Statistical Mechanics, Generalisation and Regularisation of Neural Network Models

Alan P. Dunmur

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy
to the
University of Edinburgh
1994

Abstract

There has been much recent interest in obtaining analytic results for rule learning using a neural network. In this thesis the performance of a simple neural network model learning a rule from noisy examples is calculated using methods of statistical mechanics. The free energy for the model is defined and order parameters that capture the statistical behaviour of the system are evaluated analytically. A weight decay term is used to regularise the effect of the noise added to the examples. The network's performance is estimated in terms of its ability to generalise to examples from outside the data set. The performance is studied for a linear network learning both linear and nonlinear rules. The analysis shows that a linear network learning a nonlinear rule is equivalent to a linear network learning a linear rule, with effective noise added to the training data and an effective gain on the linear rule. Examining the dependence of the performance measures on the number of examples, the noise added to the data and the weight decay parameter, it is possible to optimise the generalisation error by setting the weight decay parameter to be proportional to the noise level on the data. Hence, a weight decay is not only useful for reducing the effect of noisy data, but can also be used to improve the performance of a linear network learning a nonlinear rule.

A generalisation of the standard weight decay term in the form of a general quadratic penalty term or regulariser, which is equivalent to a general Gaussian prior on the network's weight vector, is considered. In this case an average over a distribution of rule weight vectors is included in the calculation to remove any dependence on the exact realisation of the rule. As a simple example, the case where the rule weight vector is drawn from a spherical distribution is considered. In this case it is shown that the best performance (lowest generalisation error) for noisy data is achieved with the standard weight decay; the prior distribution of network weights matches the distribution from which the rule weight vector is drawn.

The model is extended to consider different distributions of the rule weights. It is expected that when the penalty matrix models the rule weight distribution, the network's performance is optimised and this is demonstrated for a number of simple examples and is shown to be true analytically for the Gaussian priors under consideration. Hence, more complicated penalty terms can enable the network to take advantage of any known fine structure of the target rule.

A general distribution of input patterns is also considered. This alters the calculation so that the order parameters measure the overlaps between weight vectors weighted by the input distribution. The performance measures are then evaluated in terms of these rescaled order parameters. The optimal penalty matrix is unchanged by the novel input distribution. Some simple extensions of the noise model are also considered, it turns out that these do not alter the form of the optimal penalty matrix either.

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 myself, unless otherwise stated.

Part of the work contained in chapters 3 and 4 has been published in:
A P Dunmur and D J Wallace. Learning and generalisation in a linear perceptron stochastically trained with noisy data. *J. Phys. A*, **26** 5767–5779, 1993.

August 1994

Acknowledgments

I must firstly thank my supervisors, initially David Wallace and latterly David Saad, without their help and encouragement, the thesis would not be what it is now. I must also thank the other members of the neural networks group in the physics department for many helpful discussions and detailed reading of the thesis. They are in no particular order; David Barber, Peter Sollich, Glenn Marion and Ansgar West. I would also like to thank Sandra Gadd for her readings of the thesis.

There are many people who have helped with making the time in Edinburgh pass all too quickly, of these I would especially like to thank Jerry Lloyd, Anne Payne and Ian Cook, for happy days and weekends spent climbing, skiing and eating and Karen Dobbie for providing entertainment during lunchtime.

I must also thank my family; my parents for providing the inspiration and the support when needed. Last and by no means least, I must thank Rachel Kenworthy for putting up with everything and allowing me to get to know the road to Aberdeen so well.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A neural network model is a computational machine that consists of a collection of simple processing units each of which simulates a biological neuron to some extent. There has been much recent interest in "neural networks" both within the physics community and beyond. Neural networks were initially motivated as models of brain function and may be studied as such [1]. However, nowadays there is at least as much interest in the application of networks to computational tasks; this aspect of neural networks is usually called **neuro-** or **neural computation**, for a good physicist's review see [37]. Neural networks have been applied very successfully to some difficult computational tasks, *e.g.*, [68, 46]. In order to solve a problem, an algorithm is used to fix the network parameters such that the network performs the desired task. The algorithm itself has additional parameters (sometimes called **hyperparameters**) associated with it, which are often set by *ad hoc* methods that are based on empirical observations. It would be useful to be able to specify the hyperparameters for a particular task by rigorous criteria

such that the performance of the generated network is optimised.

Analysing the capabilities and performance of a neural network is an extremely difficult problem and many different approaches have been explored. Some of them are more empirical, *e.g.*, cross validation [51], Bootstrap, Jackknife [17], whilst others are more theoretical in approach, *e.g.*, the Bayesian formalism [49, 50], PAC [35], VC dimension [69]; another general approach is to use some of the methods of statistical physics [70]. The insights gained by these methods can be used to formalise rules of thumb as well as to suggest novel algorithms and networks. In order to be able to quantify how well a particular network has solved a task, it is necessary to have a measure of the network's performance. There are many possible performance measures that may be used; the actual measure chosen is dependent on the type of task under study. It may be that a network is being used as an associative memory (see *e.g.*, [37]): in this case one possible performance measure is the number of patterns the network can "memorise" without losing information, another might be the proportion of a corrupted pattern that can be recalled correctly. Another possible task could be to learn a hidden rule from a set of examples. A performance measure could be whether the network could generate the correct response for a new example. This ability (or lack) to perform on a novel example is called **generalisation**. Generalisation is a broad concept that covers many different tasks including regression and classification problems, therefore there are many different generalisation measures.

In this thesis, the performance of a simple network model is studied in terms of its ability to generalise. The formalism used to study the model is based on the standard statistical physics replica method [15] and follows the work of Seung *et al* [64]. This formalism is then used to investigate the effect of different hyper-parameters on the generalisation capability of a simple network. The task is made

more complicated by the addition of noise to the data in the problem. A specific performance measure is chosen and optimal hyper-parameters are identified that give the best generalisation. The remainder of this chapter contains a brief history of the field of neural networks, an introduction to a neural network model and an outline of the rest of the thesis.

1.1 Historical background

The initial motivation for neural network models was to use them as models of brain function. The brain is made up of approximately 10^{10} nerve cells [6] (called **neurons**) of many different types that are connected together in large clusters. In 1943 McCulloch and Pitts [52] proposed a simple nonlinear model of a neuron. Any number of these simple neurons could then be connected together by modifiable parameters to produce large networks. The essential feature of this model was the nonlinearity of the neurons and the modifiable parameters which could be set by some algorithm.

The "Hebb rule" (so called because it is based on a hypothesis of Hebb [36] that postulated how a brain learns) is an algorithm whereby a simple network can store or **memorise** a set of patterns, that is given a pattern as an input to the network, the corresponding output is produced. This algorithm is successful up to a limiting number of patterns [3, 4] after which the quality of the recalled patterns degrades rapidly. Since this algorithm is not trying to learn the underlying structure of the data it cannot be expected to perform well on rule learning problems. In 1962 Rosenblatt [59] presented an algorithm that could set the network parameters (**train** the network) to solve a simple two class classification problem

and proved that if it was possible to separate the two classes in input space using a hyperplane, *i.e.*, the task was **linearly separable**, the algorithm would solve the problem in finite time. Provided the task was solvable, the network could perform well on novel examples – it had good generalisation.

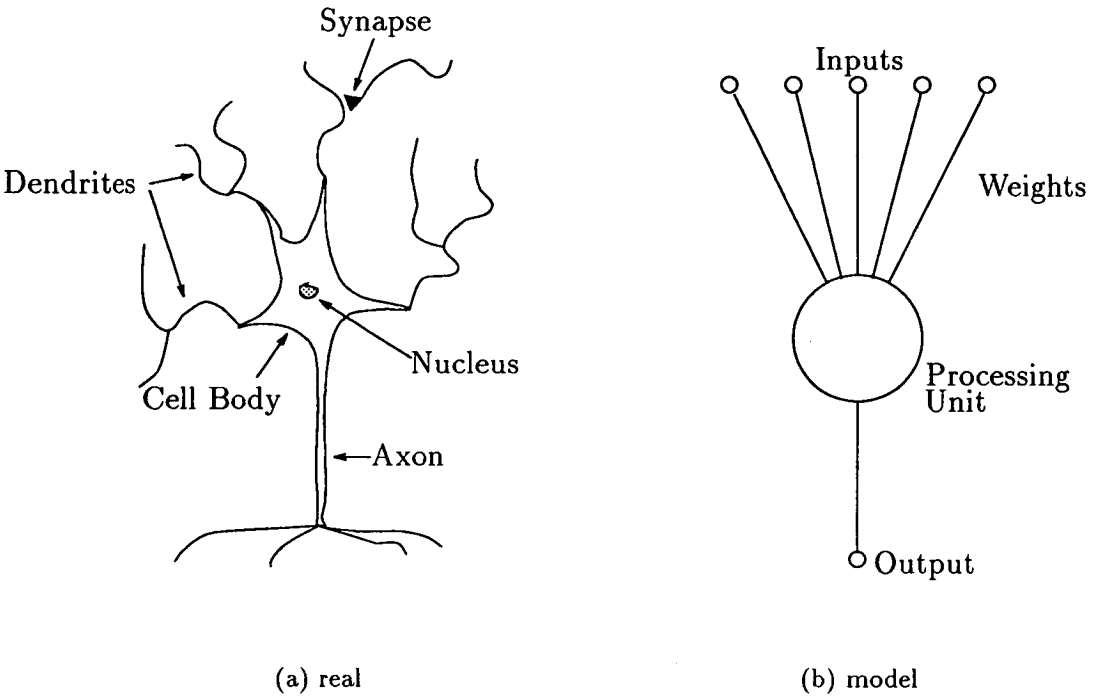
In 1969 Minsky and Papert examined the capabilities of simple networks; their study [55] suggested that simple neural network models have poor classification capabilities for all but the simplest of problems. This slowed research into the use of networks as classifiers and hence the field of neural computation. The invention (and reinvention) of **back propagation** [73, 61, 60, 57] meant that there existed a systematic algorithm for setting network parameters for networks of arbitrary configurations. This negated some of the criticisms Minsky and Papert had leveled at simple networks since these larger networks were able to solve more complicated tasks. Since then there have been other algorithms that are better at training, some new, *e.g.*, Quickprop [18]; some more traditional, *e.g.*, conjugate gradient [20, 28], *etc.*.

Physicists first became interested in neural networks when Hopfield [39] pointed out that a particular model of memory could have an energy function defined for it, this meant that the methods of statistical physics could be used to optimise its performance in terms of the number of patterns it could store [2]. This branch of research was pushed forward by Gardner who used methods from the study of spin glasses to gain additional insights into the behaviour of this type of network [25, 26]. Recently physicists have become interested in using the methods of statistical physics to study the **rule learning** problem [70]. A formalism for studying a simple network learning a rule based on the methods used by Gardner was first introduced by Györgi and Tishby [31] and further developed in Seung *et al* [64]. For simple network models, a different formalism was developed by Hertz,

Krogh, and Thorbergsson [38] which uses the spectrum of eigenvalues. These formalisms based on the methods of statistical physics along with other more probabilistic methods have allowed theoretical results for network performance to be obtained. These results can be useful for selecting appropriate networks and training algorithms for real world applications [37].

1.2 A brief introduction to neural networks

A "neural" network can be defined as a collection of simple processing units connected by a set of modifiable parameters. The simple processing units are modelled on biological neurons. There are many different types of neuron; a schematic diagram of a typical neuron is presented in Fig. 1.1(a). The neuron consists of a set of **dendrites** connected to the cell body which extends into an **axon**. The axon subsequently branches into smaller strands which interact with dendrites from other neurons. At the interaction, there are **synapses** which transmit a signal to other cells. The transmission of a signal through a synapse is a very complicated chemical process, the action of which is to raise or lower the electrical potential of the dendrite. The potential from all the dendrites is accumulated in the cell body or **soma**. If this potential reaches a certain threshold, the cell "fires" and transmits an electrical pulse down the axon. The pulse is a nonlinear function of the cell potential and may in turn be passed through synapses to other cells. The brain is assumed to "learn" through the process of adapting the synapses between neurons which then transmit more or less of the pulse to the next cell.



(a) real (b) model

Figure 1.1. Schematic diagram of a real and model neuron.

The two main characteristics of a biological neural network are the high adaptive connectivity between neurons and the nonlinearity present in a neuron. A schematic diagram of of a simple model of a neuron is presented in Fig. 1.1(b). Here a set of **inputs**, corresponding to the signals arriving at the synapses, is connected by a series of modifiable **weights** corresponding to the synapses themselves to a processing unit. The processing unit takes the weighted sum of the inputs to give the potential or **activation** of the unit. If this activation is greater than some threshold, the cell fires and transmits a pulse, if the activation is below threshold, the cell does nothing. Mathematically, this may be written as

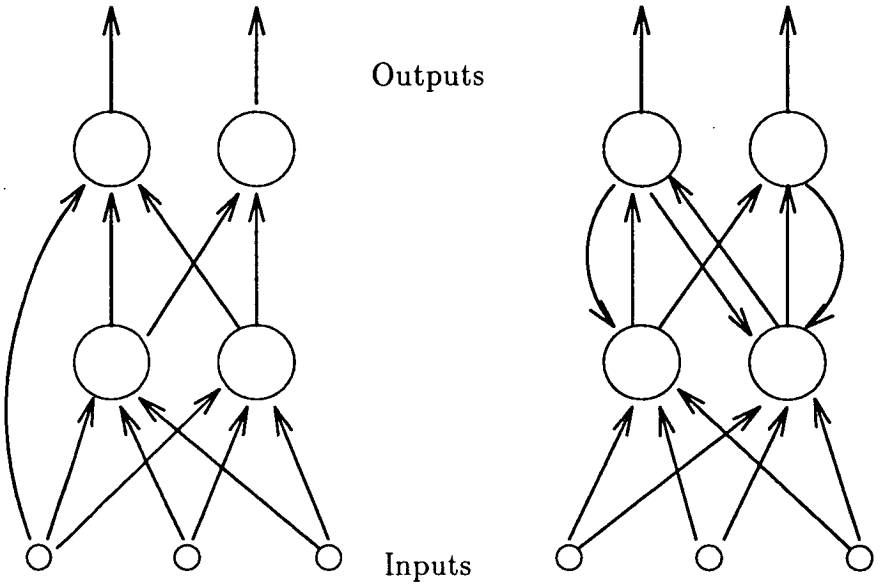
$$\sigma = \Theta(\sum_j W_j s_j - \theta),$$

where σ is the output of the unit, $\mathbf{W}$ is a vector of the weights, $\mathbf{s}$ is the vector

of inputs, θ is the threshold of the unit and $\Theta(x)$ is the Heaviside step function, that is $\Theta(x)$ equals one if $x > 0$ and zero if $x < 0$. The behaviour for $x = 0$ is undefined. The size of the pulse has been renormalised to one to make the equation simpler, but a pulse of arbitrary size could be included in the model. The components of the weight vector W_j may be positive or negative. This simple unit was the network for which Rosenblatt produced his learning rule and is known as a **perceptron** or binary perceptron to distinguish it from the models discussed in the next paragraph.

A simple generalisation of the perceptron is to replace the Heaviside step function $\Theta(x)$ with some arbitrary **activation function** $g(x)$. If $g(x)$ is simply a linear function, the network may be called a linear perceptron. A set of generalised perceptrons can be connected together to produce a powerful network that may be used for many diverse computational tasks. It can be shown that any arbitrary input/output mapping may be approximated by a network with a sufficient number of units and a particular **architecture** [40], [23].

The architecture of a network is the number and type of units and the structure of the connections between them. The parameters of the network are the weights between units and the thresholds of these units. If the architecture of the network can be redrawn such that the connections between nodes are unidirectional from inputs to outputs, the network is said to be **feed forward**, as in Fig. 1.2(a), if no such redrawing is possible, the network is called **recurrent**, as in Fig. 1.2(b). If the units in the feed forward network are perceptrons, then this network is an example of a **multi-layer perceptron** (MLP). In a MLP, there may be some nodes that are not inputs and are not directly connected to the output, these nodes are often referred to as **hidden nodes**.



(a) Feed–forward (b) Recurrent

Figure 1.2. Feed-forward and recurrent networks

MLPs are much more useful than single layer perceptrons; as an example, consider a mapping that cannot be learnt by a simple perceptron, but may be solved using a 2 layer perceptron; the logical XOR function (table 1.1). A network that is capable of solving the XOR problem is presented in Fig. 1.3. Since the weights are continuous, there is an infinite number of different weight and threshold settings that would also solve the task for the architecture used. There is also an infinity of other network architectures that may be used to solve the problem.

When using a network to solve a problem, the number of inputs and outputs is set by the data, however, as mentioned before, there are an infinite number of possible networks that may be placed between the inputs and outputs. It is very important to select a network with a suitable architecture (correct number of hidden nodes,

Inputs		Output
0	0	0
1	0	1
0	1	1
1	1	0

Table 1.1. The logical XOR function.

layers, etc) for the problem. At present the selection of a network architecture is something of a black art, although there do exist systematic algorithms for selecting a network architecture for some tasks. Some of these start from a simple network and generate a network that is capable of solving the problem by adding nodes and weights as needed, *e.g.*, upstart, [22], cascade-correlation [19], These methods are known as **constructive** since they construct a network to solve the problem. One possible drawback with these methods is that there is no reason to believe that the solved network will be the "best" solution in terms of simplicity or training time *etc.* Another group of methods are the destructive or **pruning** methods, *e.g.*, optimal brain damage [47], *etc.* These methods start with a network that is larger than needed and remove those parts of it that are unnecessary. Since the algorithms remove the least used parts of the network, it is reasonable to suppose that the final network will be close to optimal in terms of its connections and training times *etc.* Both the constructive and pruning methods contain an intuitive preference for simpler models (Occams razor).

The behaviour of the network depends on the settings of the weights between the nodes as well as the architecture. There are many possible algorithms for fixing the weights of a network which depend on the task the network is being used to solve. Some algorithms used for simple tasks are able to determine the network parameters directly, *e.g.*, the Hebb rule and the pseudo inverse solution

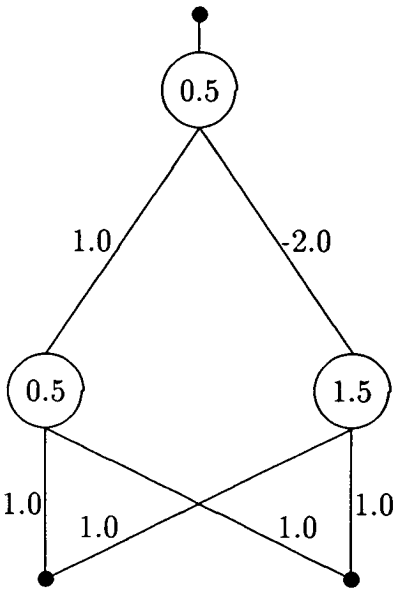


Figure 1.3. Network to solve logical XOR. The units have a step-function activation and output a 1 if the activation is greater than the threshold (the value inside the unit) and 0 otherwise. The weights connecting the units are the values beside the connections.

[37]. However, for the majority of tasks it is not possible to set the weights *a priori*. In these cases it may be possible to train the network to perform the desired task through iterative updates of the weights. There are two main classes of algorithm: If the task is presented as a set of data that gives the "correct" network response for a particular input, then the weight modifications may be done using a direct comparison between the actual and desired response, this is known as **supervised learning** [37]. If on the other hand, the task is presented as a set of inputs and the task of the network is to extract whatever correlations between the inputs it can, this is known as **unsupervised learning** *e.g.*, [42]. A third class of algorithm that sits between the two main classes mentioned above is that of **reinforcement learning** where the training algorithm receives

limited feedback during training, *e.g.*, whether the networks output is correct or incorrect.

During training it is useful to know how well the network is doing, either relative to its previous performance or absolutely for unsupervised and supervised training respectively. A **cost function** which gives a measure of the networks performance can be defined. Training is then simply the process of finding a minimum of this cost function. Ideally for the "best" performance, the minimum should be global, though this is not always possible and almost always cannot be identified.

There are an enormous number of possible tasks to which neural networks may be applied. Broadly, these may be split up into;

- Associative memory – the task is to learn a set of examples or patterns and be able to recall a particular pattern when the corresponding input is presented or (more usefully) when a corrupted input is presented. A useful cost function or performance measure may be the overlap between the retrieved memory and that stored.
- Control – learn a mapping from some initial state to a final state. There may be many possible paths between the two states. The inputs may be the state variables for a complex process and the outputs are the control variables. The path between the initial state to the desired final state is unimportant. The cost function could measure how close the process is to the desired state [58].
- Prediction – given a series of data predict the next term(s) in the series. If the next term in the series is known a cost function may be defined that gives a measure of the difference between the actual prediction and the

desired prediction [72].

- Classification – given a set of inputs, decide into which of k classes the input belongs. A data set consisting of inputs and the correct classes can then be used to define a cost function for the network. The cost function could be simply the number of misclassified patterns which can then be used to train the network [45, 46].

- Regression – fit a model to a set of data. The model may then be used for interpolating or extrapolating to new data points. A possible cost function is the difference between the predicted point and the actual position [49].

For all these learning problems, it is possible that the data set is corrupted by some noise process making the task of training the network harder. A regulariser on the network parameters can help to counteract the effect of noise.

Evaluating the theoretical performance of a network is a very complicated task, a number of different approaches have evolved. One of the more successful is the **Bayesian** approach [49]. Simple statistical arguments are used to evaluate probability distributions for weights and models given a data set and any other prior information. The Bayesian method can be shown to include Occams razor [49]. The **PAC** (Probably Almost Correct) method [35] looks at the worst case scenario of a learning task and uses this result to provide an upper bound on the performance of the network. The **VC dimension** approach [69] uses counting arguments to specify a measure of the number of possible functions the network can model. This can then be used to specify upper bounds on the error measures. The **statistical mechanics** approach is to calculate the average performance by first calculating the average free energy for the problem and using this to evaluate

order parameters that capture the statistics of the model. This method will be used in the thesis.

1.3 Outline of Thesis

Theoretical results for the performance of networks are useful for selecting appropriate networks and algorithms for specific problems. It is possible to calculate the performance of simple networks, [48, 64, 38], by making some assumptions about the model being studied. In chapter 2 the statistical mechanics formalism [64] that will be used is introduced. The network model is a simple perceptron that is trained on a set of data generated from a known perceptron rule. This enables comparisons between the network that is learning the task and the rule that produced the data to be drawn.

It is likely in a real world situation that the data used to train the network has become corrupted by some noise process. Chapter 3 extends the formalism developed in the previous chapter to allow for a data set which has been corrupted by noise. One possible method of counteracting the noise on the data is a weight decay which adds a regularising component to the training process. This has been studied previously by Krogh and Hertz [43] using a different method. In this chapter, the performance of the network is calculated using the formalism outlined in the previous chapter allowing some additional insights into the behaviour of the network to be gained. A prescription for fixing the optimal regularising parameter in terms of the networks generalisation ability is presented. Chapter 4 presents the behaviour of the model for various limits of the hyperparameters of the training algorithm.

A weight decay is shown to be equivalent to a penalty term added to the cost function. Since the effect of the weight decay is to improve the performance in the case of noisy data, a generalised penalty term may be able to increase the performance further. The calculation developed in chapter 3 is extended in chapter 5 to consider a general regularising term added to the training process. An average over a distribution of teachers is included to remove a dependence on the explicit teacher being used. The behaviour of the network is calculated for an isotropic teacher distribution and some simple regularisers. In chapter 6, the model used to calculate the performance is extended to include anisotropic Gaussian teacher distributions and anisotropic distributions of inputs. The effects of the different teacher and input distributions are discussed in terms of how well the general penalty terms perform. Finally, different noise models and the effect of general penalty terms are discussed briefly. Chapter 7 gives a short summary and presents the main conclusions of the thesis.

As an aid to the reader the notation used in the thesis is listed in Appendix A.

Chapter 2

Statistical mechanics of learning

It would be useful to know how a particular network architecture is likely to perform on different problems. This knowledge could then be used to pick the class of network that would give the "best" performance without having to experiment and waste time training sub–optimal nets. A network performs a mapping from its inputs to its outputs that depends on the architecture, the activation functions of the "neurons" and a set of network parameters – the weights. For rule learning problems, the weights may be evolved using some algorithm (trained) so that the network approximates a target mapping. Training usually tries to optimise a cost function that describes how well the model approximates the target mapping. There are many possible methods of minimising the cost function with respect to the weights; some perform a search through state space for a minimum of the cost function, *e.g.*, genetic algorithms [29], simulated annealing, [41] , etc. Others use some sort of gradient descent to find a minimum, *e.g.*, backpropagation [37]. In this chapter, cost or error functions that measure a network's

performance on a rule learning problem are introduced; a training algorithm that generates a distribution of networks is defined, this allows the concept of average performance measures to be introduced. Following Seung *et al* [64] the average performance for a simple network is calculated using terminology and methods from statistical mechanics.

2.1 Errors

Consider a network, $\mathcal{N}_{\mathbf{W}}$ with a particular architecture and a set of weights denoted by the vector, $\mathbf{W}$. The network maps a vector of inputs, $\mathbf{s}$, to a single output node, $\sigma_{\mathbf{W}}$. The case where the output is more than one node can be split up into a number of different networks with the same inputs and different outputs. The mapping defined by the network can be written as,

$$\sigma_{\mathbf{W}}(\mathbf{s}) = \mathcal{N}_{\mathbf{W}}(\mathbf{s}) \; . \tag{2.1}$$

For a particular network and an example consisting of an input $\boldsymbol{\xi}$, and corresponding output, ζ, it is possible to measure how well the network approximates the example by comparing the network's output with the example output. This measure is a function of the network's architecture and weight vector and the example. In the simplest case, the error is defined to be zero if the network correctly models the example and one if the network's output disagrees with the example output. The error measure may also be defined in such a way as to represent a distance between the network's output and the example output,

$$\epsilon(\mathcal{N}_{\mathbf{W}}; \boldsymbol{\xi}, \zeta) = f(|\sigma_{\mathbf{W}}(\boldsymbol{\xi}) - \zeta|) \; ,$$

where $\sigma_{\mathbf{W}}$ is the network output parametrised by $\mathbf{W}$. The measure $|\ldots|$ gives a distance between the example output and the network output and f is some monotonically increasing function. The function $\epsilon(\mathcal{N}_{\mathbf{W}};\boldsymbol{\xi},\zeta)$ is then a measure of the error of a given network, $\mathcal{N}_{\mathbf{W}}$, for a particular example and is called the error measure. It is natural to set $f(0)$ equal to 0 so that when the network and the target agree, the error measure is zero. Since $f(x)$ is defined in terms of a distance, the error measure is an even function, one simple choice for f is $f(x) = \frac{1}{2}x^2$, this choice has the advantages that it is continuous and differentiable everywhere which can be essential for some training algorithms. Other choices are possible and have been studied [66]. Using the quadratic form of f gives the error measure as,

$$\epsilon(\mathcal{N}_{\mathbf{W}};\boldsymbol{\xi},\zeta) = \frac{1}{2}(\sigma(\boldsymbol{\xi}) - \zeta)^2 \,. \tag{2.2}$$

This definition of the error measure will be used in later calculations. Since one particular architecture of network will be considered, the error measure may be written as $\epsilon(\mathbf{W};\boldsymbol{\xi},\zeta)$.

Given a set of examples which define the target mapping, the error measure may be summed to give a total error for the whole example set. This data set, often called the **training set**, Θ, contains p examples θ consisting of input output pairs. In this case the total error on the training set is termed the **training error**, E_t,

$$E_t(\mathcal{N}_{\mathbf{W}};\Theta) = \sum_{\theta\in\Theta} \epsilon(\mathcal{N}_{\mathbf{W}};\theta) \,. \tag{2.3}$$

This error gives no indication of how the network would perform on examples which are not members of the training set, this could be obtained using a data set consisting of examples drawn from outside the training set and would measure how well the network was able to generalise to novel examples. In this case, the

error measure is summed over a test set, Φ to give the test error, which is defined in a similar manner to the training error.

$$E_{\text{test}}(\mathcal{N}_{\mathbf{W}};\Phi) = \sum_{\theta\in\Phi} \epsilon(\mathcal{N}_{\mathbf{W}};\theta) \,.$$

The test error gives a measure of the network's performance on examples drawn from the test set. If the test set was a subset of the training set, the test error would give no new information about the network's performance on unseen examples (its generalisation ability). Hence for the test error to give useful information about a network's generalisation ability, the test set should not be a subset of the training set, [75].

The network is trained to approximate the mapping from the inputs to the output described by the training set using an algorithm that minimises a cost function defined on the training set. The examples in the training set may only partially describe the full mapping. However, it is hoped that by minimising the cost function the network will be able to learn the target mapping and hence give a low test error. It could be that the mapping described by the training set is a random mapping, *i.e.*, there is no correlation between the inputs and the corresponding output. In this case, the network may be able to store all the examples in the training set giving a training error of zero, however it would not be able to learn anything about novel examples and would thus have a non–zero test error. The task of storing a random mapping in a network has been extensively studied as a model of memory [1].

If the mapping or rule connecting the inputs to the output is non–random, it would be of more interest to know how well the network has learnt the mapping. In this case, if the test set were the whole of example space a test error error of

zero would imply that the network had learnt the rule exactly. The network's test error on the whole of example space is known as the **generalisation** error. A test error for a finite test set gives a sampled approximation to the generalisation error which improves as the size of the test set increases.

A finite set of data could be partitioned into a training set and a test set. There is a trade off between the number of examples used to train the network which may give a "better trained" network and the number of examples in the test set which give a better estimate of the generalisation error. There are a number of methods which can be utilised to improve the performance on a finite data set, *e.g.*, cross validation [34], bootstrap and jack-knife [16, 17].

2.2 Training

The process of training a network is an algorithmic method of minimising some cost function, E_c, with respect to the network parameters, the weights $\mathbf{W}$. For a particular architecture of network the cost function can be defined in terms of the network weights, $\mathbf{W}$, and the training data set Θ; a simple cost function could be the training error, E_t, defined in equation (2.3). There are many possible methods of minimising the cost function [37]. One of the simplest is gradient descent. The weight vector is modified according to,

$$\frac{\partial \mathbf{W}}{\partial t} \propto -\nabla E_c(\mathbf{W};\Theta)\,. \tag{2.4}$$

This equation updates the weight vector in the direction of steepest descent of the cost function. Depending on the initial conditions of the weight vector and

the form of the cost function there may be a situation where the weight vector becomes stuck in a local minima of the cost function. This can give a sub–optimal solution to the learning problem.

If noise were added to the update equation, there is a finite probability that a network will be able to escape local minima and possibly find the optimal solution, by the same token, it will not however necessarily remain in the global minimum. It has been shown that adding noise to the update equation can increase a network's performance in terms of its ability to generalise [27]. This is known as stochastic training which is now considered in more detail.

During stochastic training, the weights are evolved by following a standard gradient descent of the cost function, E_c and perturbing the updates with additive zero mean noise. In this case, the weight vector updates are given by a Langevin equation,

$$\frac{\partial \mathbf{W}}{\partial t} \propto -\nabla E_c(\mathbf{W};\Theta) + \boldsymbol{\eta}(t) \,, \tag{2.5}$$

where $\boldsymbol{\eta}(t)$ is zero mean dynamic noise with variance given by

$$\langle \eta_i(t)\eta_j(t')\rangle = 2T\delta_{ij}\delta(t-t') \,,$$

where T is a measure of the amount of noise. Ideally, to get to the global minimum, training is initiated with a large value of T and the amount of noise is then slowly reduced to zero using an annealing schedule so that finally, the network parameters are globally optimal [41].

Asymptotically, as time $t \rightarrow \infty$, for a fixed amount of noise added to the updates, the stochastic update rule results in an equilibrium probability distribution for

the weights [12] given by,

$$P(\mathbf{W}) = \frac{e^{-\beta E_c(\mathbf{W};\Theta)}}{\mathcal{Z}} , \tag{2.6}$$

where $\beta = \frac{1}{T}$. This distribution is well known in statistical physics and is called the Gibbs distribution; by analogy with thermodynamic terminology, T is called the temperature. It can be seen from equation (2.5) that the effect of T is to blur the trajectory of the network's weight vector through weight space. If the network is stuck in a local minima, the noise enables the weight vector to jump out of the local minima and possibly reach a lower minimum of the cost function. The partition function, $\mathcal{Z}$, is given by,

$$\mathcal{Z} = \int d\mathbf{W} \exp[-\beta E_c(\mathbf{W})] , \tag{2.7}$$

and normalises the equilibrium probability distribution, $P(\mathbf{W})$ so that the integral over weight space is equal to one.

The Gibbs distribution may also be motivated by appealing to information theoretic methods. Consider an observable, $\hat{O}(X)$, which depends on a random variable, X, drawn from a state space, Ω, and has an average value $\langle O \rangle$. One wishes to calculate the distribution of X which gives this average value, denoted $P(X)$. The standard information theoretic method to calculate $P(X)$ is the Maximum Entropy method [65] where the entropy, S, defined by

$$S = -\sum_{X \in \Omega} P(X) \ln P(X) , \tag{2.8}$$

is maximised with respect to $P(X)$, subject to the constraints that the probabilities sum to one and the distribution gives the average value of the observable.

The constraints may be written as:

$$\begin{aligned} \sum_{X\in\Omega} P(X) - 1 = 0 &= f(X)\,, \\ \sum_{X\in\Omega} P(X)\hat{O}(X) - \langle O\rangle = 0 &= g(X)\,. \end{aligned}$$

The entropy is maximised by introducing Lagrange multipliers α and β for the constraints f and g respectively. For the entropy to be a maximum,

$$dS + \alpha\, df + \beta\, dg = 0$$

Substituting the entropy and constraints into equation (2.8) and taking out a factor of $dP(X)$ leads to,

$$P(X) \propto \exp[\alpha - 1 + \beta\hat{O}(X)]\,.$$

Absorbing the factor $e^{\alpha-1}$ into the normaliser for the probability and assuming that the observable is an energy gives the Gibbs distribution. Equation (2.6) can therefore also be interpreted as an information theoretic form of the posterior weight distribution given a value of the cost function. The asymptotic post-training distribution of network parameters is known and hence averages over this distribution may be calculated.

2.3 Average performance measures

For a particular architecture of network it would be useful to give an estimate of its performance that was independent of the network parameters and the training

set used to evolve the parameters. Using the stochastic training rule eq. (2.5) a Gibbs distribution of weights is generated asymptotically eq. (2.6); if the training error eq. (2.3) is averaged over this posterior distribution, the result is an average error for an ensemble of networks trained on the same example set. The averages over the Gibbs posterior distribution are called thermal averages, denoted $\langle \ldots \rangle_T$, because they are averaging over the thermal noise introduced in the stochastic training.

Once a quantity has been averaged over the posterior distribution of weights, it still contains a dependence on the training set, Θ, since all instances of the posterior weights are generated from the same training set. This dependence may be removed by performing a second average over the training set. This average is over parameters that are "frozen" during training, hence the average is known as a quenched average and is denoted $\langle\!\langle \ldots \rangle\!\rangle$. The exact form of the quenched average depends on the parameters used to generate each training example. If each example input output pair, $(\boldsymbol{\xi}, \zeta)$ is denoted θ and the training set contains p such examples, the quenched average of a quantity $K(\Theta)$ is

$$\langle\!\langle\ K(\Theta)\ \rangle\!\rangle = \int \prod_{l=1}^{p} d\mu(\theta^l)\ K(\Theta)\ .$$

where $d\mu(\theta^l)$ contains the distribution of the quenched parameters used to generate the l^{th} example in the training set.

The quenched average of a posterior averaged quantity then gives an average quantity that is independent of the example set used, but dependent on the training algorithm, the distribution of examples in the training set and the size

of the training set. The average training error, ϵ_t can be defined as

$$\epsilon_t = \frac{1}{p} \langle\!\langle \langle E_t(\mathbf{W};\Theta)\rangle_T \rangle\!\rangle \ ,$$

where p is the number of examples in the training set Θ. The average training error gives an expected error for the network on an example drawn from a typical training set. In the absence of any other information, it is one possible performance measure. However, it does not give any information on the expected error for an unseen example. The average training error may be useful if one is only interested in storing patterns in the network. The average cost function ϵ_c may be similarly defined as the quenched average of the thermal average of the cost function.

The generalisation error introduced earlier is the network's test error on the whole of example space, that is the test set Φ is taken to be the complete example space. The measure on example space, $d\mu(\theta)$, that was used for the quenched average can be used to define the **generalisation function**

$$\epsilon(\mathbf{W}) = \int_{\Phi'} d\mu(\theta)\, \epsilon(\mathbf{W};\theta) \ , \tag{2.9}$$

where $\theta = (\boldsymbol{\xi}, \zeta)$ is an example input–output pair and Φ' is the complete example space. The generalisation function gives the expected error of a particular network for a random example picked from example space according to the measure $d\mu(\theta)$.

If there exists a weight vector, $\mathbf{W}^*$, such that $\epsilon(\mathbf{W}^*) = 0$, the network is capable of learning the underlying rule exactly. In this case the problem is said to be realisable. If $\epsilon(\mathbf{W}) > 0$ for all possible $\mathbf{W}$, the problem is said to be unrealisable.

The average generalisation error, ϵ_g can be defined from the generalisation function in a similar manner to the average training error. It is defined by

$$\epsilon_g = \langle\!\langle \langle \epsilon(\mathbf{W}) \rangle_T \rangle\!\rangle \ , \tag{2.10}$$

This error gives an estimate of the network's performance that is independent of the actual training set used. Like the average training error it depends on the training algorithm, the distribution of examples in example space and the size of the training set. Other forms of average generalisation error are possible, *e.g.* that used by Bruce and Saad [9].

The essential difference between the generalisation error defined above and that used by Bruce *et al* is the position of the thermal averaging over the posterior ensemble of weight vectors. Explicitly for a quadratic error measure, the average generalisation error, eq. (2.10) is,

$$\epsilon_g = \frac{1}{2} \left\langle\!\!\left\langle \ \left\langle \ \left\langle (\sigma_{\mathbf{W}}(\boldsymbol{\xi}) - \zeta)^2 \right\rangle_\theta \ \right\rangle_T \ \right\rangle\!\!\right\rangle ,$$

where the inner average $\langle \cdot \rangle_\theta$ is over the distribution of examples and $(\boldsymbol{\xi}, \zeta)$ is an example input–output pair. This can be compared to that used by Bruce *et al*

$$\epsilon_{BS} = \left\langle\!\!\left\langle \ (\langle \sigma_{\mathbf{W}}(\boldsymbol{\xi}) - \zeta \rangle_T)^2 \ \right\rangle\!\!\right\rangle . \tag{2.11}$$

The two generalisation errors also differ by a factor of a half due to different conventions in the definition of the errors. Thus ϵ_g measures the average error of the posterior rules, whereas ϵ_{BS} measures the error of the average posterior rule. In statistical terminology, ϵ_g measures the mean square error (MSE) and ϵ_{BS} measures the bias squared (MSE = bias^2 + variance). This difference will

affect the behaviour that each performance measure favours (see §3.8)

The average training error and the average generalisation error give performance measures for a network that are dependent only on the architecture of the network, the training algorithm and the distribution of training examples. A method of calculating these errors will now be presented.

2.4 The free energy

The network's performance measures defined in the previous section are average measures, the question arises as to whether these averages are typical for any instance of the system, *i.e.*, whether the variance of the quantities is small. This can be related to the question of self averaging in statistical physics [7], in this case, the self averaging quantity studied is the free energy F defined by,

$$F = -T \langle\langle \ln \mathcal{Z} \rangle\rangle \ , \tag{2.12}$$

where $\mathcal{Z}$ is the partition function. It can be shown that the free energy has fluctuations $O(\frac{1}{N})$ for large N [7].

