The Dynamics of On-line Learning in Radial Basis Function Networks

Jason A.S. Freeman[1] and David Saad[2]
[1]Centre for Cognitive Science, University of Edinburgh, Edinburgh EH8 9LW, UK.
[2]Department of Computer Science & Applied Mathematics
University of Aston
Birmingham B4 7ET, UK.

April 25, 1997

Abstract

On-line learning is examined for the Radial Basis Function Network, an important and practical type of neural network. The evolution of generalization error is calculated within a framework which allows the phenomena of the learning process, such as the specialization of the hidden units, to be analyzed. The distinct stages of training are elucidated, and the role of the learning rate described. 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 derivation of maximal and optimal learning rates. As well as finding the evolution of the mean system parameters, the variances of these parameters are derived and shown to be typically small. Finally, the analytic results are strongly confirmed by simulations.

1 Introduction

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 know the average deviation of the network's estimate from the target function - this is known as *generalization error*. From a practical perspective, generalization error is unavailable; it can be approximated by utilising a *test set*, again generated from f_T, which is distinct from the training set. It would be very useful if it were possible to make general statements concerning the generalization error that could be expected in the average case. In this paper, we calculate the evolution of the average generalization error, as well as the evolution of key parameters that describe the learning system, for the Radial Basis Function Network (RBF).

Several frameworks are available which facilitate analytic investigation of learning and generalization in supervised neural networks, such as the statistical physics methods (see [1] for a review), the Bayesian framework (e.g., [2]) and the PAC method [3]. 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, [4] and references therein). It has proved very difficult to obtain general results for the commonly used multilayer networks, such as the sigmoid multi-layer perceptron (MLP) and the RBF.

For RBFs, some analytic studies exist which focus primarily on generalization error: in [5, 6], average case analyses are performed employing a Bayesian framework to study RBFs under a

stochastic training paradigm. In [7], a bound on generalization error is derived under the assumption that the training algorithm finds a globally optimal solution. Details of studies of RBFs from the perspective of the PAC framework can be found in [8] and references therein. These methods focus on a training scenario in which a model is trained on a *fixed set* of examples using a stochastic training method.

There are several studies which are concerned with understanding the dynamics of *on-line* gradient descent training scenarios, whereby network parameters are modified after each presentation of an example [9, 10, 11]; these studies examine the evolution of system parameters primarily in the asymptotic regime. A similar method, based on examining the dynamics of overlaps between characteristic system vectors in on-line training scenarios has been suggested in [12, 13, 14, 15] for investigating the learning dynamics in the 'soft committee machine' (SCM). 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 [15, 17]. The training dynamics in discrete systems has been examined by several authors employing a variety of techniques [18, 19, 20, 21, 22], some of which offered improved training algorithms.

We present a method for analyzing the behaviour of RBFs in an on-line learning scenario 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 the means and the variances of these variables 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 previously been applied to MLPs [13, 14, 15]; earlier work on RBFs from an on-line learning perspective can be found in [17].

2 The RBF Network and the On-line Learning Paradigm

RBF networks have been successfully employed to perform supervised learning in many real-world tasks; they have proved a valuable alternative to MLPs. These tasks include chaotic time-series prediction [23], speech recognition [24] and data classification [25].

The RBF is a universal approximator for continuous mappings - it can approximate any continuous function to arbitrary accuracy given a sufficient number of hidden units [26]. The RBF architecture consists of a two-layer network (see figure 1) in which each layer is fully connected to its successor. For simplicity, a single output node is utilised throughout the analysis. The activation function of the hidden nodes is radially symmetric in input space; the magnitude of the activation given a particular datapoint is usually a decreasing function of the distance between the input vector of the datapoint and the centre of the basis function. The output layer computes a linear combination of the activations of the basis functions, parameterised by the weights $\boldsymbol{w}$ between hidden and output layers. The function computed by an RBF network with K hidden units is therefore:

$$f_S(\boldsymbol{\xi}) \;=\; \sum_{b=1}^{K} w_b \, s_b(\boldsymbol{\xi}) \;=\; \boldsymbol{w} \cdot \boldsymbol{s} \tag{1}$$

where $\boldsymbol{\xi}$ is the input vector applied to the input layer, s_b denotes the response of basis function b, and $\boldsymbol{s}$ represents the vector of hidden unit responses of the network.

The most common choice for the basis functions is the Gaussian function, which will be employed as the hidden unit transfer function throughout the paper. Therefore the response of basis function b to inputvector $\boldsymbol{\xi}$ is:

$$s_b(\boldsymbol{\xi}) \;=\; \exp\left(-\frac{\|\boldsymbol{\xi} - \boldsymbol{m}_b\|^2}{2\sigma_B^2}\right) \tag{2}$$

where 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.

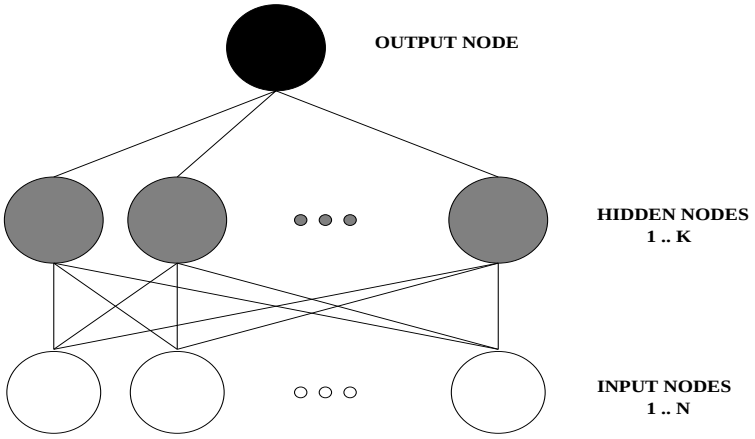


Figure 1: RBF network architecture

Two general methodologies exist which allow the adjustment of the parameters of the RBF to approximate the target function. One approach involves fixing the parameters of the hidden layer (both the basis function centres and widths) using an unsupervised technique such as clustering, setting a centre on each data point of the training set, or even picking random values (for a review see [27]). This leaves only the hidden-to-output weights $\boldsymbol{w}$ to adapt, which makes the problem linear in those weights. Although fast to train, this approach generally results in sub-optimal networks since the basis function parameters are not fixed with respect to the targets in the training data, and do not take account of the values of $\boldsymbol{w}$. The alternative is to adapt the hidden layer parameters, either just the centre positions or both centre positions and widths, in conjunction with the adaptation of $\boldsymbol{w}$. This renders the problem non-linear in the adaptable parameters, and hence requires an optimization technique, such as gradient descent, to estimate the parameters. The second approach is computationally more expensive, but usually leads to greater accuracy of approximation. This paper investigates the non-linear approach in which basis function centres are continuously modified using gradient descent to allow convergence to more optimal models.

