Learning and Generalization in Radial Basis Function Networks

Jason Alexis Sebastian Freeman

Ph.D.
University of Edinburgh
1997

Declaration

I declare that the thesis has been composed by myself, and that the work is my own original research except where otherwise indicated.

Jason Freeman,
27th July 1997

Publications

Material included in this thesis can also be found in the following publications:

J.A.S. Freeman and D. Saad, (1995). Learning and Generalization in Radial Basis Function Networks. *Neural Computation*, **7**, 1000-1020

J.A.S. Freeman and D. Saad, (1996). Radial Basis Function Networks: Generalization in Over-realizable and Unrealizable Scenarios. *Neural Networks*, **9**, 1521-1529

J.A.S. Freeman and D. Saad, (1997) Regularization and Realizability in Radial Basis Function Networks, *Mathematics of Neural Networks - Models, Algorithms and Applications*, Eds. S. W. Ellacott, J. C. Mason and I. J. Anderson, Bookseries on Operations Research/Computer Science Interfaces, **8**, 192-197, Kluwer Academic Publishers, Boston.

J.A.S. Freeman and D. Saad, (1997). On-line Learning in Radial Basis Function Networks. *Neural Computation*, **9**, 1601-1622

J.A.S. Freeman and D. Saad, (1997). Dynamics of On-line Learning in Radial Basis Function Networks. *Phys. Rev. E*, **56**, 907-918

J.A.S. Freeman, M.J. Orr and D. Saad, (1997). Statistical Theories of Learning in Radial Basis Function Networks. Chapter one of C.T. Leondes, editor, *Neural Network Systems, Techniques and Applications: Algorithms and Architectures.* Academic Press.

Acknowledgements

This thesis has benefitted greatly from discussions with Ansgar West, David Barber, Glenn Marion and Peter Sollich during my time as a member of the neural networks research group within the Department of Physics at Edinburgh. The support of the CNS research group, within the Centre for Cognitive Science at Edinburgh, was also invaluable, as were the cakes, coffee and occasional more potent refreshments at the weekly research group meeting!

Particular credit is due to my supervisors, David Saad and David Willshaw. David Saad's technical knowledge, enthusiasm and support were vital in the preparation of this work; whenever an avenue of research seemed to have reached an impasse, David Saad would help me find a way through. David Willshaw welcomed me into his research group when the Physics group was disbanded, providing me with excellent facilities and support. His vast range of knowledge of the neural network field proved invaluable.

My thanks to those who expertly administered the CNS computer network, particularly Andrew Gillies, and also Bruce Graham, Rosanna Maccagnano and Mark Orr. Further thanks is due to Mark for innumerable coffees, and to Rosanna and Betty Hughes for secretarial support.

In the non-academic sphere, special thanks are due to Helen Moyses and Becca Thomas for fantastic times and for keeping me focussed on the things that matter; also thanks to those with whom I lived, drank and/or had frequent stimulating discussions, etc., especially Mark Corti, Sarah Gingell, Hannibal Hills, Pete Hipwell, Sam Joseph, Alex Judge and Robin Young.

I'm especially grateful to my family for their love and support, and also for always helping me out in those moments of financial crisis...

This work was supported by the Engineering and Physical Sciences Research Council of the UK.

Abstract

The aim of supervised learning is to approximate an unknown target function by adjusting the parameters of a learning model in response to possibly noisy examples generated by the target function. The performance of the learning model at this task can be quantified by examining its generalization ability. Initially the concept of generalization is reviewed, and various methods of measuring it, such as generalization error, prediction error, PAC learning and the evidence, are discussed and the relations between them examined. Some of these relations are dependent on the architecture of the learning model.

Two architectures are prevalent in practical supervised learning: the multi-layer perceptron (MLP) and the radial basis function network (RBF). While the RBF has previously been examined from a worst-case perspective, this gives little insight into the performance and phenomena that can be expected in the *typical* case. This thesis focusses on the properties of learning and generalization that can be expected *on average* in the RBF.

There are two methods in use for training the RBF. The basis functions can be fixed in advance, utilising an unsupervised learning algorithm, or can adapt during the training process. For the case in which the basis functions are fixed, the typical generalization error given a data set of particular size is calculated by employing the Bayesian framework. The effects of noisy data and regularization are examined, the optimal settings of the parameters that control the learning process are calculated, and the consequences of a mismatch between the learning model and the data-generating mechanism are demonstrated.

The second case, in which the basis functions are adapted, is studied utilising the on-line learning paradigm. The average evolution of generalization error is calculated in a manner which allows the phenomena of the learning process, such as the specialization of the basis functions, to be eludicated. The three most important stages of training: the symmetric phase, the symmetry-breaking phase and the convergence phase, are analyzed in detail; the convergence phase analysis allows the derivation of maximal and optimal learning rates. Noise on both the inputs and outputs of the data-generating mechanism is introduced, and the consequences examined. Regularization via weight decay is also studied, as are the effects of the learning model being poorly matched to the data generator.

Contents

List of Figures

Chapter 1

Introduction

1.1 Supervised Learning in Neural Networks

The aim of supervised learning in neural networks is to approximate an unknown target mapping $f_T : X \to Y$, where X and Y represent the input and output space respectively, as closely as possible given a set of possibly noise-corrupted examples (the *training set* D) generated from f_T. To quantify the performance of a network at this task, one would ideally like to be able to measure how accurately the network reproduces the target function - this is known as *generalization ability*. From a practical perspective, generalization ability is unavailable as the target mapping is unknown, although attempts can be made to estimate it using further data generated from the target mapping. It would be very useful if it were possible to make general statements concerning the generalization ability that could be expected in the average case.

While many neural network architectures have been proposed, in the context of supervised learning there are two models which predominate: the multi-

layer perceptron (MLP) and the radial basis function network (RBF). Until recently, very little theory existed which addressed the problem of determining the properties of supervised learning for these architectures, particularly in the average case. While worst-case bounds have been determined under various limiting assumptions for both models (see, for instance, Barron, 1993; Haussler, 1994; Barron, 1994; Niyogi and Girosi, 1994), these bounds are in general insufficiently tight to be any guide to the *typical* performance that can be expected from these networks. If this were known, it would be possible to estimate the amount of data required to achieve desired performance levels, and to optimize training procedures. Further, there are many heuristic techniques that have been proposed to improve the performance of supervised neural networks. If the average-case properties could be calculated analytically, it would be possible to evaluate these heuristics in a well-founded manner, and to propose new theory-based procedures and techniques.

Several frameworks exist which facilitate the analytic investigation of the properties of supervised learning, such as the statistical physics methods (see Watkin *et al.*, 1993, for a review), the Bayesian framework (e.g. Mackay, 1992; Bishop, 1995), the PAC method (Haussler, 1994) and the Extended Bayesian Framework (Wolpert, 1996a), which is claimed by its author to subsume the others. These methods have primarily been applied to the simpler neural networks, such as linear and Boolean perceptrons, and various simplifications of the committee machine (see Nilsson, 1965; Schwarze, 1993, and references therein). It has proved very difficult to obtain results for the MLP and the RBF. One aspect of this thesis is the discovery of the typical learning properties of the RBF via the Bayesian theory.

Recently, another approach to investigating supervised learning, based on studying the dynamics of on-line gradient descent learning, has become promi-

nent. Initially employed by several authors to study learning processes primarily in the asymptotic regime, it has been successfully applied to the study of 'soft committee machines' (Saad and Solla, 1995a,b) and extended to MLPs (Riegler and Biehl, 1995). With extensions designed to take into account the nature of the RBF, it is possible to modify this approach to determine the average-case learning properties of the fully adaptive RBF; this is explored as another strand of the thesis.

One question that arises from examining these various frameworks is the precise meaning of the performance of a learning algorithm. There are various quantities that can be employed to calculate this performance, including generalization error, prediction probability, prediction error and the evidence. These measures are related in various ways; a further aim of the thesis is to elucidate these relationships in as much generality as possible.

1.2 The RBF Network

The RBF network consists of two layers of units which perform computation: an output layer and a hidden layer, and an additional input layer which plays no role beyond propagating the input vectors to the hidden layer. For simplicity, throughout the thesis the output layer will consist of a single unit (see figure 1.1). The number of units in the input layer, and hence the dimensionality of input space, is denoted by N, while the number of hidden units in the learning model is signified by K.

The units of the hidden layer have a transfer function that is radially symmetric in input space; this transfer function is constructed by considering the distance between a point in input space and the *centre* of the basis function,

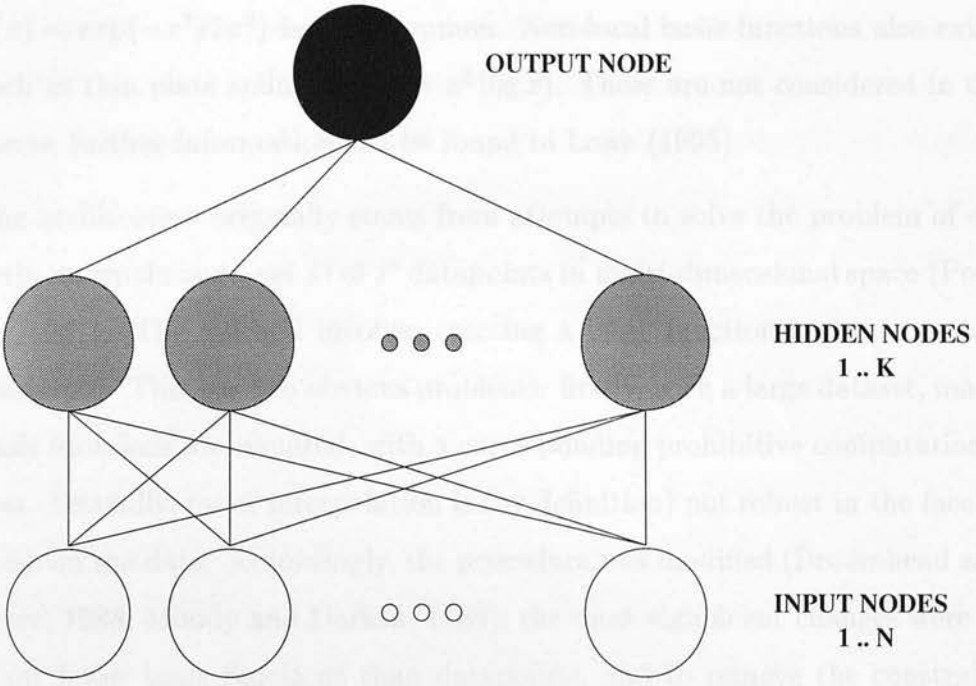


Figure 1.1: RBF network architecture. N denotes the number of input units, and the dimensionality of input space, while K signifies the number of basis functions in the learning model.

which is represented by the weights from the input layer to the hidden layer. The output layer simply performs a linear combination of the basis functions. Thus, in a general form, RBF networks perform a mapping:

$$f = \sum_b w_b \, \phi(\|\boldsymbol{m}_b - \boldsymbol{\xi}\|) \tag{1.1}$$

where w_b represents the hidden-to-output weight of basis function b, and ϕ is some function of the distance between input vector $\boldsymbol{\xi}$ and the basis function centre $\boldsymbol{m}_b$. The properties of the resulting interpolation function f are, to a large extent, not dependent on the precise form of the transfer function ϕ (Powell, 1987). Usually, localized transfer functions are used, in which $\phi(x) \to 0$ as $x \to \infty$; in particular, the Gaussian basis function

$\phi(x) = \exp(-x^2/2\sigma^2)$ is very common. Non-local basis functions also exist, such as thin plate splines ($\phi(x) = x^2 \log x$). These are not considered in the thesis; further information can be found in Lowe (1995).

The architecture originally stems from attempts to solve the problem of exactly interpolating a set D of P datapoints in multi-dimensional space (Powell, 1987). The method involves creating a basis function centred on each datapoint. This has two obvious problems: firstly, with a large dataset, many basis functions are required, with a corresponding prohibitive computational cost. Secondly, exact interpolation is (by definition) not robust in the face of noise on the data. Accordingly, the procedure was modified (Broomhead and Lowe, 1988; Moody and Darken, 1989); the most significant changes were to allow fewer basis functions than datapoints, and to remove the constraint that the basis functions are centred on datapoints. The resulting model is the one generally referred to in the context of RBF neural networks. RBFs are closely related to Parzen kernel estimators (see Scott, 1992), and can also be motivated by the theory of interpolation of noisy data (Webb, 1994), and from regularization theory (Poggio and Girosi, 1990a,b).

The RBF network is representationally powerful, being a universal approximator for continuous functions in that, given a sufficient number of hidden units, any continuous function can be approximated to desired accuracy; this is proved by Hartman *et al.* (1990) for RBFs with Gaussian hidden units, and Park and Sandberg (1993) under more general conditions on the transfer functions. It has been successfully employed in a number of real-world applications, including chaotic time-series prediction (Casdagli, 1989), speech recognition (Niranjan and Fallside, 1990) and data classification (Musavi *et al.*, 1992).

There are two commonly utilized methods for training RBFs, which are discussed in detail in chapters 3 and 5. One approach involves fixing the parameters of the hidden layer before training the hidden-to-output weights; these parameters are fixed *without regard to the target values*, relying only on the input values in the training set. This allows the use of unlabelled data[1], which other supervised learning architectures generally cannot employ. Once the hidden layer parameters are fixed, the problem is quadratic in the hidden-to-output weights and thus only requires the solution of a set of linear equations. This approach must in general result in sub-optimal solutions as the basis functions are fixed without regard to the targets. The alternative training paradigm is to adapt the hidden-layer parameters during training, either just the centre positions or both centres and widths. This renders the problem non-linear in the adaptive parameters and this requires an optimization technique, such as gradient descent, to estimate these parameters. This approach is computationally more expensive, but usually leads to greater accuracy of approximation. Obviously the first method could be used to establish a good starting position for the optimization procedure of the second.

The fact that RBF transfer functions are usually chosen to be localized gives the RBF properties that are quite distinct from the MLP. Firstly, the RBF escapes the charge often levelled at neural networks that they are uninterpretable - that it is impossible to tell what the hidden units represent. Since the area of input space covered by each basis function is known and limited, the responsibility of each hidden unit for the overall mapping is simple to determine. Localization also allows fast training: as discussed above, the centres can be fixed in position and width in advance of adapting the

[1] Unlabelled data has no associated target value.

hidden-to-output weights, or in the full supervised learning mode in which all weights are adapted, it is possible to determine efficiently which units need updating and which are not significantly affected for each datapoint (Omohundro, 1987). This has the result that, particularly in large networks, only a small fraction of hidden units need to be considered at each update step. However, localization renders the RBF susceptible to the *curse of dimensionality*: generally, given an N-dimensional input space, and given that this space can be considered to be divided into K^N hypercubes, the number of hypercubes required to fully partition the space grows exponentially in the number of dimensions. With localized basis functions, the scaling of the number of basis functions required is also exponential in N, with the result that in high-dimensional spaces, not only does the computation time become prohibitive, but the amount of data required to determine the network parameters properly also becomes very large. This makes it very important to be sure that each input dimension is relevant to the determination of the output values; further, it is possible that some of the input variables are correlated, which may allow a reduction of the dimensionality.

1.3 Structure of the Thesis

The thesis is organised as three logical units. In the first unit, consisting of chapter 2, various analytical definitions of the performance of a learning algorithm are presented, discussed and related in as much generality as possible. The second unit, encompassing chapters 3 and 4, concerns the calculation of average-case properties for RBFs in which only the hidden-to-output weights are adjustable, using the Bayesian framework. The final unit, spanning chapters 5, 6 and 7, deals with average-case analyses of RBFs in which, in addition

to the hidden-to-output weights, the positions of the centres are adaptive.

Chapter 2

Generalization

2.1 What is Generalization?

The generalization ability of a learning system is a measure of how accurately it can estimate or predict data that it has not been exposed to in the training process. Generalization ability does not exist independently of the task to be learnt; this can seem surprising, but consider the task of fitting a function to a finite set of datapoints (even without the complications of noise). In the absence of *a-priori* knowledge concerning the properties of the function underlying the data, all functions that exactly fit the data are equally valid. Note that the error made by the student on datapoints that are not in the training set (the off-training set error) is not necessarily correlated with the the performance on the training set, and, given a finite training set with inputs drawn from a space of infinite cardinality, such as the real numbers, the off-training set error is essentially equal to the error over the whole space. Further discussion of these points can be found in Wolpert (1992, 1996a,b).

Within the context of supervised learning, one is primarily interested in min-

imizing the average deviation of the estimate of the learning model from the target mapping over the entire space of possible inputs, as weighted by the measure defined over this space by the input distribution. No matter how the deviation may be defined, this quantity is termed *generalization error*; it is not available empirically with finite training data, and so must be estimated in practical use.

Analytically, generalization error can be investigated by making an assumption concerning the form of the function that is to be learnt. In general, a more specific assumption allows tighter results to be found, and vice versa. The different frameworks that exist for examining generalization error incorporate different strengths of assumptions, and thus different strengths of results: for instance, both the Bayesian approach followed in chapter 3 and the statistical mechanics approach (Watkin *et al.*, 1993) require knowledge of the input distribution, but allow average case results to be derived, while the PAC method and derivatives thereof (Haussler, 1989, 1994) (see Holden and Rayner, 1995, for an attempt to apply PAC learning to RBFs) are independent of the input distribution, but only provide weak bounds on the generalization error.

If one is unwilling to assume a functional form for the teacher, but knowledge is available concerning the conditional probability of a particular output value given an input value, then analytic properties of generalization can still be investigated by considering the *prediction error*, which is derived from the probability of the learning mechanism correctly predicting a data-point drawn from the known input-output distribution. Note that there is a many-to-one relation between the combination of teacher functional form and noise model, and the conditional probability of the output given an input; they are not equivalent.

2.2 Measuring Generalization Ability

Several different ways of measuring generalization are discussed in the following sections, including prediction probability, prediction error, generalization error and the PAC theory. The evidence (Mackay, 1992), which has been conjectured to be highly correlated with generalization ability in certain circumstances, is also examined.

2.2.1 Prediction Error

Prediction error is defined via the *prediction probability*, which is the probability that a learning system, trained on a particular set D of P examples drawn from the Cartesian product of inputspace with outputspace, $X \times Y$, according to some probability $\mathcal{P}_{X\times Y}$, will correctly predict another input-output pair, termed the *test point* T, drawn independently from that distribution. The prediction probability is defined as $\mathcal{P}(T|D)$; denoting the vector of parameters of the learning model by $\boldsymbol{w}$, the prediction probability can be written in terms of the model:

$$\mathcal{P}(T|D) = \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D)\mathcal{P}(T|\boldsymbol{w}) \tag{2.1}$$

Imposing the constraint that minimization of the training error is equivalent to maximising the likelihood of the data (Levin *et al.*, 1989) leads to the following form for the probability of the dataset given the learning model parameters and training algorithm parameters:[1]

[1]Note that, strictly, $\mathcal{P}(D|\boldsymbol{w},\beta)$ should be written $\mathcal{P}(\,(y_1,\ldots,y_P)|(\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_P),\boldsymbol{w},\beta)$ as it is desired to predict the output terms from the input terms, rather than both jointly.

$$\mathcal{P}(D|\boldsymbol{w},\beta) = \frac{\exp(-\beta E_D(\boldsymbol{w}))}{Z_D} \tag{2.2}$$

This form resembles a Gibbs distribution over the space of parameters; it also corresponds to modelling the data set as being subject to zero-mean additive Gaussian noise. The β term serves as a hyperparameter, and controls the error sensitivity - with β large, $\mathcal{P}(D|\boldsymbol{w},\beta)$ will be sharply peaked around the parameter values with lowest error, while with β small, error is tolerated to a greater extent and the distribution will be relatively spread out. $Z_D = \int_{Y^P} d^P y \, \exp\left(-\beta E_D(\boldsymbol{w})\right)$ simply normalizes the distribution.

This distribution can be realised practically by employing the Langevin training algorithm, which is simply the gradient descent algorithm with an appropriate noise term added to the weights at each update (Rögnvaldsson, 1994). Denoting standard gradient descent by the equation:

$$\Delta\boldsymbol{w} = -\eta\nabla E_D(\boldsymbol{w}) \tag{2.3}$$

where η is the learning rate, the Langevin variant is:

$$\Delta\boldsymbol{w} = -\eta\nabla E_D(\boldsymbol{w}) + \sqrt{2\eta/\beta}\vartheta \tag{2.4}$$

where ϑ is a Gaussian noise term, drawn from a distribution of zero mean and unit variance.

Furthermore, it has been shown that the gradient descent learning algorithm, considered as a stochastic process due to random order of presentation of the training data, solves a Fokker-Planck equation for which the stationary distribution can be approximated by a Gibbs distribution (Radons *et al.*,

1990).

To obtain the individual terms of equation (2.1), one can firstly apply equation (2.2) to the test point T:

$$\mathcal{P}(T|\boldsymbol{w},\beta) = \frac{\exp(-\beta E_T(\boldsymbol{w}))}{Z_T} \tag{2.5}$$

where E_T is the error on T and $Z_T = \int_Y dy\, \exp(-\beta E_T(\boldsymbol{w}))$ is the normalization.

Denoting a general prior distribution over the parameter space of the learning model W by $\mathcal{P}(\boldsymbol{w})$, the posterior distribution can be constructed:

$$\mathcal{P}(\boldsymbol{w}|D,\beta) = \frac{\mathcal{P}(\boldsymbol{w})\exp(-\beta E_D(\boldsymbol{w}))}{Z} \tag{2.6}$$

where E_D is the error on the dataset and the normalization Z is the partition function over parameter space given by $Z = \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w})\exp(-\beta E_D(\boldsymbol{w}))$.

Then prediction probability can be written as:

$$\begin{aligned} \mathcal{P}(T|D,\beta) &= \int_W d\boldsymbol{w}\, \mathcal{P}(T|\boldsymbol{w},\beta)\, \mathcal{P}(\boldsymbol{w}|D,\beta) \\ &= \frac{\int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w})\exp(-\beta E_T - \beta E_D)}{Z\, Z_T} \end{aligned} \tag{2.7}$$

Prediction error is defined as the negative log of the prediction probability: $E_{PRE} = -\log \mathcal{P}(T|D)$. This can also be interpreted as the average number of bits required to encode a novel example, given a system trained on D, so in a sense it measures the surprisingness of a new example to the system; it is connected to the Minimum Descriptive Length (see Levin *et al.*, 1989).

2.2.2 Generalization Error

There are several similar definitions of generalization error; a common theme amongst definitions is that the error is the average difference between the desired output and the estimate of the learning model: taking E_T as any measure of an error at a single point,

$$E_G = \int_X d\boldsymbol{\xi}\, \mathcal{P}(\boldsymbol{\xi}) E_T = \langle\, E_T \,\rangle \tag{2.8}$$

where $\langle \cdots \rangle$ represents an average over input space.

When utilising a stochastic training method, such as that employed in chapters 3 and 4, analytically one obtains a distribution over the space of parameters. To obtain average-case results in this situation, this distribution must be taken into account. In this case, equation (2.8) becomes:

$$E_G = \left\langle \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D) E_T \right\rangle \tag{2.9}$$

Some authors (e.g. Hansen, 1993) consider the test point to be *noisy*. Throughout this thesis, generalization error will be defined as the error between the student and the noise-free teacher, as the aim is to study how well the student emulates the *underlying* mapping. The effect of a noisy teacher is to alter the ability of the student to correctly learn the underlying mapping; there is no need to take this into account explicitly in the generalization error.

2.2.3 PAC Learning

The PAC framework, introduced by Valiant (Valiant, 1984), derives from a combination of statistical pattern recognition, decision theory and computational complexity. The basic position of PAC learning is that to learn an unknown target function successfully, an estimator should be devised which, with high probability, produces a good approximation of it, with a time complexity which is at most a polynomial function of the input dimensionality of the target function, the inverse of the accuracy required, and the inverse of the probability with which the accuracy is required. In its basic form, PAC learning deals only with two-way classification, but extensions to multiple classes and real-valued functions do exist (e.g. Haussler, 1989). PAC learning is *distribution-free*; it does not require knowledge of the input distribution, as the Bayesian framework does. The price paid for this freedom is much weaker results - the PAC framework produces worst-case results in the form of upper bounds on the generalization error, and these bounds are usually weak. It gives no insight into average-case performance of an architecture.

The basic PAC learning framework is defined as follows. A *concept class* C, is a set of subsets of input space X. Each concept $c \in C$ represents a task to be learned. A *hypothesis space* H is also a set of subsets of X, which need not equal C. For a learning model which performs a mapping $f : X \mapsto Y$, where in the simplest case of two-way classification, output space $Y = \{-1, +1\}$, a hypothesis $h \in H$ is simply the subset of X for which $f(\boldsymbol{\xi}) = +1$. Each setting of the parameters of the learning model corresponds to a function f; hence, by examining all possible parameter settings, one can associate a class of functions F with a particular model, and, through this, associate a hypothesis space with the model.

In the learning process, one is provided with a dataset D of P training examples, drawn independently from $\mathcal{P}_X$, and labelled $+1$ if the input pattern $\boldsymbol{\xi}$ is an element of concept c, and -1 otherwise. The model, during training, forms a hypothesis h via parameter adjustment, and the error of h w.r.t c is quantified as the probability of the symmetric difference Δ between c and h:

$$\text{error}(h, c) = \sum_{\boldsymbol{\xi} \in h \Delta c} \mathcal{P}_X(\boldsymbol{\xi}) \tag{2.10}$$

From this, one can define PAC learnability: the concept class C is PAC learnable by a model if, for all concepts $c \in C$ and for all distributions $\mathcal{P}_X$, it is true that when the model is given at least $p(N, \epsilon, \delta)$ training examples, where p is a polynomial, then the model can form a hypothesis h such that:

$$\Pr[\text{error}(h, c) > \epsilon] \leq \delta \tag{2.11}$$

One can think of δ as a measure of confidence, and of ϵ as an error tolerance. This is a worst-case definition, as it requires that the number of training examples must be bounded by a single fixed polynomial for all concepts $c \in C$ and all distributions $\mathcal{P}_X$. Thus, for fixed N and δ, plotting ϵ as a function of training set size gives an upper bound on *all* learning curves for the model; this bound may be very weak as compared to an average case.

The PAC framework has been extended to deal with models with a single real-valued output and adjustable hidden units (Haussler, 1994), which requires a redefinition of error as the expected *absolute* difference between the prediction of the learning model and the target. As with the basic PAC framework, results describe the worst-case scenario. The framework has also been modified by Niyogi and Girosi (1994) to make explicit the difference be-

tween errors caused by having insufficient training data and those that arise from a mismatch between the learning model and the task being learned.

2.2.4 Evidence

The *evidence* (for the hyperparameters) is defined in this context as the probability of a given dataset given certain values of the hyperparameters that control the learning process. The evidence has been postulated by Mackay (1992) to be well-correlated with generalization ability, at least when the space of possible models is appropriate to the problem in question and in the presence of sufficient data.

Although any set of hyperparameters can be employed depending on the exact type of learning system, learning will be considered to be controlled by two hyperparameters, as in Mackay (1992): γ, a parameter controlling the prior probability of a weight vector, which can be *interpreted* as a regularization parameter, although strictly this is outside the Bayesian framework from which the evidence arises, and β, which, as in section 2.2.1, is an error-sensitivity parameter.

The evidence term is defined as the probability of a dataset given the hyperparameter settings, $\mathcal{P}(D|\gamma, \beta)$, and it arises from an examination of the posterior probability of a set of learning model parameters given the dataset and the hyperparameters. Re-writing the prior over weight space in terms of γ as $\mathcal{P}(\boldsymbol{w}|\gamma)$, by Bayes' theorem:

$$\mathcal{P}(\boldsymbol{w}|D, \gamma, \beta) = \frac{\mathcal{P}(D|\boldsymbol{w}, \beta)\mathcal{P}(\boldsymbol{w}|\gamma)}{\mathcal{P}(D|\gamma, \beta)} \tag{2.12}$$

The evidence term is the normalization for the posterior, which is often

omitted as it is irrelevant to the selection of $\boldsymbol{w}$.

Recall that $\mathcal{P}(D|\boldsymbol{w},\beta) = \exp(-\beta E_D(\boldsymbol{w}))/Z_D$ and
$Z = \int d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|\gamma) \exp(-\beta E_D)(\boldsymbol{w})$. Employing a quadratic error term, Z_D becomes a Gaussian integral and thus is independent of $\boldsymbol{w}$: $Z_D = (2\pi/\beta)^{P/2}$. Using this, it is simple to obtain:

$$\mathcal{P}(D|\gamma,\beta) = \frac{Z}{Z_D} \tag{2.13}$$

Thus the evidence is proportional to the partition function over parameter space, and is therefore closely related to the free energy, $F = -(1/\beta)\log Z$, an important quantity in the statistical mechanics framework (see, for instance, Hertz *et al.*, 1989). It is of interest to relate analytically the evidence to generalization error, as certain conjectures concerning this relation have been made on intuitive grounds (MacKay, 1992).

2.3 Relating Measures of Generalization

2.3.1 Prediction Error vs Evidence

It is possible to elucidate a straightforward relationship between prediction error and evidence by exploiting the fact that the likelihood of the data has a form corresponding to a Gibbs distribution over parameter space.
Calculating the probability of a test point conditioned on the dataset by inserting the prior $\mathcal{P}(\boldsymbol{w}|\gamma)$ into equation (2.7):

$$\mathcal{P}(T|D^P,\gamma,\beta) = \frac{\int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|\gamma) \exp(-\beta E_D^P - \beta E_T)}{Z^P Z_T} \tag{2.14}$$

where the dataset D and error E_D have been explicitly labelled with the number of datapoints P, and Z^P denotes the partition function over weightspace in which the dataset has P elements.

The error function is additive, so $E_D^{P+1} = E_D^P + E_T$. The numerator of (2.14) is then the partition function over weightspace of a dataset of size $P+1$, so the prediction probability can be rewritten as:

$$\mathcal{P}(T|D^P, \gamma, \beta) = \frac{Z^{P+1}}{Z^P Z_T} \tag{2.15}$$

Eliminating the partition functions over weightspace by combining this with the definition of the evidence from equation (2.13), and recalling that Z_D is simply the normalization for the likelihood, as discussed in section 2.2.1, one obtains:

$$\mathcal{P}(T|D^P, \gamma, \beta) = \frac{\mathcal{P}(D^{P+1}|\gamma, \beta) Z_D^{P+1}}{\mathcal{P}(D^P|\gamma, \beta) Z_D^P Z_T} \tag{2.16}$$

When the test point and the dataset are drawn from the same distribution and when Z_D can be factored, such as in the case where the error measure is quadratic[2], then $Z_D^{P+1} = Z_D^P Z_T$, and:

$$\mathcal{P}(T|D^P, \gamma, \beta) = \frac{\mathcal{P}(D^{P+1}|\gamma, \beta)}{\mathcal{P}(D^P|\gamma, \beta)} \tag{2.17}$$

This equation relates prediction probability to the ratio of the evidence for a dataset of size $P+1$ to that for one of size P.

Converting the prediction probability into prediction error gives a simple

[2]In this case Z_D is a Gaussian integral.

relationship between prediction error and evidence:

$$-\log \mathcal{P}(T|D^P, \gamma, \beta) = \log \mathcal{P}(D^P|\gamma, \beta) - \log \mathcal{P}(D^{P+1}|\gamma, \beta) \tag{2.18}$$

The prediction error on predicting a new example is equal to the change in log evidence caused by adding the new example to the dataset.

2.3.2 Prediction Error vs Generalization Error

It is extremely difficult to obtain a relationship between prediction error and generalization error that is architecture-independent. Analytic considerations of generalization error rely on having or assuming information concerning the form of the function that generated the data, such as a teacher model, while prediction error is concerned with the conditional density of the output given an input.

If the learning model in question is reasonably well-trained or the error-sensitivity β is small, an equivalence of ordering relations can be developed, showing that if and only if the prediction probability for a test point T given a learning model trained on an arbitrary dataset D_1 is greater than that for the learning model trained on another arbitrary dataset D_2, then generalization error for the learning model trained on D_1 is lower than that for D_2:

$$\mathcal{P}(T|D_1) > \mathcal{P}(T|D_2) \equiv E_G(D_1) < E_G(D_2) \tag{2.19}$$

Making explicit the dependence of prediction probability on the parameters of the learning system:

$$\mathcal{P}(T|D) = \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D)\mathcal{P}(T|\boldsymbol{w}) \tag{2.20}$$

where $\mathcal{P}(T|\boldsymbol{w}) = \exp(-\beta E_T(\boldsymbol{w}))/Z_T$.

With a reasonably well-trained model, or with the error-sensitivity β small, the term βE_T will be small for all weight vectors with significant probability, and thus $\exp(-\beta E_T) \simeq 1 - \beta E_T$. The relation $\mathcal{P}(T|D_1) > \mathcal{P}(T|D_1)$ becomes:

$$\mathcal{P}(T|D_1) > \mathcal{P}(T|D_2) \equiv \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D_1)E_T < \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D_2)E_T \tag{2.21}$$

Since the test point was arbitrary, this proves eqn.(2.19).

2.3.3 Generalization Error vs Evidence for the RBF

It is possible in some cases to relate generalization error to the evidence, and hence to prediction error and prediction probability, if one has knowledge of the data-generating mechanism. In Bruce and Saad (1994), this is performed for the case of a perceptron student learning a noise-corrupted perceptron teacher. In the course of calculating average-case generalization error for the RBF in chapter 3, an analytic relation between generalization error, the evidence and prediction error will be constructed for that architecture.

Chapter 3

Stochastic Learning

This chapter investigates *average case* generalization ability for the RBF architecture, utilising the Bayesian approach in which a probability distribution is constructed over the space of possible weights of the network, conditioned on the dataset and the parameters that control the learning process. During the course of the investigation, generalization error is analytically related to the evidence, and thereby to prediction error.

Analytic investigations of generalisation error which focus on average-case results have primarily considered the one-layer perceptron, either in boolean or linear form (Bruce and Saad, 1994), and on simple extensions of this, such as the committee machine (see, for instance, Schwarze (1993)), as these architectures are analytically tractable unlike the general multi-layer perceptron.

Generalization error for the RBF has been considered analytically from a *worst-case* perspective by several authors: Niyogi and Girosi (1994) derive a bound on generalization error under the assumption that the training algorithm always finds a globally optimal solution, but require only weak constraints on the function that generated the training set; they do not consider

regularization. The paper also contains an extensive bibliography pertaining to the topic of generalization. Haussler (1994) finds worst case bounds by employing the PAC framework (see section 2.2.3). Some empirical studies also exist; for instance, Botros and Atkeson (1991) compare the performance of various choices for the basis functions.

