Universität Würzburg

Institut für Theoretische Physik

Am Hubland, D–97074 Würzburg, Germany

On–line backpropagation in two–layered neural networks

Peter Riegler and Michael Biehl

Ref.: WUE-ITP-95-018
e-mail : pr@physik.uni-wuerzburg.de
ftp : ftp.physik.uni-wuerzburg.de

On–line backpropagation in two–layered neural networks

Peter Riegler[1] and Michael Biehl
Institut für Theoretische Physik
Julius–Maximilians–Universität
Am Hubland, D–97074 Würzburg, Germany

Abstract

We present an exact analysis of learning a rule by on–line gradient descent in a two–layered neural network with adjustable hidden–to–output weights (backpropagation of error). Results are compared with the training of networks having the same architecture but fixed weights in the second layer.

PACS.: 07.05.Mh, 87.10, 02.50

The ability of neural networks to learn a rule from examples [1] has been studied successfully in a statistical mechanics context, see e.g. [2, 3, 4] for recent reviews. So far most of the analysis has been restricted to very simple networks like the single layer perceptron [1] or networks with one layer of hidden units and a fixed hidden–to–output relation, e.g. the so–called committee machine [3].

In the following we extend the recent investigation of learning by on–line gradient descent [8, 10] to two–layered networks with adjustable weights connecting the hidden units and the output. This topic is of crucial importance as systems with variable hidden–to–output relations can realize more complex classification schemes and are commonly used in practical applications of neural networks [1, 5].

The learning prescription corresponds to an on–line version of the so–called 'backpropagation of error' [1], a method widely used in practice [5]. In the theory of on–line learning (e.g. [6, 7, 8, 10] and ref. therein) it is assumed that a sequence of examples $\{\boldsymbol{\xi}^\mu, \tau^\mu\}$ for an unknown rule is provided by the environment. Here, $\boldsymbol{\xi}^\mu$ denotes an example input vector and $\tau^\mu = \tau(\boldsymbol{\xi}^\mu) \in \mathbb{R}$ is the corresponding correct rule output.

Consider a student network with a single continuous output $\sigma(\mathbf{W}, \boldsymbol{\xi})$ where $\mathbf{W}$ denotes the set of all weights in the network. The frequently used quadratic error measure

$$\epsilon(\mathbf{W}, \boldsymbol{\xi}) = \frac{1}{2}[\sigma(\mathbf{W}, \boldsymbol{\xi}) - \tau(\boldsymbol{\xi})]^2 \tag{1}$$

quantifies the degree of disagreement between the student and the rule output for a particular input. Throughout this paper we consider independently drawn

---

[1] email: pr@physik.uni-wuerzburg.de, biehl@physik.uni-wuerzburg.de

vectors $\boldsymbol{\xi}$ with uncorrelated random components of zero mean and unit variance. Denoting the average over this input distribution by $\langle \ldots \rangle_\xi$ we define the generalization error

$$\epsilon_g(\mathbf{W}) = \langle\, \epsilon(\mathbf{W}, \boldsymbol{\xi})\, \rangle_\xi . \tag{2}$$

This quantity measures the validity of the hypothesis for the rule, which is defined through the student architecture and its weights $\mathbf{W}$.

In the on–line scheme a new, uncorrelated example is presented at each learning step μ and the set of student weights is updated instantaneously. In the following we will consider an update according to the gradient of $\epsilon(\mathbf{W}, \boldsymbol{\xi}^\mu)$ with respect to the weights:

$$\mathbf{W}^{\mu+1} = \mathbf{W}^\mu - \tilde{\eta}[\sigma(\mathbf{W}^\mu, \boldsymbol{\xi}^\mu) - \tau^\mu]\ \nabla_{\mathrm{W}}\, \sigma(\mathbf{W}, \boldsymbol{\xi}^\mu)|_{\mathbf{W}^\mu} . \tag{3}$$

The learning rate $\tilde{\eta}$ controls the size of the steps made in the direction of steepest descent.

