EUROPHYSICS LETTERS 20 May 1997

Europhys. Lett., **38** (6), pp. 477-482 (1997)

Online learning from finite training sets

PETER SOLLICH[1](*) and DAVID BARBER[2]

[1] *Department of Physics, University of Edinburgh, Edinburgh EH9 3JZ, U.K.*
[2] *Neural Computing Research Group, Aston University, Birmingham B4 7ET, U.K.*

(received 23 January 1997; accepted in final form 3 April 1997)

PACS. 87.10.+e – General, theoretical, and mathematical biophysics (including logic of biosystems, quantum biology, and relevant aspects of thermodynamics, information theory, cybernetics, and bionics).
PACS. 02.50.−r – Probability theory, stochastic processes, and statistics.
PACS. 05.90.+m – Other topics in statistical physics and thermodynamics.

Abstract. – We analyse online (gradient descent) learning of a rule from a *finite* set of training examples at *non-infinitesimal* learning rates η, calculating exactly the time-dependent generalization error for a simple model scenario. In the thermodynamic limit, we close the dynamical equation for the generating function of an infinite hierarchy of order parameters using 'within-sample self-averaging'. The resulting dynamics is non-perturbative in η, with a slow mode appearing only above a finite threshold $\eta_{\min}$. Optimal settings of η for given final learning time are determined and the results are compared with offline gradient descent.

Neural networks have been the subject of much recent research because of their ability to learn rules from examples. One of the most common learning algorithms is online gradient descent: The weights of a network ('student') are updated each time a training example from the training set is presented, such that the error on this example is reduced. In offline gradient descent, 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. It becomes equivalent to offline learning only in the limit where the learning rate $\eta \to 0$ [1]. In both cases, the main quantity of interest is the evolution of the generalization error: After a given number of weight updates, how well does the student approximate the input-output mapping ('teacher' rule) underlying the training examples?

We do not consider in the following non-gradient descent learning algorithms, and also restrict ourselves to gradient descent on the most common measure of error on a training example, the squared output deviation (see eq. (1) below). For interesting recent results on more general, optimized online learning algorithms, see, *e.g.*, [2, 3].

(*) Royal Society Dorothy Hodgkin Research Fellow. Email: `P.Sollich@ed.ac.uk`

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, 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.*, [1]). An explicit analysis of the time evolution of the generalization error for finite training sets was provided by Krogh and Hertz [4] for an offline learning scenario with $\eta \to 0$. For finite η, progress has been made in particular for so-called soft committee machine network architectures [5], but only for the case of infinite training sets. In this Letter, we provide the first exact calculation of generalization performance for a scenario with both non-infinitesimal learning rate η and finite training set size of α examples per weight. A discussion of our results from a more practical point of view can be found in the conference proceedings [6].

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

$$y = \mathbf{w}^{\mathrm{T}}\mathbf{x}/\sqrt{N}.$$

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 to ensure typical outputs of $O(1)$ for input and weight components of $O(1)$. Whenever a training example $(\mathbf{x}, y)$ is presented to the network, its weight vector is updated along the gradient of the squared error on this example, *i.e.*,

$$\Delta\mathbf{w} = -\eta\, \nabla_{\mathbf{w}} \tfrac{1}{2}(y - \mathbf{w}^{\mathrm{T}}\mathbf{x}/\sqrt{N})^2 = \eta\,(y\mathbf{x}/\sqrt{N} - \tfrac{1}{N}\mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{w}) \tag{1}$$

where η is the learning rate. We are interested in online learning from finite training sets, where for each update an example is chosen randomly (with replacement) from a given set $\{(\mathbf{x}^\mu, y^\mu), \mu = 1, \ldots, p\}$ of p training examples. (The case of cyclical presentation of examples [7] is left for future study.) If example μ is chosen for update n, the weight vector is changed to

$$\mathbf{w}_{n+1} = \{1 - \eta\,\tfrac{1}{N}[\mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}} + \gamma]\}\mathbf{w}_n + \eta\, y^\mu \mathbf{x}^\mu/\sqrt{N} \tag{2}$$

Here we have also included a weight decay γ; the rescaled version $\lambda = \gamma\alpha$ (where $\alpha = p/N$ is the number of examples per weight) corresponds to the weight decay commonly used in offline learning [4]. For simplicity, all student weights are assumed to be initially zero, *i.e.*, $\mathbf{w}_{n=0} = \mathbf{0}$.