3.1 RBF Architecture and Training Methodology

The RBF architecture consists of a two-layer fully-connected network (see figure 1.1). Each hidden node is parameterised by two quantities: a centre $\boldsymbol{m}$ in input space, corresponding to the vector defined by the weights between the node and the input nodes, and a width σ_B^2.

The role of the hidden units is to perform a non-linear transformation of the input space into the space of activations of the hidden units; it is this transformation that gives the RBF a much greater representational power than the linear perceptron. The output layer computes a linear combination of the activations of the basis functions; to simplify the analyses in the thesis, a single output node is utilised, parameterised by the weight vector $\boldsymbol{w}$ between hidden and output layers.

Within this model, the basis functions are taken to be Gaussian; each hidden node has an identical width σ_B^2 corresponding to the variance of the Gaussian. The overall function f_S computed by the network is therefore:

$$f_S(\boldsymbol{\xi}) = \sum_{b=1}^{K} w_b \exp\left(-\frac{\|\boldsymbol{\xi} - \boldsymbol{m}_b\|^2}{2\sigma_B^2}\right) = \boldsymbol{w} \cdot \boldsymbol{s}(\boldsymbol{\xi}) \tag{3.1}$$

where $s(\xi)$ denotes the vector of responses of the hidden units to the input vector ξ.

One typical training methodology employed for the RBF is to fix the parameters of the first layer utilising some algorithm to ensure that the positions of the training data in input space are adequately represented by the basis functions, and then either to solve a system of linear equations or use some training algorithm such as gradient descent to set the parameters of the second layer. Training is computationally inexpensive as compared to multi-layer perceptrons.

There have been many schemes proposed for setting the parameters of the hidden units. These methods are usually unsupervised; they pay attention only to the input values of the data, and ignore the output values. Thus the problem is really the same as mixture density estimation. Note that there is no guarantee that modelling the input distribution will be useful for the task of modelling the input-output mapping; an optimal procedure for this task must set the hidden unit parameters with regard to the output values.

The simplest scheme is simply to set the basis function centres to a random subset of the input vectors in the training set. While extremely fast, this method is crude and usually leads to the use of a large number of basis functions to give reasonable performance at the function approximation task. A more refined method is forward selection (Rawlings, 1988), in which one starts with an empty set of basis functions, and then continues adding the most explanatory basis function until a heuristic stopping criterion is met. This method is applied to RBFs in (Chen *et al.*, 1989, 1991), employing an efficient implementation termed orthogonal least squares. In (Orr, 1993), the method is refined to include a more principled stopping criterion; in the case

of (Chen *et al.*, 1989, 1991), this is simply a threshold on the proportion of variance that the RBF explains. Further improvements and variations are introduced in (Chen *et al.*, 1996) and (Orr, 1995). The alternative path may also be employed, which begins with a basis function on each datapoint, and reduces the number so as to affect the network performance as little as possible (Devijver, 1982; Fununaga and Hayes, 1989).

Instead of being constrained to place basis functions on the datapoint input values, a clustering algorithm can be employed. K-means clustering, applied to the RBF by Moody and Darken (1989), partitions the data set into K disjoint subsets where similar vectors are represented by a single centre. Kohonen maps (Kohonen, 1982) have also been utilised. A more principled method of performing density estimation is the Gaussian mixture model, in which it is assumed that the input distribution was generated by a weighted mixture of Gaussians. The parameters of the mixture model can be estimated by a non-linear optimization technique, such as the EM algorithm (Dempster *et al.*, 1977).

In this chapter the hidden units have centres that are presumed to be fixed in place by a suitable process as described above; there is a single output unit with a vector of adjustable weights.

3.1.1 Data Generation

The training data D consists of P input-output pairs indexed $1 \ldots P$: $(\boldsymbol{\xi}_p, y_p)$. In order to have full control over the task to be learned, the data is assumed to be generated by a teacher RBF, and then corrupted under some noise process, with the N-dimensional input vectors being drawn from a symmetric Gaussian distribution of variance σ_ξ^2. The teacher consists of M centres each

with a position vector $\boldsymbol{n}_u$ and an identical width σ_{Bt}, while the student consists of K centres with position vector $\boldsymbol{m}_i$ and width σ_B. In general, the centres of the teacher need not correspond in position, number or width to those of the student, which implies that the learning problem may not be realizable. In this chapter, the teacher has a set of centres identical to those of the student (this is relaxed in chapter 4). Thus in the language of learning theory (Niyogi and Girosi, 1994), the approximation error is zero, and generalization error is equivalent to estimation error. The weight vector of the teacher output node is denoted by $\boldsymbol{w}^0$, so the teacher computes:

$$f_T(\boldsymbol{\xi}) = \sum_{u=1}^{M} w_u^0 \exp\left(-\frac{\|\boldsymbol{\xi} - \boldsymbol{n}_u\|^2}{2\sigma_{Bt}^2}\right) = \boldsymbol{w}^0 \cdot \boldsymbol{t}(\boldsymbol{\xi}) \tag{3.2}$$

where $\boldsymbol{t}(\boldsymbol{\xi})$ represents the vector of responses of the teacher hidden units to the input vector $\boldsymbol{\xi}$.

3.1.2 The Training Algorithm

The training algorithm for the weights that impinge on the student output node is considered stochastic in nature; modelling the noise process as zero-mean additive Gaussian noise leads to the following form for the probability of the dataset given the weights and training algorithm parameters, as discussed in section 2.2.1:

$$\mathcal{P}(D|\boldsymbol{w}, \beta) = \frac{\exp(-\beta E_D(\boldsymbol{w}))}{Z_D} \tag{3.3}$$

Labelling the noise on example p as ϑ_p, the data is therefore generated by: $y_p = f_T(\boldsymbol{\xi}_p) + \vartheta_p$.

To prevent over-dependence of the distribution of student weight vectors on the details of the noise, it is necessary to introduce a regularising factor, which can be defined in terms of the student weights, as a prior distribution over weight space:

$$\mathcal{P}(\mathbf{w}|\gamma) = \frac{\exp(-\gamma E_W)}{Z_W} \tag{3.4}$$

where E_W is a penalty term based, for instance, on the magnitude of the student weight vector[1] and $Z_W = \int_W d\boldsymbol{w} \exp(-\gamma E_W)$.

Employing Bayes' theorem, one can derive an expression for the probability of a student weight vector given the training data and training algorithm parameters (this is a specialization of equation (2.12)):

$$\begin{aligned} \mathcal{P}(\boldsymbol{w}|D,\gamma,\beta) &= \frac{\mathcal{P}(D|\boldsymbol{w},\beta)\,\mathcal{P}(\boldsymbol{w}|\gamma)}{\mathcal{P}(D|\gamma,\beta)} \\ &= \frac{\exp\left(-\beta E_D - \gamma E_W\right)}{Z_M} \end{aligned} \tag{3.5}$$

Here, $Z_M = \int d\boldsymbol{w} \exp(-\beta E_D - \gamma E_W)$ is the partition function over student space.

3.2 Generalization Error

Following section 2.2.2, generalization error E_G will be defined as the average error between the desired and actual network output. The square of the

[1]Note that for the ubiquitous $E_W = \frac{1}{2}\|\boldsymbol{w}\|^2$ penalty term, $Z_W = (2\pi/\gamma)^{\frac{K}{2}}$.

difference between desired and actual output is the typical error measure employed, which for a particular student network gives[2]:

$$E = \left\langle \left(f_T(\boldsymbol{\xi}) - f_S(\boldsymbol{\xi}) \right)^2 \right\rangle \tag{3.6}$$

From a practical viewpoint, one only has access to the empirical risk, or test error, which is the mean-sum-squared error on a set of points not employed during training. This quantity is an approximation to the expected risk, defined as the expectation of $(y - f_S(\boldsymbol{\xi}))^2$ with respect to the joint distribution $\mathcal{P}(\boldsymbol{\xi}, y)$. With an additive noise model, the expected risk simply decomposes to $E + \sigma^2$, where σ^2 is the variance of the noise. Some authors equate the expected risk with generalization error by considering the squared difference between the *noisy* teacher and the student (see, for instance, Hansen, 1993). A more detailed discussion of these quantities can be found in (Niyogi and Girosi, 1994).

If a stochastic training algorithm is employed, such as the Langevin variant of gradient descent, the resulting probability distribution over weight space (conditioned on the training data) must be taken into account. Two possibilities for average generalization error arise. If, as is usually the case practically, the algorithm selects a single weight vector from the ensemble, a procedure which will be termed Gibbs learning, then equation (3.6) becomes[3]:

[2]This definition is equivalent to the distance in the $L^2(\mathcal{P})$ norm between $f_T(\boldsymbol{\xi})$ and $f_S(\boldsymbol{\xi})$, where $L^2(\mathcal{P})$ is the set of functions whose square is integrable with respect to the measure defined by $\mathcal{P}$.

[3]It is worth noting that by taking $\frac{\beta}{\gamma} \to \infty$, the distribution of student weight vectors becomes a delta function centred on the weight vector that minimises the empirical risk. This situation is commonly considered in the computational learning theory literature, but is unrealistic for neural networks, where often only locally optimal solutions are found in practice.

$$E_G = \left\langle \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D,\gamma,\beta)\, \left(f_T(\boldsymbol{\xi}) - f_S(\boldsymbol{\xi})\right)^2 \right\rangle \tag{3.7}$$

Note that in order to obtain average-case results, an average over the posterior weight distribution in included in the definition of E_G.

A second possibility arises from considering a Bayes-optimal approach. This requires one to take the expectation of the estimate of the network, which is impractical due to the computation involved, but can be approximated by Monte-Carlo methods (Neal, 1992), or more crudely by performing a succession of training runs:

$$E_B = \left\langle \left(f_T(\boldsymbol{\xi}) - \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D,\gamma,\beta)\, f_S(\boldsymbol{\xi}) \right)^2 \right\rangle \tag{3.8}$$

These two quantities are related by:

$$\begin{aligned} E_G - E_B &= \left\langle \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D,\gamma,\beta)\, f_S(\boldsymbol{\xi})^2 - \left(\int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D,\gamma,\beta)\, f_S(\boldsymbol{\xi}) \right)^2 \right\rangle \\ &= \mathrm{Var}\left(f_S(\boldsymbol{\xi})\right) \end{aligned} \tag{3.9}$$

where $\mathrm{Var}\left(\cdots\right)$ is the variance with respect to the posterior distribution.

In order to investigate the generic performance of the architecture, it is desirable to eliminate the dependence of generalization error on the particular data-set used. An average over possible data-sets, denoted by $\ll \cdots \gg$, will be utilised for this purpose. Thus, with additive Gaussian noise ϑ on the data, one obtains:

$$\ll E \gg = \int_{X^P} d^P\boldsymbol{\xi}\, \mathcal{P}(\boldsymbol{\xi}_p) \int_{\vartheta^P} d^P\vartheta\, \mathcal{P}(\vartheta_p) \left\langle \left(f_T(\boldsymbol{\xi}) - f_S(\boldsymbol{\xi})\right)^2 \right\rangle \tag{3.10}$$

3.3 Calculation of Generalization Error

The calculation of generalization error will focus on both E_G and E_B; a link to prediction error is developed via an analytic relation between E_G and the evidence. Initially E_G is found, as E_B can be derived easily once E_G is known.

Recalling that the teacher centres are equal in number and position to those of the student and signifying the difference between student and teacher weight vectors, $\boldsymbol{w} - \boldsymbol{w}^0$, by $\boldsymbol{w}^*$, the definition of E_G becomes:

$$E_G = \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D, \gamma, \beta) \left\langle (\boldsymbol{w}^* \cdot \boldsymbol{s})^2 \right\rangle \tag{3.11}$$

Since $\boldsymbol{w}^*$ is independent of the input distribution,

$$E_G = \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D, \gamma, \beta)\, \boldsymbol{w}^{*T} \boldsymbol{G} \boldsymbol{w}^* \tag{3.12}$$

where $\boldsymbol{G} = \left\langle \boldsymbol{s}\, \boldsymbol{s}^T \right\rangle$ is a matrix describing the average responses of pairs of student basis functions to an input point. Taking the input vectors to be drawn from a symmetric Gaussian distribution with mean $\mathbf{0}$, variance σ_ξ^2, allows $\boldsymbol{G}$ to be calculated explicitly; the full expression is given in appendix A.

Employing the definition of $\mathcal{P}(\boldsymbol{w}|D, \gamma, \beta)$ as in eqn.(3.5), taking E_D as sum-squared training error with ϑ_p as the noise on training example p, and defining

$E_W = \frac{1}{2}\|\boldsymbol{w}\|^2$ as the prior over weight space allows eqn.(3.5) to be re-written as:

$$\mathcal{P}(\boldsymbol{w}|D,\gamma,\beta) = \frac{\exp(-\frac{P}{2}\boldsymbol{w}^{*T}\boldsymbol{\Lambda}^{-1}\boldsymbol{w}^* - \boldsymbol{w}^{*T}\boldsymbol{\rho} - \frac{\beta}{2}\sum_p \vartheta_p^2 - \frac{\gamma}{2}\|\boldsymbol{w}^0\|^2)}{Z_M} \tag{3.13}$$

where:

$$\begin{aligned} \boldsymbol{\Lambda}^{-1} &= \frac{\gamma}{P}\mathbf{I} + \frac{\beta}{P}\sum_p \tilde{\boldsymbol{s}}_p\tilde{\boldsymbol{s}}_p^T \\ \boldsymbol{\rho} &= \gamma\boldsymbol{w}^0 + \beta\sum_p \vartheta_p\tilde{\boldsymbol{s}}_p \end{aligned}$$

with $\tilde{\boldsymbol{s}}_p$ being the vector of responses of the student basis functions to data-point p.

At this point one can proceed in two ways: E_G can be found directly via eqn. (3.12) by integrating over the posterior distribution, or a more circuitous route can be followed which relates E_G to the evidence and prediction error in passing. Since it is much simpler, E_G will ultimately be found by direct integration, but first it will be related to the evidence.

Substituting eqn. (3.13) into eqn. (3.12), and rewriting by substituting into the resulting equation the derivative of the numerator of eqn. (3.13) with respect to the elements of the matrix $\boldsymbol{\Lambda}^{-1}$, E_G becomes:

$$E_G = -\frac{2}{P}\sum_{bc} G_{bc}\frac{1}{Z_M}\frac{\partial}{\partial\Lambda_{bc}^{-1}}\left[\int_W d\boldsymbol{w}\exp(-\gamma E_W - \beta E_D)\right] \tag{3.14}$$

The evidence is proportional to the modified partition function Z_M, so one

can immediately relate the evidence to the generalization error:

$$E_G = -\frac{2}{P}\sum_{bc} G_{bc}\frac{\partial}{\partial \Lambda_{bc}^{-1}}\left[\log \mathcal{P}(D|\gamma,\beta)\right] \tag{3.15}$$

At this point it is also possible to relate generalization error to prediction error. It is relatively simple to derive the relationship between prediction error and evidence (see section 2.3.1):

$$\log \mathcal{P}(y|\boldsymbol{\xi}, D, \gamma, \beta) = \log \mathcal{P}(D, y|\boldsymbol{\xi}, \gamma, \beta) - \log \mathcal{P}(D|\gamma,\beta) \tag{3.16}$$

Employing this relationship in equation (3.15), one arrives at:

$$E_G = -\frac{2}{P}\sum_{bc} G_{bc}\frac{\partial}{\partial \Lambda_{bc}^{-1}}\left[\log \mathcal{P}(D, y|\boldsymbol{\xi}, \gamma, \beta)\right] - \frac{\partial}{\partial \Lambda_{bc}^{-1}}\left[\log \mathcal{P}(y|\boldsymbol{\xi}, D, \gamma, \beta)\right] \tag{3.17}$$

These relations are not immediately intuitive, but it is possible to write the evidence in terms of $\boldsymbol{\Lambda}$ and some constants: recalling that the evidence is proportional to Z_M and rewriting Z_M in a manner similar to 3.13:

$$\begin{aligned} \log Z_M \;=\; & -\frac{1}{2}\log\det \boldsymbol{\Lambda}^{-1} + \frac{1}{2P}\boldsymbol{\rho}^T\boldsymbol{\Lambda}\boldsymbol{\rho} + \\ & \frac{K}{2}\log\frac{2\pi}{P} - \frac{\beta}{2}\sum_p \vartheta_p^2 - \frac{\gamma}{2}\|\boldsymbol{w}^0\|^2 \end{aligned} \tag{3.18}$$

One could then substitute this expression into eqn. (3.15) and find the derivatives analytically, but as discussed, it is simpler to find generalization error by integrating directly over the posterior. Substituting the expression for the

posterior, eqn. (3.13), into that for E_G, eqn. (3.12), and integrating:

$$E_G = \frac{\operatorname{tr} \boldsymbol{G\Lambda}}{P} + \frac{\boldsymbol{\rho}^T \boldsymbol{\Lambda G \Lambda \rho}}{P^2} \tag{3.19}$$

It remains to consider the average $\ll \cdots \gg$ over datasets and the Gaussian noise on the datasets. Performing the noise average, recalling that only $\boldsymbol{\rho}$ contains noise terms:

$$\int d^P \vartheta \, \mathcal{P}(\vartheta_p) \boldsymbol{\rho \Lambda G \Lambda \rho} = \tag{3.20}$$
$$\sum_{de} (\Lambda G \Lambda)_{de} \left(\gamma^2 w_d^0 w_e^0 + \beta^2 \sigma^2 \sum_p \tilde{s}_d^p \tilde{s}_e^p \right)$$

To progress further and perform the dataset average, it is necessary to know the form of $\boldsymbol{\Lambda}$. To this end, it will be assumed that $\boldsymbol{\Lambda}^{-1}$ is of the form:

$$\begin{bmatrix} \theta & \tilde{\theta} & \cdots & \tilde{\theta} \\ \tilde{\theta} & \theta & \cdots & \tilde{\theta} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\theta} & \tilde{\theta} & \cdots & \theta \end{bmatrix} \tag{3.21}$$

That is, all diagonal entries are equal to θ, and all off-diagonal entries are equal to $\tilde{\theta}$.

This induces $\boldsymbol{\Lambda}$ to take on the form:

$$\begin{bmatrix} \psi & \tilde{\psi} & \cdots & \tilde{\psi} \\ \tilde{\psi} & \psi & \cdots & \tilde{\psi} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\psi} & \tilde{\psi} & \cdots & \psi \end{bmatrix} \tag{3.22}$$

where:

$$\begin{aligned} \psi &= \frac{1+\phi}{\theta - \tilde{\theta}} \\ \tilde{\psi} &= \frac{\phi}{(\theta - \tilde{\theta})} \\ \phi &= -\frac{\tilde{\theta}}{\theta + \tilde{\theta}(K-1)} \end{aligned}$$

The implications of this assumption for the RBF model are twofold: firstly, the equality of diagonal entries corresponds to all the centres receiving an equal amount of activation via the training data[4]. For the particular case of a symmetric input distribution centred at the origin of input space, this assumption breaks down only for the case in which the centres are dissimilar in distance from the origin *and* the variance of the input distribution is not of sufficient magnitude for the distribution to be approximately uniform in the regions covered by the basis functions. Secondly, the equality of off-diagonal entries requires each pair of basis functions to receive a similar joint activation via the training data. This assumption is satisfied except for the case in which the centres are not approximately equidistant from each other

[4] A common procedure for selecting basis function parameters is to maximise the likelihood of the inputs of the training data under a mixture model given by a linear combination of the basis functions; constraining the priors of the mixture model to be equal encourages this property of equal activation to be satisfied.

and the spread of the basis functions is not sufficient to allow considerable overlap between each pair of receptive fields to occur.

Unfortunately, this selection of form for $\mathbf{\Lambda}^{-1}$ is not sufficient to allow the dataset average to be carried out, as the $\boldsymbol{\xi}_p$ terms do not separate into independent factors. One can approximate $\mathbf{\Lambda}^{-1}$ as:

$$\begin{aligned} \mathbf{\Lambda}^{-1} &= \frac{\gamma}{P}\mathbf{I} + \frac{\beta}{P}\sum_p \tilde{\boldsymbol{s}}_p \tilde{\boldsymbol{s}}_p^T \\ &\simeq \frac{\gamma}{P}\mathbf{I} + \beta\frac{1}{P} \ll \sum_p \tilde{\boldsymbol{s}}_p \tilde{\boldsymbol{s}}_p^T \gg \\ &= \frac{\gamma}{P}\mathbf{I} + \beta \boldsymbol{G} \end{aligned} \tag{3.23}$$

Utilising the central limit theorem, the neglected variance in the distribution of $\frac{1}{P}\sum_p \tilde{\boldsymbol{s}}_p \tilde{\boldsymbol{s}}_p^T$ decreases as $\frac{1}{P}$. Note that this implies that the calculation of generalization error holds strictly only in the asymptotic regime of large P, but it will be shown in chapter 4 via simulations that the results are a very good approximation for non-asymptotic P.

The integral over datasets can now be performed as a straightforward Gaussian, yielding the final expression for generalization error:

$$\ll E_G \gg = \frac{1}{P}\left(\operatorname{tr} \boldsymbol{G}\mathbf{\Lambda} + \frac{1}{P} \operatorname{tr} \mathbf{\Lambda}\boldsymbol{G}\mathbf{\Lambda}\mathbf{\Upsilon} \right) \tag{3.24}$$

where, for notational convenience, the matrix defined by $\Upsilon_{bc} = \gamma^2 w_b^0 w_c^0 + \beta^2\sigma^2 P G_{bc}$ has been introduced.

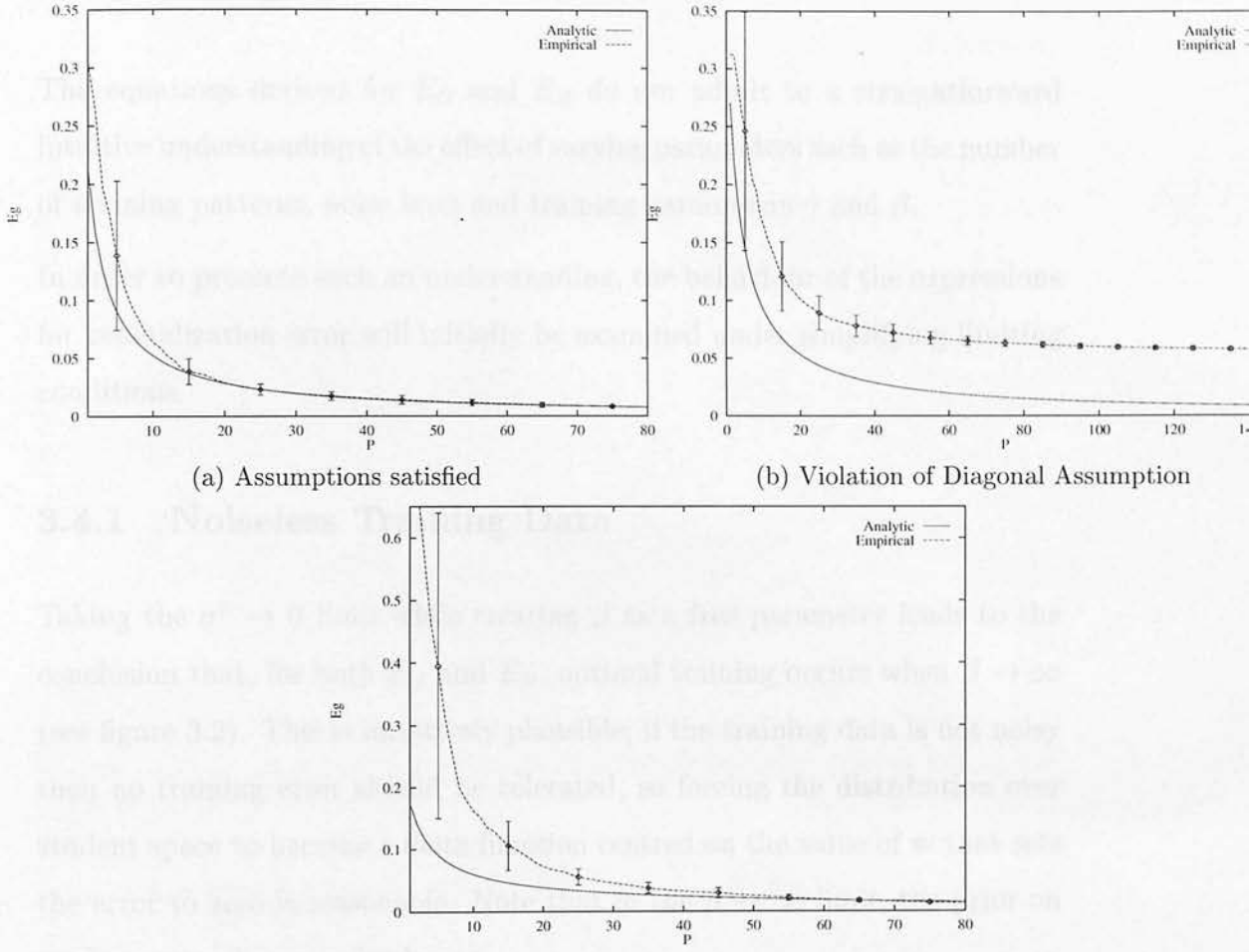
From this, via equation (3.9), one can calculate $\ll E_B \gg$:

$$\ll E_B \gg = \frac{1}{P^2}(\operatorname{tr} \boldsymbol{\Lambda G \Lambda \Upsilon}) \tag{3.25}$$

To examine the validity of the assumptions for $\boldsymbol{\Lambda}^{-1}$, simulations were conducted in which the empirical value of E_G was calculated via equation (3.20) by generating random training data and numerically evaluating $\boldsymbol{\Lambda}$. The simulations were carried out for three scenarios: firstly, the case in which the conditions for the assumption of form of $\boldsymbol{\Lambda}^{-1}$ were exactly satisfied; secondly, for certain basis functions receiving an impoverished supply of training data, thus violating the equality of diagonal entries; finally, for the interactions between different pairs of basis functions being unequal, which violates the equality of off-diagonal entries.

Comparisons of the mean values of E_G found by simulation, E_G^{SIM}, with those found analytically via equation (3.24) are shown in figure 3.1. Note that the variances of the simulation distributions quickly become negligible.

When the assumptions are satisfied, E_G^{SIM} rapidly converges to E_G. Violation of the assumption of diagonal equality gives rise to a systematic error, while violation of the off-diagonal assumption causes the convergence to slow, but introduces negligible systematic error. This lack of significant effect is explicable by an examination of the expression for $\boldsymbol{G}$ (eqn.A.3): the result of introducing differing interactions between the basis functions is simply to vary $\|\boldsymbol{m}_b + \boldsymbol{m}_c\|$; the effect of this will always be overwhelmed by that of other terms, particularly if the ratio of σ_B^2 to σ_ξ^2 is large. It can be concluded, therefore, that the calculation of generalization error is invalid only for the cases in which P is near to 0 or in which the basis functions receive significantly different levels of activation via the training data.



(a) Assumptions satisfied

(b) Violation of Diagonal Assumption

(c) Violation of Off-Diagonal Assumption

Figure 3.1: Analytic E_G (unbroken line) versus mean of E_G^{SIM} (dashed line) examining the validity of the assumption of form for $\mathbf{\Lambda}^{-1}$ under various distributions of the centres of the basis functions. The error bars are plotted at 1 standard deviation of the simulation mean. Each simulation was run 50 times with the following parameter settings (denoting the angle between $\boldsymbol{m}_b$ and $\boldsymbol{m}_c$ as $\Theta_{b,c}$):
Common to all simulations (see section 3.1 for a reminder of symbol definitions): $N = 3, K = M = 4, \sigma^2 = 1, \beta = 0.5, \gamma = 1, \sigma_\xi^2 = 2, \sigma_B^2 = 1$
Assumptions satisfied: $\forall_b$: $\|\boldsymbol{m}_b\| = 1, \forall_{b,c:b\neq c}$: $\Theta_{b,c} = 2\pi/3$
Diagonal violation: $\|\boldsymbol{m}_1\| = \|\boldsymbol{m}_2\| = 1$, $\|\boldsymbol{m}_3\| = \|\boldsymbol{m}_4\| = 4$, $\forall_{b,c:b\neq c}$: $\Theta_{b,c} = 2\pi/3$
Off-diagonal violation: $\forall_b$: $\|\boldsymbol{m}_b\| = 1$, $\Theta_{1,2} = \Theta_{3,4} = \pi/6$, $\Theta_{1,4} = \Theta_{2,3} = \pi$

3.4 Analysis of Generalization Error

The equations derived for E_G and E_B do not admit to a straightforward intuitive understanding of the effect of varying parameters such as the number of training patterns, noise level and training parameters γ and β.

In order to promote such an understanding, the behaviour of the expressions for generalization error will initially be examined under simplifying limiting conditions.

3.4.1 Noiseless Training Data

Taking the $\sigma^2 \to 0$ limit while treating β as a free parameter leads to the conclusion that, for both E_G and E_B, optimal training occurs when $\beta \to \infty$ (see figure 3.2). This is intuitively plausible; if the training data is not noisy then no training error should be tolerated, so forcing the distribution over student space to become a delta function centred on the value of $\mathbf{w}$ that sets the error to zero is reasonable. Note that in the $\beta \to \infty$ limit, the prior on student space becomes irrelevant.

3.4.2 No Weight Decay: the $\gamma \to 0$ limit

Considering the $\gamma \to 0$ limit allows one to analyse the dependence of E_G and E_B on the number of training examples, P. The assumption of the diagonal versus off-diagonal form for $\mathbf{\Lambda}^{-1}$ induces a similar form on the matrix $\boldsymbol{G}$; the diagonal and off-diagonal elements of $\boldsymbol{G}$ will be referenced by G_D and G_O respectively.

Proceeding from the final expression for generalization error, eqn. (3.24), the

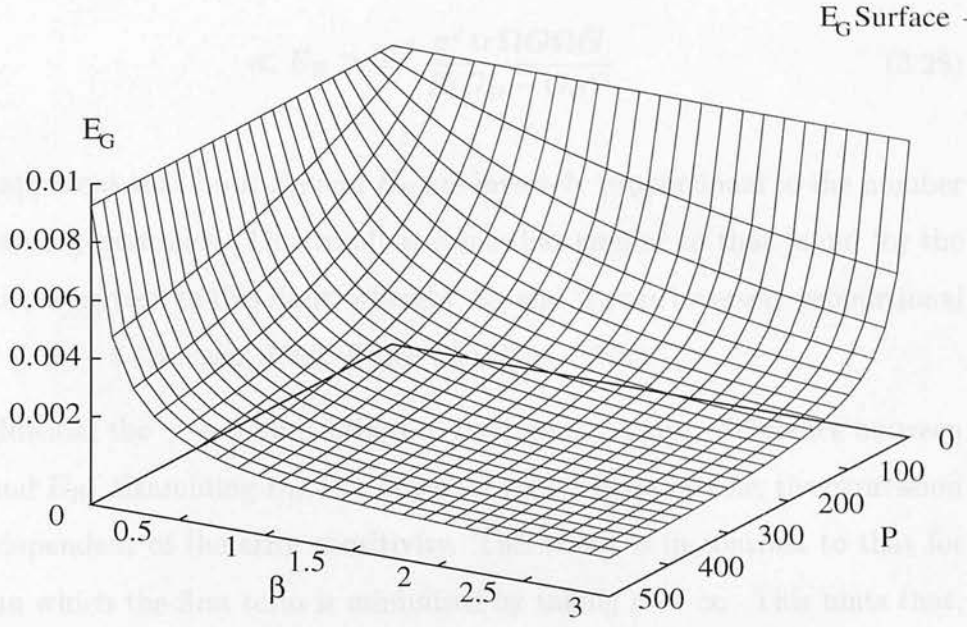


Figure 3.2: E_G as a function of number of examples P and error sensitivity β for $\sigma^2 \to 0$

form of $\mathbf{\Lambda}$ is known from eqn. (3.21). Thus the $\gamma \to 0$ limit of Λ can be found: defining, for notational convenience, the matrix $\mathbf{\Omega}$ by $\Omega_{bc} = \delta_{bc} - \frac{G_O}{G_D + G_O(K-1)}$,

$$\lim_{\gamma \to 0} \mathbf{\Lambda} = \frac{\mathbf{\Omega}}{\beta(G_D - G_O)} \tag{3.26}$$

then, straightforwardly, $\lim_{\gamma \to 0} \ll E_G \gg$ and $\lim_{\gamma \to 0} \ll E_B \gg$ can be found:

$$\ll E_G \gg \overset{\gamma \to 0}{=} \frac{\operatorname{tr} \boldsymbol{G\Omega}}{\beta P(G_D - G_O)} + \frac{\sigma^2 \operatorname{tr} \boldsymbol{\Omega G \Omega G}}{P(G_D - G_O)^2} \tag{3.27}$$

and

$$\ll E_B \gg \overset{\gamma \to 0}{=} \frac{\sigma^2 \operatorname{tr} \boldsymbol{\Omega G \Omega G}}{P(G_D - G_O)^2} \tag{3.28}$$

It is apparent that both E_G and E_B are inversely proportional to the number of training examples. This result is somewhat similar to that found for the linear perceptron in this limit, whereby E_G and E_B are inversely proportional to $P - N - 1$ (Hansen, 1993; Bruce and Saad, 1994).

In addition, the $\gamma \to 0$ limit brings to light an interesting difference between E_G and E_B. Examining E_B, it is apparent that β plays no role; the expression is independent of the error sensitivity. This result is in contrast to that for E_G, in which the first term is minimised by taking $\beta \to \infty$. This hints that, in the Bayes generalizer, it is only the *ratio* of γ to β that is important, as is the case for the linear perceptron (Bruce and Saad, 1994), while the Gibbs generalizer is dependent on both β and γ separately. This discrepancy is explicated by recalling equation (3.9); E_G consists of a term due to the variance of the student output with respect to the posterior, minimised by taking $\beta \to \infty$, and a term identical to E_B.

Both E_G and E_B are independent of N, the dimensionality of input space, in this limit.

3.4.3 The General Case: Noise and Weight Decay

To gain some understanding of the variation of E_G and E_B with P, γ and β in the general case, consider figures 3.3, 3.4, 3.5 and 3.6.

