Online learning from finite training sets

David Barber[1]
RWCP[2] Theoretical Foundation SNN[3]
University of Nijmegen, 6525 EZ Nijmegen, The Netherlands.

Peter Sollich[4]
Department of Physics, University of Edinburgh, Edinburgh EH9 3JZ, U.K.

Abstract

We analyse online gradient descent learning from *finite* training sets at *non-infinitesimal* learning rates η for both linear and non-linear networks. In the linear case, exact results are obtained for the time-dependent generalization error of networks with a large number of weights N, trained on $p = \alpha N$ examples. This allows us to study in detail the effects of finite training set size α on, for example, the optimal choice of learning rate η. We also compare online and *offline* learning, for respective optimal settings of η at given final learning time. Online learning turns out to be much more robust to *input bias* and actually outperforms offline learning when such bias is present; for unbiased inputs, online and offline learning perform almost equally well. Our analysis of online learning for *non-linear* networks (namely, soft-committee machines), advances the theory to more realistic learning scenarios. Dynamical equations are derived for an appropriate set of order parameters; these are exact in the limiting case of either linear networks or infinite training sets. Preliminary comparisons with simulations suggest that the theory captures some effects of finite training sets, but may not yet account correctly for the presence of local minima.

1 Introduction

The analysis of online (gradient descent) learning, which is one of the most common approaches to supervised learning found in the neural networks community, has recently been the focus of much attention. The characteristic

[1]Email `davidb@mbfys.kun.nl`
[2]Real World Computing Partnership
[3]Foundation for Neural Networks
[4]Royal Society Dorothy Hodgkin Research Fellow. Email `P.Sollich@ed.ac.uk`

feature of online learning is that the weights of a network ('student') are updated each time a new training example is presented, such that the error on this example is reduced. In offline learning, on the other hand, the total error on all examples in the training set is accumulated before a gradient descent weight update is made. For a given training set and starting weights, offline learning is entirely deterministic. Online learning, on the other hand, is a stochastic process due to the random choice of training example (from the given training set) for each update; in fact, it can essentially be viewed as a 'noisy' version of offline learning. The two are equivalent only in the limit where the learning rate $\eta \to 0$ (see, *e.g.*, Heskes and Kappen, 1991). For both online and offline learning, the main quantity of interest is normally the evolution of the generalization error: After a given number of weight updates, how well does the student approximate the input-output mapping ('teacher') underlying the training examples?

Most analytical treatments of online learning assume either that the size of the training set is infinite, or that the learning rate η is vanishingly small. Both of these restrictions are undesirable: In practice, most training sets are finite[5], and non-infinitesimal values of η are needed to ensure that the learning process converges after a reasonable number of updates. General results have been derived for the difference between online and offline learning to first order in η, which apply to training sets of any size (see, *e.g.*, Heskes and Kappen, 1991). However, these do not directly address the question of generalization performance. The most explicit analysis of the time evolution of the generalization error for linear networks and finite training sets was provided by Krogh and Hertz (1992) for a scenario very similar to the (linear) one we consider below. Their $\eta \to 0$ offline calculation will serve as a baseline for our work. For non-linear networks and finite η, progress has been made in particular for so-called soft committee machine network architectures (see, *e.g.*, Saad and Solla, 1995, Biehl and Schwarze, 1995), but only for the case of infinite training sets. Finite training sets present a significant analytical difficulty as successive weight updates are correlated, giving rise to highly non-trivial generalization dynamics.

This chapter is split into two main sections. In section (2), we develop the main theoretical tools required for an exact treatment of linear networks. We then build on these results in section (3), by constructing a compact, approximate theory for non-linear networks, based on similar theoretical principles to the linear theory.

[5]Online learning can also be used to learn teacher rules that vary in time. The assumption of an infinite set (or 'stream') of training examples is then much more plausible, and in fact necessary for continued adaptation of the student. We do not consider this case in the following.

2 Linear networks

In this section, we give an exact analysis of online learning in a simple linear model system. Our aim is twofold: (1) to assess how the combination of non-infinitesimal learning rates η and finite training sets (containing α examples per weight) affects online learning, and (2) to compare the generalization performance of online and offline learning. A priori, one may expect online learning to perform worse due to its inherent randomness. We show that this disadvantage is actually negligible when online and offline learning are compared on an equal footing, *i.e.*, for their respective optimal learning rates. More importantly, we will see that online learning is much more robust to *input bias* than offline learning and actually performs *better* than the offline version in the case of biased inputs.

2.1 Model definition

We consider training of a linear student network with input-output relation

$$y = \frac{1}{\sqrt{N}} \mathbf{w}^{\mathrm{T}} \mathbf{x}$$

Here $\mathbf{x}$ is an N-dimensional vector of real-valued inputs, y the single real output and $\mathbf{w}$ the weight vector of the network. '$^{\mathrm{T}}$' denotes the transpose of a vector and the factor $1/\sqrt{N}$ is introduced for convenience. In online learning, whenever a training example $(\mathbf{x}, y)$ is presented to the network, its weight vector is updated along the gradient of the squared error[6] on this example, *i.e.*,

$$\Delta\mathbf{w} = -\eta \, \nabla_{\mathbf{w}} \frac{1}{2} \left(y - \frac{1}{\sqrt{N}} \mathbf{w}^{\mathrm{T}} \mathbf{x} \right)^2 = \eta \left(\frac{1}{\sqrt{N}} y \mathbf{x} - \frac{1}{N} \mathbf{x} \mathbf{x}^{\mathrm{T}} \mathbf{w} \right)$$

where η is the learning rate. We are primarily interested in the case of online learning from finite training sets, where for each update an example is randomly chosen from a given set $\{(\mathbf{x}^\mu, y^\mu), \mu = 1 \ldots p\}$ of p training examples. If example μ is chosen for update n, the weight vector is changed to

$$\mathbf{w}_{n+1} = \left\{ 1 - \frac{\eta}{N} \left[\mathbf{x}^\mu (\mathbf{x}^\mu)^{\mathrm{T}} + \gamma \right] \right\} \mathbf{w}_n + \eta \, \frac{1}{\sqrt{N}} y^\mu \mathbf{x}^\mu \qquad \text{(online)} \qquad (2.1)$$

Here we have also included a weight decay γ. The update rule for *offline* learning is similar, but here the gradients for all p different training examples are accumulated before a weight update is made:

$$\mathbf{w}_{(r+1)p} = [1 - \eta(\lambda + \mathbf{A})] \mathbf{w}_{rp} + \frac{\eta}{\sqrt{N}} \sum_\mu y^\mu \mathbf{x}^\mu \qquad \text{(offline)} \qquad (2.2)$$

[6] We consider only squared error here, which is probably the most commonly used error measure. We also restrict our analysis to 'vanilla' gradient descent learning, excluding more sophisticated learning algorithms.

Here r is the number of offline weight updates; in order to compare online and offline learning at equal computational cost, we index the weight vectors for both cases by the number of gradient calculations, which is $n = rp$ in the offline case. The matrix

$$\mathbf{A} = \frac{1}{N}\sum_{\mu} \mathbf{x}^{\mu}(\mathbf{x}^{\mu})^{\mathrm{T}}$$

is the correlation matrix of the training inputs, and $\lambda = \gamma\alpha$ is the weight decay rescaled by the number of examples per weight, $\alpha = p/N$. We will generally use λ (rather than γ) to characterize the strength of the weight decay, for both online and offline learning. For simplicity, all student weights are assumed to be initially zero, *i.e.*, $\mathbf{w}_{n=0} = \mathbf{0}$.

The main quantity of interest to us is the *generalization error* of the student and its evolution during learning. We assume that the training examples are generated by a linear 'teacher', *i.e.*, $y^{\mu} = \mathbf{w}_*^{\mathrm{T}}\mathbf{x}^{\mu}/\sqrt{N} + \xi^{\mu}$, where ξ^{μ} is zero mean additive noise of variance σ^2. The teacher weight vector is taken to be normalized to $\mathbf{w}_*^2 = N$ for simplicity. We first investigate the case of unbiased inputs ($\langle\mathbf{x}\rangle = \mathbf{0}$), assuming that input vectors are sampled randomly from an isotropic distribution over the hypersphere $\mathbf{x}^2 = N$ (biased inputs will be considered in Section 2.4). The generalization error, defined as the average of the squared error between student and teacher outputs for random inputs, is then

$$\epsilon_{\mathrm{g}} = \frac{1}{2N}(\mathbf{w}_n - \mathbf{w}_*)^2 = \frac{1}{2N}\mathbf{v}_n^2 \quad \text{where} \quad \mathbf{v}_n = \mathbf{w}_n - \mathbf{w}_*.$$

In order to make the scenario analytically tractable, we focus on the limit $N \to \infty$ of a large number of input components and weights, taken at constant number of examples per weight $\alpha = p/N$ and updates per weight ('learning time') $t = n/N$. In this limit, the generalization error $\epsilon_{\mathrm{g}}(t)$ becomes self-averaging (see however Section 2.4) and can be calculated by averaging both over the random selection of examples from a given training set and over all training sets. Our results can be straightforwardly extended to the case of perceptron teachers with a nonlinear transfer function, as in (Sollich, 1995).

