Statistical mechanics approach to early stopping and weight decay

Siegfried Bös
Laboratory for Information Synthesis, Brain Science Institute, RIKEN, Wako-shi, Saitama 351-01, Japan
(Received 10 November 1997; revised manuscript received 13 February 1998)

Overtraining as a result of the difference between the empirical loss and the expected loss is a serious problem in neural network learning. It is known that methods such as early stopping, weight decay, or input noise can reduce overtraining. Here, these methods are studied in detail. We use a model that allows an analytical treatment. The treatment is based on an equilibrium statistical mechanics approach that is extended to its finite temperature solution. An unrealizable task that shows strong overtraining is examined. We find that overtraining can be completely avoided with each of the three methods if the parameters are optimally chosen. It is also shown that overtraining can appear in a realizable task, if the task is highly nonlinear. Also there overtraining can be avoided with each of the three methods. [S1063-651X(98)14206-X]

PACS number(s): 87.10.+e, 07.05.Mh, 05.20.−y

I. INTRODUCTION

A. Batch training

The ability to learn functional relations $\mathcal{F}$ between an input $\mathbf{x}$ and an output $\mathbf{z}$,

$$\mathcal{F}:\mathbf{x}\in\mathcal{I}\rightarrow\mathbf{z}\in\mathcal{O}, \tag{1}$$

from a finite number of examples is the major advantage of neural networks. Learning from examples becomes necessary in all applications, if no model exists, how the independent variable $\mathbf{x}$ determines the value of the dependent variable $\mathbf{z}$. If furthermore many observations are available or easily accessible, then neural networks become the first choice.

Supervised batch training using gradient descent is probably the most common training algorithm for neural networks. *Supervised* means that a set of examples $\mathcal{S}_P$ is available, consisting of P inputs $\mathbf{x}_\mu$ and the corresponding target output z^*_μ, i.e., $\mathcal{S}_P=\{(\mathbf{x}_\mu,z^*_\mu),\mu=1,\ldots,P\}$. It is then possible to define a *loss function*, i.e., $l[z^*,z]$, which measures the difference between the target output z^*_μ and the actual output z_μ. The average of this loss function over the whole set of examples is minimized by supervised training. *Batch* training or off-line training uses all examples simultaneously in each update step and repeats the updates until a certain termination condition is reached. The update, from $\eta\Delta\mathbf{W}(t)=\mathbf{W}(t+1)-\mathbf{W}(t)$, is then

$$\Delta\mathbf{W}(t)=-\sum_{\mu=1}^{P}\nabla_{\mathbf{W}}l[z^*_\mu,z_\mu(\mathbf{x}_\mu,\mathbf{W})], \tag{2}$$

where $\mathbf{W}$ denotes the adjustable parameters (weights) of the network, t counts the number of updates, and η is the learning rate. The actual output z is determined by the inputs $\mathbf{x}_\mu$ and the weights $\mathbf{W}(t)$. Another very popular training method is *on-line* learning, where only one example is used in each update, see [1].

Learning from examples has also some characteristic problems. Supervised training attempts to minimize the averaged error over the set of examples $\mathcal{S}_P$, which is called *empirical loss* or *training error*. How well the whole functional relation $\mathcal{F}$ is learned, is measured by the average of the loss function over the whole input space $\mathcal{I}$, which is called *expected loss* or *generalization error*.

If the number of examples in the training set becomes large, then the empirical loss will of course converge against the expected loss. However, for small example sets, problems commonly known as overtraining or overfitting can occur. *Overfitting* is used to denote that the network has more degrees of freedom than necessary for a specific task. The surplus in degrees of freedom results in an overadaptation to the data and a reduced generalization ability. Here, we will not deal with overfitting, for which model selection methods such as the Akaike Information Criterion [2], Network Information Criterion [3], or Bayes [4] have been proposed. However, even after an appropriate model selection has been applied, it can still happen that the variables accept wrong values due to misguidance, which is denoted as *overtraining*.

B. Avoidance of overtraining

Effective methods to avoid overtraining are early stopping, weight decay, or input noise. *Early stopping* makes use of the observation that the training algorithm specializes more and more on the specific examples of the example set. It can therefore be advantageous to terminate training before the specialization becomes too high. *Weight decay* is based on the fact that overtraining is accompanied by very large weights. The additional weight decay term reduces the size of the weights in each iteration. *Input noise* or jitter, which is random noise added to the inputs $\mathbf{x}$, can also prevent a too high specialization.

All the above procedures have one free parameter that must be optimized. These are the optimal stopping time t_{opt}, the optimal weight decay strength λ_{opt}, and the optimal level of the input noise δ_{opt}, respectively. In order to choose the parameters optimally, additional knowledge about the expected loss must be facilitated, since the empirical loss cannot provide this information.

Validation schemes, such as test-set validation or cross validation, attempt to provide this information about the expected loss. *Test-set validation* uses an additional empirical loss measured on a set of examples that are not used for

training. Since these examples are not used for training, the algorithm cannot adapt to them and the averaged loss on the test set can be used as a suitable approximation for the expected loss. The quality of the approximation increases with the size of the test set. That these examples cannot be used for training is an obvious disadvantage of this method. *Cross validation* seeks to overcome this problem at the expense of computing time. Although it uses only a small test set, the same procedure is repeated several times with different test sets. The optimal parameter is then determined by averaging over the results achieved with the different test sets.

We also want to discriminate between realizable and unrealizable tasks. In a *realizable task*, not only the training error but also the generalization error can become zero. The student is able to learn the whole task exactly. This is not possible in an *unrealizable task*, where the generalization error can only be decreased to a finite residual error E_∞. It is important to note that unrealizable tasks are more common than realizable tasks.

In the literature [5–7], it was already pointed out that several methods can improve the generalization ability. This paper follows a similar direction in that we discuss the emergence of overtraining in two models and show strategies to avoid it. We will study these questions within the framework of statistical mechanics. A detailed outline will follow at the end of the next section.

II. THE MODEL

A. Single-layer perceptron

In this paper, we restrict ourselves to single-layer perceptrons. A single-layer perceptron consists of only one layer of weights $\mathbf{W}$ between the N-dimensional input $\mathbf{x}$ and one output unit z. From the weighted sum h of the inputs x_i, the output is calculated by applying a transfer function $g(h)$, i.e.,

$$z=g(h), \quad \text{with } h=\frac{1}{\sqrt{N}}\mathbf{W}\mathbf{x}=\frac{1}{\sqrt{N}}\sum_{i=1}^{N} W_i\, x_i. \tag{3}$$

The training set for supervised learning is $\mathcal{S}_P=\{(\mathbf{x}_\mu, z_\mu^*), \mu=1,\ldots,P\}$, providing the correct output z_μ^* for each input $\mathbf{x}_\mu$. For theoretical purposes, it is very useful to assume that the correct outputs are computed by a second network, the so-called *teacher network*. By comparing the architectural complexity of the teacher and the student network it becomes apparent whether a task is realizable or unrealizable. Moreover, monitoring the training process becomes easier, since it can be described by a comparison of the variables from both networks. Variables denoting the teacher will always be indicated by an asterisk.

Using the mean-squared error and the teacher concept, we can write the loss function as

$$l(\mathbf{x},\mathbf{W}^*,\mathbf{W}):=\frac{1}{2}\left[g^*\left(\frac{\mathbf{W}^*\mathbf{x}}{\sqrt{N}}\right)-g\left(\frac{\mathbf{W}\mathbf{x}}{\sqrt{N}}\right)\right]^2. \tag{4}$$

The training error E_T is the averaged loss over the training set,

$$E_T:=\frac{1}{P}\sum_{\mu=1}^{P} l(\mathbf{x}_\mu,\mathbf{W}^*,\mathbf{W}). \tag{5}$$

The performance of the network on the whole task is measured by averaging over all possible inputs. This defines the *generalization error* E_G,

$$E_G:=\langle l(\mathbf{x},\mathbf{W}^*,\mathbf{W})\rangle_{\{\mathbf{x}\in I\}}. \tag{6}$$

Minimizing the training error will in turn minimize the generalization error, if the number of examples P is large. How this procedure works for small P is the subject of the next sections.