The main quantity of interest is the evolution of the *generalization error* of the student. 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; generalization of our results to nonlinear perceptron teachers is straightforward using the methods of [8]. The teacher weight vector is taken to be normalized to $\mathbf{w}_*^2 = N$ for simplicity, and the input vectors are assumed to be sampled randomly from an isotropic distribution over the hypersphere $\mathbf{x}^2 = N$. The generalization error, defined as the average of the squared error between student and teacher outputs for random inputs, is then

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

In order to make the scenario analytically tractable, we focus on the thermodynamic 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 and can be calculated as an 'annealed' average over the random selection of examples from a given training set followed by a 'quenched' average over all training sets.

We begin by deriving from (2) an update equation for the annealed average (denoted $\langle\ldots\rangle$) of a 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 $N\times N$ matrix. Performing the average over the random choice of training example for update n explicitly (and discarding terms of relative order $\mathrm{O}(N^{-1})$), we find

$$N\left(\langle\epsilon_{n+1}\rangle - \langle\epsilon_n\rangle\right) = \frac{\tilde\eta}{N}(\mathbf{b}-\lambda\mathbf{w}_*)^{\mathrm{T}}\mathbf{M}\langle\mathbf{v}_n\rangle - \frac{\tilde\eta}{N}\left\langle \mathbf{v}_n^{\mathrm{T}}\left[\lambda\mathbf{M}+\frac{1}{2}(\mathbf{A}\mathbf{M}+\mathbf{M}\mathbf{A})\right]\mathbf{v}_n\right\rangle$$
$$+\quad \frac{\tilde\eta^2\alpha}{N}\sum_\mu \frac{1}{N}(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{M}\mathbf{x}^\mu\left\{\frac{1}{2}(\xi^\mu)^2 - \xi^\mu\frac{1}{\sqrt{N}}(\mathbf{x}^\mu)^{\mathrm{T}}\langle\mathbf{v}_n\rangle + \frac{1}{2N}\left\langle\mathbf{v}_n^{\mathrm{T}}\mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{v}_n\right\rangle\right\} \quad (3)$$

where $\tilde\eta = \eta/\alpha$ is a rescaled learning rate, $\mathbf{A} = \frac{1}{N}\sum_\mu \mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}}$ is the correlation matrix of the training inputs, and $\mathbf{b} = \frac{1}{\sqrt{N}}\sum_\mu \xi^\mu\mathbf{x}^\mu$. We now want to transform (3) into a closed dynamical equation for $\langle\epsilon_n\rangle$. The first term in curly brackets depends on quenched variables only and therefore acts as an unproblematic constant w.r.t. the annealed average. The two terms linear in $\langle\mathbf{v}_n\rangle$ are also straightforward: An annealed average of (2) yields directly

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

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

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

which again depends only on quenched variables. The terms in (3) quadratic in $\mathbf{v}_n$ 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. We treat these 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 the *order parameter generating function*([1]) $\epsilon_n(h) = \frac{1}{2N}\mathbf{v}_n^{\mathrm{T}}\exp(h\mathbf{A})\mathbf{v}_n$. This still leaves the last term in (3), which cannot be obtained in this way due to the presence of the factors $c^\mu = \frac{1}{N}(\mathbf{x}^\mu)^{\mathrm{T}}\exp(h\mathbf{A})\mathbf{x}^\mu$. Using arguments from [10], however, it can be shown that these are 'within-sample self-averaging': Up to fluctuations of $\mathrm{O}(N^{-1/2})$, all c^μ are equal to each other and hence to the training set ('sample') average

$$\frac{1}{p}\sum_\mu c^\mu = \frac{1}{\alpha N}\operatorname{tr}\mathbf{A}\exp(h\mathbf{A}) = \frac{1}{\alpha}g'(h) \qquad \text{where} \qquad g(h) = \frac{1}{N}\operatorname{tr}\exp(h\mathbf{A}).$$