The free energy is related to the performance of the network by,

$$F = p\,\epsilon_t - TS \ , \tag{2.13}$$

where the entropy S has been introduced;

$$S = -\left\langle\!\!\left\langle \int d\mathbf{W} P(\mathbf{W}) \ln P(\mathbf{W}) \right\rangle\!\!\right\rangle \ .$$

Equation (2.13) is identical to the standard thermodynamic identity of statistical physics, [44], with the average energy replaced by p, the number of examples, times the average training error. From the free energy the average training error ϵ_t can be calculated.

Using eq. (2.13) the average training error is related to the free energy by,

$$\epsilon_t = \frac{1}{p}\frac{\partial(\beta F)}{\partial \beta} .$$

A similar result holds for the entropy.

The calculation of the free energy is a well known problem of statistical mechanics and there are many possible methods available. Amongst these are:

- The high temperature limit. The exponential in the partition function may be expanded as a power series in β. It gives the correct behaviour of the system for high temperatures (small β) but breaks down at low temperatures. This limit has been used in the study of learning, although, in most cases of interest, the noise level (temperature) is small and this method is inappropriate.

- The annealed approximation. Make the approximation, $F = -T \ln \langle\langle\ \mathcal{Z}\ \rangle\rangle$. This approximation gives a lower bound on the free energy and gives accurate results for high temperatures. It has been used with some success in the study of simple multilayer perceptrons [63]. The method does however break down for lower temperatures.

- The replica method [15]. This method is rather more complicated then

either of the previous two approximations, however it is valid for low temperatures. The complexity of the replica method may introduce more problems than it solves, since in order to simplify the calculation assumptions about the symmetry of the solutions have to be made. It turns out that the symmetry assumptions are inappropriate for some models [21].

The replica method will be used in this thesis since it gives analytic expressions for network performance at low levels of dynamic noise. Before the replica method is outlined, the network model used in the calculation will be defined.

2.5 The model

In order to simplify the calculations greatly, the simplest type of network, a single layer perceptron will be studied. The distinction between single and multi–layer perceptrons (see §1.2) will not be needed as only the former are under study and so the term perceptron will be taken to mean single layer perceptrons. The inputs, $\mathbf{s}$, and parameters of the network, $\mathbf{W}$, are assumed to be continuous valued variables. The mapping described by a perceptron, $\mathcal{N}_{\mathbf{W}}$, can be written as,

$$\sigma(\mathbf{s}) = g\left(\frac{1}{\sqrt{N}}\mathbf{W}\cdot\mathbf{s}\right) , \tag{2.14}$$

where σ is the perceptron output, $\mathbf{s}$ is a vector of inputs, $\mathbf{W}$ is a vector of the network parameters and $g(\cdot)$ is known as the **activation function**. The number of inputs is equal to the number of weights, N. The product $\mathbf{W}\cdot\mathbf{s}$ is sometimes called the **activation** of a unit.

The training or example set is assumed to have been generated from an "unseen" **teacher** network. The teacher network is another network, $\mathcal{V}$, that is parametrised by a weight vector $\mathbf{W}^0$. The output of the teacher, ζ, for an input $\boldsymbol{\xi}$ is

$$\zeta = \mathcal{V}_{\mathbf{W}^0}(\boldsymbol{\xi}) ,$$

where the teacher rule has been modelled by an architecture $\mathcal{V}$, parametrised by a set of weights $\mathbf{W}^0$. Using a teacher network each example can be generated from an input, hence the example space can be generated from the input space alone. The measure on example space, $d\mu(\theta)$ is now equal to the measure on input space $d\mu(\mathbf{s})$.

Following the analogy of the teacher, the network trying to learn the rule, $\mathcal{N}_{\mathbf{W}}$, is called the **student**. The student is trained on a set of examples generated from the teacher and randomly generated inputs picked from the input distribution.

The only knowledge the training algorithm and hence the student has of the teacher is through the example set. However, when the performance of the student is evaluated, it is useful to compare the final student network with the teacher. In order to keep the comparisons simple, the teacher will also be assumed to be a perceptron of the same size as the student. This means that direct comparisons between the student and teacher weight vectors may be made. The training example set is made up of input output pairs, where the outputs, ζ, have been generated from the inputs $\boldsymbol{\xi}$ using,

$$\zeta(\boldsymbol{\xi}) = g_0\left(\frac{1}{\sqrt{N}}\mathbf{W}^0 \cdot \boldsymbol{\xi}\right) .$$

The vector $\mathbf{W}^0$ is the teacher weight vector and $g_0(\cdot)$ is the activation function

of the teacher.

A further assumption about the training set will be made; the components of the examples are assumed to be drawn from independent zero mean Gaussian distributions of unit variance , $d\mu(s_j) = \mathcal{N}(0,1)ds_j$. The same distribution will be assumed for the distribution of test examples which is used in the calculation of the generalisation function. It is well known that in the thermodynamic limit, the distribution of binary inputs (± 1) becomes the same as continuous zero mean Gaussian variables of unit variance, hence the results are also applicable for the binary input case. The assumption that the distributions are of unit variance is reasonable since any rescaling of the components of the input vector can be absorbed by a simple renormalisation of the weight vectors, this assumption will be looked at further in chapter 6. The error measure that will be used is the quadratic one introduced in eq. (2.2). From the error measure and the distribution of examples, the generalisation function may be calculated.

2.5.1 Generalisation function

Using the definition of the error measure and the distribution of examples assumed above, it is possible to calculate the generalisation function. From eq. (2.9) and using the quadratic definition of the error measure, eq. (2.2), the generalisation function may be written as,

$$\epsilon(\mathbf{W}) = \frac{1}{2}\int d\mu(\mathbf{s})\left[g\left(\frac{1}{\sqrt{N}}\mathbf{W}\cdot\mathbf{s}\right) - g_0\left(\frac{1}{\sqrt{N}}\mathbf{W}^0\cdot\mathbf{s}\right)\right]^2 ,$$

where the integration is taken over the whole of input space. There are two possible methods of proceeding: One is to say that the activations are zero mean

random variables and appeal to the central limit theorem [8]; the other is to introduce delta functions explicitly for the activations [64]. If integral representations for these delta functions are also introduced, it is possible to evaluate the integral over the examples. This is the procedure that will be followed

After introducing the delta functions for the activations of the student and teacher (x and y respectively) and their integral representations (appendix B), the generalisation function, $\epsilon(\mathbf{W})$, can be written as,

$$\epsilon(\mathbf{W}) = \frac{1}{2}\int \frac{dx\,d\hat{x}}{2\pi}\frac{dy\,d\hat{y}}{2\pi} d\mu(\mathbf{s})\,[g(x)-g_0(y)]^2 \times \exp\left[i\hat{x}\left(x-\frac{\mathbf{W}\cdot\mathbf{s}}{\sqrt{N}}\right)+i\hat{y}\left(y-\frac{\mathbf{W}^0\cdot\mathbf{s}}{\sqrt{N}}\right)\right] .$$

Unless otherwise stated, integrations are taken to be over the complete range of a variable. Thus in the above equation, the integration is taken over $x, \hat{x}, y, \hat{y}$ from -∞ to ∞ and $\mathbf{s} \in$ input space.

The integral over input space has factored out. Using the assumed distribution for the input components ($\mathcal{N}(0,1)$) this integral can be evaluated yielding,

$$\epsilon(\mathbf{W}) = \frac{1}{2}\int \frac{dx\,d\hat{x}}{2\pi}\frac{dy\,d\hat{y}}{2\pi} [g(x)-g_0(y)]^2 \times \exp\left[ix\hat{x}+iy\hat{y}-\frac{1}{2}q_0\hat{x}^2-\frac{1}{2}\Omega^2\hat{y}^2-R\,\hat{x}\hat{y}\right] ,$$

where the following order parameters have been introduced;

$$q_0 = \frac{1}{N}\mathbf{W}\cdot\mathbf{W} , \tag{2.15}$$

$$R = \frac{1}{N}\mathbf{W}\cdot\mathbf{W}^0 , \tag{2.16}$$

$$\Omega^2 = \frac{1}{N}\mathbf{W}^0\cdot\mathbf{W}^0 . \tag{2.17}$$

In general order parameters reduce the complexity of a system, instead of having to keep track of N components, it is only necessary to consider a number of simple order parameters that capture the statistics of the system. In the above case, $\sqrt{q_0}$ is the length of the student weight vector, Ω is the length of the teacher weight vector and $R' = R/(\Omega\sqrt{q_0})$ is the cosine of the angle between the student and teacher weight vectors.

The integrals over both the conjugate variables, $\hat{x}, \hat{y}$ can be evaluated. If x, y are rescaled such that $y = y/\Omega$ and $x = (x - \frac{yR}{\Omega^2})/\sqrt{(q_0 - \frac{R^2}{\Omega^2})}$, the generalisation function is given by,

$$\epsilon(\mathbf{W}) = \frac{1}{2}\int Dy\, Dx \left[g\left(\sqrt{q_0}\{x\sqrt{1-R'^2} + yR'\}\right) - g_0(y\Omega)\right]^2 \qquad (2.18)$$

where the notation $Dx = dx \exp[-\frac{1}{2}x^2]/\sqrt{2\pi}$ has been used.

The generalisation function is dependent on the student through the order parameters, q_0 and R'. The effect of the length of the student and teacher is to multiply the activation inside their respective activation functions. When the angle between the student and teacher is zero, $R' = 1$. In this case, if the activation functions of the student and teacher are identical and binary, *i.e.*, $g(x) = g_0(x) = \mathrm{sgn}(x)$, the generalisation function will be zero for any length of the student. To learn a binary teacher using a binary student it is sufficient to generate a student weight vector in the correct direction with arbitrary length. However, if the activation functions have some other form, the length of the student gives an effective gain of the student activation function. That is, if the activation function is of the form $g(\nu\, x)$, then the length of the student gives an effective gain of $\tilde{\nu} = \nu\sqrt{q_0}$. Similarly, for the teacher, the effective gain is $\tilde{\Omega} = \Omega\nu_0$. The generalisation function can only be exactly zero if the activations

of student and teacher are of the same form and the effective gain of the student is the same as the effective gain of the teacher. The average generalisation error is simply the quenched average of the thermal average of the generalisation function.

2.5.2 The cost function

In the section on training the network (§2.2) the posterior distribution of weights was defined in terms of the cost function, E_c. If generalisation is the quantity of interest, it might be hoped that minimising the training energy would lead to good generalisation ability. In this case, the cost function could be taken to be the training energy plus a potential term which is independent of the training data Θ,

$$E_c(\mathbf{W}, \Theta) = E_t(\mathbf{W}, \Theta) + V(\mathbf{W}),$$

where $V(\mathbf{W})$ is the potential term which depends on the weight vector and not the data set. The partition function defined for the Gibbs distribution, eq. (2.7) may now be written explicitly as,

$$\mathcal{Z} = \int d\mu(\mathbf{W}) \exp\left[-\beta E_t(\mathbf{W}, \Theta)\right] , \tag{2.19}$$

where, the potential term $V(\mathbf{W})$ has been incorporated into the *a priori* measure on weight space, *i.e.*, $d\mu(\mathbf{W}) = P_0(\mathbf{W})d\mathbf{W}$, and $P_0(\mathbf{W})$ is the prior distribution on student weight vectors that depends on the potential, $V(\mathbf{W})$. The prior distribution is used in the training algorithm to renormalise or otherwise constrain the weight vectors by including any knowledge about the expected form of the

student weight vectors. The simplest distribution is to assume a spherical constraint for the weight vectors, that is renormalise the weight vector after each update step to be of fixed length. More complicated distributions will be looked at in the following chapters.

2.6 The replica method

In this section, the free energy and hence the performance of the network is calculated using the replica method introduced in [15]. The formalism follows that developed by Seung *et al* [64], based on work by Gardner [25].

The basis of the replica method is the identity,

$$\ln z = \lim_{n \longrightarrow 0} \frac{1}{n}(z^n - 1) \; .$$

From equation(2.12) and using the identity above, the free energy may be written

$$-\beta F = \lim_{n \longrightarrow 0} \frac{1}{n} \ln \langle\!\langle \; \mathcal{Z}^n \; \rangle\!\rangle \; . \tag{2.20}$$

The free energy is evaluated for arbitrary integer n and then analytically continued to $n = 0$. The annealed approximation is equivalent to setting $n = 1$.

It is necessary to calculate $\langle\!\langle \; \mathcal{Z}^n \; \rangle\!\rangle$ for integer n. This is equivalent to replicating the system n times and then training the systems in parallel (hence "replica").

The systems interact through the quenched average over the training set. Introducing an index for each replica and using equation(2.19),

$$\langle\!\langle\, \mathcal{Z}^n \,\rangle\!\rangle = \int d\mu(\Theta) \prod_{\sigma} \left\{ \int d\mu(\mathbf{W}^\sigma) \exp\left[-\beta \sum_{\theta \in \Theta} \epsilon(\mathbf{W}^\sigma; \theta) \right] \right\} . \tag{2.21}$$

This replicated partition function contains a dependence on the distribution of the training set, Θ, used. This distribution may be expanded in terms of the individual examples in the training set. That is $d\mu(\Theta) = \prod_{l=1}^{p} d\mu(\theta^l)$ (θ^l refers to an input – output pair), p is the number of examples in the training set.

The product over the replica index becomes a sum in the exponential. The order of integration may be rearranged to give,

$$\langle\!\langle\, \mathcal{Z}^n \,\rangle\!\rangle = \int \prod_{\sigma} \{d\mu(\mathbf{W}^\sigma)\} \prod_{l} \left\{ \int d\mu(\theta^l) \exp\left[-\beta \sum_{\sigma} \epsilon(\mathbf{W}^\sigma; \theta^l) \right] \right\} .$$

The dependence on the training set is through the distributions of each individual training example. In the simplest case the examples are drawn from independent identical distributions, that is $d\mu(\theta^l) = d\mu(\theta)\ \forall\ l$. The product over the example index becomes p (the number of examples) copies of the same function. The integral over the example distribution is a function of the complete set of replicated weights, $\{\mathbf{W}^\sigma\}$. Using the definition of the number of examples per weight, $\alpha = p/N$, the replicated partition function may be written as,

$$\langle\!\langle\, \mathcal{Z}^n \,\rangle\!\rangle = \int \prod_{\sigma=1}^{n} \{d\mu(\mathbf{W}^\sigma)\} \exp\{-N\alpha G_{\mathrm{r}}[\mathbf{W}^\sigma]\} , \tag{2.22}$$

where $G_{\mathrm{r}}[\mathbf{W}^\sigma]$ is known as the replicated Hamiltonian and is given by

$$G_{\mathrm{r}}[\mathbf{W}^\sigma] = -\ln \int d\mu(\theta) \exp\left[-\beta \sum_{\sigma=1}^{n} \epsilon(\mathbf{W}^\sigma; \theta) \right] . \tag{2.23}$$

The training set dependence of the replicated Hamiltonian has been reduced to an integral over the distribution of a single example and a factor depending on the number of examples in the training set. The replicated Hamiltonian is related to the average performance of the replicated networks on a single input example, θ, its form is controlled by the distribution of the examples in the training set, $d\mu(\theta)$.

2.6.1 The replicated Hamiltonian

The replicated Hamiltonian can be calculated by making use of the example distribution. In this model, the outputs of the training examples are assumed to be generated from a teacher network. This means that an example only depends on its input, s, hence $d\mu(\theta) = d\mu(\mathbf{s})$. Substituting the quadratic error measure, eq. (2.2), into eq. (2.23) gives

$$G_{\mathrm{r}}[\mathbf{W}^{\sigma}] = -\ln \int d\mu(\mathbf{s}) \exp\left[-\frac{1}{2}\beta \sum_{\sigma=1}^{n} \left(g\left(\frac{1}{\sqrt{N}}\mathbf{W}^{\sigma}\cdot\mathbf{s}\right) - g_0\left(\frac{1}{\sqrt{N}}\mathbf{W}^{0}\cdot\mathbf{s}\right)\right)^2\right]. \tag{2.24}$$

In order to calculate the replicated Hamiltonian, delta functions will be introduced that pick out the activations, $\mathbf{W}\cdot\mathbf{s}$ of all the networks (the n replicated students and the teacher). Integral representations of these delta functions facilitate integration over the input space. This integration yields order parameters that describe the behaviour of the system. The integrations over the variables introduced in the integral representations can be evaluated for particular types of network.

The dependence of the error measure on the input, $\mathbf{s}$, is removed by introducing auxiliary variables for the activations of the networks; $\mathbf{W}^\sigma \cdot \mathbf{s} = x_\sigma$ for the replicated networks and $\mathbf{W}^0 \cdot \mathbf{s} = y$ for the teacher. The auxiliary variables are linked to the activations by delta functions. Using the integral representation of the delta function (appendix B), the integral over the input space can be factored out to yield,

$$\begin{aligned} \exp(-G_{\mathrm{r}}[\mathbf{W}^\sigma]) \;=\; & \int \prod_\sigma \left\{ \frac{dx_\sigma d\hat{x}_\sigma}{2\pi} \right\} \frac{dy d\hat{y}}{2\pi} \\ & \times \exp\left[-\frac{1}{2}\beta \sum_\sigma (g(x_\sigma) - g_0(y))^2 + i \sum_\sigma x_\sigma \hat{x}_\sigma + iy\hat{y} \right] \\ & \times \int d\mu(\mathbf{s}) \exp[-iN^{-\frac{1}{2}} (\sum_\sigma \mathbf{W}^\sigma \hat{x}_\sigma + \mathbf{W}^0 \hat{y}) \cdot \mathbf{s}] \end{aligned} \tag{2.25}$$

It is now possible to perform the integral over the input vector, $\mathbf{s}$. The distribution of input vectors is assumed to be a zero mean unit variance multivariate Gaussian distribution, *i.e.*, $d\mu(\mathbf{s}) = d\mathbf{s}\ (2\pi)^{-\frac{N}{2}} \mathrm{e}^{-\frac{1}{2}\mathbf{s}^2}$. Considering the integral over the input vector space in eq. (2.25) and using the distribution assumed above gives,

$$\begin{aligned} \int \frac{d\mathbf{s}}{(2\pi)^{\frac{N}{2}}} \exp\left[-\frac{1}{2}\mathbf{s}^2 - iN^{-\frac{1}{2}} (\sum_\sigma \mathbf{W}^\sigma \hat{x}_\sigma + \mathbf{W}^0 \hat{y}) \cdot \mathbf{s} \right] \;= \\ \exp\left[-\frac{1}{2} \left(\sum_{\sigma\rho} Q_{\sigma\rho} \hat{x}_\sigma \hat{x}_\rho + \Omega^2 \hat{y}^2 + 2 \sum_\sigma R_\sigma \hat{x}_\sigma \hat{y} \right) \right] , \end{aligned} \tag{2.26}$$

where the following order parameters have been introduced:

$$Q_{\sigma\rho} = \frac{1}{N} \mathbf{W}^\sigma \cdot \mathbf{W}^\rho , \tag{2.27}$$

$$R_\sigma = \frac{1}{N} \mathbf{W}^0 \cdot \mathbf{W}^\sigma , \tag{2.28}$$

$$\Omega^2 = \frac{1}{N} \mathbf{W}^0 \cdot \mathbf{W}^0 . \tag{2.29}$$

$Q_{\sigma\rho}$ is the overlap between replicas, R_σ is the overlap between a replica and the teacher weight vectors, and Ω is the length of the teacher as before, eq. (2.17). These order parameters characterise the statistics of the system, as well as reducing the number of free parameters from Nn to $O(n^2)$ for integer n.

The replicated Hamiltonian, $G_{\rm r}$, now depends on the replicated student weight vectors only through the order parameters $Q_{\sigma\rho}$ and R_σ. Using eq. (2.25), the replicated Hamiltonian can then be written as,

$$\begin{aligned}\exp(-G_{\rm r}[Q_{\sigma\rho},R_\sigma]) &= \int\prod_\sigma\left\{\frac{dx_\sigma d\hat{x}_\sigma}{2\pi}\right\}\frac{dyd\hat{y}}{2\pi}\\ &\times\exp\left[-\frac{1}{2}\beta\sum_\sigma(g(x_\sigma)-g_0(y))^2+i\sum_\sigma x_\sigma\hat{x}_\sigma+iy\hat{y}\right]\\ &\times\exp\left[-\frac{1}{2}\left(\sum_{\sigma\rho}Q_{\sigma\rho}\hat{x}_\sigma\hat{x}_\rho+\Omega^2\hat{y}^2+2\sum_\sigma R_\sigma\hat{x}_\sigma\hat{y}\right)\right] \quad (2.30)\end{aligned}$$

The behaviour of the system is now described in terms of the order parameters defined above. Returning to the quenched average of the replicated partition function, equation (2.22), and extracting the order parameters using delta functions allows the integral over the replicated student weights to be factored out. The quenched average of the replicated partition function eq. (2.22) can be written as

$$\begin{aligned}\langle\!\langle\, \mathcal{Z}^n \,\rangle\!\rangle &= \int\prod_\sigma\{d\mu(\mathbf{W}_\sigma)\}\exp\{-N\alpha G_{\rm r}[Q_{\sigma\rho},R_\sigma]\}\\ &\times\prod_\sigma\left\{N\,dR_\sigma\delta(NR_\sigma-\mathbf{W}^0\cdot\mathbf{W}^\sigma)\right\}\\ &\times\prod_{\sigma\le\rho}\{N\,dQ_{\sigma\rho}\delta(NQ_{\sigma\rho}-\mathbf{W}^\sigma\cdot\mathbf{W}^\rho)\} \quad (2.31)\end{aligned}$$

The second product of delta functions is only over $\sigma\le\rho$ since from eq. (2.27), $Q_{\sigma\rho}$ is symmetric in the replica indices. Hence the product is over a complete set

of independent values of $Q_{\sigma\rho}$. Integral representations for the delta function are introduced (and hence the conjugate order parameters, $\hat{Q}_{\sigma\rho}$ and $\hat{R}_\sigma$), yielding,

$$\langle\!\langle\, \mathcal{Z}^n \,\rangle\!\rangle = N^{\frac{n(n+3)}{2}} \int \prod_\sigma \left\{ \frac{dR_\sigma d\hat{R}_\sigma}{2\pi i} \right\} \prod_{\sigma \le \rho} \left\{ \frac{dQ_{\sigma\rho} d\hat{Q}_{\sigma\rho}}{2\pi i} \right\} \times \exp\left\{ N(G_0[Q_{\sigma\rho}, \hat{Q}_{\sigma\rho}, R_\sigma, \hat{R}_\sigma] - \alpha G_r[Q_{\sigma\rho}, R_\sigma, \Omega]) \right\} , \quad (2.32)$$

where

$$G_0[Q_{\sigma\rho}, \hat{Q}_{\sigma\rho}, R_\sigma, \hat{R}_\sigma] = -\sum_\sigma R_\sigma \hat{R}_\sigma - \sum_{\sigma \le \rho} Q_{\sigma\rho} \hat{Q}_{\sigma\rho} + \frac{1}{N} \ln \int \prod_\sigma \{ d\mu(\mathbf{W}^\sigma) \} \times \exp\left[\sum_\sigma \hat{R}_\sigma (\mathbf{W}^\sigma \cdot \mathbf{W}^0) + \sum_{\sigma \le \rho} \hat{Q}_{\sigma\rho} (\mathbf{W}^\sigma \cdot \mathbf{W}^\rho) \right] \quad (2.33)$$

The function G_0 is constrained by the prior distribution on the replicated weights, $d\mu(\mathbf{W}^\sigma)$ and will be called the prior constrained Hamiltonian, G_0 is also the logarithm of the density of networks which have order parameters R_σ and $Q_{\sigma\rho}$.

2.6.2 The free energy per weight

The free energy F, defined in eq. (2.12), is extensive in the number of weights N, *i.e.*, it scales with N. A non-extensive quantity, the free energy per weight, $f = \frac{F}{N}$, may be defined, which is finite in the thermodynamic $N \to \infty$ limit. In the thermodynamic limit the integrals over the order parameters in eq. (2.32) can be evaluated using the saddle point method [11], in which the integral of an exponential of an extensive quantity is replaced with the exponential of the extrema of the exponent. Using this method, the definition of the free energy per

weight and eq. (2.32) and (2.12), the free energy per weight may be written as,

$$-\beta f = \lim_{n\to 0} \underset{Q_{\sigma\rho}\hat{Q}_{\sigma\rho}R_\sigma\hat{R}_\sigma}{\text{extr}} \left\{ \frac{1}{nN} \{G_0 - \alpha G_r\} \right\} , \tag{2.34}$$

where $\underset{\text{parameters}}{\text{extr}} \{\ldots\}$ indicates that the function in the braces is extremised over the parameters. The factor of $N^{n(n+3)/2}$in the partition function eq. (2.32) gives a term like $n(n+3)\ln(N)/2N$ in the free energy per weight which tends to zero in the thermodynamic limit.

The calculation of the free energy and hence the average errors has been reduced to extremising the function in eq. (2.34) with respect to the order parameters. The order parameters are still rather complicated. In order to simplify the calculation somewhat, an ansatz for the order parameters is made. The simplest approximation to make is the replica symmetric ansatz.

2.7 The replica symmetric ansatz

In order to evaluate the free energy per weight, it is useful to reduce the complexity of the order parameters. It seems reasonable as a first approximation to assume that the order parameters are the same for all the replicas, although generally, the ground state of a system is not symmetric under symmetries of the Hamiltonian. This assumption is known as the replica symmetric (RS) ansatz and may be written explicitly as,

$$Q_{\sigma\rho} = q_0\,\delta_{\sigma\rho} + q_1(1-\delta_{\sigma\rho}) , \tag{2.35}$$

$$R_\sigma = R , \tag{2.36}$$

$$\hat{Q}_{\sigma\rho} = \hat{q}_0\,\delta_{\sigma\rho} + \hat{q}_1(1-\delta_{\sigma\rho})\,, \tag{2.37}$$

$$\hat{R}_\sigma = \hat{R}\,. \tag{2.38}$$

Referring to the definitions of $Q_{\sigma\rho}$ and R_σ, equations(2.27) and (2.28), the meaning of the RS order parameters can be identified. The average length squared of a replicated weight vector is q_0. The parameter q_1 is the average unnormalised overlap between two different replicas. This can be normalised by q_0 to give q, the average cosine of the angle between replica weight vectors. This average cosine is related to the volume of student weight space that contains possible solutions. The average unnormalised overlap between a replicated student weight and the teacher is given by R. This can be normalised by the length of the teacher and the average length of a student to give the average cosine of the angle between the student and teacher weight vectors. The order parameters, q, R are the averaged versions of those introduced in the calculation of the generalisation function eq. (2.15), (2.16). The meaning of the conjugate order parameters is less clear, but some insight may be gained; this is discussed in section (2.10).

By defining the replica symmetric ansatz, the complexity of the problem has diminished, since the number of free parameters has reduced from $O(n^2)$ to $O(1)$. The free energy per weight, equation (2.34), is given by extremising over the RS order parameters, that is,

$$-\beta f = \lim_{n\to 0} \operatorname*{extr}_{q_0\hat{q}_0 q_1\hat{q}_1 R\hat{R}} \left\{ \frac{1}{nN}\left\{G_0 - \alpha G_r\right\}\right\}\,. \tag{2.39}$$

It is not unreasonable to question whether the RS ansatz is valid. This problem has been studied for other statistical mechanics models [54]. The problem of replica symmetry breaking can manifest itself as a negative entropy for discrete

systems, another criterion used is the Almeida - Thouless line [13]. There exists a hierarchy of approximations that can introduce replica symmetry breaking into the order parameters [54]. The stability of the replica symmetric solution of perceptron learning has been previously studied, [64]. Replica symmetry breaking as applied to networks can be thought of in terms of the weight space. If the weight space is connected, the average value of the replicated student weight vectors will lie within the weight space. Therefore, the RS order parameters may accurately describe the system. However, if the weight space is disconnected, for example in the case of binary weights, the average value of the replicated student weight vector does not necessarily lie within the weight space. In this case the RS order parameters may not correctly describe a typical system. Thus the RS ansatz may not be valid for some weight distributions. It is widely believed that RS is valid for connected weight spaces.

2.8 The replica symmetric Hamiltonian

The replica symmetric Hamiltonian can be calculated by substituting the RS ansatz defined in equations (2.35) and (2.36) into equation (2.30). This enables the integrals over the conjugate variables $\hat{x}$ and $\hat{y}$ to be performed, again reducing the complexity. The RS Hamiltonian is given by,

$$\begin{aligned}\exp[-G_{\mathrm{r}}] &= \int \prod_{\sigma} \left\{ \frac{dx_\sigma d\hat{x}_\sigma}{2\pi} \right\} \frac{dy d\hat{y}}{2\pi} \exp\left[-\frac{1}{2}\beta \sum_{\sigma} (g(x_\sigma) - g_0(y))^2 \right] \\ &\quad \times \exp\left[-\frac{1}{2} q_1 \sum_{\sigma \neq \rho} \hat{x}_\sigma \hat{x}_\rho - \frac{1}{2} q_0 \sum_{\sigma} \hat{x}_\sigma^2 + i \sum_{\sigma} x_\sigma \hat{x}_\sigma \right] \\ &\quad \times \exp\left[-\frac{1}{2}\Omega^2 \hat{y}^2 + iy\hat{y} - R\hat{y} \sum_{\sigma} \hat{x}_\sigma \right] . \end{aligned} \tag{2.40}$$

The integral over the conjugate variable $\hat{y}$ may be factored out and evaluated. The sum $\sum_{\sigma\neq\rho}$ can be split up into $\sum_{\sigma\rho} - \sum_{\sigma}$. The notation is simplified by rescaling y such that, $y' = \frac{y}{\Omega}$. The double sum, $\sum_{\sigma\rho}$, in the exponential may be removed using a Hubbard – Stratonovitch (HS) transformation (see appendix B). This transformation linearises the exponent at the expense of introducing another Gaussian integral over a new variable, t. Explicitly, the HS transformation gives,

$$\exp\left[-\frac{1}{2}(q_1 - \frac{R^2}{\Omega^2})\sum_{\sigma\rho}\hat{x}_\sigma\hat{x}_\rho\right] = \int Dt \exp\left[it\sqrt{q_1 - \frac{R^2}{\Omega^2}}\sum_{\sigma}\hat{x}_\sigma\right] . \tag{2.41}$$

After this term is substituted into the RS Hamiltonian, the summations over the replica index can be factored out of the exponent to give an integral over a product,

$$\exp[-G_r] = \int Dy'\, Dt \prod_{\sigma}\mathcal{J}_\sigma , \tag{2.42}$$

where

$$\mathcal{J}_\sigma = \int \frac{dx_\sigma d\hat{x}_\sigma}{2\pi} \exp\left[-\frac{1}{2}\beta(g(x_\sigma) - g_0(\Omega y'))^2 - \frac{1}{2}(q_0 - q_1)\hat{x}_\sigma^2 + i\hat{x}_\sigma(x_\sigma - \frac{R}{\Omega}y') + it\hat{x}_\sigma\sqrt{q_1 - \frac{R^2}{\Omega^2}}\right] \tag{2.43}$$

The integral above is now independent of the replica index σ, and the replica index may be dropped. This means that now the individual replicas each contribute the same factor to the replicated Hamiltonian. This agrees with the introduction of the RS ansatz where the replicas were explicitly assumed be identical on average. The replicated Hamiltonian may be rewritten as,

$$G_r = -\ln \int Dy'\, Dt\, \mathcal{J}^n , \tag{2.44}$$

with $\mathcal{J}$ given by eq. (2.43) without the replica index σ.

Consider the contribution from each of the replicas. The integral over the conjugate variable $\hat{x}$ can be evaluated. Rescaling x such that

$$x' = \frac{\left(x - \frac{R}{\Omega}y' + t\left(q_1 - \frac{R^2}{\Omega^2}\right)^{\frac{1}{2}}\right)}{\sqrt{(q_0 - q_1)}}$$

gives,

$$\mathcal{J} = \int Dx' \exp\left[-\frac{1}{2}\beta(g\left(\sqrt{(q_0 - q_1)}\,x' + \frac{R}{\Omega}y' - t\sqrt{q_1 - \frac{R^2}{\Omega^2}}\right) - g_0(\Omega\, y'))^2\right]. \tag{2.45}$$

This is similar to the expression for the generalisation function derived earlier eq. (2.18). The introduction of the normalised order parameters, $R' = R/\Omega\sqrt{q_0}$ and $q' = q_1/q_0$ simplifies the notation.

In order to evaluate the free energy, the limit as $n \to 0$ of $1/n$ times the replicated Hamiltonian is taken. Define $\mathcal{G}_r = \lim_{n\to 0} \frac{1}{n} G_r$. For small n, $\mathcal{J}^n = 1 + n \ln \mathcal{J} + O(n^2)$; substituting this result into the logarithm of equation (2.44) and using $\ln(1 + a) = a + O(a^2)$ for small a, gives,

$$\begin{aligned}\mathcal{G}_r[q_0, q_1, R] \;&=\; -\int Dy\, Dt\, \ln \int Dx \\ &\times \exp\left[-\frac{1}{2}\beta\left(g\left(\sqrt{q_0}\left\{\sqrt{(1-q')}\,x + R'y - t\sqrt{q' - R'^2}\right\}\right) - g_0(\Omega\, y)\right)^2\right].\end{aligned} \tag{2.46}$$

The replicated Hamiltonian now depends on the weight vectors only through the order parameters q_0, q_1 and R. As discussed earlier, the order parameters

$R' = R/(\Omega\sqrt{q_0})$ and $q' = q_1/q_0$ are average cosines of angles between weight vectors. The length of the teacher becomes a multiplier in the activation function $g_0(y)$. The average length of the student, $\sqrt{q_0}$ also becomes a multiplier in the student activation function, $g(x)$. The replica symmetric Hamiltonian cannot be evaluated analytically for a general student activation function, however the calculation may be done for a linear student activation function.

The replica symmetric Hamiltonian can be calculated for a linear student, whilst the teacher may use any activation function at this stage. From eq. (2.46), and assuming the student activation is linear with gain ν, that is $g(x) = \nu x$, the integral over x may be evaluated, giving,

$$\begin{aligned} \mathcal{G}_r &= \tfrac{1}{2}\ln(1+\beta\nu^2(q_0-q_1)) \\ &\quad + \frac{\beta}{2}\int Dy\,Dt\,\frac{\left(\frac{R\nu}{\Omega}y + t\nu\sqrt{q_1-\frac{R^2}{\Omega^2}} - g_0(\Omega y)\right)^2}{(1+\beta\nu^2(q_0-q_1))} \end{aligned}$$

The remaining integral over t may be evaluated without difficulty. However since there is no exact form for the activation function of the teacher, the notation $\langle g_0^2\rangle = \int Dx g_0^2(\Omega x)$ and $\langle g_0 x\rangle = \int Dx\, x\, g_0(\Omega x)$ is introduced following Bös *et al* [8] and the final result may be written as,

$$\mathcal{G}_r = \tfrac{1}{2}\ln(1+\beta\nu^2(q_0-q_1)) + \frac{\beta}{2}\frac{(q_1\nu^2 - 2\sqrt{q_0}\nu R'\langle x\, g_0\rangle + \langle g_0^2\rangle)}{(1+\beta\nu^2(q_0-q_1))}\,. \qquad (2.47)$$

The gain of the student activation function, ν can be absorbed into the order parameters by a simple renormalisation of the weights, thus in the subsequent analysis, the gain need not be considered.

2.9 RS prior constrained Hamiltonian

The RS ansatz may be applied to the prior constrained Hamiltonian G_0 in the same manner as the previous section. From equation (2.33) and substituting in the RS ansatz,

$$G_0 = -nR\hat{R} - nq_0\hat{q}_0 - \frac{1}{2}(n^2 - n)q_1\hat{q}_1 + \frac{1}{N}\ln \mathcal{I}\,, \tag{2.48}$$

where,

$$\begin{aligned}\mathcal{I} \;=\; \int \prod_{\sigma} \{d\mu(\mathbf{W}^\sigma)\} \exp\Big[\hat{R}\sum_{\sigma}(\mathbf{W}^\sigma \cdot \mathbf{W}^0) + (\hat{q}_0 - \tfrac{1}{2}\hat{q}_1)\sum_{\sigma}(\mathbf{W}^\sigma \cdot \mathbf{W}^\sigma) \\ + \tfrac{1}{2}\hat{q}_1 \sum_{\sigma\rho}(\mathbf{W}^\sigma \cdot \mathbf{W}^\rho)\Big]\,.\end{aligned} \tag{2.49}$$

Using an HS transformation the the sum over $\sigma\rho$ may be linearised at the expense of introducing a Gaussian integral over a vector $\mathbf{z}$. The replica index can be factored out since the contribution from each replica is the same. As before, the contribution from all the replicas may be expanded as a power series in n for small n. Defining $\mathcal{G}_0 = \lim_{n\to 0} \frac{1}{n} G_0$ yields,

$$\begin{aligned}\mathcal{G}_0 \;=\; & -R\hat{R} - q_0\hat{q}_0 + \tfrac{1}{2}q_1\hat{q}_1 \\ & + \frac{1}{N}\int Dz \ln \int d\mu(\mathbf{W}) \\ & \qquad \times \exp\Big[(\hat{q}_0 - \frac{1}{2}\hat{q}_1)\mathbf{W}^2 + \hat{R}(\mathbf{W}\cdot\mathbf{W}^0) + \sqrt{\hat{q}_1}\mathbf{z}\cdot\mathbf{W}\Big]\,.\end{aligned} \tag{2.50}$$

The prior constrained Hamiltonian depends on the prior distribution of student

weights through the measure, $d\mu(\mathbf{W})$ and the complete set of RS order parameters, $q_0, \hat{q}_0, q_1, \hat{q}_1, R, \hat{R}$. In order to evaluate the integral over weight space, it is necessary to assume some prior on the student weight vectors. The meaning of the non-conjugate order parameters has already been given, the meaning of the conjugate order parameters is discussed in the next section.

2.10 The conjugate order parameters

The results obtained for the replicated Hamiltonian and the prior constrained Hamiltonian can be substituted into the free energy per weight given by eq. (2.34). In order to extremise this function, the differentials with respect to the order parameters are set to zero. Considering the differentials with respect to the conjugate order parameters leads to the following set of equations:

$$R = \frac{1}{N}\int Dz \langle \mathbf{W} \rangle_z \cdot \mathbf{W}^0 , \tag{2.51}$$

$$q_0 = \frac{1}{N}\int Dz \langle \mathbf{W} \cdot \mathbf{W} \rangle_z , \tag{2.52}$$

$$q_1 = \frac{1}{N}\int Dz \langle \mathbf{W} \rangle_z \cdot \langle \mathbf{W} \rangle_z , \tag{2.53}$$

where the average $\langle \mathbf{W} \rangle_z$ is defined by,

$$\langle \mathbf{W} \rangle_z = \frac{\int d\mu(\mathbf{W})\ \mathbf{W}\ \exp\left[-\frac{1}{2}\mathbf{W}^2(\hat{q}_1 - 2\,\hat{q}_0) + (\sqrt{\hat{q}_1}\mathbf{z} + \hat{R}\mathbf{W}^0)\cdot\mathbf{W}\right]}{\int d\mu(\mathbf{W})\ \exp\left[-\frac{1}{2}\mathbf{W}^2(\hat{q}_1 - 2\,\hat{q}_0) + (\sqrt{\hat{q}_1}\mathbf{z} + \hat{R}\mathbf{W}^0)\cdot\mathbf{W}\right]} . \tag{2.54}$$

The average $\langle \ldots \rangle_z$ can be written in terms of a probability distribution for the

weight vectors given by,

$$P(\mathbf{W}) = \frac{1}{\mathcal{Z}} \exp\left[-\mathcal{H}_{\text{eff}}\right] ,$$

Where the effective Hamiltonian is given by

$$\mathcal{H}_{\text{eff}}(\mathbf{W}) = \frac{1}{2}\mathbf{W}^2(\hat{q}_1 - 2\,\hat{q}_0) - (\sqrt{\hat{q}_1}\mathbf{z} + \hat{R}\mathbf{W}^0) \cdot \mathbf{W} .$$

The Hamiltonian is made up of two terms, the first simply measures the length of the weight vector, $\mathbf{W}$. The second is a field term that looks at the overlap of the weight vector with another vector given by $\sqrt{\hat{q}_1}\mathbf{z} + \hat{R}\mathbf{W}^0$. The first term in this vector is a Gaussian random field with variance $\hat{q}_1$, the second is a bias towards the teacher with magnitude $\hat{R}$. The ground state of the Hamiltonian is achieved when the weight vector is the shortest vector that lies parallel to $\sqrt{\hat{q}_1}\mathbf{z} + \hat{R}\mathbf{W}^0$.