As a specific example, we study the learning of a rule which can be parametrized in terms of a *teacher* neural network with one hidden layer of M continuous nodes with sigmoid activation function and a single linear output unit. We will distinguish two different types of network models in the following:

In a *fully connected architecture* all M hidden units receive information from the same N input nodes. The total output of such a teacher network is given by

$$\tau(\boldsymbol{\xi}) = \sum_{n=1}^{M} v_n g(y_n) \quad \text{with} \quad y_n = \mathbf{B}_n \cdot \boldsymbol{\xi} \tag{4}$$

with weights $\mathbf{B}_n \in \mathbb{R}^N$ connecting the n–th hidden unit with the input $\boldsymbol{\xi} \in \mathbb{R}^N$ and the set $\{v_n\}_{n=1,\ldots M}$ of hidden–to–output connections.

Alternatively we consider a so–called *tree–like architecture*, where the hidden units are connected to nonoverlapping receptive fields, each of which is taken to be N–dimensional. Thus, the teacher output is also of the form (4) but with $y_n = \mathbf{B}_n \cdot \boldsymbol{\xi}_n$ where $\boldsymbol{\xi}_n$ is the n–th subset of the $M \cdot N$–dimensional input $\boldsymbol{\xi} = (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots \boldsymbol{\xi}_M)$.

As an example for a sigmoid activation function we choose $g(y) = \mathrm{erf}\left(y/\sqrt{2}\right)$ [8, 10], but our results should not depend upon this choice crucially.

The student is taken to be a network of the corresponding architecture with K hidden units, input–to–hidden weights $\mathbf{J}_i \in \mathbb{R}^N$, and adjustable hidden–to–output weights w_i, $i = 1, \ldots K$. Its output is given by

$$\sigma(\boldsymbol{\xi}) = \sum_{i=1}^{K} w_i g(x_i). \tag{5}$$

The quantities x_i are defined as $x_i = \mathbf{J}_i \cdot \boldsymbol{\xi}$ in the overlapping architecture with an N–dimensional input and $x_i = \mathbf{J}_i \cdot \boldsymbol{\xi}_i$ in the case of K nonoverlapping receptive fields. In both cases, we will scale the learning rate in equation (3) with the number of inputs to a single hidden unit: $\tilde{\eta} = \eta/N$.

Given a specific teacher network, the generalization error (2) of the student depends in the thermodynamic limit $N \to \infty$, $M, K \propto \mathcal{O}(1)$ only on the $\{w_i\}$ and the *order parameters*

$$R_{im} = \mathbf{J}_i \cdot \mathbf{B}_m, \qquad Q_{ij} = \mathbf{J}_i \cdot \mathbf{J}_j, \qquad m = 1, \dots M \quad \text{and} \quad i, j = 1, \dots K. \tag{6}$$

It is straightforward to derive from (3) recursion relations for the mean values of all these quantities by performing the average over the latest example input [8, 10]. It is furthermore possible to show that in the thermodynamic limit the overlap parameters $\{R_{im}, Q_{ij}\}$ as well as the adjustable weights $\{w_i\}$ become selfaveraging quantities [12]. Thus the description in terms of their mean values is sufficient. In the same limit one can interpret $\alpha = \mu/N$ as a 'continuous time' and obtains ordinary differential equations for the evolution of the learning network:

$$\frac{dR_{im}}{d\alpha} = \eta \langle\, \delta_i y_m \,\rangle, \qquad \frac{dQ_{ij}}{d\alpha} = \eta \langle\, \delta_i x_j + \delta_j x_i \,\rangle + \eta^2 \langle\, \delta_i \delta_j \,\rangle, \qquad \frac{dw_i}{d\alpha} = \langle\, g(x_i)\, \Delta \,\rangle \tag{7}$$
$$\text{where } \Delta = \left[\sum_{n=1}^{M} v_n g(y_n) - \sum_{i=1}^{K} w_i g(x_i) \right] \quad \text{and} \quad \delta_i = w_i\, g'(x_i)\, \Delta.$$