Thus, the last term on the r.h.s. of (3) becomes ($\partial_h = \partial/\partial h$)

$$\frac{\tilde\eta^2\alpha}{N}\,\frac{1}{\alpha}\,g'(h)\sum_\mu\frac{1}{2N}\left\langle\mathbf{v}_n^{\mathrm{T}}\mathbf{x}^\mu(\mathbf{x}^\mu)^{\mathrm{T}}\mathbf{v}_n\right\rangle = \tilde\eta^2 g'(h)\,\frac{1}{2N}\left\langle\mathbf{v}_n^{\mathrm{T}}\mathbf{A}\mathbf{v}_n\right\rangle = \tilde\eta^2 g'(h)\,\partial_h\epsilon_n(h=0)$$

Combining all the above ingredients, the dynamical equation (3) does indeed close. For $N\to\infty$, where $\langle\epsilon_n(h)\rangle\to\epsilon(t,h)$, it takes the form

$$\left[\partial_\tau + 2(\lambda+\partial_h)\right]\epsilon(\tau,h) = V_1(\tau,h) + \tilde\eta\,V_2(\tau,h) + \tilde\eta\,g'(h)\,\partial_h\epsilon(\tau,h=0) \quad (4)$$

with $V_{1,2}(\tau,h)$ known functions of quenched variables. A rescaled learning time $\tau = \tilde\eta t$ has been introduced here, which is the relevant time variable for the limit of infinitesimal

([1]) A similar approach has recently been used to study simple spin glass dynamics [9]; we are grateful to A.C.C. Coolen for pointing out this connection.

learning rate, $\eta \to 0$. The linear PDE (4) is most easily solved in terms of Laplace transforms w.r.t. τ, $\hat{\epsilon}(z,h) = \int_0^\infty d\tau\, \exp(-z\tau)\, \epsilon(\tau,h)$ etc., with the initial condition $\epsilon(\tau=0,h) = \frac{1}{2N}\mathbf{w}_*^{\mathrm{T}} \exp(h\mathbf{A})\mathbf{w}_*$. Formally treating the term $\partial_h \epsilon(\tau, h=0)$ as a known inhomogeneity, the solution is obtained from the Laplace transform of the r.h.s. of (4) by convolution (over h) with the appropriate Green's function of the differential operator $2\partial_h + z + 2\lambda$. We write the result in the form

$$\hat{\epsilon}(z,h) = \hat{v}_1(z,h) + \tilde{\eta}\,\hat{v}_2(z,h) + \tilde{\eta}\,\hat{u}(z,h)\,\partial_h \hat{\epsilon}(z,h=0). \tag{5}$$

Differentiating w.r.t. h and setting $h = 0$ then gives a simple self-consistency equation for $\partial_h \hat{\epsilon}(z,h=0)$, whose solution can be inserted back into (5). Finally, the physically relevant Laplace transform of the time-dependent generalization error $\epsilon_{\mathrm{g}}(t)$ is found by setting $h = 0$. Using $\hat{v}_2\,\partial_h \hat{u} = \hat{u}\,\partial_h \hat{v}_2$ (see (7) below), one obtains:

$$\hat{\epsilon}_{\mathrm{g}}(z) = \tilde{\eta}\int_0^\infty dt\, e^{-z\tilde{\eta}t}\epsilon_{\mathrm{g}}(t) = \hat{\epsilon}(z,h=0) = \left.\left\{\hat{v}_1(z,h) + \tilde{\eta}\,\frac{\hat{v}_2(z,h) + \hat{u}(z,h)\,\partial_h \hat{v}_1(z,h)}{1-\tilde{\eta}\,\partial_h \hat{u}(z,h)}\right\}\right|_{h=0} \tag{6}$$

The remaining average over the quenched variables can be carried out directly on $\hat{u}(z,h)$ and $\hat{v}_{1,2}(z,h)$ since these are self-averaging. The results can be expressed as averages $\langle \ldots \rangle_a$ over the (self-averaging) eigenvalue spectrum $\rho(a)$ [10, 11] of $\mathbf{A}$:

$$\begin{aligned} \hat{v}_1(z,h) &= \left\langle \frac{e^{ha}}{2(\lambda+a)^2}\left[\frac{\sigma^2 a + \lambda^2}{z} + \frac{2a(\lambda-\sigma^2)}{z+\lambda+a} + \frac{a(\sigma^2+a)}{z+2\lambda+2a}\right]\right\rangle_a \\ \hat{u}(z,h) &= \left\langle \frac{ae^{ha}}{z+2\lambda+2a}\right\rangle_a, \qquad \hat{v}_2(z,h) = \hat{u}(z,h)\,\frac{\sigma^2}{z}\left[\frac{\alpha}{2} - \left\langle \frac{a}{z+\lambda+a}\right\rangle_a\right] \end{aligned} \tag{7}$$

