Learning by Online Gradient Descent

Michael Biehl*
CONNECT, The Niels Bohr Institute
Blegdamsvej 17, 2100 Copenhagen Ø, Denmark

Holm Schwarze†
Department of Theoretical Physics, Lund University
Sölvegatan 14 A, 223 62 Lund, Sweden

Abstract

We study online gradient–descent learning in multilayer networks analytically and numerically. The training is based on randomly drawn inputs and their corresponding outputs as defined by a target rule. In the thermodynamic limit we derive deterministic differential equations for the order parameters of the problem which allow an exact calculation of the evolution of the generalization error. First we consider a single–layer perceptron with sigmoidal activation function learning a target rule defined by a network of the same architecture. For this model the generalization error decays exponentially with the number of training examples if the learning rate is sufficiently small. However, if the learning rate is increased above a critical value, perfect learning is no longer possible. For architectures with hidden layers and fixed hidden–to–output weights, such as the parity and the committee machine, we find additional effects related to the existence of symmetries in these problems.

PACS. 87.10, 02.50, 05.90

1 Introduction

Neural Networks [1, 2] can realize a classification scheme: they assign an output value to any possible input, defined by the architecture of the net, the activation functions of its units, and the actual set of network parameters or weights.

*permanent address: Institut für theoretische Physik, Universität Würzburg, Am Hubland, D–97074 Würzburg, Germany email: biehl@physik.uni-wuerzburg.de
†email: holm@thep.lu.se

The ability of such systems to learn a rule by choosing suitable weights has been studied successfully using statistical mechanics [3, 4, 5]. Mostly, the training process is interpreted as a stochastic minimization of an energy function defined in weight space, which measures the performance of the student network on a given set of examples. The term generalization is used for the student's ability to infer an unknown input/output relation from the examples and apply it to novel input data.

Statistical physics provides the tools for investigating typical properties of the equilibrium solution to this optimization problem by performing the average over random example inputs in the limit of infinite dimensionality.

The most thoroughly studied model in this context is the so–called simple perceptron, a single binary threshold unit which realizes a linearly separable classification [6, 7]. Convergence of the training process can be guaranteed for deterministic learning algorithms which yield good generalization of a linearly separable rule [3, 4, 5].

However, the learning scheme by far most commonly used in practice is gradient descent learning in multilayered networks of continuous units. It is the basis of the well known 'backpropagation of error' [8–11] and its modifications (see e.g. [1]). The objective function is very often simply the quadratic deviation of the student output from the correct one, summed over all training examples.

In offline or batch learning the evolution of the weights follows the direction of steepest descent in an energy landscape defined for a set of input/output pairs. In this paper we consider, however, learning by *online* gradient descent. The change of weights is given by the gradient of the error evaluated for only the latest in a sequence of examples. The performance on previous examples is not taken into account, and no explicit storage of a training set is necessary. Recently, the properties of online gradient–descent learning have been studied in the context of master equations for stochastic dynamics, which, in the limit of small learning rates, can be approximated by a Fokker–Planck equation (e.g. [12–15]).

Here, we study the dynamics of online learning for specific models of two–layer networks in a well–defined thermodynamic limit. This leads us to deterministic differential equations for the order parameters of the problem, which can be solved exactly. Assuming that all training inputs are drawn independently from the same distribution we study the generalization ability for different types of student networks and rules to be learned.

We introduce online learning formally in the next section. Mainly for explaining the method of analysis, we discuss —as a first example— a single neuron with a continuous sigmoidal activation function. The unknown rule is represented by a single unit of the same type (the *teacher*) but with an unknown weight vector. Thus, the problem is learnable, and we study how the generalization error decreases to zero as more and more examples have been used for training.

In our second example (sec. 4) the rule is still defined by a single neuron, but the student network consists of two hidden units with a fixed linear hidden–

to–output relation. The coefficients of the latter determine whether the rule is indeed learnable for the actual student and various scenarios can be modeled.

In section 5 we consider a student network with 2 hidden units, whose output is defined to be the product of their respective states. The rule to be learned is represented by a network of the same structure.

A summary and discussion of the results is given in the last section where we conclude with an outlook on further applications of the method.

2 The model

Consider a student network with continuous output $\sigma(\underline{J},\underline{\xi})$ where $\underline{\xi}$ is an N–dimensional input vector and $\underline{J}$ is the set of all variable weights in the net.

The desired output $\tau(\underline{\xi})$ is defined by a target rule, and the error

$$\varepsilon(\underline{J},\underline{\xi}) = \frac{1}{2}\Big[\sigma(\underline{J},\underline{\xi}) - \tau(\underline{\xi})\Big]^2 \tag{1}$$

measures the deviation of the student from the rule for a particular input $\underline{\xi}$.

The generalization error of a student with weights $\underline{J}$ is defined as

$$\varepsilon_g(\underline{J}) = \Big\langle \varepsilon(\underline{J},\underline{\xi}) \Big\rangle_{\underline{\xi}} \tag{2}$$

where $\langle \ldots \rangle_{\underline{\xi}}$ denotes the average over the distribution of inputs. In the following we consider independently drawn input vectors with uncorrelated random components of zero mean and unit variance.

At each learning step μ, a new uncorrelated vector $\underline{\xi}^\mu$ is presented, and the current weight vector $\underline{J}^\mu$ is updated according to the gradient of $\varepsilon(\underline{J}^\mu,\underline{\xi}^\mu)$ with respect to the weights:

$$\begin{aligned} \underline{J}^{\mu+1} &= \underline{J}^\mu - \frac{\eta}{N}\nabla_{\underline{J}}\,\varepsilon(\underline{J}^\mu,\underline{\xi}^\mu) \\ &= \underline{J}^\mu - \frac{\eta}{N}\Big[\sigma(\underline{J}^\mu,\underline{\xi}^\mu) - \tau(\underline{\xi}^\mu)\Big]\,\nabla_{\underline{J}}\,\sigma(\underline{J}^\mu,\underline{\xi}^\mu) \end{aligned} \tag{3}$$

Here, η is the so–called learning rate which has been explicitly scaled with the network size N. It controls the size of the steps made in the direction of steepest descent. Note, that the architecture of the student net and the activation functions of its units determine the actual form of the gradient term.

For the specific examples considered in the following, it is possible to derive from (3) recursion relations for order parameters, which in turn determine the student's properties completely in the thermodynamic limit $N \to \infty$. In the same limit we can interprete $\alpha = \mu/N$ as a continuous 'time' and solve the corresponding differential equations for the order parameters numerically. Thus, the evolution of the generalization ability in the above online learning process is obtained.