The averages are over the $M{\cdot}K$–dimensional Gaussian distribution of the $\{x_i, y_m\}$ which is determined through the correlations
$\langle\, x_i x_j \,\rangle = Q_{ij}, \quad \langle\, x_i y_m \,\rangle = R_{im}, \quad \langle\, y_m y_n \,\rangle = \mathbf{B}_m \cdot \mathbf{B}_n \equiv T_{mn}$
for overlapping architectures and
$\langle\, x_i x_j \,\rangle = \delta_{ij} Q_{ij}, \quad \langle\, x_i y_m \,\rangle = \delta_{im} R_{im}, \quad \langle\, y_m y_n \,\rangle = \delta_{mn} T_{mn}$
in the case of tree–like student and teacher networks. All averages in (7) can be performed exactly [8, 10] and a numerical integration of the differential equations yields the evolution of the overlaps and hidden–to–output weights, and thus the learning curve $\epsilon_g(\alpha)$.

In this letter we analyse only the special symmetric cases where

$$v_n = v \quad \text{and} \quad T_{mn} = T\delta_{mn} \quad \text{for all} \quad m, n = 1, 2, \dots M \ll N. \tag{8}$$

For large N, this corresponds to uncorrelated vectors B_n with independent random components of zero mean and variance $1/N$. Note, however, that the following is easily extended to more general asymmetric settings.

Here we furthermore restrict ourselves to situations where $K = M$, i.e. the rule is perfectly learnable for the student. The extension to unlearnable rules (e.g. $K < M$) and oversophisticated students ($K > M$) will be presented in a forthcoming publication [12].

For $K = M$ the number of dynamical variables in (7) is quadratic in K. However, due to the symmetric architecture of the teacher network assumed in

(8) the time evolution rapidly leads to the equality of corresponding variables in different branches of the student network (see fig. 1 for an example). Hence for $\alpha \to \infty$ the student network can be described in terms of only five variables $R = R_{ii}$, $S = R_{im}$, $Q = Q_{ii}$, $C = Q_{ij}$, and $w = w_i$ regardless of the actual number of hidden units.

a) *tree–like architecture:* Here each branch of the network receives a different part of the input vector $\boldsymbol{\xi} = (\boldsymbol{\xi}_1, .., \boldsymbol{\xi}_K)$ [3, 9]. Therefore the 'off-diagonal' order parameters S and C do not carry any significant information and hence need not to be considered.

In order to solve the asymptotics of the remaining dynamical variables R, Q, and w we linearize the equations of motion (7) for small deviations from the optimal solution $R_\infty = Q_\infty = T, w_\infty = v$. Defining $\mathbf{V}^{(3)} = (R-T, Q-T, w-v)^\top$ the linearization is of the form

$$\frac{d\mathbf{V}^{(3)}}{d\alpha} = \mathbf{m}^{(3)}\mathbf{V}^{(3)}, \tag{9}$$

where here and in the following the upper index denotes the dimension of the corresponding matrix.

For the sake of comparison we also consider the case of fixed hidden–to–output couplings in the student network. In this case we let $w \equiv v$ in order to keep the task learnable. It should be emphasized that learning the rule should be a much simpler task for such a student, as the rule parameters v_n are assumed to be known *a priori*.

Defining $\mathbf{V}^{(2)} = (R-T, Q-T)^\top$ correspondingly the asymptotics near the optimal solution $R_\infty = Q_\infty = T$ is governed by $d\mathbf{V}^{(2)}/d\alpha = \mathbf{m}^{(2)}\mathbf{V}^{(2)}$. Note that by construction the (2×2)–matrix $\mathbf{m}^{(2)}$ can be obtained from $\mathbf{m}^{(3)}$ by truncation of the last column and the last row. Both matrices can be diagonalized analytically for arbitrary parameters T and v. Denoting the eigenvalues of $\mathbf{m}^{(3)}$ by λ_i and those of $\mathbf{m}^{(2)}$ by $\tilde{\lambda}_i$ one gets