2.11 A simple prior - the spherical constraint

The expression for the prior constrained Hamiltonian, eq. (2.50), contains the prior distribution on the weights, $d\mu(\mathbf{W})$. The integral over the weights may be evaluated if an explicit form for this distribution is assumed. The spherical constraint assumes that the weight vectors are drawn from the surface of a hypersphere. This is equivalent to renormalising the weight vectors after each update step. Much previous work has been done using the spherical constraint, [24]. In this section, the spherical constraint will be introduced and the order parameters calculated.

The spherical constraint is of particular use if a binary activation function is

used, since only the direction of the weight vector is important. The length of the weight vector can be taken without loss of generality to be one, since any renormalisation in the length of the weight vectors can be absorbed into the binary activation function.

The spherical constraint reduces the number of order parameters needed. In this case, the length of a replicated student weight, $Q_{\sigma\sigma}$ is constrained to be 1. The integrals over $Q_{\sigma\sigma}$ and $\hat{Q}_{\sigma\sigma}$ are no longer needed, within this formalism this is equivalent to setting the conjugate order parameter $\hat{Q}_{\sigma\sigma} = 0 \ \forall\sigma$ and $Q_{\sigma\sigma} = 1 \ \forall\sigma$ in the replica equations.

2.11.1 Prior constrained Hamiltonian

Again assuming the replica symmetric ansatz, the prior constrained Hamiltonian is given by equation (2.50). The spherical constraint demands that $q_0 = 1$ and $\hat{q}_0 = 0$. substituting these values into equation (2.50) gives,

$$\begin{aligned} \mathcal{G}_0 &= -R\hat{R} + \tfrac{1}{2}q_1\hat{q}_1 \\ &\quad + \frac{1}{N}\int Dz \ln \int d\mu(\mathbf{W}) \\ &\qquad \times \exp\left[-\frac{1}{2}\hat{q}_1\mathbf{W}^2 + \hat{R}(\mathbf{W}\cdot\mathbf{W}^0) + \sqrt{\hat{q}_1}\mathbf{z}\cdot\mathbf{W}\right] . \end{aligned} \tag{2.55}$$

The spherical constraint is equivalent to making the *a priori* assumption that the weight vectors are selected from a hypersphere of radius N. The prior distribution of weights may then be written explicitly as,

$$d\mu(\mathbf{W}) = \delta(\mathbf{W}^2 - N)d\mathbf{W}$$

$$= \frac{d\mathbf{W}}{(2\pi)^{\frac{N}{2}}} \int \frac{d\lambda}{2\pi i} \exp[\lambda(\mathbf{W}\cdot\mathbf{W} - N)]$$

where the parameter λ arises from the integral representation of the delta function.

Substituting this into equation(2.55) and evaluating the Gaussian integral on the weight vector gives,

$$\begin{aligned}\mathcal{G}_0 &= -R\hat{R} + \tfrac{1}{2}q_1\hat{q}_1 \\ &\frac{1}{N}\int D\mathbf{z} \ln\left\{\int \frac{d\lambda}{2\pi i}(\lambda+\hat{q})^{-\frac{N}{2}}\right. \\ &\left.\times \exp\left[\frac{1}{2}\frac{(N\hat{R}^2 + \hat{q}\mathbf{z}^2)}{\lambda+\hat{q}} + \frac{(\hat{R}\sqrt{\hat{q}}\mathbf{W}^0\cdot\mathbf{z})}{\lambda+\hat{q}} + \frac{1}{2}\lambda N\right]\right\} \quad (2.56)\end{aligned}$$

Since $\mathbf{z}^2 = O(N)$ and $\mathbf{W}^0\cdot\mathbf{z} = O(N)$ then the λ integration may be performed by the saddle point method giving a saddle point λ^* independent of $\mathbf{z}$. Hence after performing the integral over $\mathbf{z}$ the prior constrained Hamiltonian may be written as,

$$\mathcal{G}_0 = -R\hat{R} + \frac{1}{2}q\hat{q} + \frac{1}{2}\left\{-\ln(\lambda^*+\hat{q}) + \lambda^* + \frac{(\hat{R}^2+\hat{q})}{\lambda^*+\hat{q}}\right\} \quad (2.57)$$

2.11.2 Replicated Hamiltonian

If a spherical linear student is assumed, the replica symmetric Hamiltonian is given by equation (2.47). Substituting in $q_0 = 1$ and $\Omega = 1$, yields,

$$\mathcal{G}_r = \tfrac{1}{2}\ln(1+\beta(1-q')) + \frac{\beta}{2}\frac{(q' - 2R'\langle xg_0\rangle + \langle g_0^2\rangle)}{(1+\beta(1-q'))^2}\,. \quad (2.58)$$

The normalised order parameters are used: $q' = q_1$ and $R' = R$, this is because the lengths of the weight vectors are constrained to be one, the overlaps are cosines.

2.11.3 Free energy

The free energy per weight for a linear student may now be calculated from $\mathcal{G}_r$ and $\mathcal{G}_0$,

$$\begin{aligned} -\beta f &= -R\hat{R} + \frac{1}{2}q\hat{q} - \frac{1}{2}\ln(\lambda + \hat{q}) + \frac{1}{2}\lambda + \frac{1}{2}\frac{(\hat{R}^2 + \hat{q})}{(\lambda + \hat{q})} \\ &\quad -\frac{\alpha}{2}\ln(1 + \beta(1-q)) - \frac{\alpha\beta}{2}\frac{\langle q - 2R\,\langle xg_0\rangle + \langle g_0^2\rangle\rangle}{(1 + \beta(1-q))} \end{aligned} \tag{2.59}$$

where all the parameters q, $\hat{q}$, R, $\hat{R}$ and λ are to be extremised over. Finding the saddle points and eliminating λ, gives,

$$R = \hat{R}(1-q) \tag{2.60}$$

$$q = (\hat{R}^2 + \hat{q})(1-q)^2 \tag{2.61}$$

$$\hat{R} = \frac{\alpha\beta\,\langle xg_0\rangle}{(1 + \beta(1-q))} \tag{2.62}$$

$$\hat{q} = \frac{\alpha\beta^2(q - 2R\,\langle x\,g_0\rangle + \langle g_0^2\rangle)}{(1 + \beta(1-q))^2} \tag{2.63}$$

These equations are the saddle point equations for a linear student learning a teacher whose activation is given by $g_0(x)$ where all the weight vectors are subject to a spherical constraint. Since the spherical constraint has been used, the student is not able to alter its effective gain, through alteration in the length of its vector.

2.11.4 Linear teacher

The simplest form for the teacher activation is to assume that it is linear like the student. In this case, the averages $\langle xg_0 \rangle$ and $\langle g_0^2 \rangle$ are simple to calculate and give the results,

$$\langle xg_0 \rangle = \nu_0 \qquad \langle g_0^2 \rangle = \nu_0^2$$

This gives the results for the saddle point equations as

$$R = \hat{R}(1-q) \tag{2.64}$$

$$q = (\hat{R}^2 + \hat{q})(1-q)^2 \tag{2.65}$$

$$\hat{R} = \frac{\alpha\beta}{(1+\beta(1-q))}\nu_0 \tag{2.66}$$

$$\hat{q} = \frac{\alpha\beta^2(q - 2R\nu_0 + \nu_0^2)}{(1+\beta(1-q))^2} \tag{2.67}$$

From these results the average training and generalisation errors may be calculated for various limits [64].

Since the saddle point equations (2.64 - 2.67) are written in terms of the averages of the teacher activation function, $\langle xg_0 \rangle$ and $\langle g_0^2 \rangle$, the performance measures may be calculated for a linear student learning a nonlinear teacher. The evaluation of $\langle xg_0 \rangle$ and $\langle g_0^2 \rangle$ can be done analytically for some activation functions. However in general it must be done numerically.

2.12 Other realisable problems

As pointed out by Bös *et al* [8], in the zero temperature limit, the exponential of the error in the Gibbs distribution behaves like a delta function, that is,

$$\exp[-\tfrac{1}{2}\beta(g(h) - g_0(h_0))^2] \propto \delta(g(h) - g_0(h_0)) \qquad \text{as } \beta \to \infty,$$

where h and h_0 are the activations of the student and teacher respectively. Hence the free energy has a term which is proportional to $\delta(g - g_0)$. Provided $g = g_0$ and the inverse function $g^{-1}(\cdot)$ exists, the free energy is independent of the activation function in the zero T limit. Thus the order parameters for the realisable case linear – linear are the same for all realisable cases which have the same student and teacher activation functions, hence the performance measures can be calculated for other realisable rules.

2.13 Concluding remarks

To summarise, this chapter introduces some theoretical aspects of training a network to approximate a mapping between a vector of real valued inputs and a single output. The network is trained to model the mapping by minimising a cost function with respect to the network parameters, the weights. The cost function is defined on the training set as the mean square difference between the network output and that given by the example set plus a potential term and is minimised by stochastic gradient descent. Stochastic gradient descent produces a posterior Gibbs distribution of network parameters that is dependent on the noise level (temperature) used in the stochastic update procedure. The posterior

distribution defines an ensemble of networks that is dependent on the temperature used in the training algorithm. Averaging over this ensemble removes the dependence on the specific noise process used to generate a particular network. The posterior distribution is dependent on the examples in the training set. An average that is independent of the training examples can be calculated be averaging over the example set, a quenched average. The average training error is defined as the quenched average of the training error averaged over the thermal ensemble of networks.

If the mapping from inputs to output is described by a rule, it is reasonable to expect the network to be capable of extrapolation to unseen examples from outside the training set. If the correct output for every single input were known, a useful performance measure would be the mean square error of the model calculated over the set of all possible examples. Practically it would be impossible to calculate all possible example pairs without having detailed knowledge of the mapping itself. Theoretically the examples are assumed to have been generated by a known teacher mapping, this mapping can be used to calculate an average error for the network over the set of all possible examples, which is known as the generalisation function. A generalisation error that is independent of the specific network and the training data used can be calculated in a similar manner to the average training error. The average generalisation error is defined as the quenched average of the thermal average of the generalisation function.

The average quantities described above are related to the quenched average of the free energy of the system. The replica method of statistical physics is used to calculate the free energy. This method considers n copies of the system simultaneously and then takes the $n \rightarrow 0$ limit analytically. The statistical behaviour of the system is captured by order parameters which describe the overlaps between

the weight vectors of different replicas as well as the overlap between replicas and the teacher. The complexity of the problem is considerably reduced by assuming the replica symmetric (RS) ansatz. Using the RS ansatz, the free energy can be calculated exactly for a linear student in the thermodynamic limit and hence, the average order parameters are evaluated.

In the following chapters, the formalism introduced in this chapter will be used to calculate the performance for networks with different prior distributions for the weights.

Chapter 3

Noisy data and weight decay - Calculation

In many real world problems, the data set that is used to train the network has been corrupted by some unknown noise process. This may cause the network to model the noise on the data rather than the underlying mapping and hence, the network may generalise poorly on unseen examples. If generalisation ability is important, it would be useful to be able to alter the cost function so that the effects of noise are minimised, *i.e.*, to try to ensure that the network only learns the underlying mapping. The noise that is added to the training data is *static* throughout the training process. In this sense, it differs from the noise that is added to the stochastic training algorithm which is dynamic. There are now two different sorts of noise to be considered; the fixed **static noise** that has corrupted the training data and the **dynamic noise** that is used in the training algorithm.

[0]Part of the work in this chapter has been published in [14]

The training set is used to constrain a network's degrees of freedom, if the training set does not contain enough information about the underlying mapping, some of the degrees of freedom may be constrained by the noise on the data. Consider a curve fitting problem as an analogy: Given a set of noisy data points generated from some "true" curve, the object is to fit a polynomial curve through the data points using least squares regression. The model parameters are the coefficients of the interpolation polynomial, the number of degrees of freedom of the model can be considered as the order of the polynomial curve that is being fitted. The training error of the network is the mean square error on the training data points and the generalisation error is the error for a random point picked from the "true" curve. If the order of the interpolated polynomial is greater than that of the "true" curve, then the interpolated polynomial may fit the noisy training data well, but give poor generalisation for a relatively small number of examples (*e.g.*, a cubic fitting a noisy linear etc). This leads to the principle of "Occam's razor" that simpler models are preferred over unnecessarily complex ones. The problem is how to formally include this principle into an algorithm.

There are many techniques that can be used to counter the problem of learning the noise rather than the data. Some involve starting from a model with a small number of degrees of freedom and adding more as necessary. This class of algorithms are known as constructive (*e.g.*, Upstart [22], cascade correlation [19], Tiling [53]). Another methodology is to start with a model that has a high number of degrees of freedom and remove those that are not needed - pruning a model (*e.g.*, optimal brain damage [47], optimal brain surgeon [33] *etc.*). There is a less severe method that is related to pruning; regularisation. In this case a term is added to the cost function that penalises complex models, *e.g.*, in the curve fitting problem it is possible to regularise the model by penalising high curvature.

The formalism developed in the previous chapter enables a prior on the student weight vectors to be chosen. Using a regulariser is equivalent to choosing a model from a distribution that disfavours those configurations that are penalised. Thus the prior distribution can be used to implement a regulariser on the student distribution. In the previous chapter, the prior distribution was chosen to be a spherical constraint that fixed the lengths of the weight vector to be unity. In this chapter the effect of using a weight decay [37] prior will be studied and the performance measures calculated. In the subsequent chapter, the performance measures calculated in this chapter will be evaluated for various limits.

3.1 Weight decay

Weight decay is a dynamic algorithm that can be used to regularise a network. At each update step, the weights are reduced by a small amount, this may be written mathematically as,

$$W_i^{new} = (1 - \epsilon) W_i^{old} ,$$

where W_i is a component of the student weight vector and ϵ is a small constant. This update rule on its own would cause all the weights to decay to zero. However, used in conjunction with standard gradient descent training and suitable values of ϵ, only those components of the student weight vector that are not well specified by the training data are reduced. Practically, after training is completed all those weights that are close to zero may be pruned.

If the network is trained using gradient descent of a cost function, weight decay

can be interpreted as adding a quadratic term to the training energy to give a new cost function, since when the cost function is differentiated, the quadratic term gives a decay term in the weight update equation. The new cost function is,

$$E_c(\mathbf{W};\Theta) = E_t(\mathbf{W};\Theta) + \frac{1}{2}\lambda \mathbf{W}\cdot\mathbf{W}\ , \tag{3.1}$$

where E_t is the training error defined in chapter 2, that is the sum of the error measure over the training set. The weight decay parameter λ is linked to the amount the weights decay, ϵ, by $\epsilon = \lambda\tau$, where τ is the size of the update step. The quadratic term in the cost function penalises large weights and is known as a penalty term.

The weight decay term may also be motivated by using a well known method of statistics, ridge regression. This method is an extension to the usual parameter fitting technique of least squares. A least squares analysis is similar to the pseudo inverse [37] solution of neural networks. For linear regression, a simple model to be fitted to the data is defined as,

$$\mathbf{y} = X\mathbf{W},$$

where $\mathbf{y}$ is a p dimensional vector containing the outputs of the training data, X is a $p \times N$ matrix containing the inputs of the examples and $\mathbf{W}$ is the vector of model parameters. The data is rescaled so that is has zero mean and variance one. The training error is minimised by the Moore - Penrose inverse [74],

$$\mathbf{W} = (X^T X)^{-1} X^T \mathbf{y}$$

The problem of setting the model parameters has been reduced to inverting the **pattern correlation matrix**, $X^T X$. If the data is ill-conditioned, (*e.g.*there is

a linear dependence in the inputs or $p < N$), the matrix $X^T X$ is near singular and has some eigenvalues that are close to zero. In this case, the estimated parameters are dominated by the smallest eigenvalues of $X^T X$. Ridge regression [74] is used by statisticians to get around the problem of having small eigenvalues in the correlation matrix. The technique adds a constant to the diagonal part of $X^T X$ and then the least squares estimate of the model parameters are given by solving

$$(X^T X + k\, I)\mathbf{W} = X^T \mathbf{y}\,,$$

where the parameter k is equivalent to the weight decay parameter and causes the disproportionate effect of the small eigenvalues to be removed.

The weight decay can be motivated as a pruning method or as a regularising prior on the student distribution. These motivations are equivalent to performing gradient descent on a cost function that is the sum of the training error and a penalty term. It has been shown that a weight decay is useful for reducing the effect of noisy data [43]. In order to investigate the effect of weight decay on noisy data, it is necessary to have a model of the static noise on the data.

3.2 Static noise model

In the previous chapter it was assumed that the training set Θ was generated from a teacher perceptron. This enabled useful comparisons to be drawn between the final average student and the teacher. The problem of studying the effect of

noise[1] added to the training set could be modelled by using a noisy teacher, *i.e.*, a teacher rule that is corrupted by some noise process. A noisy teacher was studied in the context of a linear perceptron by Krogh and Hertz [43] using an analysis of the training dynamics based on the eigenvalue spectrum of the input correlation matrix. A noise process is added to the teacher so that the distribution of training examples depends on the noise process as well as the input vector. The replica method assuming the replica symmetric ansatz will then be used to calculate the average performance of the network.

The network is assumed as before to be a single layer perceptron with N weights, $\mathbf{W}$, connecting the inputs to the output. The training set, Θ, consisting of p examples θ^l, is assumed to be produced by a teacher network that has been corrupted by a noise process. The output of the noisy teacher, σ_n, is given by,

$$\sigma_n(\mathbf{s}, \eta; \mathbf{W}_0) = g_0(\mathbf{W}_0 \cdot \mathbf{s} + \eta) \,, \tag{3.2}$$

where η is zero mean additive random noise. The distribution of a training example, θ, is now related to the distribution of noise as well as the distribution of the input vectors.

$$d\mu(\theta) = d\mu(\mathbf{s})\, d\mu(\eta) \,.$$

This means that the quenched average over the distribution of examples contains an extra integral over the noise distribution. The quenched average of the free energy may be calculated as in chapter 2.

---

[1]The distinction between static noise on the data and dynamic noise on the training should be clear from the context, if it is not, noise shall refer to static noise and dynamic noise will be labeled as such

3.3 Stochastic gradient descent

Stochastic gradient descent (§2.2), gives a Langevin equation for the weight updates of the form

$$\frac{\partial \mathbf{W}}{\partial t} = -\nabla E_t(\mathbf{W};\Theta) - \nabla V(\mathbf{W}) + \boldsymbol{\eta}(t)$$

where $V(\mathbf{W})$ is a data independent potential term. The weight decay penalty term is then equal to this potential term, $V(\mathbf{W}) = \frac{1}{2}\lambda \mathbf{W}\cdot\mathbf{W}$. This update rule gives an asymptotic Gibbs distribution of student weights with partition function,

$$\mathcal{Z} = \int d\mu(\mathbf{W}) \exp\left\{-\beta \sum_{\mathbf{s}\in\Theta} \epsilon(\mathbf{W};\mathbf{s})\right\} , \tag{3.3}$$

where $d\mu(\mathbf{W})$ is an *a priori* measure on the student weight space and β is related to the variance of the dynamic noise, $\boldsymbol{\eta}(t)$ (see §2.2). The measure may be related to the potential term; for a weight decay this gives,

$$d\mu(\mathbf{W}) = \left(\frac{\beta\lambda}{2\pi}\right)^{N/2} \exp(-\frac{1}{2}\beta\lambda\mathbf{W}^2) d\mathbf{W} , \tag{3.4}$$

where the factor outside the exponential normalises the measure so that $\int d\mu(\mathbf{W}) = 1$. The measure $d\mu(\mathbf{W})$ is equivalent to a Gaussian prior on the student weights. The free energy may be evaluated for this prior distribution and the effect on the average performance of the network calculated.

The Bayesian formalism, [49], introduces the prior on the student weights as an Gaussian distribution with the variance controlled by a single parameter. The temperature is introduced to measure the variance of the noise model postulated for the student. It turns out that the Bayesian generalisation error used by

Bruce *et al* eq. (2.11), is only dependent on the ratio of the variances of the two distributions [9], *i.e.*, $\beta\lambda$ is a natural parametrisation of the Gaussian prior for studying the generalisation error.

3.4 Free energy for noisy perceptron with weight decay

The free energy and hence the performance measures for a simple perceptron with weight decay learning a corrupted or noisy teacher may now be evaluated. The free energy is calculated using the replica method, assuming replica symmetry, which gives from eq. (2.39),

$$-\beta f = \underset{q_0 \hat{q}_0 q_1 \hat{q}_1 R \hat{R}}{\text{extr}} \left\{ \frac{1}{N} \{\mathcal{G}_0 - \alpha \mathcal{G}_r\} \right\} . \tag{3.5}$$

where $q_0, \hat{q}_0, q_1, \hat{q}_1, R, \hat{R}$ are the replica symmetric order parameters introduced in §2.7. The free energy per weight is calculated by extremising f with respect to the replica symmetric order parameters. The replicated Hamiltonian $\mathcal{G}_r$ (§2.8) is constrained by the architecture of the student and teacher networks being modelled. The prior constrained Hamiltonian $\mathcal{G}_0$ (§2.9) is dependent on the student prior chosen (in this case the weight decay or Gaussian prior).

The Gaussian prior distribution on the student weights is equivalent to the canonical distribution of statistical mechanics. In the thermodynamic limit, the *a priori* student distribution is equivalent to a microcanonical distribution with the weight vector constrained to be of length $\sqrt{N/\beta\lambda}$. This suggests there is some scale invariance associated with scaling of the student weights,$\mathbf{W}$, the temperature, β

and the weight decay λ. If the training error is independent of the length of the student weight (as for a hard threshold), then a scale transformation in the student weights can be absorbed by a renormalisation of the weight decay parameter, λ. This is not the case for the linear perceptron; in this case, the training error depends on terms quadratic in the student weights and a rescaling of the student weights can be absorbed by a rescaling of the training data outputs and the training temperature β. This would leave the posterior student distribution and hence the free energy unchanged.

The teacher weight vector is assumed to have fixed length, this is equivalent to a microcanonical distribution. This means that in the thermodynamic limit, the student and teacher weight vectors are chosen from the same type of prior distribution. Intuitively this seems to be a good idea for good generalisation with small numbers of examples, since if the student and teacher priors are poorly matched, the prior student distribution will be giving incorrect hints to the training. For a large number of patterns, the data swamps the prior student distribution and the importance of matching the prior distributions is lessened. Matching the student and teacher distributions is discussed in more detail in chapter 6.

3.4.1 Replicated Hamiltonian

The replicated Hamiltonian eq. (2.23) now contains the integration over the distribution of noise on the teacher. Substituting the new example distribution into eq. (2.23) and taking the $n \rightarrow 0$ limit gives,

$$\mathcal{G}_{\mathrm{r}}[\mathbf{W}^{\sigma}] = -\lim_{n \rightarrow 0} \frac{1}{Nn} \ln \int d\mu(\mathrm{s}) d\mu(\eta) \exp\left[-\beta \sum_{\sigma=1}^{n} \epsilon(\mathbf{W}^{\sigma}; \mathrm{s}, \eta)\right] , \qquad (3.6)$$

where the quadratic error measure,

$$\epsilon(\mathbf{W}^{\sigma};\mathbf{s},\eta) = \frac{1}{2}\left(g\left(\mathbf{W}\cdot\mathbf{s}/\sqrt{N}\right) - g_0\left(\mathbf{W}^0\cdot\mathbf{s}/\sqrt{N}+\eta\right)\right)^2 ,$$

will be used.

After extensive calculation following similar steps as in §2.8, introducing the order parameters and assuming replica symmetry, the replica symmetric Hamiltonian is

$$\begin{aligned}\mathcal{G}_{\rm r}[q_0,q_1,R] &= -\int Dy\, Dt\, d\mu(\eta)\ln\int Dx \\ &\times\exp\left[-\frac{1}{2}\beta\left(g\left(\sqrt{(q_0-q_1)}\,x+\frac{R}{\Omega}y-t\sqrt{q_1-\frac{R^2}{\Omega^2}}\right)-g_0(\Omega y+\eta)\right)^2\right] \quad (3.7)\end{aligned}$$

This equation is similar to the replica symmetric Hamiltonian for uncorrupted data eq. (2.46). The static noise dependence has introduced an integral over the noise distribution. The function $\mathcal{G}_{\rm r}$, eq. (3.7), may be evaluated analytically for a linear student as before.

Assuming the activation function of the student is linear, $g(x) = x$; introducing the notation $\langle\, g_0^2\,\rangle_\eta = \int Dx\, d\mu(\eta)\, g_0^2(\Omega x+\eta)$ and $\langle\, x\, g_0\,\rangle_\eta = \int Dx\, d\mu(\eta)\, x\, g_0(\Omega x+\eta)$ and evaluating the remaining integrals, the replica symmetric Hamiltonian for a linear student learning a noisy teacher with arbitrary activation function is,

$$\begin{aligned}\mathcal{G}_{\rm r}(q_0,q_1,R) &= \frac{1}{2}\ln(1+\beta(q_0-q_1)) \\ &+\frac{\beta}{2\,(1+\beta(q_0-q_1))}\left(q_1-2\,R\frac{\langle\, x\, g_0\,\rangle_\eta}{\Omega}+\left\langle\, g_0^2\,\right\rangle_\eta\right) . \quad (3.8)\end{aligned}$$

The only difference between this and the RS Hamiltonian for a clean teacher

is the average over the noise that takes place in $\langle x\, g_0 \rangle_\eta$ and $\langle g_0^2 \rangle_\eta$. A linear student with a finite gain, *i.e.*, $g(x) = \nu x$ could be considered. Since the effect is to rescale the order parameters, to keep the analysis simple, the gain is absorbed into the order parameters.

3.4.2 Prior constrained Hamiltonian

The weight decay term introduced in §3.1 can be written as a Gaussian prior distribution on the student weights; this gives the prior measure of the student weights in eq. (3.4). The replica symmetric prior constrained Hamiltonian, is given by eq. (2.33). The measure, $d\mu(\mathbf{W})$ eq. (3.4), can be substituted, which after some calculation gives,

$$\begin{aligned} \mathcal{G}_0 \;=\; & -R\hat{R} - q_0\hat{q}_0 + \frac{1}{2}q_1\hat{q}_1 + \frac{1}{2}\ln\beta\lambda \\ & -\frac{1}{2}\ln(\beta\lambda + \hat{q}_1 - 2\,\hat{q}_0) + \frac{(\hat{q}_1 + \hat{R}^2\Omega^2)}{2(\beta\lambda + \hat{q}_1 - 2\,\hat{q}_0)} \end{aligned} \tag{3.9}$$

This prior constrained Hamiltonian is similar to that calculated for the spherical constraint eq. (2.57), however, in this case there is a dependence on the diagonal order parameters, q_0 and $\hat{q}_0$, since the length of the student is not renormalised at each update.

3.4.3 Saddle point equations

The free energy for a linear student learning a corrupted teacher using a weight decay is given by,

$$\begin{aligned} -\beta f &= -R\hat{R} - q_0\hat{q}_0 + \frac{1}{2}q_1\hat{q}_1 - \frac{1}{2}\ln(\beta\lambda - 2\,\hat{q}_0 + \hat{q}_1) + \frac{1}{2}\ln(\beta\lambda) \\ &\quad + \frac{1}{2}\frac{\hat{R}^2\Omega^2 + \hat{q}_1}{\beta\lambda - 2\,\hat{q}_0 + \hat{q}_1} - \frac{\alpha}{2}\ln(1 + \beta(q_0 - q_1)) \\ &\quad - \frac{1}{2}\frac{\alpha\beta}{(1 + \beta(q_0 - q_1))}\left(q_1 - 2\,R\frac{\langle\, x\, g_0 \,\rangle_\eta}{\Omega} + \left\langle\, g_0^2 \,\right\rangle_\eta\right), \end{aligned} \tag{3.10}$$

with the order parameters given by their saddle point values. The free energy is simplified if the order parameters corresponding to the overlap between student and teacher and its conjugate are rescaled: $r = R/\Omega$ and $\hat{r} = \hat{R}\Omega$. The saddle point equations are,

$$q_0 = q_1 + \frac{1}{\beta\lambda + \hat{q}_1 - 2\hat{q}_0} \tag{3.11}$$

$$q_1 = (\hat{r}^2 + \hat{q}_1)(q_0 - q_1)^2 \tag{3.12}$$

$$\hat{q}_0 = \frac{1}{2}(\hat{q}_1 - \frac{\hat{r}}{\langle\, x g_0 \,\rangle_\eta}) \tag{3.13}$$

$$\hat{q}_1 = \frac{\alpha\beta^2}{(1 + \beta(q_0 - q_1))^2}\left(q_1 - 2\,r\,\langle\, x\, g_0 \,\rangle_\eta + \left\langle\, g_0^2 \,\right\rangle_\eta\right) \tag{3.14}$$

$$r = \hat{r}(q_0 - q_1) \tag{3.15}$$

$$\hat{r} = \frac{\alpha\beta}{1 + \beta(q_0 - q_1)}\,\langle\, x\, g_0 \,\rangle_\eta \tag{3.16}$$

where it is assumed that the number of training examples scales as α times the number of weights, that is $\alpha = p/N$.

The saddle point equations are for a linear student learning any nonlinear perceptron teacher. It is straightforward to check that these equations reduce to

those presented by Seung *et al* [64] for a spherical constraint and a linear teacher by setting $q_0 = 1$ and $\Omega = 1$ corresponding to a spherical normalisation on the student and teacher weights. This result will be discussed in more detail in §4.1.

3.5 Solving the saddle point equations

The saddle point equations eq. (3.11) - (3.16) can be solved to give the behaviour of a linear student using a weight decay learning an arbitrary teacher from noisy data. From eq. (3.11,3.12), define $\mathcal{Q} = q_0 - q_1$. Consider $\mathcal{Q}' = \beta\mathcal{Q}$, the saddle point equations give a quadratic equation for $\mathcal{Q}'$ that has roots given by,

$$\mathcal{Q}' = \frac{1}{2\lambda}(1-\alpha-\lambda) \pm \frac{1}{2\lambda}\sqrt{(1-\alpha-\lambda)^2 + 4\lambda}\,, \tag{3.17}$$

In the replica symmetric ansatz q_0 is identified with the average length squared of a student weight vector. Thus $q' = q_1/q_0$ is the average cosine of the angle between replica weight vectors. This quantity is independent of the gain of the student. The overlap between replicas varies from zero corresponding to the replicas spanning the whole of weight space, to one, corresponding to a single student solution. Since $q' \leq 1$, $\mathcal{Q} = (q_0 - q_1) \geq 0$. Thus only the positive root of $\mathcal{Q}$ is needed. The function $\mathcal{Q}'$ is identical to the static limit of the response function for uncorrelated patterns as calculated by Hertz *et al* [38].

After calculating $\mathcal{Q}$, the remaining order parameters may be evaluated in terms of $\phi = 1 + \frac{1}{\mathcal{Q}'}$, detailed calculation (appendix C) yields;

$$q_0 = q_1 + \frac{T}{\phi - 1} \tag{3.18}$$

$$q_1 = \frac{\alpha}{(\phi^2-\alpha)}\left(\left\langle g_0^2\right\rangle_\eta - \langle x\, g_0\rangle_\eta^2(1+\lambda)\right) + \frac{\alpha\langle x\, g_0\rangle_\eta^2}{\phi} \tag{3.19}$$

$$r = \frac{\alpha\langle x\, g_0\rangle_\eta}{\phi} \tag{3.20}$$

$$\hat{q}_0 = \tfrac{1}{2}\left(\hat{q}_1 - \frac{\hat{r}}{\langle x g_0\rangle_\eta}\right) \tag{3.21}$$

$$\hat{q}_1 = \frac{\alpha\beta^2}{(\phi^2-\alpha)}\left(\langle x\, g_0\rangle_\eta^2\lambda^2 + \left(\left\langle g_0^2\right\rangle_\eta - \langle x\, g_0\rangle_\eta^2\right)(\phi-1)^2\right) \tag{3.22}$$

$$\hat{r} = \alpha\beta\langle x\, g_0\rangle_\eta\left(1-\frac{1}{\phi}\right) \tag{3.23}$$

The parameter ϕ is independent of the temperature and the teacher activation used.

The normalised overlap between student and teacher, $R' = R/(\Omega\sqrt{q_0})$ may be written as,

$$R' = \frac{r}{\sqrt{q_0}} \tag{3.24}$$

The normalised overlap between replicas may be written in terms of the order parameters, $q' = q_1/q_0$

3.6 General teacher activation functions

The actual form of the teacher activation function appears in the order parameters through the averages, $\langle x g_0\rangle_\eta$ and $\langle g_0^2\rangle_\eta$. The averages are defined by,

$$\left\langle g_0^2\right\rangle_\eta = \int Dx\, d\mu(\eta)\, g_0^2(\nu_0(\Omega x+\eta)) \tag{3.25}$$

$$\langle x\, g_0\rangle_\eta = \int Dx\, d\mu(\eta)\, x\, g_0(\nu_0(\Omega x+\eta)) \tag{3.26}$$

where the gain on the teacher function ν_0 has been introduced explicitly. The noise may be renormalised by the length of the teacher weight vector, Ω to give $\tilde{\eta} = \eta/\Omega$. The averages are now in terms of a single gain parameter, $\tilde{\Omega} = \nu_0\Omega$, which means that the teacher gain, ν_0, can be set to unity without loss of generality. The actual distribution of the noise has variance, $\gamma^2 = \Omega^2\tilde{\gamma}^2$, where $\tilde{\gamma}^2$ is the variance of the renormalised noise. For the remainder of the thesis, the static noise considered will be the renormalised noise $\tilde{\eta}$.

Consider the average square difference between a nonlinear teacher activation and a linear model,

$$S = \tfrac{1}{2}\int Dx d\mu(\eta)(g_0(\Omega(x+\eta)) - ax)^2 \, .$$

This is minimised by $a = \langle\, xg_0 \,\rangle_\eta$, thus the average $\langle\, xg_0 \,\rangle_\eta$ is the gradient/ gain of an effective linear teacher which models the nonlinear activation function. The effective renormalised noise level of the linear model may be written as

$$\tilde{\gamma}^2_{\text{eff}} = \left\langle\, g_0^2 \,\right\rangle_\eta / \langle\, x\, g_0 \,\rangle_\eta - 1 \, . \tag{3.27}$$

This effective noise level will be useful for studying nonlinear teachers. The averages can be evaluated numerically for any arbitrary activation function. For some specific activation functions, the integrations may be done analytically.

3.6.1 Linear teacher

In this case, the activation function of the teacher is given by $g_0(x) = x$. The averages are given by, $\langle\, x\, g_0 \,\rangle_\eta = \Omega$ and $\langle\, g_0^2 \,\rangle_\eta = \Omega^2(1+\tilde{\gamma}^2)$

3.6.2 Binary teacher

The activation function of the teacher is $g_0(x) = \mathrm{sgn}(x)$. This gives for the averages,

$$\left\langle g_0^2 \right\rangle_\eta = 1 \qquad \langle x\, g_0 \rangle_\eta = \sqrt{\frac{2}{\pi(1+\tilde{\gamma}^2)}} \tag{3.28}$$

These results are independent of the length of the teacher Ω as expected for a sgn(.) activation function. These results may substituted into the order parameters.

3.7 Calculating the order parameters

The averages evaluated above may be substituted into the solved saddle point equations to further simplify the equations for specific teacher activation functions. Since the quantities of interest are average quantities, the student weight vector will refer to the average student weight vector, unless otherwise indicated. Similarly, generalisation and training errors will be average quantities.

3.7.1 Linear teacher

With a linear teacher, the form of the free energy presented in equation (3.10) is similar to that given by Seung *et al* [64] for a linear perceptron learning a linear teacher with an unrealisable threshold, that is $\sigma^l = (\mathbf{W}^0 \cdot \mathbf{s}^l + \theta)$, where θ is the teacher threshold/bias. The free energies are identical if the spherical constraint is assumed and $\beta\lambda \to 0$, that is zero weight decay. In this case the threshold is

identified with the standard deviation of the noise on the teacher. Thus in this limit, an unrealisable threshold on the teacher is actually equivalent to adding noise to the teacher, which is then averaged.

The overlap between the average student and teacher weight vectors, r, is,

$$r = \frac{\alpha\Omega}{\phi} \,. \tag{3.29}$$

Returning to the average inter–replica overlap, equation (3.19), and substituting the results, using the quadratic equation for ϕ and rearranging gives,

$$q_1 = \frac{\alpha\Omega^2}{(\phi^2 - \alpha)}(\tilde{\gamma}^2 - \lambda) + \frac{\alpha\Omega^2}{\phi} \,. \tag{3.30}$$

The average overlap between the replicas is made up of two terms, one of which depends on the difference between the noise level on the data set and the weight decay parameter. The second term is proportional to the average overlap between student and teacher weight vectors.

The results for the linear teacher may be substituted back into equation (3.14),

$$\hat{q}_1 = \frac{\alpha\beta^2\Omega^2}{(\phi^2 - \alpha)}\left(\lambda^2 + \tilde{\gamma}^2(\phi - 1)^2\right) \,.$$

These results will be used in the next chapter to calculate the network's performance for various limits. They also suggest a useful way of rewriting the order parameters for arbitrary teacher activation function.

3.7.2 Nonlinear teacher

The saddle point equations for a nonlinear teacher are given in eq. (3.18) - (3.23), the overlap between replicas and its conjugate can be rewritten in terms of an effective noise level, $\tilde{\gamma}^2_{\text{eff}}$, eq. (3.27) and the effective gain of the teacher, $\langle x\, g_0 \rangle^2_\eta$,

$$q_1 = \frac{\alpha \langle x\, g_0 \rangle^2_\eta}{(\phi^2 - \alpha)} \left(\tilde{\gamma}^2_{\text{eff}} - \lambda \right) + \frac{\alpha \langle x\, g_0 \rangle^2_\eta}{\phi} \tag{3.31}$$

$$\hat{q}_1 = \frac{\alpha\beta^2 \langle x\, g_0 \rangle^2_\eta}{(\phi^2 - \alpha)} \left(\lambda^2 + \tilde{\gamma}^2_{\text{eff}} (\phi - 1)^2 \right) . \tag{3.32}$$

The remaining order parameters are the same as for the linear teacher case up to factors of the effective gain of the teacher, $\langle x\, g_0 \rangle_\eta$. Thus a linear student learning a nonlinear teacher is equivalent to a linear student learning a noisy linear teacher with a gain given by the effective teacher gain, $\langle x\, g_0 \rangle^2_\eta$ and the effective noise level given by $\tilde{\gamma}^2_{\text{eff}}$.