2.2 Unbiased inputs

2.2.1 Outline of calculation

We begin by deriving from the online learning weight update (2.1) an update equation for the 'selection' average of the generalization error (*i.e.*, its average with respect to the random choice of training examples for each update, denoted generically by $\langle \ldots \rangle$). In fact, it will turn out to be useful to consider a slightly generalized version of the generalization error, $\epsilon_n = \frac{1}{2N}\mathbf{v}_n^{\mathrm{T}}\mathbf{M}\mathbf{v}_n$, with $\mathbf{M}$ an arbitrary $N \times N$ matrix. To get the update equation for $\langle\epsilon_n\rangle$, we first rewrite (2.1) in terms of $\mathbf{v}_n$, the difference between student and teacher weight vectors:

$$\mathbf{v}_{n+1} = \left\{1 - \eta\left[\frac{1}{N}\mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}} + \frac{\lambda}{p}\right]\right\}\mathbf{v}_n + \eta\,\frac{1}{\sqrt{N}}\xi^\mu\mathbf{x}^\mu - \frac{\eta\lambda}{p}\mathbf{w}_* \tag{2.3}$$

This can now be multiplied by its transpose, with the matrix $\mathbf{M}$ inserted, and the selection average for update n performed. Discarding terms which become negligible in the large N limit, one finds after a little algebra

$$\begin{aligned} N\left(\langle\epsilon_{n+1}\rangle - \langle\epsilon_n\rangle\right) &= \frac{\tilde\eta}{N}(\mathbf{b}-\lambda\mathbf{w}_*)^{\mathrm{T}}\mathbf{M}\left\langle\mathbf{v}_n\right\rangle - \frac{\tilde\eta}{N}\left\langle\mathbf{v}_n^{\mathrm{T}}\left[\lambda\mathbf{M} + \frac{1}{2}(\mathbf{AM}+\mathbf{MA})\right]\mathbf{v}_n\right\rangle \\ &+ \frac{\tilde\eta^2\alpha}{N}\sum_\mu\frac{1}{N}(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{M}\mathbf{x}^\mu\left\{\frac{1}{2}(\xi^\mu)^2 - \xi^\mu\frac{1}{\sqrt{N}}(\mathbf{x}^\mu)^{\mathrm{T}}\left\langle\mathbf{v}_n\right\rangle + \frac{1}{2N}\left\langle\mathbf{v}_n^{\mathrm{T}}\mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{v}_n\right\rangle\right\} \end{aligned} \tag{2.4}$$

where $\tilde\eta = \eta/\alpha$ is a rescaled learning rate, and $\mathbf{b} = \sum_\mu \xi^\mu\mathbf{x}^\mu/\sqrt{N}$. We now want to transform (2.4) into a closed dynamical equation for $\langle\epsilon_n\rangle$. This means that all selection averages need to be either eliminated or reduced to averages of the same form as $\langle\epsilon_n\rangle$. For the two terms linear in $\langle\mathbf{v}_n\rangle$, this is straightforward: The selection average of (2.1) yields directly

$$N\left(\langle\mathbf{v}_{n+1}\rangle - \langle\mathbf{v}_n\rangle\right) = \tilde\eta\left[-(\lambda+\mathbf{A})\left\langle\mathbf{v}_n\right\rangle + \mathbf{b} - \lambda\mathbf{w}_*\right].$$

Starting from $\mathbf{v}_0 = -\mathbf{w}_*$, this can easily be solved, with the result (for $N \to \infty$)

$$\langle\mathbf{v}_n\rangle = (\lambda+\mathbf{A})^{-1}\left\{\mathbf{b} - \lambda\mathbf{w}_* - \exp\left[-\tilde\eta\,t(\lambda+\mathbf{A})\right](\mathbf{b}+\mathbf{A}\mathbf{w}_*)\right\} \tag{2.5}$$

from which the selection average has now disappeared. Learning rate and learning time enter only through the combination $\tau = \tilde\eta\,t$; this rescaled time will be useful later on. In (2.4), the remaining terms quadratic in $\mathbf{v}_n$ now present the main problem. The second term on the r.h.s. shows that the evolution of $\epsilon_{\mathrm{g}} = \epsilon_n(\mathbf{M}{=}\mathbf{1})$ depends on $\epsilon_n(\mathbf{M}{=}\mathbf{A})$ which in turn depends on $\epsilon_n(\mathbf{M}{=}\mathbf{A}^2)$ and so on, yielding an infinite hierarchy of order parameters. This problem was solved in (Sollich and Barber, 1997a) by introducing an auxiliary parameter h through $\mathbf{M} = \exp(h\mathbf{A})$; all order parameters $\epsilon_n(\mathbf{M}{=}\mathbf{A}^m)$, $m = 1, 2, \ldots$, can then be obtained by differentiating $\epsilon_n(h) = \frac{1}{2N}\mathbf{v}_n^{\mathrm{T}}\exp(h\mathbf{A})\mathbf{v}_n$.

Here we choose a different route, which is somewhat more transparent and also more easily adapted to the case of biased inputs to be considered later. The main idea is to decompose the evolution of $\mathbf{v}_n$ into components defined by eigenvectors of the input correlation matrix $\mathbf{A}$. (This is equivalent to changing to a coordinate system in which $\mathbf{A}$ is diagonal, and then considering the components of $\mathbf{v}_n$ separately.) More precisely, let us order the N eigenvalues of $\mathbf{A}$ in ascending order and split them into K equal blocks, labeled by $\kappa = 1 \ldots K$, each containing N/K eigenvalues. Let $\mathbf{P}^\kappa$ be the projector matrices onto the spaces spanned by the eigenvectors of each block. Then $\mathbf{v}_n = \sum_\kappa \mathbf{P}^\kappa \mathbf{v}_n$; likewise, the generalization error is decomposed as

$$\epsilon_{\rm g} = \frac{1}{K}\sum_\kappa \epsilon_n^\kappa, \qquad \epsilon_n^\kappa = \frac{K}{2N}\mathbf{v}_n^{\rm T}\mathbf{P}^\kappa\mathbf{v}_n$$

Each of the generalization error components ϵ_n^κ obeys the update equation (2.4), with $\mathbf{M} = K\mathbf{P}^\kappa$. But these equations now become closed, because

$$\mathbf{A}\mathbf{P}^\kappa = \mathbf{P}^\kappa\mathbf{A} \approx a^\kappa \mathbf{P}^\kappa$$

where a^κ is an eigenvalue from the κ-th block (formally, this approximation becomes exact in the limit $K \to \infty$, where the spread of eigenvalues within each block tends to zero). This immediately reduces the second term on the right-hand side of (2.4) to $-2\tilde{\eta}(\lambda + a^\kappa)\left\langle \epsilon_n^\kappa \right\rangle$. Only the very last term of (2.4) now remains to be brought into a similar form. This is achieved by noting that the factors $c_\kappa^\mu = (K/N)(\mathbf{x}^\mu)^{\rm T}\mathbf{P}^\kappa\mathbf{x}^\mu$ are 'within-sample self-averaging' (Sollich and Barber, 1997a): Up to fluctuations which vanish as $O(N^{-1/2})$ for large N, all c^μ are equal to each other and hence to the training set ('sample') average

$$\frac{1}{p}\sum_\mu c_\kappa^\mu = \frac{K}{\alpha N}{\rm tr}\,\mathbf{A}\mathbf{P}^\kappa \approx \frac{a^\kappa}{\alpha}$$

The last approximation again becomes exact[7] for $K \to \infty$. The factors $c_\kappa^\mu = a^\kappa/\alpha$ can therefore be taken out of the sum over μ in (2.4), leaving the selection average

$$\sum_\mu \frac{1}{2N}\left\langle \mathbf{v}_n^{\rm T}\mathbf{x}^\mu(\mathbf{x}^\mu)^{\rm T}\mathbf{v}_n \right\rangle = \frac{1}{2}\left\langle \mathbf{v}_n^{\rm T}\mathbf{A}\mathbf{v}_n \right\rangle \approx \frac{1}{K}\sum_\kappa a^\kappa \left\langle \epsilon_n^\kappa \right\rangle$$

We now have all the ingredients to write (2.4) as a closed system of evolution equations for the ϵ_n^κ. In the large N limit, the change $N\left(\left\langle \epsilon_{n+1}^\kappa \right\rangle - \left\langle \epsilon_n^\kappa \right\rangle\right)$ due to an update becomes the time derivative $\partial_t \epsilon^\kappa$, and $\left\langle \epsilon_n^\kappa \right\rangle \to \epsilon^\kappa(t)$. Using the rescaled time $\tau = \tilde{\eta}t$ introduced above, one then has

$$\left[\partial_\tau + 2(\lambda + a^\kappa)\right]\epsilon^\kappa(\tau) = V^\kappa(\tau) + \tilde{\eta}\,W^\kappa(\tau) + \tilde{\eta}\,a^\kappa \frac{1}{K}\sum_{\kappa'} a^{\kappa'}\epsilon^{\kappa'}(\tau) \qquad (2.6)$$

Here the functions $V^\kappa(\tau)$ and $W^\kappa(\tau)$ are

$$\begin{aligned} V^\kappa &= \frac{K}{N}(\mathbf{b} - \lambda\mathbf{w}_*)^{\rm T}\mathbf{P}^\kappa \left\langle \mathbf{v}_n \right\rangle \\ W^\kappa &= a^\kappa \left[\frac{1}{2N}\sum_\mu (\xi^\mu)^2 - \frac{1}{N}\mathbf{b}^{\rm T}\left\langle \mathbf{v}_n \right\rangle\right] \end{aligned}$$