$$\begin{aligned} \lambda_1 &= \tilde{\lambda}_1 = \frac{4v^4\eta^2}{\pi^2}\left(\frac{K-1}{1+2T} + \frac{1}{\sqrt{1+4T}}\right) - \frac{4v^2\eta}{\pi\sqrt{1+2T}}, \\ \tilde{\lambda}_2 &= -\frac{2v^2\eta}{\pi(1+2T)^{(3/2)}}, \\ \lambda_{2,3} &= \frac{1}{2}\left(\tilde{\lambda}_2 + m_{33} \pm \sqrt{(\tilde{\lambda}_2 - m_{33})^2 + \frac{4m_{31}^2 T}{(1+T)^2}}\right), \end{aligned} \tag{10}$$

where $m_{33} = -(2\eta/\pi)\arcsin(T/(1+T))$, $m_{31} = 2\eta v/(\pi\sqrt{1+2T})$. Note that $\lambda_1 = \tilde{\lambda}_1$ is quadratic in η whereas the remaining eigenvalues depend linearly on η.

The asymptotics of $\mathbf{V}^{(3)}$ is governed by $\max(\lambda_1, \lambda_2)$ which depends on η, see fig. 1. For small η the asymptotic decay of $\mathbf{V}^{(3)}$ is proportional to $\exp(\lambda_2\alpha)$

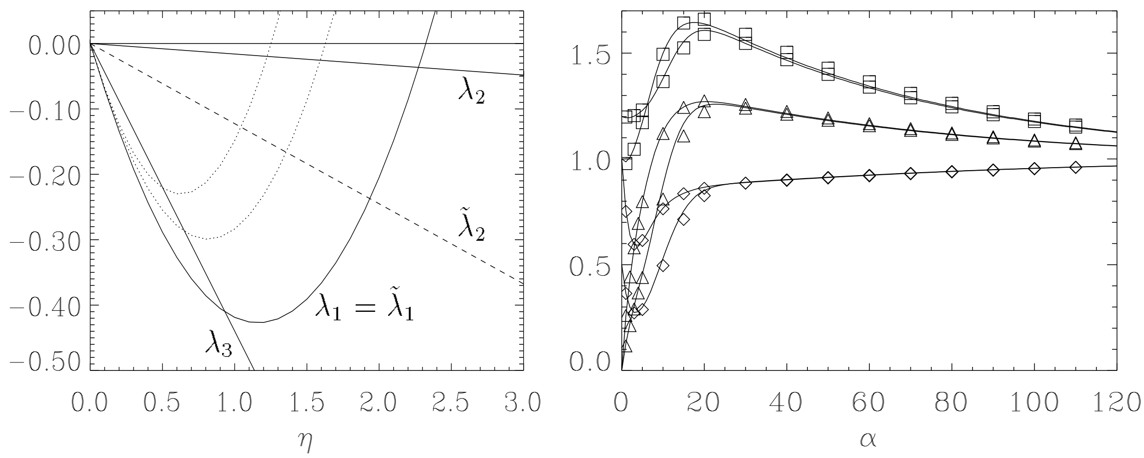

Figure 1: Learning in a tree-like architecture with two hidden units ($K = 2$) for parameters $T = v = 1$. Left: η–dependence of the eigenvalues of the linearized system (λ_i of $\mathbf{m}^{(3)}$, $\tilde{\lambda}_i$ of $\mathbf{m}^{(2)}$) governing the asymptotics of learning. The dotted lines show λ_1 for $K=3$ and $K=4$ respectively demonstrating the decrease of η_c with increasing K. Right: evolution of the dynamical variables for $\eta = 1$ ($\triangle$ R, $\square$ Q, $\diamond$ w). The symbols represent results obtained by a simulation of a system with $2N = 200$ input units averaged over 100 experiments. Errorbars would be smaller than the symbols. Note that the dynamical variables in different branches become equal rapidly in spite of the asymmetric initial conditions $R_{11}(0)$=$R_{22}(0)$=0, $Q_{11}(0)$=1.2, $Q_{22}(0)$=$w_1(0)$=1.0, and $w_2(0)$=0.5.