Examining first figure 3.3, in which E_B is plotted against P and β for a constant value of γ, it is apparent that there is a minimum in the generalization

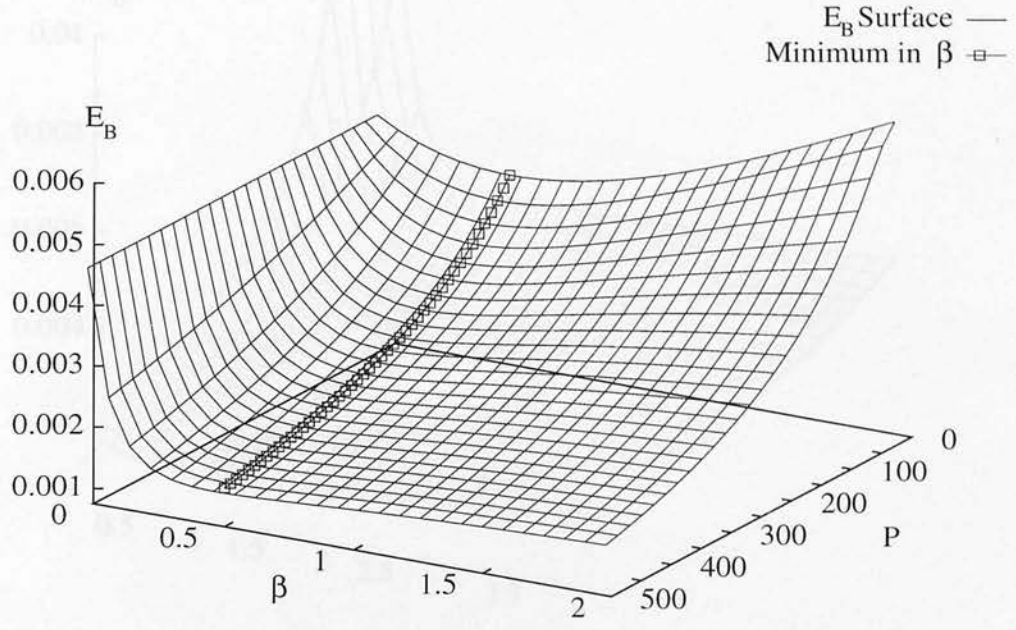


Figure 3.3: Generalization error E_B as a function of number of examples P and error sensitivity β. The minimum in E_B with respect to β is independent of P.

error surface at a constant value of β. When γ is set to its optimal value, the value of β at the minimum can be shown empirically to be inversely proportional to the variance of the noise, σ^2. Similarly, plotting E_B against P and γ (figure 3.4) demonstrates a minimum in the generalization error surface at a constant value of γ. This minimum, for β set to an optimal value, is a function of both $\|\boldsymbol{w}^0\|^2$ and $\sum_{bc} w_b^0 w_c^0$.

An entirely different pattern of results emerges for E_G. Considering figure 3.5, the optimal value of β rapidly becomes infinite as P increases. This is due to the fact that the Gibbs generalizer requires the selection of a single weight vector from the ensemble of students, so it is advantageous to penalise any training error maximally once a reasonable amount of training data is

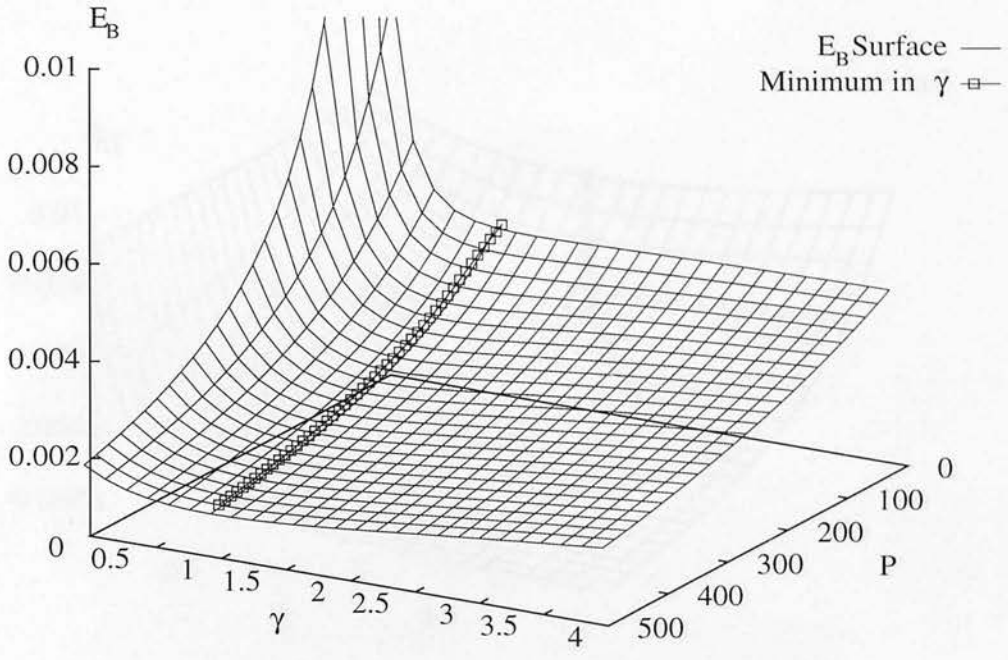


Figure 3.4: Generalization error E_B as a function of number of examples P and weight decay parameter γ. The minimum in E_B with respect to γ is independent of P.

available. The Bayes generalizer, on the other hand, employs a weighted average of students in order to make a prediction; noise on the training data output values can to some extent be compensated for by this average, and so it is not desirable to force the ensemble to become a delta function. Focussing on E_G as a function of P and γ (figure 3.6), an analogous result is apparent: the optimal value of γ is initially infinite, but as $P \to \infty$, the optimal value of γ tends to an expression similar in dependence to that for E_B.

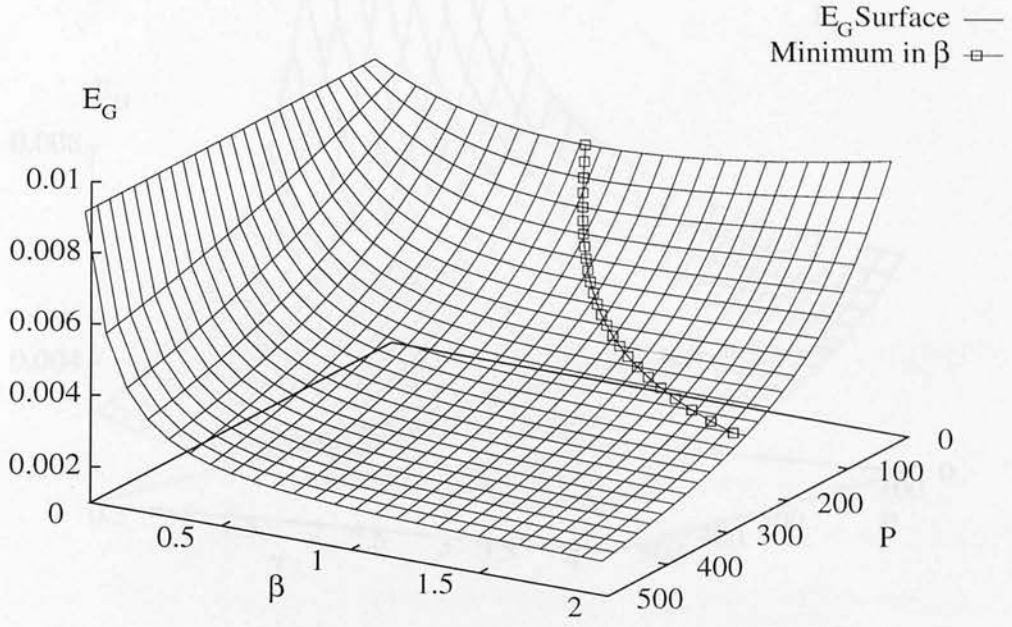


Figure 3.5: Generalization error E_G as a function of number of examples P and error sensitivity β. At the minimum in E_G with respect to β, $\beta \to \infty$ as $P \to \infty$.

3.4.4 Analytic Determination of Optimal Parameters

It is not possible to find closed-form analytic expressions for the optimal settings of β and γ for either E_G or E_B generally, but for the case in which there is no interaction between the basis functions, as may occur when the variance of the input distribution is large compared to the width of the basis functions, such expressions can be obtained; these can then be elaborated upon to some extent in order to suggest the form of the actual dependencies of β_{opt} and γ_{opt}.

For the Bayes-optimal generalizer, by minimising E_B with respect to the training parameters, the optimal settings were determined to be:

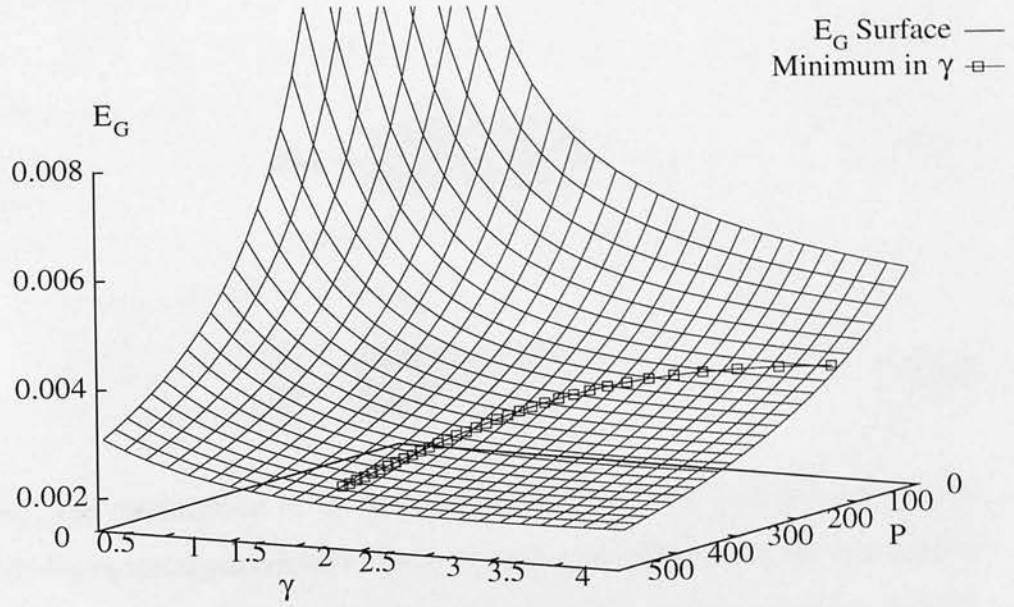


Figure 3.6: Generalization error E_G as a function of number of examples P and weight decay parameter γ. As $P \to \infty$, the value of γ at the minimum in E_G with respect to γ becomes constant.

$$\beta_{opt} = \frac{\gamma \|\boldsymbol{w}^0\|^2}{K\sigma^2} \tag{3.29}$$

and:

$$\gamma_{opt} = \frac{K\sigma^2 \beta}{\|\boldsymbol{w}^0\|^2} \tag{3.30}$$

The form of equations (3.29) and (3.30) proves that only the *ratio* of γ to β, $\frac{2K\sigma^2}{\|\boldsymbol{w}^0\|^2}$, determines whether the parameter settings are optimal.

For the Gibbs generalizer the expressions for optimal parameters are a little

more complicated:

$$\beta_{opt} = \frac{\gamma(2\gamma\|\boldsymbol{w}^0\|^2 + K)}{K(2\gamma\sigma^2 - G_D P)} \tag{3.31}$$

$$\gamma_{opt} = \frac{G_D K P \beta(2\beta\sigma^2 + 1)}{2\|\boldsymbol{w}^0\|^2 \beta G_D P - K} \tag{3.32}$$

Under this assumption of no interactions between the basis functions, the results for optimal parameters closely resemble those found for the perceptron (Bruce and Saad, 1994), an architecture which can also be viewed as having no interactions between units of the layer immediately preceding the output layer.

Allowing terms linear in the interaction parameter, G_O, leads to optimal parameters which have an additional dependence on the cross-correlation of the teacher RBF weight vector, $\sum_{bc} w_b^0 w_c^0$. For instance, the optimal ratio of γ_{opt} to β_{opt} for E_B becomes (with G_D small):

$$\frac{\gamma_{opt}}{\beta_{opt}} = \frac{2\beta\sigma^2 K G_D^2}{(G_D - G_O)G_O \sum_{bc} w_b^0 w_c^0 + (G_D - G_O)^2 \|\mathbf{w}^0\|^2} \tag{3.33}$$

The effect of admitting all terms in G_O for E_B can only be examined empirically. As in the $G_O = 0$ case, β_{opt} was found to be linearly dependent on γ, and vice versa, with the gradient of the γ_{opt} versus β dependence being the reciprocal of that for β_{opt} versus γ. This form of relationship implies that E_B can still be minimised by finding the correct ratio of γ to β; it is unnecessary to find absolute values for these quantities. Thus, the optimal values define

a straight line in training parameter space.

In the case of E_B, the dependence of γ_{opt} and β_{opt} on the noise variance σ^2 can also be found; again, as in the $G_O = 0$ case, γ_{opt} is proportional to σ^2 while β_{opt} is inversely proportional to σ^2.

Mackay (1992) also studied noisy interpolation on a linear model. His goal was to find the optimal parameters by maximising the posterior probability $\mathcal{P}(\gamma, \beta|D)$ of the hyperparameters γ and β, while the approach taken in this thesis is to minimise generalization error with respect to the hyperparameters. Despite this difference in method, some comparison between results can be made. Working from eqn. (2.22) of (Mackay, 1992), one can rewrite MacKay's expression for the optimal value of γ as:

$$\gamma = \frac{\tau}{\|\boldsymbol{w}\|^2} \tag{3.34}$$

where τ is a measure of the effective number of parameters supported by the data. Since here the teacher is known, $\tau = K$, and since in the standard Bayesian formulation employed by MacKay there is no distinction between teacher and student as such, eqn. (3.34) becomes:

$$\gamma = \frac{K}{\|\boldsymbol{w}^0\|^2} \tag{3.35}$$

If the noiselevel is known, then β can be set to[5] $1/\sigma^2$. The optimal setting for γ derived previously in eqn. (3.30), by minimising the Bayes generalization error, then matches that of eqn. (3.35) exactly. No such obvious connection could be found for β, however.

[5] This matches MacKay's definition.

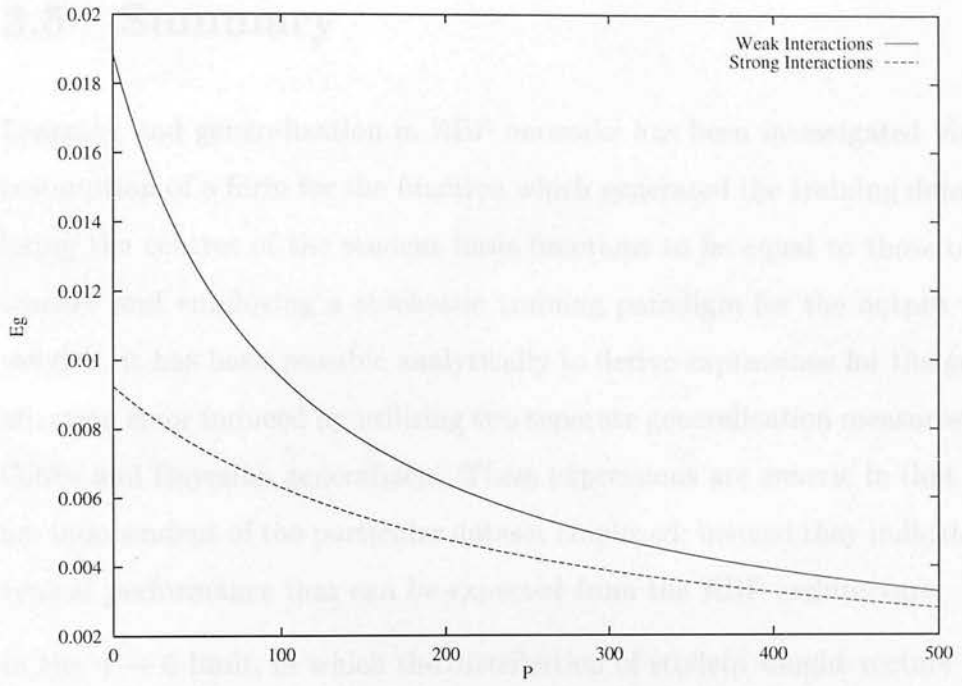


Figure 3.7: The effects of strongly versus weakly interacting hidden units. E_G versus number of training pairs is plotted for weakly-interacting hidden units (top curve) and strongly-interacting hidden units (bottom curve).

3.4.5 Interactions Between Hidden-Layer Units

The effect of joint activations between hidden-layer units, whereby a single training pair simultaneously contributes to the activation of every hidden-layer unit, is to reduce the number of training patterns required to achieve a certain level of generalization error as compared to a network in which there are no such interactions. Consider figure 3.7, in which E_G is plotted for an RBF network with highly-overlapping hidden units and for a network with small overlap: the generalization error for given P is considerably lower for the highly-overlapping version. This phenomenon is due to the fact that high overlaps allow every hidden unit to learn from every training pair, while small overlaps prevent some units from benefiting from certain training pairs.

3.5 Summary

Learning and generalization in RBF networks has been investigated via the assumption of a form for the function which generated the training data. By fixing the centres of the student basis functions to be equal to those of the teacher and employing a stochastic training paradigm for the output node weights, it has been possible analytically to derive expressions for the generalization error induced by utilising two separate generalization measures: the Gibbs and Bayesian generalizers. These expressions are generic in that they are independent of the particular dataset employed; instead they indicate the typical performance that can be expected from the RBF architecture.

In the $\gamma \to 0$ limit, in which the distribution of student weight vectors is effectively induced solely by the training data, both measures of generalization error, E_G and E_B, were found to be inversely proportional to the number of training pairs, P.

The optimal settings of the training parameters γ and β have been examined; it was determined, empirically for the general case and analytically for the simplified situation of no interactions between basis functions, that minimisation of E_B occurs when γ and β are merely set in the correct ratio. However, this result does not apply to E_G, for which each parameter must be optimised separately.

Finally, the interactions between basis functions were shown to be important for rapid learning: strong interactions allow each hidden node to adapt to every training point, while weak interactions imply some training data is effectively ignored by some hidden units.

Chapter 4

Stochastic Learning 2

In real learning scenarios, it is rare that the precise functional form of the data-generating mechanism is known. Therefore it is vital to understand how a student architecture such as the RBF reacts to cases in which it is not matched to the teacher. The student may have more representational power than the teacher, which is known as the over-realizable case, or it may not be able to emulate the teacher exactly even in the limit of infinite training data; this is known as the unrealizable case.

This chapter extends the student-teacher framework introduced previously to allow the investigation of not only the exactly realizable case, but also the unrealizable and over-realizable cases. The data-generating mechanism is a teacher RBF in which the centres and widths of the basis functions need not match those of the student, so that mismatched cases can be investigated.

To facilitate understanding of these cases, generalization error will be viewed as consisting of two components: approximation error and estimation error. Given a particular student architecture, approximation error is the error made by the optimal student of that architecture, and is due to the architec-

ture having insufficient representational power to emulate exactly the process that generated the problem to be learnt; it is an asymptotic quantity in that it cannot be overcome during the training process even in the limit of infinite training data. If the approximation error is zero, the problem is termed realizable; if not, it is termed unrealizable. Estimation error is the error due to not having selected an optimal student of the chosen architecture; it is a dynamic quantity as it changes during training, and is caused by having insufficient data, noisy data, or a learning algorithm which is not guaranteed to reach an optimal solution in the limit of infinite amounts of data. There is a trade-off between representational power and the amount of data required to achieve a particular error value (the sample complexity) in that the more powerful the student, the greater the likelihood that the approximation error can be eliminated but the larger the amount of data required to reduce the estimation error to a particular level.

Also in this chapter the limitation induced by the requirement that the positions of the basis functions of the teacher RBF are known is eliminated by introducing the idea of the degree of confidence in the student basis function positions.

4.1 Finding the Generalization Error

The learning scenario examined is similar to that in chapter 3; the aim is again to analyze average case performance, so a posterior distribution over the space of student weights is constructed, conditioned on the training data and hyperparameters of the learning process. Recapping,

$$\begin{aligned} \mathcal{P}(\boldsymbol{w}|D,\gamma,\beta) &= \frac{\mathcal{P}(D|\boldsymbol{w},\beta)\,\mathcal{P}(\boldsymbol{w}|\gamma)}{\mathcal{P}(D|\gamma,\beta)} \\ &= \frac{\exp\left(-\beta E_D - \gamma E_W\right)}{Z_M} \end{aligned} \tag{4.1}$$

The error measure is again taken to be quadratic; both the Gibbs and Bayes algorithms defined in section 3.2 are analyzed, and the dependency on a particular dataset is eliminated by averaging over all possible datasets. The input distribution is a zero-mean Gaussian with variance σ_ξ^2. The calculation is slightly more complicated than its counterpart in chapter 3 as the sets of teacher and student basis functions are no longer identical in general, so it is no longer possible to combine the student and teacher weight vectors into the single vector $\boldsymbol{w}^*$. Thus, working from eqn.(3.7), generalization error becomes:

$$\ll E_G \gg = \ll \left\langle \int_W d\boldsymbol{w}\, \mathcal{P}(\boldsymbol{w}|D,\gamma,\beta)\,(\boldsymbol{w}^0\cdot\boldsymbol{t} - \boldsymbol{w}\cdot\boldsymbol{s})^2 \right\rangle \gg \tag{4.2}$$

Defining the prior over student weight space as $E_W = \frac{1}{2}\|\boldsymbol{w}\|^2$ and the error on the training data as $E_D = \frac{1}{2}\sum_p \{\boldsymbol{w}\cdot\tilde{\boldsymbol{s}}_p - \boldsymbol{w}^0\cdot\tilde{\boldsymbol{t}}_p + \vartheta_p\}^2$, where, as in previous chapters, $\tilde{\boldsymbol{s}}_p$ and $\tilde{\boldsymbol{t}}_p$ are the counterparts of $\boldsymbol{s}$ and $\boldsymbol{t}$ for training point p such that $\tilde{s}_{p_i} = \exp(-\|\boldsymbol{\xi}_p - \boldsymbol{m}_i\|^2/2\sigma_B^2)$, and where ϑ_p is additive Gaussian noise of variance σ^2 on example p, one can calculate the posterior $\mathcal{P}(\boldsymbol{w}|D,\gamma,\beta)$. Substituting this into eqn.(4.2) leads to:

$$\ll E_G \gg = \ll \left\langle \frac{\boldsymbol{s}^T\boldsymbol{\Lambda}\boldsymbol{s} + 2(\boldsymbol{w}^0\cdot\tilde{\boldsymbol{t}})\boldsymbol{\rho}^T\boldsymbol{\Lambda}\boldsymbol{s}}{P} + \frac{\boldsymbol{\rho}^T\boldsymbol{\Lambda}\boldsymbol{s}\boldsymbol{s}^T\boldsymbol{\Lambda}\boldsymbol{\rho}}{P^2} + (\boldsymbol{w}^0\cdot\boldsymbol{t})^2 \right\rangle \gg \tag{4.3}$$

where:

$$\begin{aligned} \Lambda^{-1} &= \frac{\gamma}{P}\mathbf{I} + \frac{\beta}{P}\sum_p \tilde{\boldsymbol{s}}_p \tilde{\boldsymbol{s}}_p^T \\ \boldsymbol{\rho} &= \beta \sum_p (\vartheta_p - \boldsymbol{w}^0 \cdot \tilde{\boldsymbol{t}}_p)\tilde{\boldsymbol{s}}_p \end{aligned}$$

Note that this definition of $\boldsymbol{\rho}$ is different to that given in chapter 3 due to the possibility of the teacher and student having different numbers of basis functions.

Calculating the expectation of this quantity over both input space and the noise on the dataset,

$$\begin{aligned} \ll E_G \gg \;=\; \ll \frac{\mathrm{tr}\,\boldsymbol{G}\boldsymbol{\Lambda}}{P} + \frac{\sigma^2\beta^2 \sum_p \tilde{\boldsymbol{s}}_p^T \boldsymbol{\Lambda}\boldsymbol{G}\boldsymbol{\Lambda}\tilde{\boldsymbol{s}}_p}{P^2} + \\ \boldsymbol{w}^{0T}\left\{ \frac{\beta^2 \sum_{pq} \tilde{\boldsymbol{t}}_p \tilde{\boldsymbol{s}}_p^T \boldsymbol{\Lambda}\boldsymbol{G}\boldsymbol{\Lambda}\tilde{\boldsymbol{s}}_q \tilde{\boldsymbol{t}}_q^T}{P^2} - \frac{2\beta \sum_p \tilde{\boldsymbol{t}}_p \tilde{\boldsymbol{s}}_p^T \boldsymbol{\Lambda}\boldsymbol{L}}{P} + \boldsymbol{K} \right\} \boldsymbol{w}^0 \gg \end{aligned} \tag{4.4}$$

where $\boldsymbol{G} = \left\langle \boldsymbol{s}\,\boldsymbol{s}^T \right\rangle$, $\boldsymbol{K} = \left\langle \boldsymbol{t}\,\boldsymbol{t}^T \right\rangle$ and $\boldsymbol{L} = \left\langle \boldsymbol{s}\,\boldsymbol{t}^T \right\rangle$ are matrices concerning student, teacher and both student and teacher respectively; these matrices represent the positions of the centres via the average pairwise responses of the hidden units to an input. Full expressions for these quantities can be found in appendix A.

It remains to perform the average over the positions of the training data. This requires the use of the large P regime, but the simulations presented in section 4.4 show the validity of the results for all values of P except for P small. For some of the results that follow, it is also necessary to know the

form of $\mathbf{\Lambda}$; as in chapter 3, where required it will be assumed that $\mathbf{\Lambda}^{-1}$ has a diagonal versus off-diagonal form, such that each diagonal entry is equal to θ, and each off-diagonal entry is equal to $\tilde{\theta}$. This induces a similar form on $\mathbf{\Lambda}$, where:

$$\begin{aligned} \Lambda_{bc} &= \frac{\delta_{bc} + \phi}{\theta - \tilde{\theta}} \\ \phi &= -\frac{\tilde{\theta}}{\theta + \tilde{\theta}(K-1)} \end{aligned}$$

The equality of diagonal entries implies that each basis function receives an equal amount of activation via the training set, while the equality of off-diagonal entries requires each pairwise correlation between basis function activations to be equal: the ramifications of these restrictions are explored via computer simulations in chapter 3. One can find θ and $\tilde{\theta}$ from the definition of $\mathbf{\Lambda}^{-1}$:

$$\begin{aligned} \mathbf{\Lambda}^{-1} &= \frac{\gamma}{P}\mathbf{I} + \frac{\beta}{P}\sum_p \tilde{\boldsymbol{s}}_p \tilde{\boldsymbol{s}}_p^T \\ &\simeq \frac{\gamma}{P}\mathbf{I} + \beta \ll \frac{1}{P}\sum_p \tilde{\boldsymbol{s}}_p \tilde{\boldsymbol{s}}_p^T \gg \\ &= \frac{\gamma}{P}\mathbf{I} + \beta \boldsymbol{G} \end{aligned} \tag{4.5}$$

where variance in the mean of the distribution of $\frac{1}{P}\sum_p \tilde{\boldsymbol{s}}_p \tilde{\boldsymbol{s}}_p$ of magnitude $1/P$ has been neglected

The average over datasets can now be performed , yielding the final expression for generalization error:

$$\ll E_G \gg = \frac{1}{P}\left\{ \operatorname{tr} \boldsymbol{G\Lambda} + \sigma^2\beta^2 \operatorname{tr}\left[(\boldsymbol{G\Lambda})^2\right]\right\} + \tag{4.6}$$
$$\boldsymbol{w}^{0T}\left\{\beta^2\left[\frac{1}{P}\operatorname{tr}\boldsymbol{\Lambda G\Lambda J} + \left(1-\frac{1}{P}\right)\boldsymbol{L}^T\boldsymbol{\Lambda G\Lambda L}\right] - 2\beta\boldsymbol{L}^T\boldsymbol{\Lambda L} + \boldsymbol{K}\right\}\boldsymbol{w}^0$$

where $\boldsymbol{J}$ is a four-dimensional tensor dealing with student and teacher centre positions in the same manner as $\boldsymbol{L}$ (defined in appendix A)[1].

From $\ll E_G \gg$, recalling that the difference between E_G and E_B is simply the variance of the student output with respect to the posterior distribution, one can readily calculate $\ll E_B \gg$:

$$\ll E_B \gg = \ll E_G \gg - \frac{1}{P}\operatorname{tr}\boldsymbol{G\Lambda} \tag{4.7}$$

The expression for $\ll E_G \gg$ appears complicated, but it can be understood by decomposing it into components. The first term represents the variance of the student output, as can be seen from equation (4.7), while the second term is the error due to noise on the training data, which becomes zero in the $\sigma^2 \to 0$ limit. These two terms are purely estimation error. The final term deals with the relationship between the student and teacher, and includes both estimation error and approximation error.

[1] The trace over $\boldsymbol{\Lambda G\Lambda J}$ is over the first two indices of $\boldsymbol{J}$, resulting in a M by M matrix.

4.2 Analysis of Generalization Error

4.2.1 The Effects of Regularization

While the effects of regularization are similar for E_G and E_B, the optimal parameter settings, as found by minimising generalization error with respect to γ and β, are quite different. As discussed in chapter 3, for E_G it is necessary to optimise γ and β jointly, while for E_B, only the *ratio* of γ to β need be considered; this optimal ratio is independent of P. The discrepancy in optimisation requirements is due to the variance term in E_G, which is minimised by taking $\beta \to \infty$. These findings hold for both realizable and unrealizable cases.

To illustrate the effects of regularization in a realizable scenario, consider figure 4.1(a) where E_B, calculated from equation (4.7), is plotted versus P for three cases. The solid curve results from optimal regularization and demonstrates the lowest value of generalization error that can be achieved on average; the dot-dash curve represents the over-regularized case, in which the prior is dominant over the likelihood, showing how reduction in generalization error is substantially slowed. The dashed curve is for the highly under-regularized case, which in the $\gamma/\beta \to 0$ case gives a divergence in both E_G and E_B. Similar behaviour is also found in the linear perceptron[2] (Dunmur and Wallace, 1993).

It is important to note that in the $P \to \infty$ limit (with N fixed), the settings of γ and β are irrelevant as long as $\beta \neq 0$. All results dealing with optimization of training require the assumption of form for $\mathbf{\Lambda}$.

[2]The comparison is not exact, however, as the work of Dunmur and Wallace employs the thermodynamic limit ($N \to \infty, P \to \infty, P/N$ held constant) and focusses exclusively on E_G.

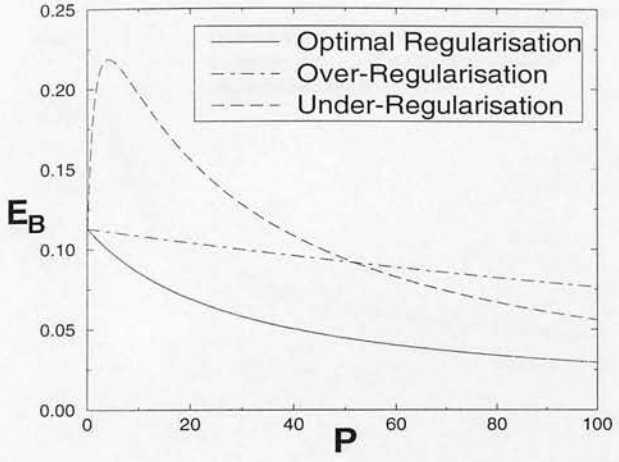
4.2.2 The Over-Realizable Scenario

Operationally, selecting a form for the student implies that one is prepared to believe that the teacher has an identical form. Therefore optimisation of training parameters must be performed on the basis of this belief. When the student is overly powerful this leads to under-regularization, as the magnitude of the teacher weight vector is believed to be larger than the true case. This is illustrated in figure 4.1(b); the dashed curve represents generalization error for the under-regularized case in which the training parameters have been optimised as if the teacher has the same form as the student, while the solid curve below represents the same student, but with training optimised with respect to the true teacher.

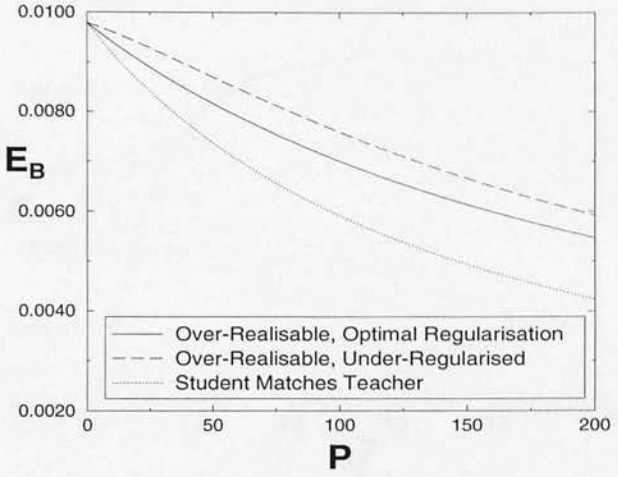
Employing an overly-powerful student can drastically slow the reduction of generalization error as compared to the case where the student matches the teacher. Even with training optimised with respect to the true teacher form, the matching student greatly out-performs the overly-powerful version due to the necessity to suppress the redundant parameters during the training process. This requirement for parameter suppression becomes stronger as the student becomes more powerful. The effect is shown in figure 4.1(b); generalization error for the matching student is given by the dotted curve, while that of the overly-powerful but correctly optimised student is given by the solid curve directly above.

4.2.3 The Unrealizable Scenario

An analogous result to that of the over-realizable scenario is found when the teacher is more powerful than the student. Optimisation of training



(a) The effects of regularization: the solid curve represents optimal regularization ($\gamma = 2.7, \beta = 1.6$), the dot-dash curve illustrates the over-regularized case ($\gamma = 2.7, \beta = 0.16$), and the dashed curve shows the highly under-regularized case ($\gamma = 2.7, \beta = 16$). The student and teacher were matched, each consisting of 3 centres at $(1, 0), (-0.5, 0.866)$ and $(-0.5, -0.866)$. Noise of variance 1 was employed. Note that this is a realizable scenario, as in chapter 3.

