropy $s = S/N = (1/N)\ln V(\Theta^{(p)})$ over all training sets and teachers,

$$\langle s \rangle_{P(\Theta^{(p)}|\mathcal{V})P(\mathcal{V})} = \frac{1}{N}\left\langle \ln V(\Theta^{(p)}) \right\rangle_{P(\Theta^{(p)}|\mathcal{V})P(\mathcal{V})} = \frac{1}{N}\lim_{n\to 0}\frac{1}{n}\ln\left\langle V^n(\Theta^{(p)})\right\rangle_{P(\Theta^{(p)}|\mathcal{V})P(\mathcal{V})}.$$

The r.h.s. is calculated for positive integer values of n and continued analytically to $n = 0$. The version space volume $V(\Theta^{(p)})$ appearing in the above expression can be written as

$$V(\Theta^{(p)}) = \int d\mathbf{w}\ \tilde{P}(\mathbf{w}) \prod_{\mu=1}^{p} \Theta\left(y^\mu f(\mathbf{x}^\mu)\right).$$

The product of Heaviside Θ-functions ensures that the binary student outputs $f(\mathbf{x}^\mu)$ are the same as the training outputs y^μ, i.e., that the integration runs only over students which are compatible with the training set. Introducing n replicas $\mathbf{w}^a$ of the student space and writing the average over training sets and teachers explicitly, one has

$$\left\langle V^n(\Theta^{(p)})\right\rangle_{P(\Theta^{(p)}|\mathcal{V})P(\mathcal{V})} = \int d\mathbf{w}_\mathcal{V}\, P(\mathbf{w}_\mathcal{V}) \int \prod_{a=1}^{n}\left(d\mathbf{w}^a\, \tilde{P}(\mathbf{w}^a)\right)$$
$$\prod_\mu \left[\int d\mathbf{x}^\mu P(\mathbf{x}^\mu|\Theta^{(\mu-1)}) \sum_{y^\mu=\pm 1} P(y^\mu|\mathbf{x}^\mu,\mathcal{V}) \prod_{a=1}^{n}\Theta\left(y^\mu f^a(\mathbf{x}^\mu)\right)\right]. \qquad (8.12)$$

Due to the assumption of noise free training outputs, one has

$$P(y^\mu|\mathbf{x}^\mu,\mathcal{V}) = \Theta(y^\mu f_\mathcal{V}(\mathbf{x}^\mu))$$

where $f_\mathcal{V}(\mathbf{x}^\mu)$ is the 'clean' teacher output. Since we have also assumed that the teacher prior $P(\mathbf{w}_\mathcal{V})$ and the student pseudo-prior $\tilde{P}(\mathbf{w})$ are identical, it follows from (8.12) that the teacher $\mathbf{w}_\mathcal{V}$ can be treated just like an additional student replica $\mathbf{w}^0$. This we do in the following, using the convention that the replica index a runs over the values $0, 1 \ldots n$ unless otherwise specified. The above teacher-student symmetry, which holds whenever the inference model is correct (see, e.g., [SOS92, WRB93] and Chapter 5), also implies the equality of the overlap parameters R_i and q_i defined in (8.5, 8.6).

In (8.12), the factors contributed by individual training examples are coupled through the dependence of the probability distribution of input $\mathbf{x}^{\mu+1}$ ($\mu = 0 \ldots p-1$) on

the existing training set $\Theta^{(\mu)}$. As in the calculation for the binary perceptron, however, we shall find that the dependence on $\Theta^{(\mu)}$ only manifests itself through overlap parameters which are self-averaging in the thermodynamic limit, so that the averages over individual training examples can be treated as decoupled. Let us therefore now focus on a specific training example $(\mathbf{x}^{\mu+1}, y^{\mu+1})$. Carrying out the sum over the training output $y^{\mu+1}$, we have a contribution

$$\prod_a \Theta\left(f^a(\mathbf{x}^{\mu+1})\right) + \prod_a \Theta\left(-f^a(\mathbf{x}^{\mu+1})\right) \tag{8.13}$$

to (8.12), which has to be averaged over $P(\mathbf{x}^{\mu+1}|\Theta^{(\mu)})$. The superscript $\mu+1$ is dropped in the following to simplify the notation. Writing the student outputs explicitly in terms of their hidden unit outputs,

$$f^a(\mathbf{x}) = \operatorname{sgn}\left(\frac{1}{\sqrt{K}}\sum_{i=1}^{K}\sigma_i^a(\mathbf{x})\right) \qquad \sigma_i^a(\mathbf{x}) = \operatorname{sgn}\left(\sqrt{\frac{K}{N}}(\mathbf{w}_i^a)^{\mathrm{T}}\mathbf{x}_i\right)$$

and using integral representations for the Θ-functions, the term (8.13) takes the form

$$\prod_a\left(\int_0^\infty dy^a \int \frac{d\hat{y}^a}{2\pi}\right)\exp\left(-i\sum_a \hat{y}^a y^a\right) \times \left[\exp\left(\frac{i}{\sqrt{K}}\sum_a \hat{y}^a \sum_i \sigma_i^a(\mathbf{x})\right) + \text{c.c.}\right]. \tag{8.14}$$

The average over $\mathbf{x}$ of the terms in square brackets can be written as a cumulant expansion

$$\left\langle \exp\left(\pm\frac{i}{\sqrt{K}}\sum_a \hat{y}^a \sum_i \sigma_i^a(\mathbf{x})\right)\right\rangle_{\mathbf{x}} = \exp\left\{-\frac{1}{2K}\sum_{ab}\hat{y}^a\hat{y}^b\sum_{ij}\left\langle\sigma_i^a(\mathbf{x})\sigma_j^b(\mathbf{x})\right\rangle_{\mathbf{x}}^{\mathrm{c}} - \frac{1}{24K^2}\sum_{abcd}\hat{y}^a\hat{y}^b\hat{y}^c\hat{y}^d\sum_{ijkl}\left\langle\sigma_i^a(\mathbf{x})\sigma_j^b(\mathbf{x})\sigma_k^c(\mathbf{x})\sigma_l^d(\mathbf{x})\right\rangle_{\mathbf{x}}^{\mathrm{c}} + \ldots\right\} \tag{8.15}$$

where the superscript 'c' on the averages refers to cumulant (or connected) averages. Cumulants involving odd powers of the $\hat{y}^a$ or σ_i^a do not occur in (8.15) because the distribution of $\mathbf{x}$ is invariant under $\mathbf{x} \to -\mathbf{x}$. This can be seen from the fact that for the TCM, $y = f(\mathbf{x}) \to -y$ for $\mathbf{x} \to -\mathbf{x}$, which implies that $\mathbf{x}$ is a bisection query (or is selected by a particular committee of $2k$ students) if and only if $-\mathbf{x}$ is. This entails in particular that $\langle\sigma_i^a(\mathbf{x})\rangle = 0$, which means that the second order cumulant in (8.15)

reduces to a normal average, while the fourth order cumulant can be written as

$$\left\langle \sigma_i^a \sigma_j^b \sigma_k^c \sigma_l^d \right\rangle^c = \left\langle \sigma_i^a \sigma_j^b \sigma_k^c \sigma_l^d \right\rangle - \left\langle \sigma_i^a \sigma_j^b \right\rangle \left\langle \sigma_k^c \sigma_l^d \right\rangle - \left\langle \sigma_i^a \sigma_k^c \right\rangle \left\langle \sigma_j^b \sigma_l^d \right\rangle - \left\langle \sigma_i^a \sigma_l^d \right\rangle \left\langle \sigma_j^b \sigma_k^c \right\rangle . \tag{8.16}$$

The terms in the expansion (8.15) can therefore be calculated from averages of products of an even number of the σ_i^a, and we will show below how this can be done. Intuitively, one would expect that although the σ_i^a (for fixed a) may be correlated, the linear combinations $K^{-1/2} \sum_i \sigma_i^a$ will become Gaussian random variables in the limit $K \to \infty$, leaving only the contribution from the second order cumulants in (8.15). We will confirm this below by showing that the fourth order term is smaller by a factor of $1/K$ than the leading second order term.

To calculate averages over products of the σ_i^a, we use the representation

$$\begin{aligned} \sigma_i^a(\mathbf{x}) &= \sum_{\sigma_i^a = \pm 1} \sigma_i^a \, \Theta\left(\sigma_i^a \sqrt{\frac{K}{N}} (\mathbf{w}_i^a)^{\mathrm{T}} \mathbf{x}_i \right) \\ &= \sum_{\sigma_i^a = \pm 1} \sigma_i^a \int_0^\infty dh \int \frac{d\hat{h}}{2\pi} \exp\left(-i\hat{h}h + i\hat{h}\sigma_i^a \sqrt{\frac{K}{N}} (\mathbf{w}_i^a)^{\mathrm{T}} \mathbf{x}_i \right) \end{aligned} \tag{8.17}$$

Multiplying a finite number of the σ_i^a and collecting the factors depending explicitly on $\mathbf{x}$, we are therefore lead to the consideration of averages of the form

$$\Gamma = \left\langle \exp\left(i \sqrt{\frac{K}{N}} \sum_i \mathbf{a}_i^{\mathrm{T}} \mathbf{x}_i \right) \right\rangle_{\mathbf{x}} \tag{8.18}$$

in which only a finite number of the $\mathbf{a}_i$, which are linear combinations of the $\mathbf{w}_i^a$, are nonzero. These averages can be evaluated using the query by committee approach, in analogy with the calculation in Appendix 5.8. The probability distribution of $\mathbf{x}$ can be written in the form

$$P(\mathbf{x}|\Theta^{(\mu)}) \propto P(\mathbf{x}) \sum_{\{y^\gamma = \pm 1\}} \prod_{\gamma=1}^{2k} \Theta\left(y^\gamma f^\gamma(\mathbf{x}) \right) \equiv P(\mathbf{x}) \, g(\mathbf{x}) \tag{8.19}$$

The index γ labels the $2k$ committee members $\mathbf{w}^\gamma$, which are sampled randomly from the student distribution after μ training examples, $P(\mathbf{w}|\Theta^{(\mu)})$; their outputs for input $\mathbf{x}$ are denoted by $f^\gamma(\mathbf{x})$. The summation over the y^γ is restricted to those combinations for which exactly k of the y^γ are $+1$ and the others are -1, according to the principle of 'maximal disagreement' between committee members. The distribution $P(\mathbf{x})$, finally, implements the assumed spherical constraint on hidden unit input vectors, $P(\mathbf{x}) \propto$

$\prod_i \delta(\mathbf{x}_i^2 - N/K)$. Introducing the hidden unit outputs σ_i^γ of the committee members, the non-trivial factor on the r.h.s. of (8.19) can be written as

$$g(\mathbf{x}) = \sum_{\{y^\gamma\}} \sum_{\{\sigma_i^\gamma = \pm 1\}} \prod_\gamma \left[\Theta\left(y^\gamma \frac{1}{\sqrt{K}} \sum_i \sigma_i^\gamma \right) \prod_i \Theta\left(\sigma_i^\gamma \sqrt{\frac{K}{N}} (\mathbf{w}_i^\gamma)^{\mathrm{T}} \mathbf{x}_i \right) \right]. \tag{8.20}$$