Once h is set to zero, a closed form for all required averages can be obtained in terms of the known 'response function' [10, 11] $G(\alpha,\lambda) = \left\langle (\lambda+a)^{-1}\right\rangle_a$.

Our main result (6) yields directly the asymptotic generalization error, $\epsilon_\infty = \epsilon_{\mathrm{g}}(t\to\infty) = \lim_{z\to 0} z\hat{\epsilon}_{\mathrm{g}}(z)$. 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 [1]. Intuitively, the larger η, the more the online 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 [1] 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 and online learning does not lead to weight diffusion because all individual updates vanish asymptotically. Numerical evaluation of ϵ_∞ shows this to be indicative of a more general trend [6]: For small training sets ($\alpha \approx 1$ or less), large learning rates $\eta = O(1)$ can be used without increasing ϵ_∞ significantly, whereas for large α, where $\epsilon_\infty = \frac{1}{2}\sigma^2[1/\alpha + \eta/(2-\eta)]$, one needs small $\eta = O(\alpha^{-1})$ to keep relative increases small. Eq. (6) also shows that as η is increased, ϵ_∞ eventually diverges at a critical learning rate $\eta_{\mathrm{c}}(\alpha,\lambda) = \alpha/\partial_h \hat{u}(z=0,h=0)$: As $\eta \to \eta_{\mathrm{c}}$, the 'overshoot' of the weight update steps becomes so large that the weights eventually diverge. Weight decay counteracts this by reducing the length of the weight vector at each update, and η_{c} therefore increases with λ (fig. 1b-d).

Consider now the large t behaviour of the generalization error $\epsilon_{\mathrm{g}}(t)$. From the Laplace transform (6,7), one sees that for small η, the most slowly decaying mode $\exp(-ct)$ of $\epsilon_{\mathrm{g}}(t)$ has a decay constant $c = \eta(\lambda + a_{\min})/\alpha$, with $a_{\min} = (1-\sqrt{\alpha})^2$ the smallest non-zero eigenvalue of $\mathbf{A}$ [10, 11]. This scales linearly with η, the size of the weight updates, as expected (fig. 1a). For small α, the condition $ct \gg 1$ for $\epsilon_{\mathrm{g}}(t)$ to have reached its asymptotic value ϵ_∞ is $\eta(1+\lambda)(t/\alpha) \gg 1$ and scales with t/α, the number of times each training example has

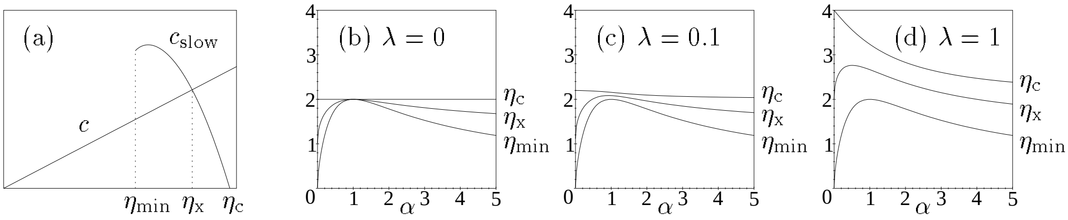

Fig. 1. – Sketch of definitions of $\eta_{\rm min}$ (minimal learning rate for slow mode), $\eta_{\rm x}$ (crossover to slow mode dominated convergence) and $\eta_{\rm c}$ (critical η; see text for details), and their dependence on α.

been used. For large α, on the other hand, the condition becomes $\eta t \gg 1$: α drops out since convergence occurs before repetitions of training examples become significant.