whereas $\mathbf{V}^{(3)} \propto \exp(\lambda_1 \alpha)$ for larger values of η. For η greater than the critical value

$$\eta_c = \frac{\pi / v^2}{\frac{K-1}{\sqrt{1+2T}} + \frac{\sqrt{1+2T}}{\sqrt{1+4T}}} \tag{11}$$

λ_1 becomes positive and the on–line backpropagation algorithm does not converge to the optimal solution. Therefore the range of learning rates that lead to perfect generalization is given by $\eta < \eta_c$. This range decreases with an increasing number of hidden units like $\eta_c \propto 1/K$ for large K. Because of $\lambda_1 = \tilde{\lambda}_1$ the same holds true for $\mathbf{V}^{(2)}$, i.e. for learning with predetermined hidden–to–output couplings [10].

In addition, the identity $\lambda_1 = \tilde{\lambda}_1$ implies a rather remarkable result : The critical learning rate is the same whether or not the hidden–to–output weights are adjustable. We therefore conclude that – at least for a finite number of hidden units K – the existence of a critical learning rate is a first layer effect.

In the regime of η small enough the asymptotics is governed by λ_2 and $\tilde{\lambda}_2$, respectively. Eq. (10) implies $\lambda_2 > \tilde{\lambda}_2$ for all values of T and v. An updating of the

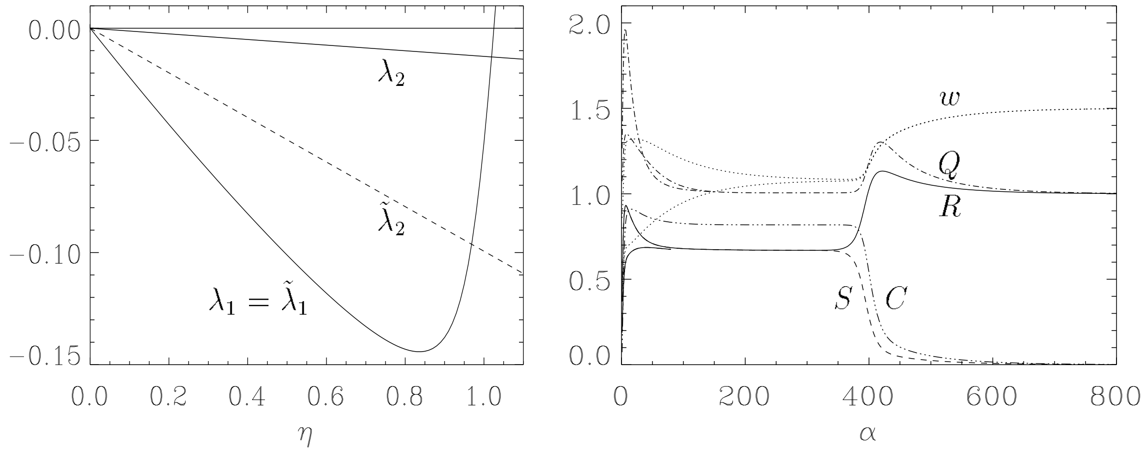

Figure 2: Learning in the fully connected architecture with two hidden units ($K = 2$) for parameters $T = 1$, $v = 1.5$. Left: eigenvalues that govern the asymptotics of the dynamical variables. Notation as in fig. 1. Right: numerical solution for the dynamical variables for $\eta = 1$ and asymmetric initial conditions $R(0) = S(0) = C(0) = 0, Q(0) = w_1(0) = 1, w_2(0) = 0.5$.

couplings in the second layer therefore leads to a slowing down of the asymptotic convergence. This is precisely what one would expect, since the adaptation of additional (hidden–to–output) couplings should decelerate the learning process. However, λ_2 and $\tilde{\lambda}_2$ are independent of K. This means that the asymptotics of the dynamical variables is independent of the number of hidden units in the small η–regime.