B. The generalization error

A main idea of the statistical mechanics approach is to make an assumption about the distribution of the inputs, such that the generalization error (6) can be calculated. For random inputs $\mathbf{x}$ from a nonpathological distribution with zero mean and unit variance, it can be assumed that the weighted sums h^* and h (3) are Gaussian distributed random numbers, if the dimension N is large. The correlations of these two Gaussians are

$$\langle (h^*)^2\rangle_x=1, \quad \langle h^*h\rangle_x=R, \quad \langle (h)^2\rangle_x=Q. \tag{7}$$

These define the two dynamical *order parameters*,

$$R:=\frac{1}{N}\sum_{i=1}^{N} W_i^* W_i, \quad Q:=\frac{1}{N}\sum_{i=1}^{N} (W_i)^2. \tag{8}$$

Here, we assume that the teacher weights $\mathbf{W}^*$ have norm one and introduce a variable gain γ for the transfer function $g^*(h^*)$. In other papers [1], the norm $T=N^{-1}\mathbf{W}^*\mathbf{W}^*$ is used instead of γ. Whether γ or $\sqrt{T}$ is used is a matter of individual taste. However, it should be emphasized that this parameter is only task dependent and remains unchanged during the training process.

The statistical mechanics approach is exact only in the thermodynamic limit, i.e., $N\to\infty$. Therefore, the variable $\alpha:=P/N$ is a more appropriate measure for the size of the example set. It can then be assumed that N and P are infinite with α remaining finite. The theory is under normal circumstances valid for reasonable system sizes such as $N\geqslant 100$. See also [8], where the author shows that $N>24$ is already large enough.

At times it is more convenient to use the normalized order parameters, $q:=\sqrt{Q}$ and $r:=R/q$, since they have more transparent interpretations. The normalized overlap r is the cosine of the angle between the two vectors $\mathbf{W}$ and $\mathbf{W}^*$. And q is the Euclidean norm of the student's weight vector.

The generalization error (6) becomes an average over the correlated Gaussians h^* and h. After a decorrelation into $\tilde{h}^*$ and $\tilde{h}$, we have the form

$$E_G(R,Q)=\langle \tfrac{1}{2}[g^*(\gamma h^*)-g(h)]^2\rangle_{\tilde{h}^*,\tilde{h}}, \tag{9}$$

with

$$h^*=:\tilde{h}^*, \quad \text{and } h=:R\tilde{h}^*+\sqrt{Q-R^2}\tilde{h}. \tag{10}$$

The average over the independent Gaussian variables is denoted by

$$\langle\ \rangle_{\tilde{h}} := \int_{-\infty}^{\infty} \frac{d\tilde{h}}{\sqrt{2\pi}} \exp\left(-\frac{\tilde{h}^2}{2}\right) \cdots . \tag{11}$$

Especially in the case where the student has a linear transfer function $g(h)=h$, the expression for the generalization error becomes very simple,

$$E_G^{\mathrm{lin}}(R,Q) = \tfrac{1}{2}(G - 2HR + Q). \tag{12}$$

The constants $G(\gamma)$ and $H(\gamma)$,

$$G := \langle [g^*(\gamma\tilde{h}^*)]^2 \rangle_{\tilde{h}^*}, \quad H := \langle g^*(\gamma\tilde{h}^*)\tilde{h}^* \rangle_{\tilde{h}^*}, \tag{13}$$

summarize the dependence on the teacher. The results hold for all teacher transfer functions, for which the constants G and H can be calculated.

As mentioned in the Introduction, we have special interest in unrealizable tasks. The essential feature of unrealizable tasks in our approach is $G \neq H^2$. Here, unrealizable tasks emerge from selecting a teacher transfer function that is different from the linear function $g(h)=h$. Interesting choices are nonlinear, sigmoid functions such as $\tanh(\gamma h^*)$ or $\mathrm{erf}(\gamma h^*)$, or the addition of Gaussian noise $\epsilon \in \mathcal{N}(0,\sigma)$ to the linear function, i.e., $\gamma h^* + \epsilon$. For this noisy linear teacher, the constants can be computed algebraically,

$$G = \gamma^2 + \sigma^2, \quad H = \gamma. \tag{14}$$

It should be noted that every task with a linear student can be expressed by an equivalent task with a noisy linear teacher, since each combination of G and H can be realized by an appropriate choice of γ and σ. The noisy linear teacher was studied by Krogh and Hertz [9] in some detail using a Greens-function approach. We will occasionally refer to their results.

For the computation of the plots, we have to choose specific values for G and H. For historical reasons and to allow a comparison with the task in Sec. VI, we have chosen $G = 0.84$ and $H = 0.78$, which corresponds to $g^*(h^*) = \tanh(5h^*)$. This is in no way a restriction.

Realizable tasks are the special case, where $G = H^2$ holds, implying a noise-free linear teacher. This cancels all terms proportional to $G - H^2$ and the behavior becomes much more simplified, see also [10].

C. Outline

In the next section we will provide the framework based on an equilibrium statistical mechanics approach necessary for the later sections. We will show how the evolution of the order parameters can be determined from the free energy. An important restriction will arise from this calculation; we will learn that analytical results can be achieved only for the linear student.

The linear unrealizable task is then discussed in Sec. IV. It is shown that overfitting occurs and how it can be avoided by early stopping, weight decay, or input noise.

In Sec. V, an alternative approach is presented. It is based on the dynamics of training and therefore is well suited to describe training processes. After a brief discussion, we will compare it to the equilibrium approach.

In Sec. VI, we will apply our knowledge to a nonlinear student learning a realizable nonlinear teacher; i.e., teacher and student use both a ''tanh'' transfer function. Also in this task, while realizable, overfitting appears due to the nonlinearity in the task. Nonlinearity is an essential requirement to turn a multilayer net into a general function approximator. This case cannot be solved exactly; however, we can deduce an approximation from what we have learned from previous sections.

The paper is concluded with a summary in Sec. VII. The characteristic features of learning unrealizable tasks are summarized in a figure showing the different regimes of learning.

III. EQUILIBRIUM APPROACH

Here, the technical framework for the later sections, IV and VI, is provided. We briefly discuss how batch training can be described by an equilibrium statistical mechanics approach. In Sec. V an alternative approach to the same problem is discussed.

A. Free energy

It can be shown that the equilibrium distribution of the noisy gradient descent training, given by

$$W_i(t+1) - W_i(t) = -\nabla_{W_i}(PE_T) + \epsilon_i(t), \tag{15}$$

where ϵ denotes Gaussian white noise, is a Gibbs-Boltzmann distribution $\mathrm{Prob}(\mathbf{W}) \sim e^{-\beta PE_T}$. With $\beta = 1/T$, we denote the inverse of the temperature T ($k_B = 1$). This was shown in [11], for a dynamics in continuous time with noise, which is uncorrelated in the sites i and in time t and has a noise level $2T$.

The partition function counts the fraction of all the states, which fulfill the spherical normalization condition with the *a priori* norm of the weights Q_0,

$$Z = \frac{\displaystyle\int_{-\infty}^{\infty} \prod_{i=1}^{N} dW_i\, e^{-\beta PE_T} \delta\left[\sum_{i=1}^{N} (W_i)^2 - NQ_0\right]}{\displaystyle\int_{-\infty}^{\infty} \prod_{i=1}^{N} dW_i\, \delta\left[\sum_{i=1}^{N} (W_i)^2 - NQ_0\right]}, \tag{16}$$

where δ denotes the delta function. The *a priori* norm Q_0 is somewhat similar to a target norm that we want to reach. Later we will see that it is sometimes better not to reach this target.

The *free energy* f contains all relevant information, including the values of the order parameters R and Q. It is defined as the logarithm of the sum of states averaged over the distribution of the examples $\mathbf{x}_\mu$,

$$-\beta f := \frac{1}{N} \langle \ln Z \rangle_{\{\mathbf{x}_\mu : \mu = 1, \ldots, P\}}. \tag{17}$$

To average the logarithm of Z over the examples $\mathbf{x}_\mu$ in Eq. (17), the currently commonly used *replica trick* is applied. Details about such a calculation can be found in [12].