There are two 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; obviously one may employ a method which is a compromise between the two extremes. 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 techniques such as the replica method are unnecessary.

3 Generalization and System Dynamics

It is difficult to examine generalization without having some *a priori* knowledge of the target function since for any finite number of datapoints, there are an infinite number of functions that will fit these points exactly. In this work, we utilise a student-teacher framework, in which a

teacher network produces the training data which is then learned by the student. This has the advantage that we can control the learning scenario precisely, facilitating the investigation of cases such as the exactly realizable case, in which the student architecture matches that of the teacher, the over-realizable case, in which the student can represent functions that cannot be achieved by the teacher, and the unrealizable case in which the student has insufficient representational power to emulate the teacher.

A training set consists of P input-output pairs $(\boldsymbol{\xi}^\mu, y^\mu)$ where $1 \leq \mu \leq P$. In the training scenario examined here, the components of the typical N dimensional input vector $\boldsymbol{\xi}^\mu$ are chosen as uncorrelated Gaussian random variables of mean 0, variance σ_ξ^2, while the scalar output y^μ is generated by applying $\boldsymbol{\xi}$ to the deterministic teacher RBF. This represents a *general* training scenario since, being universal approximators, RBF networks can approximate any continuous mapping to a desired degree. Noise is not employed in this paper; this will be investigated in a further publication. The mapping implemented by the teacher is denoted by f_T; the vector of hidden-to-output weights of the teacher is represented by an M dimensional vector $\boldsymbol{w}^0$ while the centre of teacher basis function (TBF) u is denoted by $\boldsymbol{n}_u$. The vector of teacher basis function responses to input vector $\boldsymbol{\xi}$ is represented by an M dimensional vector $\boldsymbol{t}$. For simplicity, the TBF widths are equal to those of the student; the framework does allow them to differ, but this complicates matters greatly without adding much insight. The function computed by the teacher is therefore:

$$f_T(\boldsymbol{\xi}) = \sum_{u=1}^{M} w_u^0 \exp\left(-\frac{\|\boldsymbol{\xi} - \boldsymbol{n}_u\|^2}{2\sigma_B^2}\right) \;=\; \boldsymbol{w}^0 \cdot \boldsymbol{t} \tag{3}$$

We approach the problem of calculating system evolution by replacing the set of N-dimensional vectors $\boldsymbol{m}$, which describe the position in input space of the student basis functions, by a set of macroscopic variables representing the means and variances of the overlaps: $Q_{bc} = \boldsymbol{m}_b \cdot \boldsymbol{m}_c, R_{bu} = \boldsymbol{m}_b \cdot \boldsymbol{n}_u$ and $T_{uv} = \boldsymbol{n}_u \cdot \boldsymbol{n}_v$. We will concentrate on the evolution of the means of these quantities; the relevance of their variances will be quantified and examined as well. The evolution of the system will be described in terms of the evolution of these macroscopic variables and of the hidden-to-output weights $\boldsymbol{w}$.

The definition of generalization error that we employ is the most common in the neural networks literature - the quadratic deviation between f_T and f_S:

$$E_G = \left\langle \frac{1}{2}\left(f_S - f_T\right)^2 \right\rangle \tag{4}$$

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

Substituting equations (2) and (3) into (4) gives:

$$E_G = \frac{1}{2}\left\{ \sum_{bc} w_b w_c \left\langle s_b s_c \right\rangle + \sum_{uv} w_u^0 w_v^0 \left\langle t_u t_v \right\rangle - 2\sum_{bu} w_b w_u^0 \left\langle s_b t_u \right\rangle \right\} \tag{5}$$

The variables $b, c, \ldots\ u, v, \ldots$ and will represent student and teacher centers respectively, running from 1 to K and to M accordingly. We assume the input distribution to be Gaussian, so the averages are Gaussian integrals and can be performed analytically. 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; the full expression is given in the appendix.

3.1 System Dynamics

The learning dynamics in this work follows 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 $1/N$. Expressions for the time evolution of the mean overlaps of $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be derived:

$$\begin{aligned} \langle\, \Delta \boldsymbol{Q}_{bc} \,\rangle \;\; = \;\; & \frac{\eta}{N\sigma_B^2} \langle\, [\delta_b(\boldsymbol{\xi} - \boldsymbol{m}_b^p) \cdot \boldsymbol{m}_c^p + \delta_c(\boldsymbol{\xi} - \boldsymbol{m}_c^p) \cdot \boldsymbol{m}_b^p] \,\rangle + \\ & \left(\frac{\eta}{N\sigma_B^2}\right)^2 \langle\, \delta_b \delta_c (\boldsymbol{\xi} - \boldsymbol{m}_b^p) \cdot (\boldsymbol{\xi} - \boldsymbol{m}_c^p) \,\rangle \end{aligned} \tag{6}$$

$$\langle\, \Delta \boldsymbol{R}_{bu} \,\rangle = \frac{\eta}{N\sigma_B^2} \langle\, \delta_b (\boldsymbol{\xi} - \boldsymbol{m}_b^p) \cdot \boldsymbol{n}_u \,\rangle \tag{7}$$

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, we use the same learning rate but scale it differently (with $1/K$, in agreement with results obtained by Riegler [28] for the MLP), yielding:

$$\langle\, \Delta w_b \,\rangle = \frac{\eta}{K} \langle\, (f_T - f_S) s_b \,\rangle \tag{8}$$

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[28].

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 the appendix.