[7]The large K limit needs to be taken *after* the limit $N \to \infty$ for 'within-sample self-averaging' to hold; this is why one cannot take $K = N$ from the outset.

with $\langle \mathbf{v}_n \rangle$ given by (2.5). Having derived (2.6), the rest of the calculation is fairly straightforward. Eq. (2.6) is formally solved using Laplace transforms with respect to τ, for example $\hat{\epsilon}^\kappa(z) = \int_0^\infty d\tau \, \exp(-z\tau)\, \epsilon^\kappa(\tau)$:

$$\hat{\epsilon}^\kappa(z) = \frac{1}{z + 2(\lambda + a^\kappa)} \left[\epsilon^\kappa(0) + \hat{V}^\kappa(z) + \tilde{\eta}\, \hat{W}^\kappa(z) + \tilde{\eta}\, a^\kappa \frac{1}{K} \sum_{\kappa'} a^{\kappa'} \hat{\epsilon}^{\kappa'}(z) \right] \tag{2.7}$$

with the initial condition $\epsilon^\kappa(0) = \frac{K}{2N} \mathbf{w}_*^{\mathrm{T}} \mathbf{P}^\kappa \mathbf{w}_*$. Multiplying by a^κ and summing over κ gives a self-consistency equation for $K^{-1} \sum_\kappa a^\kappa \hat{\epsilon}^\kappa(z)$ which is easily solved. Inserting the solution into (2.7) then gives an explicit expression for $\hat{\epsilon}^\kappa(z)$ and hence for the Laplace transform of the generalization error, $\hat{\epsilon}_{\mathrm{g}}(z) = K^{-1} \sum_\kappa \hat{\epsilon}^\kappa(z)$. As a final step, the average over all training sets (*i.e.*, training inputs $\mathbf{x}^\mu$ and output noises ξ^μ) is then carried out. In the end, everything can be written in terms of averages over the known eigenvalue spectrum (Hertz et al., 1989; Sollich, 1994) of the input correlation matrix $\mathbf{A}$. The explicit form of the final result (Sollich and Barber, 1997a) is rather cumbersome; we omit it here and note only the relatively simple dependence on η:

$$\hat{\epsilon}_{\mathrm{g}}(z) = \hat{\epsilon}_0(z) + \frac{\eta \hat{\epsilon}_1(z)}{1 - \eta \hat{\epsilon}_2(z)} \tag{2.8}$$

The functions $\epsilon_i(z)$ $(i = 0 \ldots 2)$ depend on α, σ^2 and λ (and, of course, z), but are independent of η. The teacher weights do no appear explicitly: because of the isotropy of the input distribution, only the length of the teacher weight vector matters once an average over training sets has been taken, and this has already been fixed to $\mathbf{w}_*^2 = N$.

The calculation of the generalization error for offline learning is much simpler than that for the online case due to the absence of the selection average. In fact, the offline weight update (2.2) can be iterated directly to yield

$$\mathbf{v}_{rp} = (\lambda + \mathbf{A})^{-1} \left\{ \mathbf{b} - \lambda \mathbf{w}_* - [1 - \eta(\lambda + \mathbf{A})]^r \left(\mathbf{b} + \mathbf{A}\mathbf{w}_* \right) \right\} \tag{2.9}$$

Multiplying this by its transpose gives directly the generalization error, and the average over training sets can then be carried out in the usual fashion (see, *e.g.*, Hertz et al., 1989). As expected on general grounds, for $\eta \to 0$ (and only then) one obtains the same result as for online learning, corresponding to the term $\hat{\epsilon}_0(z)$ in (2.8).

2.3 Discussion

We now briefly highlight some features of our exact result (2.8) for the generalization error achieved by online learning; a somewhat more detailed exposition can be found in (Sollich and Barber, 1997b). We discuss the asymptotic generalization error ϵ_∞, the convergence speed for large learning times, and the behaviour at small t; finally, we compare online and offline learning. For

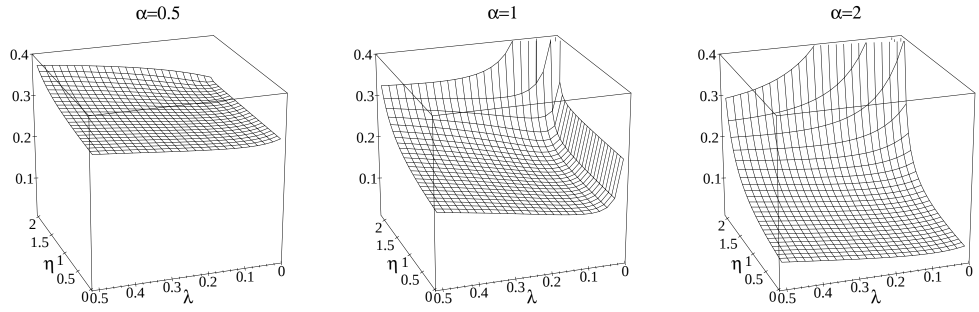

Figure 1: Asymptotic generalization error ϵ_∞ vs η and λ. α as shown, $\sigma^2 = 0.1$.

numerical evaluations, we generally take $\sigma^2 = 0.1$, corresponding to a sizable noise-to-signal ratio of $\sqrt{0.1} \approx 0.32$.

The asymptotic generalization error is found directly from (2.8) using $\epsilon_\infty = \epsilon_{\rm g}(t \to \infty) = \lim_{z\to 0} z\hat{\epsilon}_{\rm g}(z)$. As expected, it coincides with the offline result (which is *independent* of η) *only* for $\eta = 0$; as η increases from zero, it increases monotonically. Reassuringly, our calculation reproduces existing $O(\eta)$ results for this increase (Heskes and Kappen, 1991). In figure 1 we plot ϵ_∞ as a function of η and λ for $\alpha = 0.5$, 1, 2. We observe that it is minimal for $\lambda = \sigma^2$ and $\eta = 0$, as expected from corresponding results for offline learning (Krogh and Hertz, 1992)[8]. We also read off that for fixed λ, ϵ_∞ is an increasing function of η: The larger η, the more the weight updates tend to overshoot the minimum of the (total, *i.e.*, offline) training error. This causes a diffusive motion of the weights around their average asymptotic values (Heskes and Kappen, 1991) which increases ϵ_∞. In the absence of weight decay ($\lambda = 0$) and for $\alpha < 1$, however, ϵ_∞ is independent of η. In this case the training data can be fitted perfectly; every term in the total sum-of-squares training error is then zero and online learning does not lead to weight diffusion because all individual updates vanish. In general, the relative increase $\epsilon_\infty(\eta)/\epsilon_\infty(\eta = 0) - 1$ due to nonzero η depends significantly on α. For $\eta = 1$ and $\alpha = 0.5$, for example, this increase is smaller than 6% for all λ (at $\sigma^2 = 0.1$), and for $\alpha = 1$ it is at most 13%. This means that in cases where training data is limited ($p \approx N$), η can be chosen fairly large in order to optimize learning speed, without seriously affecting the asymptotic generalization error. In the large α limit, on the other hand, one finds $\epsilon_\infty = (\sigma^2/2)[1/\alpha + \eta/(2-\eta)]$. The relative increase over the value at $\eta = 0$ therefore grows linearly with α; already for $\alpha = 2$, increases of around 50% can occur for $\eta = 1$.

Fig. 1 also shows that ϵ_∞ diverges as η approaches a critical learning rate

[8]The optimal value of the *unscaled* weight decay decreases with α as $\gamma = \sigma^2/\alpha$, because for large training sets there is less need to counteract noise in the training data by using a large weight decay.

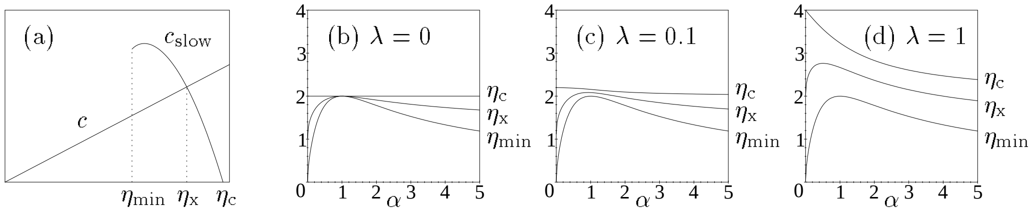

Figure 2: Sketch of definitions of η_{min} (minimal learning rate for slow mode), η_{x} (crossover to slow mode dominated convergence) and η_{c} (maximal ('critical') learning rate at which convergence still occurs), and their dependence on α.

η_{c}: As $\eta \to \eta_{\text{c}}$, the 'overshoot' of the weight update steps becomes so large that the weights eventually diverge. From the Laplace transform (2.8), one finds that η_{c} is determined by $\eta_{\text{c}}\hat{\epsilon}_2(z = 0) = 1$; it is a function of α and λ only. As shown in figure 2b-d, η_{c} increases with λ. This is reasonable, as the weight decay reduces the length of the weight vector at each update, counteracting potential weight divergences. In the small and large α limits one has $\eta_{\text{c}} = 2(1 + \lambda)$ and $\eta_{\text{c}} = 2(1 + \lambda/\alpha)$, respectively. For constant λ, η_{c} therefore decreases[9] with α (figure 2b-d).