The solution of the continuous perceptron problem was already discussed in [13]. Here, we briefly recapitulate the results necessary for further discussion. Preliminary results were presented in [14].

In [13] the following free energy was found, with Gaussian averages (11) over h_1, h_2, and h_3,

$$-\beta f=\frac{1}{2}\frac{(Q-R^2)}{Q_0-Q}+\frac{1}{2}\ln(Q_0-Q)+\alpha\left\langle \ln\left[\left\langle \exp\left(-\frac{\beta(g_1-g_2)^2}{2}\right)\right\rangle_{h_3}\right]\right\rangle_{h_1,h_2}. \tag{18}$$

The functions g_1 and g_2 are

$$g_1=g^*(\gamma h_1),$$
$$g_2=g(Rh_1+\sqrt{Q-R^2}h_2+\sqrt{Q_0-Q}h_3). \tag{19}$$

Further analytical evaluations of this integral are unfortunately only possible if the student is linear, $g(h)=h$.

B. Results for the linear student

If the student has a linear output function, the expression for the free energy (18) simplifies considerably,

$$-\beta f=\frac{1}{2}\frac{(Q-R^2)}{Q_0-Q}+\frac{1}{2}\ln(Q_0-Q)-\frac{\alpha}{2}\ln[1+\beta(Q_0-Q)]-\frac{\alpha\beta}{2}\frac{G-2RH+Q}{1+\beta(Q_0-Q)}, \tag{20}$$

using the two constants G and H from Eq. (13) again.

The values of the order parameters R and Q are taken at extreme values of the free energy. From $\partial f/\partial R=0$ and $\partial f/\partial Q=0$ it follows that

$$R(\alpha,a)=\frac{\alpha}{a}H,$$
$$Q(\alpha,a)=\frac{\alpha}{a^2-\alpha}\left(G-\frac{2-a}{a}\alpha H^2\right). \tag{21}$$

Generalization error (9) and training error (5) given by $(1/\alpha)\partial(\beta f)/\partial\beta=E_T$, are

$$E_T(\alpha,a)=A(\alpha,a)\left(\frac{a-1}{a}\right)^2+\frac{1}{2\beta a},$$
$$E_G(\alpha,a)=A(\alpha,a)+\frac{1}{2\beta(a-1)}, \tag{22}$$

using the abbreviation

$$A(\alpha,a):=\frac{1}{2(a^2-\alpha)}[a^2G-(2a-\alpha)\alpha H^2]. \tag{23}$$

All of these depend on the normalized number of examples $\alpha=P/N$ and on the temperature-dependent parameter a, defined as

$$\frac{1}{a-1}:=\chi:=\frac{1}{T}(Q_0-Q), \tag{24}$$

with $\beta=1/T$.

C. The parameter a

The physical parameter χ measures the fluctuations of the weights, i.e.,

$$\chi=\frac{\beta}{N}\left(\overline{\sum_{i=1}^{N}\langle W_i^2\rangle}-\overline{\sum_{i=1}^{N}\langle W_i\rangle^2}\right), \tag{25}$$

where the angular brackets denote the thermal average, i.e., the average over the Gibbs distribution, and the overbar is the average over the examples. The parameter χ is a measure for the thermal fluctuations of the weights. A similar χ can be found in spin-glass theory, where it is called local susceptibility (see [15], p. 35). The parameter a, which is closely related to χ, plays an important role in our further discussion.

In the literature, usually the solution at temperature zero is considered as it implies that the training error E_T accepts its absolute minimum for any number of presented examples P. At temperature zero, two cases need to be distinguished, the underdetermined case with fewer examples than variables, i.e., $P<N$, and the overdetermined case with a surplus of examples $P>N$. The storage capacity of the continuous perceptron is $\alpha_c=P_c/N=1$ and lies between these two regimes.

For $P<N$ many solutions exist, and the limit $\beta\to\infty$ must be applied first, leading to $a=1$. Then a specific solution can be chosen. This might be the one with the minimal norm $q^2=Q=Q_0$, which is identical to the *pseudoinverse solution*, see Sec. V.

If $P>N$, there are more equations than variables. Therefore, also the value of Q_0 must be optimized simultaneously with the execution of the limit $\beta\to\infty$. The differentiation of the free energy according to Q_0, i.e., $\partial f/\partial Q_0=0$, leads to

$$\beta(Q_0-Q)=\frac{1}{\alpha-1}\quad(\alpha>1). \tag{26}$$

A comparison with Eq. (24) gives $a=\alpha$. The value of a in the zero temperature limit is

$$a_0(\alpha):=\max(1,\alpha). \tag{27}$$

In the next section, we will examine the effect of other choices for $a\in[a_0,\infty[$. We will observe that solutions corresponding to higher values of a have an interesting interpretation in the context of early stopping, weight decay, or input noise.

D. Weight decay

Another method used to prevent overtraining is weight decay. The most common type of weight decay is the qua-

dratic norm of the weights added to the training energy, in order to penalize large weights, i.e.,

$$\tilde{E}_T=\frac{1}{2P}\left[\sum_{\mu=1}^{P}(z_\mu^*-z_\mu)^2+\lambda\sum_{i=1}^{N}(W_i)^2\right]. \tag{28}$$

The open parameter in this approach is the relative strength of the weight decay term λ. The weight update is then $\Delta\mathbf{W}(t)=-\nabla_{\mathbf{W}}(P\tilde{E}_T)$, which is explicitly written in Eq. (43) of Sec. V. There, it can be seen that the additional weight decay term decreases the size of the weights in each update.

The calculation of the free energy from above needs to be repeated with the additional weight decay term. The calculation is very similar to the one without weight decay. As a result, we find that only one term is added to the free energy,

$$-\beta f\rightarrow-\beta f-\frac{\beta\lambda Q_0}{2}. \tag{29}$$

As the additional term is independent of R and Q, the corresponding order parameter equations (21) remain unaffected. Only the equation for Q_0 changes.

This has a very important consequence. It directly implies that a system with weight decay can be transformed into an equivalent system without weight decay, but at another value of a, i.e., at another temperature.

The determination of Q_0, i.e., $\partial f/\partial Q_0=0$, leads to

$$-\frac{1}{2}\left[\beta\lambda+\frac{\alpha\beta}{1+\beta(Q_0-Q)}-\frac{1}{Q_0-Q}\right]=0, \tag{30}$$

which can be rewritten as the following relation between a and λ,

$$b^2\lambda+b(\lambda+\alpha-1)-1=0, \tag{31}$$

with

$$b:=\beta(Q_0-Q)=(a-1)^{-1}. \tag{32}$$

The problem is now solved. The solution from above can be used with a determined by λ.

E. Input noise

It is also known that noise on the inputs—also referred to as *jitter* [7]—can reduce overtraining. Here, we show that for the linear model, input noise and weight decay are essentially the same.

We only have to compute how the training error is affected, if an independent Gaussian noise ϵ is added to the inputs. If the noise ϵ_i has zero mean and variance δ^2, then the training error becomes

$$\tilde{E}_T=\frac{1}{2P}\left[\sum_{\mu=1}^{P}\left(z_\mu^*-\frac{1}{\sqrt{N}}\sum_{i=1}^{N}W_i(x_i+\epsilon_i)\right)^2\right],$$

$$\simeq\frac{1}{2P}\left[\sum_{\mu=1}^{P}(z_\mu^*-z_\mu^0)^2+\frac{1}{N}\left(\sum_{i=1}^{N}W_i\epsilon_i\right)^2\right], \tag{33}$$

with z_μ^0 denoting the student output of a noise free input.

