Selection of examples for a linear classifier

1996 J. Phys. A: Math. Gen. 29 1367

(http://iopscience.iop.org/0305-4470/29/7/010)

J. Phys. A: Math. Gen. **29** (1996) 1367–1380. Printed in the UK

Selection of examples for a linear classifier

Georg Jung† and Manfred Opper‡

† Physikalisches Institut, Julius-Maximilians-Universität Am Hubland, D-97074 Würzburg, Germany

‡ The Baskin Centre for Computer Engineering & Information Sciences, University of California, Santa Cruz, CA 95064, USA

Received 10 November 1995

Abstract. We investigate the problem of selecting an informative subsample out of a neural network's training data. Using the replica method of statistical mechanics, we calculate the performance of a heuristic selection algorithm for a linear neural network which avoids overfitting.

1. Introduction

Finding the optimal complexity of a neural network for learning an unknown task is one of the most interesting problems in the theory of neural computation. A popular strategy is to start with a complex network having more connections than are actually needed. Then, after training, by deleting couplings which are too small or seem to be of less importance, one hopes to end up with a network which has reasonable performance (see e.g. [23]). The strategy of clipping network weights has also been investigated in the framework of statistical mechanics (see e.g. [20, 24, 25]).

In this paper, we will look at a problem, which is in some sense dual to the problem of pruning the network weights. We consider the problem of pruning the training examples. Besides its interest from a purely theoretical viewpoint, such a problem is motivated from the so-called phenomenon of *overfitting* in network learning. If the complexity of a network is too high, it may be able to fit all training examples perfectly, but the probability of predicting the outputs on new data may drop to the value of random guessing. As a result, the learning curve, which displays the generalization error as a function of the number of examples, may show a non-monotonic behaviour. This means that an increase in the number of examples can in some regions lead to a decrease in generalization abilities.

Recently, Garces has found in numerical studies that overfitting can be avoided by deleting examples from the training set [18]. In this respect, it is interesting to find out from which selection of examples the performance of the network will benefit most. In this paper, we will study a heuristic strategy for the selection of examples in the case of a linear classifier applied to two toy problems. From an analytical viewpoint, it is simple enough to be free of the problems of replica symmetry breaking.

This approach should not be confused with the well studied problem of *learning with queries* [14–16]. In the latter case, one selects *inputs* with respect to some criterion, *before* their outputs are observed. In our case, all training examples, i.e. both inputs and outputs, are known to the learner.

0305-4470/96/071367+14$19.50

This paper is organized as follows. In the second section, we briefly review some properties of the Adaline algorithm which will be basic to our treatment. The third section explains the heuristic strategy for the example selection. In section four, we introduce two different rules to be learnt by the network. Section five contains a new approach to the statistical mechanics of the problem. Finally, in sections six and seven, the results of our calculations are presented and discussed.

2. Adaline learning

As has been shown, e.g. in [9, 3, 22, 10], the effect of overfitting can already be observed for the case of a simple linear classifier, the so-called Adaline model [13, 12, 21], which we will discuss in the following sections. For any N-dimensional input vector $\boldsymbol{\xi}$ this linear classifier is defined by the output

$$S = \frac{1}{\sqrt{N}} \boldsymbol{J} \cdot \boldsymbol{\xi} \, . \tag{1}$$

For p linearly independent input vectors $\boldsymbol{\xi}^\mu$, and arbitrary real-valued target outputs S^μ, $\mu = 1, \ldots, p$, the system of equations

$$S^\mu = \frac{1}{\sqrt{N}} \boldsymbol{J} \cdot \boldsymbol{\xi}^\mu \tag{2}$$

will always have solutions. Hence, for random inputs, where linear dependencies are unlikely, it will be possible to adjust the vector $\boldsymbol{J}$ of the N network weights J_j, $j = 1, \ldots, N$ in such a way that if $p < N$, all training examples are perfectly learnt by the network.

An explicit solution for such a weight vector is given by the pseudo-inverse solution (PSI) [11]:

$$\boldsymbol{J} = \sum_{\mu,\nu=1}^{\alpha N} S_B^\mu (C^{-1})_{\mu\nu} \boldsymbol{\xi}^\mu \tag{3}$$

where $C_{\mu\nu} = \frac{1}{N} \sum_j \xi_j^\mu \xi_j^\nu$ is the correlation matrix of the training patterns. Out of the linear space of solutions to equation (2), this is the one which has minimal squared norm $q_0 = \boldsymbol{J}^2/N$.

The restriction of our treatment to the case $p < N$ is mainly for mathematical convenience. However, especially in this region, the effect of overfitting can be observed.