We now turn to the large t behaviour of the generalization error $\epsilon_{\text{g}}(t)$. For small η, the most slowly decaying contribution to $\epsilon_{\text{g}}(t)$—the slowest 'mode'—varies as $\exp(-ct)$, its decay constant $c = \eta[\lambda + (\sqrt{\alpha} - 1)^2]/\alpha$ scaling linearly with η, the size of the weight updates, as expected (figure 2a). For larger η, the picture changes due to a new slow mode arising from the denominator of (2.8). Interestingly, this mode exists only for η above a finite threshold $\eta_{\text{min}} = 2/(\alpha^{1/2} + \alpha^{-1/2} - 1)$. For finite α, it could therefore not have been predicted from a small η expansion of $\epsilon_{\text{g}}(t)$. Its decay constant c_{slow} decreases to zero as $\eta \to \eta_{\text{c}}$, and crosses that of the normal mode at $\eta_{\text{x}}(\alpha, \lambda)$ (figure 2a). For $\eta > \eta_{\text{x}}$, the slow mode therefore determines the convergence speed for large t, and fastest convergence is obtained for $\eta = \eta_{\text{x}}$. However, it may still be advantageous to use lower values of η in order to lower the asymptotic generalization error (see below); values of $\eta > \eta_{\text{x}}$ would deteriorate both convergence speed and asymptotic performance. Fig. 2b-d shows the dependence of η_{min}, η_{x} and η_{c} on α and λ. For λ not too large, η_{x} has a maximum at $\alpha \approx 1$ (where $\eta_{\text{x}} \approx \eta_{\text{c}}$), while decaying to $\eta_{\text{x}} \approx \frac{1}{2}\eta_{\text{c}}$ for larger α. This can be explained in terms of the anisotropy of the total training error surface (Sollich and Barber, 1997a), which is strongest for $\alpha = 1$ and $\lambda \to 0$.

Consider now the small t behaviour of $\epsilon_{\text{g}}(t)$. Fig. 3 illustrates the dependence of $\epsilon_{\text{g}}(t)$ on η; comparison with simulation results for $N = 50$ clearly confirms our calculations and demonstrates that finite N effects are not sig-

[9] Conversely, for constant γ, η_{c} *increases* with α from $2(1 + \gamma\alpha)$ to $2(1 + \gamma)$: For large α, the weight decay is applied more often between repeat presentations of a training example that would otherwise cause the weights to diverge.

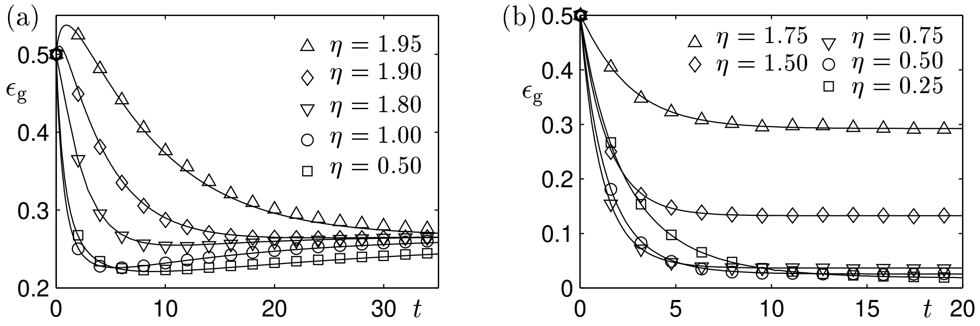

Figure 3: ϵ_g vs t for different η. Simulations for $N = 50$ are shown by symbols (standard errors less than symbol sizes). λ=10^{-4}, σ^2=0.1. (a) α=0.7, (b) α=5

nificant even for such fairly small N. For $\alpha = 0.7$ (figure 3a), we see that nonzero η acts as effective update noise, eliminating the minimum in $\epsilon_g(t)$ which corresponds to over-training (Krogh and Hertz, 1992). ϵ_∞ is also seen to be essentially independent of η as predicted for the small value of $\lambda = 10^{-4}$ chosen. For $\alpha = 5$, figure 3b clearly shows the increase of ϵ_∞ with η. It also illustrates how convergence first speeds up as η is increased from zero and then slows down again as $\eta_c \approx 2$ is approached.

Above, we saw that the *asymptotic* generalization error ϵ_∞ is minimal for $\eta = 0$. Fig. 4 shows what happens if we minimize $\epsilon_g(t)$ instead for a given *final learning time* t, corresponding to a fixed amount of computational effort for training the network. As t increases, the optimal η decreases towards zero as required by the tradeoff between asymptotic performance and convergence speed. For large t, the functional form of this decay is $\eta_{\rm opt} = (a + b \ln t)/t$ with t-independent coefficients a and b (Sollich and Barber, 1997a).

We now compare the performance of online learning to that of offline learning as calculated from (2.9). (The number of gradient calculations required for r offline weight updates is $n = rp$, corresponding to a learning time $t = n/N = r\alpha$; the generalization error $\epsilon_g(t)$ is therefore only defined for learning times t which are integer multiples of α.) To compare online and offline learning on an equal footing, we again consider optimized values of η for given final learning time t. Fig. 4b shows that the performance loss from using online instead of offline learning is actually negligible. This may seem surprising given the stochasticity of weight updates in online learning, in particular for small t. However, figure 4a shows that online learning can make up for this by allowing larger values of η to be used.

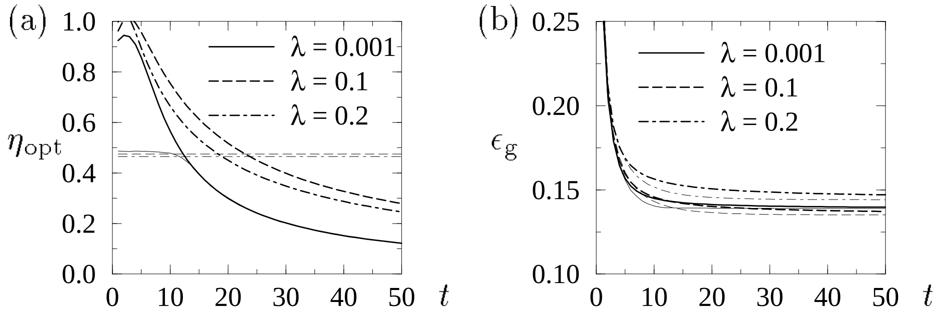

Figure 4: (a) Optimal learning rate η vs. final learning time t for online (bold) and offline learning (thin lines), and (b) resulting generalization error $\epsilon_{\rm g}$. $\alpha = 1$, $\sigma^2 = 0.1$, λ as shown. Note that although we plot offline results as continuous lines to avoid visual clutter, they are actually defined only at discrete values of the learning time, $t = r\alpha$, with r the number of offline weight updates.

2.4 Biased inputs

2.4.1 Modifications to calculation

We now investigate how online and offline learning are affected by input bias $\langle \mathbf{x} \rangle = \bar{\mathbf{x}} \neq \mathbf{0}$. As a simple scenario of this kind, consider the case where the *deviations* $\Delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$ of the inputs from their average are still distributed isotropically over a hypersphere. We choose the radius R of this hypersphere such that the average value of $\mathbf{x}^2$ is the same (N) as for the unbiased case, *i.e.*, $R^2 = N(1 - m^2)$ where $m^2 = \bar{\mathbf{x}}^2/N$ measures the size of the bias. The generalization error (the squared deviation between student and teacher outputs averaged over all inputs) now has two components,

$$\epsilon_{\rm g} = \frac{1}{2N}\left[(\bar{\mathbf{x}}^{\rm T}\mathbf{v}_n)^2 + (1 - m^2)\mathbf{v}_n^2\right] \tag{2.10}$$

As before, we consider a teacher with weight vector of length $\mathbf{w}_*^2 = N$. In the presence of input bias, however, we also need to specify the average teacher output $\bar{y} = \bar{\mathbf{x}}^{\rm T}\mathbf{w}_*/\sqrt{N}$. This parameter is not constrained by our other assumptions; however, to limit the number of free parameters in the model, we choose it to have its typical root-mean-squared value when the directions of $\mathbf{w}_*$ and $\bar{\mathbf{x}}$ are uncorrelated: $\bar{y}^2 = m^2$.

As for the case of unbiased inputs, the evolution of the generalization error is largely determined by the eigenvalue spectrum of the input correlation matrix $\mathbf{A}$. This has been determined by a number of authors (LeCun et al., 1991; Wendemuth et al., 1993; Halkjær and Winther, 1997) and shows the following features: There is a 'normal' part of the spectrum, with eigenvalues which tend to finite values as $N \to \infty$; the eigenvalues in this part of the spectrum are identical to those for the unbiased input case, expect for a rescaling by the factor $(1-m^2)$. Additionally, however, there is one isolated eigenvalue $a_N = N\alpha m^2$ which is proportional to N and exists *only* in the presence of input bias. Intuitively, this corresponds to the fact that the component of the student weights along the direction of $\bar{\mathbf{x}}$ is much more strongly determined by the training data because all input vectors have a component along $\bar{\mathbf{x}}$. Not surprisingly, therefore, the eigenvector corresponding to a_N is along the direction[10] of $\bar{\mathbf{x}}$.