In order to obtain the asymptotic decay of ϵ_g, the generalization error has to be expanded up to second order in $\mathbf{V}^{(3)}$ ($\mathbf{V}^{(2)}$ respectively). As in [8] we find that the generalization error decays proportional to $\exp(2\lambda_2\alpha)$ for small η ($\exp(2\tilde{\lambda}_2\alpha)$ if the couplings in the second layer are fixed), while $\epsilon_g \propto \exp(\lambda_1\alpha)$ for large values of η. The change of the asymptotic decay occurs at η_{opt} defined by $\lambda_1 = 2\lambda_2$ ($\tilde{\eta}_{opt}$ by $\lambda_1 = 2\tilde{\lambda}_2$). With this learning rate the fastest asymptotic decrease of the generalization error can be achieved. Note that η_{opt} is much closer to η_c than $\tilde{\eta}_{opt}$, as can be seen in fig. 1 (for $v = T = 1$ we obtain $\tilde{\eta}_{opt} = (2/3)\eta_c$ [8, 11], whereas $\eta_{opt} \approx 0.96\eta_c$). Therefore it will be very difficult to tune the learning rate optimally in practical applications of on–line backpropagation.

Fig. 1 also depicts the numerical solution of the full system of the equations of motion (7) together with simulations.

b) *fully connected architecture:* Since each hidden node receives information from all input units, a permutation symmetry is inherent in the problem, i.e. the i–th branch in the student network does not necessarily specialize on the

i–th branch in the teacher network. Without loss of generality we relabel the dynamical variables such as if this were indeed the case.

The analysis proceeds similarly to the tree–like architecture, except for the fact that the asymptotics has to be described by five dynamical variables R, S, Q, C, and w. A linearization for small deviations from the optimal solution $R_\infty = Q_\infty = T, S_\infty = C_\infty = 0, w_\infty = v$ leads to $d\mathbf{V}^{(5)}/d\alpha = \mathbf{m}^{(5)}\mathbf{V}^{(5)}$, where $\mathbf{V}^{(5)} = (R-T, S, Q-T, C, w-v)^\top$.

Again, we also consider fixed hidden–to–output couplings $w \equiv v$. This special case has been investigated recently in [10, 11]. The linearization reads $d\mathbf{V}^{(4)}/d\alpha = \mathbf{m}^{(4)}\mathbf{V}^{(4)}$, where $\mathbf{V}^{(4)} = (R-T, S, Q-T, C)^\top$. As before, $\mathbf{m}^{(4)}$ can be obtained from $\mathbf{m}^{(5)}$ by truncation of the last column and the last line.

The eigenvalues λ_i of $\mathbf{m}^{(5)}$ and $\tilde{\lambda}_i$ of $\mathbf{m}^{(4)}$ can be computed analytically for arbitrary values of T and v. However, the expressions are rather lengthy, even for a particular choice of T and v. We therefore only discuss the important features of the asymptotics, the exact expressions for the eigenvalues will be presented in [12].

The asymptotics is governed by the largest eigenvalue which we denote by λ_1 and λ_2 for $\mathbf{m}^{(5)}$ and by $\tilde{\lambda}_1, \tilde{\lambda}_2$ for $\mathbf{m}^{(4)}$. The η-dependence of the dominating eigenvalue is qualitatively the same as for the tree–like architecture: The eigenvalues $\lambda_2, \tilde{\lambda}_2$ are linear in η, whereas $\lambda_1, \tilde{\lambda}_1$ depend nonmonotonically on η, cf. fig. 2. In contrast to ref. [11] we do not observe that $\tilde{\lambda}_1$ is polynomial in η. Fig. 2 shows the eigenvalues for a particular set of parameters.

The nonmonotonic $\lambda_1, \tilde{\lambda}_1$ give rise to a critical learning rate η_c, such that for $\eta \geq \eta_c$ no perfect generalization can be achieved. Moreover, $\lambda_1 = \tilde{\lambda}_1$ holds true as before, implying that the value of η_c is independent of an adaptation of couplings in the hidden–to–output layer. Even if one had an *a priori* knowledge of v_m the maximal learning rate η_c would be the same.