3. Weighting of examples

Our basic strategy for the selection of examples is based on the fact that most iterative learning algorithms for a single-layer net result in a coupling vector which has the form of a *weighted* Hebbian rule [4, 9]. To see this, consider a learning algorithm of the *backpropagation* type which is based on the minimization of a quadratic training energy

$$E = \tfrac{1}{2} \sum_{\mu=1}^{\alpha N} (S_B^\mu - g_J(h_J^\mu))^2 \tag{4}$$

by gradient descent. Here we assume a smooth output $g_J(h_J^\mu)$ of the student net, which is defined through the internal field $h_J^\mu = \frac{1}{\sqrt{N}} \boldsymbol{J} \cdot \boldsymbol{\xi}^\mu$. Hence, during a learning step, the network couplings are changed by an amount

$$\delta J_j \propto -\frac{\partial E}{\partial J_j} \, . \tag{5}$$

It is not hard to show that the algorithm (5) for $g_J(h^\mu) = h^\mu$, when started with a zero initial vector, converges to the Adaline rule (3).

To rewrite (5) as a weighted Hebbian sum, we set

$$\delta J_j = \sum_{\mu=1}^{\alpha N} \delta x_\mu S_B^\mu \xi_j^\mu \tag{6}$$

where

$$\delta x_\mu(t) \propto \sum_{\mu=1}^{\alpha N} (1 - (S_B^\mu)^{-1} g_J) \frac{\partial g_J}{\partial h_J^\mu} \tag{7}$$

represents the weighting (in the following called the embedding strength) of the μth example in the tth learning step.

Our selection of examples will be based on the following heuristic ansatz: intuitively, we will expect that examples which have small or even negative total embedding strengths $x_\mu = \sum_t \delta x_\mu(t)$ after learning could be cast out of the training set. The latter would have an output that already has the correct sign without being learnt. This strategy may also be understood as an approximation to the AdaTron algorithm [5] which, by construction, allows for non-negative embedding strengths only. Another possibility was discussed in a paper by Garces [18] for the case of Adatron learning. Only examples which are not too hard to learn, i.e. which have positive embedding strengths below a certain value, were left in the training set.

To find an expression for the total embedding strengths, it is not necessary to solve the dynamics (5). We can get the same information directly from the final coupling vector. For the Adaline case, with $\alpha = p/N < 1$ the form of the coupling vector (3) provides us with an explicit expression for the embedding strengths which is given by

$$x_\mu = \frac{1}{S_B^\mu} \sum_\nu S_B^\nu (C^{-1})_{\mu\nu}$$

valid for $p < N$. To treat the statistical mechanics of the problem, however, a simpler implicit definition of x_μ using a suitable Lagrangian will be given in section 5. Although it is possible to obtain the coupling vector of the linear classifier for $\alpha > 1$, useful expressions for the corresponding embedding strengths are harder to find and to treat within the framework of statistical mechanics. Hence, we will restrict ourselves to the region $\alpha < 1$.

In order to keep the subsequent analysis simple, we will not assume that the Adaline algorithm has to be rerun for a second time on the reduced training set. We will rather make the ansatz, that the new weight vector is given *a priori* via single-shot learning by a weighted Hebbian sum of the form

$$\tilde{\boldsymbol{J}} = \frac{1}{\sqrt{N}} \sum_\mu f(x_\mu) S_B^\mu \boldsymbol{\xi}^\mu . \tag{8}$$

In the following, we will consider two choices for the weighting function $f(x)$. One choice will be called the *modified Adaline* rule $f(x) = x\Theta(x)$, where examples with negative embedding strengths are abandoned, and the positive ones keep their original weights. This may be considered as an approximation to the more complicated algorithm of relearning the remaining examples with the Adaline method. For comparison, we will also study the simpler choice $f(x) = \Theta(x)$ (*modified Hebbian rule*), where the remaining examples have equal weights.

4. Learning tasks

As in most theoretical studies of neural networks, the task to be learnt will be defined by a teacher network. It provides the correct outputs S_B for a given input $\boldsymbol{\xi}$. For the simplest case, the teacher is given by a single layer network with a fixed vector of couplings $\boldsymbol{B}$ and output

$$S_B = g_B(h_B) \tag{9}$$

with $h_B = \frac{1}{\sqrt{N}}\boldsymbol{B}\cdot\boldsymbol{\xi}$. The complexity of such teacher networks can be tuned by suitably varying the output function g_B. In all the following cases, an explicit mismatch between teacher and student is assumed by chosing g_B as a nonlinear function. For simplicity, we specialize on the binary case $g_B^2 = 1$. Hence, even when the teacher problem is of the simple type $g_B(h) = \mathrm{sign}(h)$, the linear output function of the student used for the training process will cause overfitting. For better comparison with the teacher net, we will measure the performance of the student network after training by its clipped output $\mathrm{sign}(h_J)$. Hence, we define the generalization error as the following average:

$$\epsilon_{\mathrm{g}} = \tfrac{1}{2}\langle|g_B(h_B) - \mathrm{sign}(h_J)|\rangle_{\xi}\,. \tag{10}$$