3 A single unit with continuous output

As a simple example of online gradient–descent learning we first consider a graded–response perceptron [16, 17] whose output is given by

$$\sigma(\underline{J}, \underline{\xi}) = g(\underline{J} \cdot \underline{\xi}) \tag{4}$$

with a nonlinear, differentiable activation function g. A standard choice for g is $g(x) = \tanh(x)$ because of its property $g'(x) = 1 - g^2(x)$. However, in order to simplify the analytic treatment of our model, it is more convenient to use the function $g(x) = \mathrm{erf}(x/\sqrt{2}) = \int_{-x}^{x} dt\, e^{-t^2/2}/\sqrt{2\pi}$ instead. Both functions are very similar in shape and we do not expect our results to depend critically upon this choice. This network is trained from a stream of examples $(\underline{\xi}^\mu, \tau_\mu)$ whose outputs τ_μ are defined by a target perceptron

$$\tau_\mu = \tau(\underline{\xi}^\mu) = g(\underline{B} \cdot \underline{\xi}^\mu) \tag{5}$$

where $\underline{B}$ is the unknown teacher weight vector, and $||\underline{B}|| = 1$. For this model the gradient descent learning rule (3) reads

$$\underline{J}^{\mu+1} = \underline{J}^\mu + \frac{\eta}{N}[g(y_\mu) - g(x_\mu)]\, g'(x_\mu)\, \underline{\xi}^\mu \tag{6}$$

with $g'(x) = \sqrt{2/\pi}\, e^{-x^2/2}$ and the abbreviations $x_\mu = \underline{J}^\mu \cdot \underline{\xi}^\mu$ and $y_\mu = \underline{B} \cdot \underline{\xi}^\mu$ for the internal fields (or net inputs) in the student and teacher network, respectively. Note that (6) formally resembles the Hebb–rule [18, 19] for the effective target outputs

$$\delta^\mu = [g(y_\mu) - g(x_\mu)]\, g'(x_\mu). \tag{7}$$

In the backpropagation algorithm, these δ^μ's play the role of the backpropagated errors.

The generalization error for this model can be calculated in a straightforward manner and expressed as a function of the overlap of the student with the teacher weight vector $R = \underline{J} \cdot \underline{B}$ and the norm of the student weight vector $Q = \sqrt{\underline{J} \cdot \underline{J}}$. If the inputs ξ_i are drawn independently from a common distribution with zero mean and unit variance, the average over inputs in (2) leads to

$$\varepsilon_g(R, Q) = \frac{1}{\pi} \arcsin\left(\frac{Q^2}{1+Q^2}\right) - \frac{2}{\pi} \arcsin\left(\frac{R}{\sqrt{2(1+Q^2)}}\right) + \frac{1}{6}. \tag{8}$$

In contrast to a simple threshold unit the generalization error of the graded–response unit explicitly depends on the length of the student vector. It vanishes only if the student vector is perfectly aligned with the teacher ($R = Q$) and has the same length ($Q = 1$).

In order to calculate the generalization error (8) at time step μ we need to compute the overlaps $R^\mu = \underline{J}^\mu \cdot \underline{B}$ and $[Q^2]^\mu = \underline{J}^\mu \cdot \underline{J}^\mu$. From the learning rule (6) using (7) we obtain the difference equations

$$\begin{aligned} R^{\mu+1} - R^\mu &= \frac{\eta}{N}\delta^\mu\, y_\mu \\ \left[Q^2\right]^{\mu+1} - \left[Q^2\right]^\mu &= \frac{2\eta}{N}\delta^\mu\, x_\mu + \frac{\eta^2}{N}\,\delta_\mu^2. \end{aligned} \tag{9}$$

These equations can be averaged over the current training input noting that the dependence on the inputs is only through the internal fields x_μ and y_μ. In the limit $N \to \infty$ these are correlated Gaussian variables with zero mean and covariances $< x_\mu^2 >= Q^2$, $< y_\mu^2 >= 1$ and $< x_\mu\, y_\mu >= R$. In the same limit, we can introduce a continuous 'time' $\alpha = \mu/N$ and rewrite (9) as differential equations

$$\begin{aligned} \frac{dR}{d\alpha} &= \eta\,\langle \delta\, y\rangle \\ \frac{d[Q^2]}{d\alpha} &= 2\eta\,\langle \delta\, x\rangle + \eta^2\big\langle \delta^2\big\rangle \end{aligned} \tag{10}$$

where $\langle \ldots \rangle$ denotes an average over the joint distribution of x and y, and where we have suppressed the index μ. For the choice $g(x) = \mathrm{erf}(x/\sqrt{2})$ the averages in (10) can be performed analytically, leading to

$$\begin{aligned} \frac{dR}{d\alpha} &= \frac{2}{\pi}\frac{\eta}{1+Q^2}\left[\frac{1+Q^2-R^2}{\sqrt{2(1+Q^2)-R^2}} - \frac{R}{\sqrt{1+2Q^2}}\right] \\ \frac{d[Q^2]}{d\alpha} &= \frac{4}{\pi}\frac{\eta}{1+Q^2}\left[\frac{R}{\sqrt{2(1+Q^2)-R^2}} - \frac{Q^2}{\sqrt{1+2Q^2}}\right] \\ &+ \frac{4}{\pi^2}\frac{\eta^2}{\sqrt{1+2Q^2}}\left[\arcsin\left(\frac{Q^2}{1+3Q^2}\right) + \arcsin\left(\frac{1+2(Q^2-R^2)}{2(1+2Q^2-R^2)}\right)\right. \\ &- \left. 2\arcsin\left(\frac{R}{\sqrt{2(1+2Q^2-R^2)}\sqrt{1+3Q^2}}\right)\right]. \end{aligned} \tag{11}$$

From the numerical solution of these equations we finally obtain the time evolution of the generalization error (8).

3.1 The role of the learning rate

Figure 1 shows the evolution of the generalization error of the graded–response perceptron for different choices of the learning rate η.

For small learning rates the generalization error smoothly decreases with increasing α and approaches the optimal value $\varepsilon_g = 0$. The speed of this approach

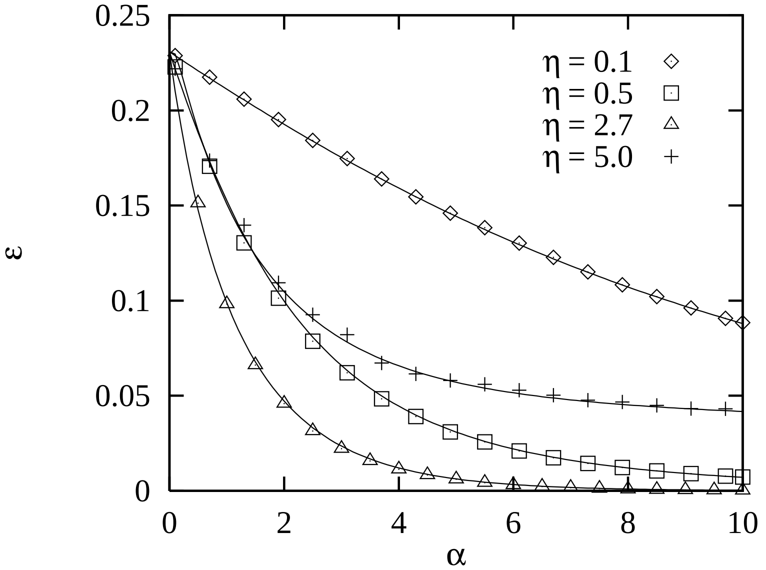

Figure 1: Generalization error of the graded–response perceptron for different learning rates. The analytic results (solid lines) are compared to simulations (symbols) for a network with $N = 100$ weights (standard error bars would be approximately the size of the symbols). All curves are for initial conditions $R(0) = 0$ and $Q(0) = 0.5$.

can be controlled by varying η. If η is chosen too large, the learning process slows down until a critical learning rate $\eta_c \approx 4.06$ is reached. For $\eta > \eta_c$ the generalization error does not decay to zero any longer but approaches a value $\varepsilon_g(\alpha \to \infty) > 0$. We have performed simulations using a network with $N = 100$ weights and found a very good agreement with our analytic results (see figure 1).