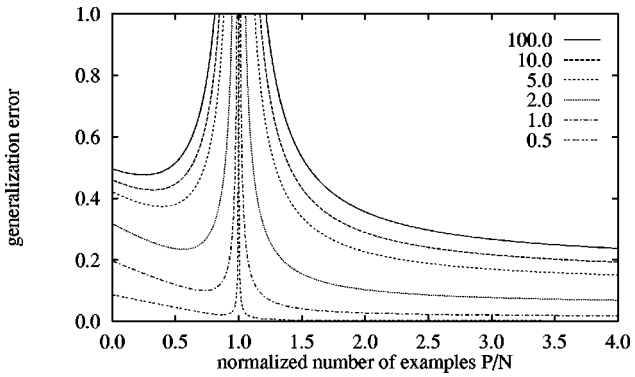

FIG. 1. Performance $E_G(\alpha)$. Generalization error E_G as a function of the normalized number of examples $\alpha=P/N$ is shown. A linear student is trained exhaustively, i.e., $a=a_0$, on an unrealizable task. The teacher uses a ''tanh'' transfer function, i.e., $g(h^*)=\tanh(\gamma h^*)$. The results for different gains γ are shown.

The W_i and the ϵ_i can be averaged independently, since the noise is independent of the inputs and therefore also of the weights. The resulting effect on the free energy is

$$-\beta f\rightarrow-\beta f-\frac{\beta}{2}Q_0\delta^2. \tag{34}$$

A comparison with Eq. (29) immediately reveals that the effect of input noise and weight decay are identical for the linear model. We simply replace λ by δ^2. The equivalence of weight decay and input noise in the linear model was already found by [9] using a Green's-function approach.

IV. LINEAR STUDENT LEARNING AN UNREALIZABLE TASK

Here, we study how a linear student learns an unrealizable task. As discussed in Sec. II, unrealizability implies $G\neq H^2$. The results for the realizable special case follow from $G=H^2$.

A. Minimal training error

To calculate the order parameters in the zero temperature limit, we insert $a_0(\alpha)$ from Eq. (27) into Eqs. (22). Below the storage capacity, that is for $\alpha<1$ and $a_0=1$, the minimum of the training error is always zero, i.e., $E_T(\alpha)=0$, and the generalization error is

$$E_G(\alpha)=\frac{G}{2}+\frac{\alpha}{2(1-\alpha)}[G-(2-\alpha)H^2]. \tag{35}$$

For $\alpha>1$ with $a_0=\alpha$, we arrive at

$$E_T(\alpha)=\frac{G-H^2}{2}\,\frac{\alpha-1}{\alpha}=\frac{G-H^2}{2}\left(1-\frac{1}{\alpha}\right),$$

$$E_G(\alpha)=\frac{G-H^2}{2}\,\frac{\alpha}{\alpha-1}=\frac{G-H^2}{2}\left(1+\frac{1}{\alpha-1}\right). \tag{36}$$

The resulting curve for the generalization error for different choices of the gain γ is shown in Fig. 1. Around the storage capacity of the perceptron $\alpha_c=1$ strong *overtraining*

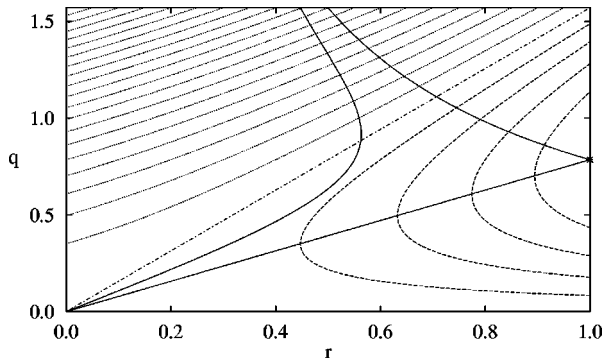

FIG. 2. The shape of $E_G(r,q)$. Contour plot of the generalization error $E_G(r,q)$ as a function of the two normalized order parameters q and r. From the minimum $E_G^{\min}=(G-H^2)/2$ at $(r,q)=(1,H)$ the contour lines for $E_G=E_G^{\min}+(H^2/10)i$ for $i=1,2,3,4$ (dashed lines) and for $i=6,7,\ldots,24$ (dotted lines) are given. The E_G^0 line corresponds to $E_G(0,0)=G/2$ or $i=5$ (dash-dotted line). The two solid lines are learning curves (see Fig. 3), the upper one corresponds to exhaustive training and the lower one to optimal training. (Parameters are $G=0.84$ and $H=0.78$, corresponding to γ is 5).

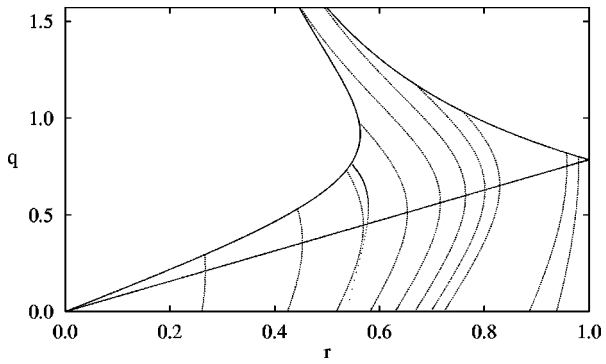

FIG. 3. Evolution of the order parameters. Parametric curves (r,q) showing the evolution of q and r as functions of the two parameters α and a are shown. On a learning curve (solid lines) the parameter α is varied and a has a certain dependence on α. This dependence is in the case of exhaustive training $a=a_0(\alpha)$ and in the case of optimal training $a=a_{\rm opt}(\alpha)$. On a training curve (dotted lines) the parameter a is decreased from infinity to a_0 while α remains fixed. Several training lines are shown for different values of $\alpha=0.1, 0.3, \ldots, 1.5, 5.0$, and 10.0. At $\alpha=0.5$, the theoretical curve is compared with a simulation of the training process (points near third line). (Parameters as in Fig. 2.)

appears, which can be seen in the divergence of the generalization error. We call the zero temperature limit *exhaustive training*, since the student net is trained until the minimal training error E_T is reached.

Note that the result of the realizable case, $G=H^2$, is simply $E_G(\alpha)=(G/2)(1-\alpha)$ for $\alpha<1$. The whole training process is completed at $\alpha=1$. This is of course a highly idealized case.

In the limit $\alpha\gg 1$, both errors approach the finite *residual error* $E_\infty:=\frac{1}{2}(G-H^2)$, with E_T coming from below and E_G from above. Both errors converge like α^{-1} against the residual error.

The two characteristic features of the unrealizable learning task—overtraining and finite residual error—have different causes. The residual error is a consequence of the unrealizability and cannot be avoided without transforming the task into a realizable task. Overtraining, however, is a result of exhaustive supervised training. The network attempts to learn all examples as well as possible, even if it implies that the generalization ability is neglected.

B. The shape of the generalization error

The fact that the system can be described with only two order parameters instead of the N dimensions of the weights allows a good illustration. We can plot the generalization error E_G over the whole range of possible values for the order parameters. Here, the normalized order parameters $q=\sqrt{Q}$ and $r=R/q$ are used. Their asymptotical limits are $r=1$ and $q=H$, which indicate that the weight vectors of student and teacher have the same direction, but are not of the same length.

In Fig. 2, $E_G(r,q)$ is plotted in the interesting range, $r\in[0,1]$ and $q\in[0,2H]$. Instead of a three-dimensional plot, the contour lines $E_G(r,q)=\text{const}$ are shown.

An important line is the $E_G(0)$ *isoline*, which remains on the initial value of the generalization error $E_G(0,0)=G/2$ for the initial conditions $q(0)=0$ and $r(0)=0$. Only combinations of (r,q) below this line correspond to an improved generalization ability. The absolute minimum of E_G is reached at $(r,q)=(1,H)$.

C. Evolution of the order parameters

The values of the order parameters r and q are connected by the training process. They are parametric curves $q(r)$ with the parameters α and a; see Eq. (21). Depending upon whether α or a is varied, we refer to them as a *learning curve* or a *training curve*, respectively. Some of these curves are shown in Fig. 3.

A learning curve follows from the variation of α with a certain choice for $a(\alpha)$. The exhaustive learning curve uses the minimal value of a, that is, $a_0(\alpha)$, and is plotted as the upper solid line in Fig. 3. Only the area below the exhaustive training curve is accessible by gradient descent batch training, if the initial conditions are $q(0)=0$ and $r(0)=0$.