In the simplest case, where the inputs are drawn from a spherically symmetric density,

$$f(\boldsymbol{\xi}) = (2\pi)^{-N/2}\mathrm{e}^{-\frac{1}{2}|\boldsymbol{\xi}|^2}$$

it is possible to describe the generalization error in terms of the angle ϕ between the two vectors $\boldsymbol{B}$ and $\tilde{\boldsymbol{J}}$.

$$\phi = \frac{\boldsymbol{B}\cdot\tilde{\boldsymbol{J}}}{|\tilde{\boldsymbol{J}}|\cdot|\boldsymbol{B}|} = \tilde{R}/\sqrt{\tilde{q}_0}\,.$$

Here $\tilde{R} = \boldsymbol{B}\cdot\tilde{\boldsymbol{J}}/N$ defines the overlap between teacher and student, and $\tilde{q}_0 = \tilde{\boldsymbol{J}}^2/N$. For simplicity, we have chosen the norm of the teacher to be $\sqrt{N}$.

We will investigate two different rules.

(i) We begin with a linearly separable task, defined by a teacher perceptron with a non-zero threshold τ. The output function is

$$g_B(h_B) = \mathrm{sign}(h_B - \tau)\,.$$

The shaded regions in figure 1 display the fraction of input space where the answers from teacher and student differ. The generalization error can be easily calculated to be

$$\epsilon_{\mathrm{g}} = \frac{1}{\pi}\cos^{-1}\left(\frac{\tilde{R}}{\sqrt{\tilde{q}_0}}\right) + \int_0^{\tau}\mathrm{D}h\left[2\Phi\left(\frac{\tilde{R}h}{\sqrt{\tilde{q}_0 - \tilde{R}^2}}\right) - 1\right] \tag{11}$$

where Dh is the Gaussian measure:

$$\mathrm{D}h = \frac{\mathrm{d}h}{\sqrt{2\pi}}\mathrm{e}^{-h^2/2} \qquad \text{and} \qquad \Phi(x) = \int_{-\infty}^{x}\mathrm{D}h.$$

(ii) The second rule is the so-called reversed-wedge problem [19]:

$$g_B(h_B) = \mathrm{sign}(h_B(h_B - \tau)^2)\,.$$

The shaded regions in figure 2 display in the same way as before the fraction of input space where the answers from teacher and student differ. The generalization error can be calculated similarly:

$$\epsilon_{\mathrm{g}} = \frac{1}{\pi} \cos^{-1}\left(\frac{\tilde{R}}{\sqrt{\tilde{q}_0}}\right) + 2\int_0^{\tau} \mathrm{D}h \left[2\Phi\left(\frac{\tilde{R}h}{\sqrt{\tilde{q}_0 - \tilde{R}^2}}\right) - 1\right]. \tag{12}$$

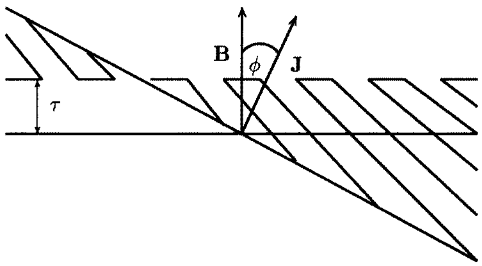

Figure 1. Geometry of input space for perceptron with threshold.

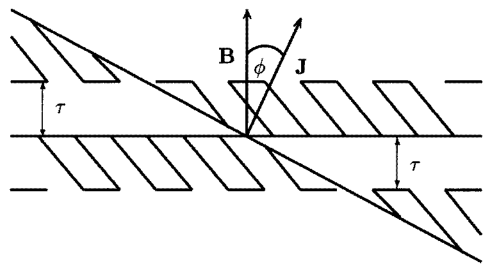

Figure 2. Geometry of input space for reversed-wedge problem.

5. Statistical mechanics: a Lagrangian approach

Following the approach of Elizabeth Gardner [1, 2], the application of statistical mechanics to network learning (for a review see [6–8]) is often based on the fact that the network configurations J_j obtained from a learning algorithm are minima of a suitable training energy E. In this case, by introducing a canonical ensemble of networks at temperature β^{-1}, the desired configuration appears as the one with maximal weight in the partition function $Z = \int \mathrm{d}J\, \mathrm{e}^{-\beta E}$, in the limit $\beta \to \infty$.

This procedure works fine [3], e.g. in order to determine the order parameters of the standard Adaline rule, which are explicit functions of the couplings. However, it does not give us any direct information on the embedding strengths, which are only implicitly related to the couplings. One possibility to obtain these quantities would be to introduce their explicit definitions within δ functions, see e.g. [17]. In our paper, we will use a new, technically more elegant, approach which is based on a Lagrangian formulation of the optimization problem rather than on a Hamiltonian E.