We can see immediately that input bias has a drastic effect on offline learning by considering eq. (2.9): For the offline learning process to converge, the product of η and the largest eigenvalue of $\lambda + \mathbf{A}$ must be less than two. In the presence of input bias, this gives the condition $\eta < 2/(N\alpha m^2)$ (neglecting λ, which gives a negligible correction for $N \to \infty$). The maximal learning rate is therefore *drastically reduced* from order unity to $O(N^{-1})$. A little reflection then shows that only the first contribution of the generalization error (2.10) decays for finite learning times; carrying out the average over training sets, one finds

$$\epsilon_{\mathrm{g}}(t = r\alpha) = \frac{1}{2} m^2 (1 - N\eta\alpha m^2)^{2r} + \frac{1}{2}(1 - m^2) \qquad (2.11)$$

The second contribution would only decay for learning times of $O(N)$, which are inaccessibly long in the limit $N \to \infty$ that we consider.

Online learning, on the other hand, is not plagued by the same problem, as we now show. Consider the first contribution to the generalization error, which we write as $\epsilon_{\mathrm{g},1} = \frac{1}{2}\delta_n^2$ with

$$\delta_n = \frac{1}{\sqrt{N}} \bar{\mathbf{x}}^{\mathrm{T}} \mathbf{v}_n$$

From the update equation (2.3) one derives that

$$\delta_{n+1} = (1 - \eta m^2)\delta_n + \eta\, \xi^\mu m^2 \qquad (2.12)$$

up to correction terms which vanish for $N \to \infty$. Starting from the initial value $\delta_0 = -\bar{y}$, this can easily be iterated and the selection average carried out to give

$$\left\langle \delta_n^2 \right\rangle = \bar{y}^2 (1 - \eta m^2)^{2n} + \eta^2 m^4 \frac{1 - (1 - \eta m^2)^{2n}}{1 - (1 - \eta m^2)^2} \frac{1}{p} \sum_\mu (\xi^\mu)^2$$

up to $O(N^{-1})$ corrections; an average over training sets then gives $p^{-1} \sum_\mu (\xi^\mu)^2 \to \sigma^2$. For $n = t = 0$, only the first term is nonzero. On the other hand, for

[10] In fact there is a small angle between this eigenvector and $\bar{\mathbf{x}}$, which however decreases as $O((\alpha N)^{-1/2})$ as N grows large. LeCun et al. (1991) claimed that this angle is exactly zero; however, their argument cannot be quite correct as it would also entail that $\mathbf{A}$ has only two different eigenvalues (whereas in reality it has a continuous spread of eigenvalues for any finite α).

nonzero learning time t (and values of the learning rate such that convergence occurs, *i.e.*, $0 < \eta < 2/m^2$) only the second term survives because $n = Nt \to \infty$ for $N \to \infty$. We therefore have for the average value of the first contribution to the generalization error:

$$\langle \epsilon_{\mathrm{g},1}(t=0) \rangle = \bar{y}^2 = m^2, \qquad \langle \epsilon_{\mathrm{g},1}(t>0) \rangle = \frac{1}{2}\sigma^2 \frac{\eta m^2}{2 - \eta m^2} \tag{2.13}$$

The discontinuous change at $t = 0$ reflects the fact that $\langle \delta_n^2 \rangle$ changes from its initial to its asymptotic value after a number of updates n which does not increase with system size N. [11]

We still have to calculate the evolution of the second component $\epsilon_{\mathrm{g},2} = (1 - m^2)\mathbf{v}_n^2/(2N)$ of the generalization error (2.10) for the case of online learning. At first sight, the $O(N)$ eigenvalue of $\mathbf{A}$ appears to complicate this task. However, the component of $\mathbf{v}_n$ along $\bar{\mathbf{x}}$, the corresponding eigenvector, contributes only negligibly to $\epsilon_{\mathrm{g},2}$:

$$\frac{1}{2N}\left(\frac{1}{|\bar{\mathbf{x}}|}\bar{\mathbf{x}}^{\mathrm{T}}\mathbf{v}_n\right)^2 = \frac{1}{Nm^2}\epsilon_{\mathrm{g},1} = O(N^{-1})$$

Thus only components of $\mathbf{v}_n$ along directions corresponding to the $O(1)$ eigenvalues of $\mathbf{A}$ need to be considered; their evolution can be calculated exactly as in Section 2.2. The only change is the rescaled eigenvalue spectrum of $\mathbf{A}$; in fact, one finds that $\epsilon_{\mathrm{g},2}/(1 - m^2)$ is exactly the same as $\epsilon_{\mathrm{g}} = \mathbf{v}_n^2/2N$ for *unbiased* inputs of length $\mathbf{x}^2 = N(1-m^2)$. It is easily checked that this change of effective input vector length can be effected by replacing λ, σ^2 and η in the expressions for ϵ_{g} by the rescaled values $\lambda' = \lambda/(1-m^2)$, $(\sigma')^2 = \sigma^2/(1-m^2)$ and $\eta' = \eta(1 - m^2)$, and so no new calculations need to be carried out.

2.4.2 Discussion

We have already mentioned that the critical learning rate for offline learning is drastically reduced to $\eta_{\mathrm{c}} = 2/N\alpha m^2$ by the presence of input bias. For online learning, η_{c} is affected in two ways: first through the 'rescaling' of η and λ explained above for the calculation of $\epsilon_{\mathrm{g},2}$, and secondly through the presence of the term $\epsilon_{\mathrm{g},1}$; eq. (2.13) shows that for the latter to remain finite one requires $\eta_{\mathrm{c}} < 2/m^2$. Fig. 5 illustrates the resulting variation of η_{c} with m^2 for several values of α and λ: As the bias increases from 0, the critical learning rate first increases until it reaches the value $2/m^2$; from that point onwards, it follows the curve $\eta_{\mathrm{c}} = 2/m^2$ (independently of α and λ) until it reaches $\eta_{\mathrm{c}} = 2$ at[12] $m^2 = 1$. In marked contrast to the case of offline learning,

[11] Note also that we have written the selection average in (2.13) explicitly because $\epsilon_{\mathrm{g},1}$ is no longer self-averaging: Each weight update (2.12) causes a change in δ_n and $\epsilon_{\mathrm{g},1}$ of order unity, and hence the fluctuations of $\epsilon_{\mathrm{g},1}$ remain nonzero even for $N \to \infty$.

[12] This is the maximal bias in our scenario since $\langle \mathbf{x}^2 \rangle = N > \bar{\mathbf{x}}^2 = Nm^2$.

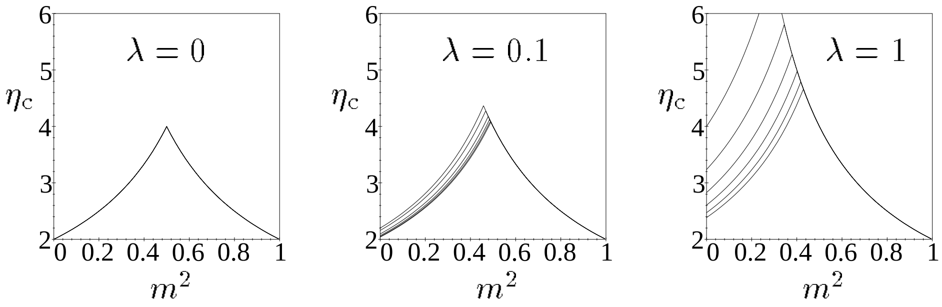

Figure 5: Critical learning rate η_c for online learning vs input bias m^2, for weight decay λ as shown and training set size $\alpha = 0, 1, \ldots, 5$ (bottom to top). Compare fig. 2 for the case of unbiased inputs.

the critical learning rate η_c for online learning therefore never decreases below values of order unity, and can actually be increased by the presence of input bias.

The different effects of input bias on the critical learning rates of online and offline learning are also reflected in the generalization performance for optimal values of η at given final learning time. For offline learning, eq. (2.11) shows that the optimal $\eta = 1/(N\alpha m^2)$, whatever the (integer) value of $r = t/\alpha$. This reduces the first contribution to the offline generalization error to zero for any $r \geq 1$, but still leaves a nonzero term $\epsilon_g = (1 - m^2)/2$ (which as explained above would start to decay only for extremely long learning times $t = O(N)$).

For online learning, on the other hand, the optimal learning rate remains of order one even in the presence of input bias. This was to be expected from the analogous results for the critical learning rate, and can be seen explicitly in fig. 6(a). Fig. 6(b) shows the resulting generalization error, which is seen to *decrease* as the input bias increases. Online learning therefore successfully exploits the presence of the input bias to achieve better generalization performance[13]. This contrasts markedly with the case of offline learning, where generalization performance (at finite learning times t) deteriorates as soon as an input bias is present[14].

[13]Wendemuth et al. (1993) view the input bias as 'additional information' which leads to improved generalization. In our case, the same conclusion can be arrived at by considering the extreme limit of maximal bias, $m^2 = 1$: In this case, the distribution of input vectors collapses to the point $\mathbf{x} = \bar{\mathbf{x}}$, and so perfect generalization is obtained after only one training example has been presented. (For noisy training outputs, more examples would be needed; the generalization error then decays roughly as $\epsilon_g \sim \sigma^2/n$, which however still gives perfect generalization $\epsilon_g = 0$ for any finite learning time t.)