Iterating the difference equations (6), (7) and (8), allows the evolution of the learning process to 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 with equation (5) to track the evolution of generalization error.

3.2 Variance and the Thermodynamic Limit

Previous work in this area [12, 13, 14, 15] 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 makes the macroscopic variables self-averaging, 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 (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). 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 [9] and also in [29] for the simpler case of the SCM; we have extended it to deal with adaptive hidden-to-output weights (see also [15]).

To quantify the effect of the variances we will derive a set of dynamical equations, parallel to those representing the dynamics of the means, for describing the dynamics of the variances. As the learning rate is usually small we will focus on first order terms in η, which dominate the dynamics, and ignore update terms of order η^2. Casting the update equations (6, 7 and 8) 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:

$$a^{p+1} = a^p + \frac{\eta}{L_a} F_a \tag{9}$$

We then assume (similar to [29]) that the update function F and the parameter a can be written in terms of a mean and fluctuation such that:

$$F_a = \overline{F}_a + \tilde{F}_a \quad \text{and} \quad a = \overline{a} + \sqrt{\tfrac{\eta}{L_a}}\tilde{a} \tag{10}$$

where $\overline{a}$ denotes an average value and $\tilde{a}$ represents a fluctuation due to the randomness of the example. The static correction terms of [29] are neglected, as in [9], as they are much smaller than the included fluctuation terms.

Combining eqns (9) and (10), and averaging with respect to the input distribution, we arrive at a set of coupled difference equations which describe the evolution of the variances:

$$\Delta \left\langle \tilde{a}\tilde{b} \right\rangle = \frac{\eta}{\sqrt{L_a L_b}} \left(\sum_c \langle \tilde{a}\tilde{c} \rangle \frac{\partial \overline{F}_b}{\partial \overline{c}} + \sum_c \left\langle \tilde{b}\tilde{c} \right\rangle \frac{\partial \overline{F}_a}{\partial \overline{c}} + \left\langle \tilde{F}_a \tilde{F}_b \right\rangle \right) \tag{11}$$

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 $\left\langle \tilde{F}_a \tilde{F}_b \right\rangle$ are given in the appendix.

It has been shown that the variances must vanish asymptotically for realizable cases [9], and we will show in section 4.6 that they are small enough throughout the evolution of the system to allow a description of the system in terms of the evolution of the means.

4 Analysing the Learning Process

Although the framework enables us to consider a wide range of cases we will limit the experiments and the analysis in this paper to realizable cases where the number of student basis functions (SBFs) equals the number of teacher basis functions (TBFs).

The system evolutions described below are obtained by iterating the difference equations (6), (7) and (8) 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 we employ. The evolutions computed describe the mean behaviour, assuming the variances are negligible; these evolutions can then be used to find the evolution of generalization error via equation ((5).

4.1 The Importance of the Learning Rate

With all the 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 (here, $\eta = 0.1$), 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, but the time period required may be prohibitively large (figure 2(a), dotted curve). 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 2(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 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 [12, 13, 14, 15].

4.2 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 continous mapping, and noise is not employed.

For this particular task we choose the teacher 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 figures 2(a) to 2(d); figure 2(a) (solid curve) shows the evolution of generalization error, calculated from equation (5), while figures 2(b) to 2(c) show the evolution of the equations for the means of $\boldsymbol{R}$, $\boldsymbol{Q}$ and $\boldsymbol{w}$ respectively, calculated by iterating equations (6), (7) and (8) 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 [14, 15]. 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 2(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 2(b) to 2(d)). The difference in the overlaps $\boldsymbol{R}$ between student centre vectors and teacher centre vectors (figure 2(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 2(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; $\forall b, w_b = 1$). 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.

4.3 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 4.2 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 3(a) and 3(b) respectively. 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 3(a), dashed curve) with that for the previous task (figure 3(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 the overlaps between student and teacher centre vectors (figure 3(b)); this should be compared with figure 2(b) which denotes the evolution of the overlaps in the previous task. Note that the plateaus in the overlaps (figure 2(b), $P = 4000$ to $P = 5 \times 10^4$) are not found for the asymmetric task.

The elimination of the symmetric phase is an extreme result caused by the small size of the student network (3 hidden units). For networks with many hidden units, one finds instead 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 used, but this adds another parameter to the training process.

4.4 Analysing the Symmetric and Symmetry-Breaking Phases

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 η, so terms of η^2 may be 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 that the student norms Q_{bc} are very similar, 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 [14], which is consistent with the numerical results, 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 equations (6), (7) and (8) under these restrictions is still rather complicated. However,