3.8 Generalisation error

The average generalisation error, $\epsilon_g = \langle\!\langle \langle \epsilon(\mathbf{W}) \rangle_T \rangle\!\rangle$, is a measure of a network's inability to solve a problem averaged over the entire data set. Assuming random, Gaussian distributed test patterns the generalisation function, $\epsilon(\mathbf{W})$, has been calculated generally in chapter 2, §2.5.1. The generalisation error is simply the quenched average of the thermal average of the generalisation function. In the replica symmetric case, the average generalisation error for a linear student

learning an arbitrary teacher is given by

$$\epsilon_g = \frac{1}{2}\left(q_0 + \left\langle g_0^2 \right\rangle - 2r\left\langle x g_0 \right\rangle\right) , \tag{3.33}$$

where the order parameters are the replica symmetric ones.

The generalisation error, eq. (3.33), measures the network's performance in learning the uncorrupted teacher. If the student weight vector exactly equals that of the teacher and the activation functions of student and teacher are identical, $\epsilon_g = 0$. The student could, however, be compared with the corrupted output of the teacher; this would make the averages of the teacher output, $\langle g_0^2 \rangle$ and $\langle x g_0 \rangle$ averages over the noise distribution as well, that is $\langle g_0^2 \rangle_\eta$ and $\langle x g_0 \rangle_\eta$ respectively. The average over the noise takes account of the student's inability to learn the uncertainty in the teacher. Hereafter, ϵ_g will refer to the network's generalisation error for learning the "clean" teacher. When the generalisation error in comparison with the corrupted teacher is referred to, the notation ϵ_g' will be used. The generalisation error compared to the clean teacher measures how closely the network has learnt the teacher rule, however, this performance measure may be rather hard to evaluate in practice since it requires knowledge of the uncorrupted rule that may not be available. Hence the corrupted generalisation error may be a more useful performance measure in certain situations.

For a linear teacher,

$$\epsilon_g = \frac{\Omega^2}{2}\left(1 + \frac{\alpha}{\phi^2 - \alpha}(\tilde{\gamma}^2 - \lambda) - \frac{\alpha}{\phi}\right) + \frac{T}{2(\phi - 1)} . \tag{3.34}$$

The generalisation error depends on the gain of the teacher, Ω, representing the square length of the teacher weight vector. This is as expected, since a linear

perceptron is used, the error is an absolute error and therefore the larger the weight vectors, the bigger the errors. The corrupted generalisation error for a linear student learning a linear teacher is given by

$$\epsilon_g{}' = \frac{\gamma^2}{2} + \epsilon_g \,,$$

where $\gamma^2 = \tilde{\gamma}^2\Omega^2$ is the total noise level on the teacher. Hence for a linear teacher it is only necessary to look at the uncorrupted generalisation error since it is related very simply to the corrupted generalisation error. For non–linear teachers, there is in general no such obvious relationship between the two generalisation errors. The corrupted generalisation error for an arbitrary teacher activation function may be written as,

$$\epsilon_g{}' = \frac{\langle x\, g_0 \rangle_\eta^2}{2}\left(1 + \tilde{\gamma}_{\rm eff}^2 + \frac{\alpha}{(\phi^2 - \alpha)}\left(\tilde{\gamma}_{\rm eff}^2 - \lambda\right) - \frac{\alpha}{\phi}\right) + \frac{T}{2(\phi - 1)} \qquad (3.35)$$

where the effective noise, $\tilde{\gamma}_{\rm eff}^2$, eq. (3.27), has been used. This gives a corrupted generalisation error that is similar to that for a linear teacher with the variance of the noise replaced by the effective noise $\tilde{\gamma}_{\rm eff}^2$ and the teacher gain Ω replaced by the effective gain $\langle x\, g_0 \rangle_\eta$. The generalisation error has an additive component due to the training temperature, thus the generalisation improves as the temperature is decreased.

Other forms of the generalisation error were discussed in §2.3. The generalisation error of Bruce *et al* [9] eq. (2.11), for a linear student and teacher is given by,

$$\epsilon_{BS} = q_1 - 2\Omega r + \Omega^2(1 + \gamma^2)$$

up to a factor of γ^2. This generalisation error is temperature *independent* and

hence optimising this generalisation error with respect to the temperature is meaningless. The difference between the errors ϵ_g and ϵ_{BS} is due to the difference between the conventional statistical mechanics view of measuring the error for a single solution from the posterior distribution and the Bayesian view of using the complete posterior distribution [64, 9]. From the Bayesian perspective optimal performance is achieved when the posterior distribution models the data distribution, *i.e.*, the variance on the posterior is equal to the variance of the noise on the data, the generalisation error is then measured as the squared bias of the posterior. The statistical mechanics view favours training to a single solution that is the MAP estimate of the student weight vector. For the Gaussian posterior under study, it turns out that the MAP estimate and the mean of the posterior are the same which means that both $\epsilon_g{}'$ and ϵ_{BS} are optimised by the same value of the weight decay parameter.

3.9 Training error

The average training error ϵ_t is defined as

$$\epsilon_t = \langle\!\langle \langle E_t \rangle_T \rangle\!\rangle = \frac{1}{\alpha}\frac{\partial(\beta f)}{\partial \beta} . \tag{3.36}$$

where f is the free energy per weight with the prior distribution on the student weight vectors kept fixed when the derivative with respect to β is taken. This is an average measure of how badly the network does on its training data set. The free energy calculated in §3.4 is the free energy related to the cost function which consists of the sum of the training energy and a term related to the prior distribution. The average training error may be calculated by differentiating βf

with the temperature dependence in the prior kept fixed. That is,

$$\epsilon_t = \frac{1}{\alpha}\frac{\partial}{\partial\beta}(\beta f|_{\beta\lambda=\text{const}})$$

After differentiating eq. (3.10) and rearranging, the average training error for a linear student learning an arbitrary teacher may be written as,

$$\epsilon_t = \frac{1}{2}\frac{\langle x\, g_0\rangle_\eta^2}{(\phi^2-\alpha)}\left(\lambda^2+\tilde{\gamma}_{\text{eff}}^2(\phi-1)^2\right)+\frac{T}{2\phi}\,. \tag{3.37}$$

The first term is in fact equal to $T^2\hat{q}_1/2\alpha$ which is a temperature independent quantity. This gives a meaning to the conjugate order parameter $\hat{q}_1$; this conjugate order parameter is α times the temperature independent part of the average training error divided by the square of the training temperature. The second term gives the temperature dependence of the training error, this term is independent of the teacher activation used and only depends on the weight decay parameter, λ and the number of patterns per weight, α through the function ϕ.

The average value of the cost function ϵ_c may also be calculated by a similar process, giving,

$$\epsilon_c = \epsilon_t + \frac{1}{2\alpha}\lambda q_0 \tag{3.38}$$

This simply adds the average of the penalty term to the average training error.

The average training error is related to the corrupted generalisation error (appendix D) by,

$$\epsilon_t = \frac{1}{(1+\mathcal{Q}')^2}\left(\epsilon_g{}' + \frac{T\mathcal{Q}'^2}{2}\right)\,, \tag{3.39}$$

where $\mathcal{Q}' = \beta\mathcal{Q}$ is the response function. This equation agrees with the result of Hansen [32] in the appropriate limit. The result suggests that the assumption

the minimising the training error will lead to better generalisation is not unreasonable. The function $1/(1+\mathcal{Q}')^2$ is a monotonically increasing function that tends to one as either α or $\lambda \rightarrow \infty$. This means that the training error is always less than the corrupted generalisation error for any learning problem. The function $1/(1+\mathcal{Q}')^2$ is only dependent on the number of patterns per weight and the weight decay parameter through the response function $\mathcal{Q}'$. The response function is dependent on the student architecture and independent of the teacher.

3.10 Optimal weight decay parameter

It has been shown by Krogh and Hertz[43] that there exists an optimal weight decay λ_{opt} which minimises the generalisation energy. This can be found by differentiating ϵ_g at finite λ.

Given the generalisation error from equation (3.34),

$$\frac{\partial \epsilon_g}{\partial \lambda} = \frac{1}{2}\frac{\partial q_0}{\partial \lambda} - \frac{\partial r}{\partial \lambda}\langle x g_0 \rangle$$

For a linear teacher, $\langle x g_0 \rangle_\eta = \langle x g_0 \rangle = \Omega$ and $\langle g_0^2 \rangle_\eta = \Omega^2(1+\tilde{\gamma}^2)$, thus the equation above reduces to,

$$\frac{\partial \epsilon_g}{\partial \lambda} = \frac{\alpha \Omega^2 \phi}{(\phi^2 - \alpha)^2}(\lambda - \tilde{\gamma}^2)\frac{\partial \phi}{\partial \lambda} + \frac{\partial \mathcal{Q}}{\partial \lambda}\,. \tag{3.40}$$

For zero T, $\frac{\partial Q}{\partial \lambda} = 0$ and putting $\frac{\partial \epsilon_g}{\partial \lambda} = 0$ yields

$$\lambda_{\text{opt}} = \tilde{\gamma}^2 \,, \tag{3.41}$$

which agrees with the result of Krogh and Hertz [43]. This result states that in order to get optimum generalisation, the width of the Gaussian prior on the student weights (the weight decay parameter λ) should be the same as the uncertainty (noise) in the training data

Now consider the linear teacher case for finite T. At finite temperature the condition, $\frac{\partial \epsilon_g}{\partial \lambda} = 0$ gives

$$4\,\alpha\lambda^2\Omega^2\left(\lambda - \tilde{\gamma}^2\right) + T\psi^{\frac{3}{2}}(\alpha - 1) - T\psi(\psi - \lambda(1 + \alpha + \lambda)) = 0 \,,$$

where $\psi = ((1 + \alpha + \lambda)^2 - 4\,\alpha)$.

This equation may be solved numerically and the results for $\Omega = 1$ are presented in Fig. 3.1. The solutions tend to infinity as $\alpha\beta \rightarrow 0.5$ from below; above this temperature, the optimum λ is infinite. This value of the weight decay parameter corresponds to having a prior weight distribution of zero weights. The large λ limit studied in the next chapter, eq. (4.15) agrees with this result since above $\alpha\beta = 0.5$ the generalisation error degrades. Thus an initial temperature for an annealing schedule may be postulated as $T_{\text{init}} = 2\alpha$.

For the larger values of T, the optimum generalisation error is not significantly less than the surrounding values and therefore training at λ_{opt} is not strictly necessary. The effect of the noise on the training set is to increase the optimal λ at low values of T/α.

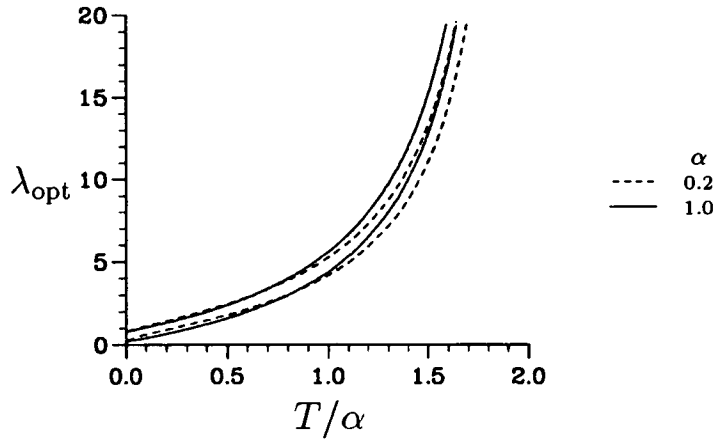


Figure 3.1. Optimal weight decay against temperature for effective gain one , $\Omega^2 = 1.0$, and two different noise levels, $\tilde{\gamma}^2 = 0.2$, (lower curves) $\tilde{\gamma}^2 = 0.8$ (upper curves)

For an arbitrary teacher the optimum weight decay for the corrupted generalisation error $\epsilon_g{}'$ may be considered since this gives a much simpler expression. The derivative of $\epsilon_g{}'$ with respect to the weight decay, λ, gives,

$$\frac{\partial \epsilon_g{}'}{\partial \lambda} = \frac{\alpha\phi}{(\phi^2-\alpha)^2}\left(\langle xg_0\rangle_\eta^2(1+\lambda) - \left\langle g_0^2\right\rangle_\eta\right)\frac{\partial \phi}{\partial \lambda} + \frac{\partial \mathcal{Q}}{\partial \lambda}\,,$$

which at zero T gives the solution,

$$\lambda_{opt} = \frac{\langle g_0^2\rangle_\eta}{\langle xg_0\rangle_\eta^2} - 1 = \tilde{\gamma}_{\text{eff}}^2\,. \tag{3.42}$$

This condition sets the weight decay equal to the effective noise level on the teacher, $\langle g_0^2\rangle_\eta - \langle xg_0\rangle_\eta^2$ divided by the effective gain of the teacher squared, $\langle xg_0\rangle_\eta^2$. This is similar to the condition for the linear teacher found previously. For nonlinear teachers, the optimum weight decay may be evaluated numerically. The optimum weight decay for a linear student learning a tanh(.) teacher for

different gains and additive noise are presented in table (3.1). It can be seen

	Gain Ω		
Noise $\tilde{\gamma}^2$	0.5	1.0	2.0
0.0	0.016	0.075	0.194
0.2	0.225	0.305	0.454
1.0	1.075	1.257	1.527

Table 3.1. Table of optimum weight decays λ_{opt} for linear student learns tanh(.) teacher.

from the table that the effect of the nonlinearity in the teacher is to increase the optimum weight decay by an amount that is related to the gain of the tanh(.) function. This can be explained, since the input examples are drawn from a uniform Gaussian distribution, the smaller gains mean that the teacher activation function is linear across more of the region from which the examples are selected, and hence the linear student is able to learn more of the teacher. For a linear student learning a binary teacher, the optimal weight decay is given by,

$$\lambda_{\text{opt}} = \frac{1}{2}\sqrt{2\pi(1+\tilde{\gamma}^2)} - 1 \ ,$$

which has a value ~ 0.25 for zero noise. Hence the generalisation of a linear student learning a binary teacher can be improved by using a weight decay set to this value.

3.11 Concluding remarks

In this chapter, the formalism developed by Seung *et al* [64] and outlined in the previous chapter is extended to enable static noise on the data set as well as a

weight decay to be studied. The model considered is a linear student learning both linear and nonlinear noisy teachers. The main performance measure used is the generalisation error. This comes in two forms, the clean generalisation error which compares the student to the uncorrupted teacher and the corrupted generalisation error. The corrupted generalisation error may be of more use in a practical situation since it is closer to the performance measure readily available. The average generalisation error ϵ_g is a rather artificial performance measure, since the test set consists of the complete example space, hence the test set contains the training set and so the error is not an error on novel examples, it is an error on a novel example plus the training set.

The case of a linear student learning a nonlinear teacher turns out to be the same as a linear student learning a linear teacher with an increased amount of noise and a different gain on the teacher.

There is an optimum weight decay parameter that minimises the generalisation error for a given training temperature and noise level on the data. This may be used to improve the performance of a network where the data has been corrupted. The prescription for setting the optimal weight decay parameter needs knowledge of the amount of noise on the data. Alternatively, this could be used as a method of estimating the noise on the data. Find the weight decay parameter that gives the optimal generalisation performance and the noise level on the data may be calculated. In the case of an unrealisable rule (linear student learning nonlinear teacher) the optimal weight decay contains a part that is reducing the error due to the non-linearity of the teacher. This shows that even when the student is poorly matched to the teacher, performance can be improved by using a weight decay.

The generalisation error studied in this thesis is that suggested by the standard statistical mechanics view; the mean square error. This is optimised by minimising both the bias and the variance of the posterior student distribution with respect to the training parameters (hyperparameters), which favours training at zero temperature, *i.e.*, selecting the MAP student weight vector. The Bayesian view says that the whole ensemble of students should be used with the variance of the posterior optimally equal to the variance of the noise on the data [9, 49]. The Bayesian generalisation error is then simply the squared bias. For the Gaussian distributions studied, it turns out that the MAP estimate and the average over the posterior distribution are the same, hence the two methods agree on the optimal weight decay parameter.

The performance measures calculated in this chapter will now be evaluated for certain limits of the main parameters. The limits considered will correspond to different training regimes.

Chapter 4

Noisy data and weight decay - Limits

In this chapter, the performance measures calculated in the previous chapter are evaluated for a number of different limits of the main parameters; the number of patterns per example, α, the weight decay parameter λ and the training temperature, T. The limits that will be looked at are,

- The zero weight decay limit, which is identical to the pseudo inverse solution. This enable comparisons between the weight decay and the pseudo inverse solution to be drawn.
- The zero temperature, $\beta \rightarrow \infty$ limit. This limit is characterised by a single student weight vector. In this limit, the generalisation error is minimised with respect to the temperature.

[0]Part of the work in this chapter has been published in [14]

- The large α limit. Here the number of examples per weight tends to infinity and so the network is able to use the data to overcome, either the noise on the data or an unsuitable prior.
- The large weight decay limit considers the effect of having a high degree of belief in the prior.
- The zero noise limit will be studied to enable comparisons between the weight decay prior and a spherical constraint to be drawn, as well as studying the effect of noise.

Both the case of a linear student learning a linear teacher and a linear student learning a nonlinear teacher will be examined under these conditions. The nonlinear teacher is chosen to have a hyperbolic tangent activation function.

4.1 Linear student, Linear Teacher

The simplest problem is where the teacher has a linear activation function with gain Ω, the student is also linear and has gain one.

4.1.1 Zero T, Zero λ

The $\lambda \to 0$ limit corresponds to training with an infinitesimally small weight decay term. However with $\lambda = 0$, the integration over the weights performed in the evaluation of the partition function eq. (3.3) is undefined. Thus the limit $T, \lambda \to 0$ is taken with $\lambda' = \beta\lambda$ constant giving a finite distribution of weights.

The zero weight decay limit is equivalent to the pseudo inverse solution. In this limit, the difference between the overlaps, $\mathcal{Q}$ eq. (3.17), gives two solutions

$$\mathcal{Q}(T,\lambda\to 0) = \begin{cases} \frac{1-\alpha}{\lambda'} & \alpha \leq 1 \\ 0 & \alpha > 1 \end{cases},$$

where $\lambda' = \beta\lambda$. Using these solutions in the solved saddle point equations (3.29), (3.30) and (3.18) to evaluate the physical order parameters gives;

$$q_0(T,\lambda\to 0) = \begin{cases} \alpha\Omega^2\left(1+\frac{\tilde{\gamma}^2}{1-\alpha}\right)+\frac{1-\alpha}{\lambda'} & \alpha \leq 1 \\ \Omega^2\left(1+\frac{\tilde{\gamma}^2}{\alpha-1}\right) & \alpha > 1 \end{cases}, \tag{4.1}$$

$$q_1(T,\lambda\to 0) = \begin{cases} \alpha\Omega^2\left(1+\frac{\tilde{\gamma}^2}{1-\alpha}\right) & \alpha \leq 1 \\ \Omega^2\left(1+\frac{\tilde{\gamma}^2}{\alpha-1}\right) & \alpha > 1 \end{cases}, \tag{4.2}$$

$$r(T,\lambda\to 0) = \begin{cases} \alpha\Omega & \alpha \leq 1 \\ \Omega & \alpha > 1 \end{cases}. \tag{4.3}$$

Substituting these values into the generalisation error, eq. (3.34) yields

$$\epsilon_g(T,\lambda\to 0) = \begin{cases} \frac{1}{2}(1-\alpha)(\Omega^2+\frac{1}{\lambda'})+\frac{\alpha\Omega^2\tilde{\gamma}^2}{2(1-\alpha)} & \text{for } \alpha \leq 1 \\ \frac{\Omega^2\tilde{\gamma}^2}{2(\alpha-1)} & \text{for } \alpha > 1 \end{cases}. \tag{4.4}$$

The average training error is given by eq. (3.37):

$$\epsilon_t(T,\lambda\to 0) = \begin{cases} 0 & \alpha \leq 1 \\ \frac{\Omega^2\tilde{\gamma}^2}{2\alpha}(\alpha-1) & \alpha > 1 \end{cases}. \tag{4.5}$$

It can be seen from eq. (4.1) that the effect of the noise added to the training set is to increase the average length of the student vectors, this in turn means that

the generalisation error is expected to be higher since, for a linear perceptron, the student can only generalise perfectly when it has the same direction and effective gain as the teacher. This cannot occur since from eq. (4.1), the noise draws the effective student gain away from the effective teacher gain Ω. This is confirmed in eq. (4.4) which shows noise degrading the generalisation error. The effect of noise can be reduced by presenting more patterns, that is increasing α. For $\alpha \leq 1$ the parameter λ' may be increased to reduce the generalisation error. Since $\lambda' = \beta\lambda$, in the zero T, λ limit, this corresponds to taking T to zero faster than λ, that is, training with an infinitesimally small but finite weight decay. This solution is the pseudo inverse solution [37].

For a finite noise level, there is a discontinuity at $\alpha = 1$ in the average generalisation error. For this value of α, the examples specify the weights, however, the noise causes the student weight vector to be arbitrarily far from the correct solution on average. As more patterns are presented, the student weights approach the teacher and so the errors decrease. Above $\alpha = 1$ the training error for noisy data is increased from zero, as seen in eq. (4.5). This is because as α increases above one, the network cannot learn the random noise present on the data and hence cannot learn the examples exactly

Some of the results presented by Krogh and Hertz [43] are equivalent to taking the zero T, λ limit with $\lambda' \to \infty$ and normalising the teacher vector to be of length one, that is $\Omega^2 = 1$. Since in this limit $\mathcal{Q} = 0$, the average overlap between replicas is always one, $q' = 1$; this means that there is only one actual student solution to the learning problem (the pseudo inverse). The average overlap between student

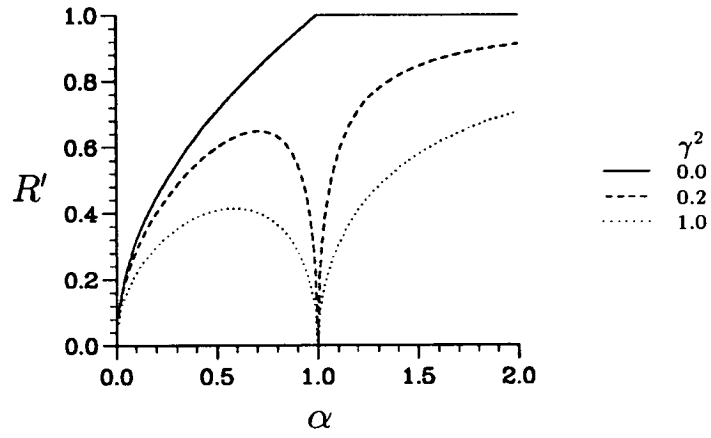


Figure 4.1. Average overlap between student and teacher weight vectors, R' against α for zero temperature and weight decay.

and teacher is given by

$$R'(T, \lambda \to 0) = \begin{cases} \sqrt{\frac{\alpha}{1+\tilde{\gamma}^2/(1-\alpha)}} & \alpha \leq 1 \\ \frac{1}{\sqrt{1+\tilde{\gamma}^2/(\alpha-1)}} & \alpha > 1 \end{cases} . \tag{4.6}$$

The function R' is presented in Fig. 4.1 for a number of different noise levels. Inserting the limits into eq. (4.4), gives the generalisation error for $T, \lambda \to 0$ and $\Omega = 1, \lambda' \to \infty$,

$$\epsilon_g(T, \lambda \to 0) = \begin{cases} \frac{1}{2}(1-\alpha) + \frac{\alpha\gamma^2}{2(1-\alpha)} & \alpha \leq 1 \\ \frac{\gamma^2}{2(\alpha-1)} & \alpha > 1 \end{cases} , \tag{4.7}$$

which is in exact agreement with the generalisation error calculated for a linear perceptron by Krogh and Hertz [43] apart from the factor of two due to a difference in definition.

4.1.2 Zero T, finite λ

Here, the temperature is held at zero whilst the effect of a finite weight decay is investigated. Zero temperature corresponds to learning with no dynamic noise on the weight updates. In this limit, $\mathcal{Q} = 0, \Rightarrow q' = 1$, *i.e.*, there is only one solution in the student weight space and hence all the replicas have the same student weight vector.

The training and generalisation errors at zero temperature are given by

$$\epsilon_t(T=0) = \frac{\Omega^2}{2}\frac{(\lambda^2 + \tilde{\gamma}^2(\phi-1)^2)}{(\phi^2-\alpha)} \tag{4.8}$$

$$\epsilon_g(T=0) = \frac{\Omega^2}{2}\left(1 - \frac{\alpha}{\phi} + \frac{\alpha}{\phi^2-\alpha}(\lambda - \tilde{\gamma}^2)\right) \tag{4.9}$$

where, $\phi = \frac{1}{2}(1+\alpha+\lambda+\sqrt{(1+\alpha+\lambda)^2 - 4\,\alpha})$. Since the gain of the teacher, Ω appears as a scale factor, a teacher of unit length will be studied, *i.e.*, $\Omega = 1$ for the plots in this section. The effect of a weight decay is to increase the initial training error above zero. This is because a non zero weight decay term takes the student's weight vector away from that which gives zero training error. Asymptotically, as the number of patterns per weight α tends to infinity, the training and corrupted generalisation errors are given by,

$$\epsilon_g'(T=0, \alpha\rightarrow\infty) = \frac{1}{2}\Omega^2\tilde{\gamma}^2(1+\frac{1}{\alpha}) + O(\alpha^{-2})\,, \tag{4.10}$$

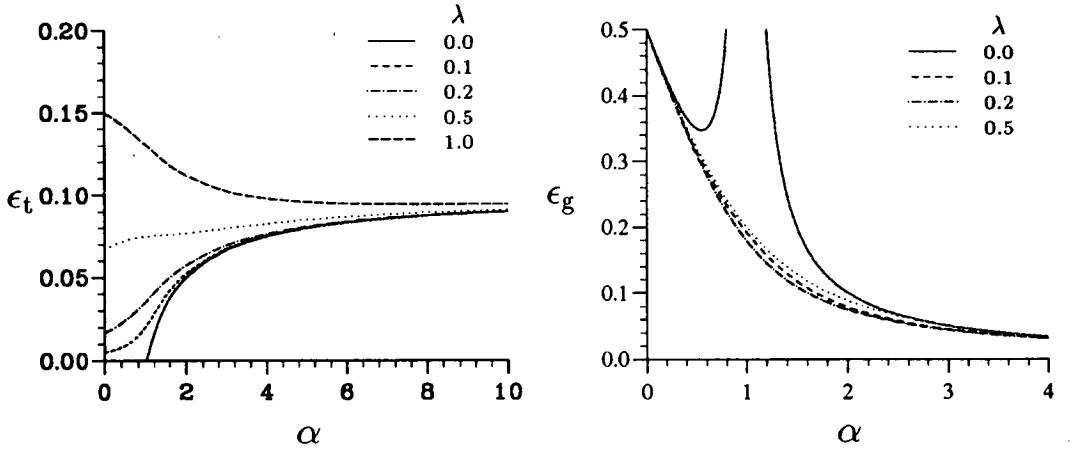
$$\epsilon_t(T=0, \alpha\rightarrow\infty) = \frac{1}{2}\Omega^2\tilde{\gamma}^2(1-\frac{1}{\alpha}) + O(\alpha^{-2})\,. \tag{4.11}$$

The result of Seung *et al* [64] that the training and corrupted generalisation error, ϵ_g', approach the same value from below and above respectively for large α can be seen to hold for all λ. The equivalence of the average training and corrupted

generalisation errors as $\alpha \to \infty$ is expected, since as the number of examples per weight tends to infinity the example set spans input space, this implies that the average error on the training set is the same as the expected error on a random input: the generalisation error. The λ independence of the asymptotic value is due to the large amount of data overwhelming the prior and hence for large α, the student is independent of the prior. The asymptotic form of the cosine of the angle between the student and teacher weight vectors is,

$$R'(T = 0, \alpha \to \infty) = 1 - \frac{\tilde{\gamma}^2}{2\,\alpha} + O(\alpha^{-2}) \,.$$

Thus as the number of examples increases, the student becomes collinear with the teacher, again the prior is not present in the asymptotic form for the same reason as above.



(a) training error (b) Generalisation error

Figure 4.2. Training and generalisation error for zero temperature, noise $\gamma^2 = 0.2$ added and different weight decays λ.

The average cost function can also be evaluated for zero temperature, it is given

by,

$$\epsilon_c(T=0) = \frac{\lambda\Omega^2}{2}\left(1 - \frac{(\lambda-\tilde{\gamma}^2)}{(\phi^2-\alpha)}\left(1 + \frac{\lambda\phi^2}{(\phi-\alpha)^2}\right)\right)$$

A plot of this is presented in Fig. 4.3 for noise $\tilde{\gamma}^2 = 0.2$ and different values of the weight decay parameter. For optimum temperature and weight decay (in the minimum generalisation error sense, see §3.10), $T = 0, \lambda = \tilde{\gamma}^2$, the cost function is constant and equals half of the actual noise level on the teacher, $\gamma^2/2$. This constant is the asymptotic training and corrupted generalisation error, *i.e.*, the error due to the noise on the data. This suggests that if the cost function is below this value, the network may not be regularised sufficiently, causing over-fitting and if the cost function is above to this value, the network may be "over regularised" causing the prior to disrupt the data.

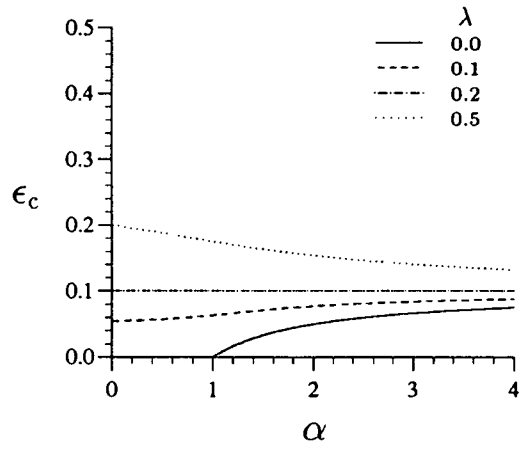
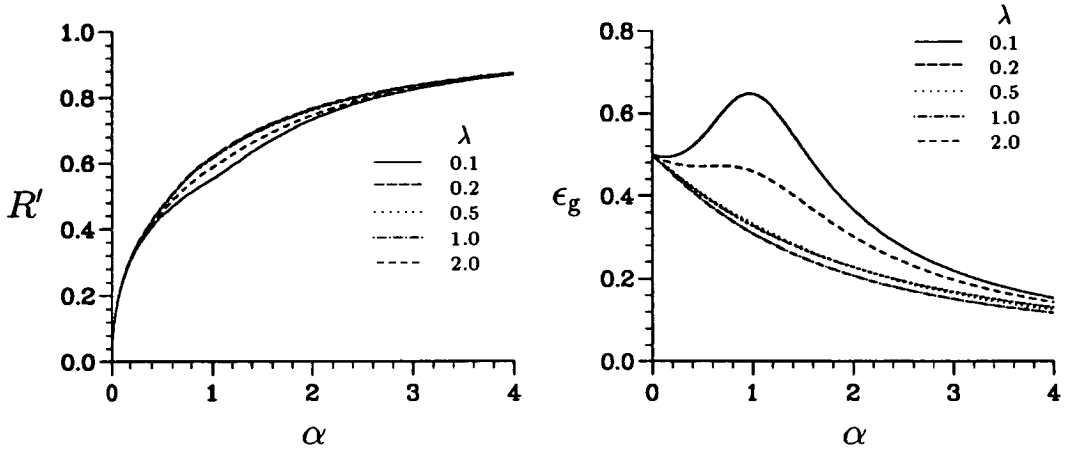


Figure 4.3. Average cost function for zero temperature , noise $\tilde{\gamma}^2 = 0.2$ added, plotted for different values of λ.

If the noise level added to the training data is increased the effect of an inappropriately small weight decay may be investigated. The generalisation error and the average overlap between student and teacher are shown in Fig. 4.4 for a large



(a) Average Overlap (b) Generalisation error

Figure 4.4. Generalisation error ϵ_g and the average overlap, R' plotted for zero temperature, noise $\tilde{\gamma}^2 = 1.0$ and different weight decays, λ.

noise level, $\tilde{\gamma}^2 = 1.0$. The effect of a weight decay that is too small is more apparent on the generalisation error than the cosine of the angle between student and teacher weight vector. This is because the generalisation error has an additive component dependent on the length of the student weight vector, which is also affected by the noise.

The smaller weight decays in Fig. 4.4 are not large enough to regularise the high noise level and a peak in the generalisation error is present for the smaller values of the weight decay parameter λ. The peak is related to the divergence that appears at $\alpha = 1$ for the pseudo inverse solution, $\lambda \rightarrow 0$, and occurs when the student is most sensitive to noise. The actual location of the peak is obtained by solving a quartic equation in α. A plot of the location of the generalisation error peak α_{max} versus the value of the weight decay parameter is presented in Fig. 4.5. There is a value of the weight decay parameter (denoted λ_{p}) above which there

is *no* peak in the generalisation error. Thus if $\lambda > \lambda_p(\gamma)$, the generalisation error decreases as the number of examples in the training set increases for all $\alpha > 0$. For $\lambda < \lambda_p$ the generalisation error can increase as more examples are presented for $\alpha < \alpha_{max}$, this suggests that the weight decay should be larger than λ_p to get continual improvement as the number of examples is increased. For $\tilde{\gamma}^2 \geq 3$ $\lambda_p = (\tilde{\gamma}^2 - 1)/2$ and $\alpha_{max}(\lambda_p) = 0$. If $\tilde{\gamma}^2 < 3$, the expression for λ_p is less simple but still depends on the noise level γ. The location of the peak for $\lambda = \lambda_p$ and $\tilde{\gamma}^2 < 3$ is at nonzero α, this can be seen in Fig. 4.5 for the curves corresponding to $\tilde{\gamma}^2 = 1.0$ and 2.0. This is because for $\tilde{\gamma}^2 < 3$, there is a local minimum in the generalisation error at $0 < \alpha < \alpha_{max}$ and as $\lambda \rightarrow \lambda_p$ from below, the local minimum and the maximum combine at $\alpha_{max}(\lambda_p) > 0$. For $\tilde{\gamma}^2 > 3$, the local minimum does not exist for $\alpha > 0$, so as $\lambda \rightarrow \lambda_p$, the location of the peak tends towards 0. As the noise level tends to infinity, the location of the peak is given by $\alpha_{max} = 1 + \lambda$ for $\lambda \ll \tilde{\gamma}^2$, as λ increases, however, the height of this peak decreases. The peak in the generalisation error implies that the network overfits the data, *i.e.*, it is under-regularised.

The peak in the generalisation error is related to a peak in the length of the student, the location of this peak may also be calculated. The smaller values of the weight decay allow the student vector to grow larger than the teacher causing a degradation of the generalisation error. This may be seen in Fig. 4.6 where the average length squared of the student is plotted for noise $\tilde{\gamma}^2 = 1.0$ and different weight decay parameters; in this case, a peak does not appear if $\lambda > \tilde{\gamma}^2/2$. As $\tilde{\gamma}^2/2 - \lambda$ becomes small the location of the maximum tends to infinity, whilst the height of the peak becomes smaller. Again as $\tilde{\gamma}^2 \rightarrow \infty$, the peak is at $\alpha_{max} = 1 + \lambda$.

As the number of examples per weight, α tends to infinity, the asymptotic form

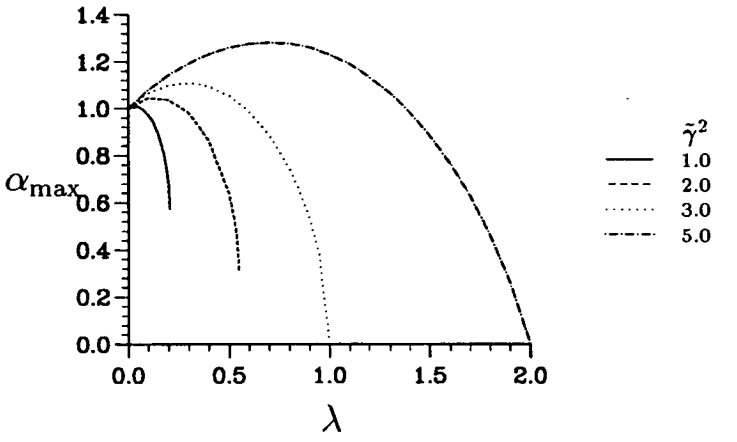


Figure 4.5. Location of maximum as a function of α of the generalisation error plotted against the weight decay parameter λ for different noise levels on the data in the zero temperature limit.

of the average student length squared at zero T is,

$$q_0(T=0, \alpha \to \infty) = \Omega^2 \left\{ 1 - \frac{1}{\alpha}(2\lambda - \tilde{\gamma}^2) + O(\alpha^{-2}) \right\}$$

This shows that for $\lambda > \tilde{\gamma}^2/2$, the average length squared of a student is less than one and thus there is not a peak in the length of the student; for$\lambda < \tilde{\gamma}^2/2$ the average length of the student weight vector is able to increase above one, as $\alpha \to \infty$, the length returns to that of the teacher.

The generalisation error plotted as a function of the weight decay parameter λ is presented in Fig. 4.7. The optimum weight decay parameter estimated in section 3.10 can be picked out. For the smaller values of α, the minimum is not significantly lower than the surrounding generalisation error, here the data is not able to specify the weights fully. As the number of examples becomes comparable to the number of weights, $\alpha \sim 1$, the minimum at λ_{opt} becomes more

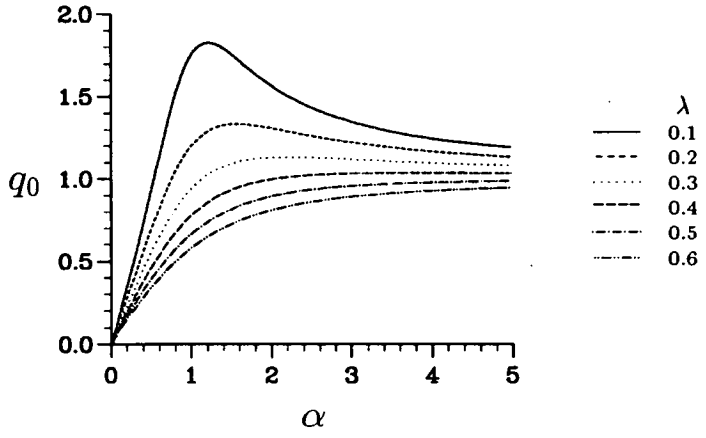


Figure 4.6. The average length squared of the student plotted against α for noise $\tilde{\gamma}^2 = 1.0$ added and different weight decays, λ.

pronounced, since here the network is most sensitive to noise. As α tends towards infinity the minimum is again not significantly lower than the surrounding values, since as $\alpha \rightarrow \infty$ the choice of prior is less important The asymptotic expansion of the corrupted generalisation error for large α, eq. (4.10), shows the lack of dependence on the prior for larger α with the first order correction independent of λ.