Using (8.19), the desired average (8.18) takes the form

$$\Gamma = \frac{\left\langle g(\mathbf{x}) \exp\left(i\sqrt{\frac{K}{N}} \sum_i \mathbf{a}_i^{\mathrm{T}} \mathbf{x}_i \right) \right\rangle}{\langle g(\mathbf{x}) \rangle} \tag{8.21}$$

of a ratio of two averages over $\mathbf{x}$ distributed according to $P(\mathbf{x})$. From (8.20), one sees that the quantities to be averaged depend on $\mathbf{x}$ only through the scalar products

$$z_i^\gamma = \sqrt{\frac{K}{N}} (\mathbf{w}_i^\gamma)^{\mathrm{T}} \mathbf{x}_i \qquad z_i = \sqrt{\frac{K}{N}} \mathbf{a}_i^{\mathrm{T}} \mathbf{x}_i.$$

In the limit $N/K \gg 1$ of interest to us, these become zero mean Gaussian variables with covariances given by

$$\left\langle z_i^\gamma z_j^\delta \right\rangle = \delta_{ij} \frac{K}{N} (\mathbf{w}_i^\gamma)^{\mathrm{T}} \mathbf{w}_j^\delta \qquad \langle z_i^\gamma z_j \rangle = \delta_{ij} \frac{K}{N} (\mathbf{w}_i^\gamma)^{\mathrm{T}} \mathbf{a}_j \qquad \langle z_i z_j \rangle = \delta_{ij} \frac{K}{N} \mathbf{a}_i^{\mathrm{T}} \mathbf{a}_j \tag{8.22}$$

As usual, the overlaps $(K/N)(\mathbf{w}_i^\gamma)^{\mathrm{T}} \mathbf{w}_j^\delta$ are assumed to be self-averaging and can therefore be replaced by $q_i^\mu = q^\mu$, the latter equality following from the assumption of hidden unit symmetry. The z_i^γ and z_i can thus be written in terms of uncorrelated Gaussian random variables $\tilde{z}_i^\gamma$, $\tilde{x}_i$ and $\tilde{z}_i$ with zero mean and unit variance, in the form

$$z_i^\gamma = \tilde{x}_i \sqrt{q^\mu} + \tilde{z}_i^\gamma \sqrt{1 - q^\mu} \tag{8.23}$$

$$z_i = \tilde{x}_i u_i + \tilde{z}_i v_i \tag{8.24}$$

with only a finite number of nonzero u_i and v_i (corresponding to the nonzero $\mathbf{a}_i$). We only need to consider explicitly the average in the numerator of (8.21), since the denominator is obtained by setting $u_i \equiv v_i \equiv 0$. The $\tilde{z}_i^\gamma$ occur only in the 'input-to-hidden' Θ-functions in $g(\mathbf{x})$, eq. (8.20), and can be averaged out immediately:

$$\left\langle \Theta\left(\sigma_i^\gamma (\tilde{x}_i \sqrt{q^\mu} + \tilde{z}_i^\gamma \sqrt{1 - q^\mu}) \right) \right\rangle_{\tilde{z}_i^\gamma} = H\left(-\sigma_i^\gamma \tilde{x}_i \sqrt{\frac{q^\mu}{1 - q^\mu}} \right).$$

Introducing integral representations

$$\Theta\left(y^\gamma \frac{1}{\sqrt{K}} \sum_i \sigma_i^\gamma\right) = \int_0^\infty dh^\gamma \int \frac{d\hat{h}^\gamma}{2\pi} \exp\left(-i\hat{h}^\gamma h^\gamma + i\hat{h}^\gamma y^\gamma \frac{1}{\sqrt{K}} \sum_i \sigma_i^\gamma\right)$$

for the remaining Θ-functions in $g(\mathbf{x})$, the sum over the σ_i^γ can be carried out by using

$$\sum_{\sigma_i^\gamma = \pm 1} \exp\left(i\hat{h}^\gamma y^\gamma \frac{1}{\sqrt{K}} \sigma_i^\gamma\right) H\left(-\sigma_i^\gamma \tilde{x}_i \sqrt{\frac{q^\mu}{1-q^\mu}}\right) = \cos\frac{\hat{h}^\gamma}{\sqrt{K}} + iy^\gamma \hat{H}_i \sin\frac{\hat{h}^\gamma}{\sqrt{K}} \tag{8.25}$$

where

$$\hat{H}_i = \hat{H}\left(\tilde{x}_i \sqrt{\frac{q^\mu}{1-q^\mu}}\right) \qquad \hat{H}(\cdot) = 1 - 2H(\cdot).$$

Collecting these results, one obtains for the numerator of (8.21)

$$\exp\left(-\frac{1}{2}\sum_i v_i^2\right) \sum_{\{y^\gamma\}} \left[\prod_\gamma \int_0^\infty dh^\gamma \int \frac{d\hat{h}^\gamma}{2\pi} \exp(-i\hat{h}^\gamma h^\gamma)\right]$$
$$\left\langle \exp\left\{i \sum_i \left[u_i \tilde{x}_i + \sum_\gamma \ln\left(\cos\frac{\hat{h}^\gamma}{\sqrt{K}} + iy^\gamma \hat{H}_i \sin\frac{\hat{h}^\gamma}{\sqrt{K}}\right)\right]\right\}\right\rangle_{\{\tilde{x}_i\}} \tag{8.26}$$

where the first factor arises from the trivial average over the $\tilde{z}_i$.

So far, all manipulations hold for TCMs with any number K of hidden units. We now specialize to the limit $K \to \infty$ and expand

$$\ln\left(\cos\frac{\hat{h}^\gamma}{\sqrt{K}} + iy^\gamma \hat{H}_i \sin\frac{\hat{h}^\gamma}{\sqrt{K}}\right) = 1 + \frac{i}{\sqrt{K}} \hat{h}^\gamma y^\gamma \hat{H}_i - \frac{1}{2K}\left(1 - \hat{H}_i^2\right)(\hat{h}^\gamma)^2 + O(K^{-3/2}) \tag{8.27}$$

When summed over $i = 1 \ldots K$, the $O(K^{-3/2})$ terms can contribute at most $O(K^{-1/2})$ and are therefore negligible for $K \to \infty$. Inserting the expansion (8.27) into (8.26), the integrals over the h^γ and $\hat{h}^\gamma$ can be carried out, each giving a factor

$$H\left(-y^\gamma \frac{c_1}{\sqrt{1-c_2}}\right)$$

where

$$c_1 = \frac{1}{\sqrt{K}} \sum_i \hat{H}_i \qquad c_2 = \frac{1}{K} \sum_i \hat{H}_i^2.$$

The summation over the y^γ then becomes trivial, and by setting $u_i \equiv v_i \equiv 0$ to obtain

the denominator of (8.21), one finds

$$\Gamma = \exp\left(-\frac{1}{2}\sum_i v_i^2\right) \frac{\left\langle \exp\left(i\sum_i u_i\tilde{x}_i\right) H^k\left(\frac{c_1}{\sqrt{1-c_2}}\right) H^k\left(-\frac{c_1}{\sqrt{1-c_2}}\right)\right\rangle_{\{\tilde{x}_i\}}}{\left\langle H^k\left(\frac{c_1}{\sqrt{1-c_2}}\right) H^k\left(-\frac{c_1}{\sqrt{1-c_2}}\right)\right\rangle_{\{\tilde{x}_i\}}} \tag{8.28}$$

Since the $\tilde{x}_i$ and hence the $\hat{H}_i$ are uncorrelated, the variance of c_2 is $O(1/K)$, while its average is $O(1)$. This means that in the $K \to \infty$ limit, c_2 can be replaced by its average

$$\bar{c}_2 = \frac{1}{K}\sum_i \left\langle \hat{H}_i^2 \right\rangle_{\tilde{x}_i} = \int D\tilde{x}\ \hat{H}\left(\tilde{x}\sqrt{\frac{q^\mu}{1-q^\mu}}\right)$$

So far we have not yet exploited the limit of a large committee, $k \to \infty$, which yields the desired exact maximum information queries[6]. This we do now by noting that the product of H-functions in (8.28) constrains c_1 to values arbitrarily close to zero for $k \to \infty$, so that we can write

$$\Gamma = \exp\left(-\frac{1}{2}\sum_i v_i^2\right) \frac{\left\langle \exp\left(i\sum_i u_i\tilde{x}_i\right)\delta(c_1)\right\rangle_{\{\tilde{x}_i\}}}{\left\langle \delta(c_1)\right\rangle_{\{\tilde{x}_i\}}} \tag{8.29}$$

in this limit. The numerator can be written explicitly as

$$\left\langle \exp\left(i\sum_i u_i\tilde{x}_i\right)\delta(c_1)\right\rangle_{\{\tilde{x}_i\}} = \int \frac{dt}{2\pi}\prod_i \int D\tilde{x}_i \exp\left(iu_i\tilde{x}_i + \frac{i}{\sqrt{K}}t\hat{H}_i\right). \tag{8.30}$$

We now make use of the fact that only a finite number s of the u_i (and v_i), which we denote by $u_{i_1}\ldots u_{i_s}$ (and $v_{i_1}\ldots v_{i_s}$, respectively), are nonzero. The numerator (8.30) therefore contains $K-s$ factors of the form

$$f\left(t^2/K\right) = \int D\tilde{x}_i \exp\left(\frac{i}{\sqrt{K}}t\hat{H}_i\right) = 1 - \frac{\bar{c}_2}{2K}t^2 + O(t^4/K^2)$$

which give

$$\left[f\left(t^2/K\right)\right]^{K-s} = \exp\left(-t^2\bar{c}_2/2\right)\left(1 + C(t;K)\right)$$

where

$$C(t;K) = \sum_{l=1}^{\infty} K^{-l} C_l(t)$$

[6]This is true if $k \to \infty$ for fixed K and N. As in Chapter 5, we have reversed the order of the limits, taking the limit $k \to \infty$ last. The fact that the correct α dependence of the entropy results (see Section 8.3) shows that this procedure is valid.