A training curve, on the other hand, follows from a variation of a for a fixed value of α. Several training curves are shown in Fig. 3 as dashed lines. Along a training curve the parameter a is reduced from infinity to a_0.

Each path (r,q) implies a certain evolution of the generalization error, which can be seen if we project Fig. 3 on top of Fig. 2. If we move along a training curve, we can clearly observe when overtraining occurs. The points where the generalization error starts to increase are connected by another solid line. How this line can be interpreted is our next topic.

D. Finite training errors

An a value larger than a_0 corresponds to a finite temperature $T>0$. A finite temperature implies that the training error does not accept its absolute minimum, i.e., $E_T>E_T^{\min}$.

On the other hand, the training error is decreased during the training process. If it does not reach its minimal value, we must assume that the training process is stopped earlier. Through simulations of the training process, we have tested

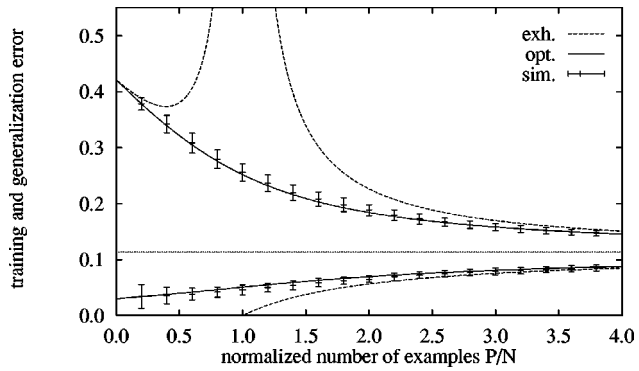

FIG. 4. Optimized performance. Comparison is shown of the performance $E_G(\alpha=P/N)$ after exhaustive training and optimal training for an unrealizable task. The upper lines denote the generalization error E_G, the lower ones denote the training error E_T. Exhaustive training (dashed lines), i.e., $a=a_0(\alpha)$, leads to overtraining. With optimal training (solid lines), i.e., $a=a_{\mathrm{opt}}(\alpha)$, overtraining can be completely avoided. Optimal training can be realized by three methods, early stopping, weight decay, or input noise, see text. Optimal early stopping was simulated (error bars) using the exact generalization error for validation. The finite residual error E_∞ is given by the dotted line. (Parameters as in Fig. 2.)

that the evolution of the order parameters can really be described by the variation of a; see points near the training curve for $\alpha=0.5$ in Fig. 3. The relevant parameter in the training process is the number of batch updates t; see Eq. (43).

We have just found an important correspondence. Decreasing temperature T, which is equivalent to reducing the value of a from infinity to a_0, can be seen an increase of the number of parallel batch training steps from 1 to $t_{\max}$.

Immediately, the question after the optimal time arises of where the training should be stopped. Unfortunately, there is no easy answer to this question, since the learning rate η [see Eq. (2)] determines the time scale. However, the optimal value of the parameter a can easily be determined.

Minimizing E_G with respect to a by $\partial E_G/\partial a=0$, neglecting the second term in Eq. (22), leads to

$$a_{\mathrm{opt}}(\alpha):=c\pm\sqrt{c^2-\alpha},\quad c=\frac{1}{2}\left(\alpha+\frac{G}{H^2}\right), \tag{37}$$

where the $+$ is the relevant solution. The corresponding generalization error is

$$E_G(\alpha)=\frac{1}{2}\left[G-\frac{\alpha}{a_{\mathrm{opt}}(\alpha)}H^2\right]. \tag{38}$$

The resulting behavior is shown in Fig. 4. It exhibits no overtraining at all.

Another indication of the optimality of this solution comes from the corresponding learning curve, which is the straight solid line shown in Fig. 2 and Fig. 3. Its analytical expression, $q(r)=Hr$, can be deduced from Eq. (21) with a few algebraic transformations.

E. Early stopping

We can confirm this result by a simulation of early stopping. Here, we do not want to bother with the limitations of an actual test-set validation or cross validation. Instead the exact generalization error is used for validation in order to receive the optimal result.

The results of the simulation using optimal validation are also shown in Fig. 4 by error bars. The error bars indicate the standard deviation in a simulation with $N=100$ averaged over 50 trials. The theoretical solution using a_{opt} is within the range of the error bars. Therefore, we can take the variation of the parameter a as a useful description of early stopping.

F. Weight decay and input noise

Next we want to discuss briefly the effect of weight decay on this problem. First we assume that the weight decay strength λ is fixed. This resembles a situation where no further knowledge about the system is available and a guess concerning the best weight decay strength must be made.

We can resolve Eq. (31) to receive a as a function of λ, i.e.,

$$a(\lambda)=\frac{1}{b_{1/2}(\lambda)}+1, \tag{39}$$

with $\lambda>0$,

$$b_{1/2}(\lambda)=\frac{1-\alpha-\lambda\pm\sqrt{(1-\alpha-\lambda)^2+4\lambda}}{2\lambda}. \tag{40}$$

Only the solution with the $+$ sign is a relevant solution. We insert $a(\lambda)$ into the equations of the order parameters (21) and the errors (22). The results clearly indicate that overtraining can be reduced using weight decay, for figures see [13].

Now we want to determine the optimal value for the weight decay strength for each α. The relation (31) between a and λ can be resolved to express λ as a function of a,

$$\lambda(a)=\frac{1+(1-\alpha)b}{b(b+1)}=\frac{a-1}{a}(a-\alpha). \tag{41}$$

We insert $a_{\mathrm{opt}}(\alpha)$ from Eq. (37) into this function to receive $\lambda_{\mathrm{opt}}(\alpha)$. The result is remarkably simple,

$$\lambda_{\mathrm{opt}}(\alpha)=\left(\frac{G}{H^2}-1\right). \tag{42}$$

The fact that the optimal weight decay strength is independent of α makes it suitable for applications. The optimal weight decay strength could be determined on a smaller sample, i.e., $\alpha'<\alpha$, using the rest of the examples for test-set validation or cross validation. Alternatively, we could fully exploit our theoretical equations and determine G from $E_T(0)=G/2$. Then H could be determined from the exhaustive training error E_T^{exh}, see Eq. (36).

As pointed out above, input noise can lead to the same effect with a noise level $\delta^2=\lambda$. Of particular interest is the optimal weight decay strength for the case where the teacher

is a noisy linear perceptron, see [9]. Then the constants are $G=\gamma^2+\sigma^2$ and $H=\gamma$, thus $\lambda_{\mathrm{opt}}(\alpha)$ becomes σ^2/γ^2. The optimal input noise level for this case is $\gamma^2\delta_{\mathrm{opt}}^2=\sigma^2$. This makes sense because for the linear student, the variance of the weights multiplied by the variance of the inputs is the variance of the outputs.

G. Summary

In this section, we have shown how overtraining—a common, but unwanted feature of unrealizable learning tasks—can be reduced. Technically speaking, overtraining can be reduced by increasing the temperature-dependent parameter a from a_0, which corresponds to a finite temperature $T>0$. This implies a simultaneous increase of the training error E_T from its minimal value $E_T^{\min}$.

The following three practical methods can have such an effect: (i) Training is stopped before the training error reaches its absolute minimum, i.e., at a time $t<t_{\max}$, or (ii) weight decay with a certain strength $\lambda>0$ is applied, or (iii) random noise with mean zero and variance $\delta>0$ is added to the inputs.

The three methods, early stopping, weight decay, and input noise, have similar positive effects in avoiding overtraining. Optimized by validation, they are able to avoid overtraining altogether. In the equilibrium statistical mechanics approach, we have found that they are equivalent, if the student is linear.

V. DYNAMICAL APPROACH

The same problem can also be addressed by an approach that is closely related to the dynamics of the training process. This approach provides the correct description of early stopping. A comparison with the above results can give further support for the equilibrium approach. Here, we briefly sketch the results, details can be found in [16].