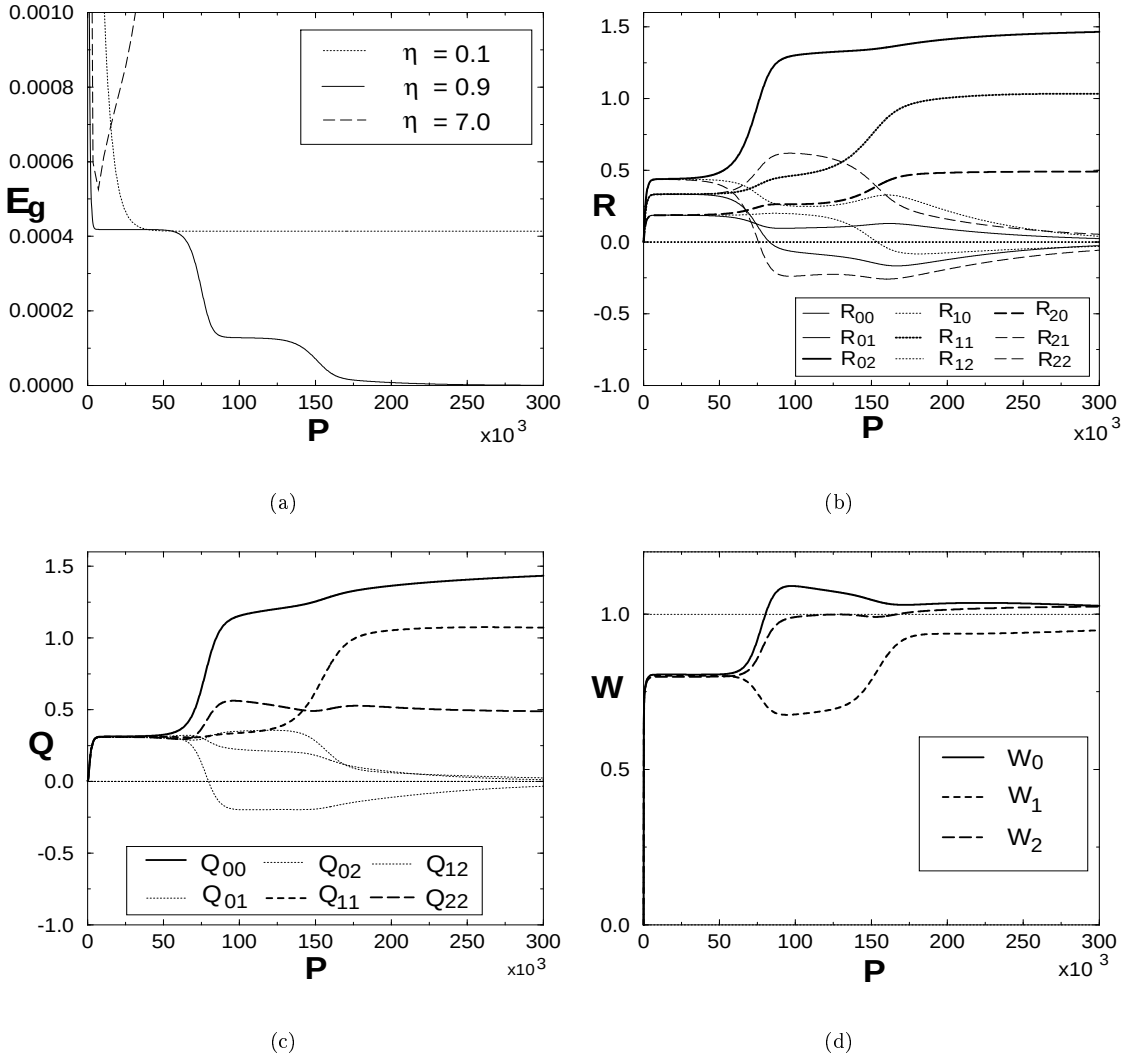
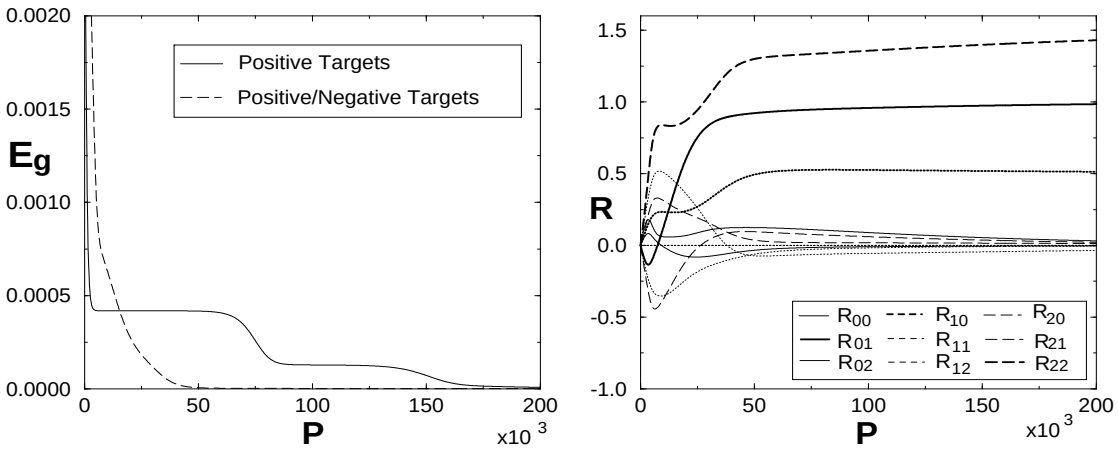


Figure 2: The exactly realizable scenario with positive TBFs. Three SBFs learn a graded, uncorrelated teacher of three TBFs with $T_{00} = 0.5, T_{11} = 1.0$ and $T_{22} = 1.5$. All teacher hidden-to-output weights are set to 1. Figure (a) describes the evolution of the generalization error as a function of the number of examples for several different learning rates ($\eta = 0.1, 0.9, 5.0$); (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.



(a) (b)

Figure 3: 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 $\boldsymbol{R}$.

since we are primarily interested in large systems, i.e., large K, we examine 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; fixed point values will be denoted like R^*:

$$R^* = \frac{1}{K\left(1+\sigma_B-\sigma_B\exp\left[\left(\frac{1}{2\sigma_B}\right)\frac{\sigma_B+1}{\sigma_B+2}\right]\right)} \tag{12}$$

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. We map the equations of motion (6), (7) 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 we are in the small η regime, components which are orthogonal to the space spanned by the teacher vectors, $\boldsymbol{m}_b^-$ 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 [16] 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 ([14]), 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 simulation. This difference 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.

4.5 Calculating the Convergence

The speed and conditions of convergence of the online gradient descent process is of great interest, both practically and theoretically. To investigate this for the RBF in the realizable case, we again use an isotropic teacher, 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: $Q_{bc} = Q\delta_{bc} + C(1-\delta_{bc}), R_{bu} = R\delta_{bu} + S(1-\delta_{bu})$ and $w_b = w$, controlled by equations (6), (7) and (8). Note that we do not expect the variances 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 4(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 [17]. Figure 4(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 [15]). 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 4(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.

4.6 Quantification of the Variance

To demonstrate that it is reasonable to consider only the mean of the updates of the system parameters, we present results quantifying 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 in [29].

To examine the effect of the variance we use a training scenario 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 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 (6), (7), (8) and (11) were iterated from these initial conditions until the means had reached an approximately steady state, thus providing a trajectory for each variance.

In figures 5(a) and 5(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 5(c) and 5(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, we believe, due to the fact that terms of η^2 are discarded in the theory.

4.7 Simulations

To demonstrate the validity of the theoretical average-case results, we compared the evolution of the system found by iterating equations (6), (7) and (8) 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.