[14]For biased inputs, we found an offline generalization error of $\epsilon_g = (1 - m^2)/2$ for optimally chosen η, which is arbitrarily close to $\frac{1}{2}$ for m^2 sufficiently small. For unbiased

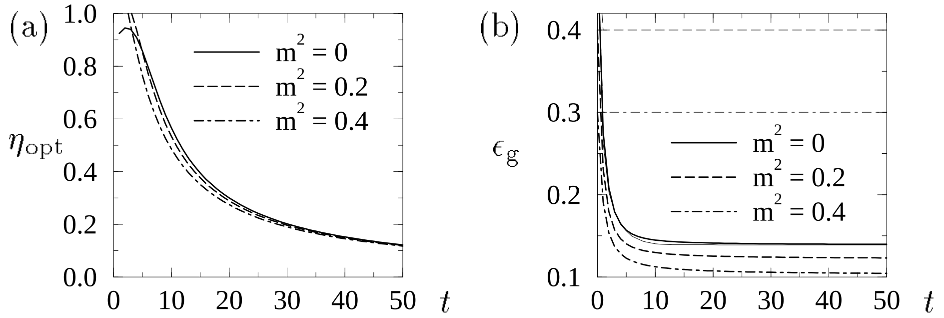

Figure 6: (a) Optimal learning rate η vs. final learning time t for online learning in the presence of input bias m^2 (values as shown; $\alpha = 1$, $\sigma^2 = 0.1$, $\lambda = 0.001$). (b) Resulting generalization error ϵ_{g}, with results for offline learning shown for comparison (thin lines). Note that while offline learning performs (marginally) better than online learning for unbiased inputs ($m^2 = 0$), it is far worse as soon as the input bias is nonzero.

2.5 Conclusions for the linear theory

We have obtained exact results for the generalization error achieved by online learning from finite training sets at non-infinitesimal learning rates. These apply directly only to the simple linear model that we have considered, but also exhibit generic features which we expect to be of general relevance. For example, the calculated dependence on η of the asymptotic generalization error ϵ_∞ and the convergence speed shows that, in general, sizable values of η can be used for training sets of limited size ($\alpha \approx 1$), while for larger α it is important to keep learning rates small. More important from a practical point of view is probably the explicit comparison between online and offline learning that our results allow us to make. To make this comparison fair, we considered the generalization performance of both algorithms for the respective optimal values of the learning rate at a given final learning time t. For unbiased inputs, we found in this way that online learning performs only marginally worse than offline learning, whereas it is in fact vastly superior as soon as there is any kind of input bias. This suggests strongly that online learning should generally be preferred over offline learning in problems where biased inputs cannot be a priori excluded.

---

inputs, on the other hand, ϵ_{g} for optimal η is generally significantly smaller than a half, as illustrated by fig. 4, for example—it can never be greater than $\frac{1}{2}$ since otherwise $\eta = 0$ would give a lower ϵ_{g}.

3 Non-linear Networks

For linear networks, we saw that the difficulties encountered with finite training sets and non-infinitesimal learning rates can be overcome by extending the standard set of descriptive ('order') parameters to include the effects of weight update correlations (Sollich and Barber, 1997b). In this section, we extend our analysis to *nonlinear* networks. The particular model we choose to study is the soft-committee machine, which is capable of representing a rich variety of input-output mappings. Its online learning dynamics has been studied comprehensively for infinite training sets(Biehl and Schwarze, 1995; Saad and Solla, 1995). In order to carry out our analysis, we adapt tools originally developed in the statistical mechanics literature which have found application, for example, in the study of Hopfield network dynamics (Coolen et al., 1996).

3.1 Model and Outline of Calculation

For an N-dimensional input vector $\mathbf{x}$, the output of the soft committee machine is given by

$$y = \sum_{l=1}^{L} g\left(\frac{1}{\sqrt{N}}\mathbf{w}_l^{\mathrm{T}}\mathbf{x}\right) \tag{3.1}$$

where the nonlinear activation function $g(h_l) = \mathrm{erf}(h_l/\sqrt{2})$ acts on the activations $h_l = \mathbf{w}_l^{\mathrm{T}}\mathbf{x}/\sqrt{N}$ (the factor $1/\sqrt{N}$ is for convenience only). This is a neural network with L hidden units, input to hidden weight vectors $\mathbf{w}_l$, $l = 1..L$, and all hidden to output weights set to 1.

We remind the reader that in online learning, the student weights are adapted on a sequence of presented examples to better approximate the teacher mapping. The training examples are drawn, with replacement, from a finite set, $\{(\mathbf{x}^\mu, y^\mu), \mu = 1..p\}$. This set remains fixed during training. Its size relative to the input dimension is denoted by $\alpha = p/N$. We take the input vectors $\mathbf{x}^\mu$ as samples from an N dimensional Gaussian distribution with zero mean and unit variance. The training outputs y^μ are assumed to be generated by a teacher soft committee machine with hidden weight vectors $\mathbf{w}_m^*$, $m = 1..M$, with additive Gaussian noise corrupting its activations and output.

The discrepancy between the teacher and student on a particular training example $(\mathbf{x}, y)$, drawn from the training set, is given by the squared difference of their corresponding outputs,

$$E = \frac{1}{2}\left[\sum_l g(h_l) - y\right]^2 = \frac{1}{2}\left[\sum_l g(h_l) - \sum_m g(k_m + \xi_m) - \xi_0\right]^2$$

where the student and teacher activations are, respectively

$$h_l = \sqrt{\frac{1}{N}}\mathbf{w}_l^{\mathrm{T}}\mathbf{x} \qquad k_m = \sqrt{\frac{1}{N}}(\mathbf{w}_m^*)^{\mathrm{T}}\mathbf{x}, \tag{3.2}$$

and ξ_m, $m = 1..M$ and ξ_0 are noise variables corrupting the teacher activations and output respectively.

Given a training example $(\mathbf{x}, y)$, the student weights are again updated by a gradient descent step with learning rate η,

$$\mathbf{w}_l' - \mathbf{w}_l = -\eta\nabla_{\mathbf{w}_l}E = -\frac{\eta}{\sqrt{N}}\mathbf{x}\partial_{h_l}E \tag{3.3}$$

As before, the generalization error is defined to be the average error that the student makes on a test example selected at random (and uncorrelated with the training set), which we write as $\epsilon_{\mathrm{g}} = \langle E\rangle$.

Although one could, in principle, model the student weight dynamics directly, this will typically involve too many parameters, and we seek a more compact representation for the evolution of the generalization error. It is straightforward to show that the generalization error depends, not on a detailed description of all the network weights, but only on the overlap parameters $Q_{ll'} = \frac{1}{N}\mathbf{w}_l^{\mathrm{T}}\mathbf{w}_{l'}$ and $R_{lm} = \frac{1}{N}\mathbf{w}_l^{\mathrm{T}}\mathbf{w}_m^*$ (Biehl and Schwarze, 1995; Saad and Solla, 1995; Sollich and Barber, 1997b). In the case of infinite α, it is possible to obtain a closed set of equations governing the overlap parameters Q, R (Saad and Solla, 1995). For finite training sets, however, this is no longer possible, due to the correlations between successive weight updates (Sollich and Barber, 1997b).

In order to overcome this difficulty, we use a technique developed originally to study statistical physics systems (Coolen et al., 1996). Initially, consider the dynamics of a general vector of order parameters, denoted by Ω, which are functions of the network weights $\mathbf{w}$. If the weight updates are described by a transition probability $T(\mathbf{w} \to \mathbf{w}')$, then an approximate update equation for Ω is

$$\Omega' - \Omega = \left\langle \int d\mathbf{w}' \, (\Omega(\mathbf{w}') - \Omega(\mathbf{w})) \, T(\mathbf{w} \to \mathbf{w}') \right\rangle_{P(\mathbf{w})\propto\delta(\Omega(\mathbf{w})-\Omega)} \tag{3.4}$$

Intuitively, the integral in the above equation expresses the average change[15] of Ω caused by a weight update $\mathbf{w} \to \mathbf{w}'$, starting from (given) initial weights $\mathbf{w}$. Since our aim is to develop a closed set of equations for the order parameter dynamics, we need to remove the dependency on the initial weights $\mathbf{w}$. The only information we have regarding $\mathbf{w}$ is contained in the chosen order parameters Ω, and we therefore average the result over the 'subshell' of all $\mathbf{w}$

[15] Here we assume that the system size N is large enough that the mean values of the parameters alone describe the dynamics sufficiently well (*i.e.*, self-averaging holds).

which correspond to these values of the order parameters. This is expressed as the δ-function constraint in equation(3.4).