and the $C_l(t)$ are polynomials in t. One therefore obtains for (8.30)

$$\left\langle \exp\left(i\sum_i u_i\tilde{x}_i\right)\delta(c_1)\right\rangle_{\{\tilde{x}_i\}} = (2\pi\bar{c}_2)^{1/2}\left\langle \exp\left(i\sum_{r=1}^{s} u_{i_r}\tilde{x}_{i_r}\right)\exp\left(-\frac{\Sigma^2}{2\bar{c}_2}\right)\left(1+\langle C(t;K)\rangle_t\right)\right\rangle \tag{8.31}$$

where the outer average is taken over the $\tilde{x}_{i_r}$, the average of $C(t;K)$ is performed over a Gaussian distribution of t with mean $-i\Sigma/\bar{c}_2$ and variance $1/\bar{c}_2$, and we have set

$$\Sigma = \frac{1}{\sqrt{K}}\sum_{r=1}^{s}\hat{H}_{i_r}.$$

The averages $\langle C_l(t)\rangle_t$ contributing to $\langle C(t;K)\rangle_t$ are simply polynomials in Σ^2 (with coefficients containing $\bar{c}_2$). Expanding

$$\exp\left(-\frac{\Sigma^2}{2\bar{c}_2}\right)\left(1+\langle C(t;K)\rangle_t\right) = \exp\left(-\frac{\Sigma^2}{2\bar{c}_2}\right)\left(1+\sum_{l=1}^{\infty}K^{-l}\langle C_l(t)\rangle_t\right) \tag{8.32}$$

in powers of Σ^2, we can deduce that in the limit $K\to\infty$, the contribution from $C(t;K)$ to the coefficient of any power of Σ^2 can be neglected compared to that from the exponential. It is crucial that this is true for *all* powers of Σ^2 since Σ^2 is itself proportional to $1/K$ and the leading terms of the expansion of the exponential in (8.32) can be cancelled when the average (8.29) is used in a calculation of averages of products of the $\sigma_i^a(\mathbf{x})$; we shall see an explicit example of this below. Neglecting $\langle C(t;K)\rangle$ in (8.31) and combining this with the corresponding result for the denominator of (8.29), obtained by setting $u_{i_r}=0$, $s=0$, $\Sigma=0$, we finally obtain

$$\Gamma = \exp\left(-\frac{1}{2}\sum_{r=1}^{s}v_{i_r}^2\right)\left\langle \exp\left[i\sum_{r=1}^{s}u_{i_r}\tilde{x}_{i_r} - \frac{1}{2\bar{c}_2K}\left(\sum_{r=1}^{s}\hat{H}_{i_r}\right)^2\right]\right\rangle, \tag{8.33}$$

the average being taken over the $\tilde{x}_{i_r}$. As pointed out above, the second term in the exponential has to be expanded to the first order in $1/K$ which contributes to the specific average of a product of the σ_i^a that one is trying to calculate. Note that the first term of this expansion gives

$$\Gamma = \exp\left(-\frac{1}{2}\sum_{r=1}^{s}(u_{i_r}^2+v_{i_r}^2)\right) \tag{8.34}$$

which, as can be verified by setting $k=0$ in (8.28), is simply the result for random

examples.

Let us now apply the result (8.33) together with the representation (8.17) to the calculation of the averages

$$\left\langle \sigma_i^a(\mathbf{x})\sigma_j^b(\mathbf{x})\right\rangle_{\mathbf{x}} \tag{8.35}$$

which give the leading term in the cumulant expansion (8.15). For $i = j$, $a = b$, one has trivially

$$\langle \sigma_i^a(\mathbf{x})\sigma_i^a(\mathbf{x})\rangle_{\mathbf{x}} = 1. \tag{8.36}$$

For $i = j$, $a \neq b$, one has

$$\sigma_i^a(\mathbf{x})\sigma_i^b(\mathbf{x}) = \sum_{\sigma_i^a,\sigma_i^b=\pm 1} \sigma_i^a\sigma_i^b \int_0^\infty dh_i^a\, dh_i^b \int \frac{d\hat{h}_i^a\, d\hat{h}_i^b}{(2\pi)^2}$$
$$\exp\left(-i\hat{h}_i^a h_i^a - i\hat{h}_i^b h_i^b + i\sqrt{\frac{K}{N}}(\hat{h}_i^a\sigma_i^a\mathbf{w}_i^a + \hat{h}_i^b\sigma_i^b\mathbf{w}_i^b)^{\mathrm{T}}\mathbf{x}_i\right) \tag{8.37}$$

so that there is only one nonzero vector

$$\mathbf{a}_i = \hat{h}_i^a\sigma_i^a\mathbf{w}_i^a + \hat{h}_i^b\sigma_i^b\mathbf{w}_i^b$$

while all $\mathbf{a}_k$ with $k \neq i$ are zero. It then follows from (8.22, 8.24) that

$$u_i^2 + v_i^2 = \frac{K}{N}\mathbf{a}_i^2 = (\hat{h}_i^a)^2 + (\hat{h}_i^b)^2 + \sigma_i^a\sigma_i^b\ \hat{h}_i^a\hat{h}_i^b\frac{K}{N}(\mathbf{w}_i^a)^{\mathrm{T}}\mathbf{w}_i^b \tag{8.38}$$

The self-averaging overlap $(K/N)(\mathbf{w}_i^a)^{\mathrm{T}}\mathbf{w}_i^b$ can be replaced by the overlap parameter $q_i^p = q$ defined in (8.6), where we have again used the assumption of hidden unit symmetry. The independence of the overlap parameter q of the replica indices a and b arises from our assumption of replica symmetry. As pointed out above, the leading contribution in $1/K$ of the result (8.33) for the required $\mathbf{x}$ average (8.18) is given by $\exp(-\frac{1}{2}(u_i^2 + v_i^2))$. Inserting this into (8.37) and using (8.38), the integrals can be carried out and one finds

$$\left\langle \sigma_i^a(\mathbf{x})\sigma_i^b(\mathbf{x})\right\rangle_{\mathbf{x}} = \sum_{\sigma_i^a,\sigma_i^b=\pm 1} \sigma_i^a\sigma_i^b\,\frac{1}{2\pi}\arccos(-\sigma_i^a\sigma_i^b q) = \frac{2}{\pi}\arcsin q \tag{8.39}$$

up to terms of $O(1/K)$. As explained above, this result is identical to that for random examples since we have only used the zeroth order term (8.34) in the $1/K$ expansion of eq. (8.33).

Let us now consider the remaining case $i \neq j$. In this case, one has

$$\left\langle \sigma_i^a(\mathbf{x})\sigma_j^b(\mathbf{x})\right\rangle_{\mathbf{x}} = \sum_{\sigma_i^a,\sigma_j^b=\pm 1} \sigma_i^a\sigma_j^b \int_0^\infty dh_i^a\, dh_j^b \int \frac{d\hat{h}_i^a\, d\hat{h}_j^b}{(2\pi)^2} \exp\left(-i(\hat{h}_i^a h_i^a + \hat{h}_j^b h_j^b)\right) \Gamma \tag{8.40}$$

where now two nonzero vectors

$$\mathbf{a}_i = \hat{h}_i^a \sigma_i^a \mathbf{w}_i^a \qquad \mathbf{a}_j = \hat{h}_j^b \sigma_j^b \mathbf{w}_j^b$$

appear in Γ. The corresponding values of u_i and v_i are determined by

$$u_i^2 + v_i^2 = \frac{K}{N}\mathbf{a}_i^2 = (\hat{h}_i^a)^2 \tag{8.41}$$

$$u_i\sqrt{q^\mu} = \frac{K}{N}(\mathbf{w}_i^\gamma)^{\mathrm{T}}\mathbf{a}_i = \hat{h}_i^a\sigma_i^a q_i^{\mu p} \tag{8.42}$$

where $\mathbf{w}_i^\gamma$ is the i-th hidden unit weight vector of any of the $2k$ committee members and we have again used the self-averaging of weight vector overlaps. It is at this point that we use the identity $q_i^{\mu p} = q_i^\mu$ which holds for the case of a correct inference model considered here (see Section 8.3). Exploiting again the assumed hidden unit symmetry $q_i^\mu = q^\mu$, eqs. (8.41, 8.42) can be solved to give

$$u_i = \hat{h}_i^a\sigma_i^a\sqrt{q^\mu} \qquad v_i = \hat{h}_i^a\sigma_i^a\sqrt{1-q^\mu}.$$

Exactly analogous expressions are obtained for u_j and v_j. Inserting these values into the result (8.33) for Γ, one has

$$\begin{aligned}\Gamma = \exp\left(-\frac{1}{2}(\hat{h}_i^a)^2 - \frac{1}{2}(\hat{h}_j^b)^2\right) - \frac{1}{2\bar{c}_2 K}\exp\left[-\frac{1}{2}(1-q^\mu)\left((\hat{h}_i^a)^2 + (\hat{h}_j^b)^2\right)\right] \times \\ \Big[\ \left\langle \exp(i\hat{h}_i^a\sigma_i^a\sqrt{q^\mu}\tilde{x}_i)\hat{H}_i^2\right\rangle_{\tilde{x}_i} \exp\left(-\frac{1}{2}q^\mu(\hat{h}_j^b)^2\right) \\ +\ \left\langle \exp(i\hat{h}_j^b\sigma_j^b\sqrt{q^\mu}\tilde{x}_j)\hat{H}_j^2\right\rangle_{\tilde{x}_j} \exp\left(-\frac{1}{2}q^\mu(\hat{h}_i^a)^2\right) \\ +\ 2\left\langle \exp(i\hat{h}_i^a\sigma_i^a\sqrt{q^\mu}\tilde{x}_i)\hat{H}_i\right\rangle_{\tilde{x}_i}\left\langle \exp(i\hat{h}_j^b\sigma_j^b\sqrt{q^\mu}\tilde{x}_j)\hat{H}_j\right\rangle_{\tilde{x}_j}\Big] + O(K^{-2})\end{aligned} \tag{8.43}$$

All the terms in this expansion which do not change sign when the sign of either σ_i^a or σ_j^b is reversed are cancelled by the sums $\sum_{\sigma_i^a=\pm 1}\sigma_i^a = \sum_{\sigma_j^b=\pm 1}\sigma_j^b = 0$ in (8.40) and hence do not contribute to the desired average (8.40). It can easily be verified that this means that only the term in the last line of (8.43) survives, which when inserted

into (8.40) leads to terms of the form

$$\int_0^\infty dh_i^a \int \frac{d\hat{h}_i^a}{2\pi} \exp\left(-i\hat{h}_i^a h_i^a - \frac{1}{2}(1-q^\mu)(\hat{h}_i^a)^2\right) \left\langle \exp(i\hat{h}_i^a \sigma_i^a \sqrt{q^\mu}\tilde{x}_i)\hat{H}_i \right\rangle_{\tilde{x}_i} =$$

$$\left\langle \hat{H}_i \, H\left(-\sigma_i^a \tilde{x}_i \sqrt{\frac{q^\mu}{1-q^\mu}}\right) \right\rangle_{\tilde{x}_i} = \frac{1}{2}\sigma_i^a \left\langle \hat{H}_i^2 \right\rangle_{\tilde{x}_i}$$

where the second equality follows from the fact that $\hat{H}_i \to -\hat{H}_i$ under $\tilde{x}_i \to -\tilde{x}_i$. This average can now be evaluated explicitly by using the result

$$\int Dz \, \hat{H}(az)\, \hat{H}(bz) = \frac{2}{\pi} \arcsin \frac{ab}{\sqrt{1+a^2}\sqrt{1+b^2}} \tag{8.44}$$

which can be verified by differentiation with respect to either of the parameters a or b (not to be confused with replica indices). One finally obtains

$$\left\langle \sigma_i^a(\mathbf{x})\sigma_j^b(\mathbf{x}) \right\rangle_{\mathbf{x}} = \sum_{\sigma_i^a, \sigma_j^b = \pm 1} (\sigma_i^a)^2 (\sigma_j^b)^2 \left(-\frac{1}{2\bar{c}_2 K}\right) 2 \left(\frac{1}{\pi} \arcsin q^\mu\right)^2 = -\frac{1}{K}\frac{2}{\pi} \arcsin q^\mu \tag{8.45}$$

where we have used (8.44) to evaluate $\bar{c}_2 = (2/\pi) \arcsin q^\mu$.