4.1.3 Zero noise, $\gamma^2 = 0$

The zero γ limit corresponds to having an uncorrupted data set. This was studied by Seung *et al* [64] using a spherical constraint on the weights and hence it is possible to draw some comparisons between a weight decay term and a spherical constraint on the weights. For the weight decay case, the additional order parameter q_0 is associated with the mean length of a student weight vector. Using

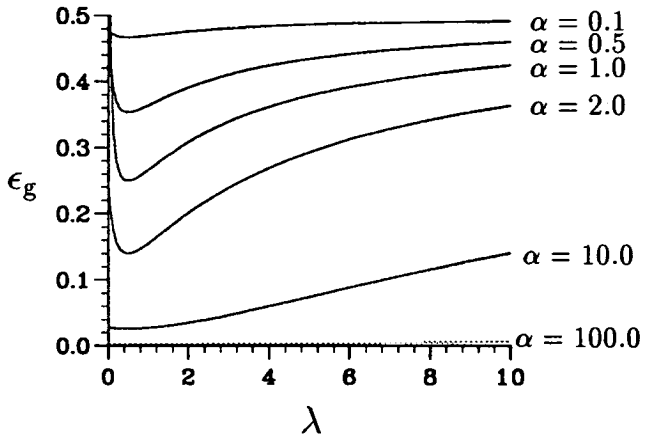


Figure 4.7. The generalisation error, ϵ_g against the weight decay parameter, λ for different values of α and noise variance 0.5 added to the data.

a spherical constraint, the length of the student weight vector, q_0, as well as the length of the teacher vector, Ω are constrained to be one.

In the zero T, λ limit with $\gamma^2 = 0$, from eq. (4.1)

$$q_0(T, \lambda \to 0) = \begin{cases} \alpha\Omega^2 + \frac{1-\alpha}{\lambda'} & \alpha \leq 1 \\ \Omega^2 & \alpha > 1 \end{cases} .$$

Therefore having a Gaussian distribution of weights such that $\lambda' = \beta\lambda = \frac{1}{\Omega^2}$ will result in $q_0 = \Omega^2$ for all α. This means that the average length squared of the student vector is equal to the length squared of the teacher, which is similar to a spherical constraint but not identical, since it is the average rather than actual length of the student weight vector which lies on the sphere. To mimic the spherical constraint of Seung *et al*, Ω is set to be one.

From equations (4.1) and (4.2) with zero γ, T, λ and assuming the distribution of

weights above, that is $\lambda' = \frac{1}{\Omega^2}$:

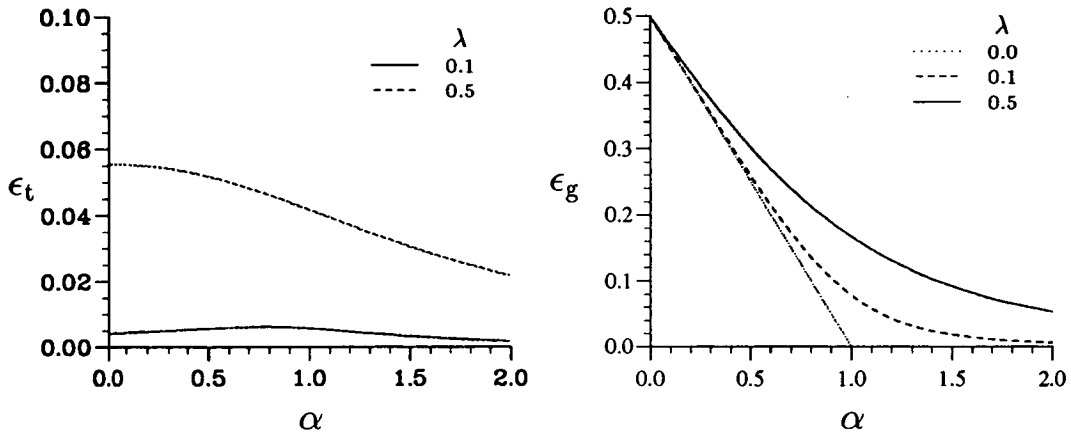
$$q'(\lambda' = 1/\Omega^2) = \begin{cases} \alpha & \alpha \leq 1 \\ 1 & \alpha > 1 \end{cases} .$$

Hence for $\alpha > 1$, the average overlap between replicas is one and therefore all the replicas tend towards the same vector and so there is only one solution within the student weight space. For $\alpha \leq 1$, the number of possible student solutions is greater than one due to the fact that there are less than N equations specifying N unknowns, therefore the system has some freedom to find a solution. The same distribution of weights gives, from eq. (4.3), the average overlap between student and teacher,

$$R'(\lambda' = 1/\Omega^2) = \begin{cases} \alpha & \alpha \leq 1 \\ 1 & \alpha > 1 \end{cases} .$$

The average overlap with the teacher tends towards one as α increases through 1. This then makes the generalisation error $\epsilon_g = 0$ for $\alpha > 1$, as can be seen from eq. (4.4). For $\alpha > 1$ the training set more than specifies the student and so there can only be one solution, namely the teacher. In the region, $\alpha \leq 1$, $\epsilon_g = (1 - \alpha)\Omega^2$. The above results again agree with Seung *et al* for $\Omega^2 = 1$.

At zero temperature, zero λ and zero γ^2, ϵ_t is zero for all α. This is as expected, since in this case the student can always learn the data set exactly, that is, the problem is realisable. At finite values of λ, with zero static noise, the training and generalisation errors are increased from their values at $\lambda = 0$, as can be seen in Fig. 4.8. At finite temperature, the errors are also increased. The presence of the weight decay in the cost function means that the learning algorithm no longer enforces $E_t = 0$ and thus the training error and generalisation error are non-zero.



(a) Training error (b) Generalisation error

Figure 4.8. The generalisation error,ϵ_g and training error ϵ_t for zero noise on the data set($\gamma = 0$) and zero temperature plotted for different weight decay parameters λ.

4.1.4 Finite T

At finite temperature the performance measures are altered slightly. The temperature dependence adds a term to the generalisation and training errors as in eq. (3.37) and (3.34). The difference between the overlaps (T times the response function $\mathcal{Q}'$), $\mathcal{Q}$ is temperature dependent and may be written in terms of the parameter ϕ as $\mathcal{Q} = T/(\phi - 1)$ which grows linearly with the temperature. The temperature dependent contribution to the generalisation error is equal to $\mathcal{Q}/2$, hence better generalisation error is achieved at zero temperature. The measure of the overlap between student solutions is $q' = q_1/q_0$. As the number of examples tends to infinity, this is given by,

$$q' = 1 - \frac{T}{\alpha\Omega^2} + O(\alpha^{-2}) .$$

The temperature increases the number of possible student solutions by not enforcing zero training error, this reduces the average overlap between solutions as seen above.

The training error for a linear student learning a linear teacher is given by eq. (3.37),

$$\epsilon_t = \frac{1}{2}\left(\frac{\Omega^2}{(\phi^2-\alpha)}\left(\lambda^2+\tilde{\gamma}^2(\phi-1)^2\right)+\frac{T}{\phi}\right). \tag{4.12}$$

The effect of the training noise is to increase the training error. When the temperature divided by the effective gain on the teacher T/Ω^2 and the weight decay λ are equal to the optimal weight decay parameter, $\lambda_{\text{opt}} = \tilde{\gamma}^2$, the training error is constant for all α with value $\tilde{\gamma}^2/2$, *i.e.*, the error due to the noise. These parameter values are identified with the optimal case of the network minimising the variance of the student output [9].

The asymptotic behaviour of the training and generalisation errors at finite temperature are given by,

$$\epsilon_g = \frac{1}{2\alpha}(T+\Omega^2\tilde{\gamma}^2)+O(\alpha^{-2}) \tag{4.13}$$

$$\epsilon_t = \frac{1}{2}\left\{\Omega^2\tilde{\gamma}^2+\frac{1}{\alpha}(T-\Omega^2\tilde{\gamma}^2)\right\}+O(\alpha^{-2}). \tag{4.14}$$

These results are in agreement with those presented by Seung *et al* for large α. The large α behaviour of the perceptron using a weight decay is identical to that predicted using a spherical constraint. The prior has no influence on the large α behaviour as expected since in this limit, the data is able to swamp the prior.

4.1.5 Large λ

The large λ limit corresponds to penalising weight vectors heavily or alternatively to narrowing the Gaussian distribution from which the student weight vectors are drawn. The generalisation error in this limit is

$$\epsilon_g = \frac{1}{2}\left\{\Omega^2 + \frac{(T - 2\,\alpha\Omega^2)}{\lambda} - \frac{\alpha}{\lambda^2}(3\,\Omega^2(\alpha+1) - \gamma^2 + T) + O(\lambda^{-3})\right\}. \quad (4.15)$$

Thus for large weight decays, the amount of noise in the data set has little effect on the generalisation error since it appears at $O(\lambda^{-2})$. The order λ^{-1} term increases the error for $T > 2\,\alpha\Omega^2$ which suggests that with large weight decays, better results can be obtained by training at temperatures less than $2\,\alpha\Omega^2$. See §3.10

4.2 Linear student, Nonlinear teacher

The performance measures may be calculated for nonlinear teachers. The averages $\langle\, x\, g_0 \,\rangle_\eta$ and $\langle\, g_0^2 \,\rangle_\eta$ for an arbitrary teacher activation function can be calculated numerically. The order parameters and performance measures may be rewritten in terms of the effective linear gain of the teacher, $\langle\, x g_0 \,\rangle_\eta$ and the effective noise on the teacher, $\gamma_{\rm eff}^2 = \langle\, g_0^2 \,\rangle - \langle\, x g_0 \,\rangle_\eta^2$ by using eq. (3.31) and (3.32). Hence the results for a linear student learning a nonlinear teacher can be written in terms of a linear student learning an effective linear teacher with some effective noise added to the training data.

The gain of the teacher activation function is controlled by the length of the

teacher vector Ω, this in turn controls the gradient of the teacher activation function around the origin. Since the examples are chosen from a Gaussian distribution centred on the origin, if the teacher activation is predominantly linear in this region, the linear student will be relatively successful at generalising since the student will have a similar response for similar activations. This corresponds to using a small teacher gain, Ω. However, if the teacher gain is large, the teacher activation function will be nonlinear in most of the region from which the examples will be chosen; in this case the linear students generalisation ability will be poor.

Using the saddle point equations eq. (3.18) - (3.23), the learning curves and the average overlaps between student and teacher may be calculated for similar limits as in the previous section. A typical nonlinear activation function is the tanh(.) function. In the remainder of this section, curves will be presented for a linear student learning a nonlinear teacher with a tanh(.) activation function in the presence of noise using a weight decay.

4.2.1 Zero T, Zero λ

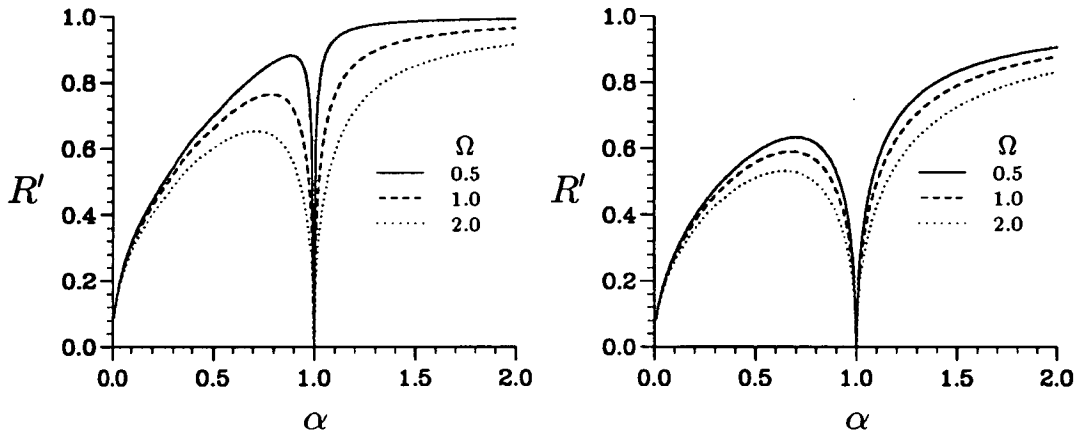
As for the linear teacher, for this limit the difference between the overlaps, $\mathcal{Q}$ is given by eq. (3.17). That is, $\mathcal{Q} = (1-\alpha)/\lambda'$, where $\lambda' = \beta\lambda$ for $\alpha \leq 1$ and $\mathcal{Q} = 0$ for $\alpha > 1$. If the $\lambda' \rightarrow \infty$ limit is considered, this is equivalent to the pseudo inverse solution, in this limit, $\mathcal{Q} = 0$ for all α. The normalised average overlap between student and teacher, R', plotted against the number of patterns per weight, α, is presented in Fig. 4.9 for two noise levels. As $\alpha \rightarrow 1$, the normalised overlap in both cases tends towards zero. This means that the student weight is moving away

from the teacher vector. The fact that the overlap reduces to zero is an artifact of the infinite system, one would expect that for a finite system, the overlap would reduce to a non zero value. This singularity at $\alpha = 1$ is similar to that found for a linear teacher when noise was added to the training data. However the effect is present in this case even when there is no noise on the training data. In the zero weight decay limit, the uncorrupted generalisation error is, from eq. (3.34),

$$\epsilon_g = \begin{cases} \frac{1}{2}\langle g_0^2 \rangle - \alpha \langle xg_0 \rangle \langle xg_0 \rangle_\eta + \frac{\alpha\left(\langle g_0^2 \rangle_\eta - \alpha \langle xg_0 \rangle_\eta^2\right)}{2(1-\alpha)} & \alpha \leq 1 \\ \frac{1}{2}\langle g_0^2 \rangle - \alpha \langle xg_0 \rangle \langle xg_0 \rangle_\eta + \frac{(\alpha-2)\langle xg_0 \rangle_\eta^2 + \langle g_0^2 \rangle_\eta}{2(\alpha-1)} & \alpha > 1 \end{cases}$$

The generalisation error compared to the noisy teacher is given by replacing the averages $\langle xg_0 \rangle$ and $\langle g_0^2 \rangle$ with their noisy equivalents in the equation above. The asymptotic value of the corrupted generalisation error, ϵ_g', for large α is given by the effective noise level of the teacher, $\gamma^2 = \langle xg_0 \rangle_\eta^2 \tilde{\gamma}_{\text{eff}}^2 = \langle g_0^2 \rangle_\eta - \langle xg_0 \rangle_\eta^2$. In the zero noise limit, the generalisation error will always have a singularity at $\alpha = 1$ as long as $\langle xg_0 \rangle^2 \neq \langle g_0^2 \rangle$, for a linear teacher the equality holds and thus the singularity disappears as was seen in the previous section. In the zero noise limit, the discontinuity is due to the fact that the linear student is modelling a nonlinear teacher. As α approaches one, each component of the student vector is fixed by an example. The examples that give an activation in the nonlinear region of the teacher activation function move the average student weight vector away from the solution that is correct on the linear part of the teacher activation function causing a singularity as in the case of the noisy linear teacher.

Noise added to the nonlinear teacher increases the effective noise level and hence reduces the overlap between student and teacher weight vectors for a particular number of examples presented as well as increasing the generalisation error.



(a) No added noise.

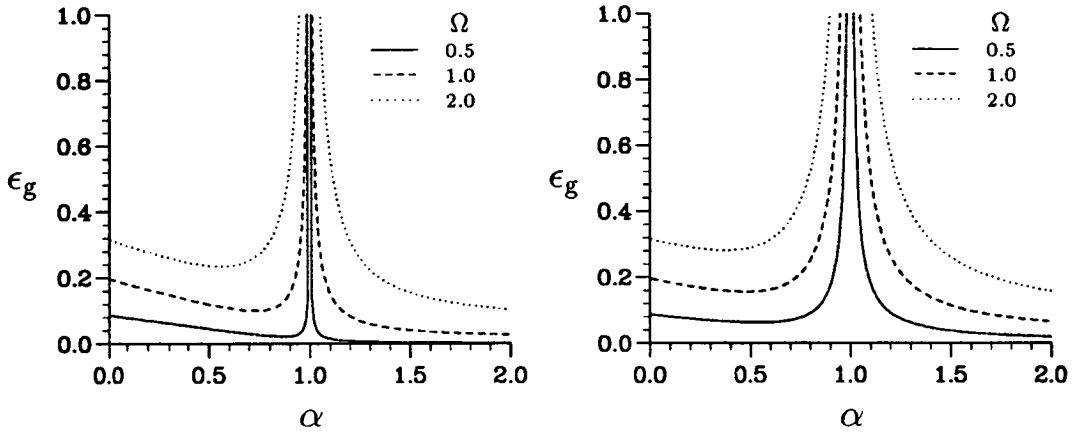
(b) Noise $\tilde{\gamma}^2 = 0.2$ added.

Figure 4.9. Normalised overlap between student and teacher, R', plotted against α for zero temperature and weight decay. The three curves correspond to three different effective gains, Ω (0.5,1.0,2.0).

The zero λ, pseudo-inverse limit of the average generalisation error, ϵ_g, plotted against the number of examples per weight, α, is presented in Fig. 4.10. There is a divergence at $\alpha = 1$ that corresponds to the drop in R' as seen in the previous figure. This discontinuity again shows that as $\alpha \rightarrow 1$, the student is learning the effective noise due to the nonlinear part of the teacher activation. The effect of noise is to spread the peak around the discontinuity at $\alpha = 1$.

4.2.2 Zero T, finite λ

A finite weight decay term is now considered. In the zero temperature limit, $\mathcal{Q}$ is always zero and the average overlap between solutions is one, *i.e.*, there is only one student solution. The average generalisation error ϵ_g is plotted against α for



(a) No added noise.

(b) Noise $\tilde{\gamma}^2 = 0.2$ added.

Figure 4.10. Average generalisation error ϵ_g against α for zero temperature and weight decay. The three curves correspond to three different effective gains, Ω.

different noise levels and different values of the effective weight decay parameter, λ in Fig. 4.11. This should be compared to the zero weight decay limit in Fig. 4.10. In the present figure Fig. 4.11, the divergence at $\alpha = 1$ has been removed. This is due to the weight decay constraining the effective noise. For the larger values of noise shown in the figure there is a residual part of the divergence present as a peak in the generalisation error due to the noise overwhelming the weight decay. The actual location of the peak in the generalisation error may be calculated as for the linear student learning a linear teacher, giving the same qualitative behaviour.

In Fig. 4.12 the average training error, ϵ_t is plotted against α for zero temperature. The training error for nonzero weight decay tends towards the asymptotic value for zero weight decay. The weight decay parameter increases the training error above the value it has when no weight decay is present.

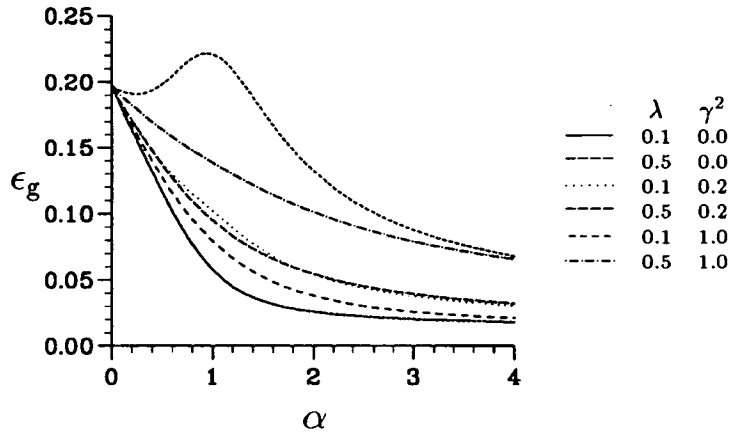


Figure 4.11. Average generalisation error ϵ_g against α for zero temperature and finite weight decay. Effective gain of teacher 1.0.

4.2.3 Large α limit

As for the linear teacher case, the asymptotic expansion of the performance measures may be calculated. The expansions for the average overlap between student and teacher, R', the average overlap between replicas, q', the corrupted generalisation error, $\epsilon_g{}'$ and the training error, ϵ_t are the same as the linear teacher case with the noise and gain replaced by the effective noise $\tilde{\gamma}^2_{\text{eff}}$ and effective gain $\langle xg_0 \rangle_{\text{eff}}$ respectively.

The asymptotic value of the uncorrupted generalisation error, ϵ_g as $\alpha \to \infty$, is given by

$$\frac{1}{2}\left(\left\langle g_0^2 \right\rangle + \langle xg_0 \rangle_\eta^2 - 2\langle xg_0 \rangle_\eta \langle xg_0 \rangle\right).$$

This compares to the simpler asymptotic value for the corrupted generalisation error, $\frac{1}{2}\left(\langle g_0^2 \rangle - \langle xg_0 \rangle_\eta^2\right)$, which is equal to the asymptotic training error. This

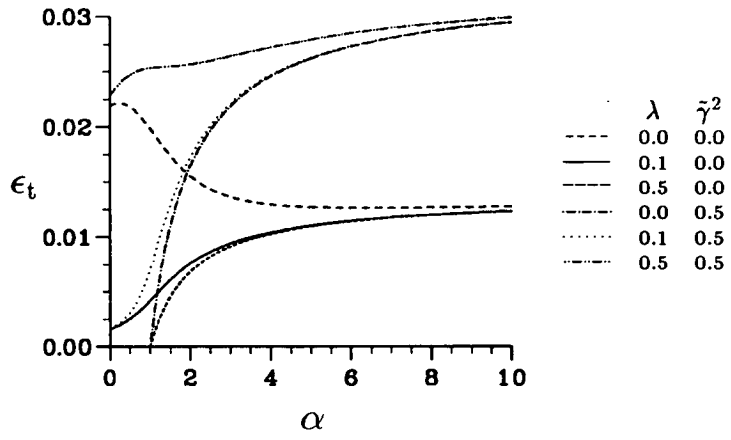


Figure 4.12. Average training error ϵ_t against α for zero temperature and finite weight decay. Effective gain 1.0

may be an argument for using the corrupted generalisation error as the performance measure: not only is it possible to calculate in practice, it also gives the same asymptotic value as the training error for large numbers of patterns.

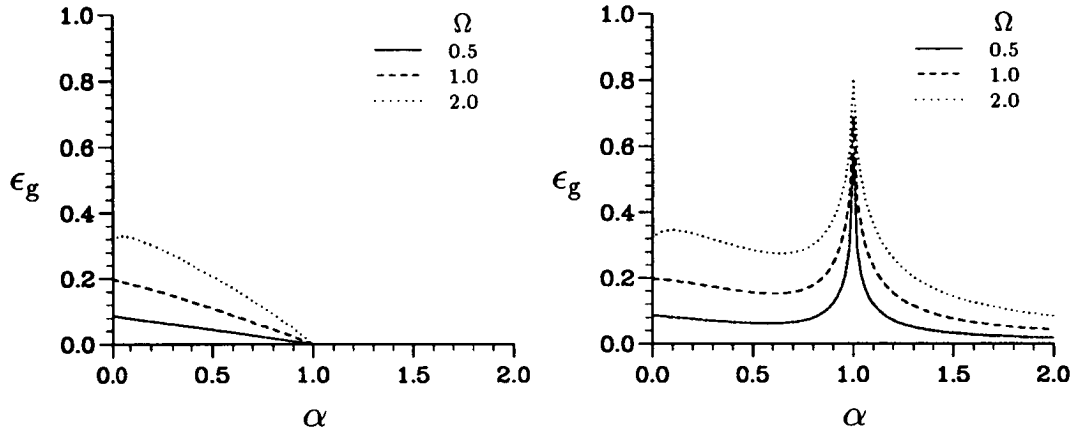
The asymptotic values of the generalisation and training errors are independent of the weight decay parameter chosen, they are dependent solely on the activation function and gain of the teacher, *i.e.*, the architecture of the teacher. The asymptotic value of the training error also gives an indication of how well the student should be expected to do, since it is equal to the residual error due to the noise on the teacher.

4.3 Other realisable rules

It is possible to to evaluate the performance for a student with an invertible nonlinear activation function, provided that the teacher has the same form of activation function, *i.e.*, $g(x) = g_0(kx)$ for some k (see §2.12). At zero temperature, the order parameters for these other rules are identical to those calculated for the linear teacher and student, the generalisation error can be calculated numerically. Plots for a student with tanh(.) activation function learning a teacher with tanh(.) activation function are presented in Fig. 4.13.

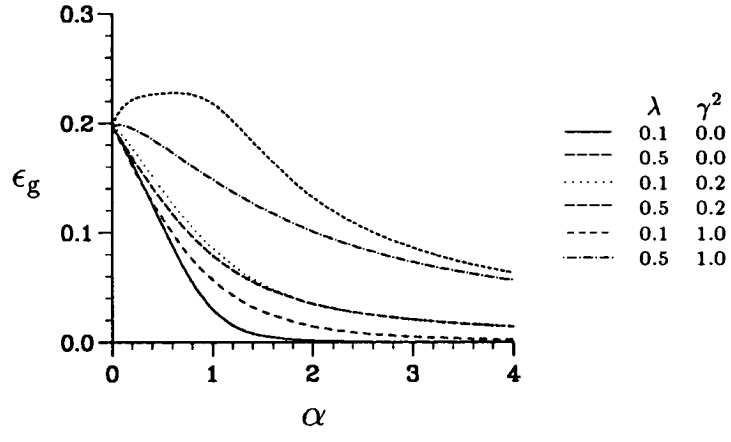
The graphs show the same form as those calculated for the linear student learning a linear teacher, as expected since the order parameters are the same, the performance must be qualitatively similar. The tanh(.) activation function provides a squashing of the peaks seen in the linear case.

The generalisation error of the system at zero α tends towards 0.5 as the effective gain of the teacher increases. This is because when $\alpha = 0$, the weight decay is not used, and the solution picked is the pseudo inverse, which gives an average student length of zero. In this case, the generalisation error is simply the average of the teacher activation function over the activation distribution, that is, the average of the square of a tanh(.) function with gain Ω over a Gaussian distribution of variance one; as Ω increases this tends to 0.5. It is not possible to use this method to calculate the performance of a binary student learning a binary teacher because the sgn(.) function is not invertible.



(a) No added noise.

(b) Noise $\tilde{\gamma}^2 = 0.2$ added.



(c) Non zero weight decay

Figure 4.13. Average generalisation error ϵ_g against α for zero temperature and weight decay with a nonlinear student (represented by a tanh(.) activation function) learning a nonlinear teacher with the same activation function. The three curves correspond to three different gains of the teacher.

4.4 Concluding remarks

In this chapter, the performance measures calculated in the previous chapter for a linear perceptron plus weight decay learning a noisy teacher are evaluated for a number of different limits of the parameters. The results show that the behaviour for a linear student learning a nonlinear teacher is qualitatively similar to a linear student learning a noisy teacher. For the linear student and teacher case and noisy data, there is a divergence in the generalisation error around $\alpha = 1$ in the zero weight decay limit. This is caused by the noise pulling the student solution away from the true rule. The divergence is present in the zero weight decay limit for a linear student learning a noise free nonlinear student, again emphasising the equivalence of a nonlinear teacher to a noisy linear teacher.

The divergence in the generalisation error could be controlled by using a finite weight decay parameter. However if the weight decay parameter was too small, the divergence could appear as a peak in the generalisation error meaning that there was not a continual improvement in performance as more examples were presented to the network.

The weight decay parameter and temperature that optimise the generalisation error, causes the cost function to be constant with the value of the error due to the noise. This implies that if the parameters are set such that the cost function is reduced below the error due to the teacher (noise on teacher or nonlinear teacher or both) the generalisation error is not optimised. This may be used as a method of selecting appropriate parameters for training a network. The training error was constant at the level of the error due to the noise/nonlinear teacher when the temperature and the weight decay parameter were proportional to the

optimum weight decay parameter. These parameter values minimise the variance of the student outputs. The results show that the cost function consisting of a penalty term plus the training error does indeed improve a networks performance in terms of its ability to generalise. The question arises as to whether a different penalty term may improve the networks performance further. This will be studied in the next chapter.

Chapter 5

A general penalty term

In the previous chapter, the effect of a weight decay term added to the training algorithm was studied for a simple network. Weight decay is equivalent to gradient descent on a cost function including the quadratic length of the student weight vector as a penalty term. The penalty term could be generalised to be any quadratic form of the student weight vector using some penalty matrix, Λ. Quadratic penalty terms have been used in general additive models of statistics [34] as a method of regularising multivariate linear regression; the generalised penalty term is related to a penalised least squares regression analysis in a similar way to weight decay and ridge regression.

This chapter will introduce the general penalty term and then calculate the free energy for a simple network trained using a general penalty term. From this free energy, the order parameters may be identified and hence the performance measures of the network calculated. The behaviour of the performance measures can then be studied for different penalty matrices.

5.1 Introducing the general penalty term

The basic calculation is similar to that performed in the previous chapter. However, in this case, the penalty term is written as a matrix term which allows correlations between components to be included.

The cost function is now defined as the usual sum of the error measure over the training set (the training error) plus a general penalty term,

$$\frac{1}{2}\mathbf{W}^T\Lambda\mathbf{W} \tag{5.1}$$

where the penalty matrix Λ is a positive definite symmetric $N \times N$ matrix and the superscript ^{T} denotes a transposed vector. The previous case of a standard weight decay term is reached if the matrix Λ is a multiple λ of the identity matrix, I.

With no information from the training data and using a gradient descent update rule, the general quadratic penalty term produces a weight decay term like,

$$\mathrm{W}_i^{\mathrm{new}} = (1 - \tau\lambda_{ii})\mathrm{W}_i^{\mathrm{old}} - \tau\sum_{j\neq i}\mathrm{W}_j^{\mathrm{old}}\lambda_{ij}\ , \tag{5.2}$$

where τ is the update step size, and λ_{ij} are the components of the penalty matrix Λ. The decay term includes the standard weight decay as well as a bias term that depends on the other components of the student weight vector. If the off diagonal terms of the penalty matrix are all negative the bias term tends to force the new student component towards the weighted average value of the other components.

As in the previous calculation, the model is a simple linear perceptron stochastically trained giving a posterior Gibbs distribution of student weights. The training data is generated from a known teacher network and may be corrupted by additive noise. The strategy adopted is to calculate the free energy of the system via the quenched average of the logarithm of partition function using the replica method. Returning to eq. (3.5) the free energy per weight f is given by,

$$-\beta f = \mathcal{G}_0 - \alpha \mathcal{G}_r \, ,$$

where $\mathcal{G}_r$ is the RS Hamiltonian which depends on the architecture of the student network being considered and $\mathcal{G}_0$ is the RS prior constrained Hamiltonian. For the linear student being studied, $\mathcal{G}_r$ is given by eq. (2.33). For an arbitrary student prior $\mathcal{G}_0$ is given by eq. (2.50),

$$\mathcal{G}_0 = -R\hat{R} - q_0\hat{q}_0 + \frac{1}{2}q_1\hat{q}_1 + \mathcal{J} \, , \tag{5.3}$$

where $\mathcal{J}$ is,

$$\mathcal{J} = \frac{1}{N}\int D\mathbf{z} \ln \int d\mu(\mathbf{W}) \exp\left[(\hat{q}_0 - \tfrac{1}{2}\hat{q}_1)\mathbf{W}^T\mathbf{W} + \mathbf{W}^T(\hat{R}\mathbf{W}^0 + \sqrt{\hat{q}_1}\mathbf{z})\right] \, .$$

The notation $D\mathbf{z} = \prod_i Dz_i$ has been used. As in the calculation for the standard weight decay, the penalty term is considered as a prior on the student weight vectors, giving a prior distribution of weights,

$$d\mu(\mathbf{W}) = (2\pi)^{-\frac{N}{2}} |\beta\Lambda|^{\frac{1}{2}} \exp\left(-\frac{\beta}{2}\mathbf{W}^T\Lambda\mathbf{W}\right) \, .$$

This distribution may be substituted into the equation for $\mathcal{J}$ above. The integral over the weight vector $\mathbf{W}$ may now be evaluated using the general formula for

multivariate Gaussian integration [30], giving,

$$\mathcal{J} = \frac{1}{2}\ln|\beta\Lambda| - \frac{1}{2}\ln|C| + \frac{1}{N}\int D\mathbf{z}\exp\left[\frac{1}{2}\mathbf{r}^T C^{-1}\mathbf{r}\right] ,$$

where

$$C = (\hat{q}_1 - 2\hat{q}_0)I + \beta\Lambda , \tag{5.4}$$

and $\mathbf{r} = \hat{R}\mathbf{W}^0 + \sqrt{\hat{q}_1}\mathbf{z}$. The vector $\mathbf{r}$ is identical to the field term in the effective Hamiltonian introduced in §2.10. The integral over the Gaussian random field, $\mathbf{z}$, may be evaluated to give the result,

$$\begin{aligned}\mathcal{G}_0 &= -R\hat{R} - q_0\hat{q}_0 + \frac{1}{2}q_1\hat{q}_1 + \frac{1}{2N}\ln|\beta\Lambda| - \frac{1}{2N}\ln|C| \\ &\quad + \frac{1}{2N}\hat{R}^2\mathbf{W}^{0T}C^{-1}\mathbf{W}^0 + \frac{1}{2N}\hat{q}_1\mathrm{Tr}\ C^{-1}\end{aligned}$$

The free energy for the standard weight decay was dependent on the teacher vector only through its length Ω. This meant that the teachers were implicitly assumed to be drawn from a distribution with fixed length (the micro-canonical ensemble). However, in the general case calculated above the Hamiltonian contains a dependence on the explicit form of the teacher weight vector through the term $\frac{1}{2N}\hat{R}^2\mathbf{W}^{0T}C^{-1}\mathbf{W}^0$, thus it is necessary to select a specific form for the teacher. Since the teacher weight vector is constant throughout the generation of the training data, it may be considered a quenched parameter of the system. Hence, the quenched average could include an average over the teacher distribution, given by the measure, $d\mu(\mathbf{W}^0)$.

5.2 Averaging over the teacher

The dependence of the free energy on the actual teacher weight is removed by assuming a distribution of teacher weight vectors and averaging over this distribution. This distribution of teachers can be thought of as a prior on the teacher weight vectors. There are many possible teacher priors the simplest being the spherical distribution used in the previous chapters.

If the teacher vector is assumed to be drawn from a spherical prior, the replica symmetric Hamiltonian for a linear student is unchanged from eq. (3.8), since this equation only depends on the teacher through the length of the teacher weight vector, Ω. The prior constrained Hamiltonian, however is altered to include the teacher average,

$$\begin{aligned} \mathcal{G}_0 &= -R\hat{R} - q_0\hat{q}_0 + \frac{1}{2}q_1\hat{q}_1 + \frac{1}{2N}\ln|\beta\Lambda| - \frac{1}{2N}\ln|C| \\ &\quad + \frac{1}{2N}\int d\mu(\mathbf{W}^0)\hat{R}^2\mathbf{W}^{0T}C^{-1}\mathbf{W}^0 + \frac{1}{2N}\hat{q}_1\mathrm{Tr}\ C^{-1}\ . \end{aligned} \tag{5.5}$$

It is now necessary to evaluate the integral over the spherical teacher distribution, $d\mu(\mathbf{W}^0) = \delta(\mathbf{W}^0\cdot\mathbf{W}^0 - N\Omega^2)\,d\mathbf{W}^0/a$, where a is a constant such that the integral $\int d\mu(\mathbf{W}^0) = 1$. The average over the teacher distribution is of the form, $\int d\mu(\mathbf{W})\mathbf{W}^T\mathrm{A}\mathbf{W}$ where A is some matrix. Since $\mathbf{W}^T\mathrm{A}\mathbf{W}$ is a scalar, the trace of the average of this quantity is also a scalar. Consider

$$m = \mathrm{Tr}\left(\frac{1}{a}\int d\mathbf{W}\,\delta(\mathbf{W}^2 - N\Omega^2)\mathbf{W}^T\mathrm{A}\mathbf{W}\right)\ .$$

Since matrices commute under the trace, this is equal to the trace of the average

of $\mathbf{W}\mathbf{W}^T$A. Let, $\int d\mu(\mathbf{W})\mathbf{W}\mathbf{W}^T = M$, then M is symmetric and commutes with any rotation matrix. By Schur's lemma [10] M must be a multiple of the identity matrix. The trace of M is given by Tr $(\int d\mu(\mathbf{W})\mathbf{W}^T\mathbf{W}) = N\Omega^2$, therefore, $M = \Omega^2 I$ and $m = \Omega^2$Tr A. Using this result in eq. (5.5) yields,

$$\begin{aligned}\mathcal{G}_0 &= -R\hat{R} - q_0\hat{q}_0 + \frac{1}{2}q_1\hat{q}_1 + \frac{1}{2}\ln\beta + \frac{1}{2N}\ln|\Lambda| - \frac{1}{2N}\ln|C| \\ &\quad + \frac{1}{2N}(\hat{R}^2\Omega^2 + \hat{q}_1)\text{Tr } C^{-1} .\end{aligned}$$

This again reduces to the appropriate result of the previous chapter for a penalty matrix which is a multiple λ of the identity matrix.

5.3 Free energy for general penalty term

The free energy per weight for a general penalty term may be calculated from the replica symmetric Hamiltonian and the prior constrained Hamiltonian as in the previous chapters. From the free energy, the order parameters can be derived.

The replicated Hamiltonian $\mathcal{G}_r$ has already been calculated for a linear student trained on noisy data generated by a spherical teacher and is given in eq. (3.8). Therefore the free energy may be written as,

$$\begin{aligned}-\beta f &= -R\hat{R} - q_0\hat{q}_0 + \frac{1}{2}q_1\hat{q}_1 + \frac{1}{2N}\ln|\beta\Lambda| - \frac{1}{2N}\ln|C| \\ &\quad + \frac{1}{2N}(\hat{R}^2\Omega^2 + \hat{q}_1)\text{Tr } C^{-1} - \frac{\alpha}{2}\ln(1+\beta(q_0-q_1)) \\ &\quad - \frac{\alpha\beta}{2(1+\beta(q_0-q_1))}\left(q_1 - 2\frac{R}{\Omega}\langle xg_0\rangle_\eta + \langle g_0^2\rangle_\eta\right) \qquad (5.6)\end{aligned}$$

Generally the free energy for a teacher drawn from a spherical distribution depends on the penalty matrix through the eigenvalues of the matrix C and hence the eigenvalues of the penalty matrix. Thus any off-diagonal penalty matrix is equivalent to its diagonalised form. This is only true for teacher weight vectors and example inputs drawn from isotropic distributions since the free energy is invariant under a rotation of the weight and a corresponding rotation of input space.

Differentiating yields the saddle point equations for the order parameters. (See appendix E):

$$q_0 = q_1 + \frac{1}{N}\mathrm{Tr}\ C^{-1} \tag{5.7}$$

$$q_1 = (\hat{r}^2 + \hat{q}_1)\frac{1}{N}\mathrm{Tr}\ C^{-2} \tag{5.8}$$

$$r = \hat{r}\frac{1}{N}\mathrm{Tr}\ C^{-1} \tag{5.9}$$

$$\hat{q}_0 = \frac{1}{2}(\hat{q}_1 - \frac{\hat{r}}{\langle xg_0 \rangle_\eta}) \tag{5.10}$$

$$\hat{q}_1 = \frac{\alpha\beta^2}{(1+\beta(q_0-q_1))^2}\left(q_1 - 2r\langle xg_0\rangle_\eta + \left\langle g_0^2 \right\rangle_\eta\right) \tag{5.11}$$

$$\hat{r} = \frac{\alpha\beta}{1+\beta(q_0-q_1)}\langle xg_0\rangle_\eta \tag{5.12}$$

where $r = R/\Omega, \hat{r} = \hat{R}\Omega$. The order parameters are similar to those for the standard weight decay, the dependence on the penalty matrix arises through the inverse of the matrix C.