For larger η, the denominator of (6) can produce an additional pole of $\hat{\epsilon}_{\rm g}(z)$ on the real z-axis, giving rise to a new slow mode. From the maximum of $\partial_h \hat{u}(z, h=0)$ w.r.t. z (for real z), one finds that this mode exists only for η above the finite threshold $\eta_{\rm min} = 2/(\alpha^{1/2}+\alpha^{-1/2}-1)$. For finite α, it is therefore non-perturbative, *i.e.*, could not be predicted from a small η expansion of $\epsilon_{\rm g}(t)$. Its decay constant $c_{\rm slow}$ decreases to zero as $\eta \to \eta_{\rm c}$, and crosses that of the normal mode at $\eta_{\rm x}(\alpha, \lambda)$ (fig. 1a). For $\eta > \eta_{\rm x}$, the slow mode therefore determines the convergence speed for large t, and fastest convergence is obtained for $\eta = \eta_{\rm x}$. From fig. 1b-d, we see that for λ not too large, $\eta_{\rm x}$ has a maximum at $\alpha \approx 1$ (where $\eta_{\rm x} \approx \eta_{\rm c}$); for larger α, on the other hand, it decays as $\eta_{\rm x} = 1 + 2\alpha^{-1/2} \approx \frac{1}{2}\eta_{\rm c}$. This is because for $\alpha \approx 1$ the eigenvalue spectrum of $\mathbf{A}$ is very broad, extending from $(1-\sqrt{\alpha})^2 \to 0$ to $(1+\sqrt{\alpha})^2 \approx 2$ [10, 11]. This corresponds to a (total training) error surface which is very anisotropic around its minimum in weight space. The steepest directions determine $\eta_{\rm c}$ and convergence along them would be fastest for $\eta = \frac{1}{2}\eta_{\rm c}$ (as in the isotropic $\alpha \to \infty$ case). However, the overall convergence speed is determined by the shallow directions, which require maximal $\eta \approx \eta_{\rm c}$ for fastest convergence.

The general points made above are illustrated by the finite t behaviour of $\epsilon_{\rm g}(t)$ shown in fig. 2. The theoretical values, which are obtained by numerical inversion of the Laplace transform (6), are seen to be in excellent agreement with simulations results for $N = 50$. This shows that finite size effects are generally not significant even for such fairly small N.

Above, we saw that the *asymptotic* generalization error ϵ_∞ is minimal for $\eta = 0$. Fig. 3 shows what happens if we minimize $\epsilon_{\rm g}(t)$ 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

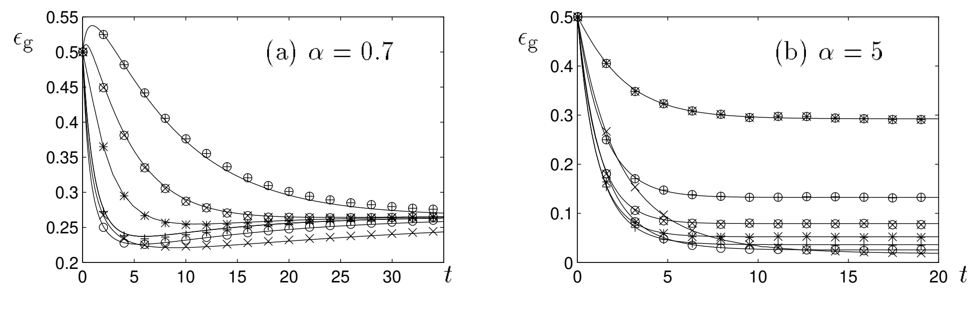

Fig. 2. – $\epsilon_{\rm g}$ vs t for different η. Simulations for $N = 50$ are shown by symbols (standard errors less than symbol sizes). $\lambda = 10^{-4}$, $\sigma^2 = 0.1$, α as shown. The learning rate η increases from below (at large t) over the range (a) $0.5 \ldots 1.95$, (b) $0.5 \ldots 1.75$.