Inserting the results (8.36, 8.39, 8.45) into (8.15), one finds

$$\left\langle \exp\left(\pm \frac{i}{\sqrt{K}} \sum_a \hat{y}^a \sum_i \sigma_i^a(\mathbf{x})\right) \right\rangle_{\mathbf{x}} =$$

$$\exp\left[-\frac{1}{2}(1-q_{\text{eff}}) \sum_a (\hat{y}^a)^2 - \frac{1}{2}(q_{\text{eff}} - q_{\text{eff}}^\mu) \left(\sum_a \hat{y}^a\right)^2 + O(\text{`}\hat{y}^4\text{'})\right] \tag{8.46}$$

in terms of the effective overlap parameters

$$q_{\text{eff}} = \frac{2}{\pi} \arcsin q \qquad q_{\text{eff}}^\mu = \frac{2}{\pi} \arcsin q^\mu.$$

Evaluating the fourth order cumulants (8.16) in the same manner as demonstrated above for the second order averages, one can show that the $O(\text{`}\hat{y}^4\text{'})$ term in (8.46) is at most $O(1/K)$ for large K. This confirms our expectation that the linear combinations $K^{-1/2}\sum_i \sigma_i^a$ become Gaussian random variables for $K \to \infty$.

The result (8.46) can now be inserted into (8.14), yielding an expression exactly analogous to that obtained for learning from random examples in the binary perceptron (see, e.g., [GT90]). Details of the remaining standard manipulations leading to the saddle point form of the entropy, eq. (8.7), are therefore omitted. We only note

that the product over μ, which by exponentiation is converted into a sum, is replaced by an integral over $\alpha' = \mu/N$ in the thermodynamic limit, with the corresponding replacement of the $q_{\rm eff}^{\mu}$ by a continuous function $q_{\rm eff}(\alpha')$.

8.6.2 Constructive query selection algorithm

Let us now consider how the above calculation is modified for the constructive query selection algorithm proposed in Section 8.4. The manipulations leading to (8.14) remain unchanged. However, now there is no need to resort to a cumulant expansion to average the terms in square brackets in (8.14) over the input vector $\mathbf{x}$. This is because the hidden unit input vectors $\mathbf{x}_i$ are uncorrelated by definition of the constructive algorithm. This allows the decomposition

$$\left\langle \exp\left(\pm\frac{i}{\sqrt{K}}\sum_a \hat{y}^a \sum_i \sigma_i^a(\mathbf{x}_i)\right)\right\rangle_{\mathbf{x}} = \prod_i \left\langle \exp\left(\pm\frac{i}{\sqrt{K}}\sum_a \hat{y}^a \sigma_i^a(\mathbf{x}_i)\right)\right\rangle_{\mathbf{x}_i} \tag{8.47}$$

where we have changed our earlier notation $\sigma_i^a(\mathbf{x})$ for the hidden unit outputs to $\sigma_i^a(\mathbf{x}_i)$ in order to emphasize that they only depend on the inputs to the respective hidden unit. Focusing on one of the factors on the r.h.s. of (8.47), i.e., fixing i, we use a representation similar to (8.17) to write

$$\exp\left(\pm\frac{i}{\sqrt{K}}\sum_a \hat{y}^a \sigma_i^a(\mathbf{x}_i)\right) = \sum_{\{\sigma^a\}} \exp\left(\pm\frac{i}{\sqrt{K}}\sum_a \hat{y}^a\sigma^a\right) \prod_a \Theta\left(\sigma^a\sqrt{\frac{K}{N}}(\mathbf{w}_i^a)^{\rm T}\mathbf{x}_i\right) \tag{8.48}$$

where the summation is over the $n+1$ binary outputs $\sigma^a = \pm 1$ $(a = 0 \ldots n)$ of the i-th hidden unit of the student replicas. We now carry out the average of the r.h.s. over the input $\mathbf{x}_i$. By definition of our constructive algorithm, $\mathbf{x}_i$ is chosen to be orthogonal to the average weight vector $\overline{\mathbf{w}}_i^{\mu}$ of the hidden unit (where the average is taken over the version space defined by the existing training set $\Theta^{(\mu)}$) and otherwise random, i.e., according to $P(\mathbf{x})$. In the limit $N/K \gg 1$ that we consider, this means that the scalar products

$$z^a = \sqrt{\frac{K}{N}}(\mathbf{w}_i^a)^{\rm T}\mathbf{x}_i$$

become zero mean Gaussian random variables with covariances

$$\left\langle z^a z^b\right\rangle_{\mathbf{x}_i} = \frac{K}{N}(\mathbf{w}_i^a)^{\rm T}\mathbf{w}_i^b - \frac{(K/N)(\mathbf{w}_i^a)^{\rm T}\overline{\mathbf{w}}_i^{\mu}\cdot(K/N)(\mathbf{w}_i^b)^{\rm T}\overline{\mathbf{w}}_i^{\mu}}{(K/N)(\overline{\mathbf{w}}_i^{\mu})^2}.$$

As usual, the weight vector overlaps are assumed to be self-averaging and are replaced by the corresponding overlap parameters

$$\begin{aligned}\frac{K}{N}(\mathbf{w}_i^a)^{\mathrm{T}}\mathbf{w}_i^b &= q_i^p + \delta_{ab}(1-q_i^p) = q + \delta_{ab}(1-q)\\ \frac{K}{N}(\mathbf{w}_i^a)^{\mathrm{T}}\overline{\mathbf{w}}_i^\mu &= q_i^{\mu p} = q^\mu\\ \frac{K}{N}(\overline{\mathbf{w}}_i^\mu)^2 &= q_i^\mu = q^\mu\end{aligned}$$

where we have again used the assumptions of replica symmetry and hidden unit symmetry and the fact that $q_i^{\mu p} = q_i^\mu$. This leads to the representation

$$z^a = \tilde{x}\sqrt{q-q^\mu} + \tilde{z}^a\sqrt{1-q}$$

of the z^a in terms of uncorrelated unit variance Gaussian variables $\tilde{x}$ and $\tilde{z}^a$. The average over the $\tilde{z}^a$ is trivial and gives

$$\left\langle \Theta\left(\sigma^a(\tilde{x}\sqrt{q-q^\mu} + \tilde{z}^a\sqrt{1-q})\right)\right\rangle_{\tilde{z}^a} = H\left(-\sigma^a\tilde{x}\sqrt{\frac{q-q^\mu}{1-q}}\right) = \frac{1}{2}(1+\sigma^a\hat{H}_x)$$

where

$$\hat{H}_x = \hat{H}\left(\tilde{x}\sqrt{\frac{q-q^\mu}{1-q}}\right).$$

The summation over the σ^a in (8.48) can now be carried out as in (8.25), yielding for the desired average (8.47)

$$\left\langle \exp\left(\pm\frac{i}{\sqrt{K}}\sum_a \hat{y}^a\sum_i \sigma_i^a(\mathbf{x}_i)\right)\right\rangle_{\mathbf{x}} = \left[\left\langle\prod_a\left(\cos\frac{\hat{y}^a}{\sqrt{K}} \pm i\hat{H}_x\sin\frac{\hat{y}^a}{\sqrt{K}}\right)\right\rangle_{\tilde{x}}\right]^K$$

Due to the occurrence of the K-th power, which arises from the product over i in (8.47), we only have to expand the average over $\tilde{x}$ up to terms of order $1/K$:

$$\begin{aligned}&\left\langle\prod_a\left(\cos\frac{\hat{y}^a}{\sqrt{K}} \pm i\hat{H}_x\sin\frac{\hat{y}^a}{\sqrt{K}}\right)\right\rangle_{\tilde{x}} =\\ &\quad\left\langle 1 \pm \frac{i}{\sqrt{K}}\hat{H}_x\sum_a\hat{y}^a - \frac{1}{2K}(1-\hat{H}_x^2)\sum_a(\hat{y}^a)^2 - \frac{1}{2K}\hat{H}_x^2\left(\sum_a\hat{y}^a\right)^2\right\rangle_{\tilde{x}}.\end{aligned} \quad (8.49)$$

Since $\hat{H}_x$ is an odd function of $\tilde{x}$, the $O(1/\sqrt{K})$ term vanishes while the remaining terms give

$$\left\langle \exp\left(\pm\frac{i}{\sqrt{K}}\sum_a \hat{y}^a \sum_i \sigma_i^a(\mathbf{x}_i)\right)\right\rangle_{\mathbf{x}} = \exp\left[-\frac{1}{2}(1-a)\sum_a (\hat{y}^a)^2 - \frac{1}{2}a\left(\sum_a \hat{y}^a\right)^2\right] \tag{8.50}$$

where the coefficient a (not to be confused with a replica index) is

$$a = \left\langle \hat{H}_x^2 \right\rangle_{\tilde{x}} = \frac{2}{\pi}\arcsin\left(\frac{q-q(\alpha')}{1-q(\alpha')}\right) \tag{8.51}$$

in agreement with the definition (8.11); the second equality in (8.51) follows by applying (8.44). Eq. (8.50) is the analogue of (8.46) for the constructive query selection algorithm; standard manipulations leading to the expression (8.7) for the average entropy, with the modified value of γ as defined in (8.11), are again omitted.

8.6.3 Generalization error

Finally, let us sketch how the generalization error (8.4) for $K \to \infty$ and the $O(1/K)$ correction given in (8.9) is calculated. The generalization error as the probability that the outputs $f(\mathbf{x})$ and $f_\nu(\mathbf{x})$ of a given student and teacher, respectively, disagree on a random test input, is defined by

$$\epsilon_g = \langle \Theta\left(-f(\mathbf{x})f_\nu(\mathbf{x})\right)\rangle_{P(\mathbf{x})} = 2\left\langle \Theta\left(-f(\mathbf{x})\right)\Theta\left(f_\nu(\mathbf{x})\right)\right\rangle_{P(\mathbf{x})}$$

where the second equality follows from the invariance of the distribution $P(\mathbf{x})$ of random test inputs under $\mathbf{x} \to -\mathbf{x}$. Introducing dummy variables σ_i, τ_i for the hidden unit outputs of student and teacher and using the decoupling of the hidden unit input vectors $\mathbf{x}_i$, one has

$$\epsilon_g = 2\sum_{\{\sigma_i,\tau_i=\pm 1\}} \Theta\left(-\frac{1}{\sqrt{K}}\sum_i \sigma_i\right)\Theta\left(\frac{1}{\sqrt{K}}\sum_i \tau_i\right) \times \prod_i \left\langle \Theta\left(\sigma_i\sqrt{\frac{K}{N}}\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_i\right)\Theta\left(\tau_i\sqrt{\frac{K}{N}}\mathbf{w}_{\nu,i}^{\mathrm{T}}\mathbf{x}_i\right)\right\rangle_{\mathbf{x}_i} \tag{8.52}$$