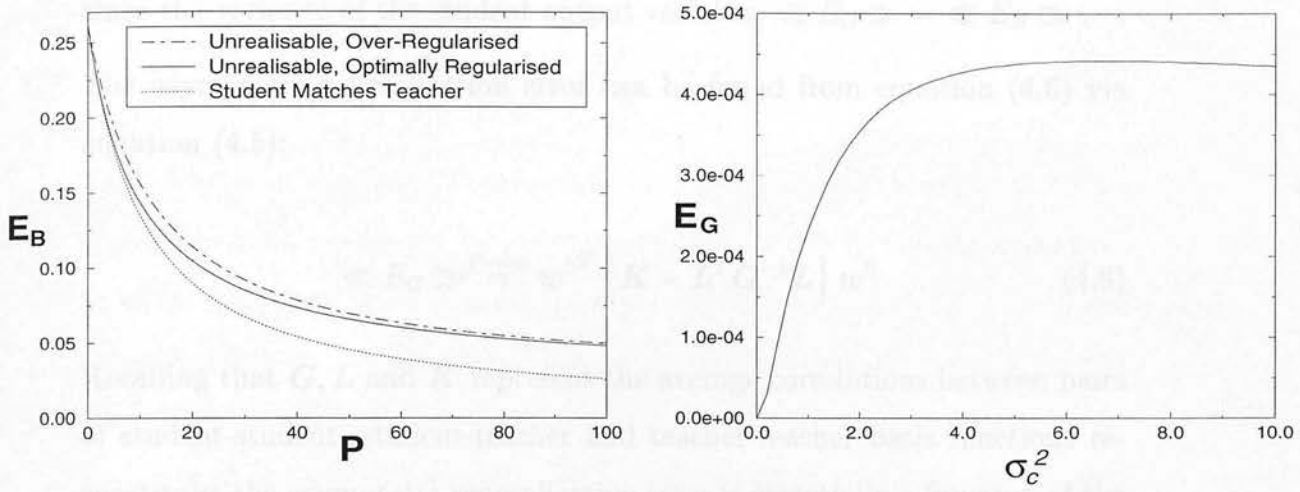


(b) The Over-realizable Case: the dashed curve shows the over-realizable case with training optimised as if the student matches the teacher ($\gamma = 3.59, \beta = 2.56$), the solid curve illustrates the over-realizable case with training optimised with respect to the true teacher ($\gamma = 3.59, \beta = 1.44$), while the dotted curve is for the student matching the teacher ($\gamma = 6.52, \beta = 4.39$). All the curves were generated with one teacher centre at $(1, 0)$; the over-realizable curves had two student centres at $(1, 0)$ and $(-1, 0)$. Noise with variance 1 was employed.

Figure 4.1: Regularization and the Over-realizable Case

parameters under the belief that the teacher has the same form as the student leads to over-regularization, due to the assumed magnitude of the teacher weight vector being greater than the actual magnitude. This effect is shown in figure 4.2(a), in which the dot-dash curve denotes generalization error for the over-regularized case based on the belief that the teacher matches the student, while the solid curve below shows the error for an identical student when the parameters of the true teacher are known; this knowledge permits optimal regularization.

The most significant effect of the teacher being more powerful than the stu-



(a) The Unrealizable case: the dot-dash curve denotes the case where the student is optimised as if the teacher is identical to it ($\gamma = 2.22, \beta = 1.55$); the solid curve demonstrates the student optimised with knowledge of the true teacher ($\gamma = 2.22, \beta = 3.05$), while, for comparison, the dotted curve shows a student which matches the teacher ($\gamma = 2.22, \beta = 1.05$). The curves were generated with two teacher centres at $(1, 0)$ and $(-1, 0)$; the unrealizable curves employed a single student at $(1, 0)$; noise of variance 1 was utilised.

(b) Approximation Error versus the belief parameter, σ_c^2. Approximation error initially increases with the uncertainty in teacher centre position, but as the uncertainty increases, the teacher centres become further from the centre of the input distribution. This causes the target function to eventually approach zero in the region in which input is likely, and thus approximation error will also reduce to zero. This process can be seen to begin from $\sigma_c^2 \simeq 7.6$ in this particular example.

Figure 4.2: The Unrealizable Case and the Belief Parameter

dent is the fact that the approximation error is no longer zero, as the teacher can never be exactly emulated by the student. This is illustrated in figure 4.2(a), where the dotted curve represents the learning curve when the student matches the teacher (and has a zero asymptote), while the two upper curves show an under-powerful student, and have non-zero asymptotes.

In order to consider the effect of a mismatch between student and teacher, the infinite example limit was calculated. In this limit, the variance of the student output and error due to noise on the training data both disappear, as do transient errors due to the relation between student and teacher, leaving

only the error that cannot be overcome within the training process. Note that since the variance of the student output vanishes, $\ll E_G \gg = \ll E_B \gg$.

The asymptotic generalization error can be found from equation (4.6) via equation (4.5):

$$\ll E_G \gg \overset{P\to\infty}{=} \boldsymbol{w}^{0T} \left\{ \boldsymbol{K} - \boldsymbol{L}^T \boldsymbol{G}^{-1} \boldsymbol{L} \right\} \boldsymbol{w}^0 \tag{4.8}$$

Recalling that $\boldsymbol{G}, \boldsymbol{L}$ and $\boldsymbol{K}$ represent the average correlations between pairs of student-student, student-teacher and teacher-teacher basis functions respectively, the asymptotic generalization error is essentially a function of the correlations between hidden unit responses. There is a also a dependence on input-space dimension, basis function width and input distribution variance via the normalisation constants, and on the hidden-to-output weights of the teacher. In the realizable case $\boldsymbol{G} = \boldsymbol{L} = \boldsymbol{K}$, and so it can be seen that the asymptotic error disappears. Note that this result is independent of the assumption of diagonal-offdiagonal form for $\boldsymbol{\Lambda}$.

4.2.4 Dependence of Estimation Error on Training Set Size

In the limit of no weight decay, it is simple to show that the estimation error portion of the generalization error is inversely proportional to the number of training examples.

From equation (4.6), using equation (4.5), the estimation error is:

$$\ll E_G \gg_{EST} = \frac{K}{P}\left\{\frac{1}{\beta} + \sigma^2\right\} + \frac{1}{P}\boldsymbol{w}^{0^T}\left\{\mathrm{tr}\,\boldsymbol{G}^{-1}\boldsymbol{J} - \boldsymbol{L}^T\boldsymbol{G}^{-1}\boldsymbol{L}\right\}\boldsymbol{w}^0 \quad (4.9)$$

Taking $\gamma \to 0$, the only P-dependencies are in the $1/P$ prefactors. This result has been confirmed by simulations, carried out in the same manner as those described in section 4.4; plotting the log of the averaged empirical generalization error versus $\log P$ gives a gradient of -1. It is also apparent that, with no weight decay, the best policy is to set $\beta \to \infty$, to eliminate the variance of the student output. This corresponds to selecting the student weight vector most consistent with the data, regardless of the noise level. This result is also independent of the form of $\boldsymbol{\Lambda}$.

4.3 Removing the Dependence on a Specific Teacher

The results described so far still have a dependence on knowing the weights and centre positions of the teacher RBF. Since this scenario is rarely the case practically, it is preferable to relax this assumption, while bearing in mind that it is *impossible* to examine generalization without some *a priori* belief in the data generation mechanism (Wolpert, 1996a,b).

The requirement that the teacher centres are known will be replaced by a single parameter, corresponding to degree of confidence in the student centres; each teacher centre will be considered to be drawn from a Gaussian distribution centred on a specific student with variance given by the confidence parameter σ_c^2. Thus, for each student centre, regions are defined centred on

the student in which the corresponding teacher centre is believed to lie with a certain probability[3].

For simplicity, the exact knowledge of the teacher weight vector will be replaced by the belief that it is drawn from a Gaussian distribution of mean zero and variance σ_w^2.

By varying the degree of confidence parameter, one can smoothly control the severity of the unrealizability, which supersedes the dichotomy between realizable and unrealizable cases. Absolute realizability is only regained in pathological cases, such as $\sigma_c^2 \to \infty$ or $\sigma_w^2 = 0$.

The typical generalization performance of the network can now be found by averaging E_G and E_B with respect to the teacher centres and weights, producing[4]:

$$\begin{aligned} \ll E_G \gg \; = \; & \frac{1}{P}\left\{ \operatorname{tr} \boldsymbol{G\Lambda} + \sigma^2\beta^2 \operatorname{tr} \boldsymbol{G\Lambda G\Lambda} \right\} + \\ & \beta^2\sigma_w^2 (\Lambda G \Lambda)_{cb} \left\{ \frac{1}{P} J'_{bcuu} + \left(1 - \frac{1}{P}\right) L'_{bcuu} \right\} - \\ & 2\beta\sigma_w^2 \Lambda_{cb} L'_{bcuu} + \sigma_w^2 \operatorname{tr} \boldsymbol{K}' \end{aligned} \tag{4.10}$$

and, again,

$$\ll E_B \gg = \ll E_G \gg - \frac{\operatorname{tr} \boldsymbol{G\Lambda}}{P} \tag{4.11}$$

[3]Mathematically, it is quite feasible to postulate a different degree of confidence in *each* student centre; however, this complicates the analysis and increases the number of free parameters without adding much in the way of insight, and so is not presented.

[4]The Einstein summation convention of summation over all repeated indices is employed

where $\boldsymbol{J}^{'}, \boldsymbol{K}^{'}$ and $\boldsymbol{L}^{'}$ (defined explicitly in appendix A) are the counterparts of $\boldsymbol{J}, \boldsymbol{K}$, and $\boldsymbol{L}$ averaged with respect to the teacher centres and weights. As before, $\ll E_G \gg$ consists of three parts: student variance, noise effects and the relationship between student and teacher, consisting of both approximation and estimation error. Only the latter part is affected by the belief parameter, σ_c^2. Figure 4.2(b) demonstrates the variation of the approximation error with σ_c^2: the approximation error initially increases monotonically with σ_c^2, but as the uncertainty increases, the teacher centres become further from the centre of the input distribution. This eventually causes the averaged approximation error to decrease, as the target function approaches zero in the region where input is likely.

4.4 Validation of the Analytic Results

In order to validate the analytic results, simulations were carried out for three realizable scenarios: well-regularized, over-regularized and under-regularized training. The simulations involved exhaustive training of RBF networks using the Langevin update procedure. Specifically, a network of three units in two-dimensional input space was employed, the centres of both student and teacher being at $(1, 0), (-0.5, 0.866)$ and $(-0.5, -0.866)$ and having width 0.707. The noise in the update step was set to $2/\beta$. For each simulation curve, 100 training runs were performed, with the generalization error being approximated by the error on a large, noiseless test set. The results for the well-regularized and under-regularized cases are presented in figure 4.3; the over-regularized case was qualitatively similar to the well-regularized case. An excellent fit between analytic and simulated results for all curves is apparent for $P > 100$, where the effects of ignoring the variance in the dataset

average become negligible. In the region where P is small, the means of the simulations fluctuate about the analytic curves for the well-regularized case, but for the under-regularized case, the analytic mean is somewhat larger than the simulation result. In the small P region, the under-regularized case is particularly vulnerable to the approximation employed in the dataset average as $\mathbf{\Lambda}$ will be dominated by the correlations between hidden unit responses to the dataset, rather than by the prior. Note that the errorbars are also large in this region, as the distribution of student weights is relatively unconstrained.

The simulations presented are for the case of a specific teacher network but simulations have also been carried out for the situation where a belief parameter was specified, with similar results.

4.5 Summary

Learning and generalization in RBFs has been analysed within a stochastic training paradigm by assuming that the training data has been generated by a teacher RBF, but one for which the centre positions and widths need not correspond to those of the student RBF, thus allowing the analysis of unrealizable and over-realizable cases.

The effects of regularization have been examined: under-regularization initially causes very poor generalization, but this can be overcome rapidly with the addition of more training data. Over-regularization is initially less damaging, but requires a large quantity of training data in order to overcome the effect.

The case in which the student is of greater representational power than the teacher has been examined; it was found that there is a tendency to under-

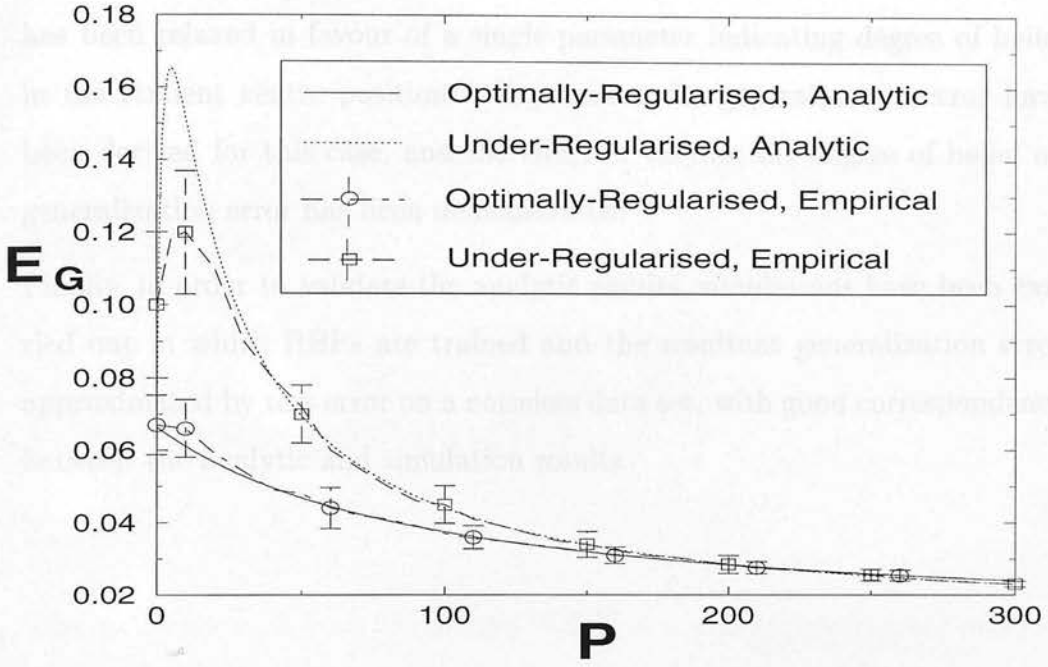


Figure 4.3: Simulation results showing the validity of the calculation of E_G and E_B. The curves are for a realizable case with three centres at $(1, 0)$, $(-0.5, 0.866)$ and $(-0.5, -0.866)$, with centre width 0.707 and noise of variance $2/\beta$. The empirical curves were generated by exhaustive training at each value of P, and represent averages over 100 trials. The error bars denote 1 standard deviation of the empirical distribution.

regularize due to over-estimating the complexity of the teacher. Even when optimal regularization is applied, the power of the student causes an increase in sample complexity as compared to the correct student. An analogous effect was found when the teacher has greater representational power than the student, in that under-estimating the complexity of the teacher leads to over-regularization. The primary effect of the unrealizable case is that the generalization error does not become zero in the limit of an infinite number of examples; the remaining component, the approximation error, has been exactly calculated.

The requirement that the exact positions of the teacher centres is known has been relaxed in favour of a single parameter indicating degree of belief in the student centre positions. Expressions for generalization error have been derived for this case, and the effect of varying the degree of belief on generalization error has been demonstrated.

Finally, in order to validate the analytic results, simulations have been carried out in which RBFs are trained and the resultant generalization error approximated by test error on a noiseless data set, with good correspondence between the analytic and simulation results.

Chapter 5

On-line Learning

The previous chapters on the subject of RBF learning focussed on the analysis of networks with a single adaptive layer. While representationally powerful, being capable of universal approximation of continuous functions, in general it is impossible to fully optimize the parameters of such networks. This is due to the two-stage training process in which the parameters of the hidden layer are fixed, usually without regard to the labels of the training data, before adapting the hidden-to-output weights.

It has proved very difficult to analyze the learning and generalization properties of networks with more than one adaptive layer. As discussed, while several tools exist which facilitate the analytic investigation of learning and generalization in supervised neural networks, such as the statistical physics methods (see Watkin *et al.*, 1993, for a review), the Bayesian framework (e.g. MacKay, 1992) and the PAC method (Haussler, 1994), these tools have principally been applied to simple networks, such as linear and boolean perceptrons, and various simplifications of the committee machine (see, for instance, Schwarze (1993) and references therein).

Recently an approach based on studying the dynamics of on-line gradient descent training scenarios has been used by several authors (Heskes and Kappen, 1991; Leen and Orr, 1994; Amari, 1993) to examine the evolution of system parameters primarily in the asymptotic regime. A similar approach, based on examining the dynamics of overlaps between characteristic system vectors in on-line training scenarios has been suggested in Saad and Solla (1995a,b) for investigating the learning dynamics in the SCM (Biehl and Schwarze, 1995). This approach provides a complete description of the learning process, formulated in terms of the overlaps between vectors in the system, and can be easily extended to include general two-layer networks (Riegler and Biehl, 1995).

This chapter presents a method for analyzing the behaviour of RBFs in an *on-line* learning scenario whereby network parameters are modified after each presentation of an example, which allows the calculation of generalization error as a function of a set of variables characterizing the properties of the adaptive parameters of the network. The dynamical evolution of these variables in the average case can be found, allowing not only the investigation of generalization ability, but also allowing the internal dynamics of the network, such as specialization of hidden units, to be analyzed. This tool has also been applied to MLPs (Saad and Solla, 1995a,b; Riegler and Biehl, 1995).

5.1 Training Paradigms and Non-linear Optimization

Although the single adaptive layer training method investigated in the previous RBF-related chapters generally gives sub-optimal solutions, the problem

is linear in the adaptive weights, and thus is fast to solve and amenable to analysis. Adopting the alternative method in which the hidden layer parameters (either just the centre positions or both centre positions and widths) are adapted simultaneously renders the problem non-linear in the adaptable parameters, and hence requires an optimization technique, such as gradient descent, to estimate these parameters. This second approach is computationally more expensive, but usually leads to greater accuracy of approximation. This chapter investigates the non-linear approach in which basis function centres are continuously modified to allow convergence to more optimal models.

There are two common methods in use for gradient descent. In *batch learning*, one attempts to minimize the additive training error over the entire dataset; adjustments to parameters are performed once the full training set has been presented. The alternative approach, examined here, is *on-line learning*, in which the adaptive parameters of the network are adjusted after each presentation of a new datapoint[1]. There has been a resurgence of interest analytically in the on-line method, as technical difficulties caused by the variety of ways in which a training set of given size can be selected are avoided, so complicated techniques such as the replica method (Hertz *et al.*, 1989) are unnecessary.

5.2 On-line learning in RBF networks

This chapter examines a gradient descent on-line training scenario on a continuous error measure. As in previous chapters, the trained model (student) is an RBF network consisting of K basis functions. The centre of student

[1] Obviously one may employ a method which is a compromise between the two extremes.

basis function (SBF) b is denoted by $\boldsymbol{m}_b$ while the hidden-to-output weights of the student are represented by $\boldsymbol{w}$. Training examples will consist of input-output pairs $(\boldsymbol{\xi}, y)$. The components of $\boldsymbol{\xi}$ are uncorrelated Gaussian random variables of mean 0, variance σ_ξ^2, while y is generated by applying $\boldsymbol{\xi}$ to a deterministic teacher RBF, but one in which the number M and position of the hidden units need not correspond to that of the student, which allows investigation of over-realizable and unrealizable cases[2]. The mapping implemented by the teacher is denoted by f_T, and that of the student by f_S. The hidden-to-output weights of the teacher are $\boldsymbol{w}^0$ while the centre of teacher basis function u is given by $\boldsymbol{n}_u$. The vector of student basis function responses to input vector $\boldsymbol{\xi}$ is represented by $\boldsymbol{s}(\boldsymbol{\xi})$, while those of the teacher are denoted by $\boldsymbol{t}(\boldsymbol{\xi})$. The overall functions computed by the networks are therefore[3]:

$$f_S(\boldsymbol{\xi}) = \sum_{b=1}^{K} w_b \exp\left(-\frac{\|\boldsymbol{\xi} - \boldsymbol{m}_b\|^2}{2\sigma_B^2}\right) = \boldsymbol{w} \cdot \boldsymbol{s}(\boldsymbol{\xi}) \tag{5.1}$$

$$f_T(\boldsymbol{\xi}) = \sum_{u=1}^{M} w_u^0 \exp\left(-\frac{\|\boldsymbol{\xi} - \boldsymbol{n}_u\|^2}{2\sigma_B^2}\right) = \boldsymbol{w}^0 \cdot \boldsymbol{t}(\boldsymbol{\xi}) \tag{5.2}$$

This notation is the same as in chapters 3 and 4, but note that the student centre vectors $\boldsymbol{m}$ are now adaptive. As previously, N will denote the dimensionality of input space and P the number of examples presented.

While the centres of the basis functions (input-to-hidden weights) and the hidden-to-output weights are considered adjustable, for simplicity the widths of the basis functions are fixed to a common value σ_B. The framework allows the investigation of the case where these widths are also adaptive, but this

[2]This represents a *general* training scenario since, being universal approximators, RBF networks can approximate any continuous mapping to a desired degree.

[3]Indices b,c,d and e will always represent SBFs, while u and v will represent those of the teacher.

adds greatly to the complexity of the analysis. For analytical convenience, the evolutions of the centres of the basis functions are redescribed in terms of the overlaps $Q_{bc} \equiv \boldsymbol{m}_b \cdot \boldsymbol{m}_c$, $R_{bu} \equiv \boldsymbol{m}_b \cdot \boldsymbol{n}_u$ and $T_{uv} \equiv \boldsymbol{n}_u \cdot \boldsymbol{n}_v$, where T_{uv} is constant and describes characteristics of the task to be learnt.

Previous work in this area (Biehl and Schwarze, 1995; Saad and Solla, 1995a,b; Riegler and Biehl, 1995) has relied upon the thermodynamic limit[4]. This limit allows one to ignore fluctuations in the updates of the means of the overlaps due to the randomness of the training examples, and permits the difference equations of gradient descent to be considered as differential equations. The thermodynamic limit is hugely artificial for local RBFs; as the activation is localized, the $N \to \infty$ limit implies that a basis function responds only in the vanishingly unlikely event that an input point falls exactly on its centre; there is no obvious reasonable rescaling of the basis functions[5]. The curse of dimensionality, discussed on section 1.2, is at its most potent here. The price paid for not taking this limit is that one has no *a priori* justification for ignoring the fluctuations in the update of the adaptive parameters due to the randomness of the training example. In this chapter and chapter 6, both the means and variances of the adaptive parameters are calculated, showing that the fluctuations are practically negligible.

5.3 Calculating the Generalization Error

Generalization error measures the average dissimilarity over input space between the desired mapping f_T and that implemented by the learning model

[4] $P \to \infty, N \to \infty$ and $P/N = \alpha$, where α is finite.

[5] For instance, utilizing $\exp\left(-\frac{\|\boldsymbol{\xi}-\boldsymbol{m}_b\|^2}{2N\sigma_B^2}\right)$ eliminates all directional information as the cross-term $\boldsymbol{\xi} \cdot \boldsymbol{m}_b$ vanishes in the thermodynamic limit.

f_S. This dissimilarity is taken as quadratic deviation:

$$E_G = \left\langle \frac{1}{2}\left[f_S - f_T\right]^2 \right\rangle \tag{5.3}$$

where $\langle \cdots \rangle$ denotes an average over input space with respect to the measure $\mathcal{P}(\boldsymbol{\xi})$.

Substituting the definitions of equations (5.1) and (5.2) into (5.3) leads to:

$$E_G = \frac{1}{2}\left\{ \boldsymbol{w}^T \left\langle \boldsymbol{s}\boldsymbol{s}^T \right\rangle \boldsymbol{w} + \boldsymbol{w}^{0T} \left\langle \boldsymbol{t}\boldsymbol{t}^T \right\rangle \boldsymbol{w}^0 - 2\boldsymbol{w}^T \left\langle \boldsymbol{s}\boldsymbol{w}^{0T} \right\rangle \boldsymbol{w}^0 \right\} \tag{5.4}$$

Since the input distribution is Gaussian, the averages are Gaussian integrals and can be performed analytically; the resulting expression for generalization error is given in appendix B. Each average has dependence on combinations of $\boldsymbol{Q}$,$\boldsymbol{R}$ and $\boldsymbol{T}$ depending on whether the averaged basis functions belong to student or teacher.

5.4 System Dynamics

Expressions for the time evolution of the overlaps $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be derived by employing the gradient descent rule, $\boldsymbol{m}_b^{p+1} = \boldsymbol{m}_b^p + \frac{\eta}{N\sigma_B^2}\delta_b(\boldsymbol{\xi} - \boldsymbol{m}_b)$, where $\delta_b = (f_T - f_S)w_b s_b$ and η is the learning rate which is explicitly scaled with $1/N$:

$$\begin{aligned} \langle \Delta Q_{bc} \rangle \;=\; & \frac{\eta}{N\sigma_B^2} \langle \left[\delta_b(\boldsymbol{\xi} - \boldsymbol{m}_b^p)\cdot \boldsymbol{m}_c^p + \delta_c(\boldsymbol{\xi} - \boldsymbol{m}_c^p)\cdot \boldsymbol{m}_b^p\right] \rangle + \\ & \left(\frac{\eta}{N\sigma_B^2}\right)^2 \langle\, \delta_b\delta_c(\boldsymbol{\xi} - \boldsymbol{m}_b^p)\cdot(\boldsymbol{\xi} - \boldsymbol{m}_c^p) \,\rangle \end{aligned} \tag{5.5}$$

$$\langle \Delta R_{bu} \rangle = \frac{\eta}{N\sigma_B^2} \langle \delta_b(\boldsymbol{\xi} - \boldsymbol{m}_b^p) \cdot \boldsymbol{n}_u \rangle \tag{5.6}$$

The hidden-to-output weights can be treated similarly. In general one may choose different learning rates for the dynamics of the centres and of the hidden-to-output weights. Here, the same learning rate is used, but it is scaled differently (with $1/K$, in agreement with results obtained by Riegler (1997) for the MLP, yielding:

$$\langle \Delta w_b \rangle = \frac{\eta}{K} \langle (f_T - f_S) s_b \rangle \tag{5.7}$$

Note that scaling the learning rate with $1/K$ does not make a significant difference in this case, since the thermodynamic limit has not been employed for N, in comparison to the exact MLP calculation where adiabatic elimination should be employed for restoring the self-averaging properties of the overlaps (Riegler, 1997).

These averages are again Gaussian integrals, so can be carried out analytically. The averaged expressions for $\Delta \boldsymbol{Q}, \Delta \boldsymbol{R}$ and $\Delta \boldsymbol{w}$ are given in appendix B.

By iterating equations (5.5), (5.6) and (5.7), the evolution of the learning process can be tracked. This allows one to examine facets of learning such as specialization of the hidden units. Since generalization error depends on $\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{w}$, one can also use these equations in conjunction with equation (5.4) to track the evolution of generalization error.

5.5 Analyzing the Learning Process

The set of system evolutions described in the following sections are obtained by iterating the difference equations (5.5), (5.6) and (5.7) from random initial conditions sampled from the following distributions: Q_{bb} and w_b are sampled from $U[0, 10^{-4}]$, while $Q_{bc,b\neq c}$ and R_{bc} from a uniform distribution $U[0, 10^{-5}]$, which represent random correlations expected by arbitrary initialization of systems of the size employed. The evolution describes the mean behaviour of the overlaps and hidden-to-output weights, assuming the variances are negligible; these mean behaviours can then be used to find the evolution of generalization error via equation (5.4).

5.5.1 The Importance of the Learning Rate

With all the teacher basis functions (TBFs) positive, analysis of the time evolution of the generalization error, overlaps and hidden-to-output weights for various settings of the learning rate reveals the existence of three distinct behaviours. If η is chosen to be too small , there is a long period in which there is no specialization of the SBFs, and no improvement in generalization ability: the process becomes trapped in a symmetric subspace of solutions; this is the symmetric phase. Given asymmetry in the student initial conditions (i.e. in $\boldsymbol{R}$, $\boldsymbol{Q}$ or $\boldsymbol{w}$), or of the task itself, this subspace will always be escaped and the task eventually solved, but the time period required may be prohibitively large (figure 5.1(a), dotted curve, $\eta = 0.1$). The length of the symmetric phase increases with the symmetry of the initial conditions. At the other extreme, if η is set too large, an initial transient takes place quickly, but there comes a point from which the student vector norms grow

extremely rapidly, until the point where, due to the finite variance of the input distribution and local nature of the basis functions, the SBFs are no longer activated during training (figure 5.1(a), dashed curve, with $\eta = 7.0$). In this case, the generalization error approaches a finite value as $P \to \infty$ and the task is not solved. Between these extremes lies a region in which the symmetric subspace is escaped quickly, and $E_G \to 0$ as $P \to \infty$ for the realizable case (figure 5.1(a), solid curve, with $\eta = 0.9$). The SBFs become specialized and, asymptotically, the teacher is emulated exactly.

These results for the learning rate are qualitatively similar to those found for SCMs and MLPs (Biehl and Schwarze, 1995; Saad and Solla, 1995a,b; Riegler and Biehl, 1995).

5.5.2 An Example of System Evolution

There are four distinct phases in the learning process, which are described with reference to an example of learning an exactly realizable task. This task consists of a network of 3 student basis functions (SBFs) learning a *graded* teacher of 3 TBFs, where *graded* implies that the square norms of the TBFs (diagonals of $\boldsymbol{T}$) differ from one another; for this task, $T_{00} = 0.5, T_{11} = 1.0$, and $T_{22} = 1.5$. As previously stated, the widths of the student basis functions are considered fixed and equal to those of the teacher for simplicity; also note that the teacher always produces a continuous mapping, and noise is not employed.

For this particular task the teacher is chosen to be uncorrelated, with the off-diagonals of $\boldsymbol{T}$ set to 0, and the teacher hidden-to-output weights $\boldsymbol{w}^0$ to 1. The learning process is illustrated in figure 5.1; figure 5.1(a) (solid curve) shows the evolution of generalization error, calculated from equation (5.4),

while figures 5.1(b) to 5.1(d) show the evolution of the equations for the means of $\boldsymbol{R}$, $\boldsymbol{Q}$ and $\boldsymbol{w}$ respectively, calculated by iterating equations (5.5), (5.6) and (5.7) from random initial conditions as described above. Input dimensionality $N = 8$, learning rate $\eta = 0.9$, input variance $\sigma_\xi^2 = 1$ and basis function width $\sigma_B = 1$ were employed.

The picture that emerges mirrors that of the SCM and MLP (Saad and Solla, 1995b; Riegler and Biehl, 1995). Initially, there is a short *transient* phase in which the overlaps and hidden-to-output weights evolve from their initial conditions until they reach an approximately steady value ($P = 0$ to $P = 4000$). The *symmetric* phase then begins, which is characterized by a plateau in the evolution of the generalization error (see figure 5.1(a), solid curve, $P = 4000$ to $P = 5 \times 10^4$), corresponding to a lack of differentiation amongst the hidden units; they are unspecialized and learn an average of the hidden units of the teacher, so that the student centre vectors and hidden-to-output weights are similar (figures 5.1(b) to 5.1(d)). The difference in the overlaps $\boldsymbol{R}$ between student centre vectors and teacher centre vectors (figure 5.1(b)) is *only* due to the difference in the lengths of various teacher centre vectors; if the overlaps were normalized, they would be identical. The symmetric phase is followed by a *symmetry-breaking* phase in which the SBFs learn to specialize, and become differentiated from one another ($P = 5 \times 10^4$ to $P = 1.7 \times 10^5$). Finally there is a long *convergence* phase, as the overlaps and hidden-to-output weights reach their asymptotic values. Since the task is realizable, this phase is characterized by $E_G \to 0$ (figure 5.1(a), solid curve), and by the student centre vectors and hidden-to-output weights approaching those of the teacher (i.e. $Q_{00} = R_{00} = 0.5, Q_{11} = R_{11} = 1.0, Q_{22} = R_{22} = 1.5$, with the off-diagonal elements of both $\boldsymbol{Q}$ and $\boldsymbol{R}$ being zero; $\forall b, w_b = 1$). The arbitrary labels of the SBFs were permuted to match those of the teacher.

These phases are generic in that they are observed, sometimes with some variation such as a series of symmetric and symmetry-breaking phases, in every on-line learning scenario for RBFs so far examined.

One should point out that the formalism describes the evolution of the means (and the variances) from certain initial conditions. Convergence of the dynamics to sub-optimal attractive fixed points (local minima) may occur if the starting point is within the corresponding basin of attraction. No local minima have been observed in the solutions, which may be an artifact of the system dimensionality.

5.5.3 Task Dependence

The symmetric phase is a phenomenon which depends on the symmetry of the task as well as that of the initial conditions. One would expect a shorter symmetric phase in inherently asymmetric tasks. To examine this, a task similar to that of section 5.5.2 was employed, with the single change being that the sign of one of the teacher hidden-to-output weights was flipped, thus providing two categories of targets: positive and negative. The initial conditions of the student remained the same as in the previous task, with $\eta = 0.9$.

The evolution of generalization error and the overlaps for this task are shown in figures 5.2(a) and 5.2(b) respectively. The dividing of the targets into two categories effectively eliminates the symmetric phase; this can be seen by comparing the evolution of the generalization error for this task (figure 5.2(a), dashed curve) with that for the previous task (figure 5.2(a), solid curve). There is no longer a plateau in the generalization error. Correspondingly, the symmetries between SBFs break immediately, as can be seen by examining

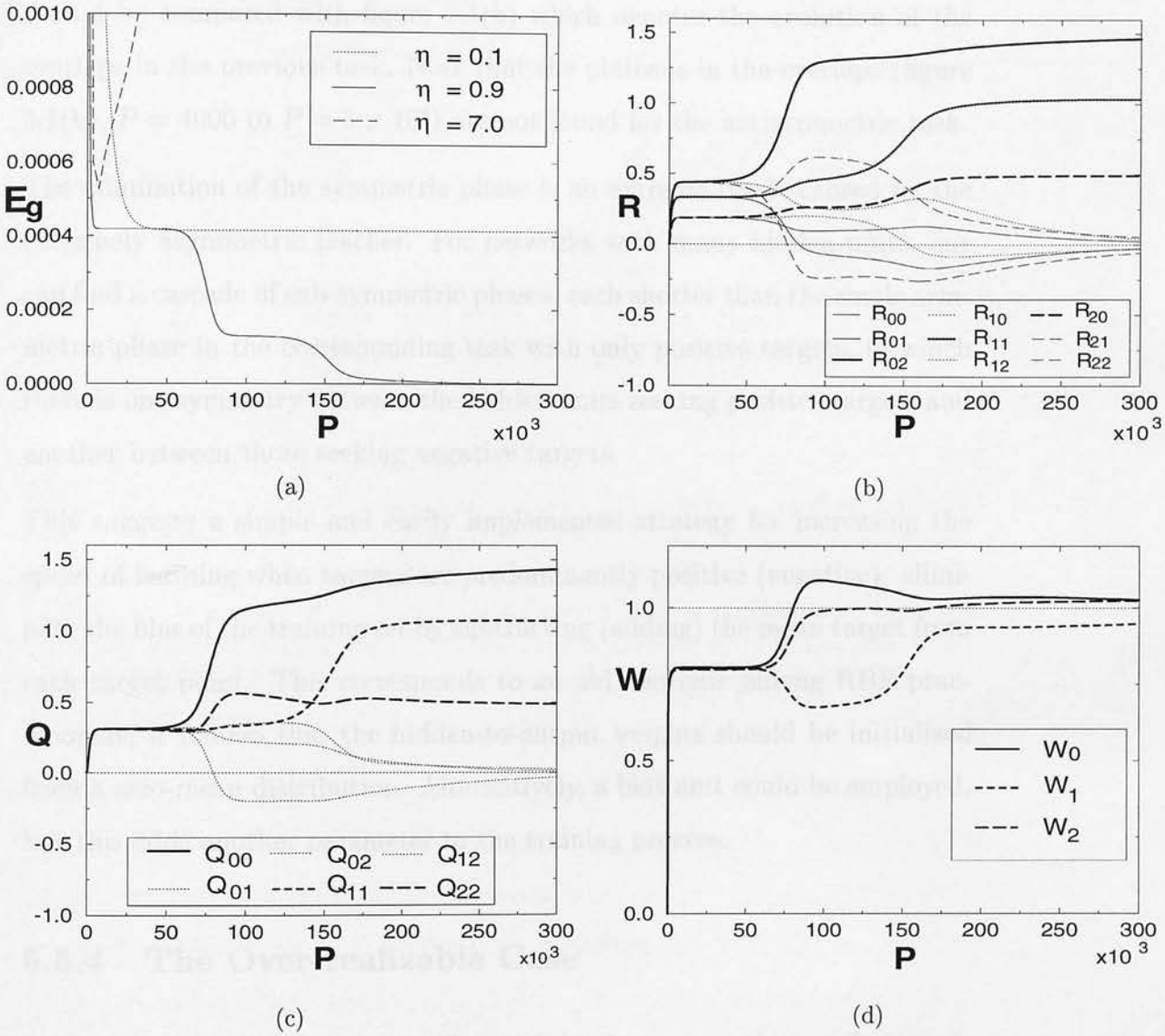


Figure 5.1: The exactly realizable scenario with positive TBFs. Three SBFs learn a graded, uncorrelated teacher of three TBFs with $T_{00} = 0.5, T_{11} = 1.0$ and $T_{22} = 1.5$. All teacher hidden-to-output weights are set to 1. Figure (a) describes the evolution of the generalization error as a function of the number of examples for several different learning rates ($\eta = 0.1, 0.9, 7.0$); (b) and (c) follow the evolution of overlaps between student and teacher centre vectors and among student centre vectors respectively, while (d) monitors the evolution of the mean hidden-to-output weights.