We can investigate the asymptotic behavior of the generalization error in greater detail by linearizing (11) around its fixed points given by $dR/d\alpha = d[Q^2]/d\alpha = 0$. As can be easily verified, (11) has a fixed point at $(R, Q) = (1, 1)$ for all values of η. Linearizing around this point yields a system of linear differential equations for the deviations $r = 1 - R$ and $q = 1 - Q$, given by

$$\begin{pmatrix} r' \\ q' \end{pmatrix} = \underline{\underline{A}} \begin{pmatrix} r \\ q \end{pmatrix}, \quad \text{with} \quad \underline{\underline{A}} = \frac{4}{3}\sqrt{5}\,\frac{\eta}{\eta_c} \begin{pmatrix} -\frac{2}{3} & \frac{1}{2} \\ \frac{1}{3} - \frac{\eta}{\eta_c} & \frac{\eta}{\eta_c} - \frac{1}{2} \end{pmatrix} \tag{12}$$

with $\eta_c = \sqrt{5/3}\,\pi \approx 4.06$. The eigenvalues of $\underline{\underline{A}}$ are $\lambda_1 = 4\frac{\sqrt{5}}{3}\frac{\eta}{\eta_c}(\frac{\eta}{\eta_c} - 1)$ and $\lambda_2 = -2\sqrt{5}\,\eta/(9\eta_c)$ (see figure 2). Therefore, for subcritical values of η the parameters R and Q approach their optimal value exponentially fast, with $r, q \propto e^{\lambda(\eta)\,\alpha}$, where $\lambda(\eta) = \max(\lambda_1, \lambda_2)$. Note that as $\eta \to \eta_c$ the relaxation time $-1/\lambda_1$ diverges like $(\eta_c - \eta)^{-1}$ (*critical slowing down*). For $\eta > \eta_c$, one of the two eigenvalues becomes positive and $(R, Q) = (1, 1)$ is not an attractive fixed point any longer. However, in this regime we can numerically find a second fixed point of (11) with $R, Q \neq 1$. Hence, perfect learning is not possible and $\varepsilon_g(\alpha \to \infty) > 0$.

The eigenvalues of the linearized system governing the approach to this sub-optimal fixed point are also shown in figure 2. Note, that for $4.45 \lesssim \eta \lesssim 5.05$ the eigenvalues have imaginary parts corresponding to oscillations around the fixed points. However, these oscillations are strongly damped due to the larger real parts of the eigenvalues.

If the learning rate is greater than $\eta_d = \pi / \arcsin(1/3) \approx 9.24$, no fixed point exists and $R, Q \to \infty$ as $\alpha \to \infty$.

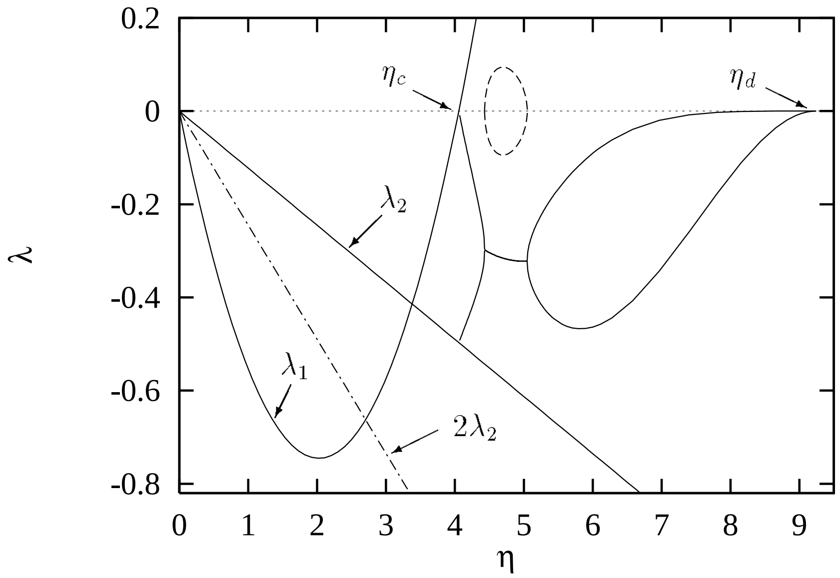

Figure 2: Eigenvalues of the linearized system (12) governing the asymptotic behavior of the generalization error of the single graded response unit. The dashed lines correspond to the imaginary parts of the eigenvalues. The dashed–dotted line represents $2\lambda_2$.

In order to find the learning rate η_{opt}, which yields the fastest asymptotic decay of the generalization error, we expand (8) to second order in r and q

$$\varepsilon_g \approx \frac{2}{\sqrt{3}} \frac{1}{\pi}(r - q) - \frac{1}{3\pi\sqrt{3}} \left[(r - q)^2 + q^2 \right]. \tag{13}$$

Since the eigenvector corresponding to the eigenvalue λ_2 is $(1, 1)^\top$, this mode cannot contribute to the asymptotic behavior of the linear combination $(r - q)$. Hence, for small η, $(r - q)$ decays faster than r and q, and the behavior of ε_g

depends on the actual order of $(r-q)$ compared to the quadratic terms. If, for given η, q^2 is larger than $(r-q)$, the generalization error decays proportional to $e^{2\lambda_2\alpha}$, while we have $\varepsilon_g \propto e^{\lambda_1\alpha}$ if $(r-q)$ is larger than q^2. This change of the asymptotic behavior happens at $\eta_{\rm opt} = 2\eta_c/3 \approx 2.704$, where λ_1 and $2\lambda_2$ coincide (see figure 2). Therefore, the fastest asymptotic decay of ε_g is achieved for $\eta_{\rm opt}$. However, it should be noted that this discussion only holds for the *asymptotic* behavior. The initial decay of the generalization error is faster for values of η different from $\eta_{\rm opt}$, and an online adjustment of the learning rate would be desirable in order to optimize the network behavior [20].

Even though learning a simple perceptron with another perceptron is a learnable task, this simple example illustrates the importance of a proper choice of the learning rate in an online gradient–descent scheme. The convergence becomes slow if the learning rate differs from the optimal one, and a large learning rate causes a failure to converge to the optimal solution. This result is in agreement with the behavior observed by many authors for the backpropagation algorithm. Furthermore, the step–size dependence of offline gradient–descent minimization schemes is well known. In one dimension Newton's method utilizes the fact that the optimal step size for an iterative minimization of a quadratic cost function by gradient descent is $1/(2\mu)$, where μ is the second derivative of the cost function. If the step size is larger than twice this value the procedure does not converge. In higher dimensions, minimizing a quadratic error surface corresponds to batch learning in a single linear unit. In this case, the largest step size which guarantees convergence is the inverse of the largest eigenvalue of the Hessian (see e.g. [22, 23]). Note, however, that these considerations do not apply directly to online–learning, where the gradient of the training error is calculated with respect to one example only.