The scalar products

$$z = \sqrt{\frac{K}{N}}\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_i \qquad z_\nu = \sqrt{\frac{K}{N}}\mathbf{w}_{\nu,i}^{\mathrm{T}}\mathbf{x}_i$$

become zero mean Gaussian variables in the limit $N/K \gg 1$, with covariances

$$\left\langle z^2 \right\rangle_{\mathbf{x}_i} = \left\langle z_\nu^2 \right\rangle_{\mathbf{x}_i} = 1 \qquad \langle z z_\nu \rangle_{\mathbf{x}_i} = \frac{K}{N} \mathbf{w}_i^{\mathrm{T}} \mathbf{w}_{\nu,i} = R_i = R$$

from the definition (8.5) of the overlap parameters R_i and the assumption of hidden unit symmetry[7]. Writing $z_\nu = zR + \tilde{x}\sqrt{1-R^2}$, in terms of a zero mean Gaussian variable $\tilde{x}$ uncorrelated with z and of unit variance, one has

$$\left\langle \Theta\left(\sigma_i \sqrt{\frac{K}{N}} \mathbf{w}_i^{\mathrm{T}} \mathbf{x}_i\right) \Theta\left(\tau_i \sqrt{\frac{K}{N}} \mathbf{w}_{\nu,i}^{\mathrm{T}} \mathbf{x}_i\right) \right\rangle_{\mathbf{x}_i} = \left\langle \Theta\left(\sigma_i z\right) \Theta\left(\tau_i (zR + \sqrt{1-R^2}\tilde{x})\right) \right\rangle$$

$$= \int Dz\, H\left(-\sigma_i \tau_i z \frac{R}{1-R^2}\right) = \frac{1}{4}\left(1 + \sigma_i \tau_i \frac{2}{\pi} \arcsin R\right). \tag{8.53}$$

The form of this result is intuitively obvious, since for $\sigma_i \tau_i = -1$, the average simply corresponds to half the probability that the i-th hidden units, which are binary perceptrons with weight vectors $\mathbf{w}_i$ and $\mathbf{w}_{\nu,i}$, differ in their outputs for the random inputs $\mathbf{x}_i$. This probability is (compare eq. (5.21))

$$\epsilon = \frac{1}{\pi} \arccos R = \frac{1}{2}\left(1 - \frac{2}{\pi} \arcsin R\right) = \frac{1}{2}(1 - R_{\mathrm{eff}})$$

where

$$R_{\mathrm{eff}} = \frac{2}{\pi} \arcsin R \tag{8.54}$$

as defined in (8.5). It follows that

$$\epsilon_{\mathrm{g}} = 2^{1-K} \sum_{\{\sigma_i, \tau_i = \pm 1\}} \Theta\left(-\frac{1}{\sqrt{K}} \sum_i \sigma_i\right) \Theta\left(\frac{1}{\sqrt{K}} \sum_i \tau_i\right) \epsilon^l (1-\epsilon)^{K-l} \tag{8.55}$$

where l is the number of $i \in \{1 \ldots K\}$ for which $\sigma_i \tau_i = -1$. This representation of the generalization error is useful for determining its large α behaviour for fixed K. For $\alpha \to \infty$, R and R_{eff} tend to one, so that $\epsilon \to 0$; hence the low l-terms in (8.55) dominate. Counting the possible configurations of the $\{\sigma_i, \tau_i\}$ for $l = 1$ and $l = 2$, one obtains

[7] Below, we only describe the calculation of the generalization error for the case of hidden unit symmetry. The $K \to \infty$ result (8.4) for the general case ($R_i \neq R_j$ for some i, j) can be obtained by trivial modifications.

$$\begin{aligned}\epsilon_g &= 2^{1-K}\left\{\binom{K}{L+1}\binom{L+1}{1}\epsilon(1-\epsilon)^{K-1}\right. \\ &\quad \left.+\left[\binom{K}{L+1}\binom{L+1}{2}+\binom{K}{L+2}\binom{L+2}{2}\right]\epsilon^2(1-\epsilon)^{K-2}+O(\epsilon^3)\right\} \\ &= \frac{K!}{(2^L L!)^2}\left[\epsilon - L\epsilon^2 + O(\epsilon^3)\right] \qquad (8.56)\end{aligned}$$

where we have set $L = \frac{1}{2}(K-1)$ (remember that K was assumed to be odd, so that L is an integer). Evaluating the prefactor for large K using Stirling's formula,

$$\frac{K!}{(2^L L!)^2} \approx \sqrt{\frac{2K}{\pi}} \qquad (8.57)$$

and neglecting the $O(\epsilon^2)$ term gives the asymptotic form of the generalization error

$$\epsilon_g \approx \sqrt{\frac{2K}{\pi}}\,\epsilon = \sqrt{\frac{2K}{\pi}}\,\frac{1}{\pi}\arccos R$$

stated in the text (eq. (8.10)). Comparing the contributions of order ϵ and ϵ^2 in (8.56), we see that this expression is valid as long as $\epsilon \ll 1/L = O(1/K)$, corresponding to $\epsilon_g \ll O(1/\sqrt{K})$.

To obtain the large K behaviour of the generalization error for finite α, it is easiest to go back to (8.52) and introduce integral representations for the Θ-functions that remain once the result for the averages (8.53) is inserted. Using (8.54), one obtains

$$\begin{aligned}\epsilon_g = 2^{1-2K} \sum_{\{\sigma_i,\tau_i=\pm 1\}} \int_0^\infty dx\,dy \int \frac{d\hat{x}}{2\pi}\frac{d\hat{y}}{2\pi} \exp\left(-i\hat{x}x - i\hat{y}y\right) \\ \times \exp\left(i\frac{\hat{x}}{\sqrt{K}}\sum_i \sigma_i - i\frac{\hat{y}}{\sqrt{K}}\sum_i \tau_i\right)\prod_i (1+\sigma_i\tau_i R_{\rm eff}) \qquad (8.58)\end{aligned}$$

The contributions from different i now factorize, and the sums over the σ_i and τ_i can be performed to give

$$\epsilon_g = 2\int_0^\infty dx\,dy \int \frac{d\hat{x}}{2\pi}\frac{d\hat{y}}{2\pi} \exp\left(-i\hat{x}x - i\hat{y}y\right)\left[\cos\frac{\hat{x}}{\sqrt{K}}\cos\frac{\hat{y}}{\sqrt{K}} + R_{\rm eff}\sin\frac{\hat{x}}{\sqrt{K}}\sin\frac{\hat{y}}{\sqrt{K}}\right]^K .$$

Expanding the K-th power explicitly, one obtains

$$\epsilon_g = 2\sum_{l=0}^{K}\binom{K}{l}(I_l)^2 R_{\rm eff}^l \qquad (8.59)$$

where

$$\begin{aligned} I_l &= \int_0^\infty dx \int \frac{d\hat{x}}{2\pi} \exp(-i\hat{x}x) \cos^{K-l}(\hat{x}/\sqrt{K}) \sin^l(\hat{x}/\sqrt{K}) \\ &= \int \frac{d\hat{x}}{2\pi i} \frac{1}{\hat{x} - i\epsilon} \cos^{K-l}(\hat{x}/\sqrt{K}) \sin^l(\hat{x}/\sqrt{K}) \end{aligned} \tag{8.60}$$

with $\epsilon \to 0^+$. For even l, one can replace $1/(\hat{x} - i\epsilon)$ by $i\pi\delta(\hat{x})$ since the remaining part of the integrand is odd; this yields

$$I_0 = 1/2 \qquad I_{2l} = 0 \ \text{ for } \ l \geq 1. \tag{8.61}$$

For odd l, one can replace $1/(\hat{x} - i\epsilon)$ by $1/\hat{x}$; for simplicity of notation, we also relabel $\hat{x} \to x$. For large K, one has

$$\cos^K(x/\sqrt{K}) = e^{-\frac{1}{2}x^2} \left(1 - \frac{x^4}{12K} + O(1/K^2)\right)$$

and we discard the last term since we are only interested in contributions up to $O(1/K)$. However, we have to respect the periodicity of the cosine in order to get the correct $O(1/K)$ term and therefore replace

$$\cos^K(x/\sqrt{K}) = \sum_{m=-\infty}^{\infty} (-1)^m f(x - m\pi\sqrt{K}) \qquad f(x) = e^{-\frac{1}{2}x^2} \left(1 - \frac{x^4}{12K}\right).$$

Rescaling the integration variable by $\sqrt{K}$, one then has from (8.60)

$$2\pi i\, I_l = \int \frac{dx}{x} \tan^l x \sum_m (-1)^m f(\sqrt{K}(x - m\pi))$$

For large K, the factor $f(\sqrt{K}(x - m\pi))$ is vanishingly small except in regions of width $O(1/\sqrt{K})$ around the points $x = m\pi$. The remaining part of the integrand, $(\tan^l x)/x$, can therefore be Taylor expanded around these points, yielding a sum of Gaussian integrals for I_l which can be carried out explicitly. From the $m = 0$ term, one finds the contribution

$$I_l^{m=0} = \frac{K^{-l/2}}{\sqrt{2\pi i}} \left[(l-2)!! + \frac{1}{K}\left(\frac{l}{3} l!! - \frac{(l+2)!!}{12}\right) + O(1/K^2)\right]$$

while for $m \neq 0$ one gets

$$I_l^{m\neq 0} = \frac{K^{-l/2}}{\sqrt{2\pi i}} \left(-\frac{(-1)^m\, l!!}{K m^2 \pi^2} + O(1/K^2)\right).$$

Here we have used the notation $l!! = l(l-2)\cdots 3\cdot 1$ (remember that l is odd). The sum over m is performed by using

$$\sum_{m=1}^{\infty} \frac{(-1)^m}{m^2} = -\frac{\pi^2}{12}$$

and yields

$$I_l = \frac{K^{-l/2}}{\sqrt{2\pi i}} \left[(l-2)!! + \frac{l}{4K} l!! + O(1/K^2) \right]. \tag{8.62}$$

Inserting (8.61, 8.62) into (8.59) and using that

$$\binom{K}{l} \frac{1}{K^l} = \frac{1}{l!} \left(1 - \frac{l(l-1)}{2K} + O(1/K^2) \right)$$

one obtains after a bit of algebra

$$\epsilon_g = \frac{1}{2} - \frac{1}{\pi} \sum_{l \text{ odd}}^{K} \frac{((l-2)!!)^2}{l!} \left[1 + \frac{l}{2K} + O(1/K^2) \right] R_{\text{eff}}^l.$$

Comparing this with the Taylor expansions of the functions $\arccos R_{\text{eff}}$ and $(1-R_{\text{eff}}^2)^{-1/2}$, one reads off that for large K

$$\epsilon_g = \frac{1}{\pi} \arccos R_{\text{eff}} - \frac{1}{2\pi K} \frac{R_{\text{eff}}}{\sqrt{1-R_{\text{eff}}^2}} + O(1/K^2).$$

This completes the derivation of the results (8.4, 8.9) given in the text.