We will make explicit use of the fact that the coupling vector is the solution of the following constrained optimization problem: minimize $\frac{1}{2}\boldsymbol{J}^2$ under the constraints

$$1/\sqrt{N}\, \boldsymbol{J} \cdot \boldsymbol{\xi}^{\mu} = S_B^{\mu} \qquad \forall \mu\,.$$

By introducing Lagrange parameters x_μ together with the Lagrange function

$$\mathcal{L}(\boldsymbol{J}, \{S_B^{\mu} x_\mu\}) = \tfrac{1}{2}\boldsymbol{J}^2 - \sum_{\mu} S_B^{\mu} x_\mu \left(\frac{1}{\sqrt{N}} \boldsymbol{J} \cdot \boldsymbol{\xi}^{\mu} - S_B^{\mu}\right) \tag{13}$$

we can solve the optimization problem by finding the point in $(N + P)$-dimensional space of J_j's and x_μ's where $\mathcal{L}$ is stationary. Setting the partial derivative of $\mathcal{L}$ with respect to the J_j's equal to zero, we see that

$$\boldsymbol{J} = \frac{1}{\sqrt{N}} \sum_{\mu} x_\mu S_B^{\mu} \boldsymbol{\xi}^{\mu}\,. \tag{14}$$

Thus, the x_μ's actually coincide with the embedding strengths. However, the stationary point of $\mathcal{L}$ is a *saddle point*, not a minimum. In order to keep integrals finite, we will work with the following complex partition function:

$$Z = \int \prod_j \mathrm{d}J_j \prod_\mu \mathrm{d}y_\mu \exp\left[-\beta\mathcal{L}(\boldsymbol{J}, \{\mathrm{i}y_\mu\}\right]. \tag{15}$$

Using the saddle-point method, by suitably deforming the contour of integration of the y_μ's, we find that in the limit $\beta \to \infty$, the dominant contribution to Z comes from the stationary point of the Lagrangian $\mathcal{L}$.

Using the complex distribution in Z, we are able to calculate any average of functions of the embedding strengths by identifying y_μ with $-\mathrm{i}S_B^\mu x_\mu$ at the saddle point. We are particularly interested in the distribution $W(\tilde{J}_l)$ of an arbitrary component of the new student vector $\tilde{J}_l(\{x_\mu\})$. This enables us to calculate the order parameters necessary to describe the generalization ability of a network with the new couplings $\tilde{J}_l$. The characteristic function $\tilde{\omega}(k) = \left\langle \mathrm{e}^{\mathrm{i}k\tilde{J}_l(\{x_\mu\})}\right\rangle_\xi$ of this random variable is expressed as a further average over the complex distribution defined by the partition function (15). We will denote this average by $\langle\cdots\rangle_\beta$ and get

$$\tilde{\omega}(k) = \lim_{\beta\to\infty}\left\langle\left\langle \exp\left[\mathrm{i}k\tilde{J}_l\left(\left\{\mathrm{i}\frac{y_\mu}{S_B^\mu}\right\}\right)\right]\right\rangle_\beta\right\rangle_\xi \tag{16}$$

$$= \lim_{\beta\to\infty}\left\langle Z^{-1}\int_{-\infty}^{+\infty}\prod_j \mathrm{d}J_j \prod_\mu \frac{\mathrm{d}y_\mu}{\sqrt{2\pi}} \exp\left(-\frac{\beta}{2}\boldsymbol{J}^2 + \mathrm{i}\beta\sum_\mu y_\mu\left(\frac{1}{\sqrt{N}}\boldsymbol{J}\cdot\boldsymbol{\xi}^\mu - S_B^\mu\right) + \mathrm{i}k\tilde{J}_l\right)\right\rangle_\xi. \tag{17}$$

Introducing n replicas and the local field of the teacher, we see that in the $n \to 0$ limit only the lth component of the student vector contributes:

$$\tilde{\omega}(k) = \lim_{\beta\to\infty,\ n\to 0}\left\langle \int_{-\infty}^{+\infty}\prod_a \mathrm{d}J_{la}\prod_{\mu,a}\frac{\mathrm{d}y_\mu^a}{\sqrt{2\pi}}\prod_\mu\frac{\mathrm{d}h^\mu\,\mathrm{d}v^\mu}{2\pi}\exp\left(-\frac{\beta}{2}\sum_a J_{la}^2 - \mathrm{i}\beta\sum_{\mu,a} g_B(h^\mu)y_\mu^a\right.\right.$$
$$\left.\left. -\mathrm{i}\sum_\mu h^\mu v^\mu + \frac{\mathrm{i}}{\sqrt{\mathrm{N}}}\sum_\mu \xi_l^\mu\left(\beta\sum_a J_{la}y_\mu^a + B_l v^\mu + kf\left(\frac{\mathrm{i}y_\mu^b}{g_B(h^\mu)}\right)g_B(h^\mu)\right)\right)\right\rangle_\xi.$$