4 The Soft–Committee Machine

After having illustrated the basic features of online gradient–descent learning in a simple perceptron we now turn to networks with hidden units. As an example for this situation we consider a fully connected two–layer network with 2 hidden units of the type described above and with the hidden–to–output weights fixed to +1. The overall output of this machine is given by

$$\sigma\Big(\underline{J}_1, \underline{J}_2, \underline{\xi}\Big) = g_o\Big[g\Big(\underline{J}_1 \cdot \underline{\xi}\Big) + g\Big(\underline{J}_2 \cdot \underline{\xi}\Big)\Big], \tag{14}$$

where we choose a linear output unit with $g_o(x) = \beta x$, $\beta > 0$. Hence, the overall output is proportional to the average 'decision' of the hidden units. This network is trained to implement a simple task defined by a teacher perceptron as in (5). Note that this is a learnable task (i.e. $\varepsilon_g^{\rm opt} = 0$) only for the choices $\beta = 1/2$ and $\beta = 1$, and an unrealizable rule for all other values of β. For $\beta = 1/2$ the optimal generalization error is achieved for a symmetric student network with

$\underline{J}_1 = \underline{J}_2 = \underline{B}$, while for $\beta = 1$ the optimal student is a specialized one with $\underline{J}_1 = \underline{B}$ and $||\underline{J}_2|| = 0$ or vice versa. Therefore, this simple model includes a variety of possible learning scenarios.
Similar to the single unit, the generalization error for this model can be expressed in terms of the relevant overlaps $R_l = \underline{J}_l \cdot \underline{B}$, $Q_l = \sqrt{\underline{J}_l \cdot \underline{J}_l}$ and $C = \underline{J}_1 \cdot \underline{J}_2$ $(l = 1, 2)$. We obtain

$$\begin{aligned}
\varepsilon_g(R_1, R_2, Q_1, Q_2, C) &= \sum_{l=1}^{2} \left[\frac{\beta^2}{\pi} \arcsin\left(\frac{Q_l^2}{1+Q_l^2}\right) - \frac{2\beta}{\pi} \arcsin\left(\frac{R_l}{\sqrt{2(1+Q_l^2)}}\right) \right] \\
&+ \frac{2\beta^2}{\pi} \arcsin\left(\frac{C}{\sqrt{1+Q_1^2}\sqrt{1+Q_2^2}}\right) + \frac{1}{6}. \qquad (15)
\end{aligned}$$

Again, from the learning rule (3) we obtain difference equations for the relevant parameters, which in the limit $N \to \infty$ can be written as the differential equations:

$$\begin{aligned}
\frac{dR_l}{d\alpha} &= \eta \left\langle \delta_l \, y \right\rangle \\
\frac{d[Q_l^2]}{d\alpha} &= 2\eta \left\langle \delta_l \, x_l \right\rangle + \eta^2 \left\langle \delta_l^2 \right\rangle \\
\frac{d[C]}{d\alpha} &= \eta \left\langle \delta_1 \, x_2 + \delta_2 \, x_1 \right\rangle + \eta^2 \left\langle \delta_1 \, \delta_2 \right\rangle \qquad (16)
\end{aligned}$$

where the averages are over the joint distribution of the x_l's and y_l's, and with $\delta_l = [g(y_l) - g(x_l)]\, g'(x_l)$ analogous to (7) for the individual hidden units. All the averages can be performed analytically (see appendix A), and from the numerical solution of the resulting set of five coupled differential equations we get the time evolution of the generalization error (15) and its asymptotic value.

First we will discuss the case $\beta = 1$. It can easily be seen from (27)–(29) that the network will always evolve symmetrically with $R_1 = R_2$ and $Q_1 = Q_2$ if it is started from symmetric initial conditions. Only if the initial conditions break this symmetry, can the system leave the symmetric subspace and approach the optimal solution. Figure 3 shows the evolution of the student–teacher overlaps $R_{1,2}$ for two different choices of initial conditions.

Initially, the overlaps rapidly approach values close to a fixed point which is stable within the symmetric subspace. However, due to the non–symmetric initial conditions the system does not evolve within this subspace, and eventually the repulsive mode takes over. For large values of α, the system approaches the optimal fixed point which breaks the symmetry between hidden units. This delayed repulsion is due to the fact that the corresponding positive transverse eigenvalue of the linearized system is very small compared to the absolute values of the negative longitudinal ones. As can be seen from figure 3, the number of learning steps necessary to escape from the symmetric fixed point is very sensitive

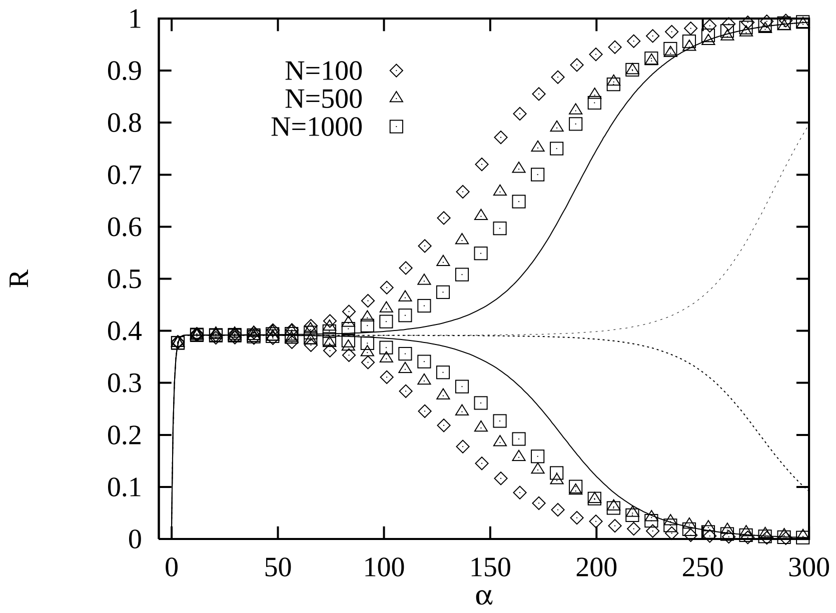

Figure 3: Evolution of the parameters R_1 and R_2 of the soft–committee machine with $\beta = 1$ and learning rate $\eta = 1$. The solid line represents the solution of the differential equations for nonsymmetric initial conditions ($R_1 = R_2 = Q_1 = C = 0$ and $Q_2 = 0.1$). The symbols show the results of simulations for different system sizes. The dotted lines correspond to the analytic solution for the initial conditions $R_1 = R_2 = Q_1 = C = 0$ and $Q_2 = 0.01$.

to the degree of asymmetry in the initial conditions. For the same reason we find strong finite size effects in our simulations in the region, where the specialization of hidden units occurs.

In order to study the η– and β–dependence of the network behavior we have numerically looked for stationary solutions of the differential equations and computed the eigenvalues of the corresponding linearized system; the results are shown in figure 4.

For fixed $\beta = 1$ (figure 4a) and small $\eta < \eta_c \approx 1.89$ the network behavior is qualitatively similar to the situation at $\eta = 1$. The initial generalization behavior is dominated by a repulsive symmetric fixed point while for large training sets the network approaches the optimal nonsymmetric solution. A careful numerical evaluation of the fixed–point equations shows the existence of a further symmetric fixed point. However, this fixed point is repulsive even within the symmetric subspace and does not influence the network performance. As in the single unit there is a critical learning rate $\eta_c \approx 1.89$, above which the optimal fixed point becomes unstable and the network approaches a suboptimal but still nonsymmetric solution. If the learning rate is increased above $\eta_s \approx 2.17$, the network does not realize the necessity of a specialization any longer, and the symmetric fixed point