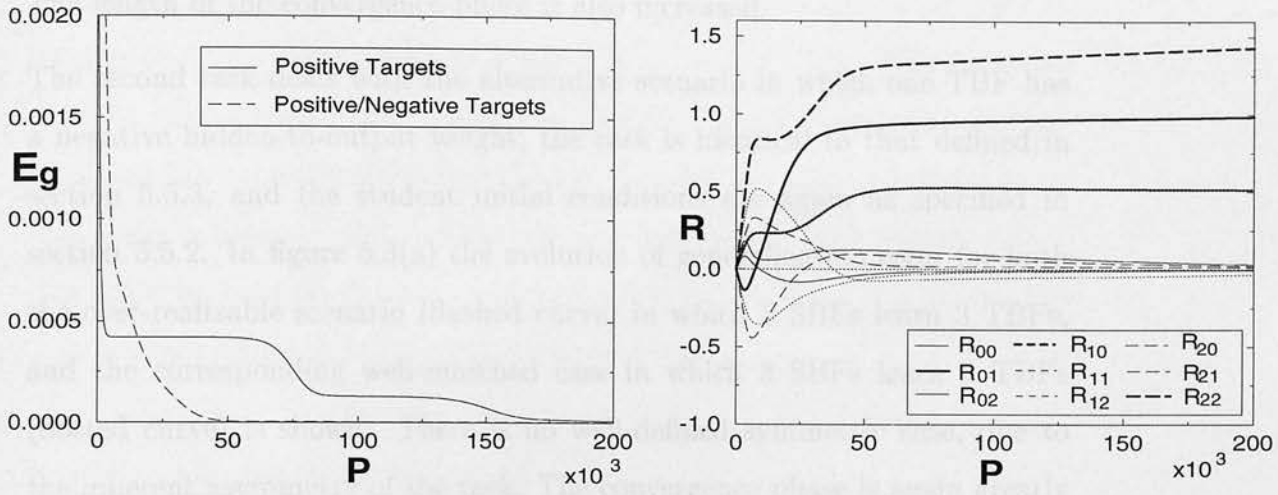
the overlaps between student and teacher centre vectors (figure 5.2(b)); this should be compared with figure 5.1(b) which denotes the evolution of the overlaps in the previous task. Note that the plateaus in the overlaps (figure 5.1(b), $P = 4000$ to $P = 5 \times 10^4$) are not found for the antisymmetric task.

The elimination of the symmetric phase is an extreme result caused by the extremely asymmetric teacher. For networks with many hidden units, one can find a cascade of sub-symmetric phases, each shorter than the single symmetric phase in the corresponding task with only positive targets, in which there is one symmetry between the hidden units seeking positive targets and another between those seeking negative targets.

This suggests a simple and easily implemented strategy for increasing the speed of learning when targets are predominantly positive (negative): eliminate the bias of the training set by subtracting (adding) the mean target from each target point. This corresponds to an old heuristic among RBF practitioners; it follows that the hidden-to-output weights should be initialized from a zero-mean distribution. Alternatively, a bias unit could be employed, but this adds another parameter to the training process.

5.5.4 The Over-realizable Case

In real-world problems the exact form of the data-generating mechanism is rarely known. This leads to the possibility that the student may be overly powerful, in that it is capable of fitting surfaces more complicated than that of the true teacher. It is important to gain insight into how architectures will respond given such a scenario in order to be confident that they can be used successfully when the true teacher is unknown.



(a) (b)

Figure 5.2: The exactly realizable scenario defined by a teacher network with a mixture of positive and negative TBFs. Three SBFs learn a graded, uncorrelated teacher of three TBFs with $T_{00} = 0.5, T_{11} = 1.0$ and $T_{22} = 1.5$. $w_0^0 = 1, w_1^0 = -1, w_2^0 = 1$. (a) describes the evolution of the generalization error for this case and presents for comparison the evolution in the case of all positive TBFs, while (b) shows the evolution of the overlaps between student and teacher centres $\boldsymbol{R}$.

Intuitively, one might expect that a student that is well-matched to the teacher will learn faster than one which is overly powerful. Figure 5.3(a) shows two tasks, each of which compares the over-realizable scenario with the well-matched case. The first task, consisting of 3 TBFs, is identical to that detailed in section 5.5.2, and hence has only positive targets. The performance of a well-matched student of 3 SBFs is compared with an over-realizable scenario in which 5 SBFs learn the 3 TBFs. Comparison of the evolution of generalization error between these learning scenarios is shown in figure 5.3(a); the solid curve represents the well-matched scenario, while the dot-dash curve illustrates the over-realizable scenario. The length of the symmetric phase is significantly increased with the overly-powerful student.

The length of the convergence phase is also increased.

The second task deals with the alternative scenario in which one TBF has a negative hidden-to-output weight; the task is identical to that defined in section 5.5.3, and the student initial conditions are again as specified in section 5.5.2. In figure 5.3(a) the evolution of generalization error for both the over-realizable scenario (dashed curve) in which 5 SBFs learn 3 TBFs, and the corresponding well-matched case in which 3 SBFs learn 3 TBFs (dotted curve) is shown. There is no well-defined symmetric case, due to the inherent asymmetry of the task. The convergence phase is again greatly increased in length; this appears to be a general feature of the over-realizable scenario.

Given that the student is overly powerful, there appears to be, *a priori*, several remedies available to the student. It could: eliminate the excess nodes, form cancellation pairs (in which two students exactly cancel one another), or devise more complicated fitting schemes.

To examine the actual responses of the student, the evolution of the overlaps between student and teacher and of the hidden-to-output weights for the particular scenario described by the second trial detailed above are presented in figures 5.3(b) and 5.3(c) respectively. Looking first at figure 5.3(c), it is apparent that w_3 approaches zero (short-dashed curve), indicating that SBF 3 is entirely eliminated during training. Thus 4 SBFs remain to emulate 3 TBFs. The negative TBF 1 is exactly emulated by SBF 0, as $T_{11} = 1, w_1^0 = -1$ and $R_{01} = 1, w_0 = -1$ (solid curve on both figure 5.3(b) and 5.3(c)), while, similarly, SBF 2 exactly emulates TBF 2 (long-dashed curve, both figures). This leaves SBF 1 and SBF 4 to emulate TBF 0. Looking at figure 5.3(c), dotted and dot-dash curves, both student hidden-to-output weights

approach 0.5, exactly half that of the hidden-to-output weight of TBF 0; looking at 5.3(b), both SBFs have 0.5 overlap with TBF 0. This indicates that the sum of both students emulates TBF 0. Thus elimination and fitting involving the non-cancelling combination of nodes was found; in these trials and many others, no pairwise cancellation was found. One presumes that this could be induced by very careful selection of the initial conditions, but that it is not found under normal circumstances.

5.5.5 Analysis of the Symmetric Phase

The symmetric phase, in which there is no specialization of the hidden units, can be analyzed in the realizable case by employing a few simplifying assumptions. It is a phenomenon that is predominantly associated with small values of η/N, so terms of $(\eta/N)^2$ are neglected. The hidden-to-output weights are clamped to $+1$. The teacher is taken to be *isotropic*: TBF centres have *identical norms* of 1, each having no overlap with the others, therefore $T_{uv} = \delta_{uv}$. This has the result, also observed in the numerical solutions, that the student norms Q_{bb} are very similar in this phase, as are the student-student correlations, so $Q_{bb} \equiv Q$ and $Q_{bc,b\neq c} \equiv C$ where Q becomes the square norm of the SBFs, and C is the overlap between any two different SBFs.

Following the geometric argument of Saad and Solla (1995b), in the symmetric phase, the SBF centres are confined to the subspace spanned by the TBF centres. Since $T_{uv} = \delta_{uv}$, the SBF centres can be written in the orthonormal basis defined by the TBF centres, with the components being the overlaps $\boldsymbol{R}$: $m_b = \sum_{u=1}^{M} R_{bu}\boldsymbol{n}_u$. As the teacher is isotropic, the overlaps are independent of both b and u and thus can be written in terms of a single parameter R. Further, this reduction to a single overlap parameter leads to

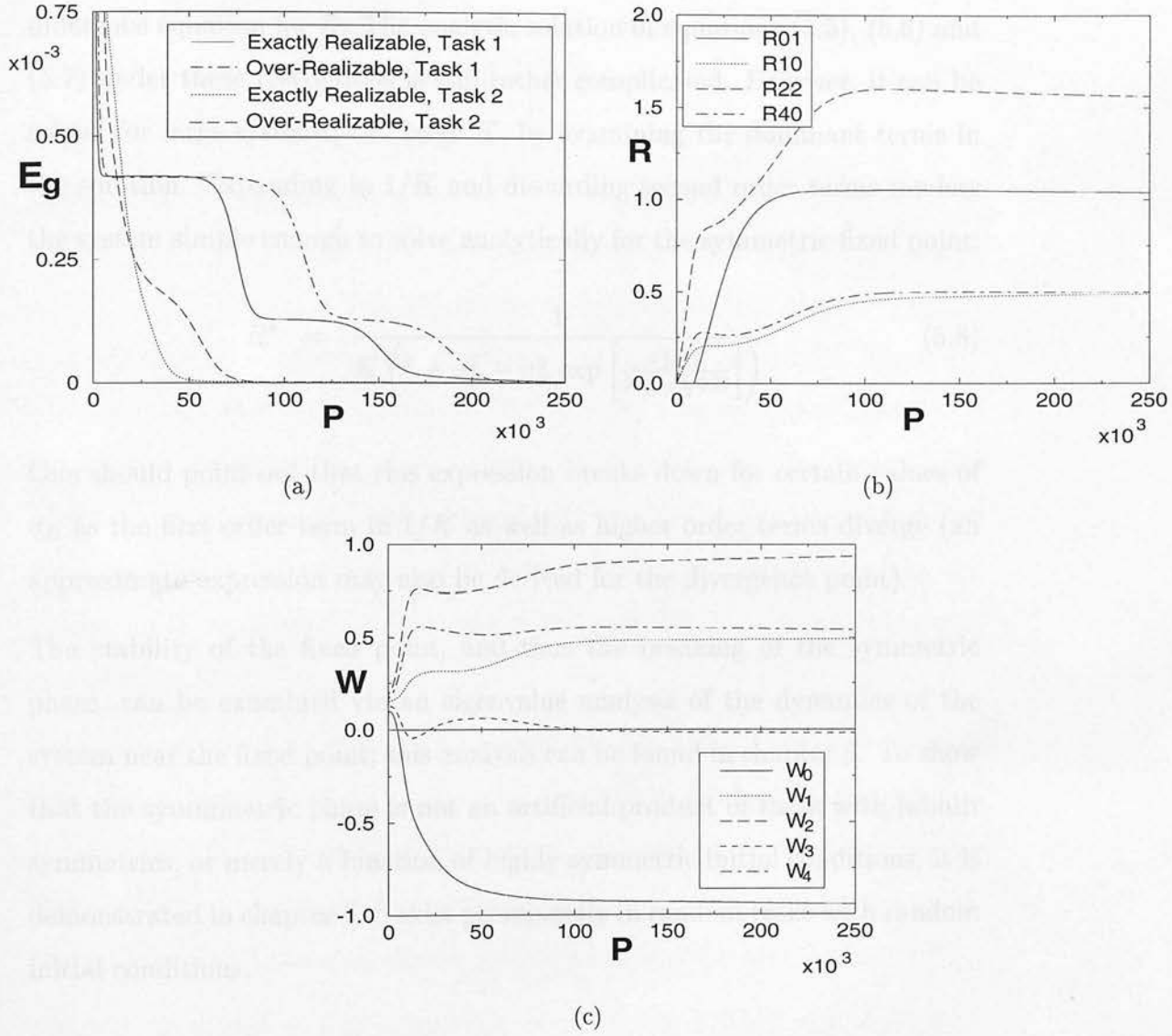


Figure 5.3: The over-realizable scenario. Figure (a) describes the evolution of the generalization error in two tasks; each task is learnt by a well-matched student (exactly realizable), and an overly-powerful student (over-realizable). Figures (b) and (c) show the evolution of the overlaps $\boldsymbol{R}$ and the hidden-to-output weights $\boldsymbol{w}$ for the over-realizable case in the second task, in which the teacher RBF includes a mixture of positive and negative hidden-to-output weights. In this scenario, five SBFs learn a graded, uncorrelated teacher of three TBFs with $T_{00} = 0.5, T_{11} = 1.0$ and $T_{22} = 1.5$. $w_0^0 = 1, w_1^0 = -1, w_2^0 = 1$.

$Q = C = MR^2$, so the evolution of the overlaps can be described as a single difference equation for R. The analytic solution of equations (5.5), (5.6) and (5.7) under these restrictions is still rather complicated. However, it can be solved for large systems, i.e. large K, by examining the dominant terms in the solution. Expanding in $1/K$ and discarding second order terms renders the system simple enough to solve analytically for the symmetric fixed point:

$$R^* = \frac{1}{K\left(1 + \sigma_B^2 - \sigma_B^2 \exp\left[\frac{\sigma_B^2+1}{2\sigma_B^2(\sigma_B^2+2)}\right]\right)} \tag{5.8}$$

One should point out that this expression breaks down for certain values of σ_B as the first order term in $1/K$ as well as higher order terms diverge (an approximate expression may also be derived for the divergence point).

The stability of the fixed point, and thus the breaking of the symmetric phase, can be examined via an eigenvalue analysis of the dynamics of the system near the fixed point; this analysis can be found in chapter 6. To show that the symmmetric phase is not an artificial product of tasks with inbuilt symmetries, or merely a function of highly symmetric initial conditions, it is demonstrated in chapter 6 to exist prominently in random tasks with random initial conditions.

5.5.6 Analysis of the Convergence Phase

To gain insight into the convergence of the on-line gradient descent process in a realizable scenario, a similar simplified learning scenario to that utilized in the symmetric phase analysis was employed. The hidden-to-output weights are again fixed to $+1$, and the teacher is defined by $T_{uv} = \delta_{uv}$.

The scenario can be extended to adaptable hidden-to-output weights; this is presented in chapter 6. As in the symmetric phase, the fact that $T_{uv} = \delta_{uv}$ allows the system to be reduced to four adaptive quantities: $Q \equiv Q_{bb}, C \equiv Q_{bc,b\neq c}, R \equiv R_{bb}$ and $S \equiv R_{bc,b\neq c}$.

Linearizing this system about the known fixed point of the dynamics, $Q = 1, C = 0, R = 1, S = 0$, yields an equation of the form $\Delta\mathbf{x} = \boldsymbol{A}\mathbf{x}$ where $\mathbf{x} = \{1 - R, 1 - Q, S, C\}$ is the vector of deviations from the fixed point. The eigenvalues of the matrix $\boldsymbol{A}$ control the converging system: these are presented in figure 5.4(a) for $K = 10$. In every case examined, there is a single critical eigenvalue λ_c that controls the stability and convergence rate of the system (shown in bold), a non-linear subcritical eigenvalue, and two subcritical linear eigenvalues. The value of η at $\lambda_c = 0$ determines the maximum learning rate for convergence to occur; for $\lambda_c > 0$ the fixed point is unstable.

The convergence of the overlaps is controlled by the critical eigenvalue, therefore, the value of η at the single minimum of λ_c determines the optimal learning rate (η_{opt}) in terms of the fastest convergence of the system to the fixed point. An examination of globally optimal learning rates for the SCM, calculated via variational methods, can be found in (Saad and Rattray, 1997).

Examining η_c and η_{opt} as a function of K (figure 5.4(b)), one finds that both quantities scale as $1/K$; the maximum and optimal learning rates are inversely proportional to the number of hidden units of the student. Numerically, the ratio of η_{opt} to η_c is approximately $2/3$.

Finally, the relationship between basis function width and η_c is plotted in figure 5.4(c). When the widths are small, η_c is very large as it becomes unlikely that a training point will activate any of the basis functions. For

$\sigma_B^2 > \sigma_\xi^2, \eta_c \sim 1/\sigma_B^2$.

5.6 Summary

On-line learning, in which the adaptive parameters of the network are updated at each presentation of a data point, was examined for the RBF using gradient descent learning. The analytic method presented allows the calculation of the evolution of generalization error and of the specialization of the hidden units.

This method was used to elucidate the stages of training and the role of the learning rate. There are four stages of training: a short transitory phase in which the adaptive parameters move from the initial conditions to the symmetric phase; the symmetric phase itself, characterized by lack of differentiation amongst hidden units; a symmetry-breaking phase in which the hidden units become specialized, and a convergence phase in which the adaptive parameters reach their final values asymptotically. Three regimes were found for the learning rate: small, giving unnecessarily slow learning, intermediate, leading to fast escape from the symmetric phase and convergence to the correct target, and too large, which results in a divergence of SBF norms and failure to converge to the correct target.

Examining the exactly realizable scenario, it was shown that employing both positive and negative targets leads to much faster symmetry breaking. The over-realizable case was also studied, showing that over-realizability extends both the length of the symmetric phase and that of the convergence phase.

The symmetric phase for realizable scenarios was analyzed and the value of the overlaps at the symmetric fixed point found.

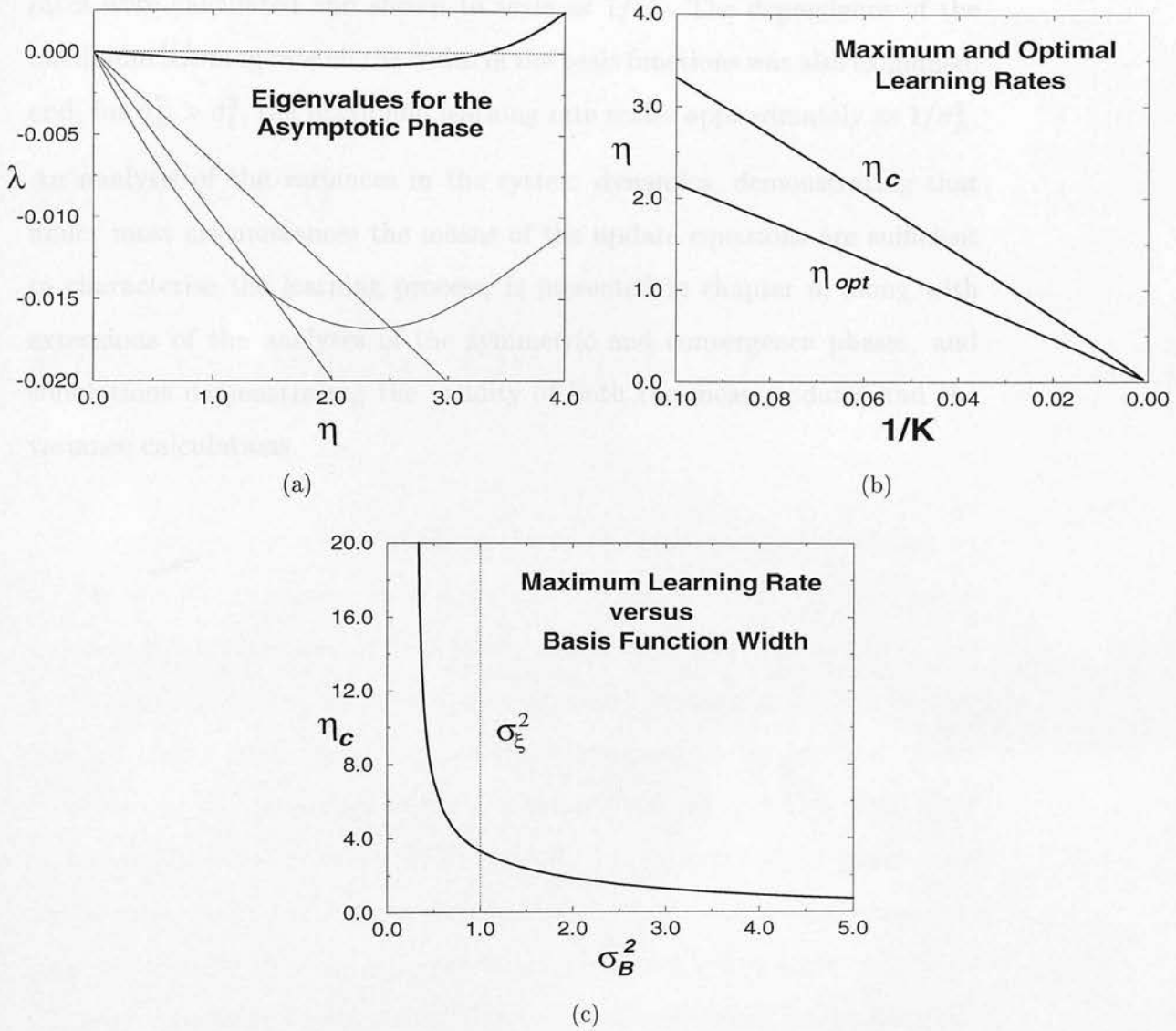


Figure 5.4: Convergence and symmetric phases. Figure (a) shows the eigenvalues controlling the dynamics of the system for the convergence phase (detailed in section 5.5.6), linearized about the asymptotic fixed point in the realizable case, as a function of η. The critical eigenvalue is shown in bold. Figure (b) denotes the maximum and optimal convergence phase learning rates, found from the critical eigenvalue; these quantities scale as $1/K$, while figure (c) shows the maximum convergence phase learning rate as a function of basis function width.

The convergence phase was also studied; both maximum and optimal learning rates were calculated and shown to scale as $1/K$. The dependence of the maximum learning rate on the width of the basis functions was also examined, and, for $\sigma_B^2 > \sigma_\xi^2$, the maximum learning rate scales approximately as $1/\sigma_B^2$.

An analysis of the variances in the system dynamics, demonstrating that under most circumstances the means of the update equations are sufficient to characterise the learning process, is presented in chapter 6, along with extensions of the analyses of the symmetric and convergence phases, and simulations demonstrating the validity of both the mean updates and the variance calculations.

Chapter 6

Extensions to On-line Learning

This chapter significantly expands on the on-line learning analysis of chapter 5. The use of the mean update equations is justified via an analytic calculation of the average fluctuations in the system dynamics; the results of this analysis are confirmed by experiments of actual learning in RBF networks. The behaviour of the system in the symmetric phase is further understood by analysing the properties of the symmetric fixed point, which sheds light on major differences between RBFs and MLPs. The convergence phase analysis is extended to the more realistic situation of adaptive hidden-to-output weights. Finally, further simulations are presented which show the excellent correspondence between the theoretical results obtained by iterating the mean update equations and results from training real RBF networks.

6.1 System Dynamics

The system dynamics derived in chapter 5 are employed unchanged. Recapping, the gradient descent rule $\boldsymbol{m}_b^{p+1} = \boldsymbol{m}_b^p + \frac{\eta}{N\sigma_B^2}\delta_b(\boldsymbol{\xi} - \boldsymbol{m}_b)$ (where

$\delta_b = (f_T - f_S)w_b s_b$ and learning rate η is explicitly scaled with $1/N$) is used as the basis from which expressions for the time evolution of the mean overlaps of $\boldsymbol{Q}$ and $\boldsymbol{R}$ are derived. The hidden-to-output weights are treated similarly, except that the learning rate is scaled with $1/K$ rather than $1/N$. The averaged equations for $\Delta\boldsymbol{Q}, \Delta\boldsymbol{R}$ and $\Delta\boldsymbol{w}$ can be found in appendix B.

6.2 Variance and the Thermodynamic Limit

Other recent work in this area (Biehl and Schwarze, 1995; Saad and Solla, 1995a,b; Riegler and Biehl, 1995) has relied upon the thermodynamic limit (i.e., $P \to \infty, N \to \infty$ and $P/N = \alpha$, where α is finite). Taking this limit allows one to ignore fluctuations in the updates of the means of the overlaps due to the randomness of the training examples, and permits the difference equations of gradient descent to be considered as differential equations. As discussed in chapter 5, the thermodynamic limit is hugely artificial for local RBFs as the activation is localized. The price paid for not taking this limit is that one has no *a priori* justification for ignoring the fluctuations in the update of the adaptive parameters due to the randomness of the training example.

By making assumptions as to the form of these fluctuations, it is possible to derive equations describing their evolution; the method is mentioned in (Heskes and Kappen, 1991) and also in (Barber *et al.*, 1996) for the simpler case of the SCM, and is based on the Van Kampen small fluctuation expansion (Kampen, 1992); it is extended in this thesis to deal with adaptive hidden-to-output weights (see also Riegler and Biehl, 1995).

To quantify the effect of the variances, a set of dynamical equations will

be derived, parallel to those representing the dynamics of the means, for describing the dynamics of the variances. To simplify the explanation, the update equations (5.5), (5.6) and (5.7) are cast here into a general form, where a represents a generic system parameter and the scaling parameter L_a is set to N for $\boldsymbol{Q}$ and $\boldsymbol{R}$, and to K for w. As the scaled learning rate is usually small, the focus will be on first order terms in η/L_a, which dominate the dynamics, and update terms of order $(\eta/L_a)^2$ will be ignored.

Thus the update equations can all be represented by a single generic equation:

$$a^{p+1} = a^p + \frac{\eta}{L_a} F_a \tag{6.1}$$

It is then assumed (similar to Barber *et al.*, 1996) that the update function F and the parameter a can be written in terms of a mean and fluctuation such that:

$$F_a = \overline{F}_a + \widehat{F}_a \quad \text{and} \quad a = \overline{a} + \sqrt{\tfrac{\eta}{L_a}}\,\widehat{a} \tag{6.2}$$

where $\overline{a}$ denotes an average value and $\widehat{a}$ represents a fluctuation due to the randomness of the example. The bias term that arises from the Van Kampen expansion is neglected (as in Heskes and Kappen, 1991) as it is typically an order $\sqrt{\eta/L}$ smaller than the fluctuation term; it is caused by the interaction between the non-linear learning rules and the fluctuations, whereby if a fluctuation in a particular direction in parameter space does not result in the same restoring effect as that in the opposite direction, there is a net bias (see Wiegerinck and Heskes, 1996).

Combining eqns (6.1) and (6.2), and averaging with respect to the input distribution, gives a set of coupled difference equations which describe the

evolution of the variances:

$$\Delta\left\langle \widehat{a}\widehat{b}\right\rangle = \frac{\eta}{\sqrt{L_a L_b}}\left(\sum_c \left\langle \widehat{a}\widehat{c}\right\rangle \frac{\partial \overline{F}_b}{\partial \overline{c}} + \sum_c \left\langle \widehat{b}\widehat{c}\right\rangle \frac{\partial \overline{F}_a}{\partial \overline{c}} + \left\langle \widehat{F}_a \widehat{F}_b \right\rangle\right) \tag{6.3}$$

The update to the variances is composed of an instantaneous fluctuation which depends only on the current example, denoted by $\left\langle \widehat{F}_a \widehat{F}_b \right\rangle$, and a set of terms dependent on the current variances. Thus eqn. (6.3) describes the evolution of the *cumulative* variances, not merely those due to the randomness of the current example.

Applying this general method to each pair of adaptive quantities allows the evolution of the variances for the entire system to be calculated. The averages are again Gaussian and so are analytically tractable; the expressions that result for the instantaneous variances are given in appendix B.

It has been shown that the variances must vanish asymptotically for realizable cases (Heskes and Kappen, 1991); the equations derived above are employed in section 6.3.3 to demonstrate that the variances are small enough throughout the evolution of the system to allow a description of the system dynamics in terms of the evolution of the means.

6.3 Analysing the Learning Process

6.3.1 Analysing the Symmetric and Symmetry-Breaking Phases

The symmetric phase analysis of chapter 5 is extended in this section to examine the dynamics of the evolving system near the symmetric fixed point.

The same assumptions employed previously are used here: terms of η^2 are neglected as the symmetric phase is predominantly associated with small η; the teacher is isotropic ($T_{uv} = \delta_{uv}$), so the student norms Q_{bb} are similar (denoted by Q) as are the student correlations Q_{bc} (denoted by C).

In the symmetric phase, the SBF centres are mostly confined to the subspace spanned by the TBF centres. Since $T_{uv} = \delta_{uv}$, the SBF centres can be written in the orthonormal basis defined by the TBF centres, with the components being the overlaps $\boldsymbol{R}$: $\boldsymbol{m}_b = \sum_{u=1}^{M} R_{bu}\boldsymbol{n}_u$. As the teacher is isotropic, the overlaps are independent of both b and u and thus can be written in terms of a single parameter R. Further, this reduction to a single overlap parameter leads to $Q = C = MR^2$, so the evolution of the overlaps can be described as a single difference equation for R. The analytic solution of this equation, giving the value of R (and thus Q, C and S) at the symmetric fixed point is given in eqn (5.8); fixed point values will be denoted like R^*.

The stability of the fixed point, and thus the breaking of the symmetric phase, can be examined via an eigenvalue analysis of the dynamics of the system near the fixed point. The equations of motion (5.5), (5.6) are mapped to equations of deviations from the symmetric fixed point via $r = R - R^*, s = S - S^*, q = Q - Q^*, c = C - C^*$. Remembering the geometrical argument above, the student weight vectors can be expanded in terms of the student-teacher overlaps; as the calculation is in the small η regime, components which are orthogonal to the space spanned by the teacher vectors, $\boldsymbol{m}_b^{\perp}$, may be neglected, so that the student norms Q and overlaps C are completely determined by the student-teacher overlaps. Writing these overlaps as: $R_{bu} = R\delta_{bu} + S(1 - \delta_{bu})$ gives the relations $Q = R^2 + S^2(K - 1)$ and $C = 2RS + S^2(K - 2)$. If these relations are expanded to first order in the deviations r and s, it can be seen that $q = c = 2R^*(r + s(K - 1))$, so that $Q^* = C^*$ is

preserved to first order; this is also consistent with the truncated equations of motion if they too are expanded to first order. Thus the dynamical quantities reduce to three: r, s and c.

Performing an eigenvalue analysis on the resulting system reveals one dominant positive eigenvalue (λ) that scales with K and represents a perturbation which breaks the symmetries between the hidden units by amplifying asymmetries in the initial conditions (see (Biehl *et al.*, 1996) for a detailed analysis of this for the SCM); the remaining modes, which also scale with K, are irrelevant as they preserve the symmetry. This result is in contrast to that for the SCM (Saad and Solla, 1995b), in which the dominant eigenvalue scales with $1/K$. This implies that for RBFs the more hidden units in the network, the *faster* the symmetric phase is escaped, resulting in negligible symmetric phases for large systems, while in SCMs the opposite is true; this result has been confirmed by simulations in which, for the RBF, the length of symmetric phase is found to scale as $1/K$. This difference across architectures is caused by the contrast between the localized nature of the basis function in the RBF network and the global nature of sigmoidal hidden nodes in SCM. In the SCM case, small perturbations around the symmetric fixed point result in relatively small changes in error since the sigmoidal response changes very slowly as one modifies the weight vectors. On the other hand, the Gaussian response decays exponentially as one moves away from the centre, so small perturbations around the symmetric fixed point result in massive changes that drive the symmetry breaking. When K increases the error surface looks very rugged emphasising the peaks and increasing this effect, in contrast to the SCM case where more sigmoids means a smoother error surface. Note that this result applies to *realizable* tasks, in which the student is of the same complexity as the teacher, and to isotropic teachers. Whether the length

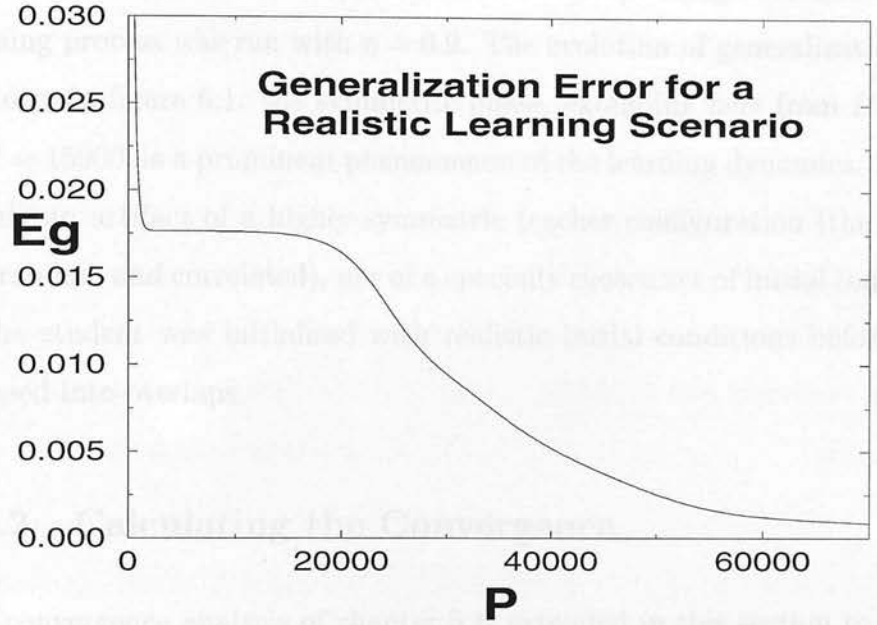


Figure 6.1: Generalization error for a realistic learning task showing the existence and importance of the symmetric phase. A student of 10 hidden units learns a realizable task in which the teacher also has 10 hidden units, with $N = 5$. The symmetric phase is a prominent feature of the learning dynamics.

of the symmetric phase scales as $1/K$ for more complex teachers remains a subject for research.