Chapter 9

Summary and Outlook

Let us summarize the main results of the work presented in this thesis. For perfectly learnable problems, we found qualitative differences between linear, 'invertible' and nonlinear, 'non-invertible' rules, which suggested that query learning is generally more useful for the latter than for the former. However, queries also yield significant improvements for linear rules when the number of training examples is of the order of the number of parameters of the student (provided that the training data is not too noisy). This case may occur quite frequently in practice, where the total number of training examples can be severely limited. Based on our general arguments about the crucial role of 'invertibility' of the rule, which essentially allows perfect generalization after presentation of a finite number of noise free training examples, we expect that the results for simple nonlinear, invertible rules (such as perceptrons with nonlinear, monotonic output functions) would be similar to those for the linear case. We also found that in situations where the training algorithm is ill-matched to the learning problem at hand, queries can perform worse than random examples if they are not explicitly selected to minimize the generalization error (but rather the entropy, for example). However, this phenomenon was found to be confined to situations in which the generalization error has not decreased significantly from the value it had before any training examples had been received; it may thus not be a problem of great practical relevance. It would be interesting to explore whether this still holds for query learning of more complicated rules.

For imperfectly learnable problems with linear students, we studied minimum entropy queries. Minimum *teacher space* entropy queries proved to be counterproductive, due to lack of feedback about the progress of the student in learning the rule; for minimum *student space* queries, which are the only practical choice if the teacher space is unknown, the structure of the student space was seen to dominate the efficacy of

query learning. We pointed out that this result, if it holds more generally, has significant practical implications: it would enable predictions about the usefulness of query learning on the basis of *how* one is trying to learn (as determined by the student space and the training algorithm), independently of *what* one is trying to learn. One way of checking the potential generality of this conclusion would be to calculate finite size effects in a scenario with linear students and nonlinear perceptron teachers, in order to see whether the functional form of the teacher manifests itself in a qualitatively new form. This would be straightforward using the response function methods described in Chapter 7.

In Chapter 5, we investigated query learning in situations where knowledge about the teacher space is not available. This lead us to the definition of query learning assuming the inference model is correct, which was shown to have to significant drawbacks: The actual optimization of a given objective function is no longer guaranteed, even when selecting queries expressly for this purpose. More importantly, we found that for discrete (in particular, binary) output students, query learning can perform worse than random examples even for an infinitely large number of training examples; we termed this the problem of 'self-confirming hypotheses far from the truth' due to the fact that query selection can produce training sets which will continually reconfirm a wrong hypothesis about the true rule. In our investigation of the effect of different noise processes corrupting training examples with binary outputs, we also saw that, depending on the form of the noise, the discrete nature of a rule can be 'smoothed out'. This reduces the efficacy of query learning to a level that we had previously associated with continuous output rules. We pointed out the need for a more careful investigation of this point, in particular with regard to typical noise models that might occur in practice.

To overcome the problems of query learning assuming the inference model is correct, we proposed to combine query learning with inference model selection or adaptation. For the scenarios considered, the results were encouraging; in particular, the problem of self-confirming hypotheses was avoided and the generalization performance achieved by queries was consistently better than that for learning from random examples. A more detailed study of this topic would therefore appear to be worthwhile, in particular given the rather limited range of scenarios that we have explored in this context.

In the final two chapters, we tried to gauge the importance of effects that can occur in neural networks which would realistically be used in practical supervised learning problems. Chapter 7 dealt with finite size effects, and we found that the results derived in the thermodynamic limit of infinite system size are generally valid even for fairly small systems with a number of parameters of the order of tens or hundreds. This

is certainly encouraging and suggests that the thermodynamic limit can often make accurate predictions for real-world problems. Exceptions to this rule were regions in the neighbourhood of phase transitions in the learning behaviour. A further study of this topic in the context of learning problems with first order phase transitions (see, e.g., [SST92]) would therefore be an interesting extension.

In Chapter 8, we investigated query learning in a simple multi-layer network with binary outputs, the tree committee machine. Two main conclusions emerged: Firstly, the form of the relationship between entropy or information gain and generalization error was found to be modified compared to single-layer networks; this raised a number of interesting questions regarding the efficacy of query learning for minimum entropy in more complex multi-layer networks. Secondly, and more importantly from a practical point of view, we found a computationally cheap *constructive* algorithm for query selection which does yield significantly better generalization performance than learning from random examples. It has been outlined how this algorithm could be adapted to more powerful neural networks; it remains to be seen whether a calculation of the resulting generalization performance is tractable and whether qualitatively new features of query learning will emerge.

We hope that the above results shed some light on the general capabilities and limitations of query learning; the possible extensions mentioned above and in the course of the preceding chapters, should help to make the picture more complete. There are, however, still many missing pieces of the puzzle: One major point that we have not addressed is query learning in multi-layer networks with sigmoidal transfer functions for the hidden and output units. Since such networks implement continuous input-output mappings, one might naively say that they should show qualitatively the same behaviour as linear networks. However, as the presence of local minima in the training error 'surface' of such networks makes clear, they are not 'invertible' in the sense that linear networks are; for any number of (noise free) training examples, there will typically be a finite (and possibly large) number of networks which are compatible with the training set. Some of these networks will be simple transformations of one another (related, for example, by a permutation of the hidden units and the corresponding weights), but there may also be other 'accidental degeneracies'. One could therefore classify multi-layer networks with continuous output functions as 'continuous non-invertible' networks. It is tempting to think that the efficacy of query learning in these networks would lie between the two extremes of 'invertible' networks on the one hand and discrete, 'non-invertible' networks (such as the binary perceptron) on the other hand. This hypothesis is based on the number of networks compatible with a sufficiently large training set, which is one for invertible, larger than one and finite for

continuous non-invertible, and infinity for discrete non-invertible networks. This topic certainly deserves further study; an investigation of query learning in single-layer perceptrons with non-monotonic continuous output functions might be a useful starting point.

Another topic that we have not tackled is query learning for 'over-sophisticated' students, which due to their functional form can learn rules which are more complex than the actual teacher. It is unclear how the sub-optimally large number of parameters of such students would affect query learning. On the one hand, one could argue that a lot of these parameters might be irrelevant to the actual predictions that the student makes, and that queries (especially for minimum student space entropy) could waste a large number of training examples trying to determine these irrelevant parameters (D. Cohn, private communication). On the other hand, one might turn the argument around and say that the post-training distribution of irrelevant parameters will only be affected very weakly by new training examples, if at all, and that consideration of the *expected* entropy decrease would force queries to concentrate on the determination of relevant parameters. A logical extension of an investigation of over-sophisticated students would be to consider 'universal' students, making use of the universal approximation properties of multi-layer networks (see, e.g., [HSW89]). Such universal networks would have to have an infinitely large number of hidden units, but one can choose appropriate priors over the weights which give sensible priors over the corresponding input-output mappings [Nea94]. With such universal students, problems associated with query learning of imperfectly learnable rules would no longer occur. In fact, one could extend the idea even further and, discarding the neural network representation of students altogether, consider learning with more general models such as, for example, Gaussian processes [Wil95]. It would be exciting to see what form query selection for minimum entropy, say, would take in such models, and how such queries would improve generalization performance.

In line with our general philosophy, we have concentrated mainly on theoretical questions about query learning so far. Having reviewed our results along with possible extensions and perspectives for future work, a brief discussion of challenges arising from the practical application of query learning is now in order. One major point is of course the implementation of query selection algorithms. If one wants to retain the principled approach of selecting queries to optimize a given objective function, then the first step would be the calculation of the relevant averages of the objective function over the student post-training distribution, the teacher posterior and the distribution of the unknown new training output to obtain the function $\epsilon(\Theta^{(p)}, x)$ defining a query selection algorithm (see Chapter 2). This can in principle be accomplished using Monte

Carlo techniques (see, e.g., [GS90, Nea93]), as has been demonstrated by Paass and Kindermann [PK95] for several toy learning scenarios. Another approach would be to choose the functional form of the students and the training algorithm in such a way that at least some of the averages can be carried out analytically (see, e.g., [CGJ95]). Finally, one could also consider using cheaply computable 'proxies' for objective functions; in [PK95], for example, the 'current loss' (essentially the generalization error calculated for a specific input x, rather than averaged over the input distribution) was proposed as a proxy for the 'future loss' (the expected generalization error after query x and the corresponding output have been added to the training set), and this appeared to produce acceptable results. Having obtained the function $\epsilon(\Theta^{(p)}, x)$, the next step is the identification of an input which optimizes it. If a stream of random inputs is available cheaply, this can be done by query *filtering*; however, it has to be borne in mind that query filtering times may become very large as learning proceeds [FSST93]. Also, the situation may often not be as simple as for maximum information gain queries in binary output models, where the globally optimal value of the objective function ($\ln 2 = 1$ bit) is known *a priori*; one would then have to come up with a prescription as to when to accept an input yielding only a 'local' optimum of the objective function. For query *construction*, the most direct way of finding the optimum of the objective function is by an exhaustive search of a set of candidate inputs which cover the input space reasonably well (see, e.g., [PK95]). In high-dimensional input spaces, however, this may well be infeasible, and one might have to rely, for example, on gradient descent techniques as in [CGJ95], or restrict the search to the neighbourhood of previous training inputs [Coh94]. An additional complication could arise in scenarios where the distribution $P(x)$ of random inputs is highly structured. As explained in Chapter 2, queries should only be chosen from regions of inputs space where $P(x)$ is nonzero, since otherwise the corresponding teacher output may be ill-defined. If these regions consist of a number of well-separated 'clusters', query construction may become a highly non-trivial task (see also [Fre93]). One will then probably be forced back to query filtering; alternatively, one could first try to find the structure of the input distribution by some unsupervised learning technique such as vector quantization or Kohonen networks (see, e.g., [HKP91]) and then use this to map inputs to a representation in which random inputs have a near-uniform distribution.

Query learning, certainly in its principled form based on objective function optimization, can be computationally expensive in practical applications. This is particularly true if Monte Carlo techniques have to be used to carry out all or some of the necessary averages. There are several ways in which this problem could be tackled: One of the most obvious suggestions would be to cut down on the computational effort

of query learning by selecting queries in batches, rather than one by one as assumed in our analysis. This approach has been studied by simulations; the results reviewed in, for example, [FTK89], suggest that it might only result in a small loss in generalization performance, even for comparatively large batches (see also [CAL94]). An alternative to this would be to mix queries and random examples; as we saw in Section 3.4.1, single queries are normally more effective after previous random examples than after queries, and this could be expected to offset the loss in performance due to the interspersed random examples, at least to some degree.

Mixing of random examples and queries would most likely also make query learning (which in practical situations will often have to be carried out using the assumption that the inference model is correct) more robust against inference model misspecification. In light of the risks associated with query learning with incorrect inference models (see Chapter 5), this would be a very desirable property. It could also be achieved by several other methods; beyond those already mentioned at the end of Chapter 6, one could, for example, 'prime' query learning with an initial set of random training examples, select an inference model on the basis of this training set, and then proceed with sequential query selection as usual. A related approach was investigated in [CM93], and appeared to have desirable statistical properties. In situations where one is reasonably confident that the correct inference model is contained in a finite collection of models, one might also contemplate using query learning initially to select the most appropriate model (see, e.g., [PR93] and references therein), with subsequent queries selected to learn the teacher rule within this (hopefully correct) inference model. Finally, there might also be some potential for incorporating the techniques of 'robust experimental design' into practical query selection algorithms. The approach typically taken in robust design (see, e.g., [Ber85, BB86]) is to assume that the correct inference model is a member of a certain (often infinite) family of models. Queries are then chosen to maximize the minimum of the objective function over all models, or to maximize its average over a 'hyper-prior' expressing how likely the different inference models are deemed to be[1]. This approach obviously makes the query selection process even more computationally intensive. So far, it has therefore been explored mainly for simple 'almost linear' or one-dimensional scenarios (see, e.g., [CN82, SY84, DS91, Det94, DJ94, Wie94, CF95]); it remains to be seen whether it can be extended to more complex scenarios.