The update, $\mathbf{W}(t+1)=\mathbf{W}(t)+\eta\Delta\mathbf{W}(t)$, using gradient descent has the following form, $\Delta\mathbf{W}(t):=-\nabla_{\mathbf{W}}(P\tilde{E}_T)$, with η denoting the *learning rate* and t counting the batch training steps. If the function $g(h)$ is linear, then the derivative $g'(h)$ vanishes and we get the so-called adaptive linear or *adaline* rule [17]. The update including weight decay has the form

$$\Delta W_i(t)=\frac{1}{\sqrt{N}}\sum_{\mu=1}^{P}[z_\mu^*-z_\mu(t)]x_{i\mu}-\lambda W_i(t). \tag{43}$$

It is possible to find an explicit solution for the weights. For $P<N$, this is

$$W_i(t)=\frac{\eta}{\sqrt{N}}\sum_{\mu,\nu=1}^{P}z_\mu^*\left\{\frac{\mathbf{1}-[(1-\eta\lambda)\mathbf{1}-\eta\mathbf{C}]^t}{\mathbf{1}-[(1-\eta\lambda)\mathbf{1}-\eta\mathbf{C}]}\right\}_{\mu\nu}x_{i\nu},$$

where $\mathbf{1}$ is the identity matrix, and

$$\mathbf{C}_{\mu\nu}=\frac{1}{N}\sum_{i=1}^{N}x_{i\mu}x_{i\nu}. \tag{44}$$

The solution is consistent with the initial conditions, $W_i(0)=0$ and $W_i(1)=\eta N^{-1/2}\Sigma_{\mu=1}^{P}z_\mu^*x_{i\mu}$, which corresponds to *Hebbian* learning. Asymptotically, after an infinite number of time steps, it yields the *pseudoinverse* weights including weight decay,

$$W_i(t\to\infty)=\frac{1}{\sqrt{N}}\sum_{\mu,\nu=1}^{P}z_\mu^*[(\lambda\mathbf{1}+\mathbf{C})^{-1}]_{\mu\nu}x_{i\nu}. \tag{45}$$

The solution for $P>N$ is very similar, see [16].

The explicit solution for the weight vector (44) can be inserted into the definition of the order parameters. The typical dynamics of the order parameters can then be determined by averaging over the input distribution.

A. Results

Here, we cannot go into detail about this approach. Only the results necessary for the comparison to the results of the equilibrium approach are presented.

The dynamical equations of $R(t)$ and $Q(t)$ can be written in a compact form if we define a constant $c=\min(\alpha,1)$,

$$\begin{aligned}R(\alpha,\lambda,\eta,t)&=cHI_{111},\\ Q(\alpha,\lambda,\eta,t)&=c(G-H^2)I_{221}+cH^2I_{222}.\end{aligned} \tag{46}$$

The integrals $I_{lmn}(\alpha,\lambda,\eta,t)$ are

$$I_{lmn}=\int_{\xi_{\min}}^{\xi_{\max}}d\xi\rho(\xi)\frac{[1-(1-\eta\lambda-\eta\xi)^t]^l}{[\lambda+\xi]^m}\xi^n, \tag{47}$$

with $l,m,n\in\{1,2,3\}$. The density of the eigenvalues $\rho(\xi)$ of the matrix $\mathbf{C}$ is

$$\rho(\xi)=\frac{1}{2\pi\xi c}\sqrt{(\xi_{\max}-\xi)(\xi-\xi_{\min})}. \tag{48}$$

Finally, the maximal and the minimal eigenvalues are $\xi_{\max,\min}:=(1\pm\sqrt{\alpha})^2$. The time-dependent integrals converge only if the learning rate is smaller than the maximal learning rate,

$$\eta_{\max}=\frac{2}{\lambda+\xi_{\max}}=\frac{2}{\lambda+1+2\sqrt{\alpha}+\alpha}. \tag{49}$$

The results of the dynamical approach are the correct description for early stopping and also for weight decay. A comparison of the results of the two approaches can provide interesting further insight into the validity of the equilibrium approach.

B. Comparison

As in Fig. 3, we can plot the evolution of the order parameters by parametric curves (r,q). In the dynamical approach, the parameters are t, η, and α for early stopping and additionally λ for weight decay. For the learning rate η, we have to assume that it is smaller than the maximum $\eta_{\max}$. The solution will then converge and the value of η determines the scaling of t.

In Fig. 5, we compare how the two approaches, equilibrium and dynamical, describe early stopping. The evolution

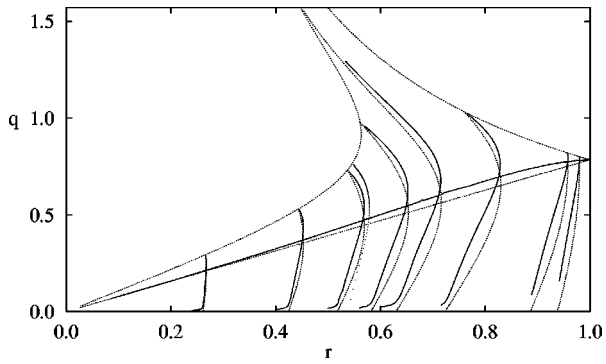

FIG. 5. Comparison of dynamical and equilibrium approach. The evolution of the order parameters (r,q), as it is described by the two approaches, is shown in a similar way to Fig. 3. The results of the equilibrium approach are given by dotted lines for $\alpha=0.1$, 0.3, . . . , 0.9, 1.5, 5.0, and 10.0. The solid lines are the corresponding results of the dynamical approach with $\eta=0.01$. The optimal evolution slightly deviates from the equilibrium result, however, the effect on the performance of E_G^{opt} is minimal. The dots near $\alpha=0.5$ are results of a simulation. (Parameters as in Fig. 2.)

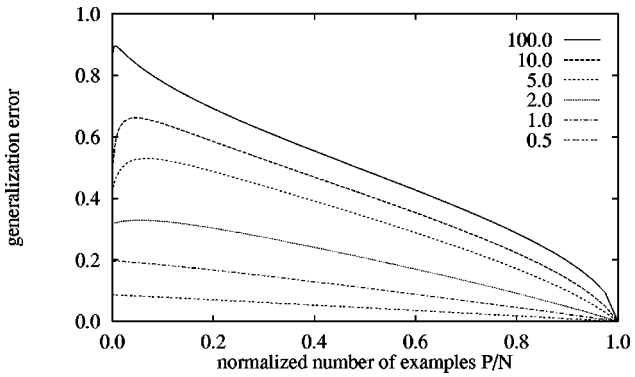

FIG. 6. Performance $E_G(\alpha)$. The generalization error E_G as a function of $\alpha=P/N$, for the problem, tanh perceptron learns tanh perceptron, after exhaustive training. The results for different gains γ of the teacher transfer function $\tanh(\gamma h^*)$ are shown. In this realizable task, exhaustive training can lead to overtraining, if the gain γ is higher than a critical gain γ_c.

of the order parameters is shown for different α. The results of the dynamical approach will change slightly with different choices of the learning rate η. However, they will never exactly coincide with the result of the equilibrium approach. The description of early stopping by the equilibrium approach is therefore not exact, but nevertheless a good and useful approximation. In the case of weight decay, which is not shown in Fig. 5, both approaches yield exactly the same results.

VI. NONLINEAR STUDENT LEARNS REALIZABLE TASK

The knowledge acquired above is now applied to a study of a realizable task, where a nonlinear student must learn an identical nonlinear teacher, i.e., $g(h)=\tanh(h)$ and $g^*(h^*)=\tanh(\gamma h^*)$.

A. Minimal training error

The minimal training error below the storage capacity, $\alpha_c=1$, is always zero. This implies that for every example, the outputs of teacher and student are identical, i.e., $z_\mu^*=z_\mu$. If the transfer function of the student is invertible, then an *alternative loss function* can be used, which has the same minimum,

$$\hat{l}:=\frac{1}{2}[g^{-1}(g^*(h_\mu^*))-h_\mu]^2=\frac{1}{2}[h_\mu^*-h_\mu]^2. \qquad (50)$$

The second equality in Eq. (50) holds only if the transfer functions of teacher and student are identical. By using the alternative loss function, training becomes independent of the transfer function.