After averaging and introducing appropriate order parameters, assuming replica symmetry, we obtain in the limit $n \to 0$

$$\tilde{\omega}(k) = \exp\left(-\frac{k^2}{2}A + \mathrm{i}kB_l b\right) \tag{18}$$

with $A = \alpha F + 2\alpha^2\gamma G/\delta - \alpha^3 Q\gamma^2/\delta^2$ and $b = \alpha T - \alpha^2\gamma U/\delta$. These constants are defined by the following order parameters, which again can be calculated within the replica framework:

$$\begin{aligned} Q_{ab} &= \langle y^a y^b\rangle \qquad & F &= \left\langle f^2\left(\frac{\mathrm{i}y}{g_B(h)}\right)g_B^2(h)\right\rangle \\ G_{ab} &= \left\langle \mathrm{i}y^a f\left(\frac{\mathrm{i}y^b}{g_B(h)}\right)g_B(h)\right\rangle \qquad & U &= \langle yv\rangle \\ S &= \langle \mathrm{i}y g_B(h)\rangle \qquad & T &= \left\langle \mathrm{i}v f\left(\frac{\mathrm{i}y}{g_B(h)}\right)g_B(h)\right\rangle. \end{aligned} \tag{19}$$

Finally, $\gamma = \beta(G_{aa} - G_{ab})$ and $\delta = 1 + \alpha\beta(Q_{aa} - Q_{ab})$.

Explicit expressions for these quantities are given in appendix B. By a Fourier transform of (18) we obtain the Gaussian distribution

$$W(\tilde{J}_l) = \frac{1}{\sqrt{2\pi A}} \exp\left(-\frac{(\tilde{J}_l - B_l b)^2}{2A}\right). \tag{20}$$

Hence, using the self-averaging property of order parameters, the overlap of the new weight vector with the teacher and the corresponding norm are given by

$$\tilde{R} = \frac{1}{N}\sum_{l=1}^{N} \langle \tilde{J}_l \rangle B_l = b$$

and

$$\tilde{q}_0 = \frac{1}{N}\sum_{l=1}^{N} \langle \tilde{J}_l^2 \rangle = A + b^2 .$$

Using these order parameters, the generalization error for the different tasks of section 4 can be calculated from the results of section 4.

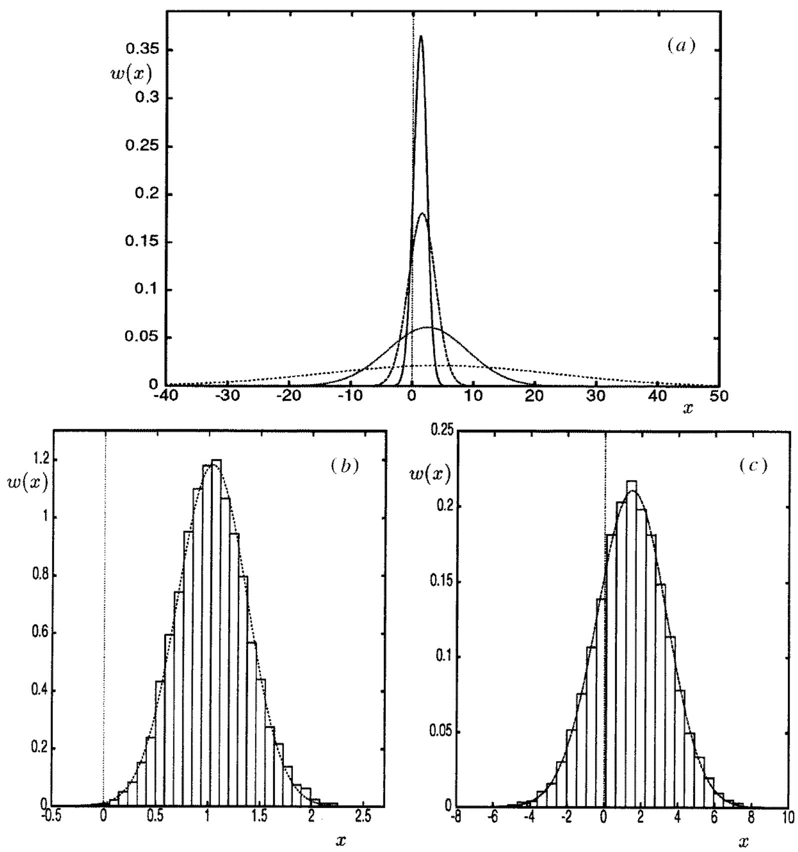

Figure 3. Distribution of embedding strengths for signum teacher with threshold $\tau = 0$. (*a*) Theoretical results for $\alpha = 0.4$, 0.6, 0.8, 0.9 (upper to lower curves), (*b*) theoretical results (broken) and simulations (histogram) for $\alpha = \frac{30}{321}$, (*c*) same as (*b*) but for $\alpha = \frac{180}{321}$.