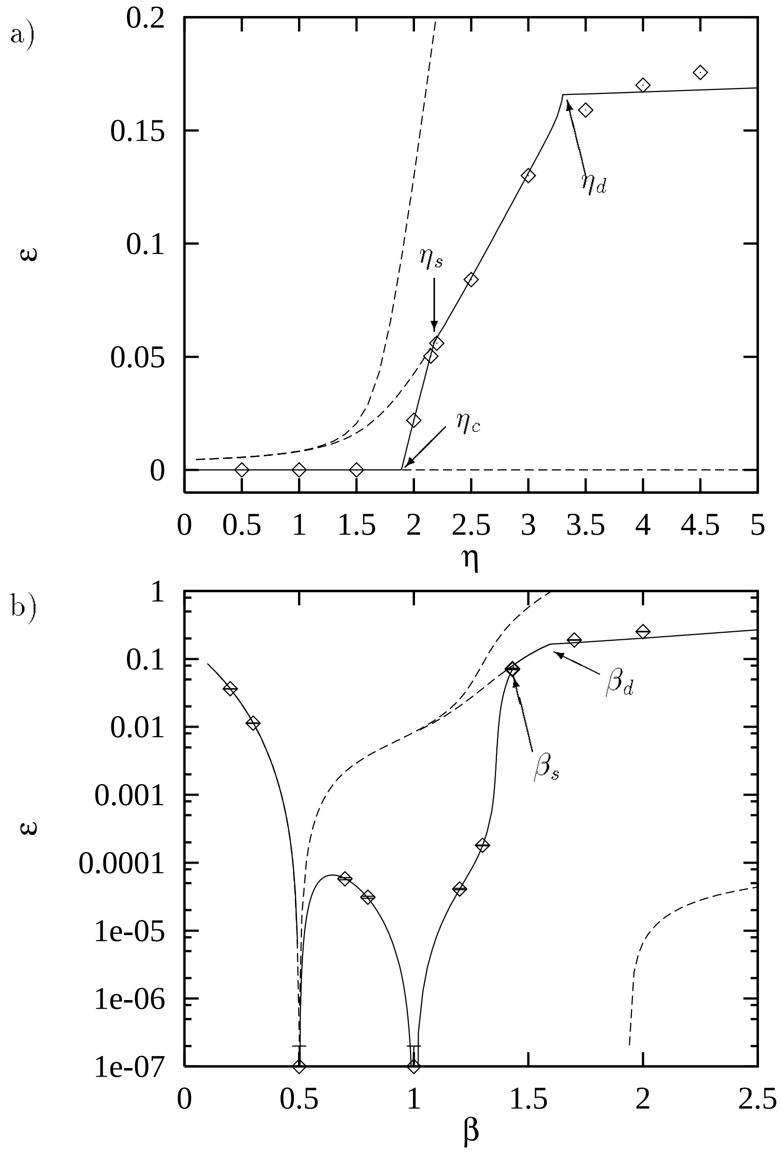

Figure 4: Asymptotic generalization error of the soft–committee a) as a function of the learning rate η for fixed $\beta = 1$ and b) as a function of β for fixed $\eta = 1$. Solid lines indicate attractive fixed points while dashed lines indicate repulsive fixed points. The symbols correspond to estimates of the asymptotic generalization error extrapolated from simulations (if not included, standard error bars would be smaller than the size of the symbols).

becomes the stable one. A further increase of the learning rate above $\eta_d \approx 3.29$ causes the norms of the student weight vectors to diverge, and the algorithm even fails to find an approximative solution. Finally we note, that in contrast to the single unit, the speed of both the escape from the symmetric region and the approach to the optimal solution below η_c are influenced by the choice of the learning rate. Therefore, tuning the learning rate to obtain the optimal behavior is a more difficult task than in the single unit.

Figure 4b shows the asymptotic properties of the soft–committee as a function of the gain parameter β in the linear output unit with $\eta = 1$ fixed. For small $\beta \leq 0.5$ the network approaches a stable symmetric fixed point. The corresponding residual error depends on the choice of β: For $\beta = 1/2$ the rule is realizable for the student network ($\varepsilon_g(\alpha \to \infty) = 0$) while it is unrealizable for $\beta < 1/2$. Again, we find a second symmetric fixed point, which is repulsive for all values of β and in general does not influence the dynamical evolution. Its generalization error is very close to that of the stable fixed point, closer than the resolution in figure 4b. In the region $1/2 < \beta < \beta_s \approx 1.44$ the fixed point structure is again similar to the one described for $\eta = \beta = 1$: A stable nonsymmetric fixed point with $R_1 \neq R_2$ and a pair of repulsive symmetric fixed points, one of which is attractive within the symmetric subspace and influences the initial behavior of the training process for a wide range of initial conditions. The residual error vanishes only at $\beta = 1$ and sharply increases even for a slightly mismatched gain parameter. For large values of β we find a behavior similar to the large–η regime: In the region $1.44 \lesssim \beta < \beta_d \approx 1.59$ one of the symmetric fixed points becomes the stable one and for $\beta \geq \beta_d$ the norm of the student vector diverges while R remains bounded.

5 The Soft–Parity Machine

Here we consider a student–network with two hidden units again, but with nonoverlapping receptive fields: each of the units is connected to only half of the input nodes. The output of the net is taken to be the product of their respective states, as an example for a nonlinear hidden–to–output relation:

$$\sigma(\underline{J}_1, \underline{J}_2, \underline{\xi}) = g(\underline{J}_1 \cdot \underline{\xi}_1)\, g(\underline{J}_2 \cdot \underline{\xi}_2) \quad \text{with} \quad g(x) = \mathrm{erf}(x/\sqrt{2}). \tag{17}$$

Here $\underline{J}_{1,2} \in I\!R^N$, and we consider the total input to be $2N$–dimensional, consisting of uncorrelated N–dimensional vectors $\underline{\xi}_{1,2}$. Thus, the normalization of the vectors and the definition of overlaps are formally the same as in the above cases.

The error is calculated with respect to a teacher of the same structure, with unknown, normalized $\underline{B}_{1,2}$. The generalization error is determined through the order parameters $R_l = \underline{B}_l \cdot \underline{J}_l$ and $Q_l = \sqrt{\underline{J}_l \cdot \underline{J}_l}$ as

$$\begin{aligned}\varepsilon_g(R_1,R_2,Q_1,Q_2) &= \frac{1}{9} + \frac{4}{\pi^2}\arcsin\left(\frac{Q_1^2}{1+Q_1^2}\right)\arcsin\left(\frac{Q_2^2}{1+Q_2^2}\right) \\ &- \frac{8}{\pi^2}\arcsin\left(\frac{R_1}{\sqrt{2(1+Q_1^2)}}\right)\arcsin\left(\frac{R_2}{\sqrt{2(1+Q_2^2)}}\right). \quad (18)\end{aligned}$$

Note, that no cross overlaps of the type $\underline{J}_1 \cdot \underline{B}_2$ or $\underline{J}_1 \cdot \underline{J}_2$ enter, because the inputs to the hidden units are taken to be drawn independently.

We consider the learning procedure

$$\underline{J}_l^{\mu+1} = \underline{J}_l^{\mu} + \frac{\eta}{N}\,\delta_l^{\mu}\,\underline{\xi}_l^{\mu} \qquad\qquad l = 1,2 \qquad (19)$$

where $\delta_{1,2} = \left[\tau(\underline{\xi}^{\mu}) - \sigma(\underline{J}_1^{\mu},\underline{J}_2^{\mu})\right] g'(x_{1,2}^{\mu})\, g(x_{2,1}^{\mu})$ and $x_{1,2}^{\mu} = \underline{J}_{1,2}^{\mu} \cdot \underline{\xi}_{1,2}^{\mu}$.