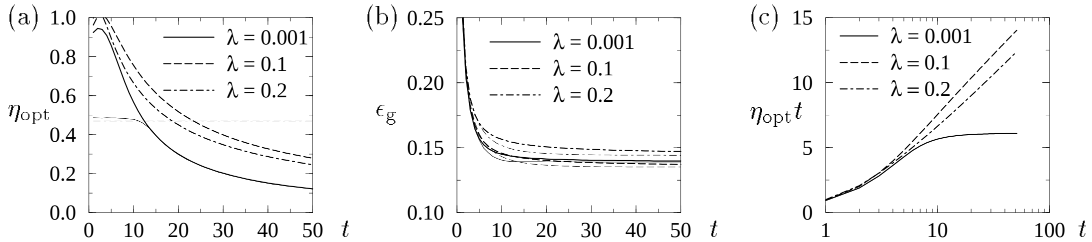

Fig. 3. – (a) Optimal η vs. final learning time t for online (bold) and offline learning (thin lines), and (b) resulting $\epsilon_{\rm g}$; (c) scaling of $\eta_{\rm opt}t$ with $\ln t$. $\alpha = 1$, $\sigma^2 = 0.1$, λ as shown.

η decreases towards zero as required by the tradeoff between asymptotic performance and convergence speed: Minimizing $\epsilon_{\rm g}(t) \approx \epsilon_\infty + {\rm const} \cdot \exp(-ct) \approx c_1 + \eta c_2 + c_3 \exp(-c_4 \eta t)$ leads to $\eta_{\rm opt} = (a + b \ln t)/t$ (with some constants a, b, $c_{1\ldots4}$). Although derived for small η, this functional form also provides a good description down to fairly small t (fig. 3c), where $\eta_{\rm opt}$ becomes large. For $\lambda < \sigma^2$, however, where for small η the generalization error $\epsilon_{\rm g}(t)$ has a minimum at some finite t (a phenomenon normally referred to as 'over-training' [11]; see fig. 2a), $\eta_{\rm opt}$ behaves asymptotically as $\eta_{\rm opt} \propto t^{-1}$, without the $\ln t$ factor. This corresponds to a fixed effective learning time ηt required to reach this minimum.

In fig. 3b, we also compare the performance of online learning to that of offline learning (calculated from the appropriate finite η version of [4]), again with optimized values of η for given t. The performance loss from using online instead of offline (gradient descent) learning is seen to be negligible. This may seem surprising given the stochasticity of weight updates in online learning, in particular for small t. However, fig. 3a shows that online learning can make up for this by allowing larger values of η to be used. This advantage should become even more pronounced for input distributions with non-zero mean: for online learning, $\eta_{\rm c}$ is not significantly affected, whereas for the offline case a drastic reduction (by a factor of $O(N^{-1})$) can result [12]. This issue deserves future study, as does dynamic (t-dependent) optimization of η; performance improvements over optimal t-independent η may however be small [7]. We also hope to extend our approach to more complicated network architectures in which the crucial question of learning dynamics with local minima can be addressed.

REFERENCES

[1] Heskes T. M. and Kappen B., *Phys. Rev. A*, **44** (1991) 2718
[2] Opper M., *Phys. Rev. Lett.*, **77** (1996) 4671 and references therein
[3] Kinouchi O. and Caticha N., *Phys. Rev. E*, **52** (1995) 2878
[4] Krogh A. and Hertz J. A., *J. Phys. A*, **25** (1992) 1135
[5] Saad D. and Solla S. A., *Phys. Rev. Lett.*, **74** (1995) 4337; *Phys. Rev. E*, **52** (1995) 4225; Biehl M. and Schwarze H., *J. Phys. A*, **28** (1995) 643
[6] Sollich P. and Barber D., in *Advances in Neural Information Processing Systems 9*, edited by Mozer M. C., Jordan M. I. and Petsche T., (MIT Press, Cambridge, MA) 1997 (in press)
[7] Luo Z.-Q., *Neural Comp.*, **3** (1991) 226; Heskes T. and Wiegerinck W., *IEEE Trans. Neural Netw.*, **7** (1996) 919
[8] Sollich P., *J. Phys. A*, **28** (1995) 6125
[9] Bonilla L. L. *et al.*, *Europhys. Lett.*, **34** (1996) 159; *Phys. Rev. B*, **54** (1996) 4170
[10] Sollich P., *J. Phys. A*, **27** (1994) 7771
[11] Hertz J. A., Krogh A. and Thorbergsson G. I., *J. Phys. A*, **22** (1989) 2133
[12] Watkin T. L. H., Rau A. and Biehl M., *Rev. Modern Phys.*, **65** (1993) 499