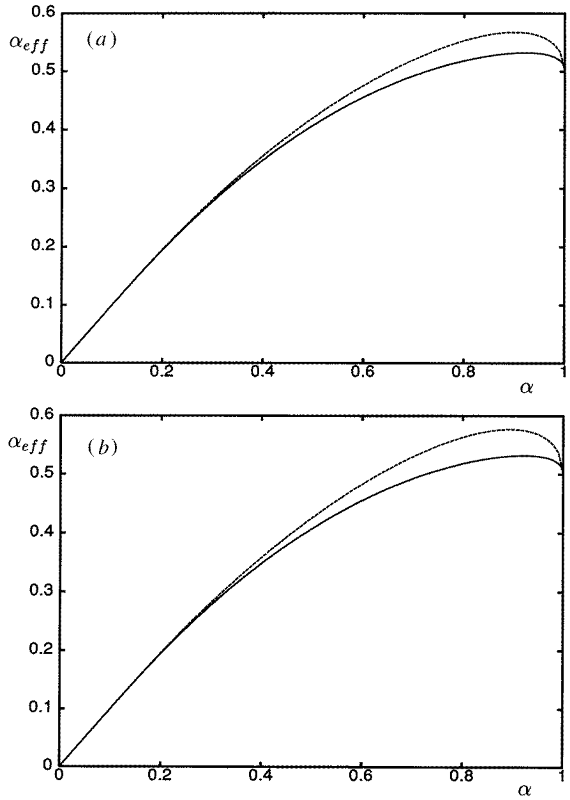

Figure 4. Relative size $\alpha_{\rm eff} = p'/N$ of the pruned training set, versus relative size $\alpha = p/N$ of the original training set for (*a*) perceptron with threshold τ (full curve, $\tau = 0$; broken curve, $\tau = 2.4$) and (*b*) reversed-wedge problem (full curve, $\tau = 0$; broken curve, $\tau = 1.2$).

6. Results

In this section we present the learning curves of our algorithms applied to the learning tasks of section 4.

For both learning tasks, the distribution $P(x)$ of embedding strengths (see figure 3 and appendix A) broadens as α increases and shows an increasing fraction of negative x_μ. Hence, using our algorithm, a mostly increasing fraction of examples, which approaches $\frac{1}{2}$ as $\alpha \to 1$, will be cast out of the training set. In figure 4, we have displayed the relative number of remaining examples $\alpha_{\rm eff} = p'/N$ examples, for both learning tasks. Here

$$\alpha_{\rm eff} = \alpha \int_0^{+\infty} P(x)\,{\rm d}x\,.$$

Figure 5 shows the generalization error for the modified Adaline algorithm with weight function $f(x) = x\Theta(x)$, learning a perceptron with threshold τ. The broken curve was obtained for the standard Adaline algorithm, where for $\alpha \to 1$ only the trivial generalization $\varepsilon_{\rm g} = \frac{1}{2}$ is achieved. Obviously, by selecting examples, this overfitting phenomenon vanishes. The little symbols on the curves are results of numerical simulations of the algorithm. Since the modified algorithm achieves smaller errors than the minimum of the broken curve, it performs better than an Adaline algorithm applied to a random selection of $\alpha_{\rm eff}N$ examples. It is interesting to note that both $\tilde{R}$ and $\tilde{q}_0$ diverge for $\alpha \to 1$ as in the case of normal Adaline learning [3]. The ratio $\tilde{R}/\sqrt{\tilde{q}_0}$, however, which enters the formulae for $\varepsilon_{\rm g}$, remains finite.

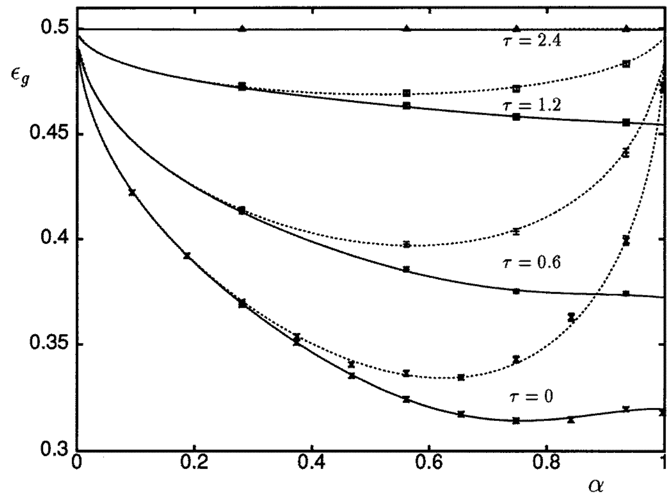

Figure 5. Generalization error for modified Adaline algorithm (full curves) compared with regular Adaline (broken curves). Task: perceptron with threshold. The symbols with error bars represent results achieved by a perceptron with $N = 321$ input units averaged over 100 draws of different learning sets.