This does not mean that the symmetric phase can be ignored for realistically-sized networks, however. Even with a teacher that is not particularly symmetric, this phase can play a significant role in the learning dynamics. To demonstrate this, a teacher RBF of 10 hidden units with $N = 5$ was constructed with the teacher centres generated from a Gaussian distribution $\mathcal{N}[0, 0.5]$. Note that this teacher must be correlated as the number of centres is larger than the input dimension. A student network, also of 10 hidden units, was constructed with all weights initialised from $\mathcal{N}[0, 0.05]$.

The networks were then mapped into the corresponding overlaps, and the learning process was run with $\eta = 0.9$. The evolution of generalization error is shown in figure 6.1: the symmetric phase, extending here from $P = 2000$ to $P = 15000$, is a prominent phenomenon of the learning dynamics. It is not merely an artifact of a highly symmetric teacher configuration (the teacher was random and correlated), nor of a specially chosen set of initial conditions, as the student was initialised with realistic initial conditions before being mapped into overlaps.

6.3.2 Calculating the Convergence

The convergence analysis of chapter 5 is extended in this section to include adaptive hidden-to-output weights.

Again an isotropic teacher is used, defined by $T_{uv} = \delta_{uv}$ and $w_u^0 = 1$. This means the evolution of each student hidden unit will be very similar, so the evolving system can be simplified to 5 adaptive variables Q, C, R, S and w, defined by: $Q_{bc} = Q\delta_{bc} + C(1 - \delta_{bc}), R_{bu} = R\delta_{bu} + S(1 - \delta_{bu})$ and $w_b = w$; these quantities are controlled by equations (5.5), (5.6) and (5.7). Note that the variances are not expected to play a significant role in defining the maximal and optimal learning rates as they have been shown to vanish in the asymptotic regime.

Linearizing these equations about the known fixed point of the dynamics, $Q = 1, C = 0, R = 1, S = 0, w = 1$ yields the eigenvalues controlling the rate of convergence and the stability. There is a single (non-linear in η) critical eigenvalue, λ_1, which controls stability, a linear eigenvalue, λ_2, which can influence convergence rate, and three further eigenvalues which play no significant role, being much smaller for all values of η. The eigenvalues are

illustrated in figure 6.2(a) for a network of 10 hidden units with input dimension $N = 10$. The maximum learning rate, defined by the crossing of the zero line, can be seen to be controlled solely by λ_1; note that this maximum only applies during convergence, not necessarily during the other phases of learning. The theory predicts a maximum learning rate of $\eta = 33$ for this scenario; the accuracy of the method was tested by training real RBF networks by initializing them near the known fixed point, and determining the value of η at which convergence failed to occur, which in this case was $\eta = 32.3$ with standard deviation of 0.8.

The rate of convergence, defined for particular η by the smaller of λ_1 and λ_2, is optimized either by setting η to the minimum of λ_1 or to the intersection of λ_1 with λ_2, depending on the exact learning scenario (e.g., for other teacher vector lengths or basis widths).

It is interesting to compare the convergence of the system with adaptive hidden-to-output weights to that where the hidden-to-output weights are fixed (chapter 5). Figure 6.2(b) shows the two significant eigenvalues for both cases in identical scenarios. λ_1 is unchanged, so the maximum learning rate is unaffected and is therefore a function of the hidden layer, not the output layer (this is also true for the MLP (Riegler and Biehl, 1995)). This implies that the exact form of the learning rule for the hidden-to-output weights is irrelevant to the maximum learning rate. Further, even if the correct values of the hidden-to-output weights were known in advance, this would not affect the maximum learning rate. Note that with fixed hidden-to-output weights, the gradient of λ_2 becomes much steeper and in fact does not affect the rate of convergence, which is controlled solely by λ_1.

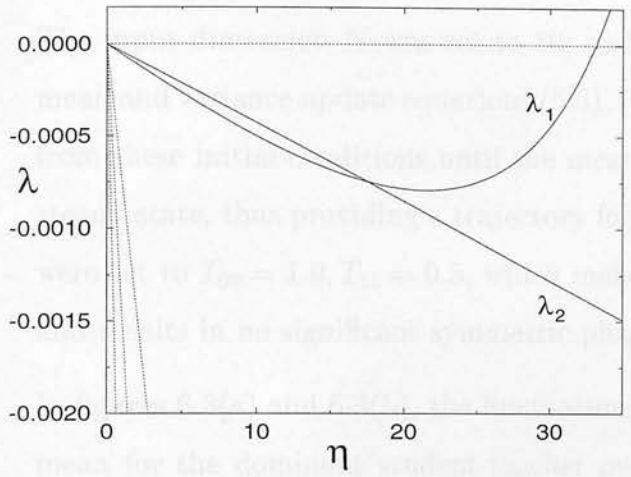
The scaling of the maximum and optimal learning rates with the number

of hidden units can also be found. For both fixed and adaptive hidden-to-output weights, the maximum learning rate scales as $1/K$. For fixed hidden-to-output weights, the optimal learning rate also scales as $1/K$, while for adaptive hidden-to-output weights, the situation is more complicated. In parameter regions where the convergence rate is optimized by minimising λ_1, the optimal learning rate again scales as $1/K$; however, in regions where optimization is achieved by finding the intersection of λ_1 and λ_2, η changes at a slower rate than $1/K$. These effects are illustrated in figure 6.2(c), in which maximum and optimal learning rates are plotted against $1/K$. Note that as K increases, $\eta_{\rm opt}$ approaches $\eta_{\rm C}$ rapidly for the adaptive hidden-to-output case (λ_2 becomes less steep), implying that it becomes difficult to optimize the process and still obtain convergence to the correct fixed point.

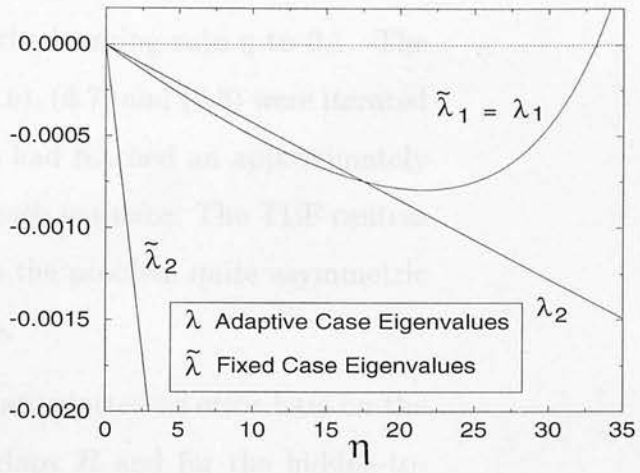
6.3.3 Quantification of the Variance

To demonstrate that it is reasonable to consider only the mean of the updates of the system parameters, results are presented which quantify the effect of the variance for a typical case, showing that its contribution is negligible in comparison with the mean values. In pathological cases in which the task and the initial conditions of the system are highly symmetric, it is possible to obtain variances which are much larger than those which typically occur - this issue is explored for the SCM by Barber *et al.* (1996).

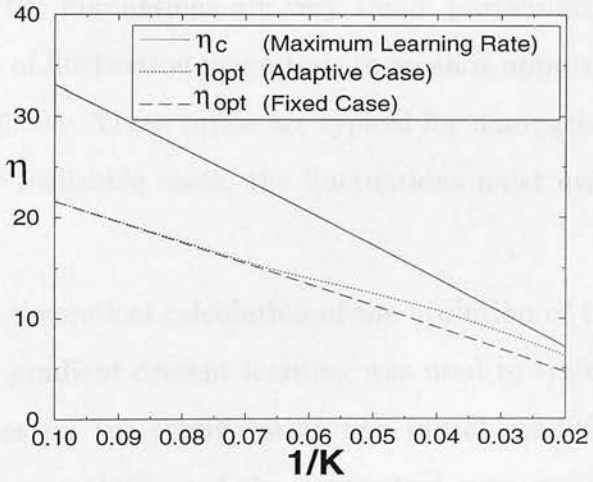
In order to quantify the variances, a training scenario is constructed in which a student network comprising two SBFs is trained on examples generated by a two node teacher. The initial conditions were constructed by randomly initialising the weights of an RBF network by drawing each input-to-hidden and hidden-to-output weight from U[0,0.1], and then mapping the network



(a)



(b)



(c)

Figure 6.2: Convergence Phase with Adaptive Hidden-to-Output Weights. Figure (a) shows the eigenvalues for the system with adaptive hidden-to-output weights. Only λ_1 and λ_2 are significant; λ_1 controls the maximum learning rate, while λ_2 can influence the optimal learning rate. Figure (b) compares the eigenvalues for systems with adaptive and fixed hidden-to-output weights, showing that λ_1 is unaffected. Figure (c) shows the scaling of the maximum and optimal learning rates with K. The maximum learning rate η_c scales with $1/K$; for fixed hidden-to-output weights, the optimal learning rate η_{opt} also scales with $1/K$, while for adaptive weights, η_{opt} rapidly approaches η_c.

into the appropriate system parameters, so as to provide realistic conditions. The input dimension N was set to 10, and the learning rate η to 0.1. The mean and variance update equations (5.5), (5.6), (5.7) and (6.3) were iterated from these initial conditions until the means had reached an approximately steady state, thus providing a trajectory for each variance. The TBF centres were set to $T_{00} = 1.0, T_{11} = 0.5$, which makes the problem quite asymmetric and results in no significant symmetric phase.

In figures 6.3(a) and 6.3(b), the fluctuations are plotted as error bars on the mean for the dominant student-teacher overlaps $\boldsymbol{R}$ and for the hidden-to-output weights $\boldsymbol{w}$ (fluctuation magnitudes for $\boldsymbol{Q}$ are very similar to those of $\boldsymbol{R}$). The magnitudes of the fluctuations are very small, particularly so for $\boldsymbol{R}$. For $\boldsymbol{w}$, the peak ratio of fluctuation magnitude to mean is approximately 0.012, while for $\boldsymbol{R}$, it is 0.008. These ratios are typical for non-pathological scenarios. Note that for realizable cases, the fluctuations must eventually disappear.

To demonstrate that the theoretical calculation of the evolution of the variances gives valid results, gradient descent learning was used to train actual RBF networks 1000 times for the configuration and initial conditions described above. The average evolutions of the parameters were employed to calculate empirical fluctuations about the means. The results of this are plotted in figures 6.3(c) and 6.3(d), in which the theoretical fluctuations are shown versus the simulation fluctuations - it can be seen that there is very good agreement between the theory and simulation. The slight discrepancy up to about $P = 1.5 \times 10^6$ is believed to be due to the fact that terms of η^2 are discarded in the theory.

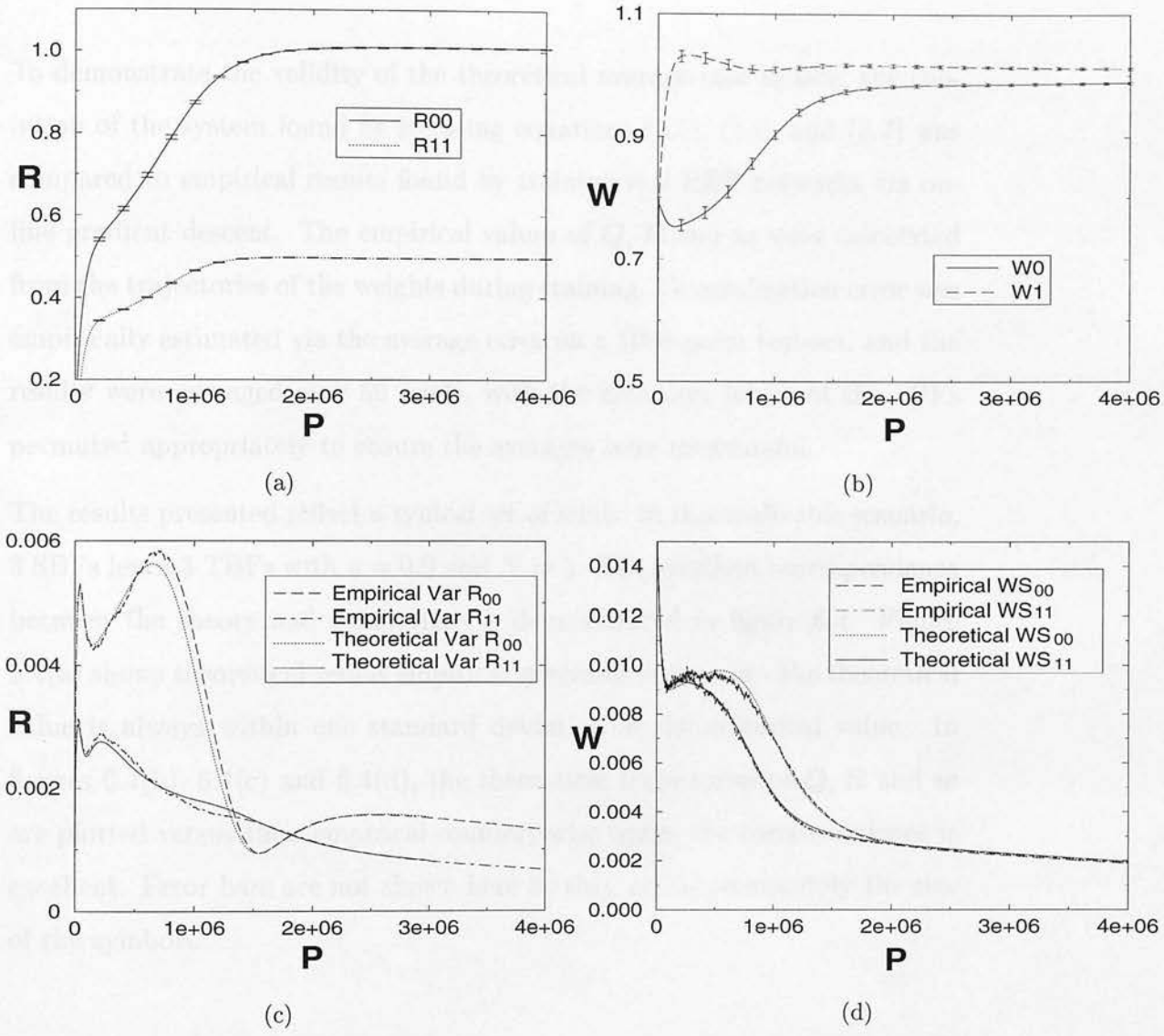


Figure 6.3: Quantification of the Variances. Figures (a) and (b) show the theoretical variances, plotted as errorbars on the mean, for the dominant overlaps R_{00} and R_{11} and for the hidden-to-output weights w_0 and w_1 respectively, for a realizable task involving two SBFs learning two TBFs. The fluctuations are negligible; this is typically true, unless the task and initial conditions are highly symmetric. Figures (c) and (d) compare the theoretical variances to those from simulations in which RBFs were trained 1000 times on the above task. The variances for the dominant overlaps and hidden-to-output weights are shown, and it can be seen that there is an excellent correspondence.

6.3.4 Simulations

To demonstrate the validity of the theoretical average-case results, the evolution of the system found by iterating equations (5.5), (5.6) and (5.7) was compared to empirical results found by training real RBF networks via on-line gradient descent. The empirical values of $\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{w}$ were calculated from the trajectories of the weights during training. Generalization error was empirically estimated via the average error on a 1000-point test-set, and the results were averaged over 50 trials, with the arbitrary labels of the SBFs permuted appropriately to ensure the averages were meaningful.

The results presented reflect a typical set of trials: in this realizable scenario, 3 SBFs learn 3 TBFs with $\eta = 0.9$ and $N = 5$. The excellent correspondence between the theory and simulations is demonstrated in figure 6.4. Figure 6.4(a) shows theoretical versus empirical generalization error - the theoretical value is always within one standard deviation of the empirical value. In figures 6.4(b), 6.4(c) and 6.4(d), the theoretical trajectories of $\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{w}$ are plotted versus their empirical counterparts; again, the correspondence is excellent. Error bars are not shown here as they are approximately the size of the symbols.

6.4 Summary

On-line learning using the gradient descent algorithm has been examined for the RBF by employing a method which allows the calculation of generalization error as well as the elucidation of the features of the learning process, such as the specialization of the hidden units.

The symmetric phase was analysed (for the realizable case), and the value of

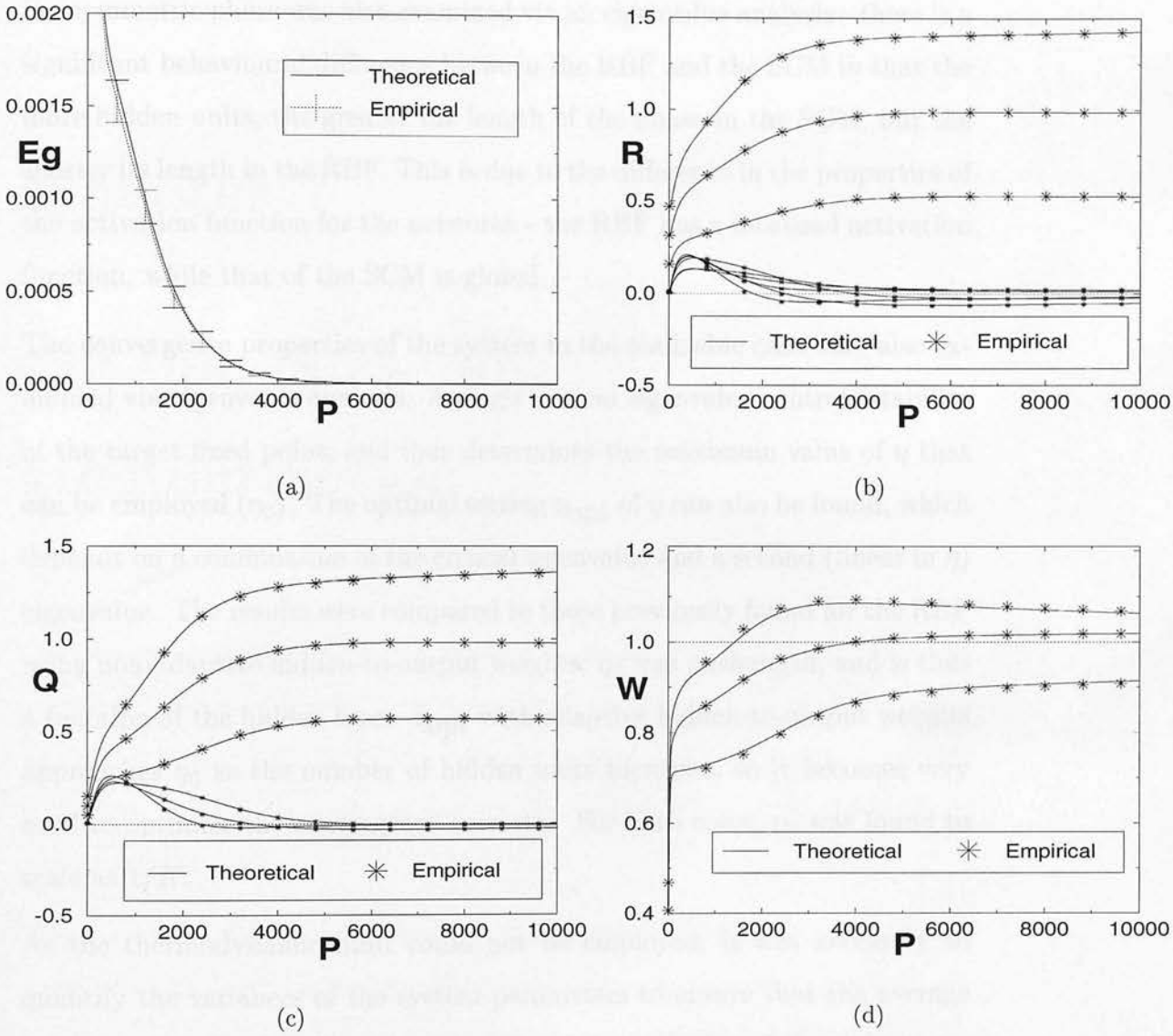


Figure 6.4: Comparison of theoretical results with simulations. The simulation results are averaged over 50 trials; the labels of the student hidden units were permuted where necessary to make the averages meaningful. Empirical generalization error was approximated with the test error on a 1000 point test set. Error bars on the simulations are at most the size of the larger asterisks for the overlaps (figures (b) and (c)), and at most twice this size for the hidden-to-output weights (figure (d)). Input dimensionality $N = 5$, learning rate $\eta = 0.9$, input variance $\sigma_\xi^2 = 1$ and basis function width $\sigma_B^2 = 1$.

the system parameters at the symmetric fixed point found. The breaking of the symmetric phase was also examined via an eigenvalue analysis - there is a significant behavioural difference between the RBF and the SCM in that the more hidden units, the greater the length of the phase in the SCM, but the shorter its length in the RBF. This is due to the difference in the properties of the activation function for the networks - the RBF has a localized activation function, while that of the SCM is global.

The convergence properties of the system in the realizable case were also examined via eigenvalue analysis. A single critical eigenvalue controls stability of the target fixed point, and thus determines the maximum value of η that can be employed (η_c). The optimal setting η_{opt} of η can also be found, which depends on a combination of the critical eigenvalue and a second (linear in η) eigenvalue. The results were compared to those previously found for the RBF using non-adaptive hidden-to-output weights; η_c was unchanged, and is thus a function of the hidden layer. η_{opt} with adaptive hidden-to-output weights approaches η_c as the number of hidden units increases, so it becomes very hard to optimize the convergence correctly. For both cases, η_c was found to scale as $1/K$.

As the thermodynamic limit could not be employed, it was necessary to quantify the variances of the system parameters to ensure that the average value was meaningful. Equations describing the evolution of these variances were derived, and it was shown that, for a typical case, the variances are small. The equations for the evolution of the means and the variances were shown to be valid descriptions of the real system via simulations.

Chapter 7

On-line Noise and Regularization

The on-line learning framework studied in chapters 5 and 6 addresses the properties of learning in the noise-free case. This chapter extends the analysis of on-line learning to the more realistic scenario in which the training data is corrupted by noise, and also adapts the framework to allow the study of regularization.

The situation of learning from corrupted examples in neural networks has been examined from a variety of perspectives, including the Bayesian approach employed in chapters 3 and 4, equilibrium statistical mechanics (Watkin *et al.*, 1993, and references therein), which has been used primarily to investigate simple networks, and non-equilibrium approaches (Amari *et al.*, 1996). Noisy on-line learning in the SCM has recently been examined from a similar perspective to that considered here (Saad and Solla, 1997).

The student-teacher framework employed previously is once again exploited in this chapter. Two classes of noise are examined: additive Gaussian output

noise, which is added to the output of the teacher to corrupt the dataset, and Gaussian input noise, which is applied to the inputs of the teacher, rather than directly to the dataset; for RBFs, input noise can also be seen as a type of model noise in which the basis function positions of the teacher are corrupted. The addition of each type of noise affects the system dynamics differently, although at low noise levels the familiar phases of learning described in chapter 5 remain qualitatively similar.

The issue of regularization via weight decay (also known as zero-order regularization or ridge regression) is examined, both in the noisy cases and in the over-realizable case in which the student has more representational power than the teacher.

Section 7.1 briefly recaps on the on-line learning framework and introduces the particular learning scenario that is employed throughout most of this chapter, sections 7.2 and 7.3 discuss output noise and input noise respectively, while section 7.4 details the effects of regularization in both noisy and over-realizable cases. This chapter has a more phenomenological flavour than those that precede it, as much of the work is exploratory and the addition of noise and regularization complicate the analysis.

7.1 System Dynamics

The framework employed is similar to that studied previously in chapters 5 and 6. The hidden unit positions of the student and teacher are again mapped onto the overlaps $Q_{bc} \equiv \boldsymbol{m}_b \cdot \boldsymbol{m}_c$, $R_{bu} \equiv \boldsymbol{m}_b \cdot \boldsymbol{n}_u$ and $T_{uv} \equiv \boldsymbol{n}_u \cdot \boldsymbol{n}_v$, where T_{uv} is constant and describes the characteristics of the task. The generalization error is calculated as a function of these overlaps and of the

hidden-to-output weights via eqn. (5.3). The time evolutions of the overlaps and hidden-to-output weights are found by calculating the average updates to these quantities using the gradient descent algorithm; in the noiseless, unregularized case, these average updates are identical to those found in eqns. (5.5), (5.6) and (5.7). However, to investigate the various types of noise and regularization, these equations must be modified; these modifications are detailed separately for each case in the appropriate section.

As a control for the effects of noise and regularization, a base case is first established in which the standard update equations of chapter 5 are employed. This realizable case involves a student of 3 SBFs learning an ungraded, uncorrelated teacher of 3 TBFs, with $T_{uu} = 1, T_{uv,u\neq v} = 0$ and $w^0 = 1$; the input dimension $N = 5$ and the learning rate $\eta = 0.5$. Throughout the chapter, the adaptive parameters were initialized in the same way as detailed in section 5.5.

The evolution of generalization error for the control case is depicted in figure 7.1(a), The system passes through the usual four phases of transient ($P = 0$ to 1000), symmetric ($P = 1000$ to 6000), symmetry-breaking ($P = 6000$ to 13000) and convergence ($P = 13000$ and onwards). Figures 7.1(b), 7.1(c) and 7.1(d) show the evolution of the SBF-SBF overlaps ($\boldsymbol{Q}$), the SBF-TBF overlaps ($\boldsymbol{R}$) and the hidden-to-output weights ($\boldsymbol{w}$), respectively.

7.2 Corrupting Examples With Additive Output Noise

To understand the effects of learning under noisy conditions, this first scenario deals with the addition of uncorrelated Gaussian noise to the output

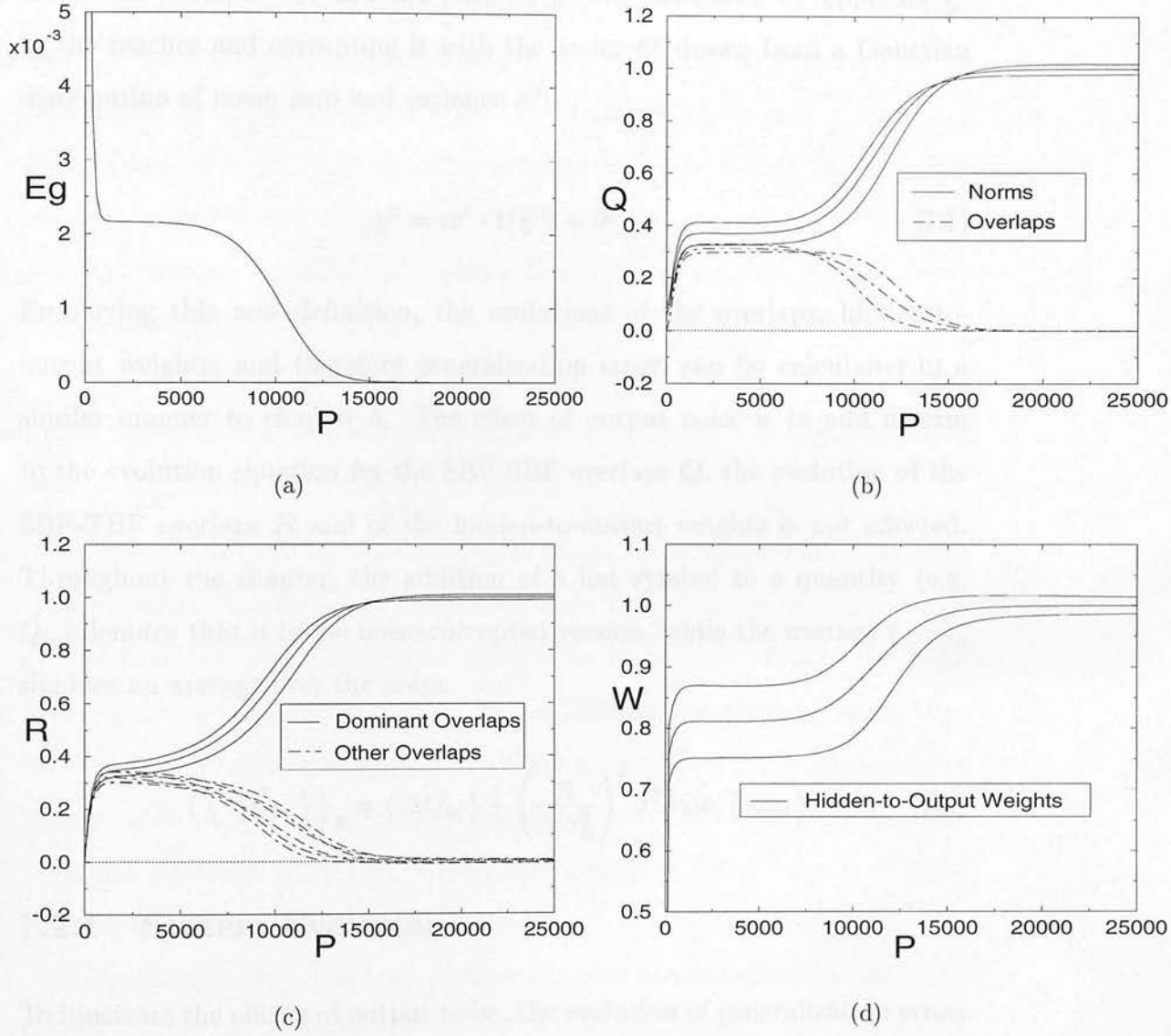


Figure 7.1: The noiseless, unregularized control case. Figure (a) shows the evolution of generalization error; figures (b), (c) and (d) illustrate the evolution of $\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{w}$ respectively. Note the familiar 4 phases: the transient, symmetric, symmetry-breaking and convergence phases.

of the teacher network. The training examples are of the form $(\boldsymbol{\xi}^p, y^p)$ where the input vector $\boldsymbol{\xi}^p$ is, as before, drawn from a Gaussian distribution of zero mean and variance σ_ξ^2, and the outputs y^p are calculated by applying $\boldsymbol{\xi}^p$ to the teacher and corrupting it with the scalar ϑ^p drawn from a Gaussian distribution of mean zero and variance σ^2:

$$y^p = \boldsymbol{w}^0 \cdot \boldsymbol{t}(\boldsymbol{\xi}^p) + \vartheta^p \tag{7.1}$$

Employing this new definition, the evolutions of the overlaps, hidden-to-output weights, and therefore generalization error, can be calculated in a similar manner to chapter 5. The effect of output noise is to add a term to the evolution equation for the SBF-SBF overlaps $\boldsymbol{Q}$; the evolution of the SBF-TBF overlaps $\boldsymbol{R}$ and of the hidden-to-output weights is not affected. Throughout the chapter, the addition of a hat symbol to a quantity (e.g. $\hat{Q}_{bc}$) denotes that it is the noise-corrupted version, while the average $\langle \cdots \rangle_\vartheta$ signifies an average over the noise:

$$\left\langle \left\langle \Delta \hat{Q}_{bc} \right\rangle \right\rangle_\vartheta = \langle \Delta Q_{bc} \rangle + \left(\frac{\eta}{N\sigma_B^2} \right)^2 \sigma^2 w_b w_c \langle s_b s_c \rangle \tag{7.2}$$

7.2.1 System Evolution

To illustrate the effects of output noise, the evolution of generalization error, the overlaps and hidden-to-output weights for the test scenario presented above are plotted in figures 7.2(a) to 7.2(d). Generalization error for various settings of the noise variance σ^2 is depicted in 7.2(a); qualitatively, for low noise levels, the system undergoes the same four-phase process as in the noiseless case. The most salient difference is that the asymptotic generaliza-

tion error is non-zero and increases with noise level. Certainly for low noise levels, the asymptotic error is proportional to the noise variance σ^2.

The effect of the noise on the $\boldsymbol{Q}$ and $\boldsymbol{R}$ overlaps is revealed in figures 7.2(b) and 7.2(c) respectively, for a noise level of $\sigma^2 = 1$. Both the norms and other overlaps of $\boldsymbol{Q}$ are increased beyond their levels in the noiseless case, the norms from 1.0 to 1.05 and the other overlaps from 0.0 to 0.05, showing that the lengths of the hidden unit centre vectors are increased. The dominant overlaps between the SBFs and TBFs, however, are decreased from 1.0 to 0.95, indicating a failure to learn the correct directions of the TBF weight vectors. The hidden-to-output weights are indirectly affected during the earlier stages of learning via the dependence of w on Q, but asymptotically take on the same values as in the noise-free case.

Increasing the noise level has the effect of changing the *qualitative* behaviour of the system: figures 7.3(a) and 7.3(b) show the evolution of the overlaps $\boldsymbol{Q}$ and $\boldsymbol{R}$ respectively for systems affected by high levels of noise ($\sigma^2 = 5$). The symmetric phase is eliminated, and generalization error increases after the transient phase until it reaches a plateau; this increase corresponds to increasing the lengths of the SBF norms in arbitrary directions until they stabilise due to lack of activation as they become further from the area of input space with significant probability mass (figure 7.3(a)). The SBF-TBF overlaps all collapse to a single value, indicating that no specialization has taken place (figure 7.3(b)), while the lack of correlation between SBFs indicates norm growth in arbitrary directions.