We present the results from 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. Figure 6(a) shows theoretical versus empirical generalization error - the theoretical value is always within one standard deviation of the empirical value. In figures

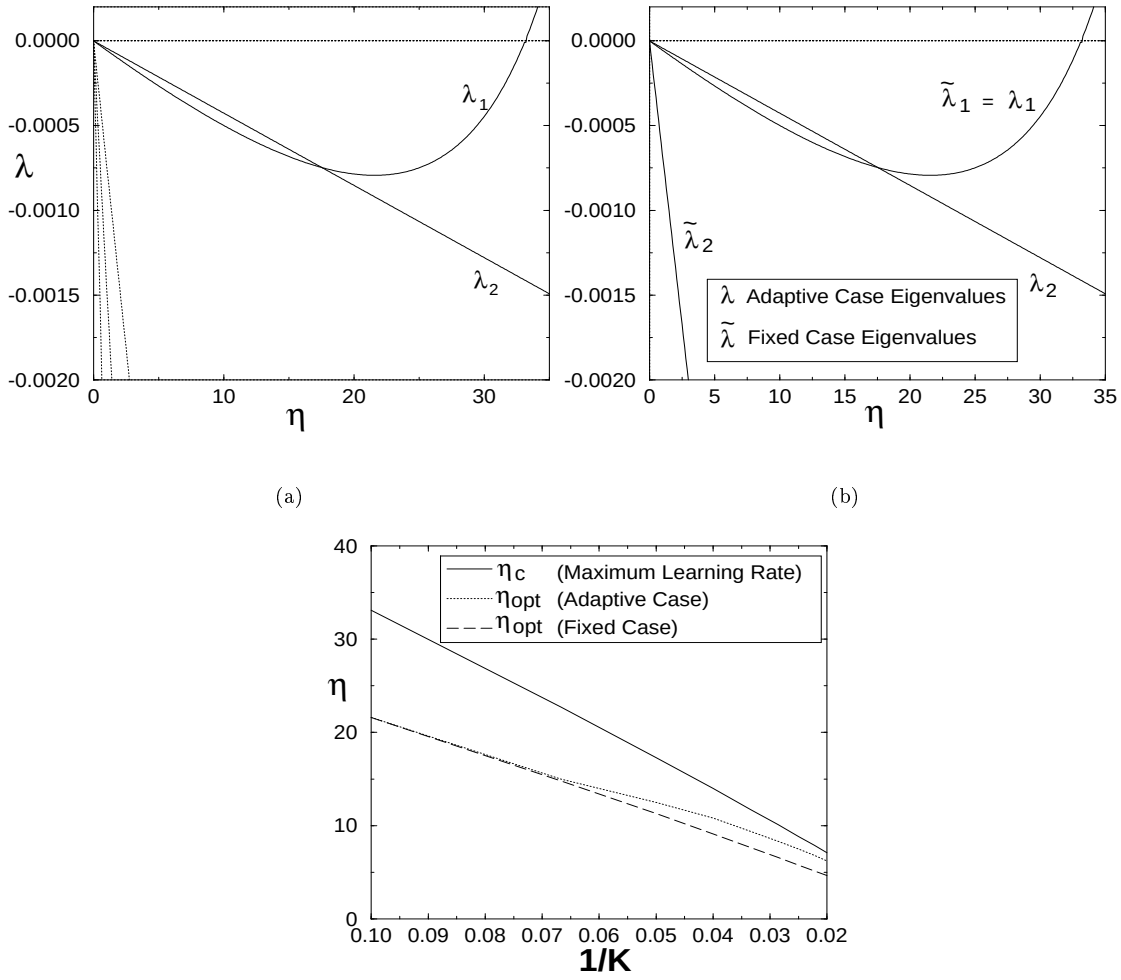
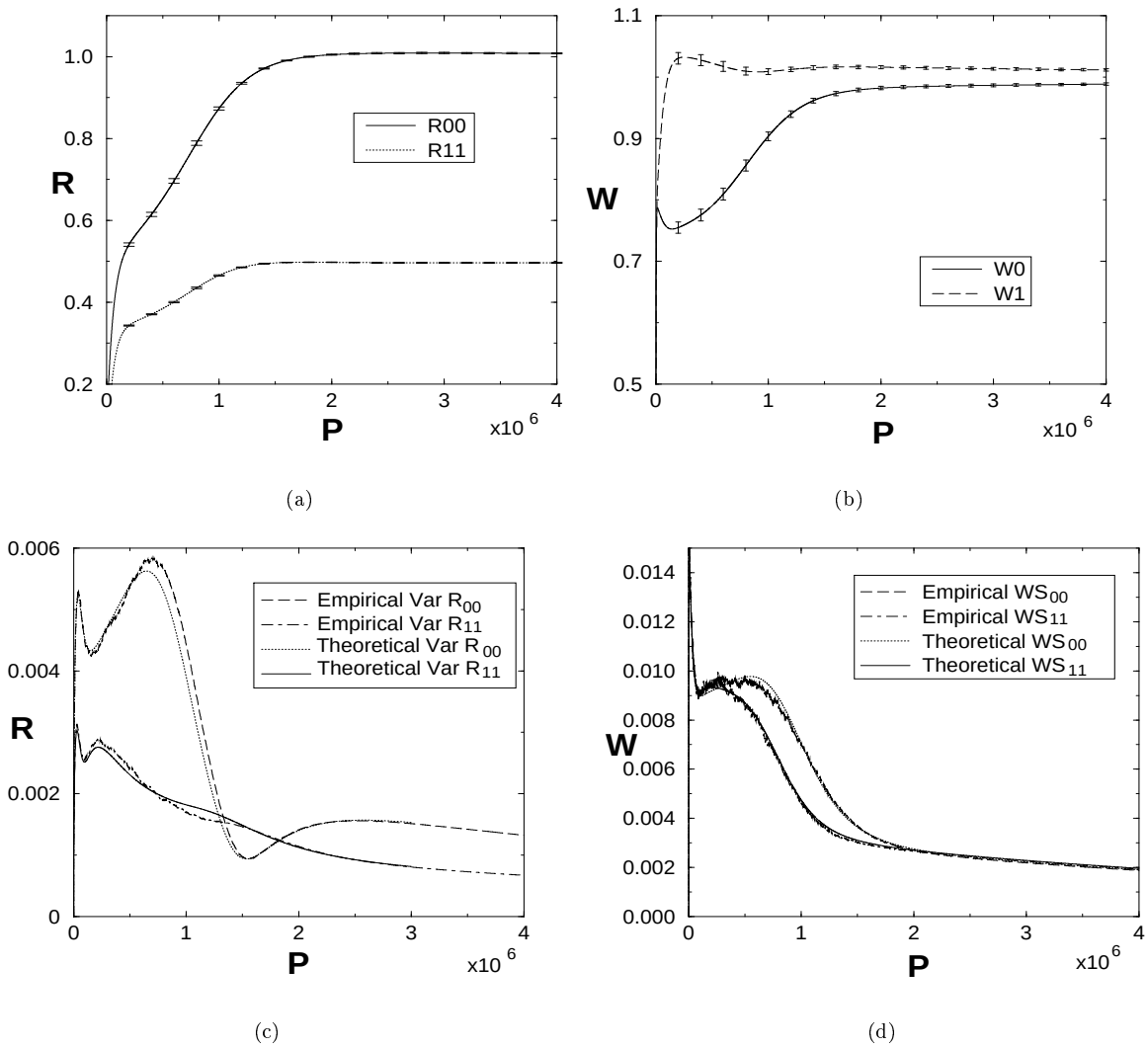