It is clear that if the integral in (3.4) depends on $\mathbf{w}$ only through $\Omega(\mathbf{w})$, then the average is unnecessary and the resulting dynamical equations are exact. This is in fact the case for $\alpha \to \infty$ and $\Omega = \{Q, R\}$, the standard order parameters mentioned above (Saad and Solla, 1995). If this cannot be achieved, one should choose a set of order parameters to obtain approximate equations which are as close as possible to the exact solution. The motivation for our choice of order parameters is based on the linear perceptron case treated in Section 2 where, in addition to the standard parameters Q and R, the overlaps projected onto eigenspaces of the training input correlation matrix $\mathbf{A} = \frac{1}{N}\sum_{\mu=1}^{p} \mathbf{x}^\mu (\mathbf{x}^\mu)^{\mathrm{T}}$ are required[16]. We therefore split the eigenvalues of $\mathbf{A}$ into K equal blocks ($\kappa = 1 \ldots K$) containing $N' = N/K$ eigenvalues each, ordering the eigenvalues such that they increase with κ. We then define projectors $\mathbf{P}^\kappa$ onto the corresponding eigenspaces and take as order parameters:

$$Q^\kappa_{ll'} = \frac{1}{N'}\mathbf{w}_l^{\mathrm{T}}\mathbf{P}^\kappa \mathbf{w}_{l'} \quad R^\kappa_{lm} = \frac{1}{N'}\mathbf{w}_l^{\mathrm{T}}\mathbf{P}^\kappa \mathbf{w}^*_m \quad U^\kappa_{ls} = \frac{1}{N'}\mathbf{w}_l^{\mathrm{T}}\mathbf{P}^\kappa \mathbf{b}_s \tag{3.5}$$

where the $\mathbf{b}_s$ are linear combinations of the noise variables and training inputs,

$$\mathbf{b}_s = \frac{1}{\sqrt{N}} \sum_{\mu=1}^{p} \xi_s^\mu \mathbf{x}^\mu. \tag{3.6}$$

As $K \to \infty$, these order parameters become functionals of a continuous variable[17].

The updates for the order parameters (3.5) due to the weight updates (3.3) can be found by taking the scalar products of (3.3) with either projected student or teacher weights, as appropriate. This then introduces the following activation 'components',

$$h^\kappa_l = \sqrt{\frac{K}{N'}}\mathbf{w}_l^{\mathrm{T}}\mathbf{P}^\kappa \mathbf{x} \qquad k^\kappa_m = \sqrt{\frac{K}{N'}}(\mathbf{w}^*_m)^{\mathrm{T}}\mathbf{P}^\kappa \mathbf{x} \qquad c^\kappa_s = \sqrt{\frac{K}{N'}}\mathbf{x}^{\mathrm{T}}\mathbf{P}^\kappa \mathbf{b}_s \tag{3.7}$$

so that the student and teacher activations are $h_l = \frac{1}{K}\sum_\kappa h^\kappa_l$ and $k_m = \frac{1}{K}\sum_\kappa k^\kappa_m$, respectively. For the linear perceptron, the chosen order parameters form a complete set - the dynamical equations close, without need for the average in (3.4).

[16] The reader may wonder why the order parameters Q and R did not show up explicitly in our treatment of the linear case in Section 2. This is because R can be calculated directly (simply take the scalar product of (2.5) with $\mathbf{w}^*$). Given R, Q and the generalization error $\hat{\epsilon}_{\mathrm{g}}$ are trivially related because $\hat{\epsilon}_{\mathrm{g}} = (Q - 2R + 1)/2$ in the linear case.

[17] Note that the limit $K \to \infty$ is taken *after* the thermodynamic limit, i.e., $K \ll N$. This ensures that the number of order parameters is always negligible compared to N (otherwise self-averaging would break down).

For the nonlinear case, we now sketch the calculation of the order parameter update equations (3.4). Taken together, the integral over $\mathbf{w}'$ (a sum of p discrete terms in our case, one for each training example) and the subshell average in (3.4), define an average over the activations (3.2), their components (3.7), and the noise variables ξ_m, ξ_0. These variables turn out to be Gaussian distributed with zero mean, and therefore only their covariances need to be worked out. One finds that these are in fact given by the naive training set averages. For example,

$$\begin{aligned} \langle h_l^\kappa k_m \rangle &= \frac{1}{p}\sum_\mu \frac{K}{N}(\mathbf{w}_l)^{\mathrm{T}}\mathbf{P}^\kappa \mathbf{x}^\mu (\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{w}_m^* \\ &= \frac{K}{\alpha N}(\mathbf{w}_l)^{\mathrm{T}}\mathbf{P}^\kappa \mathbf{A}\mathbf{w}_m^* = \frac{a_\kappa}{\alpha}R_{lm}^\kappa, \end{aligned} \tag{3.8}$$

where we have used $\mathbf{P}^\kappa\mathbf{A} = a_\kappa \mathbf{P}^\kappa$ with a_κ 'the' eigenvalue of $\mathbf{A}$ in the κ-th eigenspace; this is well defined for $K \to \infty$ (see (Sollich, 1994) for details of the eigenvalue spectrum). The correlations of the activations and noise variables explicitly appearing in the error in (3.3) are calculated similarly to give,

$$\begin{aligned} &\langle h_l h_{l'} \rangle = \frac{1}{K}\sum_\kappa \frac{a_\kappa}{\alpha}Q_{ll'}^\kappa \\ &\langle h_l k_m \rangle = \frac{1}{K}\sum_\kappa \frac{a_\kappa}{\alpha}R_{lm}^\kappa \qquad \langle k_m k_{m'} \rangle = \frac{1}{K}\sum_\kappa \frac{a_\kappa}{\alpha}T_{mm'}^\kappa \\ &\langle h_l \xi_s \rangle = \frac{1}{K}\sum_\kappa \frac{1}{\alpha}U_{ls}^\kappa \qquad \langle k_m \xi_s \rangle = 0 \qquad \langle \xi_s \xi_{s'} \rangle = \delta_{ss'}\sigma_s^2 \end{aligned} \tag{3.9}$$

where the final equation defines the noise variances. The $T_{mm'}^\kappa$ are projected overlaps between teacher weight vectors, $T_{mm'}^\kappa = \frac{1}{N}(\mathbf{w}_m^*)^{\mathrm{T}}\mathbf{P}^\kappa \mathbf{w}_{m'}^*$. We will assume that the teacher weights and training inputs are uncorrelated, so that $T_{mm'}^\kappa$ is independent of κ. The required covariances of the 'component' activations are

$$\begin{array}{lllllllll} \langle k_m^\kappa h_l \rangle &=& \frac{a_\kappa}{\alpha}R_{lm}^\kappa & \langle k_m^\kappa k_{m'} \rangle &=& \frac{a_\kappa}{\alpha}T_{mm'}^\kappa & \langle k_m^\kappa \xi_s \rangle &=& 0 \\ \langle c_s^\kappa h_l \rangle &=& \frac{a_\kappa}{\alpha}U_{ls}^\kappa & \langle c_s^\kappa k_{m'} \rangle &=& 0 & \langle c_s^\kappa \xi_{s'} \rangle &=& \frac{a_\kappa}{\alpha}\sigma_s^2\delta_{ss'} \\ \langle h_l^\kappa h_{l'} \rangle &=& \frac{a_\kappa}{\alpha}Q_{ll'}^\kappa & \langle h_l^\kappa k_{m'} \rangle &=& \frac{a_\kappa}{\alpha}R_{lm}^\kappa & \langle h_l^\kappa \xi_s \rangle &=& \frac{1}{\alpha}U_{ls}^\kappa \end{array} \tag{3.10}$$

Using equation (3.3) and the definitions (3.7), we can now write down the dynamical equations, replacing the number of updates n by the continuous variable $t = n/N$ in the limit $N \to \infty$:

$$\partial_t R_{lm}^\kappa \quad = \quad -\eta \,\langle k_m^\kappa \partial_{h_l} E \rangle$$

$$\begin{aligned}
\partial_t U^{\kappa}_{ls} &= -\eta \left\langle c^{\kappa}_s \partial_{h_l} E \right\rangle \\
\partial_t Q^{\kappa}_{ll'} &= -\eta \left\langle h^{\kappa}_l \partial_{h_{l'}} E \right\rangle - \eta \left\langle h^{\kappa}_{l'} \partial_{h_l} E \right\rangle + \eta^2 \frac{a_\kappa}{\alpha} \left\langle \partial_{h_l} E \partial_{h_{l'}} E \right\rangle
\end{aligned} \tag{3.11}$$

where the averages are over zero mean Gaussian variables, with covariances (3.9,3.10). Using the explicit form of the error E, we have

$$\partial_{h_l} E = g'(h_l) \left[\sum_{l'} g(h_{l'}) - \sum_m g(k_m + \xi_m) - \xi_0 \right] \tag{3.12}$$

which, together with the equations (3.11) completes the description of the dynamics. The Gaussian averages in (3.11) can be straightforwardly evaluated in a manner similar to the infinite training set case (Saad and Solla, 1995), and we omit the rather cumbersome explicit form of the resulting equations.

We note that, in contrast to the infinite training set case, the student activations h_l and the noise variables c_s and ξ_s are now correlated through equation (3.10). Intuitively, this is reasonable as the weights become correlated, during training, with the examples in the training set. In calculating the generalization error, on the other hand, such correlations are absent, and one has the same result as for infinite training sets. The dynamical equations (3.11), together with (3.9,3.10) constitute our main result. They are exact for the limits of either a linear network ($R, Q, T \to 0$, so that $g(x) \propto x$) or $\alpha \to \infty$, and can be integrated numerically in a straightforward way. In principle, the limit $K \to \infty$ should be taken but, as shown below, relatively small values of K can be taken in practice.