The order parameters for the nonlinear realizable task that correspond to the minimal training error $E_T=0$ are the same as the ones for the simplest realizable learning task, where a linear student learns a noise-free linear teacher. They follow from Eq. (21) if we insert $G=H^2=\gamma^2$ and a_0,

$$R=\gamma\alpha, \quad Q=\gamma^2\alpha. \qquad (51)$$

However, the generalization error still depends on the nonlinear functions; see Eq. (9). The resulting behavior of $E_G(\alpha)$ is shown in Fig. 6 for different values of the gain γ.

It is surprising to see that some of the curves actually increase for small values of $\alpha=P/N$. If the gain γ, which is the level of nonlinearity, is higher than some γ_c, then exhaustive training shows overtraining for small α.

The *critical gain* γ_c can be determined by a linear approximation of the nonlinear student. For small α, both order parameters R and Q of Eq. (51) are small, such that tanh function of the student in Eq. (9) can be approximated by a linear function, i.e., $\tanh(\epsilon)\to\epsilon$ for small ϵ. The behavior of the generalization error (9) for small arguments becomes

$$E_G(\epsilon)=E_G(0)-\frac{\epsilon}{2}[2H(\gamma)-\gamma], \qquad (52)$$

with $H(\gamma)$ from Eq. (13).

The slope of E_G at $\alpha=0$ is positive instead of negative if γ becomes larger than $2H(\gamma)$. Since the upper limit of $H(\gamma)$ is $\sqrt{2/\pi}=0.7979$, the critical gain will be smaller then 1.6. The numerical solution gives $\gamma_c=1.3371$.

B. The reason for overtraining

Again we evaluate the generalization error as a function of the two normalized order parameters, i.e., $r:=R/q$ and $q:=\sqrt{Q}$. Figure 7 shows $E_G(r,q)$ for $r\in[0,1]$ and $q\in[0,1.2\gamma]$. We have chosen the gain $\gamma=5$, because it is an intermediate level of nonlinearity and shows the overtraining effect well enough.

As in the unrealizable case, we can project the evolution of the order parameters onto Fig. 7. The learning curve for exhaustive training, see Eqs. (35) and (36), is a straight line with $q(r)=\gamma r$. If the gain γ is higher than γ_c, the isoline $E_G(0,0)$ starts with a lower slope than $q(r)=\gamma r$. The result is overtraining, if exhaustive training is applied. In other words, the network increases the norm of the weights as if the task were a linear one, and this is much too fast for a highly nonlinear task.

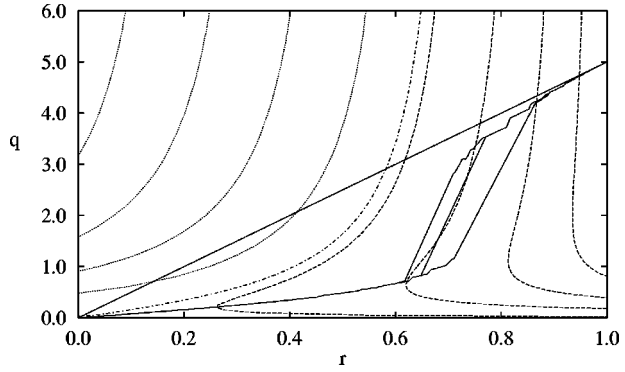

FIG. 7. Shape of generalization error $E_G(r,q)$. Contour plot of the generalization error $E_G(r,q)$ as a function of r and q, similar to that in Fig. 2. From the minimum $E_G^{\min}=0$ at $(r,q)=(1,\gamma)$, the contour lines for $E_G=0.1i$ for $i=1,2,3,4$ (dashed lines) and for $i=6,7,8,9$ (dotted lines) are given. The dashed-dotted line corresponds to $E_G=E_G(0,0)=G/2$. Two learning curves for exhaustive training (upper solid line) and optimal training (lower solid lines) are also shown. (Parameter: gain $\gamma=5$.)

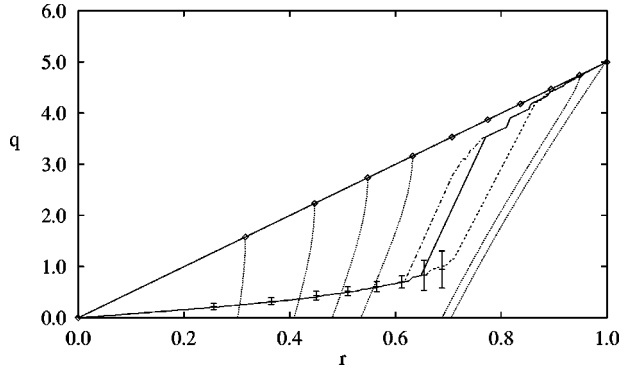

FIG. 8. Evolution of the order parameters. Parametric curves (r,q) as functions of the parameters α and a. Two learning curves, resulting from a variation of α, are shown, which correspond to exhaustive training $a=1$ (straight solid line) and optimal training, a numerically optimized (lower solid line). Marks on the learning curves indicate the values of $\alpha=0.1, 0.2, \ldots, 0.9, 0.99$. Training curves are computed by reducing the parameter a from ∞ to 1 for fixed α. Examples for $\alpha=0.1,0.2,0.3,0.4,0.9,0.99$ (dotted lines) are shown. On the training curves between $\alpha\simeq 0.5$ and 0.7, there are two minima for the generalization error E_G. The resulting hysteresis is shown by the double-dashed and dash-dotted lines. They indicate that the solution stays longer in the first minimum before it jumps to the other one, depending on the initialization. The solid line is the location of the absolute minimum. Results of a simulation of early stopping are given by error bars. (Parameter as in Fig. 7.)

C. Training with errors and early stopping

Now we apply the same method as in Sec. IV. The exact solution for nonlinear training at finite temperatures is not available. Therefore, we assume that the linear order parameters for finite temperatures or nonminimal a can be a useful approximation. These are

$$R(\alpha,a)=\gamma\alpha\frac{1}{a},\quad Q(\alpha,a)=\gamma^2\alpha\frac{a+a\alpha-2\alpha}{a(a^2-\alpha)},\quad (53)$$

with a defined in Eq. (24). The corresponding training curves are plotted in Fig. 8.

The optimal value of a that minimizes E_G can be calculated numerically. The resulting generalization error is plotted in Fig. 9; it shows no overtraining.

An interesting side effect is the hysteresis that appears in the neighborhood of the local minima of $E_G(a)$. Simple early stopping will become stuck in this local minimum, especially since the global minimum jumps quite a distance.

To test the validity of the approximation, we simulate early stopping for this problem, shown in Fig. 9. For validation, the actual generalization error is used. The theoretical approximation is within the range of the error bars of the early stopping simulation.

VII. DISCUSSION

In this work we have developed a model that presents some of the characteristic features of feedforward learning. The approach provides interesting insights, especially in unrealizable tasks, where overtraining and a finite residual error appear. Yet, it is simple enough to allow an analytical description.

We have shown that overtraining can be avoided completely, if an optimal training strategy is applied. Several strategies have been discussed, each of which can obtain the same optimal effect. Either training is stopped at a finite training error before the system overspecializes on the examples, or the weights are reduced in each training step, or alternatively input noise is added to the examples. If the respective parameter of the these methods, stopping time t, weight decay strength λ, or noise level δ, is optimally chosen, then overtraining can be avoided and the system decreases monotonously to the residual error.

The occurrence of overtraining in the realizable, nonlinear task (see Sec. VI) was a counterexample to the widespread belief that overtraining appears only in unrealizable tasks. We have shown that overtraining can also be caused by a high level of nonlinearity.

The new interpretation of the finite temperature solution of the equilibrium statistical mechanics approach is also interesting, from a technical point of view. It should be possible to apply it to other problems.