Figure 4: Convergence Phase. Figure (a) shows the eigenvalues for the system with adaptive hidden-to-output weights. Only λ_1 and λ_2 are significant; λ_1 controls the maximum learning rate, while λ_2 can influence the optimal learning rate. Figure (b) compares the eigenvalues for systems with adaptive 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 $\eta_{\rm c}$ scales with $1/K$; for fixed hidden-to-output weights, the optimal learning rate $\eta_{\rm opt}$ also scales with $1/K$, while for adaptive weights, $\eta_{\rm opt}$ rapidly approaches $\eta_{\rm c}$.

6(b), 6(c) and 6(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.



(a) (b)

(c) (d)

Figure 5: 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.

5 Conclusion

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

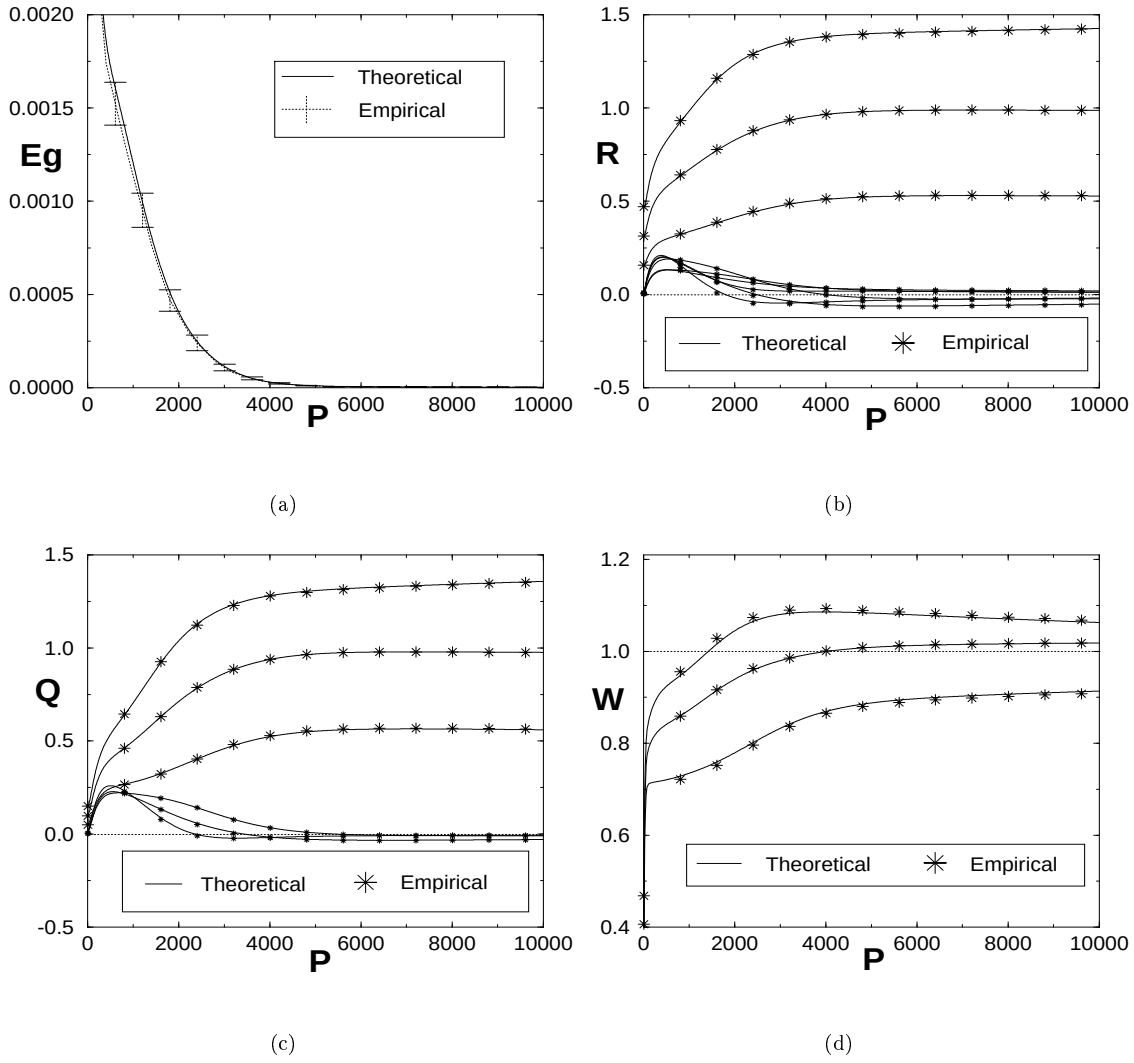


Figure 6: Comparison of theoretical results with simulations. The simulation results are averaged over 50 trials; the labels of the student hidden units were permuted where necessary to make the averages meaningful. Empirical generalization error was approximated with the test error on a 1000 point test 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 $\sigma_B^2 = 1$.

features of the learning process, such as the specialization of the hidden units.