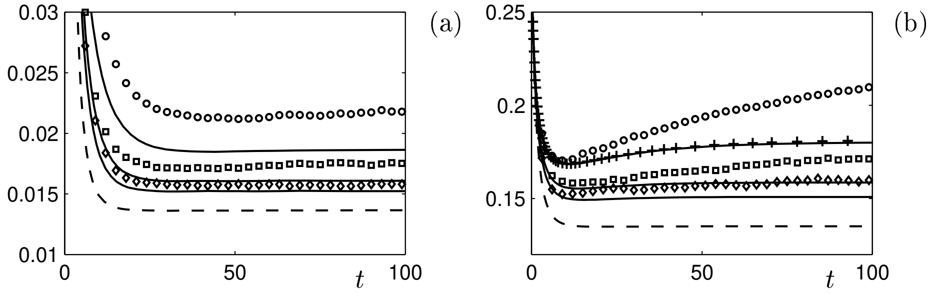

Figure 7: ϵ_{g} vs t for student and teacher with one hidden unit ($L = M = 1$); $\alpha = 2, 3, 4$ from above, learning rate $\eta = 1$. Noise of equal variance was added to both activations and output (a) $\sigma_1^2 = \sigma_0^2 = 0.01$, (b) $\sigma_1^2 = \sigma_0^2 = 0.1$. Simulations for $N = 100$ are shown by circles; standard errors are of the order of the symbol size. The bottom dashed lines show the infinite training set result for comparison. $K = 10$ was used for calculating the theoretical predictions; the curved marked "+" in (b), with $K = 20$ (and $\alpha = 2$), shows that this is large enough to be effectively in the $K \to \infty$ limit.

3.2 Results and Discussion

We now discuss the main consequences of our result (3.11), comparing the resulting predictions for the generalization dynamics, $\epsilon_{\rm g}(t)$, to the infinite training set theory and to simulations. Throughout, the teacher overlap matrix is set to $T_{ij} = \delta_{ij}$ (orthogonal teacher weight vectors of length $\sqrt{N}$).

In figure(7), we study the accuracy of our method as a function of the training set size for a nonlinear network with one hidden unit at two different noise levels. The learning rate was set to $\eta = 1$ for both (a) and (b). For small activation and output noise ($\sigma^2 = 0.01$), figure(7a), there is good agreement with the simulations for α down to $\alpha = 3$, below which the theory begins to underestimate the generalization error, compared to simulations. Our finite α theory, however, is still considerably more accurate than the infinite α predictions. For larger noise ($\sigma^2 = 0.1$, figure(7b)), our theory provides a reasonable quantitative estimate of the generalization dynamics for $\alpha > 3$. Below this value there is significant disagreement, although the qualitative behaviour of the dynamics is predicted quite well, including the overfitting phenomenon beyond $t \approx 10$. The infinite α theory in this case is qualitatively incorrect.

In the two hidden unit case, figure(8), our theory captures the initial evolution of $\epsilon_{\rm g}(t)$ very well, but diverges significantly from the simulations at larger t; nevertheless, it provides a considerable improvement on the infinite α theory. One reason for the discrepancy at large t is that the theory predicts that different student hidden units will always specialize to individual teacher hidden units for $t \to \infty$, whatever the value of α. This leads to a decay of $\epsilon_{\rm g}$ from a plateau value at intermediate times t. In the simulations, on the other hand, this specialization (or symmetry breaking) appears to be inhibited or at least delayed until very large t. This can happen even for zero noise and

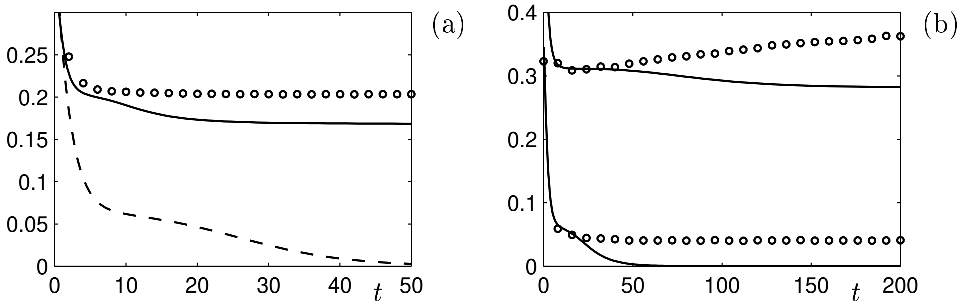

Figure 8: $\epsilon_{\rm g}$ vs t for two hidden units ($L = M = 2$). Left: $\alpha = 0.5$, with $\alpha = \infty$ shown by dashed line for comparison; no noise. Right: $\alpha = 4$, no noise (bottom) and noise on teacher activations and outputs of variance 0.1 (top). Simulations for $N = 100$ are shown by small circles; standard errors are less than the symbol size. Learning rate $\eta = 2$ throughout.

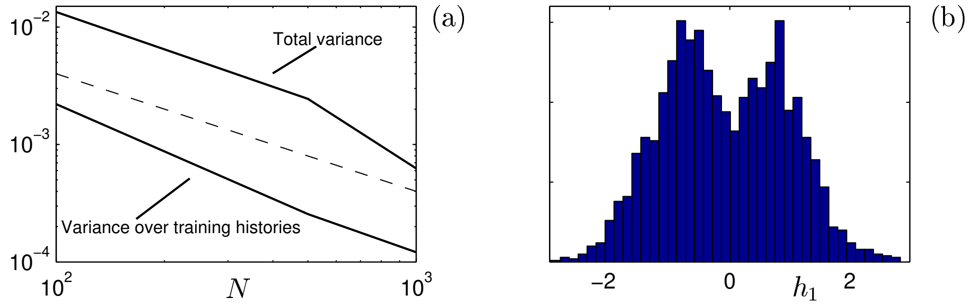

Figure 9: (a) Variance of $\epsilon_{\mathrm{g}}(t = 20)$ vs input dimension N for student and teacher with two hidden units ($L = M = 2$), $\alpha = 0.5$, $\eta = 2$, and zero noise. The bottom curve shows the variance due to different random choices of training examples from a fixed training set ('training history'); the top curve also includes the variance due to different training sets. Both are compatible with the $1/N$ decay expected if self-averaging holds (dotted line). (b) Distribution (over training set) of the activation h_1 of the first hidden unit of the student. Histogram from simulations for $N = 1000$, all other parameter values as in (a).

$\alpha \geq L$, where the training data should should contain enough information to force student and teacher weights to be equal asymptotically. The reason for this is not clear to us, and deserves further study. Our initial investigations, however, suggest that symmetry breaking may be strongly delayed due to the presence of saddle points in the training error surface with very 'shallow' unstable directions.

When our theory fails, which of its assumptions are violated? It is conceivable that multiple local minima in the training error surface could cause self-averaging to break down; however, we have found no evidence for this, see figure(9a). On the other hand, the simulation results in figure(9b) clearly show that the implicit assumption of Gaussian student activations – as discussed before eq. (3.8) – can be violated.

3.3 Conclusions for the non-linear theory

In summary, the main theoretical contribution of this section is the extension of online learning analysis for finite training sets to *nonlinear* networks. Our approximate theory does not require the use of replicas and yields ordinary first order differential equations for the time evolution of a set of order parameters. Its central implicit assumption (and its Achilles' heel) is that the student activations are Gaussian distributed. In comparison with simulations, we have found that it is more accurate than the infinite training set analysis at predicting the generalization dynamics for finite training sets, both qualitatively and also quantitatively for small learning times t. Future work will have to show whether the theory can be extended to cope with non-Gaussian student activations without incurring the technical difficulties of dynamical replica theory (Coolen et al., 1996) (see also Coolen et al., this volume), and whether this will help to capture the effects of local minima and, more generally, 'rough' training error surfaces.

References

Biehl, M. and Schwarze, H. (1995). Learning by online gradient descent. *Journal of Physics A*, 28:643–656.

Coolen, A. C. C., Laughton, S. N., and Sherrington, D. (1996). Modern Analytic Techniques to Solve the Dynamics of Recurrent Neural Networks. In Toutretzky, D. S., Mozer, M. C., and Hasslemo, M. E., editors, *Advances in Neural Information Processing Systems NIPS 8*. MIT Press.

Halkjær, S. and Winther, O. (1997). The effect of correlated input data on the dynamics of learning. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*, pages 169–175, Cambridge, MA. MIT Press.

Hertz, J. A., Krogh, A., and Thorbergsson, G. I. (1989). Phase transitions in simple learning. *Journal of Physics A*, 22:2133–2150.

Heskes, T. and Kappen, B. (1991). Learning processes in neural networks. *Physical Review A*, 44:2718–2762.

Krogh, A. and Hertz, J. A. (1992). Generalization in a linear perceptron in the presence of noise. *Journal of Physics A*, 25:1135–1147.

LeCun, Y., Kanter, I., and Solla, S. A. (1991). Eigenvalues of covariance matrices - application to neural- network learning. *Physical Review Letters*, 66(18):2396–2399.

Saad, D. and Solla, S. A. (1995). Online learning in soft committee machines. *Physical Review E*, 52:4225.

Sollich, P. (1994). Finite size effects in learning and generalization in linear perceptrons. *Journal of Physics A*, 27:7771–7784.

Sollich, P. (1995). Learning unrealizable tasks from minimum entropy queries. *Journal of Physics A*, 28:6125–6142.

Sollich, P. and Barber, D. (1997a). On-line learning from finite training sets. *Europhysics Letters*, 38:477–482.

Sollich, P. and Barber, D. (1997b). Online learning from finite training sets: An analytical case study. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*, pages 274–280, Cambridge, MA. MIT Press.

Wendemuth, A., Opper, M., and Kinzel, W. (1993). The effect of correlations in neural networks. *Journal of Physics A*, 26(13):3165–3185.