5.4 Solving the saddle point equations

The saddle point equations eq. (5.7) – (5.12) may be solved to discover the behaviour of the system. As in the previous chapter consider the difference between the overlaps $\mathcal{Q} = q_0 - q_1$. This is given by,

$$\mathcal{Q} = \frac{1}{N}\mathrm{Tr}\; C^{-1} \,. \tag{5.13}$$

The parameter ϕ may be defined as before as, $\phi = 1 + \frac{1}{\mathcal{Q}'}$ This then means that the overlap r can be written as,

$$r = \frac{\alpha}{\phi} \langle\, x g_0 \,\rangle_\eta \;,$$

where the effective gain of the teacher, $\langle\, x g_0 \,\rangle_\eta$, has been used The equations for the remaining order parameters are simplified if the notation

$$\frac{1}{\tilde{\phi}^2} = \frac{\beta^2}{(1+\mathcal{Q}')^2}\frac{1}{N}\mathrm{Tr}\; C^{-2} \,, \tag{5.14}$$

is introduced. Using this equation the average overlap between replicas is,

$$q_1 = \frac{\alpha \langle\, x g_0 \,\rangle_\eta^2}{\tilde{\phi}^2 - \alpha}\left(\alpha - \frac{2\alpha}{\phi} + 1 + \tilde{\gamma}_{\mathrm{eff}}^2\right) \,, \tag{5.15}$$

where the effective noise level, $\tilde{\gamma}_{\mathrm{eff}}$, eq. (3.27) has been used. This result may be substituted into $\hat{q}_1$ to give,

$$\hat{q}_1 = \frac{\alpha\beta^2(\phi-1)^2}{\phi^2(\tilde{\phi}^2 - \alpha)} \langle\, x g_0 \,\rangle_\eta^2 \left(\tilde{\phi}^2(1+\tilde{\gamma}_{\mathrm{eff}}^2) + \alpha^2 - 2\frac{\alpha\tilde{\phi}^2}{\phi}\right) \,. \tag{5.16}$$

The average length of the student, q_0 may be written as $q_1 + T/(\phi - 1)$. These results may be substituted into the equations for the generalisation and training errors to evaluate the performance of the network using different penalty terms.

5.5 Generalisation error

The generalisation error for the general weight decay in terms of the order parameters is given by the same formula as that for the standard weight decay case, eq. (3.33), since its dependence on the different prior weight constraint is only through the order parameters themselves. For completeness, it is included here,

$$\epsilon_g = \frac{1}{2}(q_0 + \left\langle g_0^2 \right\rangle - 2r \left\langle x g_0 \right\rangle) . \tag{5.17}$$

The values of the order parameters obtained from the saddle point equations may be substituted into this equation to give the uncorrupted generalisation error. The corrupted generalisation error, ϵ_g', §3.8 may be written in terms of ϕ and $\tilde{\phi}$ as,

$$\epsilon_g' = \frac{\left\langle x g_0 \right\rangle_\eta^2}{2(\tilde{\phi}^2 - \alpha)} \left(\tilde{\phi}^2(1 + \tilde{\gamma}_{\text{eff}}^2) + \alpha^2 - 2\frac{\alpha\tilde{\phi}^2}{\phi} \right) + \frac{T}{2(\phi - 1)} .$$

The form of the uncorrupted generalisation error ϵ_g is more complicated. Again the corrupted generalisation error for a nonlinear teacher is equivalent to a linear teacher with a different noise level and effective gain on the linear teacher. The generalisation error has a temperature dependent part that only depends on the hyperparameters through the function ϕ, which is related to the response

function. The temperature dependent part of the generalisation error is independent of the noise level on the data or the distribution of teachers. This makes sense since the temperature arises as part of the training algorithm and is not correlated to the training data.

5.6 Training error

The average training error is calculated by differentiating the free energy divided by the temperature with respect to the inverse temperature, β whilst keeping the penalty term, $\beta\Lambda$ constant. This leads to the formula

$$\epsilon_t = \frac{(\phi-1)^2 \langle xg_0 \rangle_\eta^2}{2\phi^2(\tilde{\phi}^2-\alpha)} \left(\tilde{\phi}^2(1+\tilde{\gamma}_{\rm eff}^2) + \alpha^2 - 2\frac{\alpha\tilde{\phi}^2}{\phi} \right) + \frac{T}{2\phi} . \qquad (5.18)$$

The first (temperature independent) term is given by $\hat{q}_1/2\alpha\beta^2$ and is changed from the standard weight decay case by the introduction of the function $\tilde{\phi}$. The second term gives the temperature dependence of the training error. As in the standard weight decay case, the zero temperature training error is related to the zero temperature corrupted generalisation error by, $\epsilon_t = \epsilon_g'/(1+\mathcal{Q}')^2$. The factor is dependent on the response function which depends in turn on the penalty matrix used.

The average cost function may also be calculated and is given by,

$$\epsilon_c = \epsilon_t + \frac{1}{N}\mathrm{Tr}\ \left(\Lambda(C^{-1} + (\hat{r}^2 + \hat{q}_1)C^{-2}) \right) .$$

The term added to the training energy is equivalent to the average value of the

penalty term that appeared in the average cost function for the standard weight decay eq. (3.38).

5.7 Diagonal penalty matrix

The simplest form of the penalty matrix is the standard weight decay, *i.e.*, $\Lambda = \lambda I$. The generalisation of this case is to consider a general diagonal matrix Λ given by,

$$\Lambda_{ij} = \lambda_i \delta_{ij}$$

where Λ_{ij} are the components of the penalty matrix and the λ_i are the individual weight decays on the components of the student weight vector. Since the free energy eq. (5.6) is invariant under a transformation to the eigenbasis of Λ, the case of diagonal penalty matrix is general.

The matrix C may now be written as

$$C_{ij} = \{\beta\lambda_i + \hat{q}_1 - 2\hat{q}_0\}\delta_{ij}$$

The saddle point equations depend only on the trace of the inverse of C. Since the matrix is diagonal, the inverse is simple to calculate and its trace is given by,

$$\text{Tr } C^{-1} = \sum_i \frac{1}{\beta\lambda_i + \hat{q}_1 - 2\hat{q}_0} \tag{5.19}$$

The difference between the intra-replica overlap, q_0 and the inter-replica overlap, q_1, $\mathcal{Q}$ is given by $\mathcal{Q} = \frac{1}{N}\text{Tr } C^{-1}$. The average overlaps are given in terms of $\tilde{\phi}$

which is in terms of the trace of the inverse squared, hence Tr $C^{-2} = \sum_i (\beta\lambda_i + \hat{q}_1 - 2\hat{q}_0)^{-2}$. If $\lambda_i = \lambda \ \forall i$ this simply reduces to the standard weight decay, where $\tilde{\phi} = \phi$. The response function for the standard weight decay is given by the solution to a quadratic equation. For m independent weight decay parameters in the diagonal penalty matrix the response function is the solution to a polynomial of order $m + 1$.

5.7.1 Two weight decay elements - Linear teacher

In this case, assume that the weights are ordered such that the weight decay on the first kN components is λ_1 and the weight decay on the remaining $(1-k)N$ components is λ_2 $(0 \leq k \geq 1)$. That is

$$\lambda_i = \begin{cases} \lambda_1 & \text{for } i \leq kN \\ \lambda_2 & \text{for } i > kN \end{cases} \tag{5.20}$$

Substituting this assumption into eq. (5.13) and using eq. (5.19) and (5.12) gives,

$$\mathcal{Q} = \frac{k}{\beta\lambda_1 + \frac{\alpha\beta}{1+\beta\mathcal{Q}}} + \frac{1-k}{\beta\lambda_2 + \frac{\alpha\beta}{1+\beta\mathcal{Q}}}$$

After defining the response function $\mathcal{Q}' = \beta\mathcal{Q}$ as before, this equation may be rearranged to give the cubic equation

$$\begin{aligned} 0 \ = \ & \lambda_1\lambda_2\mathcal{Q}'^3 + (\lambda_2(\alpha - k) + \lambda_1(\alpha - (1-k)) + 2\lambda_1\lambda_2)\mathcal{Q}'^2 \\ & + (\lambda_2(\alpha - 2k) + \lambda_1(\alpha - 2(1-k)) + \lambda_1\lambda_2 + \alpha^2 - \alpha)\mathcal{Q}' \\ & - \alpha - k\lambda_2 - \lambda_1(1-k) \end{aligned} \tag{5.21}$$

For the case, $\lambda_1 = \lambda_2 = \lambda$, the cubic equation factors into a linear term and a quadratic term that gives the response function for the standard weight decay. The limits, $k = 0$ and $k = 1$ also reduce to the standard weight decay.

From eq. (5.21) above, the coefficients of the general cubic, $x^3 + a_2\, x^2 a_1\, x + a_0$, are

$$\begin{aligned} a_0 &= -\frac{\alpha}{\lambda_1\lambda_2} - \frac{k}{\lambda_1} - \frac{1-k}{\lambda_2} \\ a_1 &= 1 + \frac{1}{\lambda_1}(\alpha - 2k) + \frac{1}{\lambda_2}(\alpha - 2(1-k)) + \frac{\alpha}{\lambda_1\lambda_2}(\alpha - 1) \\ a_2 &= 2 + \frac{1}{\lambda_1}(\alpha - k) + \frac{1}{\lambda_2}(\alpha - (1-k)) \end{aligned}$$

Appendix F gives the conditions that a positive cubic, $x^3 + a_2\, x^2 + a_1\, x + a_0$, has only one positive real root. These conditions may be written as:

$$\begin{aligned} \text{either} \quad & a_0 < 0 \text{ and } a_2 \geq 0 \\ \text{or} \quad & a_0 < 0 \text{ and } a_2 < 0 \text{ and } a_1 < 0\,. \end{aligned}$$

For the integration over student weight space to be defined, Λ is assumed to be positive definite. This means $\lambda_i > 0$ for all i. Thus since, $\lambda_1, \lambda_2, \alpha > 0$ and $0 \leq k \leq 1$, $a_0 < 0$ for all α, λ and k. Now, if $a_2 \geq 0$, then there is only one positive root of the cubic equation. However, if $a_2 < 0$, this implies that $\alpha \leq 1$, since if $\alpha > 1$, then neither $\alpha - k$ or $\alpha - 1 + k$ could be negative and hence a_2 would be greater than zero. Now $a_1 = a_2 - 1 - \frac{k}{\lambda_1} - \frac{(1-k)}{\lambda_2} - \frac{\alpha(1-\alpha)}{\lambda_1\lambda_2}$. Hence for $\alpha < 1$, and $a_2 < 0$, $a_1 < 0$. Thus there is only ever one positive real root of the cubic equation.

Appendix F gives the roots of a cubic in terms of two parameters $p_1 = a_2^2/9 - a_1/3$ and $p_2 = a_1 a_2/6 - a_0/2 - a_2^3/27$. The positive root is then given by

$$\mathcal{Q}' = 2\sqrt{p_1}\cos\frac{\vartheta}{3} - \frac{a_2}{3} \tag{5.22}$$

where $\vartheta = \cos^{-1}(p_2/\sqrt{p_1^3})$

To evaluate the other saddle point equations, it is necessary to evaluate Tr C^{-2} and hence $\tilde{\phi}^2$.

$$\frac{1}{N}\text{Tr } C^{-2} = \frac{k}{(\beta\lambda_1 + \hat{q}_1 - 2\hat{q}_0)^2} + \frac{(1-k)}{(\beta\lambda_2 + \hat{q}_1 - 2\hat{q}_0)^2}$$

Substituting this into the definition of $\tilde{\phi}$ and using $1 + \mathcal{Q}' = \frac{\phi}{(\phi-1)}$ gives,

$$\frac{1}{\tilde{\phi}^2} = \frac{k(\phi-1)^2}{(\lambda_1\phi + \alpha(\phi-1))^2} + \frac{(1-k)(\phi-1)^2}{(\lambda_2\phi + \alpha(\phi-1))^2}$$

The order parameters may now be evaluated in terms of ϕ and $\tilde{\phi}$ using eq. (5.14) - (5.16). The parameter ϕ is evaluated from the response function given by eq. (5.22). The behaviour of the network may be investigated for some limiting cases of the network parameters.

Zero T, zero λ_1 limit

The limit as one of the weight decay parameters is reduced to zero may be investigated. It makes no difference which parameter is chosen since the equations are symmetric under the transformation $\lambda_1 \to \lambda_2$ and $k \to (1-k)$, hence the limit,

$\lambda_1 \to 0$ will be investigated with $\lambda_1 \ll \lambda_2$. As for the standard weight decay, the limit $\lambda_1 \to 0$ is taken at the same time as taking the zero temperature limit and defining $\lambda' = \beta\lambda_1$ as constant.

As $\lambda_1 \to 0$, $\vartheta = 0$ for $\alpha < k$ and $\vartheta = \pi$ for $\alpha > k$. The solution of the cubic equation for the response function, $\mathcal{Q}'$, is given by,

$$\mathcal{Q}' = \begin{cases} \frac{1}{\lambda_1}(k - \alpha) + O(\lambda_1^0) & \alpha < k \\ O(\lambda_1^0) & \alpha > k \end{cases}$$

To discover the behaviour of the $\alpha > k$ root, it is necessary to return to the complete cubic equation. Since the solution is of $O(\lambda_1^0)$ the terms in λ_1 may be ignored, which leaves a quadratic equation with positive root,

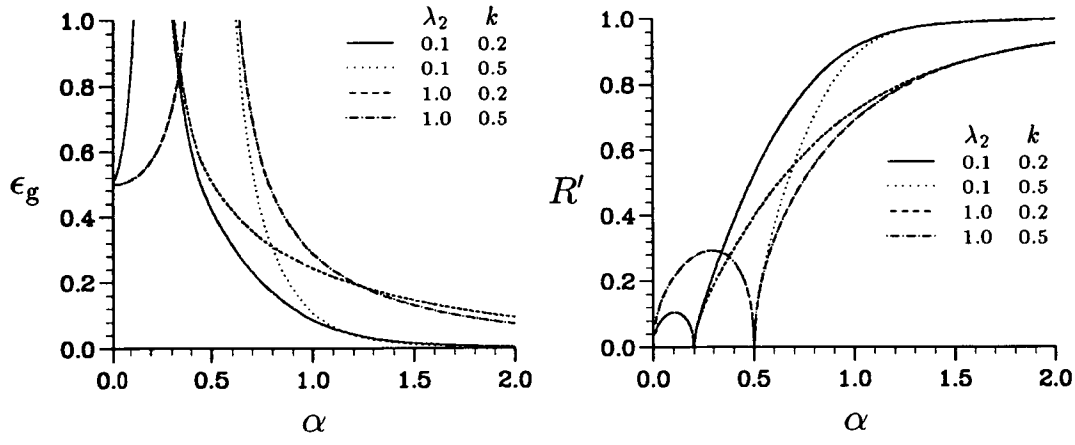
$$\mathcal{Q}'_{\alpha>k} = \frac{\alpha - \alpha^2 + 2k\lambda_2 - \alpha\lambda_2 + \alpha\sqrt{\psi}}{2\lambda_2(\alpha - k)}$$

where $\psi = (1 + \alpha + \lambda_2)^2 - 4(\alpha + k\lambda_2)$. Hence for zero T, λ_1, $\mathcal{Q} = \mathcal{Q}'/\beta$ may be written.

$$\mathcal{Q} = \begin{cases} \frac{1}{\lambda'}(k - \alpha) & \alpha < k \\ O & \alpha > k \end{cases}$$

The parameters ϕ and $\tilde{\phi}$ may now be evaluated and substituted into the equations for the order parameters, hence the performance of the network may be calculated. From the response function it can be seen that the behaviour for $\alpha \leq k$ is independent of the second weight decay, λ_2. As $\alpha \to k$, it can be postulated that the examples fix Nk components of the student weight and the nonzero weight decay parameters provide constraints on the remaining $(1 - k)N$ components. In the presence of noise, it would be expected that the generalisation error would

have a similar discontinuity at $\alpha = k$ as was seen with the standard weight decay. However for zero noise it could be naïvely expected that the generalisation error will decrease smoothly and approach zero as $\alpha \to k$. The performance measures, ϵ_g, R', are plotted in Fig. 5.1 for zero noise and the $\lambda_1 \to 0$ limit.



(a) Generalisation error (b) Overlap

Figure 5.1. Generalisation error and R' for zero temperature and noise in the $\lambda_1 \to 0$ limit.

Surprisingly, the curves in Fig. 5.1 show a discontinuity present in the performance measures for zero noise. When α approaches k, the behaviour of the generalisation error for zero noise is,

$$\epsilon_g \sim \frac{k(1-k)}{2|\epsilon|} + O(\epsilon^0)$$

where $\epsilon = \alpha - k$. This result may be compared to the behaviour of the zero weight decay limit in the presence of noise studied in the previous chapter, eq. (4.4). If the number of examples is rescaled so that the discontinuity is at $\alpha' = 1$, the behaviour is equivalent to the pseudo inverse solution with noise of variance

$(1-k)$ added to the training data. The fact that this behaviour is independent of the non-zero weight decay parameter is because the λ_2 is assumed to be >> than λ_1. If λ_2 is of the same order as λ_1, the divergence becomes a peak that is reduced as $\lambda_1, \lambda_2 \rightarrow 0$.

The overlap between student and teacher, r, is simply α for $\alpha < k$, for $\alpha > k$ the solution is more complicated. The divergence in the generalisation error and the reduction of the cosine of the angle between student and teacher is due to the length of the student growing to infinity. This in turn is due to zero eigenvalues in the pattern correlation matrix (see §3.1) and the zero weight decay components. The difference between the weight decays is acting like noise added to the data set.

The generalisation error as $\alpha \rightarrow \infty$ for zero noise is given by $\epsilon_g = (1-k)\lambda_2^2/\alpha^2 + O(\alpha^{-3})$. Thus asymptotically, the behaviour depends on the values of both k and the non-zero weight decay, λ_2. Lower generalisation error is achieved with larger values of k, since this corresponds to training with a larger proportion of the weight decays set to zero. The asymptotic generalisation error may be compared to that for the standard weight decay under similar conditions, *i.e.*, $\epsilon_g = \lambda^2/\alpha^2 + O(\alpha^{-3})$; it can be seen that the effect of the second weight decay parameter is equivalent to a scaled standard weight decay parameter as α becomes large. Thus the more patterns that are presented the less the effect of the zero weight decay on the kN components.

The average training energy for the case $\lambda_1 \rightarrow 0$ is plotted in Fig. 5.2. Here the effect of the non-zero weight decay acting as noise is clearly seen. For $\alpha < k$, the training energy is zero as would be expected, since the model has enough degrees of freedom to store the training examples exactly. However as the number of

patterns per weight increases above k, the training energy rapidly increases as the non-zero weight decay acts as noise on the examples. As the number of examples is increased, the effect of the weight decay noise is reduced and the training error tends to the standard asymptotic value, $\tilde{\gamma}^2/2$.

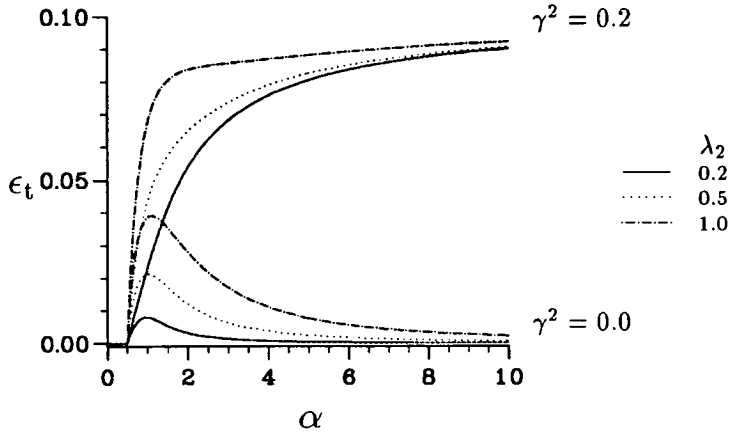
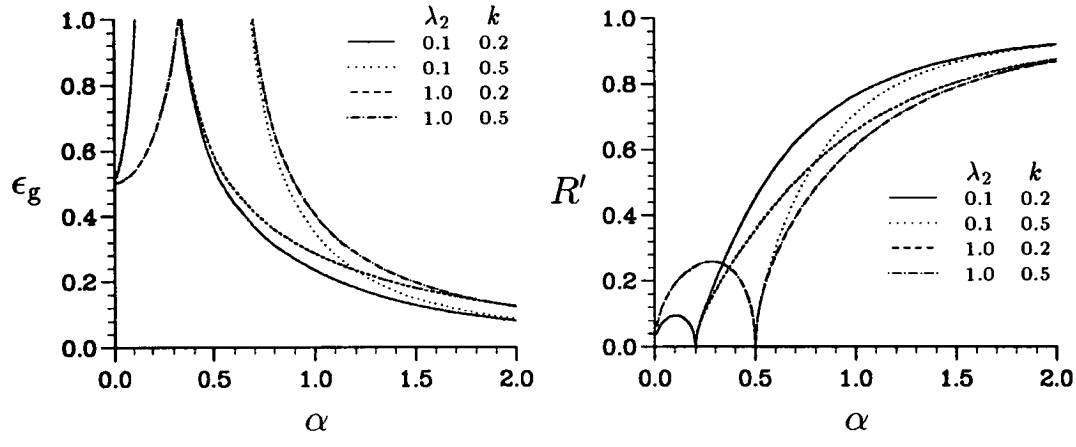


Figure 5.2. Training error $\lambda_1 \to 0$ $k = 0.5$, $\lambda_2 = 0.2, 0.5, 1.0$. The upper three curves are for a noise level of $\tilde{\gamma}^2 = 0.2$ and the lower curves are zero noise added.

The performance measures plotted in Fig. 5.1 may also be calculated for non-zero noise. This gives the curves seen in Fig. 5.3. These curves are similar to those for zero noise. This is a difference between this limit and that seen in the zero weight decay limit for the standard weight decay. The discontinuities at $\alpha = k$ are still present and the effect of the noise is to broaden the discontinuity peak. Around $\alpha = k$, the generalisation can be expanded, $\epsilon_g \sim k(\tilde{\gamma}^2 + 1 - k)/(2|\epsilon|)$, where $\epsilon = \alpha - k$. This is equivalent to a rescaled number of examples having a discontinuity at $\alpha' = 1$ due to noise of variance $\tilde{\gamma}^2 + 1 - k$.



(a) Generalisation error (b) Overlap

Figure 5.3. Generalisation error and R' for zero temperature in the $\lambda_1 \to 0$ limit. The noise is $\tilde{\gamma}^2 = 0.2$. The curves correspond to different values of k and λ_2.

Finite λ_1, λ_2

In this case both weight decay parameters are given finite values. The performance measures may be evaluated in terms of ϕ and $\tilde{\phi}$ numerically; the results are shown for various values of the weight decay parameters.

In the limit of no noise added to the linear teacher, optimum generalisation error is achieved in the zero weight decay limit, thus it is expected that lower generalisation error will be achieved with smaller weight decays. The generalisation error for a number of different combinations of the two weight decay parameters is presented in Fig. 5.4. The graph presents the average generalisation error for $\lambda_1 = 1.0$ and $\lambda_2 = 0.1$ with different values of k. The minimum value of the generalisation error for these values of λ_1 and λ_2 is achieved when $k = 0$, this is as expected since this is the smallest weight decay possible given the two fixed

values of the diagonal elements of the penalty matrix. Again the effect of the interaction between the different decay parameters is to act like noise for smaller values of α. This means that networks with a lower linear average weight decay $(k\lambda_1 + (1-k)\lambda_2)$ can have a larger generalisation error than networks using a larger standard weight decay for smaller values of α.

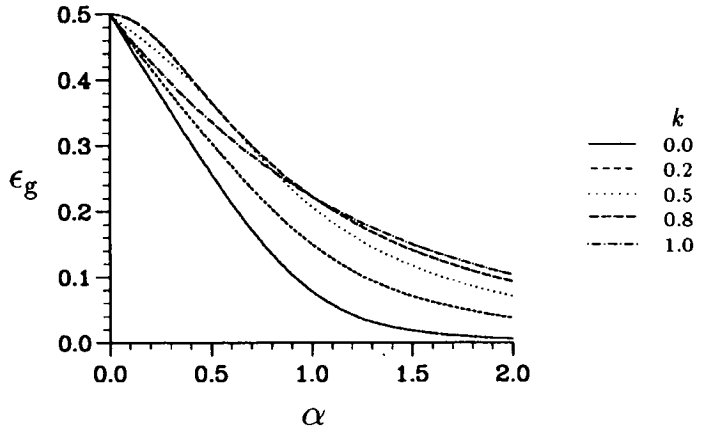
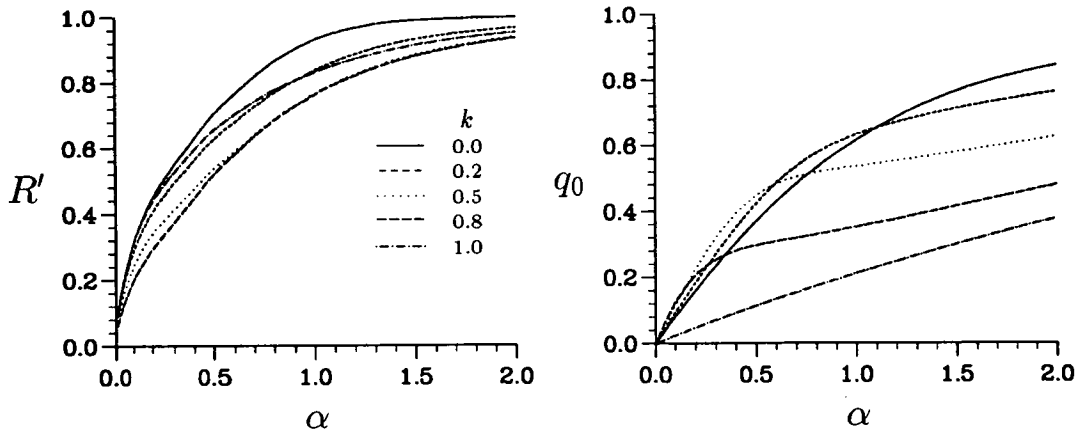


Figure 5.4. Generalisation error for zero temperature and noise. The curves are for $\lambda_1 = 1.0$ and $\lambda_2 = 0.1$.

In Fig. 5.5(a), the average overlap between student and teacher is presented for the same parameter values as those used for the generalisation error in Fig. 5.4. Again as expected, the smallest weight decay gives the maximum overlap between student and teacher. However for small α, the next largest overlap is given by $k = 1$, this does not give a correspondingly good generalisation error, since the large weight decay produces a relatively small length of student weight and this adversely affects the generalisation error. The graph in Fig. 5.5(b) presents the average length squared of a student. Ideally if the student exactly modeled the teacher this would be one. The presence of the weight decay causes the length to be reduced from this value. For $\alpha < 1 - k$ the student has enough degrees of

freedom so that it is reduced by an amount that appears to be proportional to the smaller weight decay. As $\alpha \sim 1 - k$, the length of the student grows more slowly as the larger weight decay becomes used.

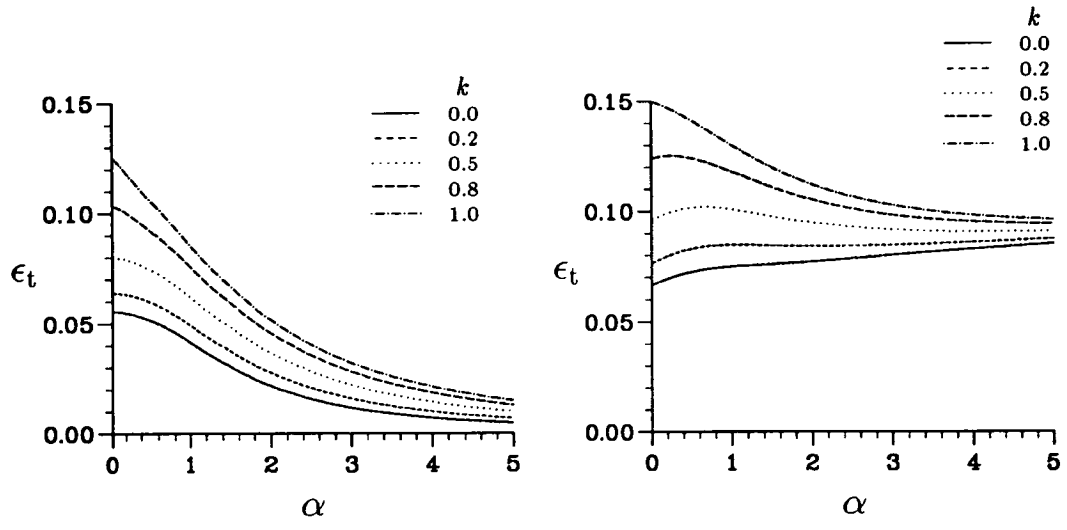


(a) Teacher overlap (b) Length squared of student

Figure 5.5. Average overlap between student and teacher and average length squared of a student for zero temperature and noise and fixed $\lambda_1 = 1.0$ and $\lambda_2 = 0.1$. The legend is the same for both graphs.

The average training error for finite λ_1, λ_2 and zero temperature is presented in Fig. 5.6. The average training error is directly related to the weighted linear average weight decay, $k\lambda_1 + (1 - k)\lambda_2$ (unlike the generalisation error). The asymptotic values of the training error are as for the standard weight decay.

Static noise is added to the training data and the effect on the generalisation error is plotted in Fig. 5.7 for a noise level of 0.2. The best generalisation error for these values of the weight decay parameters is achieved by the lowest weight decay, $k = 0$ and $\lambda = 0.1$. For small numbers of patterns, $\alpha <\sim 1$, the difference between the weight decays acts as noise causing the generalisation error to increase above that for the standard weight decay. As α increases above ~ 1, the generalisation error

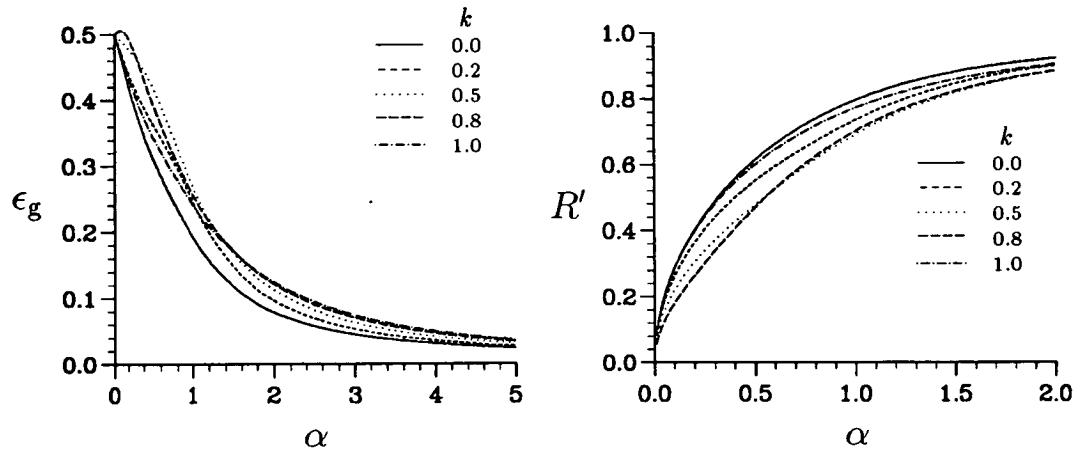


(a) Zero noise (b) $\tilde{\gamma}^2 = 0.2$

Figure 5.6. Training error for zero temperature, two weight decays, $\lambda_1 = 1.0, \lambda_2 = 0.1$.

using the larger standard weight decay becomes worse than using the mixture of weight decays. Here there is enough data to prevent the student from learning the noise on that data as well as that introduced by the difference between the two weight decays.

The plot of the average overlap Fig. 5.7(b) has a different ordering of the curves for the same values of λ_1, λ_2. The overlap between student and teacher is similar for both cases with a single weight decay (*i.e.*, $k = 0, 1$) until $\alpha \rightarrow 1$. The seeming disagreement between the generalisation error and overlap can be explained by looking at the average length of the student. For the larger weight decay (in this case), the student is shorter than is necessary and thus the generalisation error is increased. It could be that for larger values of noise, the smaller weight decay would give a larger generalisation error and the larger weight decay could give the minimum generalisation error. This is shown in Fig. 5.8, which presents

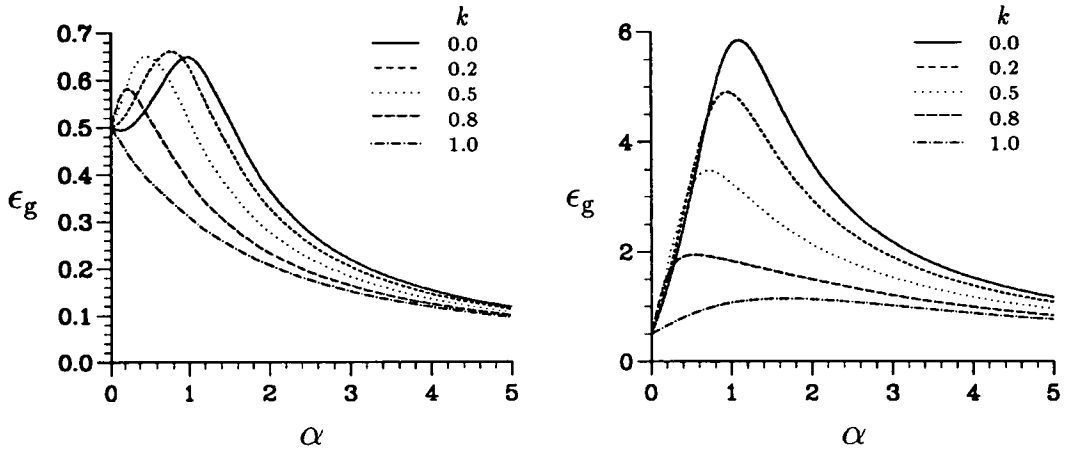


(a) Generalisation error (b) Overlap

Figure 5.7. Generalisation error and overlap for $\lambda_1 = 1.0, \lambda_2 = 0.1$, noise of variance $\tilde{\gamma}^2 = 0.2$ was added.

the generalisation error for two weight decays, $\lambda_1 = 1.0, \lambda_2 = 0.1$ and noise of variance $\tilde{\gamma}^2 = 1.0, 10.0$ added. This difference is due to the fact that for the smaller weight decays, the network sets an average weight vector that is longer than the teacher and this increases the generalisation error. The larger weight decays are needed to counteract the effect of the larger noise.

A peak in the generalisation error that was observed for the standard weight decay, Fig. 4.5, can be seen in Fig. 5.8. The maximum is due to the network learning the noise For the standard weight decay case, $k = 0$ *i.e.*, $\lambda = 0.1$, the peak is around $\alpha = 1 + \lambda$ for both $\tilde{\gamma}^2 = 1.0$ and $\tilde{\gamma}^2 = 10.0$. As the mixing between the two weight decays is increased, the location of the maximum shifts to lower values of α as k increases. For the smaller noise level, $\tilde{\gamma}^2 = 1.0$, at $k = 1$, the weight decay parameter is optimal and there is no maximum in the generalisation error. For the larger noise level, the maximum is again at $\alpha \sim 1+\lambda$.



(a) Noise $\tilde{\gamma}^2 = 1.0$ (b) Noise $\tilde{\gamma}^2 = 10.0$

Figure 5.8. Generalisation error for zero temperature and different weight decays, $\lambda_1 = 1.0, \lambda_2 = 0.1$ and noise $\tilde{\gamma}^2 = 1.0, 10.0$.

The height of the peak is reduced as the weighted linear average of the weight decay is increased.

The effect of having large noise and two weight decays which bracket the noise may also be investigated. This is presented in Fig. 5.9, where the generalisation error for $\lambda_1 = 5.0, \lambda_2 = 0.1$ and $\tilde{\gamma}^2 = 1.0$ is plotted. The peak in the generalisation error is present as before for the smaller weight decays with the mixing shifting the peak to the right as k increases. The larger weight decay gives better generalisation for $\alpha < 2.5$, here the curves cross and the smaller weight decay gives improved generalisation.

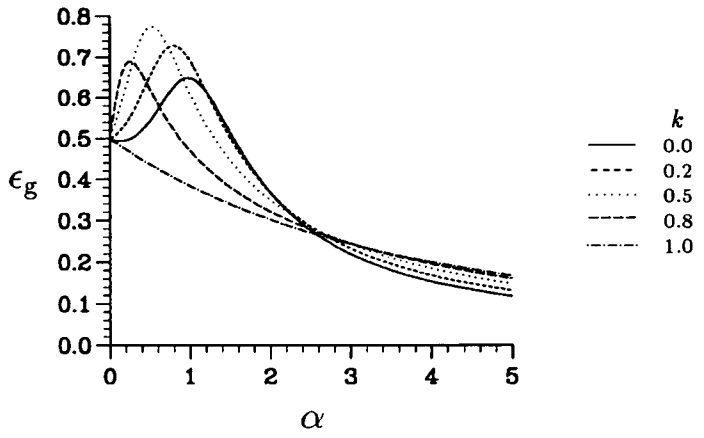


Figure 5.9. Generalisation error for zero temperature and two component penalty matrix, $\lambda_1 = 5.0, \lambda_2 = 0.1$ and noise $\tilde{\gamma}^2 = 1.0$.

5.7.2 Two weight decay elements - Nonlinear teacher

As in the previous chapter, the performance may be evaluated for a linear student learning a nonlinear teacher. Since all that is required is the numerical evaluation of the functions $\langle xg_0 \rangle_\eta, \langle g_0^2 \rangle_\eta$ eq. (3.25) and (3.26) and their uncorrupted analogues.

Zero T, zero λ_1

The curves for zero λ_1 and small gain look very similar to those produced for the linear teacher, apart from a difference in scaling. However for a larger gain of the teacher function, $\tilde{\Omega} = 10.0$, the generalisation error for a teacher with a hyperbolic tangent activation function is as in Fig. 5.10. The discontinuity at $\alpha = k$ is still present, however, there is also a plateau appearing around $\alpha = 1$

for the smaller value of the weight decay parameter. In this case, it appears that the nonlinearity is causing the performance of the network to degrade when the non-zero part of the weight decay is not large enough to cope with the nonlinear part of the teacher.

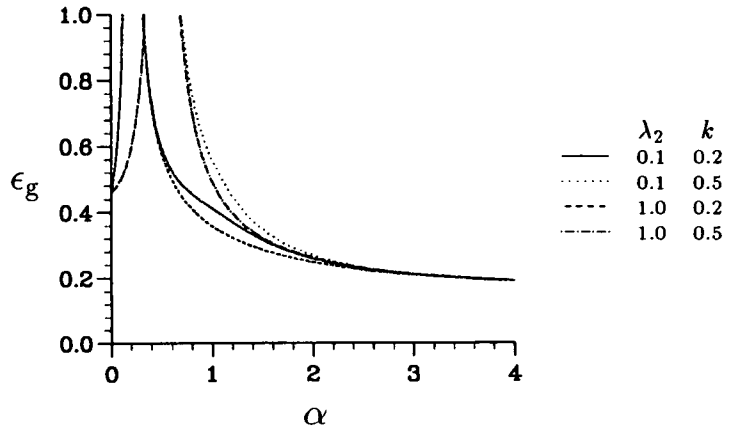


Figure 5.10. Generalisation error for zero temperature and noise. The curves are for a linear student learning a tanh(.) teacher gain $\tilde{\Omega} = 10.0$.

The generalisation error for a student learning a tanh(.) teacher in the presence of noise is presented in Fig. 5.11. This figure shows a similar form to the high gain example in the previous figure, An optimal weight decay parameter can be identified for $\alpha > k$, since the generalisation is better for the weight decay parameter that matches the variance of the noise.