The four distinct stages of training were highlighted - initially there is a short transient phase as the parameters move from their initial values into the symmetric phase, in which the hidden units are undifferentiated. Specialization gradually develops in the third, symmetry-breaking, phase, as the hidden units move towards their particular destinations; finally there is a convergence phase in which the parameters asymptotically reach their final values. The role of the learning rate was also examined - with a small learning rate η, training proceeds unnecessarily slowly, with a long trapping time in the symmetric phase. With η too high, the process does not converge to the correct fixed point; the magnitudes of the student centre vectors grow until the centre plays no part in the learning process. Between these extremes lies a range in which the process converges quickly to the correct target.

The relative importance of the stages of training depends to a large extent on the nature of the task itself. When the task is highly symmetric, the symmetric phase becomes dominant; in this case it would be desirable to introduce artificial methods of breaking the symmetry of the student. For very asymmetric tasks, the symmetric phase may be over quickly or even non-existent. Since in practical use the task is usually understood poorly, it is important to understand the behaviour of the network over a whole range of tasks.

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

As a next stage we intend to analyze the use of noise and regularizers within on-line learning for the RBF. We expect the addition of output noise to the teacher to affect the asymptotic values of the overlaps, and produce non-zero asymptotic generalization error; it may also change the length and values of the overlaps during the symmetric phase. We also expect that a learning rate decay scedule will be required for converging to the optimal generalization error. The addition of input noise to the teacher is expected to have a similar effect, perhaps with the sensitivity of the training process to the noise being greater.

Appendix

Generalization Error:

$$E_G = \frac{1}{2}\left\{\sum_{bc} w_b w_c I_2(b,c) + \sum_{uv} w_u^0 w_v^0 I_2(u,v) - 2\sum_{bu} w_b w_u^0 I_2(b,u)\right\} \tag{13}$$

$\Delta \boldsymbol{Q}, \Delta \boldsymbol{R}$ and $\Delta \boldsymbol{w}$:

$$\begin{aligned}\langle \Delta Q_{bc}\rangle &= \frac{\eta}{N\sigma_B^2}\left\{w_b\left[\overline{J}_2(b;c) - Q_{bc}\overline{I}_2(b)\right] + w_c\left[\overline{J}_2(c;b) - Q_{bc}\overline{I}_2(c)\right]\right\} + \\ &\left(\frac{\eta}{N\sigma_B^2}\right)^2 w_b w_c\left\{\overline{K}_4(b,c) + Q_{bc}\overline{I}_4(b,c) - \overline{J}_4(b,c;b) - \overline{J}_4(b,c;c)\right\}\end{aligned} \tag{14}$$

$$\langle \Delta R_{bu}\rangle = \frac{\eta}{N\sigma_B^2} w_b\left\{\overline{J}_2(b;u) - R_{bu}\overline{I}_2(b)\right\} \tag{15}$$

$$\langle \Delta w_b\rangle = \frac{\eta}{K}\overline{I}_2(b) \tag{16}$$

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

$$\overline{I}_2(b) = \sum_u w_u^0 I_2(b,u) - \sum_d w_d I_2(b,d) \tag{17}$$

$$\overline{J}_2(b;c) = \sum_u w_u^0 J_2(b,u;c) - \sum_d w_d J_2(b,d;c) \tag{18}$$

$$\begin{aligned}\overline{I}_4(b,c) &= \sum_{de} w_d w_e I_4(b,c,d,e) + \sum_{uv} w_u^0 w_v^0 I_4(b,c,u,v) - \\ &2\sum_{du} w_d w_u^0 I_4(b,c,d,u)\end{aligned} \tag{19}$$

$$\begin{aligned}\overline{J}_4(b,c;f) &= \sum_{de} w_d w_e J_4(b,c,d,e;f) + \sum_{uv} w_u^0 w_v^0 J_4(b,c,u,v;f) - \\ &2\sum_{du} w_d w_u^0 J_4(b,c,d,u;f)\end{aligned} \tag{20}$$

$$\begin{aligned}\overline{K}_4(b,c) &= \sum_{de} w_d w_e K_4(b,c,d,e) + \sum_{uv} w_u^0 w_v^0 K_4(b,c,u,v) - \\ &2\sum_{du} w_d w_u^0 K_4(b,c,d,u)\end{aligned} \tag{21}$$

I, J and K:

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

$$U_{ij} = \begin{cases} Q_{ij} & \text{if } i,j \text{ both refer to SBFs} \\ R_{ij} & \text{if } i \text{ refers to a SBF and } j \text{ to a TBF} \\ T_{ij} & \text{if } i,j \text{ both refer to TBFs} \end{cases} \tag{22}$$

$$I_2(i,j) = (2l_2\sigma_\xi^2)^{-N/2} \exp\left[\frac{-U_{ii} - U_{jj} + (U_{ii} + U_{jj} + 2U_{ij})/2\sigma_B^2 l_2}{2\sigma_B^2}\right] \tag{23}$$

$$J_2(i,j;f) = \left(\frac{U_{if} + U_{jf}}{2l_2\sigma_B^2}\right) I_2(i,j) \tag{24}$$

$$\begin{aligned} I_4(i,j,f,g) = {} & (2l_4\sigma_\xi^2)^{-N/2} \exp\left[\frac{-U_{ii} - U_{jj} - U_{ff} - U_{gg}}{2\sigma_B^2}\right] \times \\ & \exp\left[\frac{U_{ii} + U_{jj} + U_{ff} + U_{gg} + 2(U_{ij} + U_{if} + U_{ig} + U_{jf} + U_{jg} + U_{fg})}{4l_4\sigma_B^4}\right] \end{aligned} \tag{25}$$

$$J_4(i,j,f,g;h) = \left(\frac{U_{ih} + U_{jh} + U_{fh} + U_{gh}}{2l_4\sigma_B^2}\right) I_4(i,j,f,g) \tag{26}$$

$$\begin{aligned} K_4(i,j,f,g) = {} & \left(\frac{2Nl_4\sigma_B^4 + U_{ii} + U_{jj} + U_{ff} + U_{gg}}{4l_4\sigma_B^4} + \right. \\ & \left. \frac{2(U_{ij} + U_{if} + U_{ig} + U_{jf} + U_{jg} + U_{fg})}{4l_4^2\sigma_B^4}\right) I_4(i,j,f,g) \end{aligned} \tag{27}$$