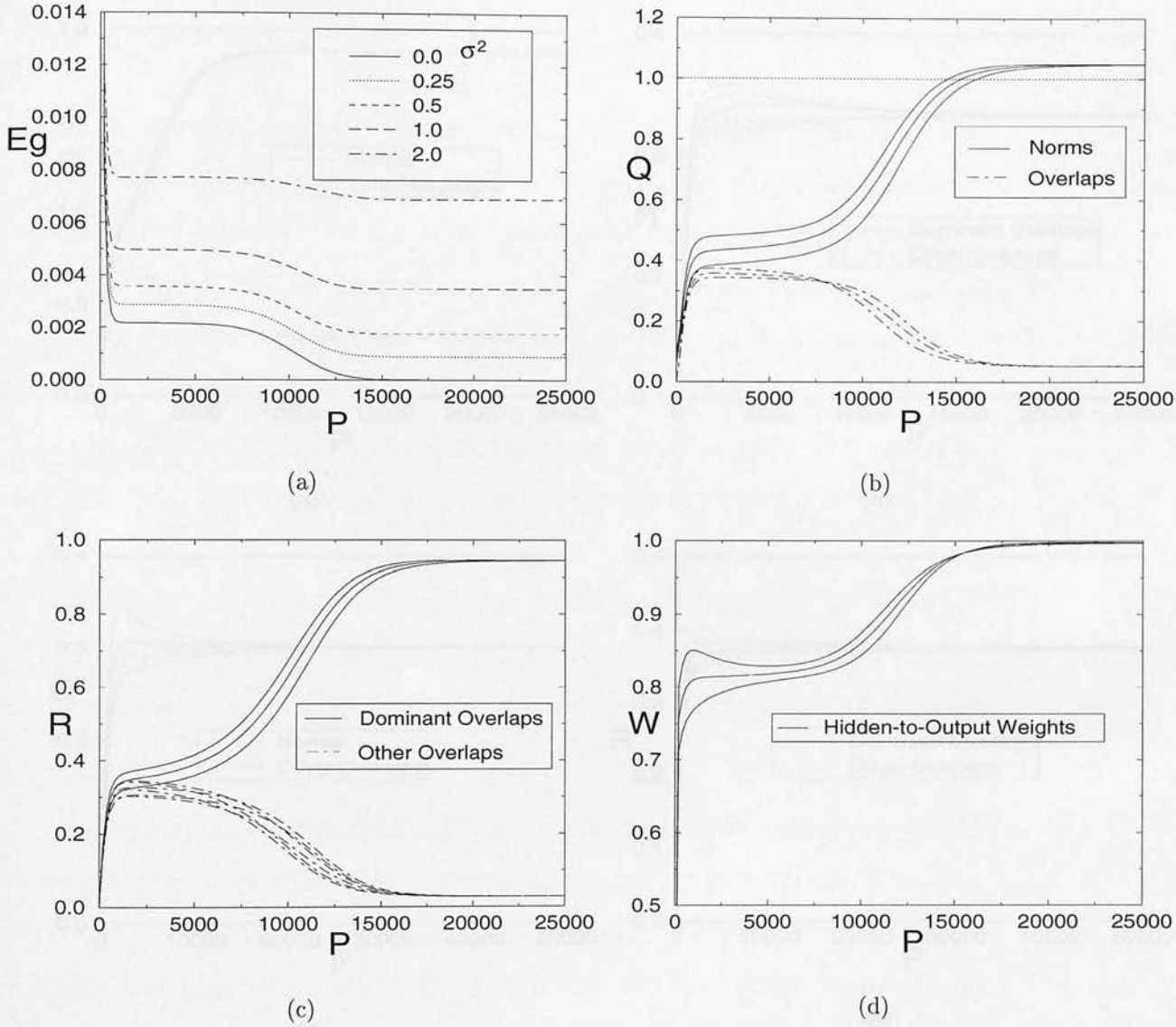


Figure 7.2: On-line learning with output noise. Figure (a) shows the evolution of generalization error for 5 levels of noise. With a fixed learning rate, the asymptotic error is non-zero when output noise is present. Figures (b), (c) and (d) illustrate the evolution of $\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{w}$ respectively for $\sigma^2 = 1.0$. The SBF norms and overlaps are increased; the overlaps between the SBFs and the TBFs they emulate (labelled *dominant overlaps* on figure (c)) are decreased while those between SBFs and the other TBFs are increased. The hidden-to-output weights are not significantly affected.

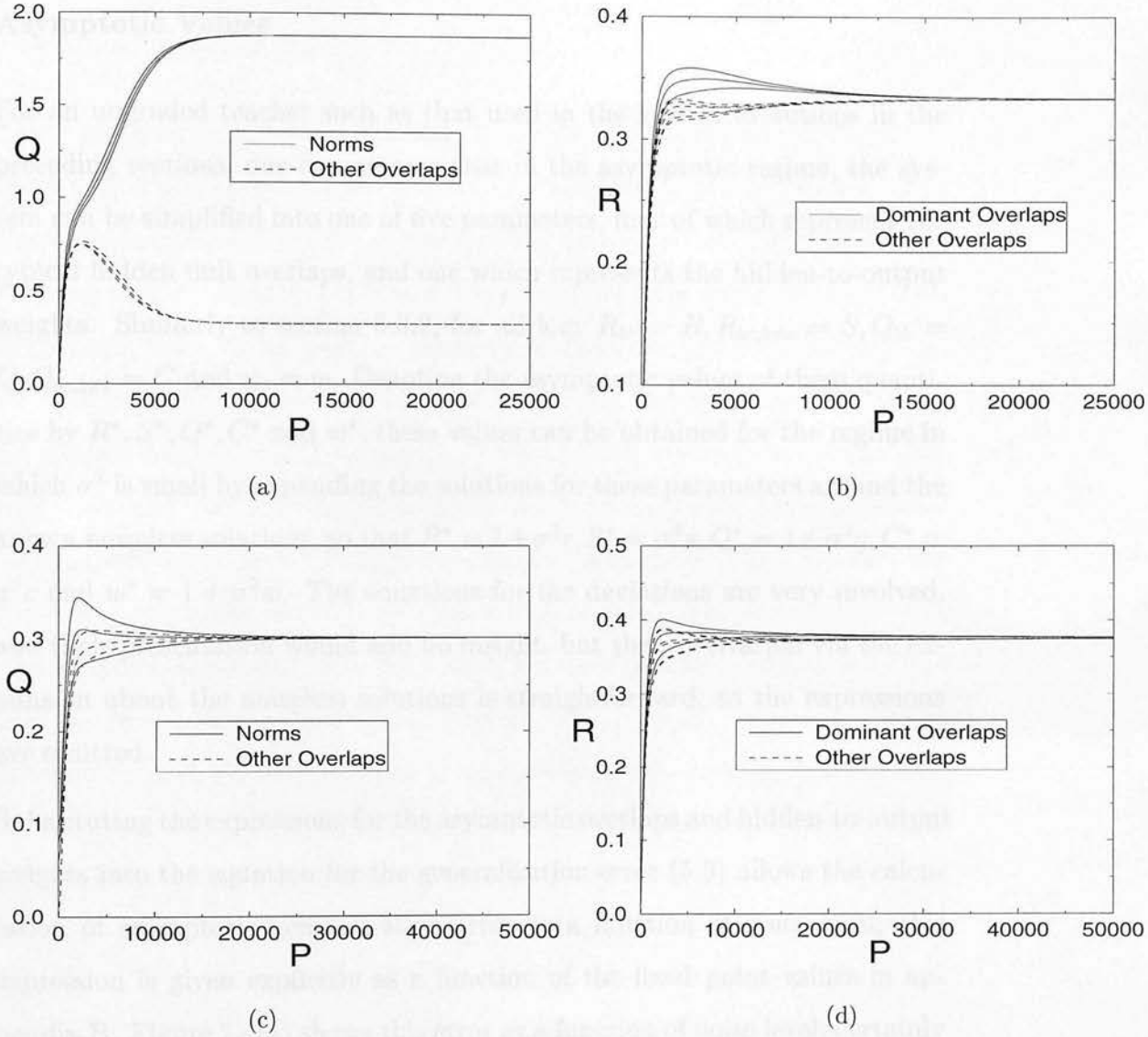


Figure 7.3: On-line learning with high levels of noise. Figures (a) and (b) illustrate the effects of high levels of output noise ($\sigma^2 = 5$) on the SBF-SBF and SBF-TBF overlaps respectively. The SBF-SBF norms increase dramatically, while the SBF-TBF overlaps all collapse to the same value, indicating that the problem is not solved. With high levels of input noise ($\sigma^2 = 0.25$), the SBF-SBF overlaps (figure (c)) are not distinguished and become small, and the SBF-TBF overlaps (figure (d)) also collapse to similar values.

7.2.2 Convergence Phase

Asymptotic Values

For an ungraded teacher such as that used in the system evolutions in the preceding sections, one can assume that in the asymptotic regime, the system can be simplified into one of five parameters, four of which represent the typical hidden unit overlaps, and one which represents the hidden-to-output weights. Similarly to section 6.3.2, for all b, c: $R_{bb} = R, R_{bc,b\neq c} = S, Q_{bb} = Q, Q_{bc,b\neq c} = C$ and $w_b = w$. Denoting the asymptotic values of these quantities by R^*, S^*, Q^*, C^* and w^*, these values can be obtained for the regime in which σ^2 is small by expanding the solutions for these parameters around the known noiseless solutions, so that $R^* = 1+\sigma^2 r, S^* = \sigma^2 s, Q^* = 1+\sigma^2 q, C^* = \sigma^2 c$ and $w^* = 1 + \sigma^2 w$. The equations for the deviations are very involved, and their presentation would add no insight, but their derivation via the expansion about the noiseless solutions is straightforward, so the expressions are omitted.

Substituting the expressions for the asymptotic overlaps and hidden-to-output weights into the equation for the generalization error (5.3) allows the calculation of asymptotic generalization error as a function of noise level; this expression is given explicitly as a function of the fixed point values in appendix B. Figure 7.4(a) shows this error as a function of noise level; certainly for low levels of noise, the error is proportional to the amount of noise (solid line). Also plotted are the values of asymptotic error found via the full system evolution (star symbols), showing an excellent correspondence between the analytic and dynamic results for low noise levels, and a slight mismatch at high noise levels in which the error from the system evolution is lower than that found analytically, due to the fact that the expansion employed to find

the asymptotic values becomes less valid as the noise increases.

Maximal Learning Rates

While training with a finite learning rate in the output noise case, there will always be a non-zero asymptotic error, the magnitude of which depends on the noise level and the value of the learning rate. However, above a certain learning rate, even convergence to a suboptimal solution does not occur. This maximum value of the learning rate can be calculated in a similar manner to that presented in section 6.3.2 by linearizing the dynamical equations for output noise (7.2), (5.6) and (5.7) about their asymptotic values. The eigenvalues of the resulting Jacobian of the system describe the exponential convergence characteristics of the overlaps and hidden-to-output weights; the value of η at which one of these eigenvalues becomes zero defines the value of η at which the asymptotic fixed point becomes unstable, and thus defines the maximal learning rate η_C for convergence to occur. The relationship between η_C and σ^2, derived from this procedure, is plotted in figure 7.4(b); η_C decreases with increasing noise level. The precise relationship between the quantities is difficult to determine because of the complexity of the derived expression, but plotting η_C versus $\log \sigma^2$ produces a linear graph, implying that η_C is a linear function of $\log \sigma^2$. These results have been confirmed by simulations by iterating the full system dynamics from initial conditions very close to the asymptotic fixed point, and varying the learning rate until convergence failed to occur, thus giving η_C.

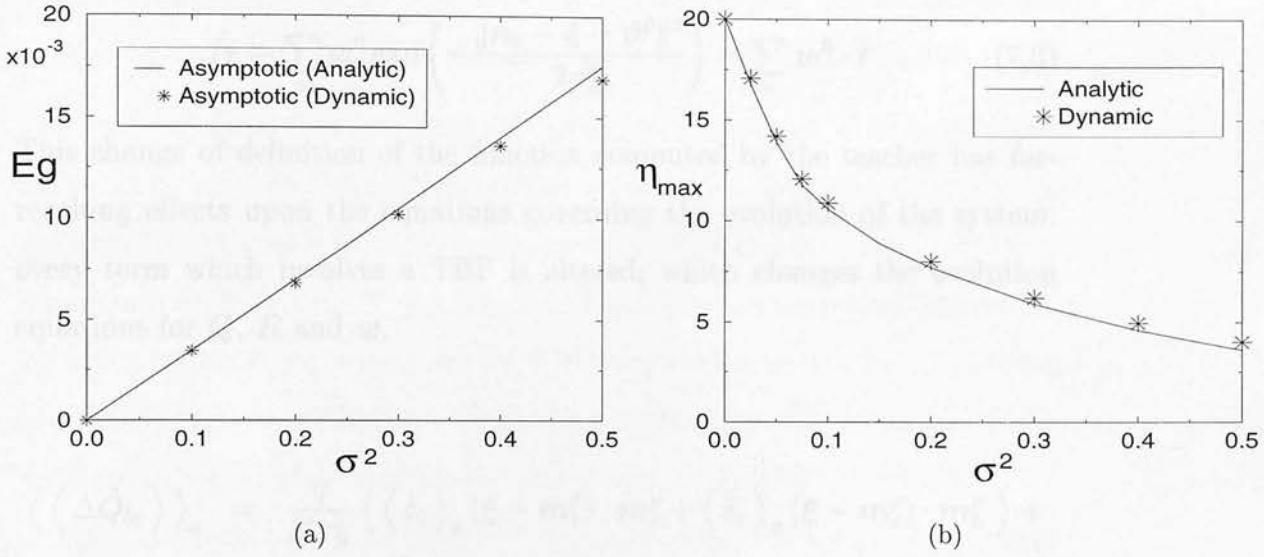


Figure 7.4: Asymptotic error as a function of noise level. Figure (a) shows the asymptotic generalization error as a function of noise level, calculated by expanding the asymptotic overlaps and hidden-to-output weights around the noiseless solutions (solid line). The asymptotic values found by the full system evolution are shown for comparison (star symbols). Figure (b) denotes the maximum learning rate η_C, also as a function of noise level; the solid line represents the analytic solution, while the star symbols show the excellent correspondence with results based on the full system evolution.

7.3 Corrupting Examples With Input Noise

The second type of noise considered is additive input noise, which is implemented here by adding uncorrelated Gaussian noise to each component of the input vector of the teacher. Note that for the RBF, input noise is equivalent to a form of model noise in which the positions of the TBFs are corrupted.

Denoting the noise on example p by the vector $\boldsymbol{\vartheta}^p$, generated by sampling each component from a uncorrelated Gaussian distribution of mean 0, variance σ^2, the function computed by the teacher becomes:

$$\hat{f}_T = \sum_u \boldsymbol{w}^0 \exp\left(-\frac{\|\boldsymbol{n}_u - \boldsymbol{\xi} - \boldsymbol{\vartheta}^p\|^2}{2\sigma_B^2}\right) = \sum_u \boldsymbol{w}^0 \cdot \hat{\boldsymbol{t}} \tag{7.3}$$

This change of definition of the function computed by the teacher has far-reaching effects upon the equations governing the evolution of the system: every term which involves a TBF is altered, which changes the evolution equations for $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{w}$.

$$\begin{aligned}\left\langle\left\langle \Delta\hat{Q}_{bc} \right\rangle\right\rangle_\vartheta &= \frac{\eta}{N\sigma_B^2}\left\langle\left\langle \hat{\delta}_b \right\rangle\right\rangle_\vartheta (\boldsymbol{\xi} - \boldsymbol{m}_b^p)\cdot\boldsymbol{m}_c^p + \left\langle \hat{\delta}_c \right\rangle_\vartheta (\boldsymbol{\xi} - \boldsymbol{m}_c^p)\cdot\boldsymbol{m}_b^p \Big\rangle + \\ &\left(\frac{\eta}{N\sigma_B^2}\right)^2 \left\langle\left\langle \hat{\delta}_b\hat{\delta}_c \right\rangle\right\rangle_\vartheta (\boldsymbol{\xi} - \boldsymbol{m}_b^p)\cdot(\boldsymbol{\xi} - \boldsymbol{m}_c^p) \Big\rangle \end{aligned} \tag{7.4}$$

$$\left\langle\left\langle \Delta\hat{R}_{bu} \right\rangle\right\rangle_\vartheta = \frac{\eta}{N\sigma_B^2}\left\langle\left\langle \hat{\delta}_b \right\rangle\right\rangle_\vartheta (\boldsymbol{\xi} - \boldsymbol{m}_b^p)\cdot\boldsymbol{n}_u \Big\rangle \tag{7.5}$$

$$\langle\langle \Delta\hat{w}_b \rangle\rangle_\vartheta = \frac{\eta}{K}\left\langle \left(\left\langle \hat{f}_T \right\rangle_\vartheta - f_S\right) s_b \right\rangle \tag{7.6}$$

The full averaged expressions for $\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{w}$ are presented in appendix B.

7.3.1 System Evolution

Examples of system evolution under conditions of input noise are presented in figure 7.5. The evolution of generalization error for various levels of noise variance is depicted in figure 7.5(a). For low noise levels, the system passes through the same four stages of training (transitory, symmetric, symmetry-breaking and convergence) as found in the noiseless case. In contrast to the

effects of output noise, the length of the symmetric phase increases as the noise level increases (although it is somewhat less sharply delineated).

The value of generalization error at convergence increases with noise level, as would be expected. The cause of this increase for the input noise case is not identical to that for output noise, however; figures 7.5(b) to 7.5(d) show the system evolution for $\sigma^2 = 1/16$: the SBF norms are *decreased* (figure 7.5) for input noise, whereas they are increased for output noise. The remaining SBF-SBF overlaps are increased, as with output noise. The dominant SBF-TBF overlaps are also decreased (figure 7.5(c)), while the other SBF-TBF overlaps are increased. These effects are caused by the SBFs being less able to distinguish between the TBFs due to the noise. Also in contrast to the output noise case, the hidden-to-output weights are asymptotically affected (figure 7.5(d)), with their convergence value being reduced. The evolution of the system is far more sensitive to input noise than to output noise; corrupting the input vector affects *all* the hidden units of the teacher, altering the non-linear response of each unit in a correlated manner, while output noise affects only the weighted, summed output of the teacher network.

Adding high levels of input noise ($\sigma^2 = 0.25$) causes a change in the behaviour of the system. The evolution of generalization error for this case is presented in figure 7.5(a), dot-dash line; the system does not escape the symmetric phase. As illustrated in figures 7.3(c) and 7.3(d), the SBF-SBF overlaps ($\boldsymbol{Q}$) collapse to a single value (there is no specialization) as do the SBF-TBF overlaps ($\boldsymbol{R}$). The system reaches a fixed point and the problem is not solved.

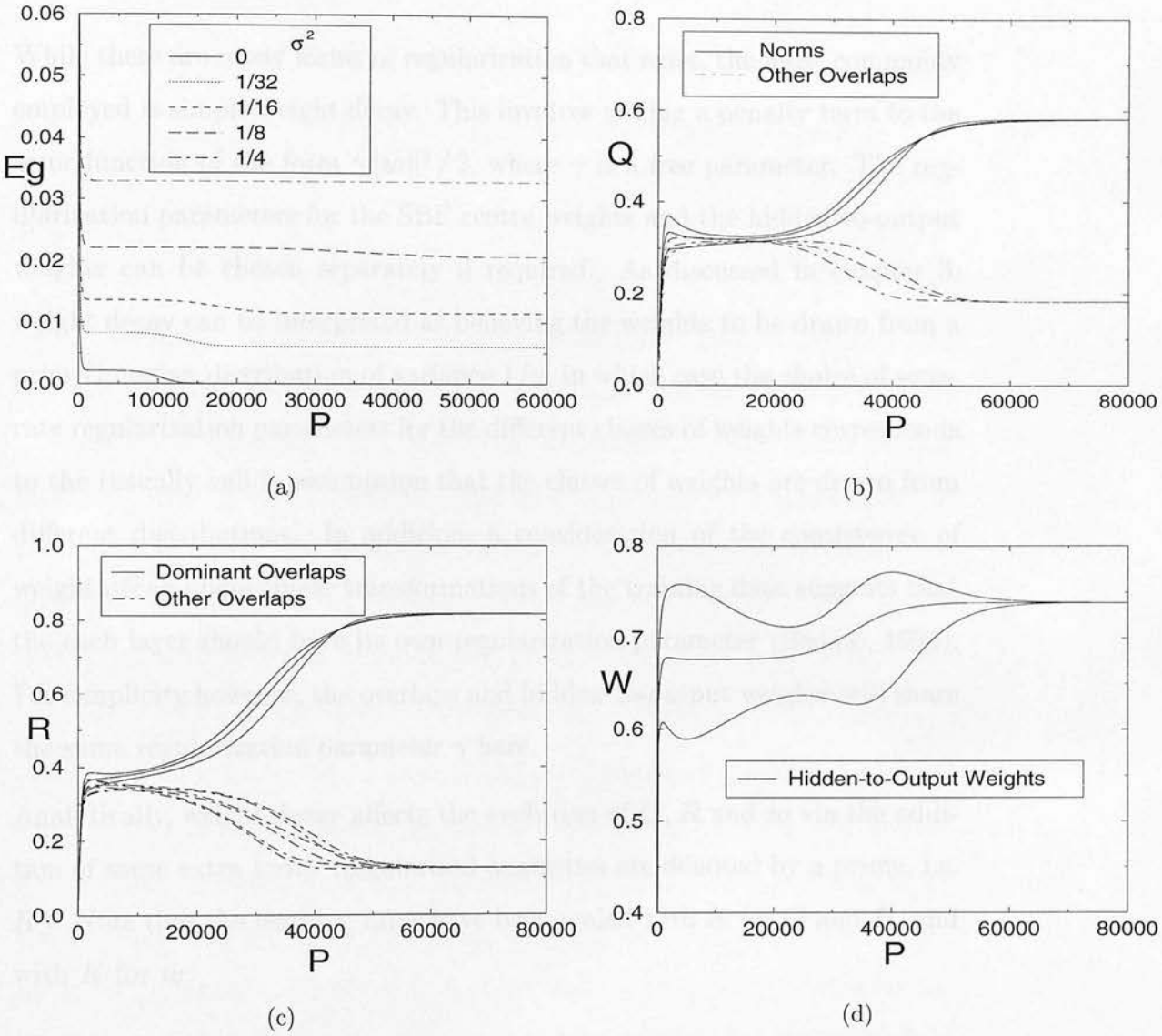


Figure 7.5: On-line learning with input noise. Figure (a) shows the generalization error for 5 levels of noise (with $\sigma^2 = 1/4$, the problem is not solved). As with output noise, the asymptotic error is non-zero with a fixed learning rate. Figures (b), (c) and (d) illustrate the evolution of $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{w}$ respectively for $\sigma^2 = 1/8$. The SBF norms are decreased while the other SBF-SBF overlaps are increased; the dominant SBF-TBF overlaps are decreased, and the other SBF-TBF overlaps are increased. Unlike the output noise case, the hidden-to-output weights are significantly affected and reduced.

7.4 Regularization

While there are many forms of regularization that exist, the most commonly employed is simple weight decay. This involves adding a penalty term to the error function of the form $\gamma \|\boldsymbol{w}\|^2 / 2$, where γ is a free parameter. The regularization parameters for the SBF centre weights and the hidden-to-output weights can be chosen separately if required. As discussed in chapter 3, weight decay can be interpreted as believing the weights to be drawn from a prior Gaussian distribution of variance $1/\gamma$, in which case the choice of separate regularization parameters for the different classes of weights corresponds to the (usually valid) assumption that the classes of weights are drawn from different distributions. In addition, a consideration of the consistency of weight decay under linear transformations of the training data suggests that the each layer should have its own regularization parameter (Bishop, 1995). For simplicity however, the overlaps and hidden-to-output weights will share the same regularization parameter γ here.

Analytically, weight decay affects the evolution of $\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{w}$ via the addition of some extra terms (regularized quantities are denoted by a prime, i.e. R'). Note that the learning rates have been scaled with N for $\boldsymbol{Q}$ and $\boldsymbol{R}$, and with K for $\boldsymbol{w}$:

$$\left\langle \Delta w_b' \right\rangle = \left\langle \Delta w_b \right\rangle - \frac{\eta}{K} \gamma w_b \tag{7.7}$$

$$\left\langle \Delta R_{bu}' \right\rangle = \left\langle \Delta R_{bu} \right\rangle - \frac{\eta}{N} \gamma R_{bu} \tag{7.8}$$

$$\left\langle \Delta Q'_{bc} \right\rangle = \langle \Delta Q_{bc} \rangle - 2\frac{\eta}{N}\gamma Q_{bc} + \left(\frac{\eta}{N}\right)^2 \left\{\gamma^2 Q_{bc}^2 + \gamma(w_b s_c + w_c s_b)(f_S - f_T)\right\} \tag{7.9}$$

7.4.1 System Evolution

There are many combinations of noise, regularization and realizability that can be examined within the framework. The base case employed previously, in which 3 SBFs learn 3 TBFs, is used as the foundation from which regularization of realizable systems corrupted by output noise and input noise are examined. Of particular interest is the regularization of over-realizable cases in which the student is representationally more powerful than the teacher: this is analysed for both noiseless and noisy scenarios by employing a new base case (detailed below) of 5 SBFs learning 2 TBFs.

Generally, with low levels of regularization, the system passes through the usual four phases: transient, symmetric, symmetry-breaking and convergence. Regularization prolongs the symmetric phase, with a resultant decrease in learning speed, by decreasing the instability of the symmetric fixed point. With higher levels of regularization, this can lead to the system becoming trapped in the symmetric phase in learning scenarios that are successfully solved without regularization.

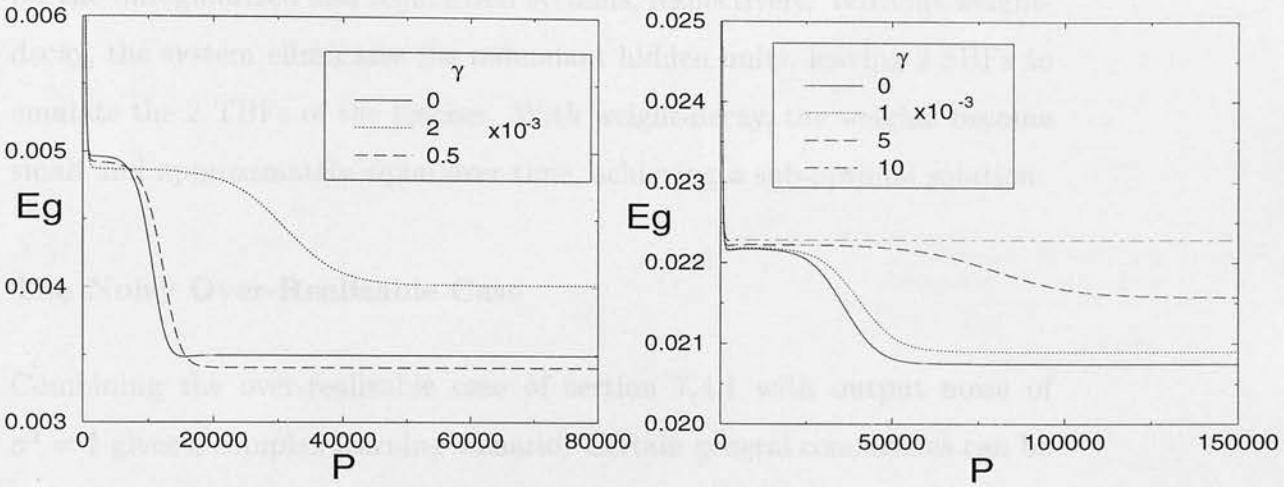
Examples Corrupted With Output Noise

It has been claimed for the SCM that regularization via weight decay does not improve system performance in noisy cases, and been hypothesized that

this is a generic feature of on-line learning due to the absence of a additive, stationary error surface defined over a fixed, finite training set. However, certainly for the RBF, it is possible with careful selection of the regularization parameter to improve the asymptotic generalization error to some extent. A noisy training scenario was constructed by taking the base case and adding additive output noise of $\sigma^2 = 1$. Various levels of regularization were applied, and the effect of this on generalization error is shown in figure 7.6(a). With $\gamma = 0.5 \times 10^{-3}$ (dashed line), which was found to be optimal, asymptotic generalization error falls from 3.5×10^{-4} to 3.4×10^{-4}, a small improvement of 3%. Note that regularization is still detrimental during most of the non-asymptotic phase. Over-regularization is extremely detrimental to learning (figure 7.6(a), dotted line); with $\gamma = 2 \times 10^{-3}$, the symmetric phase is approximately trebled in length, and the asymptotic generalization error is increased from 3.5×10^{-4} to 4.0×10^{-4}.

Examples Corrupted With Input Noise

By taking the base case and corrupting the teacher inputs with uncorrelated Gaussian noise on each component of variance $\sigma_\xi^2 = 0.125$, the effects of regularization of systems corrupted by input noise was examined. Generalization error for various levels of regularization ($\gamma = 0, 1, 5$ and 10×10^{-3}) is shown in figure 7.6(b): in each case, regularization increased the length of the symmetric phase, thus increasing the time required for learning, and also increased the asymptotic error. With $\gamma = 10 \times 10^{-3}$ (figure 7.6(b), dot-dash curve), the system fails to solve the task and remains trapped in the symmetric phase, as the symmetric fixed point has become stable due to the regularization.



(a) (b)

Figure 7.6: Regularization in noisy on-line learning. Figure (a) shows the effects of different levels of regularization when the teacher is corrupted with output noise of variance $\sigma^2 = 1$. Careful choice of the regularization parameters allows a slight reduction in asymptotic generalization error, while over-regularization leads to a poor solution. Figure (b) illustrates the effects of regularizing the input noise case with input noise $\sigma^2 = 1/8$; regularization is always counterproductive, and can lead to a failure to solve problems which are solved without regularization.

The Over-realizable Case

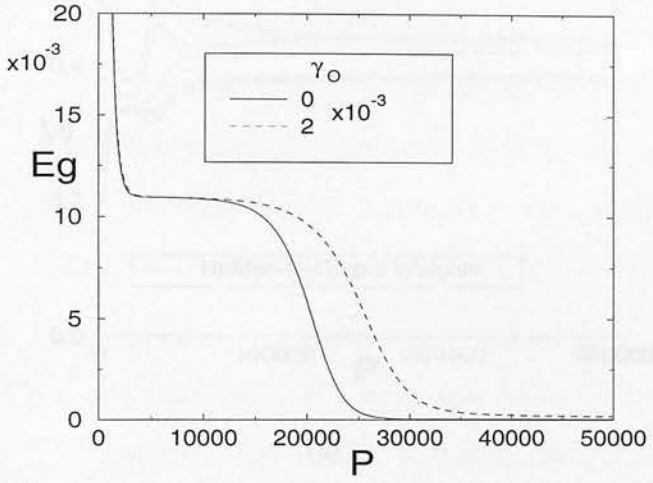
To investigate the over-realizable case, a new base case was established in which a student of 5 basis functions learns a teacher of 2 basis functions with $T_{uu} = 1, T_{uv,u\neq v} = 0$ and $w^0 = 1$ with $N = 5$ and $\eta = 0.5$. The initial conditions were as detailed in section 5.5.

Typical results are presented in figure 7.7. Regularizing the system again leads to an increase in symmetric phase length and asymptotic error. The reason for this is that, *a priori*, weight decay encourages all weights to be *small and equal*; it does not promote the elimination of excess hidden units. Figures 7.7(b) and 7.7(c) compare the hidden-to-output weight evolutions

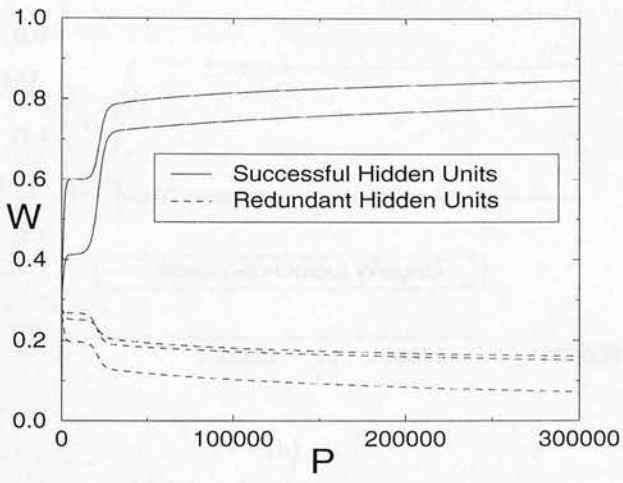
for the unregularized and regularized systems, respectively. Without weight-decay, the system eliminates the redundant hidden units, leaving 2 SBFs to emulate the 2 TBFs of the teacher. With weight-decay, the weights become small and approximately equal over time, achieving a sub-optimal solution.

The Noisy Over-Realizable Case

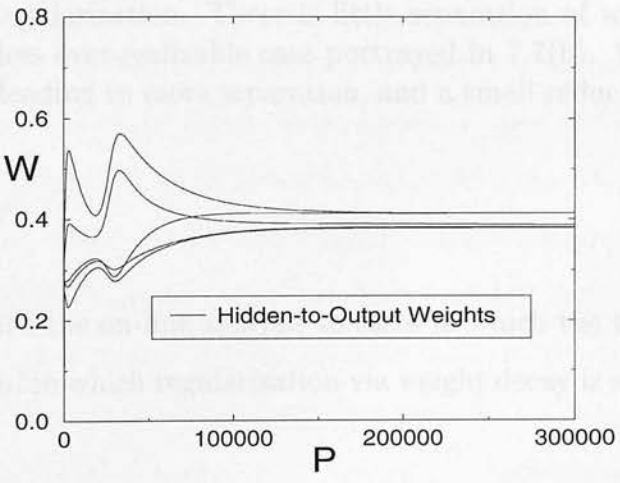
Combining the over-realizable case of section 7.4.1 with output noise of $\sigma^2 = 1$ gives a complex learning scenario. Certain general conclusions can be drawn, however. If unregularized, the student, being more representationally powerful than the teacher, learns details of the noise and does not eliminate the redundant hidden units. This can be seen by comparing the hidden-to-output weights of the noisy case (figure 7.8(a)), in which the redundant weights do not approach 0, with those of the equivalent noiseless case (figure 7.7(b)) in which the 3 redundant units are eventually eliminated. Applying regularization, with $\gamma = 5 \times 10^{-3}$, improves the separation between the necessary hidden-to-output weights of the student and those that are redundant (figure 7.8(b)), and can lead to a small improvement in generalization ability. As discussed in (Bishop, 1995), with a quadratic error function it is easy to show that weight decay has the effect of suppressing the weight vector of the solution in the directions of weight space in which the error function changes only slowly; it has little effect in directions which are important to the solution (i.e. where the error changes rapidly). Thus components of the solution that are primarily due to noise are suppressed, while those that are based on the underlying structure are retained.