The above discussion shows that although we have provided some answers to our original question 'how useful is query learning?', many more questions have been raised

[1]This applies if the sign of the objective function for query selection is such that its optimum is a maximum; otherwise the roles of 'maximum' and 'minimum' have to be reversed.

by our results, and much work remains to be done. Given this fact and the enormous potential for fruitful interaction between the disciplines of statistics, computer science, physics, and others, query learning looks set to remain an exciting area of research for years to come.

Bibliography

[AD91] A C Atkinson and A N Donev. *Optimum experimental designs.* Clarendon Press, Oxford, 1991.

[AH78] A Ash and A Hedayat. An introduction to design optimality with an overview of the literature. *Communications in Statistics, Part A - Theory and Methods*, 7:1295–1325, 1978.

[AK95] D Angluin and M Kharitonov. When won't membership queries help. *Journal of Computer and System Sciences*, 50:336–355, 1995.

[AM93] S Amari and N Murata. Statistical theory of learning curves under entropic loss criterion. *Neural Computation*, 5:140–153, 1993.

[Ami89] D J Amit. *Modelling Brain Function.* Cambridge University Press, Cambridge, 1989.

[Ang88] D Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[AT87] M P Allen and D J Tildesley. *Computer Simulations of Liquids.* Clarendon Press, Oxford, 1987.

[Atk82] A C Atkinson. Developments in the design of experiments. *International Statistical Review*, 50:161–177, 1982.

[Bau91] E Baum. Neural network algorithms that learn in polynomial time from examples and queries. *IEEE Transactions on Neural Networks*, 2:5–19, 1991.

[BB86] J Berger and L M Berliner. Robust Bayes and empirical Bayes analysis with ϵ-contaminated priors. *Annals of Statistics*, 14:461–486, 1986.

[BBS91] A G Barto, S J Bradtke, and S P Singh. Real-time learning and control using asynchronous dynamic programming. Technical Report COINS 91-57, University of Massachusetts, Amherst, MA, 1991.

[BD87] G Box and N Draper. *Empirical Model-Building and Response Surfaces.* Wiley, New York, 1987.

[BEH89] A Blumer, A Ehrenfeucht, and D Haussler. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36:929–965, 1989.

[Ber85] J Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer, NewYork, 1985.

[BH95] P F Baldi and K Hornik. Learning in linear neural networks - a survey. *IEEE Transactions on Neural Networks*, 6:837–858, 1995.

[BHS92] E Barkai, D Hansel, and H Sompolinsky. Broken symmetries in multilayered perceptrons. *Physical Review A*, 45:4146–4161, 1992.

[BKO93] S Bös, W Kinzel, and M Opper. Generalization ability of perceptrons with continuous outputs. *Physical Review E*, 47:1384–1391, 1993.

[BL91] E Baum and K Lang. Constructing hidden units using examples and queries. In R.P. Lippmann, J.E. Moody, and D S Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 904–910, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[Bor75] D M Borth. A total entropy criterion for the dual problem of model discrimination and parameter estimation. *Journal of the Royal Statistical Society, Series B*, 37:77–87, 1975.

[BR88] A Blum and R L Rivest. Training a 3-node neural network is NP-complete. In D S Touretzky, editor, *Advances in Neural Information Processing Systems 1*, pages 494–501, San Mateo, CA, 1988. Morgan Kaufmann.

[BS93] M Biehl and H Schwarze. Learning drifting concepts with neural networks. *Journal of Physics A*, 26:2651–2665, 1993.

[BS94] A Bruce and D Saad. Statistical mechanics of hypothesis evaluation. *Journal of Physics A*, 27:3355–3363, 1994.

[BSS] D Barber, P Sollich, and D Saad. Finite size effects in on-line learning of multi-layer neural networks. Submitted to *Europhysics Letters.*

[BSS95] D Barber, D Saad, and P Sollich. Finite-size effects and optimal test set size in linear perceptrons. *Journal of Physics A*, 28:1325–1334, 1995.

[BSW90] A G Barto, R S Sutton, and C J C H Watkins. Learning and sequential decision making. In M Gabriel and J Moore, editors, *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, pages 539–602, Cambridge, MA, 1990. MIT Press.

[BV92] G L Bilbro and D E Van Den Brout. Maximum entropy and learning theory. *Neural Computation*, 4:839–853, 1992.

[BW91] W L Buntine and A S Weigend. Bayesian back-propagation. *Complex Systems*, 5:603–643, 1991.

[BY86] K Binder and A P Young. Spin-glasses - experimental facts, theoretical concepts, and open questions. *Reviews of Modern Physics*, 58:801–976, 1986.

[Cac94] C Cachin. Pedagogical pattern selection strategies. *Neural Networks*, 7:175–181, 1994.

[CAL90] D Cohn, L Atlas, and R Ladner. Training connectionist networs with queries and selective sampling. In D Touretzky, editor, *Advances in Neural Information Processing Systems 2*, San Mateo, CA, 1990. Morgan Kaufmann.

[CAL94] D Cohn, L Atlas, and R Ladner. Improving generalization with active learning. *Machine learning*, 15:201–221, 1994.

[CF95] D Cook and V Fedorov. Constrained optimization of experimental designs. *Statistics*, 26:129–178, 1995.

[CGJ95] D A Cohn, Z Ghahramani, and M I Jordan. Active learning with statistical models. In G Tesauro, D S Touretzky, and T K Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 705–712, Cambridge, MA, 1995. MIT Press.

[CM93] P Chaudhuri and P A Mykland. Nonlinear experiments - optimal design and inference based on likelihood. *Journal of the American Statistical Association*, 88:538–546, 1993.

[CN82] R D Cook and C J Nachtsheimer. Model-robust, linear optimal design. *Technometrics*, 24:49–54, 1982.

[Coh90] D Cohn. A local approach to optimal queries. In D S Touretzky et al., editors, *Proceedings of Connectionist Models Summer School (CMMS-90), San Diego*, pages 173–182, San Mateo, CA, 1990. Morgan Kaufmann.

[Coh94] D A Cohn. Neural network exploration using optimal experimental design. In J D Cowan, G Tesauro, and J Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 679–686, San Francisco, CA, 1994. Morgan Kaufmann.

[Cra46] H Cramér. *Mathematical Methods of Statistics.* Princeton University Press, Princeton, 1946.

[CRL94] D Cohn, E A Riskin, and R Ladner. Theory and practice of vector quantizers trained in small training sets. *Machine learning*, 16:54–65, 1994.

[CS70] P A K Covey-Crump and S D Silvey. Optimal regression designs with previous observations. *Biometrika*, 57:551–566, 1970.

[DeG62] M H DeGroot. Uncertainty, information, and sequential experiments. *Annals of Mathematical Statistics*, 33:404–419, 1962.

[Det94] H Dette. Robust designs for multivariate polynomial regression on the d-cube. *Journal of Statistical Planning and Inference*, 38:105–124, 1994.

[DGPB91] D Derrida, R B Griffiths, and A Prügel-Bennett. Finite-size effects and bounds for perceptron models. *Journal of Physics A*, 24:4907–4940, 1991.

[DJ94] W DuMouchel and B Jones. A simple Bayesian modification of D-optimal design to reduce dependence on an assumed model. *Technometrics*, 36:37–47, 1994.

[DS91] A DasGupta and W J Studden. Robust Bayesian experimental-designs in normal-linear models. *Annals of Statistics*, 19:1244–1256, 1991.

[Dun] A P Dunmur. Nonlinear rule learning by a simple perceptron. Unpublished.

[DW93] A P Dunmur and D J Wallace. Learning and generalization in a linear perceptron stochastically trained with noisy data. *Journal of Physics A*, 26:5767–5779, 1993.

[Eat83] M L Eaton. *Multivariate Statistics - A Vector Space Approach.* John Wiley, New York, 1983.

[EKT+92] A Engel, H M Köhler, F Tschepke, H Vollmayr, and A Zippelius. Storage capacity and learning algorithms for two-layer neural networks. *Physical Review A*, 45:7590–7609, 1992.

[El-91] M A El-Gamal. The role of priors in active Bayesian learning in the sequential statistical decision theory framework. In W T Grandy Jr. and L H Schick, editors, *Maximum entropy and Bayesian Methods, Laramie 1990*, pages 33–38, Dordrecht, 1991. Kluwer.

[ER90] B Eisenberg and R Rivest. On the sample complexity of PAC-learning using random and chosen examples. In M Fulk and J Case, editors, *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 154–162, San Mateo, CA, 1990. Kaufmann.

[Fed72] V V Fedorov. *Theory of Optimal Experiments.* Academic Press, New York, 1972.

[Fel70] W Feller. *Introduction to Probability Theory and Its Applications*, volume 1. John Wiley, New York, 3rd edition, 1970. (1st edition 1950).

[Fre93] Y Freund. *Data filtering and distribution modeling algorithms for machine learning.* PhD thesis, University of Santa Cruz, September 1993.

[FSST93] Y Freund, H S Seung, E Shamir, and N Tishby. Information, prediction, and query by committee. In S J Hanson, J D Cowan, and C Lee Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 483–490, San Mateo, CA, 1993. Morgan Kaufmann.

[FTK89] I Ford, D M Titterington, and C P Kitsos. Recent advances in nonlinear experimental design. *Technometrics*, 31:49–60, 1989.

[FTW85] I Ford, D M Titterington, and C F J Wu. Inference and sequential design. *Biometrika*, 72:545–551, 1985.

[Gar88] E Gardner. The space of interactions in neural network models. *Journal of Physics A*, 21:257–270, 1988.

[GP82] J Gladitz and J Pilz. Bayes designs for multiple linear regression on the unit sphere. *Math. Operationsforschung und Statistik, Series Statistics*, 13:491–506, 1982.

[GS90] A E Gelfand and A F M Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:389–409, 1990.

[GSW89] E J Gardner, N Stroud, and D J Wallace. Training with noise and the storage of correlated patterns in a neural network model. *Journal of Physics A*, 22:2019–2030, 1989.

[GT90] G Györgyi and N Tishby. Statistical theory of learning a rule. In W Theumann and R Köberle, editors, *Neural Networks and Spin Glasses*, pages 3–36. World Scientific, Singapore, 1990. Abbreviated version published as [Gyö90].

[Gul89] S F Gull. Developments in maximum entropy data analysis. In J Skilling, editor, *Maximum Entropy and Bayesian Methods (MaxEnt 1988)*, pages 53–71, Dordrecht, 1989. Kluwer.