Instantaneous Variances

Defining, for brevity:

$$\overline{KIJJ}_4(i,j,f,g) = \overline{K}(i,j,f,g) + U_{if}U_{jg}\overline{I}_4(i,j) - U_{jg}\overline{J}_4(i,j,f) - U_{if}\overline{J}_4(i,j,g) \tag{28}$$

Variances:

$$\begin{aligned} \Delta Q_{bc}\Delta Q_{de} = {} & 1/\sigma_B^4 \left\{w_b w_d\, \overline{KIJJ}_4(b,d,c,e) + w_b w_e\, \overline{KIJJ}_4(b,e,c,d) + \right. \\ & \left. w_c w_d\, \overline{KIJJ}_4(c,d,b,e) + w_c w_e\, \overline{KIJJ}_4(c,e,b,d)\right\} \end{aligned} \tag{29}$$

$$\Delta Q_{bc}\Delta R_{du} = 1/\sigma_B^4 \left\{w_b w_d\, \overline{KIJJ}_4(b,d,c,u) + w_c w_d\, \overline{KIJJ}_4(c,d,b,u)\right\} \tag{30}$$

$$\Delta R_{bu}\Delta R_{cv} = 1/\sigma_B^4\, w_b w_c\, \overline{KIJJ}_4(b,c,u,v) \tag{31}$$

$$\Delta Q_{bc}\Delta w_d \;=\; 1/\sigma_B^2 \;\left\{w_b\;\left(\overline{J}_4(b,d,c) - Q_{bc}\overline{I}_4(b,d)\right) + w_c\;\left(\overline{J}_4(c,d,b) - Q_{bc}\overline{I}_4(c,d)\right)\right\} \tag{32}$$

$$\Delta R_{bu}\Delta w_d \;=\; 1/\sigma_B^2\; w_b\;\left\{\overline{J}_4(b,d,u) - R_{bu}\overline{I}_4(b,d)\right\} \tag{33}$$

$$\Delta w_b \Delta w_c \;=\; \overline{I}_4(b,c) - Q_{bc}\overline{I}_2(b)\overline{I}_2(c) \tag{34}$$

Other Quantities:

$$l_2 \;=\; \frac{2\sigma_\xi^2 + \sigma_B^2}{2\sigma_B^2\sigma_\xi^2} \tag{35}$$

$$l_4 \;=\; \frac{4\sigma_\xi^2 + \sigma_B^2}{2\sigma_B^2\sigma_\xi^2} \tag{36}$$

Acknowledgements The authors would like to thank Ansgar West, David Barber and Bernhard Schottky for useful discussions. D.S. would like to thank the Leverhulme Trust for their support (F/250/K).

References

[1] T. Watkin, A. Rau, and M. Biehl, Reviews of Modern Physics **65**, 499 (1993).

[2] D. MacKay, Neural Computation **4**, 415 (1992).

[3] D. Haussler, in *Foundations of Knowledge Acquisition: Machine Learning*, edited by A. Meyrowitz and S. Chipman (Kluwer, Boston, 1994), Chap. 9.

[4] H. Schwarze, J. Phys. A: Math. Gen. **26**, 5781 (1993).

[5] J. Freeman and D. Saad, Neural Computation **7**, 1000 (1995).

[6] J. Freeman and D. Saad, Neural Networks **9**, 1521 (1996).

[7] P. Niyogi and F. Girosi, Technical report No. 1467, AI Laboratory, Massachusetts Institute of Technology (unpublished).

[8] S. Holden and P. Rayner, IEEE Trans. on Neural Networks **6**, 368 (1995).

[9] T. Heskes and B. Kappen, Phys. Rev. A. **44**, 2718 (1991).

[10] T. Leen and G. Orr, in *Advances in Neural Information Processing Systems*, edited by J. Cowan, G. Tesauro, and J. Alspector (Morgan Kaufmann, San Mateo, CA, 1994), Vol. 6, pp. 477–484.

[11] S. Amari, Neurocomputing **5**, 185 (1993).

[12] M. Biehl and H. Schwarze, J. Phys. A: Math. Gen. **28**, 643 (1995).

[13] D. Saad and S. Solla, Phys. Rev. Lett. **74**, 4337 (1995).

[14] D. Saad and S. Solla, Phys. Rev. E. **52**, 4225 (1995).

[15] P. Riegler and M. Biehl, J. Phys. A: Math. Gen. **28**, L507 (1995).

[16] M. Biehl, P. Rieglar and C. Wohler, J. Phys. A: Math Gen. **29**, 4769 (1996).

[17] J. Freeman and D. Saad, Neural Computation, In Press, 1997.

[18] J. Kim and H. Sompolinsky, Phys. Rev. Lett. **76**, 3021 (1996).

[19] H. S. N. Barkai and H. Sompolinsky, Phys. Rev. Lett. **75**, 1415 (1995).

[20] M. Biehl and P. Riegler, Europhys. Lett. **28**, 525 (1994).

[21] O. Kinouchi and N. Caticha, J. Phys. A: Math. Gen. **26**, 6161 (1993).

[22] M. Copelli and N. Caticha, J. Phys. A: Math. Gen. **28**, 1615 (1995).

[23] M. Casdagli, Physica **35D**, 335 (1989).

[24] M. Niranjan and F. Fallside, Computer Speech and Language **4**, 275 (1990).

[25] M. Musavi *et al.*, Neural Networks **5**, 595 (1992).

[26] E. Hartman, J. Keeler, and J. Kowalski, Neural Computation **2**, 210 (1990).

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

[28] P. Riegler, (unpublished).

[29] D. Barber, D. Saad, and P. Sollich, Europhys. Lett. **34**, 151 (1996).