(a)

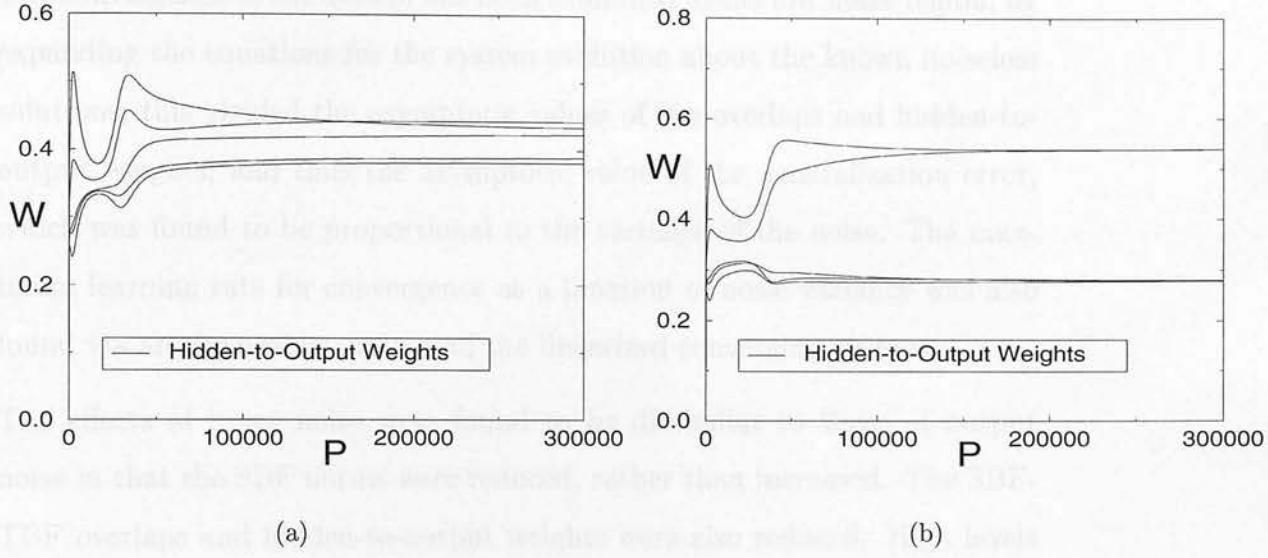


(b)



(c)

Figure 7.7: Regularization in the noiseless over-realizable case. Figure (a) shows the increase in generalization error and time required to solve the problem when regularization is used. The reason for this is illustrated in figures (b) and (c): with no regularization (b), the redundant hidden units are slowly eliminated. Weight decay encourages all the weights to be small and equal (c), leading to a sub-optimal solution.



(a) (b)

Figure 7.8: Regularization in the noisy over-realizable case. Figure (a) shows the hidden-to-output weights for the over-realizable scenario with output noise $\sigma^2 = 1$, without regularization. There is little separation of weights, in contrast to the noiseless over-realizable case portrayed in 7.7(b). Weight decay is applied in (b), leading to more separation, and a small reduction in generalization error.

7.5 Summary

This chapter has extended the on-line analysis to cases in which the teacher is corrupted by noise, and in which regularization via weight decay is applied to the student.

Both output noise and input noise have been examined. Output noise has the effect of increasing the magnitude of the SBF norms and overlaps, while decreasing the overlaps between the SBFs and the TBFs, thus reducing the specialization of the SBFs. The hidden-to-output weights were not significantly affected by the noise. High levels of noise caused a quantitative change in the system in that the symmetric phase was eliminated, yet the overlaps between SBFs and TBFs ($\boldsymbol{R}$) became similar (so no specialization occurred).

The convergence of the system has been examined in the low noise region, by expanding the equations for the system evolution about the known noiseless solutions; this yielded the asymptotic values of the overlaps and hidden-to-output weights, and thus the asymptotic value of the generalization error, which was found to be proportional to the variance of the noise. The maximum learning rate for convergence as a function of noise variance was also found via an eigenvalue analysis of the linearized converging system.

The effects of input noise were found to be dissimilar to those of output noise in that the SBF norms were reduced, rather than increased. The SBF-TBF overlaps and hidden-to-output weights were also reduced. High levels of input noise led to the system becoming trapped in a symmetric phase in which there was no specialization of SBFs whatsoever.

Regularization via weight decay was examined and applied to noiseless, noisy and over-realizable cases. In the noiseless case, regularization always degraded generalization performance, and made the symmetric fixed point more attractive so that the RBF failed to solve some tasks that are solved without regularization as it remained trapped in the symmetric phase. In the case of output noise, a small improvement in generalization performance could be obtained via careful selection of the regularization parameters, although over-regularization was extremely detrimental to performance. However, with input noise, no improvement could be found, and again the symmetric phase became more attractive, to the point where scenarios which had been successfully solved without regularization becoming insoluble. Regularization of the over-realizable case was studied, but again no improvement could be obtained in the noiseless case; performance was degraded and the excess SBFs were no longer eliminated. Regularizing the complex (output) noise-corrupted over-realizable case led to minor improvements via increasing the

specialization of the dominant SBFs on the target TBFs while decreasing the hidden-to-output weights of the spurious SBFs.

Chapter 8

Conclusion

> *Those Magicians, who have attempted to make the sword the sole or even the principal weapon, have only destroyed themselves, not by the destruction of combination, but by the destruction of division.*
>
> Aleister Crowley, Magick

The average-case properties of learning and generalization in the RBF network have been elucidated in a well-founded manner, and various methods of measuring generalization ability have been considered and related.

The first unit of the thesis concerns the relations between measures of generalization. It proved straightforward to relate the prediction probability, and therefore prediction error, to the evidence measure: the prediction probability on estimating a new test point given a dataset is simply the ratio of the evidence given the union of this dataset with the test point to the evidence given the original dataset. Thus prediction error for the test point is the change in log evidence caused by adding the test point to the dataset.

It is considerably more difficult to relate generalization error to the predic-

tion probability, as calculating generalization error relies on having some functional knowledge of the data-generating mechanism. However, if the learning model is reasonably well-trained, such that the posterior distribution over the parameters of the model is concentrated in regions which give a small error, an ordering relation is obtained which demonstrates that the prediction probability given dataset D_1 being greater than that for D_2 is equivalent to the generalization error on D_1 being less than that on D_2. If there is knowledge of the form of the data-generating mechanism, it is possible in some cases to make much more precise statements concerning the relationship between prediction probability and generalization error. This is performed for the RBF in chapter 3, although the resulting relationship is not as intuitive as those above.

The second unit of the thesis deals with the analysis of RBFs in which the parameters of the hidden units are fixed prior to training. By assuming a functional form for the teacher mechanism, which included Gaussian output noise, it proved possible to calculate the typical generalization error given the number of training points, within a stochastic training paradigm. By averaging over possible datasets, this average error was calculated independently of the exact data, and it was found that, with no weight decay, the error decreases as $1/P$. It was also possible to find the optimal settings of the hyperparameters which control the learning process. The effects of setting these hyperparameters to suboptimal values was also examined: under-regularizing leads to very poor initial performance as the student models the noise rather than the underlying structure, but this is overcome rapidly with the addition of more training data. Over-regularizing is less detrimental initially, but requires a great deal of data to recover from.

By extending the framework to encompass the case in which the teacher

mechanism does not match that of the learning model, it was possible to examine the case in which the student had greater representational power than the teacher, and the converse. With the student more powerful, there is a tendency to under-regularize due to over-estimating the complexity of the teacher. Given sufficient data, this problem can be overcome, but far more data is required than if the student matches the teacher. When the teacher is more powerful, there is a component of the error that cannot be overcome through training, and there is also a tendency to over-regularize as the complexity of the teacher is under-estimated. The requirement that the teacher is known was relaxed by incorporating uncertainty into the knowledge of the teacher model; the error initially increases with the uncertainty, but then decreases with extreme uncertainty as, probabilistically, the teacher model has no structure in the region of space modelled by the student. Finally, simulations confirm the validity of the analytic results.

The third thesis unit concerns the analysis of RBFs in which the positions of the basis function centres are adaptive, as well as the hidden-to-output weights. By employing the on-line learning paradigm, it proved possible to find equations for not only the average evolution of generalization error, but also the dynamics of the hidden units. These average equations were solved iteratively to elucidate the learning process. There are four typical stages of training: initially, there is a transitory phase as the parameters of the network adapt from their initial conditions; this is followed by the symmetric phase, characterised by a lack of differentiation between the basis functions. Given asymmetries in the task or initial conditions, the units specialize in the symmetry-breaking phase. Finally there is a exponential convergence phase as the network reaches its asymptotic state. There are three learning rate regimes: too small a learning rate leads to slow learning, an overly large

learning rate causes the system to fail to converge to the correct target, while between these extremes lies a region in which the problem is solved rapidly.

The symmetric and symmetry-breaking phases were examined via an eigenvalue analysis of the symmetric fixed point, with the result that a fundamental difference between the soft committee machine and the RBF in the realizable case was uncovered. Increasing the number of basis functions *decreases* the time taken to escape the symmetric phase in the RBF, while it has the opposite effect for the SCM. This difference is hypothesized to be due to the localized nature of the basis functions for the RBF model; unlike the SCM case, in which the basis functions are non-local, small perturbations about the symmetric fixed point result in massive changes in error, and increasing the number of basis functions emphasizes this effect by increasing the ruggedness of the error surface. The convergence phase was also analysed by examining the properties of the asymptotic fixed point, allowing the calculation of the maximal and optimal learning rates. In the case where the hidden-to-output weights are clamped to a fixed value, both maximal and optimal rates scale as $1/K$. However, when the analysis is extended to cover adaptive hidden-to-output weights, the maximal learning rate remains unchanged while the optimal rate rapidly approaches the maximal rate as network size increases. This implies that the maximal learning rate is purely a function of the hidden layer, also noted by Riegler and Biehl (1995) for the MLP, and that it is hard to optimise learning rates for fully-adaptive networks, especially those with many hidden units.

Quantification of the variances of the average on-line update equations demonstrates the validity of the approach for RBFs, while simulations confirm the accuracy of the results for both the mean equations and the variances.

The analysis of on-line learning in RBFs was extended to the cases in which the data-generating mechanism is corrupted by either output or input noise. Output noise has the effect of increasing the lengths of the SBF norms in arbitrary directions while decreasing the degree of specialization of the SBFs on their targets, with the result that, when employing a finite learning rate, asymptotic generalization error is non-zero. A high level of output noise eliminates the symmetric phase while maintaining the lack of specialization of the hidden units, and can lead to the SBF norms becoming so large that the units are no longer activated during training, effectively becoming redundant. An examination of the convergence properties of the system under conditions of output noise revealed that, at least for low noise levels with a fixed learning rate, the asymptotic generalization error is proportional to the variance of the noise. The maximum learning rate was found via an eigenvalue analysis of the asymptotic fixed point, and was shown to be a monotonically decreasing function of the noise variance.

Corrupting the teacher with input noise has the opposite effect to that of output noise in that the SBF norms are suppressed. The degree of specialization of the SBFs is again reduced, however, as are the magnitudes of the hidden-to-output weights. Again this leads to non-zero asymptotic generalization error with finite learning rate. The addition of high levels of input noise causes trapping of the system in the symmetric phase with no specialization whatsoever of the hidden units.

The use of regularization via weight decay was investigated for on-line learning. In the exactly realizable case without noise, regularization significantly degrades generalization performance and learning speed; in some cases, tasks solved without regularization remain trapped in the symmetric phase with regularization. In the presence of output noise, weight decay can give a small

improvement in asymptotic generalization error with careful selection of the regularization parameter. However, no such improvement could be discerned in the case of input noise. Considering an over-realizable task in which the student has more representational power than the teacher, without noise, performance is again hampered by regularization. With the addition of output noise, again a small improvement in generalization performance can be obtained as regularization increases the specialization of the student hidden units on their targets.

Several avenues suggest themselves for further exploration. The focus of the thesis has been on regression; classification could also be considered. Within the fixed hidden-unit paradigm, it may be possible to analyze the various methods for setting the hidden unit parameters to determine the implications for generalization performance. Empirical comparisons between methods can be found, but little analysis is so far available.

Considering the on-line framework, scope exists for greater analysis of the effects of noise. While the convergence properties in the presence of output noise were calculated, this has not yet been performed for input noise. The noisy symmetric phase can also be examined in a similar manner to the analyses in chapters 5 and 6. The study of the effects of regularization could also be expanded: weight decay is a very simple form of regularization, and more complicated schemes such as soft weight-sharing, which can achieve significantly better results, could also be analysed.

The symmetric phase is a significant yet undesirable portion of the learning dynamics in realistic tasks. Since the phase is caused by undifferentiated overlaps between student and teacher, these could be artificially broken through the introduction of extra terms in the error function penalizing over-

lap similarity. The effect of these terms should be annealed over time, to allow convergence to the correct targets.

Finally, the on-line framework considers datasets to be generated by applying the teacher function to a sample from the input distribution, providing a potentially infinite pool of datapoints; in practice, datasets are finite and on-line training often proceeds by cycling through the datapoints a number of times - the analysis of RBFs could be modified to deal with this case.

Appendix A

Stochastic Learning Quantities

$$g_0 = \left(\frac{\sigma_B^2}{2\sigma_\xi^2 + \sigma_B^2} \right)^{N/2} \tag{A.1}$$

$$g_1 = \frac{\sigma_\xi^2}{2\sigma_\xi^2 + \sigma_B^2} \tag{A.2}$$

$$G_{bc} = g_0 \exp\left[\frac{-\|\boldsymbol{m}_b\|^2 - \|\boldsymbol{m}_c\|^2 + g_1 \|\boldsymbol{m}_b + \boldsymbol{m}_c\|^2}{2\sigma_B^2} \right] \tag{A.3}$$

$$j_0 = \left(\frac{\sigma_B^2 \sigma_{Bt}^2}{\sigma_B^2 \sigma_{Bt}^2 + 2\sigma_\xi^2(\sigma_B^2 + \sigma_{Bt}^2)} \right)^{N/2} \tag{A.4}$$

$$j_1 = \left(\frac{2\sigma_\xi^2 \sigma_B^2 \sigma_{Bt}^2}{\sigma_B^2 \sigma_{Bt}^2 + 4\sigma_\xi^2(\sigma_B^2 + \sigma_{Bt}^2)} \right) \tag{A.5}$$

$$j_2 = \frac{2\sigma_{Bt}^2 \sigma_c^2 - 2j_1 \sigma_c^2 + \sigma_{Bt}^4}{2\sigma_{Bt}^4 \sigma_c^2} \tag{A.6}$$

$$j_3 = (2j_2\sigma_c^2)^{-N/2} \tag{A.7}$$

$$J_{bcuv} = j_0 \exp\left[-\frac{\|\boldsymbol{m}_b\|^2 + \|\boldsymbol{m}_c\|^2}{2\sigma_B^2} - \frac{\|\boldsymbol{n}\|^2 + \|\boldsymbol{n}_v\|^2}{2\sigma_{Bt}^2} + \frac{j_1}{4}\left\|\frac{\boldsymbol{m}_b + \boldsymbol{m}_c}{\sigma_B^2} + \frac{\boldsymbol{n}_u + \boldsymbol{n}_v}{\sigma_{Bt}^2}\right\|^2\right] \tag{A.8}$$

$$J'_{bcuu} = j_3 \exp\left[\frac{-\|\boldsymbol{m}_b\|^2 - \|\boldsymbol{m}_c\|^2}{2\sigma_B^2} + \frac{j_1\|\boldsymbol{m}_b + \boldsymbol{m}_c\|^2}{4\sigma_B^4} - \frac{\|\boldsymbol{m}_u\|^2}{2\sigma_c^2} + \frac{1}{4j_2}\left\|\frac{j_1(\boldsymbol{m}_b + \boldsymbol{m}_c)}{2\sigma_B^2\sigma_{Bt}^2} + \frac{\boldsymbol{m}_v}{\sigma_c^2}\right\|^2\right] \tag{A.9}$$

$$k_0 = \left(\frac{\sigma_{Bt}^2}{2\sigma_\xi^2 + \sigma_{Bt}^2}\right)^{N/2} \tag{A.10}$$

$$k_1 = \frac{\sigma_\xi^2}{2\sigma_\xi^2 + \sigma_{Bt}^2} \tag{A.11}$$

$$k_2 = \frac{\sigma_{Bt}^2 + 2\sigma_c^2(1 - k_1)}{2\sigma_c^2\sigma_{Bt}^2} \tag{A.12}$$

$$k_3 = (2\sigma_c^2 k_2)^{-N/2} \tag{A.13}$$

$$K_{uv} = k_0 \exp\left[\frac{-\|\boldsymbol{n}_u\|^2 - \|\boldsymbol{n}_v\|^2 + k_1\|\boldsymbol{n}_u + \boldsymbol{n}_v\|^2}{2\sigma_{Bt}^2}\right] \tag{A.14}$$

$$K'_{uu} = k_3 \exp\left[\frac{\|\boldsymbol{m}_u\|^2(2\sigma_c^2 k_2 - 1)}{4\sigma_c^4 k_2}\right] \tag{A.15}$$

$$l_0 = \left(\frac{\sigma_B^2\sigma_{Bt}^2}{\sigma_B^2\sigma_{Bt}^2 + \sigma_\xi^2(\sigma_B^2 + \sigma_{Bt}^2)}\right)^{N/2} \tag{A.16}$$

$$l_1 = \frac{2\sigma_\xi^2\sigma_B^2\sigma_{Bt}^2}{\sigma_B^2\sigma_{Bt}^2 + \sigma_\xi^2(\sigma_B^2 + \sigma_{Bt}^2)} \tag{A.17}$$

$$l_2 = \frac{2\sigma_{Bt}^2\sigma_c^2 - l_1\sigma_c^2 + \sigma_{Bt}^4}{\sigma_{Bt}^4\sigma_c^2} \tag{A.18}$$

$$l_3 = (2\sigma_c^2 l_2)^{-N/2} \tag{A.19}$$

$$L_{bv} = l_0 \exp\left(-\frac{\|\boldsymbol{m}_b\|^2}{2\sigma_B^2} - \frac{\|\boldsymbol{n}_v\|^2}{2\sigma_{Bt}^2} + \frac{l_1}{4}\left\|\frac{\boldsymbol{m}_b}{\sigma_B^2} + \frac{\boldsymbol{n}_v}{\sigma_{Bt}^2}\right\|^2\right) \tag{A.20}$$

$$\begin{aligned} L'_{bcu} \;=\; & l_3 \exp\left[\left(\frac{l_1}{4\sigma_B^4} - \frac{1}{2\sigma_B^2}\right)(\|\boldsymbol{m}_b\|^2 - \|\boldsymbol{m}_c\|^2) - \frac{\|\boldsymbol{m}_u\|^2}{2\sigma_c^2} + \right. \\ & \left. \frac{1}{4l_2}\left\|l_1\frac{\boldsymbol{m}_b + \boldsymbol{m}_c}{2\sigma_B^2\sigma_{Bt}^2} + \frac{\boldsymbol{m}_u}{\sigma_c^2}\right\|^2\right] \end{aligned} \tag{A.21}$$

Appendix B

On-line Learning Quantities

Generalization Error:

$$E_G = \frac{1}{2}\left\{\sum_{bc} w_b w_c I_2(b,c) + \sum_{uv} w_u^0 w_v^0 I_2(u,v) - 2\sum_{bu} w_b w_u^0 I_2(b,u)\right\} \quad \text{(B.1)}$$

ΔQ, ΔR and Δw:

$$\begin{aligned}\langle \Delta Q_{bc}\rangle &= \frac{\eta}{N\sigma_B^2}\left\{w_b\left[\overline{J}_2(b;c) - Q_{bc}\overline{I}_2(b)\right] + w_c\left[\overline{J}_2(c;b) - Q_{bc}\overline{I}_2(c)\right]\right\} \quad \text{(B.2)}\\ &+ \left(\frac{\eta}{N\sigma_B^2}\right)^2 w_b w_c\left\{\overline{K}_4(b,c) + Q_{bc}\overline{I}_4(b,c) - \overline{J}_4(b,c;b) - \overline{J}_4(b,c;c)\right\}\end{aligned}$$

$$\langle \Delta R_{bu}\rangle = \frac{\eta}{N\sigma_B^2} w_b\left\{\overline{J}_2(b;u) - R_{bu}\overline{I}_2(b)\right\} \quad \text{(B.3)}$$

$$\langle \Delta w_b \rangle = \frac{\eta}{K} \overline{I}_2(b) \tag{B.4}$$

$\overline{I}, \overline{J}$ and $\overline{K}$:

$$\overline{I}_2(b) = \sum_u w_u^0 I_2(b,u) - \sum_d w_d I_2(b,d) \tag{B.5}$$

$$\overline{J}_2(b;c) = \sum_u w_u^0 J_2(b,u;c) - \sum_d w_d J_2(b,d;c) \tag{B.6}$$

$$\begin{aligned} \overline{I}_4(b,c) = & \sum_{de} w_d w_e I_4(b,c,d,e) + \sum_{uv} w_u^0 w_v^0 I_4(b,c,u,v) - \\ & 2 \sum_{du} w_d w_u^0 I_4(b,c,d,u) \end{aligned} \tag{B.7}$$

$$\begin{aligned} \overline{J}_4(b,c;f) = & \sum_{de} w_d w_e J_4(b,c,d,e;f) + \sum_{uv} w_u^0 w_v^0 J_4(b,c,u,v;f) - \\ & 2 \sum_{du} w_d w_u^0 J_4(b,c,d,u;f) \end{aligned} \tag{B.8}$$

$$\begin{aligned} \overline{K}_4(b,c) = & \sum_{de} w_d w_e K_4(b,c,d,e) + \sum_{uv} w_u^0 w_v^0 K_4(b,c,u,v) - \\ & 2 \sum_{du} w_d w_u^0 K_4(b,c,d,u) \end{aligned} \tag{B.9}$$

I, J and K:

To render the notation more compact, a generic overlap parameter U is introduced; indices i, j, f, g and h may therefore apply to SBFs or TBFs as appropriate.

$$U_{ij} = \begin{cases} Q_{ij} & \text{if } i,j \text{ both refer to SBFs} \\ R_{ij} & \text{if } i \text{ refers to a SBF and } j \text{ to a TBF} \\ T_{ij} & \text{if } i,j \text{ both refer to TBFs} \end{cases} \tag{B.10}$$

$$I_2(i,j) = (2l_2\sigma_\xi^2)^{-N/2} \exp\left[\frac{-U_{ii} - U_{jj} + (U_{ii} + U_{jj} + 2U_{ij})/2\sigma_B^2 l_2}{2\sigma_B^2}\right] \tag{B.11}$$

$$J_2(i,j;f) = \left(\frac{U_{if} + U_{jf}}{2l_2\sigma_B^2}\right) I_2(i,j) \tag{B.12}$$

$$I_4(i,j,f,g) = (2l_4\sigma_\xi^2)^{-N/2} \exp\left[\frac{-U_{ii} - U_{jj} - U_{ff} - U_{gg}}{2\sigma_B^2}\right] \times \exp\left[\frac{U_{ii} + U_{jj} + U_{ff} + U_{gg} + 2(U_{ij} + U_{if} + U_{ig} + U_{jf} + U_{jg} + U_{fg})}{4l_4\sigma_B^4}\right] \tag{B.13}$$

$$J_4(i,j,f,g;h) = \left(\frac{U_{ih} + U_{jh} + U_{fh} + U_{gh}}{2l_4\sigma_B^2}\right) I_4(i,j,f,g) \tag{B.14}$$

$$K_4(i,j,f,g) = \left(\frac{2Nl_4\sigma_B^4 + U_{ii} + U_{jj} + U_{ff} + U_{gg}}{4l_4\sigma_B^4} + \frac{2(U_{ij} + U_{if} + U_{ig} + U_{jf} + U_{jg} + U_{fg})}{4l_4^2\sigma_B^4}\right) I_4(i,j,f,g) \quad \text{(B.15)}$$

Instantaneous Variances

Defining, for brevity:

$$\overline{KIJJ}_4(i,j,f,g) = \overline{K}(i,j,f,g) + U_{if}U_{jg}\overline{I}_4(i,j) - U_{jg}\overline{J}_4(i,j,f) - U_{if}\overline{J}_4(i,j,g) \quad \text{(B.16)}$$

Variances:

$$\Delta Q_{bc}\Delta Q_{de} = 1/\sigma_B^4 \left\{w_b w_d\, \overline{KIJJ}_4(b,d,c,e) + w_b w_e\, \overline{KIJJ}_4(b,e,c,d) + w_c w_d\, \overline{KIJJ}_4(c,d,b,e) + w_c w_e\, \overline{KIJJ}_4(c,e,b,d)\right\} \quad \text{(B.17)}$$

$$\Delta Q_{bc}\Delta R_{du} = 1/\sigma_B^4 \left\{w_b w_d\, \overline{KIJJ}_4(b,d,c,u) + w_c w_d\, \overline{KIJJ}_4(c,d,b,u)\right\} \quad \text{(B.18)}$$

$$\Delta R_{bu}\Delta R_{cv} = 1/\sigma_B^4\, w_b w_c\, \overline{KIJJ}_4(b,c,u,v) \quad \text{(B.19)}$$

$$\Delta Q_{bc}\Delta w_d = 1/\sigma_B^2 \left\{ w_b \left(\overline{J}_4(b,d,c) - Q_{bc}\overline{I}_4(b,d) \right) + w_c \left(\overline{J}_4(c,d,b) - Q_{bc}\overline{I}_4(c,d) \right) \right\} \quad \text{(B.20)}$$

$$\Delta R_{bu}\Delta w_d = 1/\sigma_B^2 \, w_b \left\{ \overline{J}_4(b,d,u) - R_{bu}\overline{I}_4(b,d) \right\} \quad \text{(B.21)}$$

$$\Delta w_b \Delta w_c = \overline{I}_4(b,c) - Q_{bc}\overline{I}_2(b)\overline{I}_2(c) \quad \text{(B.22)}$$

Other Quantities:

$$l_2 = \frac{2\sigma_\xi^2 + \sigma_B^2}{2\sigma_B^2\sigma_\xi^2} \quad \text{(B.23)}$$

$$l_4 = \frac{4\sigma_\xi^2 + \sigma_B^2}{2\sigma_B^2\sigma_\xi^2} \quad \text{(B.24)}$$

On-line Noise Quantites

$$\begin{aligned} \frac{2}{3^{-N/2}} E_G = {} & (w^*)^2 \left(K \exp[-Q^*/3] + K(K-1)\exp[-2Q^*/3 + C^*/3] \right) \\ & + \left(K \exp[-1/3] + K(K-1)\exp[-2/3] \right) \\ & - 2w^* \left(K \exp\left[\frac{-Q^* - 1 + (Q^* + 1 + 2R^*)/3}{2} \right] \right. \\ & \left. + K(K-1)\exp\left[\frac{-Q^* - 1 + (Q^* + 1 + 2S^*)/3}{2} \right] \right) \end{aligned} \quad \text{(B.25)}$$

Bibliography

Amari, S. (1993). Backpropagation and stochastic gradient descent learning. *Neurocomputing*, **5**, 185–196.

Amari, S., Murata, N., Müller, K., Finke, M., and Yang, H. (1996). Statistical theory of overtraining – is cross-validation asymptotically effective? In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 176–182. MIT Press.

Barber, D., Saad, D., and Sollich, P. (1996). Finite-size effects in on-line learning of multilayer neural networks. *Europhys. Lett.*, **34**, 151–156.

Barron, A. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, **39**, 930–945.

Barron, A. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning*, **14**, 115–133.

Biehl, M. and Schwarze, H. (1995). Learning by online gradient descent. *J. Phys. A: Math. Gen.*, **28**, 643.

Biehl, M., Riegler, P., and Wohler, C. (1996). Transient dynamics of online learning in 2-layered neural networks. *J. Phys. A: Math. Gen.*, **29**, 4769–4780.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford.

Botros, S. and Atkeson, C. (1991). Generalization properties of radial basis functions. In R. Lippman, J. Moody, and D. Touretzky, editors, *Advances in Neural Information Processing Systems*, volume 3, pages 707 – 713. Morgan Kaufmann, San Mateo, CA.

Broomhead, D. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, **2**, 321–355.

Bruce, A. and Saad, D. (1994). Statistical mechanics of hypothesis evaluation. *J. Phys. A: Math. Gen.*, **27**, 3355 – 3363.

Casdagli, M. (1989). Nonlinear prediction of chaotic time series. *Physica*, **35D**, 335–356.

Chen, S., Billings, S., and Luo, W. (1989). Orthogonal least squares methods and their application to non-linear system identification. *Intl. Journal of Control*, **50**(5), 1873–1896.

Chen, S., Cowan, C., and Grant, P. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, **2**(2), 302–309.

Chen, S., Chng, E., and K.Alkadhimi (1996). Regularised orthogonal least squares algorithm for constructing radial basis function networks. *International Journal of Control*, **64**(5), 829–837.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society B*, **39**(1), 1–38.

Devijver, P. (1982). *Pattern Recognition: a Statistical Approach*. Prentice-Hall, Englewood Cliffs, NJ.

Dunmur, A. and Wallace, D. (1993). Learning and generalisation in a linear perceptron stochastically trained with noisy data. *J. Phys. A: Math. Gen.*, **26**, 5767 – 5779.

Fununaga, K. and Hayes, R. (1989). The reduced Parzen classifier. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**(4), 423–425.

Hansen, L. (1993). Stochastic linear learning: Exact test and training error averages. *Neural Networks*, **6**, 393–396.

Hartman, E., Keeler, J., and Kowalski, J. (1990). Layered neural networks with gaussian hidden units as universal approximators. *Neural Computation*, **2**, 210–215.

Haussler, D. (1989). Generalizing the PAC model for neural net and other learning applications. Technical Report UCSC-CRL-89-30, University of California, Santa Cruz.

Haussler, D. (1994). The probably approximately correct (PAC) and other learning models. In A. Meyrowitz and S. Chipman, editors, *Foundations of Knowledge Acquisition: Machine Learning*, chapter 9. Kluwer, Boston.

Hertz, J., Krogh, A., and Palmer, R. (1989). *Introduction to the Theory of Neural Computation*, volume I of *Santa Fe Institute Lecture Notes*. Addison Wesley.

Heskes, T. and Kappen, B. (1991). Learning processes in neural networks. *Phys. Rev. A.*, **44**, 2718–2726.

Holden, S. and Rayner, P. (1995). Generalization and PAC learning: some new results for the class of generalized single-layer networks. *IEEE Trans. on Neural Networks*, **6**(2), 368–380.

Kampen, N. V. (1992). *Stochastic Processes in Physics and Chemistry*. North-Holland, Amsterdam.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, **43**, 59–69.

Leen, T. and Orr, G. (1994). Optimal stochastic search and adaptive momentum. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 477–484, San Mateo, CA. Morgan Kaufmann.

Levin, E., Tishby, N., and Solla, S. (1989). A statistical approach to learning and generalisation in layered neural networks. In *Colt '89: 2nd Workshop on Computational Learning Theory*, pages 245–260.

Lowe, D. (1995). On the use of nonlocal and non positive definite basis functions in RBF networks. Number 409 in Proc. 4th IEE Int'l. Conference on ANN, pages 206–211. Cambridge, UK.

MacKay, D. (1992). Bayesian interpolation. *Neural Computation*, **4**, 415–447.

Mackay, D. (1992). *Bayesian Methods for Adaptive Models*. Ph.D. thesis, Caltech.

Moody, J. and Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, **1**(2), 281–294.

Musavi, M., Ahmed, W., *et al.* (1992). On the training of radial basis function classifiers. *Neural Networks*, **5**(4), 595–603.

Neal, R. (1992). Bayesian training of backpropagation networks by the hybrid monte carlo method. Technical report, Dept. of Computer Science, University of Toronto.

Nilsson, N. (1965). *Learning Machines*. New York: McGraw-Hill. Reprinted as *The Mathematical Foundations of Learning Machines*, Morgan Kaufmann, (1990).

Niranjan, M. and Fallside, F. (1990). Neural networks and radial basis functions in classifying static speech patterns. *Computer Speech and Language*, **4**, 275–289.

Niyogi, P. and Girosi, F. (1994). On the relationship between generalization error, hypothesis complexity and sample complexity for radial basis functions. Technical Report A.I. Memo 1467, AI Laboratory, Massachusetts Institute of Technology.

Omohundro, S. (1987). Efficient algorithms with neural network behaviour. *Complex Systems*, **1**, 273–347.

Orr, M. (1993). Regularised centre recruitment in radial basis function networks. Technical report, Centre for Cognitive Science, University of Edinburgh.

Orr, M. (1995). Regularisation in the selection of radial basis function centres. *Neural Computation*, **7**(3), 606–623.

Park, J. and Sandberg, I. (1993). Approximation and radial basis function networks. *Neural Computation*, **5**(2), 305–316.

Poggio, T. and Girosi, F. (1990a). Networks for approximation and learning. *Proceedings of the IEEE*, **78**(9), 1481–1497.

Poggio, T. and Girosi, F. (1990b). Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, **247**, 978–982.

Powell, M. (1987). Radial basis functions for multivariable interpolation: a review. In J. Mason and M. Cox, editors, *Algorithms for Approximation*, pages 143–167. Clarendon Press, Oxford.

Radons, G., Schuster, H., and Werner, D. (1990). Drift and diffusion in backpropagation learning. In R. Eckmiller *et al.*, editors, *Parallel Processing in Neural Systems and Computers*. Elsevier Science Publishers, North Holland.

Rawlings, J. (1988). *Applied Regression Analysis*. Wadsworth & Brooks/Cole, Pacific Grove, CA.

Riegler, P. (1997). *Dynamics of On-line Learning in Neural Networks*. Ph.D. thesis, Institut füt Theoretische Physik, Universität Würzburg.

Riegler, P. and Biehl, M. (1995). On-line backpropagation in two-layered neural networks. *J. Phys. A: Math. Gen.*, **28**, L507–513.

Rögnvaldsson, T. (1994). On langevin updating in multilayer perceptrons. *Neural Computation*, **6**, 916–926.

Saad, D. and Rattray, M. (1997). Globally optimal parameters for on-line learning in multilayer neural networks. Preprint.

Saad, D. and Solla, S. (1995a). Exact solution for on-line learning in multilayer neural networks. *Phys. Rev. Lett.*, **74**, 4337–4340.

Saad, D. and Solla, S. (1995b). On-line learning in soft committee machines. *Phys. Rev. E.*, **52**, 4225–4243.

Saad, D. and Solla, S. A. (1997). Learning with noise and regularizers in multilayer neural networks. In M. C. Mozer, J. M. I., and P. T., editors, *Advances in Neural Information Processing Systems*, volume 9, pages 260–266. MIT Press.

Schwarze, H. (1993). Learning a rule in a multilayer neural network. *J. Phys. A: Math. Gen.*, **26**, 5781–5794.

Scott, D. (1992). *Multivariate Density Estimation: Theory, Practice and Visualization*. New York: John Wiley.

Valiant, L. (1984). A theory of the learnable. *Comm. ACM*, **27**, 1134–1142.

Watkin, T., Rau, A., and Biehl, M. (1993). The statistical mechanics of learning a rule. *Reviews of Modern Physics*, **65**, 499–556.

Webb, A. (1994). Functional approximation by feed-forward networks: a least-squares approach to generalization. *IEEE Trans. on Neural Networks*, **5**(3), 363–371.

Wiegerinck, W. and Heskes, T. (1996). How dependencies between successive examples affect on-line learning. *Neural Computation*, **8**(8), 1743–1766.

Wolpert, D. (1992). On the connection between in-sample testing and generalisation error. *Complex Systems*, **6**, 47–94.

Wolpert, D. (1996a). The lack of a-priori distinctions between learning algorithms. *Neural Computation*, **8**(7), 1341–1390.

Wolpert, D. (1996b). The existence of a-priori distinctions between learning algorithms. *Neural Computation*, **8**(7), 1391–1420.