[Gyö90] G Györgyi. Inference of a rule by a neural network with thermal noise. *Physical Review Letters*, 64:2957–2960, 1990.

[Han93] L K Hansen. Stochastic linear learning: Exact test and training error averages. *Neural Networks*, 6:393–396, 1993.

[HB95] T Hofmann and J Buhmann. Multidimensional scaling and data clustering. In G Tesauro, D S Touretzky, and T K Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 459–466, Cambridge, MA, 1995. MIT Press.

[HBH93] R M Heiberger, D K Bhaumik, and B Holland. Optimal data augmentation strategies for additive models. *Journal of the American Statistical Association*, 88:926–938, 1993.

[HCOM91] J-N Hwang, J J Choi, S Oh, and RJ Marks II. Query-based learning applied to partially trained multilayer perceptrons. *IEEE Transactions on Neural Networks*, 2:131–136, 1991.

[HK70] A Hoerl and R Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 8:27–51, 1970.

[HK92] L Holmström and P Koistinen. Using additive noise in back-propagation training. *IEEE Transactions on Neural Networks*, 3:24–38, 1992.

[HKP91] J Hertz, A Krogh, and R G Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, California, 1991.

[HKST94] D Haussler, M Kearns, H S Seung, and N Tishby. Rigorous learning curve bounds from statistical mechanics. In *Proceedings of the Seventh Annual ACM Workshop on Computational Learning Theory (COLT '94)*, pages 76–87, 1994.

[HKT89] J A Hertz, A Krogh, and G I Thorbergsson. Phase transitions in simple learning. *Journal of Physics A*, 22:2133–2150, 1989.

[HSW89] H M Hornik, M Stinchcombe, and H White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[Joh78] F John. *Partial Differential Equations.* Springer, New York, 3rd edition, 1978.

[KC87] A I Khuri and J A Cornell. *Response Surfaces (Designs and Analyses).* Marcel Dekker, New York, 1987.

[KC92] O Kinouchi and N Caticha. Optimal generalization in perceptrons. *Journal of Physics A*, 25:6243–6250, 1992.

[KH92a] A Krogh and J A Hertz. Generalization in a linear perceptron in the presence of noise. *Journal of Physics A*, 25:1135–1147, 1992.

[KH92b] A Krogh and J A Hertz. A simple weight decay can improve generalization. In J E Moody, S J Hanson, and R P Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 950–957, San Mateo, CA, 1992. Morgan Kaufmann Publishers.

[KMT93] S R Kulkarni, S K Mitter, and J N Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11:23–35, 1993.

[KO91] W Kinzel and M Opper. Dynamics of learning. In E Domany, J L van Hemmen, and K Schulten, editors, *Models of Neural Networks*, pages 149–171. Springer, Berlin, 1991.

[KR90] W Kinzel and P Rujan. Improving a network generalization ability by selecting examples. *Europhysics Letters*, 13:473–477, 1990.

[Kro92] A Krogh. Learning with noise in a linear perceptron. *Journal of Physics A*, 25:1119–1133, 1992.

[KS92] Y Kabashima and S Shinomoto. Learning curves for error minimum and maximum likelihood algorithms. *Neural Computation*, 4:712–719, 1992.

[KS95] Y Kabashima and S Shinomoto. Learning a decision boundary from stochastic examples: Incremental algorithms with and without queries. *Neural Computation*, 7:158–172, 1995.

[Lin56] D V Lindley. On the measure of the amount of information provided by an experiment. *Annals of Mathematical Statistics*, 27:986–1005, 1956.

[LKS91] Y Le Cun, I Kanter, and S A Solla. Eigenvalues of covariance matrices - application to neural network learning. *Physical Review Letters*, 66:2396–2399, 1991.

[LTS90] E Levin, N Tishby, and S A Solla. A statistical approach to learning and generalization in layered neural networks. *Proceedings of the IEEE*, 78:1568–1574, 1990.

[Mac92a] D J C MacKay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.

[Mac92b] D J C MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4:720–736, 1992.

[Mac92c] D J C MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4:590–604, 1992.

[Mac92d] D J C MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.

[McC81] P McCullagh. Discussion of 'Randomized allocation of treatments in sequential experiments' by J A Bather. *Journal of the Royal Statistical Society, Series B*, 43:286–287, 1981.

[MD89] G J Mitchison and R M Durbin. Bounds on the learning capacity of some multi-layer networks. *Biological Cybernetics*, 60:345–356, 1989.

[MEZ93] P Majer, A Engel, and A Zippelius. Perceptrons above saturation. *Journal of Physics A*, 26:7405–7416, 1993.

[Mit82] T M Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[MKC89] R H Myers, A I Khuri, and W H Carter, Jr. Response surface methodology: 1966-1988. *Technometrics*, 31, 1989.

[MPV87] M Mézard, G Parisi, and M A Virasoro. *Spin Glass Theory and Beyond.* World Scientific, Singapore, 1987.

[MS95] G Marion and D Saad. Data-dependent hyperparameter assignment. 1995. *Annals of Mathematics and Artificial Intelligence.* In press; extended version to be submitted to *Journal of Physics A.*

[MT92] W Maass and G Turán. Lower bound methods and separation results for online learning- models. *Machine Learning*, 9:107–145, 1992.

[Mun92] P W Munro. Repeat until bored: A pattern selection strategy. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 1001–1008, San Mateo, CA, 1992. Morgan Kaufmann Publishers.

[Nea93] R M Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[Nea94] R M Neal. Priors for infinite networks. Technical report, University of Toronto, 1994. Available by anonymous ftp from: `ftp.cs.toronto.edu`, file `/pub/radford/pin.ps.Z`.

[OH91] M Opper and D Haussler. Generalization performance of bayes optimal classification algorithm for learning a perceptron. *Physical Review Letters*, 66:2677–2680, 1991.

[OKKN90] M Opper, W Kinzel, J Kleinz, and R Nehl. On the ability of the optimal perceptron to generalize. *Journal of Physics A*, 23:L581–L586, 1990.

[Opp89] M Opper. Learning in neural networks: Solvable dynamics. *Europhysics Letters*, 8:389–392, 1989.

[Opp94] M Opper. Learning and generalization in a two-layer neural network: The role of the Vapnik-Chervonenkis dimension. *Physical Review Letters*, 72:2113–2116, 1994.

[Pil91] J Pilz. *Bayesian Estimation and Experimental Design in Linear Regression Models.* John Wiley, Chichester, 2nd edition, 1991. (1st edition Teubner, Leipzig, 1983).

[PK95] G Paass and J Kindermann. Bayesian query construction for neural network models. In G Tesauro, D S Touretzky, and T K Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 443–450, Cambridge, MA, 1995. MIT Press.

[Plu94] M Plutowski. *Selecting Training Exemplars for Neural Network Learning.* PhD thesis, University of California, San Diego, 1994.

[PR93] F Pukelsheim and J L Rosenberger. Experimental design for model discrimination. *Journal of the American Statistical Association*, 88:642–649, 1993.

[PW93] M Plutowski and H White. Selecting concise training sets from clean data. *IEEE Transactions on Neural Networks*, 4:305–318, 1993.

[Saa] D Saad. General gaussian priors for improved generalization. *Neural Computation.* Submitted.

[SH92] H Schwarze and J Hertz. Generalization in a large committee machine. *Europhysics Letters*, 20:375–380, 1992.

[Sil80] S D Silvey. *Optimal design.* Chapman and Hall, London, 1980.

[SN95] K K Sung and P Niyogi. Active learning for function approximation. In G Tesauro, D S Touretzky, and T K Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 593–600, Cambridge, MA, 1995. MIT Press.

[Sol94] P Sollich. Finite-size effects in learning and generalization in linear perceptrons. *Journal of Physics A*, 27:7771–7784, 1994.

[SOS92] H S Seung, M Opper, and H Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory (COLT '92), Pittsburgh, 1992*, pages 287–294, New York, 1992. ACM.

[Spo95] M Sporre. Unrealizable learning in binary feed-forward neural networks. 1995. Submitted to *Journal of Physics A.*

[SR95] N Szilas and E Ronco. Action for learning in non-symbolic systems. In *Proceeding of the European Conference on Cognitive Science, Saint-Malo, France, April 1995*, pages 55–63, Le Chesnay, France, 1995. INRIA.

[SST92] H S Seung, H Sompolinsky, and N Tishby. Statistical-mechanics of learning from examples. *Phys. Rev. A*, 45:6056–6091, 1992.

[Sto74] M Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36:111–147, 1974.

[Sto77] C J Stone. Cross-validation: A review. *Math. Operationsforschung, Series Statistics*, 9:1–51, 1977.

[SY84] J Sacks and D Ylvisaker. Some model robust designs in regression. *Annals of Statistics*, 12:1324–1348, 1984.

[TM92] S B Thrun and K Möller. Active exploration in dynamic environments. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, San Mateo, CA, 1992. Morgan Kaufmann Publishers.

[Val84] L G Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

[VC71] V Vapnik and A Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probabil. Appl.*, 16:264–280, 1971.

[Wat93] T L H Watkin. Optimal learning with a neural network. *Europhysics Letters*, 21:871–876, 1993.

[Whi89] H White. Learning in artificial neural networks: A statistical perspective. *Neural Computation*, 1:425–464, 1989.

[Wie94] D P Wiens. Robust designs for approximately linear-regression - M-estimated parameters. *Journal of Statistical Planning and Inference*, 40:135–160, 1994.

[Wil95] C K I Williams. Regression with Gaussian processes. 1995. *Annals of Mathematics and Artificial Intelligence.* In press.

[WL92] D H Wolpert and A Lapedes. An investigation of exhaustive learning. Technical Report SFI TR 92-04-20, Santa Fe Institute, Santa Fe, 1992.

[Wol92] D H Wolpert. On the connection between in-sample testing and generalization error. *Complex Systems*, 6:47–94, 1992.

[Wol93] D H Wolpert. On the use of evidence in neural networks. In S J Hanson, J D Cowan, and C Lee Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 539–546, San Mateo, CA, 1993. Morgan Kaufmann.

[Wol95] D H Wolpert. Off-training set error and a priori distinctions between learning algorithms. Technical Report SFI TR 95-01-003, Santa Fe Institute, Santa Fe, 1995.

[WR92] T L H Watkin and A Rau. Selecting examples for perceptrons. *Journal of Physics A*, 25:113–121, 1992.

[WRB93] T L H Watkin, A Rau, and M Biehl. The statistical-mechanics of learning a rule. *Reviews of Modern Physics*, 65:499–556, 1993.

[WS93] K Y M Wong and D Sherrington. Neural networks optimally trained with noisy data. *Physical Review E*, 47:4465–4482, 1993.

[WS95] D H Wolpert and C E M Strauss. What Bayes has to say about the evidence procedure. In G Heidbrecher, editor, *Maximum Entropy and Bayesian Methods (MaxEnt 1994)*, Dordrecht, 1995. Kluwer. In press.

[WSW93] D H Wolpert, C E M Strauss, and D R Wolf. Alpha, evidence, and the entropic prior. In A Mohammaddjafari and G Demoment, editors, *Maximum Entropy and Bayesian Methods (MaxEnt 1992)*, pages 113–120, Dordrecht, 1993. Kluwer.