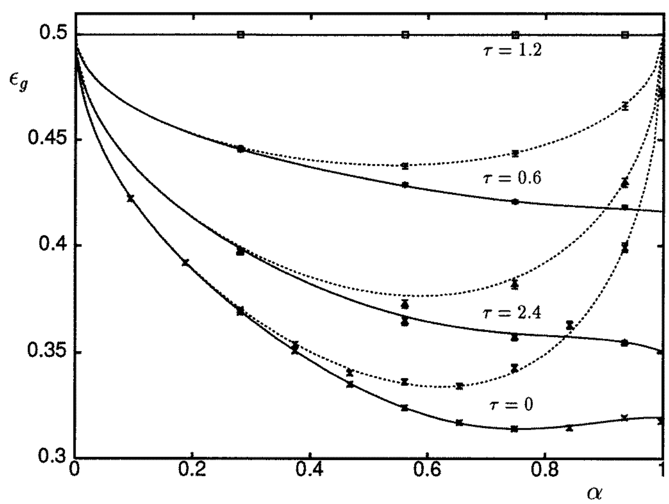

Figure 6. Same as in figure 5, but for the reversed-wedge problem.

For the same task, figure 7 shows the result for the modified Hebb method $f(x) = \Theta(x)$. This yields, except for α close to 1, a slight reduction of performance compared to the modified Adaline case. The broken curves are the results for Hebbian learning of a *random* selection of the same number of examples $p' = \alpha_{\text{eff}} N$. However, both sets of curves are displayed as functions of the initial size α of the training set. This comparison again proves that the selected examples contain more information about the learning task than a random subset of examples.

Figures 6 and 8 display the performance of the algorithms in the case of the reversed-wedge problem. The overall behaviour is roughly the same as in the case of the threshold perceptron.

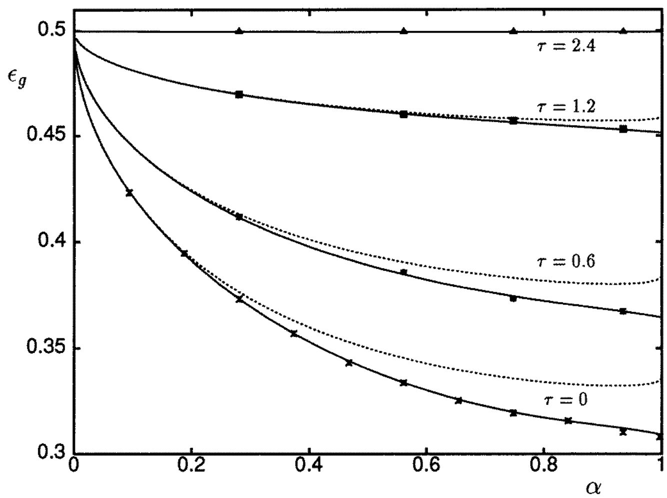

Figure 7. Generalization error for modified Hebb learning (full curves). Task: a perceptron with threshold. The broken curves show regular Hebb learning using a *random* selection of $p' = \alpha N$ examples. Simulations are done analogously to figure 5.

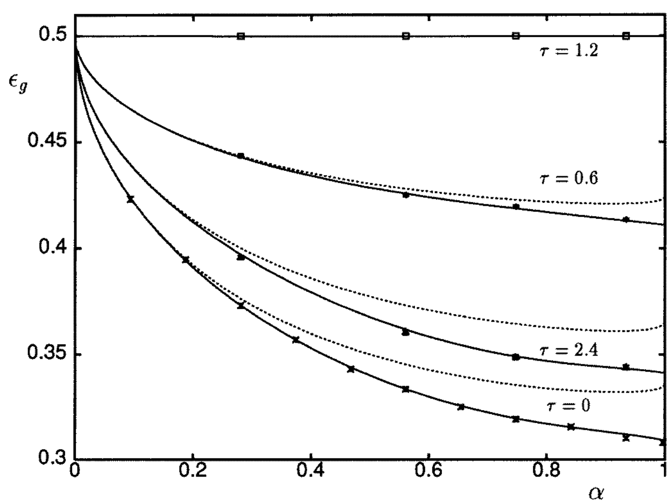

Figure 8. Same as in figure 7, but for the reversed-wedge problem. In most cases the modified Hebb rule achieves better results than the modified Adaline algorithm except for the realizable problem $\tau = 0$.