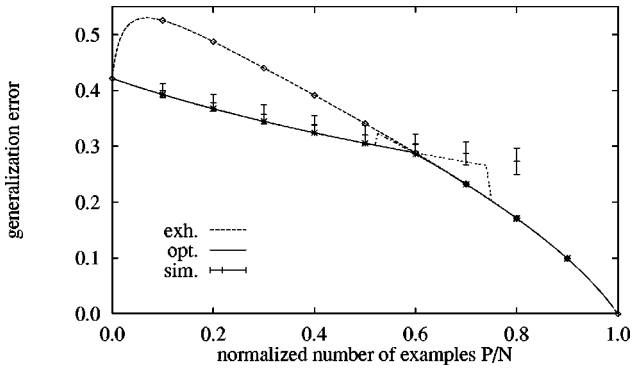

FIG. 9. Optimized performance. Comparison of the performance $E_G(\alpha)$ after exhaustive training (dashed line) and after optimized training (solid line). The marks show the values of $\alpha=0.1, 0.2, \ldots, 1.0$. Simulation results of early stopping are indicated by error bars. (Parameters as in Fig. 7.)

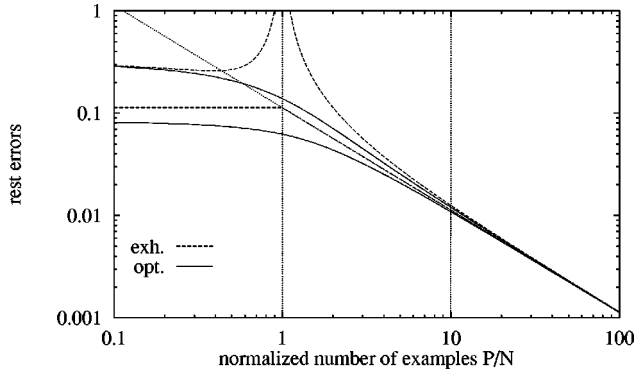

FIG. 10. Different regimes of learning. The performance of the rest generalization error $E_G - E_\infty$ (upper two lines) and the rest training error $E_\infty - E_T$ (lower lines) is split into characteristic regimes. Exhaustive training (dashed lines) and optimal training (solid lines) are shown. The dotted diagonal line is the asymptotical scaling E_∞/α. Three regimes, storage regime, generalization regime, and asymptotical regime are divided by dotted vertical lines, they are discussed in the text. (Parameters as in Fig. 2.)

We also believe that the results provide some insight into early stopping, weight decay, and input noise. While the characteristics of these methods are correctly described, it should be noted that the finer details such as the equivalence of early stopping, weight decay, and input noise, are consequences of the linearity in the model and will not hold in general.

We believe that the results achieved on the rather simple linear perceptron display some characteristic features of learning in feedforward networks. As an example, we will discuss the regimes in the learning curves, see Fig. 10.

The learning curves split at the storage capacity $\alpha_c = 1$ into two regimes, storage regime and generalization regime. In the *storage regime*, below $\alpha_c = 1$, all examples are learned by heart, resulting in a zero training error. However, the generalization ability is neglected in this regime, resulting in overtraining. In the *generalization regime* above α_c, a surplus of examples makes perfect learning impossible. The network is forced to generalize in order to minimize the training error. Information on the whole task is extracted from the examples and the generalization error decreases.

It is useful to define an additional subregime, which we call *asymptotical regime* stretching from α_{asy} to infinity. In the asymptotical regime, the number of examples is sufficiently large to make some simplifying assumptions valid. For example, it can be assumed that E_T approximates E_G well enough, such that the difference between exhaustive training and optimal training becomes negligible. Furthermore, both rest errors possess the same asymptotical convergence rate, which is α^{-1} in this model. In our example, α_{asy} is of the order 10, as shown in Fig 10.

These regimes are characteristic for learning in neural networks. For more general models only a few results are well studied, such as the asymptotical behavior [18]. Also the negligible effect of early stopping in the asymptotical range was found for general learning scenarios [19]. It should be emphasized that a much richer behavior is located below the asymptotical regime, which can already be seen in our rather simple single-layer model.

While detailed analytical studies for the more complicated, nonlinear, multilayer networks may not be possible, extensive simulations should always be accessible. For a first step in this direction, see [20]. A more thorough understanding of the whole training process, including the effects of early stopping and weight decay would be desirable, if neural networks shall become a useful and reliable tool for learning functional relations.

ACKNOWLEDGMENTS

I would like to thank S. Amari, D. Bollé, W. Kinzel, R. Kühn, J. van Mourik, K. R. Müller, and especially M. Opper for useful discussions at different stages of this work. I want to thank E. Helle and P. Pedroso for support concerning the presentation.

[1] Some references on-line are O. Kinouchi and N. Caticha, J. Phys. A **25**, 6243 (1992); M. Biehl and H. Schwarze, *ibid.* **26**, 2651 (1993); D. Saad and S. A. Solla, Phys. Rev. E **52**, 4225 (1995); S. Bös, N. Murata, S. Amari, and K.-R. Müller, Brain Science Institute RIKEN Technical Report No. TR-BSI-1997-21.

[2] H. Akaike, IEEE Trans. Autom. Control. **19**, 716 (1974).

[3] N. Murata, S. Yoshizawa, and S. Amari, in *Advances in Neural Information Processing Systems 5*, edited by S. J. Hanson, J. D. Cowan, and R. P. Lippmann (Morgan-Kaufmann, San Mateo, CA, 1993), p. 607.

[4] D. J. C. MacKay, Neural Comput. **4**, 415 (1992); **4**, 448 (1992).

[5] L. Holmström and P. Koistinen, IEEE Trans. Neural Netw. **3**, 24 (1992).

[6] J. Sjöberg and L. Ljung, report.

[7] R. Reed, R. J. Marks II, and S. Oh, IEEE Trans. Neural Netw. **6**, 529 (1995).

[8] P. Sollich, in *Advances in Neural Information Processing Systems 7*, edited by G. Tesauro, D. Touretzky, and T. Leen (The MIT Press, Cambridge, MA, 1995), p. 207.

[9] A. Krogh and J. Hertz, J. Phys. A **25**, 1117 (1992); **25**, 1135 (1992); in *Advances in Neural Information Processing Systems 4*, edited by J. E. Moody, S. J. Hanson, and R. P. Lippmann (Morgan-Kaufmann, San Mateo, CA, 1992), p. 950.

[10] J. Hertz, in *Statistical Mechanics of Neural Networks*, edited by L. Garrido, Lecture Notes in Physics Vol. 398 (Springer, Heidelberg, 1990), p. 137.

[11] R. Kühn and S. Bös, J. Phys. A **26**, 831 (1993).

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

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

[14] Preliminary results of this work have been published by S. Bös, in *International Conference on Artificial Neural Networks*

95, edited by F. Fogelman Soulié and P. Gallinari (EC & Cie, Paris, 1985), p. 111; S. Bös, in *Advances in Neural Information Processing Systems 8*, edited by D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (The MIT Press, Cambridge, MA, 1996), p. 218; S. Bös, in *International Conference on Artificial Neural Networks 96*, edited by C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, Lecture Notes in Computer Sciences Vol. 1112 (Springer, Berlin, 1996), p. 551.

[15] K. H. Fischer and J. A. Hertz, *Spin Glasses* (Cambridge University Press, Cambridge, 1991).

[16] S. Bös and M. Opper, in *Advances in Neural Information Processing Systems 9*, edited by M. C. Mozer, M. I. Jordan, and T.Petsche (The MIT Press, Cambridge, MA, 1997), p. 141; S. Bös and J. Opper, J. Phys. A (to be published).

[17] B. Widrow and M. E. Hoff, Jr., in *1960 IRE WESCON Convention Record*, Part 4 (IRE, New York, 1960), p. 96.

[18] H. S. Seung, H. Sompolinsky, and N. Tishby, Phys. Rev. A **45**, 6056 (1992); S. Amari, N. Fujita, and S. Shinomoto, Neural Comput. **4**, 605 (1992).

[19] S. Amari, N. Murata, K.-R. Müller, M. Finke, and H. Yang, IEEE Trans. Neural Netw. **8** 985 (1997).

[20] K.-R. Müller, M. Finke, N. Murata, K. Schulten, and S. Amari, Neural Comput. **8**, 1085 (1996).