Finite λ_1, λ_2

When both weight decay parameters are finite and the teacher has a nonlinear activation function, the performance of the network may also be calculated. The

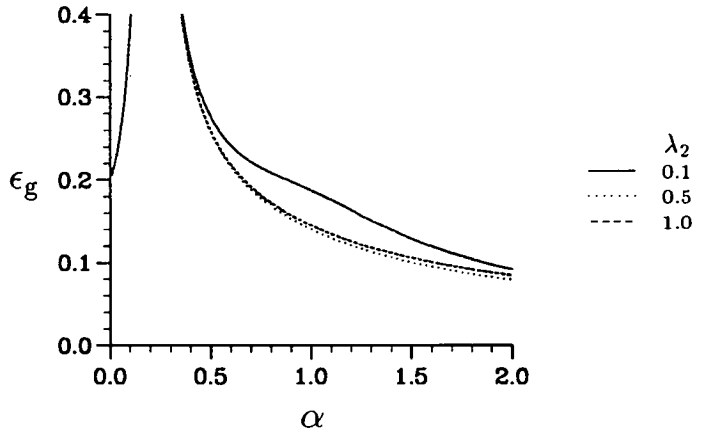
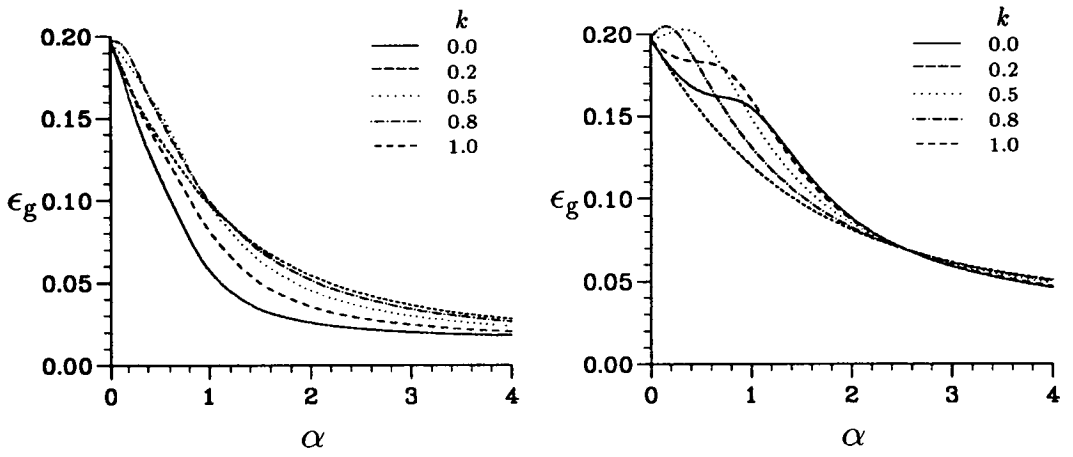


Figure 5.11. Generalisation error for zero temperature. Noise variance, $\tilde{\gamma}^2 = 0.5$ added. The curves are for a linear student learning a tanh(.) teacher, gain $\tilde{\Omega} = 1.0$.

graphs in Fig. 5.12 show the generalisation error for two weight decays, $\lambda_1 = 1.0, \lambda_2 = 0.1$. and two different noise levels. The curves correspond to different values of the parameter k. The curve for zero noise is similar to that for a linear teacher; the best generalisation error is given by the minimum average weight decay, that is, $k = 0(\lambda = 0.1)$. In the case of the noisy teacher, the best generalisation is initially achieved with the higher weight decay on all the components of the weight vector. Above $\alpha \sim 2$, there is a transition where the generalisation is improved with the smaller weight decay. This can be explained as follows; initially, with a small number of patterns, the high weight decay is needed to deal with the nonlinearity of the teacher and the noise on the data. However as the numbers of examples increases, the need for the weight decay is diminished and thus there comes a point where the smaller weight decay is all that is needed. If a close to optimal standard weight decay is used, the generalisation is better than the combination of the weight decays. The average training error

for a nonlinear teacher is of the same form as that for the linear teacher plotted in the previous section.



(a) Zero noise (b) $\tilde{\gamma}^2 = 0.5$

Figure 5.12. Generalisation error for $\lambda_1 = 1.0, \lambda_2 = 0.1$.

The main result of this section is that having two different eigenvalues of the penalty matrix acts like effective noise on the data for a spherical teacher and better performance is achieved with the standard weight decay. This is because the model has not included any fine structure of the teacher. The teacher was assumed to have been drawn from a simple spherical distribution, and thus the optimum student prior is to select a distribution that matches the teacher distribution. In the thermodynamic limit, a standard weight decay is similar to a spherical prior on the student and hence models the spherical teacher distribution.

It might be expected that if the teacher were drawn from a more complicated distribution, a general penalty term would improve the performance, this is studied in the next chapter.

5.8 Off-diagonal penalty matrix

Whilst the diagonal penalty matrix is general, more insight into the action of other penalty matrices can be obtained by considering an explicit off-diagonal form. The off-diagonal terms could be used to introduce correlations between the components of the student weight vector. One of the simplest off-diagonal penalty matrices is one with a constant parameter on the diagonal and another constant in all off-diagonal places, that is,

$$\Lambda = \begin{pmatrix} \lambda_1 & \lambda_2 & \dots & \lambda_2 \\ \lambda_2 & \lambda_1 & \dots & \lambda_2 \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_2 & \lambda_2 & \dots & \lambda_1 \end{pmatrix} \tag{5.23}$$

This penalty matrix then gives a weight decay term that looks like (using eq. (5.2)),

$$W_i^{\text{new}} = (1 - \tau(\lambda_1 - \lambda_2))W_i^{\text{old}} - \tau\lambda_2 \sum_j W_j^{\text{old}} ,$$

This is equivalent to a standard weight decay term of magnitude $\lambda_1 - \lambda_2$ coupled with an additional term that reduces the size of the component by an amount that is proportional to the average size of a component of the weight vector.

This could be used to implement a sort of soft weight sharing [56], where rather than prune the network by decaying some of the weights to zero, the complexity of the network is reduced by combining some of the weights, thus reducing the number of free parameters. For a particular component, W_i, the stable fixed

point of eq. (5.8) is given by,

$$W_i = -\frac{\lambda_2}{\lambda_1 - \lambda_2} \sum_j W_j \, .$$

Hence if $\lambda_2 = -\frac{\lambda_1}{N-1}$ all components are equal to the average value, $W_i = \langle W \rangle$ where $\langle W \rangle = \frac{1}{N} \sum_j W_j$. In the thermodynamic limit, a general penalty matrix that contains this form of weight decay can be obtained with λ_2 of order $O(1/N)$. The off-diagonal penalty terms will be rescaled so that $\lambda_2 = \lambda_2/N$ in eq. (5.8); the penalty term that decays the components towards the average is then achieved by setting $\lambda_2 = -\lambda_1$.

With the off diagonal terms set to the appropriate value in the absence of any information from the training data, the components of the weight vector will all tend to their average value. In the general case of this penalty term (λ_1 on-diagonal, λ_2/N off-diagonal), the eigenvalues of the matrix C eq. (5.4) are then, $\beta(\lambda_1 - \lambda_2/N) + \hat{q}_1 - 2\hat{q}_0$, $N-1$ times and $\beta(\lambda_1 + \lambda_2) + \hat{q}_1 - 2\hat{q}_0$ once. With the spherical teacher prior this is equivalent to a diagonal penalty matrix with $\lambda_1' = \lambda_1 - \lambda_2/N$, $\lambda_2' = \lambda_1 + \lambda_2$ and $k = 1/N$ where the prime, $'$, refers to the diagonal weight decay elements; this gives a $1/N$ correction term to the standard weight decay behaviour. However, since the free energy is only exact in the thermodynamic limit, and the corrections to the free energy are of order N^{-1}, it would be necessary to calculate these corrections before being able to observe a difference between the standard weight decay behaviour and the decay term introduced above using a spherical teacher prior. It is expected that using a different teacher prior may enable the student to use the information contained in the non-standard penalty terms, this is studied in the next chapter.

5.9 Concluding remarks

In this chapter, the performance for a linear student network regularised by a general quadratic penalty term and trained on an arbitrary teacher is calculated. The performance measures are evaluated analytically for explicit forms of the penalty matrix.

The performance of the system is invariant under a rotation of weight vector space and a corresponding rotation in input vector space. This means that the behaviour of the network only depends on the eigenvalues of the penalty matrix, hence a diagonal penalty matrix is considered without loss of generality. The simplest generalisation of the simple weight decay is considered, this is a diagonal penalty matrix with two different weight decays on the diagonal. In the case of one weight decay tending to zero, there is a discontinuity in the performance measures as the number of patterns per weight tends towards the fraction of weight decays that were zero. This can be understood in terms of the non-zero weight decay introducing an effective noise on the data. With both weight decays non-zero, again effective noise is introduced due to the difference between the weight decays. A standard weight decay performs better than the more complicated weight decay.

A penalty term that decayed the components of the student weight towards their average value is also considered. This penalty matrix introduces $1/N$ corrections to the standard weight decay free energy. It is not possible to calculate these corrections using the replica method, though other methods may yield the finite size results [67]. This penalty term will be considered in the next chapter for anisotropic teacher priors.

The behaviour of the system in this chapter shows that the student prior should describe any knowledge about the teacher as accurately as possible. If the student prior is too tightly constrained, this can adversely affect the performance. For a student learning a spherical teacher, the optimal penalty matrix is the standard weight decay, this can be understood as the student prior that matches the teacher prior. It is expected that the contribution of a generalised penalty term will be more significant when more detailed knowledge about the structure of the teacher is included. This will be studied in the next chapter.

Chapter 6

Extensions to the model

The previous chapter has shown that a general penalty term when applied to the problem of learning a teacher chosen from a spherical distribution doesn't necessarily improve the performance of the network and in some cases it can degrade the performance. However, it can be postulated that if more information about the teacher were included into the model, then non-standard penalty terms would improve performance. There may be other information that could be included in the model through the distribution of inputs or the noise model. In this chapter the order parameters for the more general case where the teacher and input vectors are drawn from a general Gaussian distribution and the input distribution is anisotropic are calculated. The performance of the network in some limits of these distributions is then considered. Finally, different noise models are discussed in terms of the different input and teacher distributions. It will be shown that in some cases a particular penalty matrix improves the performance from the standard weight decay case.

6.1 General teacher and input distributions

The model is now extended to allow for the possibility that the teacher and/or the data has been drawn from an anisotropic Gaussian distribution.

6.1.1 General teacher prior

In the previous calculations, the teacher was assumed (either implicitly or explicitly) to be drawn from a spherical distribution. In a real world problem it is possible that some of the teacher components are correlated. To model these correlations, the teacher can be assumed to be drawn from a general Gaussian prior. The measure on the teacher weight vector space is,

$$d\mu(\mathbf{W}^0) = (2\pi)^{-\frac{N}{2}} |\Sigma_0|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\mathbf{W}^{0T}\Sigma_0^{-1}\mathbf{W}^0\right] d\mathbf{W}^0, \tag{6.1}$$

where Σ_0 is the covariance matrix of the prior teacher distribution. The free energy per weight is averaged over this distribution to remove the explicit dependence on the teacher weight vector.

6.1.2 General input distribution

Previously it was assumed that the components of the input patterns were independently and identically distributed with unit variance (isotropic). A non-unit variance for the components of the input patterns may be absorbed by a renormalisation of both the student and teacher weights and the weight decay parameters.

However, if there are correlations between the components of the input patterns, there may be no simple renormalisation of the weights and the network parameters that removes the correlations without altering the free energy. This is the condition considered in this section. The measure on input space is written as,

$$d\mu(\mathbf{s}) = \frac{d\mathbf{s}}{(2\pi)^{\frac{N}{2}}|\Sigma_s|^{\frac{1}{2}}} \exp\left[-\tfrac{1}{2}\mathbf{s}^T{\Sigma_s}^{-1}\mathbf{s}\right] , \tag{6.2}$$

where Σ_s is the covariance of the input distribution.

6.2 Free energy calculation revisited

As in the previous chapters a linear student perceptron learning a teacher perceptron is considered. The quenched average of the free energy of the system is calculated by the replica method assuming replica symmetry. Returning to eq. (2.25), the replicated Hamiltonian is given by,

$$\begin{aligned} \exp(-G_{\mathrm{r}}[\mathbf{W}^\sigma]) \;=\; & \int \prod_\sigma \left\{ \frac{dx_\sigma d\hat{x}_\sigma}{2\pi} \right\} \frac{dy d\hat{y}}{2\pi} \\ & \times \exp\left[-\frac{1}{2}\beta \sum_\sigma (g(x_\sigma) - g_0(y))^2 + i \sum_\sigma x_\sigma \hat{x}_\sigma + iy\hat{y} \right] \\ & \times \int d\mu(\mathbf{s}) \exp[-iN^{-\frac{1}{2}}(\sum_\sigma \mathbf{W}^\sigma \hat{x}_\sigma + \mathbf{W}^0 \hat{y}) \cdot \mathbf{s}] \, . \end{aligned} \tag{6.3}$$

The general multivariate distribution of inputs eq. (6.2) is assumed and the integral over the input patterns may be evaluated. The definition of the order parameters is altered so that they are now measured in a different metric,

$$Q_{\sigma\rho} \;=\; \frac{1}{N}\mathbf{W}^{\sigma T}\Sigma_s \mathbf{W}^\rho \, , \tag{6.4}$$

$$R_\sigma = \frac{1}{N}\mathbf{W}^{0T}\Sigma_s\mathbf{W}^\sigma , \tag{6.5}$$

These order parameters are weighted by the covariance matrix of the input patterns. The introduction of the replica symmetric ansatz, averaging over the teacher distribution eq. (6.1) and subsequent calculation leads to the RS Hamiltonian for a linear student,

$$\begin{aligned}\mathcal{G}_r &= \frac{1}{2}\ln(1+\beta(q_0-q_1)) \\ &+\frac{\beta}{2(1+\beta(q_0-q_1))}\left(q_1 - \frac{2R}{\sqrt{\frac{1}{N}\mathrm{Tr}\ \Sigma_0\Sigma_s}}\left\langle xg_0\right\rangle_{\eta\Sigma_0\Sigma_s} + \left\langle g_0^2\right\rangle_{\eta\Sigma_0\Sigma_s}\right)\end{aligned}$$

where the order parameters are the replica symmetric versions of the weighted ones introduced above, eq. (6.4), (6.5) (*c.f.* §2.7) and the averages over the teacher are defined as,

$$\begin{aligned}\left\langle xg_0\right\rangle_{\eta\Sigma_0\Sigma_s} &= \int Dx\, d\mu(\eta) xg_0\left(\sqrt{\frac{1}{N}\mathrm{Tr}\ \Sigma_0\Sigma_s}\ x+\eta\right) , \\ \left\langle g_0^2\right\rangle_{\eta\Sigma_0\Sigma_s} &= \int Dx\, d\mu(\eta)\left(g_0\left(\sqrt{\frac{1}{N}\mathrm{Tr}\ \Sigma_0\Sigma_s}\ x+\eta\right)\right)^2 .\end{aligned} \tag{6.6}$$

The calculation of the prior constrained Hamiltonian is altered, due to the different definition of the order parameters and the general teacher prior eq. (6.1). The penalty term added to the cost function is assumed to be the general quadratic term introduced in the previous chapter. After some calculation the prior constrained Hamiltonian is,

$$\begin{aligned}\mathcal{G}_0 &= -R\hat{R} - q_0\hat{q}_0 + \frac{1}{2}q_1\hat{q}_1 + \frac{1}{2N}\ln|\beta\Lambda| - \frac{1}{2N}\ln|C| \\ &+\frac{1}{2N}\hat{R}^2\mathrm{Tr}\ \Sigma_s C^{-1}\Sigma_s\Sigma_0 + \frac{1}{2N}\hat{q}_1\mathrm{Tr}\ \Sigma_s C^{-1} .\end{aligned} \tag{6.7}$$

where Λ is the penalty matrix eq. (5.1) and the matrix C is given by,

$$C = \beta\Lambda + (\hat{q}_1 - 2\hat{q}_0)\Sigma_s$$

The free energy may be obtained, using eq. (2.34), in terms of $\mathcal{G}_r$, eq. (6.6), and $\mathcal{G}_0$, eq. (6.7), this may be differentiated to give the saddle point equations,

$$q_0 = q_1 + \frac{1}{N}\mathrm{Tr}\ D^{-1}\ , \tag{6.8}$$

$$q_1 = \hat{q}_1\frac{1}{N}\mathrm{Tr}\ D^{-2} + \hat{R}^2\frac{1}{N}\mathrm{Tr}\ D^{-2}\Sigma_s\Sigma_0\ , \tag{6.9}$$

$$R = \hat{R}\frac{1}{N}\mathrm{Tr}\ D^{-1}\Sigma_s\Sigma_0\ , \tag{6.10}$$

$$\hat{q}_0 = \frac{1}{2}\hat{q}_1 - \frac{\alpha\beta}{2(1+\beta(q_0-q_1))}\ , \tag{6.11}$$

$$\hat{q}_1 = \frac{\alpha\beta^2}{(1+\beta(q_0-q_1))^2}\left(q_1 - 2R\frac{\langle xg_0\rangle_{\eta\Sigma_0\Sigma_s}}{\sqrt{\frac{1}{N}\mathrm{Tr}\ \Sigma_s\Sigma_0}} + \left\langle g_0^2\right\rangle_{\eta\Sigma_0\Sigma_s}\right)\ , \tag{6.12}$$

$$\hat{R} = \frac{\alpha\beta}{(1+\beta(q_0-q_1))}\frac{\langle xg_0\rangle_{\eta\Sigma_0\Sigma_s}}{\sqrt{\frac{1}{N}\mathrm{Tr}\ \Sigma_s\Sigma_0}}\ , \tag{6.13}$$

with $D = C{\Sigma_s}^{-1}$. Writing the matrix D out in full,

$$D = \beta\Lambda{\Sigma_s}^{-1} + (\hat{q}_1 - 2\hat{q}_0)I\ . \tag{6.14}$$

These equations are identical to the case of a linear student using a quadratic penalty term with matrix, $\Lambda{\Sigma_s}^{-1}$, learning a teacher chosen from a distribution with covariance $\Sigma_s\Sigma_0$ with inputs chosen from an isotropic distribution. Hence the effect of the anisotropic input distribution is to change the definition of the order parameters so that they measure the weighted overlaps between weight vectors as well as to renormalise the penalty term and the teacher distribution.

It can be seen from eq. (6.8) and eq. (6.14) that the response function $\beta Q =$

$\beta(q_0 - q_1)$ for Gaussian input and teacher distributions and a penalty matrix Λ is the same as that for an isotropic input distribution with unit variance and penalty matrix $\Lambda\Sigma_s^{-1}$, *i.e.*, independent of the teacher distribution selected. This makes sense, since the response function measures the statistics of the student, and is not directly related to the teacher.

The saddle point equations may be solved analytically for particular forms of the input and teacher distributions and the penalty term. If the penalty term is the standard weight decay ($\Lambda = \lambda I$), the teacher chosen from a spherical prior ($\Sigma_0 = \Omega^2 I$) and the covariance matrix of the input distribution, Σ_s, is a multiple σ^2 of the identity matrix, then the saddle point equations are identical to the standard weight decay saddle point equations with a renormalised weight decay parameter, $\lambda' = \lambda/\sigma^2$. This result agrees with the scaling behaviour mentioned earlier in §3.4.

The average generalisation error is given by eq. (5.17) in terms of the saddle point values of the order parameters. The average training error is obtained by differentiating the free energy with respect to the inverse temperature whilst keeping the student prior constant and can be obtained from the corrupted generalisation error,

$$\epsilon_t = \frac{1}{(1+Q')^2}\left(\epsilon_g' + \frac{1}{2}TQ'^2\right) .$$

The performance of the network under different limits of the teacher and input distributions will now be considered. Initially, the input distribution will be fixed so that the effect of the novel teacher distribution can be studied.

6.3 Anisotropic teacher distribution

The distribution of inputs is assumed to be uniform as in the previous calculations, *i.e.*, $\Sigma_s = I$, this means that the matrix D in the saddle point equations is $\beta\Lambda + \hat{q}_1 - 2\hat{q}_0$. The response functions are independent of the teacher distribution used, and hence those calculated in the previous chapter may be used. The effect of different penalty matrices on a couple of teacher distributions will be studied. It has been shown in the previous chapters that the behaviour of some of the performance measures for a non-linear teacher are equivalent to a linear teacher with an effective gain and noise level, hence only noisy linear teachers will be considered, since a non-linear teacher will have qualitatively the same behaviour.

6.3.1 Anisotropic linear teacher

Consider a teacher prior that has a diagonal covariance matrix with one value, Ω_1^2, for a fraction k of the components and a different value, Ω_2^2, for the remaining fraction, *i.e.*,

$$(\Sigma_0)_{ij} = \begin{cases} \delta_{ij}\Omega_1^2 & i \leq kN \\ \delta_{ij}\Omega_2^2 & i > kN \end{cases} .$$

Since the penalty term is equivalent to the student prior, the penalty matrix is assumed to model the teacher prior, *i.e.*,

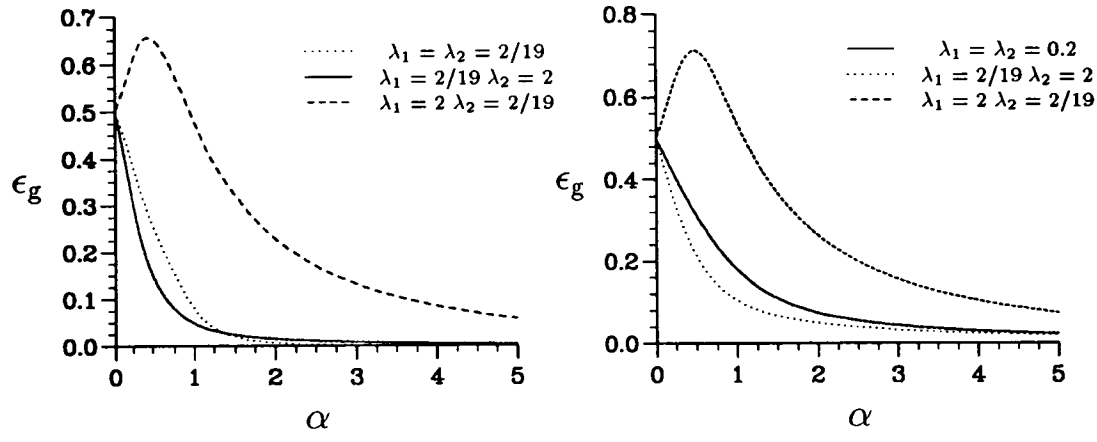
$$\Lambda_{ij} = \begin{cases} \delta_{ij}\lambda_1 & i \leq kN \\ \delta_{ij}\lambda_2 & i > kN \end{cases} ,$$

this is the penalty term studied in §5.7.1, thus the response function is that calculated previously, eq. (5.22), and hence the saddle point equations and performance measures can be calculated.

The average generalisation error for a linear student learning a teacher with a linear activation function is plotted in Fig. 6.1 for zero and non-zero noise and different values of the weight decay parameters. In the zero noise case, Fig. 6.1(a), the solid curve corresponds to the penalty matrix that captures the correct form of the teacher distribution. Initially this achieves better generalisation than the standard weight decay (dotted curve) then as alpha grows past $\alpha \sim 1.2$ the standard weight decay gives better performance. This behaviour can be explained by the fact that as α increases the data is able to specify the teacher and hence the penalty matrix that gives smaller decays will give lower generalisation error.

For the non-zero noise case, Fig. 6.1(b), the solid curve is the optimal standard weight decay for this noise level ($\lambda = \tilde{\gamma}^2 = 0.2$). The dotted curve can be numerically identified as the optimal weight decay parameters for this noise level, value of k and the form of the teacher covariance matrix, $\lambda_1 = \gamma^2/\Omega_1^2$ and $\lambda_2 = \gamma^2/\Omega_2^2$, where γ^2 is the actual noise level on the teacher. The dashed curve indicates what happens when the optimal parameters are swapped. The optimal weight decays make sense, since where the variance of the teacher is large, the weight decay term is small to let the student model the teacher, and conversely, where the variance of the teacher is small the weight decay is large to keep the student close to the teacher weight vector. Where the teacher is longer, the weight decay parameter is smaller, thus implying that the effect of noise on these components is less and *vice versa.*

If there is more information included in the teacher prior, the penalty matrix is



(a) Zero noise (b) Noise $\tilde{\gamma}^2 = 0.2$

Figure 6.1. Generalisation error versus α for a diagonal teacher, $\Omega_1^2 = 1.9, \Omega_2^2 = 0.1, k = 0.5$

able to use this information to improve the average generalisation error compared to the standard weight decay. There is an optimal form for the penalty matrix that sets the student prior to be equivalent to the teacher prior. Of more interest may be the case where the penalty matrix is close to the teacher prior but not exactly matched.

6.3.2 Highly correlated teacher

In the previous chapter, a penalty term was introduced that favoured correlations between the components of the teacher (§5.8). Consider the extreme case where all the components of the teacher are correlated with one another. In this case,

all the elements of the teacher covariance matrix are equal, *i.e.*,

$$(\Sigma_0)_{ij} = \Omega^2 \quad \forall\, i,j\ .$$

(This matrix is not strictly invertible. However it may be made invertible by adding a small constant ϵ to the diagonal. The results hold in the $\epsilon \rightarrow 0$ limit.) This teacher prior has only one degree of freedom, this means that there is only one parameter that the student needs to learn to model the teacher exactly. Using this teacher prior, the non-conjugate order parameters are given by,

$$q_0 = q_1 + \frac{1}{N}\mathrm{Tr}\ C^{-1}\ , \tag{6.15}$$

$$q_1 = \hat{q}_1 \frac{1}{N}\mathrm{Tr}\ C^{-2} + \hat{r}^2 \frac{1}{N}\sum_{ij}(C^{-1})^2_{ij}\ , \tag{6.16}$$

$$r = \hat{r}\frac{1}{N}\sum_{ij}(C^{-1})_{ij}\ , \tag{6.17}$$

where $C = \beta\Lambda + \hat{q}_1 - 2\hat{q}_0$. The conjugate order parameters are given by eq. (6.12) - (6.13).

The penalty term is assumed to be that studied in §5.8, *i.e.*,

$$\Lambda_{ij} = \delta_{ij}\,\lambda_1 + (1-\delta_{ij})\frac{\lambda_2}{N}\ .$$

When $\lambda_1 = -\lambda_2$ this penalty term penalises differences between components of the student weight vector. The response function $\mathcal{Q}'$ is simply $\frac{1}{N}\mathrm{Tr}\ \beta C^{-1}$.This can be calculated simply from the eigenvalues of C and in the thermodynamic limit is given by,

$$\mathcal{Q}' = \frac{1}{2\lambda_1}\left(1-\alpha-\lambda_1+\sqrt{(1+\alpha+\lambda_1)^2-4\alpha}\right)\ ,$$

which is the same as for a standard weight decay term up to terms of order $1/N$. This is because the response function is simply related to the sum of the eigenvalues of C and in this case, the off diagonal terms only add corrections of $O(1/N)$ to the eigenvalues. The other physical order parameters may be obtained in terms of $\phi = 1 + \frac{1}{\mathcal{Q}'} = \lambda_1(1 + \mathcal{Q}') + \alpha$ and $\phi' = (1 + \mathcal{Q}')/(\beta \sum_{ij} c_{ij}^{-1}) = (\lambda_1 + \lambda_2)(1 + \mathcal{Q}') + \alpha$. The order parameters, r and q_1 are,

$$r = \frac{\alpha \langle xg_0 \rangle_\eta}{\phi'} \tag{6.18}$$

$$q_1 = \frac{\alpha \langle xg_0 \rangle_\eta^2}{\phi^2 - \alpha} \left(\frac{\alpha}{\phi'^2}(\phi^2 - 2\phi') + 1 + \tilde{\gamma}_{\text{eff}}^2 \right) \tag{6.19}$$

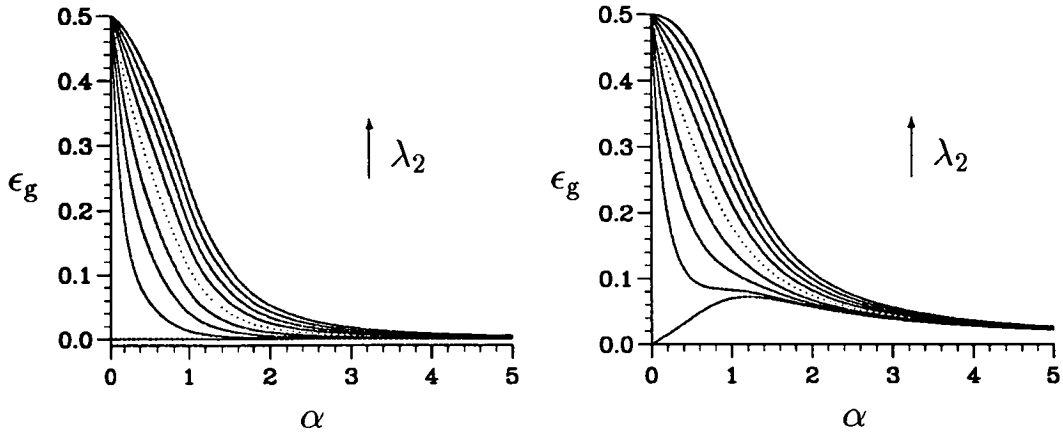
where $r = R/\Omega$ and $\tilde{\gamma}_{\text{eff}}^2 = \langle g_0^2 \rangle_\eta / \langle xg_0 \rangle_\eta^2 - 1$ has been used. The performance measures may now be evaluated for this teacher distribution and penalty term.

The corrupted generalisation error is given by,

$$\epsilon_g{}' = \frac{\langle xg_0 \rangle_\eta^2 \phi^2}{2(\phi^2 - \alpha)} \left(\frac{(\phi' - \alpha)^2}{\phi'^2} + \tilde{\gamma}_{\text{eff}}^2 \right) . \tag{6.20}$$

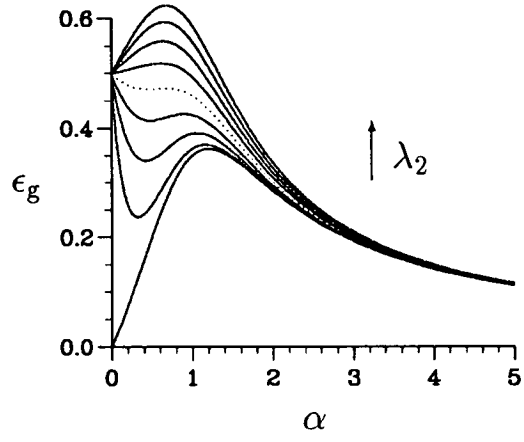
When $\lambda_1 = -\lambda_2$, $\phi' = \alpha$ and thus, $\epsilon_g{}' = \phi^2 \tilde{\gamma}_{\text{eff}}^2 / 2(\phi^2 - \alpha)$, in this case the generalisation error tends to zero as the weight decay parameter, λ_1, tends to infinity independently of the noise level.

The uncorrupted generalisation error, ϵ_g, is plotted for a linear teacher and a range of different penalty terms in Fig. 6.2. In the noise free case, the generalisation error for $\lambda_2 = -\lambda_1$ is equal to zero for all $\alpha > 0$ (at $\alpha = 0$ the generalisation error must be 0.5). This is because the teacher has a single degree of freedom and the student prior also has this form; the presentation of a single pattern is sufficient for the student to learn the teacher exactly. The case $\lambda_2 = -\lambda_1$ is



(a) Zero noise

(b) Noise 0.2



(c) Noise 1.0

Figure 6.2. Generalisation error for a highly correlated teacher and different amounts of noise. $\lambda_1 = 0.2, \lambda_2 = -0.2 \ldots 0.2$ in steps of 0.05. The dotted line corresponds to $\lambda_2 = 0$, *i.e.*, the standard weight decay.

the case where the penalty matrix prefers the components of the student weight vector to be equal. This student prior obviously matches the teacher and so should give the optimum generalisation. As α increases, the curves tend towards zero, again showing that given enough information, the prior is unimportant; the data swamps the prior.

For $\lambda_2 = -\lambda_1$ and non-zero noise, initially, the average generalisation error is small, then as the number of examples presented grows it increases up to a maximum at $\alpha = 1+\lambda_1$. The height of the maximum is $\tilde{\gamma}^2/(4(\lambda_1+\sqrt{\lambda_1(1+\lambda_1)}))$. The position of the maximum matches the discontinuity at $\alpha = 1$ for the pseudo-inverse solution trained on noisy data. As $\lambda_1 \to \infty$, the position of the maximum tends to infinity, but its height tends to zero. This means that the optimal penalty matrix is to take $\lambda_1 \to \infty$.

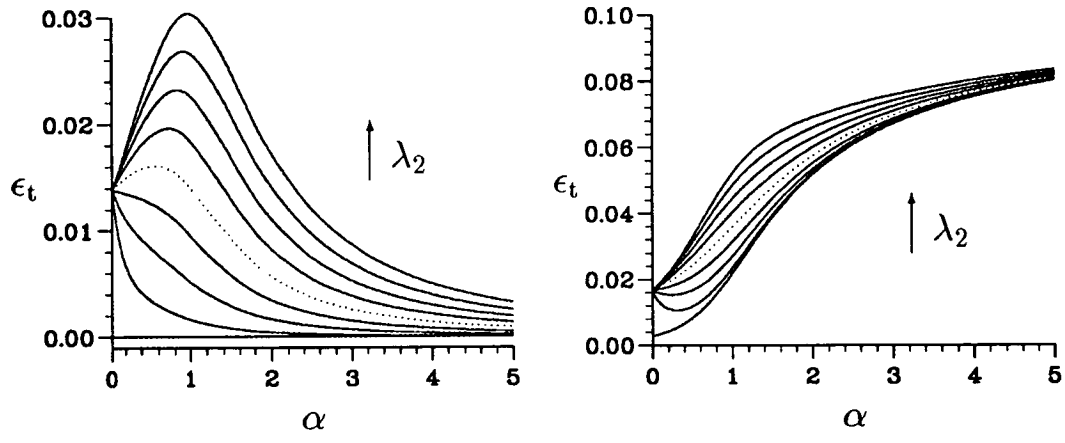
The curves show that when the student prior is able to take advantage of more information about the teacher, the performance is improved over the standard weight decay (the dotted line in Fig. 6.2). It can be postulated that for a teacher selected from a prior that contains a number of correlations, the optimal penalty term will be some where between the standard weight decay and the decay to average penalty term.

The average training may also be calculated. At zero temperature it is given by,

$$\epsilon_t = \frac{\langle xg_0 \rangle_\eta^2}{2(\phi^2 - \alpha)} \left((\lambda_1 + \lambda_2)\frac{\phi^2}{\phi'^2} + \tilde{\gamma}_{\text{eff}}^2(\phi - 1)^2 \right)$$

which is similar to the standard weight decay case eq. (3.37). When $\lambda_2 = -\lambda_2$, the average training error is the same as for the pseudo inverse solution. The training error is plotted for different weight decays in Fig. 6.3, again for $\lambda_1 = -\lambda_2$ and

zero noise, $\epsilon_t = 0 \; \forall \; \alpha > 0$, since a single noise free pattern enables the student to learn the teacher exactly. The standard weight decay is the dotted curve plotted in Fig. 6.3. The off-diagonal terms improve the performance in the case of a highly correlated teacher prior from the standard weight decay.



(a) Zero noise (b) $\tilde{\gamma}^2 = 0.2$

Figure 6.3. Training error for $\lambda_1 = 0.2, \lambda_2 = -0.2 \ldots 0.2$ in steps of 0.05. The dotted line corresponds to $\lambda_2 = 0$.

In a real world problem, it is possible that some of the input components are correlated, this has been dealt with previously by introducing weight sharing [56], where student weights that are close together in value are replaced with a single weight. This is what the off-diagonal penalty matrix achieves by introducing correlations between the weights. In the thermodynamic limit, the off-diagonal penalty term has the advantage that it does not degrade the performance when the teacher is drawn from a spherical distribution.

6.4 General input distribution

The effect of a general Gaussian input distribution on the model is now discussed. If the renormalised teacher distribution is assumed to be isotropic, *i.e.*, $\Sigma_s\Sigma_0 = \Omega^2 I$ and hence $\Sigma_0 = \Omega^2\Sigma_s^{-1}$, then the model is equivalent to a linear student learning a spherical teacher with a quadratic penalty term parametrised by $\Lambda\Sigma_s^{-1}$. In the previous chapter, it was suggested that the optimal penalty matrix for a spherical teacher is a standard weight decay, this implies that for optimal performance, $\Lambda = \lambda\Sigma_s$ where λ is a parameter that is determined by the noise level on the teacher and the length of the renormalised teacher, Ω^2. This result agrees with matching the student and teacher priors, since both the student and teacher priors have covariance $\propto \Sigma_s^{-1}$.

If the teacher is assumed to be drawn from a spherical distribution, *i.e.*, $\Sigma_0 = \Omega^2 I$, the optimum penalty matrix is suggested to be that which makes the covariance of the renormalised student prior, $(\Lambda\Sigma_s^{-1})^{-1}$ model the covariance of the renormalised teacher prior, $\Omega^2\Sigma_s$, up to a factor that is dependent on the noise level present on the teacher. This is achieved with $\Lambda \propto I$, *i.e.*, a standard weight decay. Thus in this case, the anisotropic input distribution has not affected the optimum penalty term. This can be explained by observing that the input distribution affects the student and teacher weight vectors equally and hence, the student gains no new information by modelling the input distribution. Hence the input distribution does not affect the optimal penalty matrix. It does, however, affect the interpretation of the order parameters for the Gaussian priors considered.

6.5 Optimal student prior

The previous calculations have suggested that the optimal student prior is that which models the distribution of teacher weights. The optimal penalty matrix for a linear student learning a linear teacher may be obtained simply by using the optimal learning algorithm of Watkin, [71]. The optimal student weight vector is equal to the teacher weight vector averaged over the posterior teacher distribution (sometimes referred to as the version space [70]). The posterior teacher distribution is the set of possible teachers that could have produced the data set, *i.e.*, the probability distribution of the teacher given the data. The average of the teacher weight vector over the posterior teacher distribution can be evaluated, this solution can be matched to the student obtained asymptotically for the zero temperature Gibbs learning algorithm and the optimal penalty matrix obtained,

$$\Lambda = \gamma^2 {\Sigma_0}^{-1} , \tag{6.21}$$

which is independent of the distribution of inputs. The optimal penalty matrix is obtained by matching the student prior to the teacher prior. For the case of the highly correlated teacher, the size of the optimal penalty term is infinite as observed earlier. This can be evaluated by considering a small multiple ϵ of the identity added to the covariance matrix of the teacher distribution and then considering the limit as $\epsilon \to 0$.

In the case of nonlinear teachers, it is likely that the optimal penalty term will be closely related to that identified above since a nonlinear teacher is simply a reparametrisation of the linear teacher case.

For distributions other than Gaussian it is harder to show that it is optimal to

match the form of the student prior to that of the teacher prior, though it is likely that this is true for tasks using continuous weights. Problems could arise if the average of the teacher over the version space was not a member of the accessible student weight space, as happens for binary weights; the optimal student prior is less clear in this case.

6.6 Different noise distributions

The noise on the training data considered in the previous chapters was simply added to the activation of the teacher inside the teacher activation function. There are other possible noise models that could be considered. This section will examine some of the simpler noise models.