In continuous time $\alpha = \mu/N$ one arrives at the system of differential equations

$$\begin{aligned}\frac{dR_l}{d\alpha} &= \eta\,\langle \delta_l y_l\rangle \\ \frac{d[Q_l^2]}{d\alpha} &= 2\eta\,\langle \delta_l x_l\rangle + \eta^2\Big\langle \delta_l^2 \Big\rangle. \quad (20)\end{aligned}$$

The averages are over the joint density of all internal fields, which in this case factorizes: $P(x_1,x_2,y_1,y_2) = P(x_1,y_1)\,P(x_2,y_2)$. The full form of (20) can be found in the appendix. Note however, that the average $\langle \delta_l^2\rangle$ cannot be performed analytically in general.

The differential equations conserve a physical symmetry of the type $R_1 = R_2, \quad Q_1 = Q_2$ between the hidden units. In fact it can be shown analytically, that —in the subspace of $Q_1 = Q_2$— the system is stable against small perturbations from $R_1 = R_2$. Simulations of the algorithm for finite N also indicate that this symmetry is favored for general initial conditions and no nonsymmetric fixed points of (20) were found numerically. Therefore, we restrict the following discussion to the simplified two–dimensional system of differential equations where $R_1 = R_2 = R$ and $Q_1 = Q_2 = Q$ in (31,32).

Due to the fact that $\sigma(\underline{J}_1,\underline{J}_2,\underline{\xi}) = \sigma(-\underline{J}_1,-\underline{J}_2,\underline{\xi})$ for this type of network, the actual sign of the overlap R for any fixed point (R,Q) is determined by the initial conditions and is otherwise irrelevant. Note, that $dR/d\alpha$ is an odd function of R, whereas $dQ/d\alpha$ is even. For simplicity we consider only non–negative values of R in the following.

From equ. (19) it is clear already that $\underline{J}_1 = \underline{J}_2 = 0$, i.e. $R = Q = 0$, is a steady state of the learning procedure. Furthermore, if the student starts from any initial configuration having zero overlap with the teacher weights, it will never leave the subspace with $R = 0$. For $\eta > \eta_c$ the corresponding asymptotic value of Q diverges.

Thus, *a priori knowledge* is required for successful learning in this model. Any nonzero initial overlap will eventually yield nontrivial generalization, because $R = 0$ is repulsive. A similar effect was recently observed in online unsupervised learning [21].

The second obvious fixed point is the 'perfect student' $R = Q = 1$. A linearization of the system around this point reveals a behavior qualitatively very similar to the case of a single unit, see sec. 3. Provided the learning rate is sufficiently small, $\eta < \eta_c \approx 6.165$, the system approaches $R = Q = 1$ exponentially fast in α. The decay is asymptotically given by

$$\begin{aligned} R, Q \;\; &\propto \;\; e^{\lambda(\eta)\alpha} \quad \text{where} \quad \lambda(\eta) = \max(\lambda_1, \lambda_2) \quad \text{with} \\ & \lambda_1 \approx -0.2450\,\eta + 0.03975\,\eta^2 \quad \text{and} \quad \lambda_2 \approx -0.7461\,\eta \end{aligned} \tag{21}$$

The evolution of the generalization error for fixed η and different initial conditions is shown in figure 5. Note again, that ε_g explicitely depends on Q, hence the α–dependence of the generalization error even for $R = 0$.

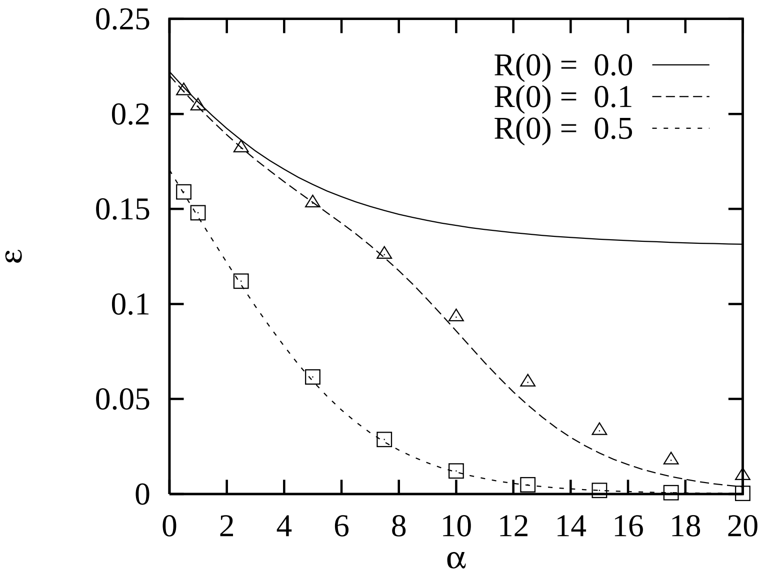

Figure 5: Evolution of the generalization error of the soft–parity machine for fixed $\eta = 3$ and different initial overlaps ($Q(0) = 1$ for all curves). Simulations were done with $N = 200$, averaged over 100 independent runs. Standard error bars would be smaller than the size of the symbols.

Learning slows down critically as $\eta \to \eta_c$. For even larger learning rates the first eigenvalue is positive, therefore (1,1) becomes unstable and a new attractive fixed point appears with a corresponding $\varepsilon_g(\alpha \to \infty) > 0$. Like for the single unit, for $\eta > \eta_d = \pi / \arcsin(1/3)$ no fixed point exists and both $Q(\alpha \to \infty)$ and $R(\alpha \to \infty)$ diverge.

In the vicinity of $R = Q = 1$ we find for the generalization error

$$\varepsilon_g \approx \frac{8}{3\sqrt{3}\pi}(r-q) + \frac{2}{3\pi^2}\left[(q+2r)^2 - 8r^2\right] - \frac{4\sqrt{3}}{27\pi}\left[(r-2q)^2 - 2q^2\right] \tag{22}$$

where $r = 1 - R$ and $q = 1 - Q$. Again, the eigenvalue λ_2 corresponds to a decay along $r = q$, and this implies, like for the single unit, that the relaxation of the generalization error is according to

$$\varepsilon_g(\alpha) \propto \begin{cases} e^{2\lambda_2 \alpha} & \text{for} \quad 0 < \eta < \eta_{\text{opt}} \\ e^{\lambda_1 \alpha} & \text{for} \quad \eta_{\text{opt}} \le \eta < \eta_c \end{cases} \tag{23}$$

In the sense of the discussion in section 3 the optimal learning rate would be defined by the condition $\lambda_1 = 2\lambda_2$, yielding $\eta_{\text{opt}} \approx 2.411$.

6 Summary and Outlook

We have studied an exactly solvable model of online gradient–descent learning in multilayer networks. For different student/teacher pairs and randomly drawn training examples we have investigated the generalization performance of the online learners. Averaging over the distribution of inputs in the thermodynamic limit allows us to write down deterministic differential equations for the order parameters, which can be solved numerically. For a single graded–response unit learning a realizable rule with sufficiently small learning rates we find an exponential decay of the generalization error. However, if the learning rate is increased above a critical value, the network approaches a suboptimal fixed point with a non–vanishing generalization error instead. Not surprisingly, this behavior is similar to that of a linear unit [22, 23], because in the vicinity of the optimal solution $\underline{B}$ the error surface is to leading order quadratic in $(\underline{B} - \underline{J})$ as in the linear case.