Again, we find $\lambda_2 > \tilde{\lambda}_2$ for all values of T, v. The additional updating of the second layer makes the asymptotic convergence much slower. As can be seen in fig. 2 λ_2 is very close to zero. In contrast to the tree–like architecture both λ_2 and $\tilde{\lambda}_2$ are K–dependent.

The asymptotics of the generalization error is as in the nonoverlapping case. However, here the optimal learning rate η_{opt} is very close to η_c due to the nonomonotonicity of $\lambda_1 = \tilde{\lambda}_1$ and the K–dependence of λ_2 and $\tilde{\lambda}_2$.

Fig. 2 shows the numerical solution of the equations of motion (7) for the fully connected architecture. Note the plateaus at intermediate values of α, which result from the existence of symmetric fixed points of (7). A complete discussion of this effect can be found in [10, 11], the extension to adjustable $\{w_i\}$ will be given in [12].

In summary we have presented an exactly solvable model for the training of multi–layer neural networks by on–line backpropagation of error. As a specific example we have discussed two–layered networks with a single linear output unit and adjustable hidden–to–output weights.

In this letter we have restricted the discussion to learnable rules defined through symmetric teacher networks with K hidden units. As an obvious extension we will furthermore investigate the learning with mismatched student architectures ($K \neq M$), e.g. the case of an unlearnable rule.

The analysis of the asymptotic learning curves yields the same critical rates $\eta_c(K)$ as in the simpler case of an *a priori* known hidden–to–output relation. For $\eta \leq \eta_{opt}$ (with η_{opt} rather close to η_c), however, the asymptotic decrease of ϵ_g is much slower because of the required adaptation of the additional weights. Due to the increased flexibility of the network it is capable of learning more complex rules, but this ability is acquired at the cost of higher computational effort.

So far we have chosen the same learning rate $\tilde{\eta} = \eta/N$ everywhere in the network. The extension to a different rate η_w in the second layer is straightforward as long as η_w remains of order $\mathcal{O}(1/N)$, otherwise the description in terms of the mean values of $\{w_i\}$ is insufficient. It remains an open problem how to analyse for example the practically relevant case of $\eta_w \propto \mathcal{O}(1/K)$ [1] in a similar fashion.

Acknowledgements: We thank D. Saad and S. A. Solla for providing ref. [11] to us prior to publication. We are grateful to G. Reents and A. Scharnagl for a critical reading of the manuscript. P. Riegler was supported by the Deutsche Forschungsgemeinschaft.

References

[1] J.A. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation*, Addison–Wesley (Redwood City, CA), 1991.

[2] S. Seung, H. Sompolinksy, and N. Tishby, Phys.Rev. A **45**, 6056 (1992).

[3] T.L.H. Watkin, A. Rau, and M. Biehl, Rev.Mod.Phys. **65**, 499 (1993).

[4] M. Opper and W. Kinzel, in: *Physics of Neural Networks III*, eds. E. Domany, J.L. van Hemmen, and K. Schulten, Springer (Berlin), in press.

[5] Y. Chauvin and D.E. Rumelhart (eds.), *Backpropagation: Theory, Architectures, and Applications*, Erlbaum (Hillsdale, NJ), 1995.

[6] S. Amari, IEEE Trans. Elect. Comput. **EC–16**, 299 (1967) and Neurocomp. **5**, 185 (1993).

[7] T. Heskes and B. Kappen, in *Mathematical Foundations of Neural Networks,* ed. J.G. Taylor, Elsevier (Amsterdam), 1993.

[8] M. Biehl and H. Schwarze, J.Phys. A **28**, 643 (1995).

[9] M. Copelli and N. Caticha, J.Phys. A **28**, 1615 (1995).

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

[11] D. Saad and S. A. Solla, Phys.Rev. E, in press (1995).

[12] P. Riegler and M. Biehl, in preparation.