6.6.1 Noise on teacher inputs

In section 3.2 the noise was introduced as zero mean Gaussian random noise on the activation of the teacher. An alternative noise model is to introduce noise on the inputs of the teacher, that is the output of the teacher is given by,

$$\sigma_0(\mathbf{s}) = g_0(\mathbf{W}^0 \cdot (\mathbf{s} + \boldsymbol{\eta})),$$

where $\boldsymbol{\eta}$ is a random vector drawn from a zero mean multivariate Gaussian distribution with covariance matrix Γ. The quenched average now contains an average over this noise vector. Since the vector of noise is a Gaussian random variable, then $\mathbf{W}^0 \cdot \boldsymbol{\eta}$ is also a zero mean Gaussian random variable with variance

$\frac{1}{N}\mathbf{W}^{0T}\Gamma\mathbf{W}^{0}$. After performing a teacher average, the variance of the noise becomes, $\frac{1}{N}\mathrm{Tr}\ \Gamma\Sigma_0$, where the teacher has been drawn from a Gaussian distribution with covariance Σ_0, eq. (6.1). Hence the noisy teacher inputs are equivalent to adding noise with variance $\frac{1}{N}\mathrm{Tr}\ \Gamma\Sigma_0$ to the teacher activation function, *i.e.*, the simple noise model discussed in §3.2 with a variance dependent on the covariance of the teacher as well as the noise. This means that the results for the optimal penalty term discussed previously still hold for this noise distribution.

6.6.2 Noise on data inputs

Noise on the teacher activation or on the teacher inputs models an error *within* the teacher, *i.e.*, the teacher is unreliable. Another possible noise model is to consider the case where the teacher itself is uncorrupted, but the inputs in the training set have been corrupted. In this case, the error measure is given by,

$$\epsilon(\mathbf{W};\mathbf{s},\boldsymbol{\eta}) = \frac{1}{2}\left(g(\mathbf{W}\cdot(\mathbf{s}+\boldsymbol{\eta})) - g_0(\mathbf{W}\cdot\mathbf{s})\right)^2 \ ,$$

where $\boldsymbol{\eta}$ is again a zero mean random vector drawn from a Gaussian distribution covariance Γ. Since both $\mathbf{s}$ and $\boldsymbol{\eta}$ are assumed to be Gaussian variables, the sum, $\mathbf{s}' = \mathbf{s}+\boldsymbol{\eta}$ is also a Gaussian variable with covariance $I+\Gamma$ for the case of uniform inputs. The error measure may be rewritten in terms of the sum of the input and noise $\mathbf{s}'$. This is equivalent to an error measure with inputs drawn from a distribution with covariance matrix $I+\Gamma$ and noise of variance $\frac{1}{N}\mathrm{Tr}\ \Gamma$ added to the teacher activation. This is again equivalent to the noise model in §3.2 but the distribution of inputs has been altered. In the previous section §6.4 it was observed that for Gaussian priors a different input distribution didn't affect

the optimal penalty term which was set to model the teacher prior, hence the different noise model does not affect the optimal penalty term.

The simple noise models introduced here do not affect the optimal penalty term that minimises the generalisation error. However in the case of noise added to the training data inputs, the distribution of examples is altered and this affects the interpretation of the order parameters.

6.7 Concluding Remarks

The calculations in the previous chapters assumed a simple isotropic distribution of inputs and a spherical teacher prior. The model used in the present chapter is extended to include a general Gaussian input distribution and teacher prior. The performance of a linear student learning a noisy teacher with a weight vector selected from an anisotropic distribution is considered. A diagonal teacher distribution with *two different* variances is considered. The penalty term is assumed to be of the same form, *i.e.*, *not* the standard weight decay. The optimal weight decay parameters are identified and these give lower generalisation error than the standard weight decay term.

A teacher drawn from a highly correlated distribution is also considered. In this case, the penalty matrix minimised differences between the components of the student. With the optimal penalty term the student is able to learn the teacher with the presentation of a single uncorrupted pattern. Penalty terms that interpolate between the optimal and the standard weight decay are also considered. There is a gradual improvement in performance as the student prior

approaches the teacher prior.

The effect of the general input distribution is to redefine the order parameters so that they are scaled by the covariance matrix of the input distribution. This means that the performance measures are dominated by the components with the largest variance. This is an argument for renormalising the input data to be of unit variance before training so that some components do not have a disproportionate affect on the performance measures.

The motivation for the standard weight decay term is to prefer models that have fewer weights, the motivation for minimising the difference between components is to prefer models that reduce the number of free parameters by sharing weights. Thus in practice the teacher prior is likely to be located somewhere between the spherical and highly correlated distributions. In this case, a penalty term that lies between the standard weight decay and the difference between components prior would give optimal generalisation.

For Gaussian priors better performance in terms of the average generalisation error, ϵ_g, is achieved by having a student prior that matches the teacher distribution. The size of the prior in relation to the training error is related to the noise level on the data. It is an open question whether this result is true for non-Gaussian priors.

Alternative noise models are introduced, which prove to be equivalent to the simple noise model introduced in §3.2. The case of noise added to the training set inputs means that the distribution of input examples is altered, which in turn means that the order parameters are weighted by the noise matrix. The optimal penalty matrix is still that which models the teacher prior. Thus the general

penalty term can improve the networks performance where it models the teacher distribution. If noise that affected the distribution of the teacher weights is added to the training data, then this could be countered by using a different penalty term.

Chapter 7

Summary and conclusions

In this thesis a linear student network learning a rule or mapping defined by a set of noisy examples is studied where the rule to be learnt is of both linear and nonlinear forms. The average performance of the network in terms of its ability to generalise is calculated for different learning scenarios which are characterised by a regulariser or prior on the student's network parameters. The optimal regulariser in terms of the generalisation ability for each learning scenario is evaluated. Finally, alternative rule and data distributions along with extensions to the noise model are considered.

The student is assumed to be a linear perceptron with continuous inputs and weights, and the data set is assumed to be generated by a known teacher network and a set of random inputs selected from an isotropic Gaussian distribution. The teacher is also taken to be a perceptron but with an arbitrary (linear or nonlinear) activation function. The similarity between the student and teacher networks enables direct comparisons to be drawn to discover how well the student

models the teacher. Noisy data is modeled by adding zero mean Gaussian noise to the activation of the teacher network. The student is trained on the corrupted example set using a stochastic gradient descent algorithm parametrised by a training temperature. A cost function is defined as the standard training error plus a potential term, which regularises the student's network parameters. Using the cost function, a free energy for the system is defined and calculated using the replica method of statistical mechanics assuming replica symmetry. Order parameters which capture the statistics of the system are evaluated from the free energy, these are then used to study the performance of the student network.

One facet of the student's performance is the ability to generalise to unseen examples, which is measured by the average generalisation error. There are two forms of the generalisation error studied; the corrupted generalisation error where the output of the student is compared to the corresponding noisy teacher output, or the uncorrupted generalisation where the student is compared to the "clean" teacher output. The corrupted generalisation error is a measure that is more likely to be calculated in practice, however the clean generalisation error gives more accurate information about how well the student has learned the teacher rule. The performance measures are used to compare the different penalty terms which are equivalent to selecting prior distributions of student weight vectors.

The standard weight decay penalty term added to the cost function for a linear student learning both a linear and nonlinear teacher is investigated. This penalty term is equivalent to picking an isotropic Gaussian distribution for the student prior. The response function as calculated by Hertz *et al* [38] arises naturally from the order parameters and hence the formalism gives an alternative method of calculating the response function for different priors. The order parameters and

performance measures for a nonlinear teacher, with the exception of the uncorrupted generalisation error, are equivalent to those for a linear student learning a linear teacher with a different gain and effective noise level on the linear teacher. Thus a linear student learning a nonlinear rule can be expected to achieve some success at generalisation, with the degree of success bounded by the effective noise level. The correlation between the student and teacher weight vectors depends on the teacher activation function only through the effective noise level. Thus as the number of examples increases, the student will be able to learn the teacher weight vector even though both the corrupted and uncorrupted generalisation error will be non-zero since the student activation function does not match the teacher's. There is a simple relation between the corrupted generalisation error and the training error which agrees with the results of Hansen [32] and so the expectation that a lower training error will give a lower generalisation error is justified for this model.

For both the linear and nonlinear teacher scenarios both the corrupted and uncorrupted generalisation errors are minimised with respect to the noise level on the training algorithm (temperature) when it is set to zero. The corrupted generalisation error is optimised with respect to the weight decay parameter when the weight decay is equal to the effective noise level on the teacher for both linear and nonlinear teachers. At these values of weight decay and training temperature the average cost function, consisting of the training error plus the standard weight decay penalty term, is constant for any number of examples in the training set and equal to the residual error due to the noise on the data. If the cost function is reduced below this residual error the data is overfitted and the generalisation ability of the network starts to degrade. Conversely, if the cost function is not reduced to the level of the residual error the data is under-fitted. This gives a

possible check on the optimality of the order parameters, and in a real problem may be used to indicate whether the data is under- or overfitted. The average training error is constant and equal to the residual error when the weight decay parameter and the training temperature both equal the noise level. These values of the parameters are those identified by Bruce and Saad [9] that minimise the variance of the student network's output. If the training data is uncorrupted, the optimal weight decay parameter for the model is infinitesimally small, which is equivalent to the pseudo inverse solution. Thus it can be seen that a weight decay term is able to improve generalisation for the case of a linear student learning a noisy linear or nonlinear teacher.

For a linear student learning a linear teacher, the optimum weight decay parameter for finite temperatures is calculated numerically. There is a temperature above which the optimum weight decay is infinite. This can be interpreted as the value of the temperature where the dynamic noise introduced in the stochastic training algorithm swamps the information contained in the data and could suggest an initial temperature for an annealing schedule.

Since the weight decay penalty term was seen to improve the generalisation ability of a network trained on noisy data, it can be postulated that there is a more general form of the penalty term that may give further improvements in the generalisation ability. A simple generalisation of the standard weight decay is to consider a generalised quadratic penalty term which is equivalent to a multivariate Gaussian prior on the student weights. The free energy is averaged over a prior distribution of teacher weight vectors to remove an explicit dependence on the teacher weight vector. Initially a spherical teacher prior is assumed and it turns out that the free energy is invariant under a rotation of the weight space and a corresponding rotation of the input space, this means that any penalty

matrix is equivalent to its diagonal form. This invariance under rotation is due to the averaging over the spherical teacher distribution. For a specific form of the generalised penalty term the uncorrupted generalisation error is not improved over that achieved by the standard weight decay and hence it is postulated that optimal generalisation is achieved by choosing a student prior (and hence penalty term) that models the teacher prior.

In order to investigate the effect of different teacher priors, the teacher is assumed to be drawn from a general anisotropic Gaussian distribution. The effect of different penalty terms can then be evaluated. When the penalty term models the teacher prior the performance is improved and it is shown analytically that for Gaussian priors on the student and teacher weight vectors and a linear student learning a linear teacher, the generalisation is optimised by selecting a student prior that models the teacher distribution up to a constant of proportionality that is equal to the noise level on the data.

The effect of a general Gaussian distribution of inputs is also considered. The order parameters are now weighted by the covariance matrix of the input distribution, this means that the order parameters no longer measure overlaps between weight vectors since components of the input that have large variance will dominate the performance measures. The different input distribution does not alter the form of the optimal penalty matrix. This makes sense, since the different input distribution does not affect the teacher distribution, so as seen before, for a linear student learning a linear teacher the optimal penalty term is that which sets the student prior to be equivalent to the teacher distribution.

It can be postulated that the novel penalty terms may be able to improve the performance for a different noise model added to the data. Gaussian noise added

to the inputs of the teacher is shown to be equivalent to simple noise added to the activation of the teacher. Noise added to the inputs of the training data is also considered and shown to be equivalent to training with a different input distribution along with the simple noise model. Hence the form of the optimal penalty matrix is not dependent of the noise distribution that is added to the data set. However, if the teacher weights are corrupted by some fixed noise, this would alter the distribution from which the teacher is selected and means that the optimal penalty term would include this noise distribution. Otherwise the optimal penalty term only depends on the noise through its effective variance.

The distributions assumed in this thesis were Gaussian since it is simple to calculate averages over these distributions, further work could be to consider other forms of distribution, though this would inevitably make the calculations more complicated. The results are exact in the thermodynamic limit, *i.e.*, the number of weights and inputs, N, tends to infinity, numerical simulations could be carried out to check the results hold for finite N systems and the $O(1/N)$ corrections could be calculated using other methods [67]. The formalism could possibly be extended to multilayer networks using some of the methods introduced recently, [62] to give results for networks that are of more practical use, however it is likely that this will make the calculation much more complicated and may in fact not be tractable.

Bibliography

[1] D. Amit. *Modelling Brain Function.* Cambridge University Press, Cambridge, 1989.

[2] D. Amit, H. Gutfreund, and H. Sompolinsky. Spin-glass models of neural networks. *Physical Review A*, **32** 1007–1018, 1985.

[3] D. Amit, H. Gutfreund, and H. Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, **55** 1530–1533, 1985.

[4] D. Amit, H. Gutfreund, and H. Sompolinsky. Information storage in neural networks with low levels of activity. *Physical Review A*, **35** 2293–2303, 1987.

[5] J.A. Anderson and E. Rosenfeld, editors. *Neurocomputing: Foundations of Research.* MIT Press, Cambridge, 1988.

[6] R. Beale and T. Jackson. *Neural Computing - an introduction.* Adam Hilger, 1990.

[7] K. Binder and A.P. Young. Spin glasses: Experimental facts, theoretical concepts, and open questions. *Reviews of Modern Physics*, **58** 801–976, 1986.

[8] S. Bös, W. Kinzel, and M. Opper. Generalisation ability of perceptrons with continuous outputs. *Phys Rev E*, **47** 1384–1391, 1993.

[9] A.D. Bruce and D. Saad. Statistical mechanics of hypothesis evaluation. *J.Phys. A*, **27** 3355–3363, 1994.

[10] J.-Q. Chen. *Group representation theory for physicists.* World Scientific, 1989.

[11] E.T. Copson. *Asymptotic expansions.* Cambridge University Press, 1971.

[12] P.H. Damgaard and H. Hüffel. Stochastic quantization. *Physics Reports*, **152** 227–398, 1987.

[13] J.R.L. de Almedia and J. Thouless. Stability of the sherrington-kirkpatrick solution of a spin glass model. *J. Phys. A*, **11** 983–990, 1978.

[14] A.P. Dunmur and D.J. Wallace. Learning and generalisation in a linear perceptron stochastically trained with noisy data. *J. Phys. A*, **26** 5767–5779, 1993.

[15] S.F. Edwards and P.W. Anderson. Theory of spin glasses. *J. Phys. F*, **5** 965, 1975.

[16] B. Efron. Bootstrap methods, another look at the jackknife. *Ann. Statist.*, **7** 1-26, 1979.

[17] B. Efron and R. LePage. Introduction to bootstrap. In R Lepage and L Billard, editors, *Exploring the Limits of Bootstrap*, pages 3–10. Wiley, 1992.

[18] S.E. Fahlman. Fast-learning variations on back-propagation: An empirical study. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 38–51, Pittsburg 1988, 1989. Morgan Kaufmann, San Mateo.

[19] S.E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 524–532, Denver 1989, 1990. Morgan Kaufmann, San Mateo.

[20] R. Fletcher. *Practical Methods of optimisation. Volume 1. Unconstrained optimization.* Wiley, 1980.

[21] J.F. Fontanari and R. Meir. Learning from examples in weight-constrained neural networks. *J. Phys. A*, **25** 1149–1168, 1992.

[22] M Frean. The upstart algorithm: A method for constructing and training feedforward neural networks. *Neural Computation*, **2** 198–209, 1990.

[23] K. Funahashi. On the approximate realization of continuous-mappings by neural networks. *Neural Networks*, **2** 183–192, 1989.

[24] E. Gardner. Maximum storage capacity in neural networks. *Europhysics Letters*, **4** 481–485, 1987.

[25] E. Gardner. The space of interactions in neural network models. *J. Phys. A*, **21** 257–270, 1988.

[26] E. Gardner and B. Derrida. Optimal storage properties of neural network models. *J. Phys. A*, **21** 271–284, 1988.

[27] E. Gardner, N. Stroud, and D.J. Wallace. Training with noise: application to word and text storage. In R. Eckmiller, editor, *Neural Computers: From Computational Neuroscience To Computer Design*, pages 251–260. Springer-Verlag, 1987.

[28] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press, 1981.

[29] D.E. Goldberg. *Genetic Algorithms in search optimization and machine learning.* Addison-Wesley, 1989.

[30] F.A. Greybil. *Matrices with applications in statistics.* Wadsworth International Group, 1983.

[31] G. Györgyi and N. Tishby. Statistical theory of learning a rule. In W K Theumann and R Köberle, editors, *Neural Networks and Spin Glasses*, pages 3–36. World Scientific, Singapore, 1990.

[32] L.K. Hansen. Stochastic linear learning: Exact test and training error averages. *Neural Networks*, **6** 393–396, 1993.

[33] B. Hassibi and D.G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In S.J. Hanson, J.D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 164–171, Denver 1992, 1993. Morgan Kaufmann, San Mateo.

[34] T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models.* Chapman and Hall, 1990.

[35] D. Haussler. The probably approximately correct (pac) and other learning models. In A Meyrowitz and S Chipman, editors, *Foundations of knowledge acquisition: Machine Learning.* Kluwer, 1994.

[36] D.O. Hebb. *The Organization of Behavior.* Wiley, New York, 1949. Partially reprinted in [5].

[37] J.A. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the theory of Neural Computation.* Redwood City: Addison-Wesley, 1991.

[38] J.A. Hertz, A. Krogh, and G.I. Thorbergsson. Phase transitions in simple learning. *J. Phys. A*, **22** 2133–2150, 1989.

[39] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, **79**, 1982. Reprinted in [5].

[40] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, **2** 359–366, 1989.

[41] S. Kirkpatrick, C.D. Gelatt Jr., , and M.P. Vecchi. Optimization by simulated annealing. *Science*, **220**, 1983. Reprinted in [5].

[42] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, **43**, 1982. Reprinted in [5].

[43] A. Krogh and J.A. Hertz. Generalization in a linear perceptron in the presence of noise. *J. Phys. A*, **25** 1135–1147, 1992.

[44] Landau and Lifshitz. *Statistical physics.* Permagon press, 1980.

[45] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, **1** 541–551, 1989.

[46] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Handwritten digit recognition with a back-propagation network. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 396–404, Denver 1989, 1990. Morgan Kaufmann, San Mateo.

[47] Y. Le Cun, J.S. Denker, and S.A. Solla. Optimal brain damage. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 598–605, Denver 1989, 1990. Morgan Kaufmann, San Mateo.

[48] E. Levin, N. Tishby, and S. Solla. A statistical approach to learning and generalization in layered neural networks. In *Proc. 2nd Workshop on Computational Learning Theory.* Morgan Kaufmann, 1989.

[49] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, **4** 415–447, 1992.

[50] D.J.C. MacKay. A practical bayesian framework for backprop networks. *Neural Computation*, **4** 698–714, 1992.

[51] M. Marron. A comparison of cross-validation techniques in density estimation. *Annals of statistics*, **15** 152–162, 1987.

[52] W.S. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, **5**, 1943. Reprinted in [5].

[53] M. Mézard and J.-P. Nadal. Learning in feedforward layered networks: The tiling algorithm. *Journal of Physics A*, **22** 2191–2204, 1989.

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

[55] M.L. Minsky and S.A. Papert. *Perceptrons.* MIT Press, Cambridge, 1969.

[56] S.J. Nowlan and G.E. Hinton. Simplifying neural networks by soft weight sharing. *Neural Computation*, **4** 473–493, 1992.

[57] D.B. Parker. Learning logic. Technical Report TR–47, Center for Computational Research in Economics and Management Science, Massachusetts Institute of Technology, Cambridge, MA, 1985.

[58] D.A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1, pages 305–313, Denver 1988, 1989. Morgan Kaufmann, San Mateo.

[59] F. Rosenblatt. *Principles of Neurodynamics.* Spartan, New York, 1962.

[60] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, **323** 533–536, 1986. Reprinted in [5].

[61] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, Cambridge, 1986.

[62] D. Saad. Explicit symmetries and the capacity of multilayer neural networks. *J. Phys. A*, **27** 2719–2734, 1994.

[63] H. Schwarze and J. Hertz. Learning from examples in fully connected commitee machines. *Phys. Rev. A*, **26** 4919–4936, 1993.

[64] H.S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning. *Phys. Rev. A*, **45** 6056–6091, 1992.

[65] C.R. Smith and W.T. Grandy Jr, editors. *Maximum-entropy and Bayesian methods in inverse problems.* Dordrecht Lancaster : D. Reidel, 1985.

[66] S.A. Solla, E. Levin, and M. Fleisher. Accelerated learning in layered neural networks. *Complex Systems*, **2** 625–639, 1988.

[67] P. Sollich. Finite size effects in learning and generalisation in linear perceptrons. Submitted to *J. Phys. A*, 1994.

[68] G. Tesauro and T.J. Sejnowski. A "neural" network that learns to play backgammon. In D.Z. Anderson, editor, *Neural Information Processing Systems*, pages 442–456, Denver 1987, 1988. American Institute of Physics, New York.

[69] V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, **16** 264–280, 1971.

[70] T.H.L. Watkin, A. Rau, and M. Biehl. The statistical mechanics of learning a rule. *Rev. Mod. Phys*, **65** 499–556, 1993.

[71] T.L.H. Watkin. Optimal learning in a neural network. *Europhys. Letts*, **21** 871–876, 1993.

[72] A.S. Weigend, B.A. Huberman, and D.E. Rumelhart. Predicting the future: a connectionist approach. *Int. Jour. of Neural Syst.*, **1** 193–209, 1990.

[73] P. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* PhD thesis, Harvard University, 1974.

[74] G.B. Wetherill. *Regression Analysis with Applications.* Chapman and Hall, 1986.

[75] D. Wolpert. On the connection between in-sample testing and generalization error. *Complex Systems*, **6** 47–94, 1992.

Appendix A

Notation

This section lists the main notation that is commonly used. Some of the symbols also have other meanings, however the differences should be clear by context.

α	The number of examples per weight (p/N).
β	$1/T$.
$\epsilon(\mathbf{W};\mathbf{s})$	The error measure.
$\epsilon(\mathbf{W})$	The generalisation function.
ϵ_{g}	The average uncorrupted generalisation error.
ϵ_{g}'	The average corrupted generalisation error.
ϵ_{t}	The average training error.
ϵ_{c}	The average cost function.
F	The free energy.
f	The free energy per weight, F/N.
$g(x)$	The activation function of the student.

$g_0(x)$	The activation function of the teacher.
G_r	The replicated Hamiltonian.
G_0	The prior constrained Hamiltonian.
$\mathcal{G}_r$	$\lim_{n\to 0} G_r/n$.
$\mathcal{G}_0$	$\lim_{n\to 0} G_0/n$.
γ^2	The variance of noise on the data.
$\tilde{\gamma}^2$	The variance of noise on the data normalised by the length of the teacher squared, γ^2/Ω^2.
Γ	The covariance matrix of a general Gaussian noise distribution.
λ	The weight decay parameter.
Λ	The general penalty matrix.
N	The number of weights and inputs.
n	The number of replicas.
Ω^2	The squared length of the teacher, $\frac{1}{N}\mathbf{W}^0 \cdot \mathbf{W}^0$.
p	The number of examples in the training set.
ϕ	$1 + 1/\mathcal{Q}'$.
q_0	The replica symmetric average intra-replica overlap, $\frac{1}{N}\mathbf{W}^\sigma \cdot \mathbf{W}^\sigma$.
q_1	The replica symmetric average inter-replica overlap, $\frac{1}{N}\mathbf{W}^\sigma \cdot \mathbf{W}^\rho$.
$\hat{q}_0$	The conjugate order parameter to q_0.
$\hat{q}_1$	The conjugate order parameter to q_1.
q'	The average cosine between replica solutions q_1/q_0.
$\mathcal{Q}$	The difference between overlaps, $q_0 - q_1$.
$\mathcal{Q}'$	The response function, $\beta\mathcal{Q}$.
R	The replica symmetric average overlap between student and teacher $\frac{1}{N}\mathbf{W}^\sigma \cdot \mathbf{W}^0$.
$\hat{R}$	The conjugate order parameter to R,

r	The average overlap between student and teacher divided by the length of the teacher, R/Ω.
$\hat{r}$	The conjugate order parameter to r, $\hat{R}\Omega$.
R'	The average cosine of the angle between student and teacher $R/\Omega\sqrt{q_0}$.
$\mathbf{s}$	An input vector.
Σ_0	The covariance matrix of a general Gaussian distribution of teacher weight vectors.
Σ_s	The covariance matrix of a general Gaussian input distribution.
T	The variance of noise added to the stochastic training algorithm, the temperature.
Θ	The training set.
θ	An example consisting of an input-output pair.
$\mathbf{W}$	A set of network parameters, the student weights.
$\mathbf{W}^0$	The weights of the teacher.
$\boldsymbol{\xi}$	The input of an example.
ζ	The output of an example.

Appendix B

Integral Identities

This appendix states a couple of useful integral identities.

B.1 Delta function - integral representation

A delta function may be written as,

$$\delta(a-b) = \int_{-\infty}^{\infty} \frac{dx}{2\pi} \exp[ix(a-b)] \,,$$

or

$$\delta(a-b) = \int_{-i\infty}^{i\infty} \frac{dx}{2\pi i} \exp[-x(a-b)] \,.$$

B.2 Hubbard Stratonovitch transformation

This is a useful trick for linearising quadratic exponential terms at the expense of introducing an integration over an additional variable. Explicitly it is,

$$\exp\left[\tfrac{1}{2}b^2\right] = \int_{-\infty}^{\infty} \frac{dx}{\sqrt{2\pi}} \exp\left[-\tfrac{1}{2}x^2 + bx\right] .$$

B.3 Gaussian Integration

The notation,

$$Dx = \frac{dx}{\sqrt{2\pi}} \exp\left[-\tfrac{1}{2}x^2\right]$$

is commonly used.

Appendix C

Solving standard saddle point equations

The saddle point equations for a linear student with a weight decay learning a noisy teacher are given by eq. (3.12 - 3.16) for completeness written here,

$$q_0 = q_1 + \frac{1}{\beta\lambda + \hat{q}_1 - 2\hat{q}_0} \tag{C.1}$$

$$q_1 = (\hat{r}^2 + \hat{q}_1)(q_0 - q_1)^2 \tag{C.2}$$

$$\hat{q}_0 = \frac{1}{2}(\hat{q}_1 - \frac{\hat{r}}{\langle x g_0 \rangle_\eta}) \tag{C.3}$$

$$\hat{q}_1 = \frac{\alpha\beta^2}{(1+\beta(q_0-q_1))^2}\left(q_1 - 2r\langle x g_0 \rangle_\eta + \left\langle g_0^2 \right\rangle_\eta\right) \tag{C.4}$$

$$r = \hat{r}(q_0 - q_1) \tag{C.5}$$

$$\hat{r} = \frac{\alpha\beta}{1+\beta(q_0-q_1)}\langle x g_0 \rangle_\eta \tag{C.6}$$

From equations (C.1) and (C.2), define $\mathcal{Q} = q_0 - q_1$ and using equation (C.6)

gives

$$\frac{1}{\mathcal{Q}} = \beta\tilde{\lambda} + \frac{\alpha\beta}{(1+\beta\mathcal{Q})} \langle x\, g_0 \rangle_\eta$$

Consider $\mathcal{Q}' = \beta\mathcal{Q}$, the equation above leads to the quadratic equation,

$$\tilde{\lambda}\mathcal{Q}'^2 + \mathcal{Q}'\left(\alpha + \tilde{\lambda} - 1\right) - 1 = 0 \tag{C.7}$$

This equation only has roots,

$$\mathcal{Q}' = \frac{1}{2\,\tilde{\lambda}}\left(1 - \alpha - \tilde{\lambda}\right) + \frac{1}{2\,\tilde{\lambda}}\sqrt{\left(1-\alpha-\tilde{\lambda}\right)^2 + 4\,\tilde{\lambda}}\,,$$

where $\tilde{\lambda} = \frac{\lambda}{\nu^2}$, which gives eq. (3.17).

So having calculated, $\mathcal{Q}'$ and hence $\mathcal{Q}$, $\hat{r}$ may be evaluated,

$$\hat{r} = \frac{\alpha\beta}{1+\mathcal{Q}'} \langle x\, g_0 \rangle_\eta$$

Define $\phi = 1 + \frac{1}{\mathcal{Q}'}$ and thus from equation (C.5),

$$r = \frac{\alpha \langle x\, g_0 \rangle_\eta}{\phi}$$

Now, substituting the values obtained for $r, \hat{r}$ and equation (C.4) into equation (C.2) and using the definition of ϕ gives,

$$q_1 = \frac{\alpha}{(\phi^2 - \alpha)}\left(\alpha \langle x\, g_0 \rangle_\eta^2 \left(1 - \frac{2}{\phi}\right) + \left\langle g_0^2 \right\rangle_\eta\right) \tag{C.8}$$

Substituting the definition of ϕ into the quadratic equation for ϕ gives a quadratic equation in ϕ,

$$\phi^2 - \phi(1 + \alpha + \lambda) + \alpha = 0$$

Using this equation,

$$\alpha\left(1 - \frac{2}{\phi}\right) = \frac{(\phi^2 - \alpha)}{\phi} - (1 + \lambda)$$

and thus eq. (3.19),

$$q_1 = \frac{\alpha}{(\phi^2 - \alpha)}\left(\left\langle g_0^2 \right\rangle_\eta - \langle x\, g_0 \rangle_\eta^2 (1 + \lambda)\right) + \frac{\alpha \langle x\, g_0 \rangle_\eta^2}{\phi}$$

This result may be substituted into $\hat{q}_1$, eq. (C.4), using $\alpha + \alpha\lambda - \alpha^2/\phi = \alpha\phi - \alpha^2$, and $(\phi - 1)(\phi - \alpha) = \lambda\phi$, gives, eq. (3.22),

$$\hat{q}_1 = \frac{\alpha\beta^2}{(\phi^2 - \alpha)}\left(\langle x\, g_0 \rangle_\eta^2 \lambda^2 + \left(\left\langle g_0^2 \right\rangle_\eta - \langle x\, g_0 \rangle_\eta^2\right)(\phi - 1)^2\right) .$$

Appendix D

Relation between ϵ_t and ϵ_g'

The corrupted generalisation error is given by eq. (3.35),

$$\epsilon_g{}' = \frac{\langle x\, g_0 \rangle_\eta^2}{2} \left(1 + \tilde{\gamma}_{\text{eff}}^2 + \frac{\alpha}{(\phi^2 - \alpha)} \left(\tilde{\gamma}_{\text{eff}}^2 - \lambda \right) - \frac{\alpha}{\phi} \right) + \frac{T}{2(\phi - 1)}$$

Rearranging and using, $\phi + \lambda\phi - \alpha = \phi^2 - \alpha\phi$ yields,

$$\epsilon_g{}' = \frac{\langle x\, g_0 \rangle_\eta^2}{2(\phi^2 - \alpha)} \left(\tilde{\gamma}_{\text{eff}}^2 \phi^2 + (\phi - \alpha)^2 \right) + \frac{T}{2(\phi - 1)}$$

This can be compared to the average training error, eq. (3.37),

$$\epsilon_t = \frac{1}{2} \frac{\langle x\, g_0 \rangle_\eta^2}{(\phi^2 - \alpha)} \left(\lambda^2 + \tilde{\gamma}_{\text{eff}}^2 (\phi - 1)^2 \right) + \frac{T}{2\phi} \,.$$

Since, $(\phi - 1)(\phi - \alpha) = \lambda\phi$,

$$\epsilon_t = \frac{(\phi - 1)^2}{\phi^2} \left(\epsilon_g{}' + \frac{T}{2(\phi - 1)^2} \right)$$

Now, $\mathcal{Q}' = 1/(\phi - 1)$ and $(1 + \mathcal{Q}') = \phi/(\phi - 1)$, hence eq. (3.39),

$$\epsilon_t = \frac{1}{(1 + \mathcal{Q}')^2} \left(\epsilon_g{}' + \frac{T\mathcal{Q}'^2}{2} \right) ,$$

Appendix E

General weight decay, order parameters.

For a linear student using a general quadratic penalty term learning a noisy teacher, the free energy per weight is written as,

$$\begin{aligned} -\beta f &= -R\hat{R} - q_0\hat{q}_0 + \frac{1}{2}q_1\hat{q}_1 + \frac{1}{2}\ln 2\pi + \frac{1}{2}\Omega^2 - \frac{1}{2N}\ln|C| \\ &\quad + \frac{1}{2N}(\hat{R}^2\Omega^2 + \hat{q}_1)\mathrm{Tr}\; C^{-1} - \frac{\alpha}{2}\ln(1+\beta(q_0-q_1)) \\ &\quad - \frac{\alpha\beta}{2(1+\beta(q_0-q_1))}\left(q_1 - 2\frac{R}{\Omega}\langle xg_0\rangle_\eta + \left\langle g_0^2\right\rangle_\eta\right) \end{aligned} \tag{E.1}$$

Differentiating with respect to the order parameters, $R, \hat{R}, q_1$ and q_1, and setting the derivatives equal to zero gives,

$$\hat{r} = \frac{\alpha\beta}{(1+\beta(q_0-q_1))}\langle xg_0\rangle_\eta$$

$$
\begin{aligned}
r &= \hat{r}\frac{1}{N}\mathrm{Tr}\ C^{-1} \\
\hat{q}_0 &= -\frac{\alpha\beta}{2(1+\beta(q_0-q_1))} + \frac{\alpha\beta^2}{2(1+\beta(q_0-q_1))^2}\left(q_1 - 2r\left\langle xg_0\right\rangle_\eta + \left\langle g_0^2\right\rangle_\eta\right) \\
\hat{q}_1 &= \frac{\alpha\beta^2}{(1+\beta(q_0-q_1))^2}\left(q_1 - 2r\left\langle xg_0\right\rangle_\eta + \left\langle g_0^2\right\rangle_\eta\right)
\end{aligned}
$$

where, $r = R/\Omega$ and $\hat{r} = \hat{R}\Omega$.

In order to calculate the remaining order parameter, we need the following results,

Consider $C = xA + yB$,

$$\frac{\partial}{\partial x}\ln|C| = \mathrm{Tr}\ AC^{-1}$$

similarly

$$\frac{\partial}{\partial x}\mathrm{Tr}\ C^{-1} = -\mathrm{Tr}\ A\left(C^{-1}\right)^2$$

Hence, the remaining order parameters are,

$$
\begin{aligned}
q_0 &= -\frac{1}{2N}\frac{\partial}{\partial\hat{q}_0}\ln|C| + (\hat{r}^2+\hat{q}_1)\frac{1}{2N}\frac{\partial}{\partial\hat{q}_0}\mathrm{Tr}\ C^{-1} \\
&= q_1 + \frac{1}{N}\mathrm{Tr}\ C^{-1} \\
q_1 &= \frac{1}{N}\frac{\partial}{\partial\hat{q}_1}\ln|C| - \frac{1}{N}\mathrm{Tr}\ C^{-1} + (\hat{r}^2+\hat{q}_1)\frac{1}{2N}\frac{\partial}{\partial\hat{q}_1}\mathrm{Tr}\ C^{-1} \\
&= (\hat{r}^2+\hat{q}_1)\frac{1}{N}\mathrm{Tr}\ C^{-2}
\end{aligned}
$$

Appendix F

Cubic equations

In previous chapters, the roots of a cubic equation were needed, hence, consider the general cubic equation given by

$$f(x) = x^3 + a_2\,x^2 + a_1\,x + a_0$$

F.1 Number of positive roots

Differentiating w.r.t. x gives

$$f'(x) = 3\,x^2 + 2\,a_2\,x + a_1$$

Thus the maximum and minimum of f are given by the two points,

$$\begin{aligned} x_1 &= \frac{1}{3}(-a_2 + \sqrt{a_2^2 - 3\,a_1}) \\ x_2 &= \frac{1}{3}(-a_2 - \sqrt{a_2^2 - 3\,a_1}) \end{aligned}$$

Differentiating a second time gives

$$f''(x) = 2(3\,x + a_2)$$

Thus $f''(x_1) = 2\sqrt{a_2^2 - 3\,a_1} > 0$ and $f''(x_2) = -2\sqrt{a_2^2 - 3\,a_1} < 0$, therefore, x_1 is a minimum and x_2 is a maximum. $x_1 > x_2$ always.

The maxima occurs on the left of the origin if $x_2 < 0$, this leads to the condition

$$a_2 > -\sqrt{a_2^2 - 3\,a_1}$$

For $a_2 > 0$ this condition is always true, for $a_2 < 0$ this condition is true if $a_1 < 0$. thus the conditions on the maximum lying on the left of the origin may be written

$$x_2 < 0 \text{ if } \begin{cases} a_2 \geq 0 \\ a_2 < 0 \text{ and } a_1 < 0 \end{cases}$$

Now, $f(0) = a_0$, so if $a_0 < 0$ and $x_2 < 0$, then there is a maximum on the left of the origin and the value of f as it crosses the y axis is less than zero, thus there

is only one positive root if;

$$a_0 < 0 \text{ and } a_2 \geq 0$$
$$a_0 < 0 \text{ and } a_2 < 0 \text{ and } a_1 < 0$$

F.2 The roots

The roots of $f(x) = 0$ can be given in terms of two parameters, p_1, p_2,

$$\begin{aligned} p_1 &= \frac{1}{9}a_2^2 - \frac{1}{3}a_1 \\ p_2 &= \frac{1}{6}(a_1 a_2 - 3a_0) - \frac{1}{27}a_2^3 \end{aligned}$$

There are three types of solution which are identified by

- $p_2^2 - p_1^3 > 0$ One real root, pair of complex roots,
- $p_2^2 - p_1^3 = 0$ All roots real, at least two equal,
- $p_2^2 - p_1^3 < 0$ All roots real.

The roots themselves are in terms of the further parameters,

$$\begin{aligned} s_1 &= \left[p_2 + (p_2^2 - p_1^3)^{\frac{1}{2}}\right]^{\frac{1}{3}} \\ s_2 &= \left[p_2 - (p_2^2 - p_1^3)^{\frac{1}{2}}\right]^{\frac{1}{3}} \end{aligned}$$

The roots are then

$$\begin{aligned} x_1 &= (s_1 + s_2) - \frac{1}{3}a_2 \\ x_2 &= -\frac{1}{2}(s_1 + s_2) - \frac{1}{3}a_2 + i\frac{\sqrt{3}}{2}(s_1 - s_2) \\ x_3 &= -\frac{1}{2}(s_1 + s_2) - \frac{1}{3}a_2 - i\frac{\sqrt{3}}{2}(s_1 - s_2) \end{aligned}$$

Since the only roots of interest are positive real ones, the cases where $p_2^2 - p_1^3 \geq 0$ are easily solved, since in this case, s_1, s_2 are both real and x_1 is then the only real root.

For the case $p_2^2 - p_1^3 < 0$, s_1 can be written in terms of polar coordinates as,

$$s_1 = \sqrt{p_1}\left\{\cos\left(\frac{1}{3}\cos^{-1}\left(\frac{p_2}{\sqrt{p_1^3}}\right)\right) + i\,\sin\left(\frac{1}{3}\cos^{-1}\left(\frac{p_2}{\sqrt{p_1^3}}\right)\right)\right\}$$

Similarly for s_2. The largest angle that $\theta = \cos^{-1}\left(p_2/\sqrt{p_1^3}\right)$ can be is π, so $cos(\theta/3) \geq 0$ and $s_1 + s_2 \geq 0$, hence the largest real root is again given by x_1.