For both the soft–committee machine learning a rule defined by a single unit and the soft–parity machine learning from another soft–parity machine the asymptotic approach to the fixed point is exponentially fast as in the case of the single unit. Again, we find critical values of the learning rate, above which perfect learning becomes impossible. In contrast to the single unit, the two–layer systems show additional features related to their internal symmetries. The output of the committee machine is invariant under permutations of the hidden–unit weight vectors. Correspondingly, we find fixed points of the differential equations for the order parameters that also obey this symmetry and strongly influence the small–α behavior of the learning dynamics. Even though these fixed points are unstable, the repulsion from the symmetric subspace is slow compared to the attraction within the symmetric subspace. A similar effect of delayed learning was recently observed in the equilibrium behavior of offline Gibbs learning in large committee machines with binary threshold units [24, 25]. In this model perfect generalization required a specialization of hidden units. However, for small training sets the equilibrium solution was a committee–symmetric one with poor generalization ability. Only for sufficiently large training sets could the network realize the necessity of breaking this symmetry.

The output of the soft–parity machine is invariant under a simultaneous change of sign of both weight vectors. The corresponding fixed point of the differential equations is $R = Q = 0$, a student that has not inferred any information about the rule. Again, a similar situation of 'memorization without generalization' was observed in the offline equilibrium behavior of the corresponding model with binary units [26, 27]: for small training sets the existence of a local minimum of the free energy with $R = 0$ causes the student network to fail completely.

It would be desirable to gain further understanding of these similarities between off–and online learning, also in order to understand to what extent results from statistical mechanics carry over to stochastic online gradient–descent strategies, whose equilibrium distributions are not of the Gibbs type [15].

Recently, Kabashima studied online learning in a parity machine with binary units according to the so–called least–action algorithm [28]. There, a loss of generalization is observed when only noisy training outputs are available. It would be interesting to study how such noisy example outputs or inputs influence the outcome of our model, in particular with respect to the dependence of the asymptotic behavior upon the learning rate. Furthermore our studies should be extended to a more detailed analysis of situations in which the rule is unlearnable for the student.

It should also be possible to apply the method to the minimization of more sophisticated cost functions, such as *entropic* or *well formed* error measures [1], as well as modified learning schemes, e.g. [29].

Acknowledgements

We would like to thank W. Kinzel and G. Reents for useful discussions. H. Schwarze was supported by the Swedish Natural Science Research Council. M. Biehl acknowledes financial support by the Deutsche Forschungsgemeinschaft. He thanks the Department of Theoretical Physics at Lund University, where this work was completed.

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] B. Müller and J. Reinhardt, *Neural Networks,* Springer (Berlin), 1990.

[3] H.S. Seung, H. Sompolinsky, and N. Tishby, Phys.Rev. **A45**, 6056 (1992).

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

[5] W. Kinzel and M.Opper, *Statistical Mechanics of Generalization*, to appear in *Physics of Neural Networks III,* series eds. E. Domany, J.L. van Hemmen, and K. Schulten, Springer (Berlin), 1994.

[6] F. Rosenblatt, *Principles of Neurodynamics*, Spartan (New York), 1962.

[7] M.L. Minsky and S. Papert, *Perceptrons*, MIT Press (Cambridge, Mass.), 1969 and 1988.

[8] A.E. Bryson and Y.-C. Ho, *Applied Optical Control,* Blaisdell (New York), 1969.

[9] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, in *Parallel Distributed Processing I,* eds. D.E. Rumelhart, J.L. McClelland, *et al.*, MIT Press (Cambridge), 1986.

[10] Y. Le Cun, in *Disordered Systems and Biological Organization,* eds. E. Bienenstock, F. Fogelman–Soulié, and G. Weisbuch, Springer (Berlin), 1986.

[11] R.P. Lippmann, Neural Computation **1**, 1 (1989).

[12] G. Radons, H. Schuster, and D.Werner, in *Parallel Processing in Neural Systems and Computers,* eds. R. Eckmiller *et al.*, Elsevier Science Publishers (North Holland), 1990.

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

[14] T.K. Leen and G.B. Orr, in *International Joint Conference on Neural Networks,* IEEE, 1992.

[15] L.K. Hansen, R. Pathria, and P. Salamon, J.Phys. **A 26**, 63 (1993).

[16] R. Kühn, S. Bös, and J.L. van Hemmen, Phys.Rev. **A 43**, 2084 (1991).

[17] S. Bös, W. Kinzel, and M. Opper, Phys.Rev. **E 47**, 1384 (1993).

[18] D.O. Hebb, *The Organization of Behavior*, Wiley (New York), 1949.

[19] F. Vallet, Europhys.Lett. **9**, 315 (1989).

[20] Y. LeCun, P.Y. Simard, and B. Pearlmutter, in *Advances in Neural Information Processing Systems V,* eds. S.J. Hanson, J.D. Cowan, and C.L. Giles, Morgan Kaufmann (San Mateo), 1993.

[21] M. Biehl, Europhys.Lett. **25**, 391 (1994).

[22] B. Widrow and S.D. Stearns, *Adaptive Signal Processing,* Prentice–Hall (Eaglewood Cliffs, NJ), 1985.

[23] Y. Le Cun, I. Kanter, and S.A. Solla, in *Advances in Neural Information Processing Systems III*, eds. R.P. Lippmann, J.E. Moody, and D.S. Touretzky, Morgan Kaufmann (San Mateo), 1991.

[24] H. Schwarze, J.Phys. **A 26**, 5781 (1993).

[25] H. Schwarze and J. Hertz, in *Advances in Neural Information Processing Systems VI*, eds. J.D. Cowan, G. Tesauro, and J. Alspector, Morgan Kaufmann (San Francisco, CA), 1994.

[26] D. Hansel, G. Mato, and C. Meunier, Europhys.Lett. **20**, 471 (1992).

[27] M. Opper, Phys.Rev.Lett. **72**, 2113 (1994).

[28] Y. Kabashima, J.Phys. **A 27**, 1917 (1994).

[29] J. Hertz, A. Krogh, B. Lautrup, and T. Lehman, *Non–Linear Back–Propagation: Doing Back–Propagation without Derivatives of the Activation Function,* CONNECT–preprint, 1994.

A The Soft–Committee Machine

In the limit $N \to \infty$ the average over internal fields in (16) is an average over the distribution

$$P(x_1, x_2, y) = \det(\underline{\underline{C}})^{-1/2}(2\pi)^{-3/2}\ \exp\left[-\frac{1}{2}(x_1, x_2, y)\,\underline{\underline{C}}^{-1}\,(x_1, x_2, y)^{\top}\right] \tag{24}$$

with the covariance matrix

$$\underline{\underline{C}} = \begin{pmatrix} Q_1^2 & C & R_1 \\ C & Q_2^2 & R_2 \\ R_1 & R_2 & 1 \end{pmatrix}. \tag{25}$$

After a lengthy but straightforward calculation using $g(x) = \mathrm{erf}(x/\sqrt{2})$, $g'(x) = \sqrt{2/\pi}\, e^{-x^2/2}$ and the identity

$$\int_{-\infty}^{+\infty} \frac{dx}{\sqrt{2\pi}}\, e^{-x^2/2}\, \mathrm{erf}\,(ax)\, \mathrm{erf}\,(bx) = \frac{2}{\pi}\ \arcsin\left(\frac{2ab}{\sqrt{(1+2a^2)\,(1+2b^2)}}\right) \tag{26}$$

we arrive at the differential equations

$$\frac{dR_1}{d\alpha} = \frac{2}{\pi}\frac{\eta}{1+Q_1^2}\left[\frac{1+\Delta_1}{\sqrt{2+Q_1^2+\Delta_1}} - \frac{R_1}{\sqrt{1+2Q_1^2}} - \frac{(1+Q_1^2)R_2 - CR_1}{\sqrt{\Lambda}}\right] \tag{27}$$

$$\begin{aligned}
\frac{d[Q_1^2]}{d\alpha} &= \frac{4}{\pi}\frac{\eta}{1+Q_1^2}\left[\frac{R_1}{\sqrt{2+Q_1^2+\Delta_1}} - \frac{Q_1^2}{\sqrt{1+2Q_1^2}} - \frac{C}{\sqrt{\Lambda}}\right] \\
&+ \frac{4}{\pi^2}\frac{\eta^2}{\sqrt{1+2Q_1^2}}\left[\arcsin\left(\frac{Q_1^2}{1+3Q_1^2}\right) + \arcsin\left(\frac{2\Delta+Q_2^2}{\Lambda_1}\right)\right. \\
&+ 2\arcsin\left(\frac{C}{\sqrt{\Lambda_1}\sqrt{1+3Q_1^2}}\right) - 2\arcsin\left(\frac{R_1}{\sqrt{1+3Q_1^2}\sqrt{2(1+Q_1^2+\Delta_1)}}\right) \\
&- \left. 2\arcsin\left(\frac{R_2(1+2Q_1^2)-2R_1C}{\sqrt{\Lambda_1}\sqrt{2(1+Q_1^2+\Delta_1)}}\right) - 2\arcsin\left(\frac{1+2\Delta_1}{2(1+Q_1^2+\Delta_1)}\right)\right]
\end{aligned} \tag{28}$$

$$\begin{aligned}
\frac{dC}{d\alpha} &= \frac{2}{\pi}\frac{\eta}{1+Q_1^2}\left[\frac{(1+Q_1^2)R_2 - CR_1}{\sqrt{2+Q_1^2+\Delta_1}} - \frac{C}{\sqrt{1+2Q_1^2}} - \frac{Q_2^2+\Delta}{\sqrt{\Lambda}}\right] \\
&+ \frac{4}{\pi^2}\frac{\eta^2}{\sqrt{\Lambda}}\left[\arcsin\left(\frac{Q_1^2+\Delta}{\Lambda_1}\right) - 2\arcsin\left(\frac{(1+Q_2^2)R_1 - CR_2}{\sqrt{\Lambda_1}\sqrt{(1+\Delta_1)\,(1+\Delta_2)-\Delta^2+\Lambda}}\right)\right. \\
&+ \left.\arcsin\left(\frac{C}{\sqrt{\Lambda_1\Lambda_2}}\right) + \frac{1}{2}\arcsin\left(\frac{(1+\Delta_1)\,(1+\Delta_2)-\Delta^2}{(1+\Delta_1)\,(1+\Delta_2)-\Delta^2+\Lambda}\right)\right] \\
&+ (1 \longleftrightarrow 2)
\end{aligned} \tag{29}$$

where we have used the abbreviations

$$
\begin{array}{rl}
\Delta = Q_1^2 Q_2^2 - C^2 & \Lambda = (1+Q_1^2)(1+Q_2^2) - C^2 \\
\Delta_1 = Q_1^2 - R_1^2 & \Lambda_1 = (1+2Q_1^2)(1+Q_2^2) - 2C^2 \\
\Delta_2 = Q_2^2 - R_2^2 & \Lambda_2 = (1+Q_1^2)(1+2Q_2^2) - 2C^2 .
\end{array}
\tag{30}
$$

The equations for $dR_2/d\alpha$ and $d[Q_2^2]/d\alpha$ are similar to (27) and (28), respectively, just with the indices 1 and 2 interchanged.

B The Soft–Parity Machine

The system of differential equations (20) reads

$$
\begin{aligned}
\frac{dR_1}{d\alpha} &= \frac{4\eta}{\pi^2(1+Q_1^2)}\left[\frac{1+Q_1^2-R_1^2}{\sqrt{2(1+Q_1^2)-R_1^2}}\arcsin\left(\frac{R_2}{\sqrt{2(1+Q_2^2)}}\right)\right. \\
&\quad \left. - \frac{R_1}{\sqrt{1+2Q_1^2}}\arcsin\left(\frac{Q_2^2}{(1+Q_2^2)}\right)\right]
\end{aligned}
\tag{31}
$$

$$
\begin{aligned}
\frac{d[Q_1^2]}{d\alpha} &= \frac{8\eta}{\pi^2(1+Q_1^2)}\left[\frac{R_1}{\sqrt{2(1+Q_1^2)-R_1^2}}\arcsin\left(\frac{R_2}{\sqrt{2(1+Q_2^2)}}\right)\right. \\
&\quad \left. - \frac{Q_1^2}{\sqrt{1+2Q_1^2}}\arcsin\left(\frac{Q_2^2}{1+Q_2^2}\right)\right] \\
&\quad + \frac{4\eta^2}{\pi^2\sqrt{1+2Q_1^2}}\left[\arcsin\left(\frac{1+2(Q_1^2-R_1^2)}{2+4Q_1^2-2R_1^2}\right)\left\langle g^2(x_2)g^2(y_2)\right\rangle\right. \\
&\quad - 2\arcsin\left(\frac{R_1}{\sqrt{2+4Q_1^2-2R_1^2}\sqrt{1+3Q_1^2}}\right)\left\langle g^3(x_2)g(y_2)\right\rangle \\
&\quad \left. + \arcsin\left(\frac{Q_1^2}{1+3Q_1^2}\right)\left\langle g^4(x_2)\right\rangle\right]
\end{aligned}
\tag{32}
$$

and correspondingly for R_2 and Q_2.

Averages are over $P(x_1, y_1) \cdot P(x_2, y_2)$, where $P(x_l, y_l)$ is identical with the distribution for the single unit (section 3).

In the simplified case of physical symmetry, $R_1 = R_2 = R$ and $Q_1 = Q_2 = Q$, the linearization around the fixed point $(R = 1, Q = 1)$ is of the form

$$\begin{pmatrix} r' \\ q' \end{pmatrix} = \underline{\underline{A}} \begin{pmatrix} r \\ q \end{pmatrix} \tag{33}$$

where $r = 1 - R$ and $q = 1 - Q$. The matrix $\underline{\underline{A}}$ is given by

$$\begin{aligned}
\underline{\underline{A}} &= -\frac{\eta}{9\pi^2} \begin{pmatrix} -6 + \frac{8}{\sqrt{3}}\pi & 9 - 2\sqrt{3}\pi \\ -6 - \frac{4}{\sqrt{3}}\pi & 9 + 2\sqrt{3}\pi \end{pmatrix} \\
&\quad + \frac{2\eta^2}{\sqrt{3}\pi^2} \begin{pmatrix} 0 & 0 \\ -\frac{2}{5}\sqrt{\frac{3}{5}} + \frac{dG}{dR}\Big|_{(1,1)} & \frac{2}{5}\sqrt{\frac{3}{5}} + \frac{dG}{dQ}\Big|_{(1,1)} \end{pmatrix} \\
&\text{with} \\
G &= \arcsin\left(\frac{1}{4}\right)\left[\left\langle g^4(x)\right\rangle - 2\left\langle g^3(x)g(y)\right\rangle + \left\langle g^2(x)g^2(y)\right\rangle\right] \\
\text{and } \frac{dG}{dQ}\Big|_{(1,1)} &= -\frac{dG}{dR}\Big|_{(1,1)} \approx 0.1183 \cdot \arcsin\left(\frac{1}{4}\right).
\end{aligned} \tag{34}$$