7. Conclusion

Using statistical mechanics, we have analysed a simple strategy for pruning the training set of examples in the case of a linear classifier network. By rewriting the coupling vector as a Hebbian sum, the natural concept of the weight of an example is introduced. Deleting the examples with negative weights from the training set avoids the overfitting phenomenon. By using different weight functions $f(x)$, our analysis might be extended to other types of example selections, e.g. erasing the ones which are too hard to learn [18]. In this case, we expect that the network's performance will benefit most from such a procedure when α is sufficiently larger than 1. However, such an analysis requires a different mathematical treatment than that of section 5. Further, it might be interesting to see whether similar strategies can be developed for more complicated networks.

Acknowledgments

We are grateful to Wolfgang Kinzel and Ronny Meir for stimulating discussions. G Jung was supported by a DFG grant and M Opper by a Heisenberg fellowship of DFG.

Appendix A. Distribution of embedding strengths

In this appendix we will briefly derive the calculation of the distribution of embedding strengths x_μ. Defining the characteristic function

$$\omega(k) = \left\langle \mathrm{e}^{\mathrm{i}kx_\nu} \right\rangle_\xi = \left\langle \lim_{\beta\to\infty} \left\langle \mathrm{e}^{-ky_\nu/S_B^\nu} \right\rangle_\beta \right\rangle_\xi \tag{A1}$$

and introducing replicas one gets

$$\begin{aligned}\omega(k) = \lim_{\beta\to\infty, n\to 0} \Bigg\langle \int_{-\infty}^{+\infty} \prod_{j,a} \mathrm{d}J_{ja} \prod_{\mu,a} \mathrm{d}y_\mu^a \prod_\mu \frac{\mathrm{d}h^\mu\,\mathrm{d}v^\mu}{2\pi} \\ \times \exp\Bigg(-\frac{\beta}{2}\sum_{j,a} J_{ja}^2 - \mathrm{i}\sum_\mu h^\mu v^\mu + \frac{\mathrm{i}}{\sqrt{\mathrm{N}}}\sum_{\mu,j}\xi_j^\mu \Big(\beta\sum_a J_{ja}y_\mu^a + B_j v^\mu\Big) \\ - \frac{ky_\nu^b}{g_B(h^\nu)} - \mathrm{i}\beta\sum_{\mu,a} y_\mu^a g(h^\mu)\Bigg)\Bigg\rangle_\xi .\end{aligned}$$

Upon averaging over the inputs and defining order parameters $R_a = \langle BJ_a \rangle$ and $q_{ab} = \langle J_a J_b \rangle$, we obtain in replica symmetry

$$\begin{aligned}\omega(k) &= \lim_{\beta\to\infty, n\to 0} \int_{-\infty}^{+\infty} \prod_\mu \mathrm{D}h^\mu \prod_{\mu,a} \mathrm{d}y_\mu^a \exp\Bigg(-\frac{\beta}{2}\chi\sum_{\mu,a}(y_\mu^a)^2 \\ &\quad -\frac{\beta^2}{2}(q-R^2)\sum_\mu\Big(\sum_a y_\mu^a\Big)^2 + \mathrm{i}\sum_{\mu,a} y_\mu^a\,(Rh^\mu - g_B(h^\mu)) - \frac{ky_\nu^b}{g_B(h^\nu)}\Bigg) \\ &= \lim_{\beta\to\infty}\int_{-\infty}^{+\infty} \mathrm{D}h\,\mathrm{D}z\, \exp\left(\frac{k^2}{\beta\chi g_B^2(h)} + \frac{\mathrm{i}k}{\chi}\Big(1 - \frac{Rh}{g_B(h)}\Big) - \mathrm{i}\frac{kz\sqrt{q-R^2}}{\chi g_B(h)} \right)\end{aligned}$$

with $\chi = \beta(q_0 - q)$. Assuming binary outputs, $|g_B(h)| = 1$ this finally yields

$$P(x) = \frac{\chi}{\sqrt{2\pi(q-R^2)}} \int_{-\infty}^{+\infty} \mathrm{D}h\, \exp\left(-\frac{\big(1 - \chi x - Rhg_B(h)\big)^2}{2(q-R^2)} \right). \tag{A2}$$

The order parameters R and q_0 can be obtained by the techniques introduced in [3, 10] and read

$$R = \alpha\int_{-\infty}^{+\infty} \mathrm{D}h\, hg_B(h) \qquad q_0 = \frac{\alpha\int_{-\infty}^{+\infty}\mathrm{D}h\, g_B^2(h) - R^2}{1-\alpha} \tag{A3}$$

for $\alpha < 1$. Explicit results for continuous functions $g_B(h)$ are given in [10].

Finally, the parameter χ can be determined using equation (3) together with (A2)

$$q_0 = \frac{1}{N}\boldsymbol{J}^2 = \frac{1}{N}\sum_{\mu,\nu} g_B(h^\nu)(C^{-1})_{\mu\nu} g_B(h^\mu) = \frac{1}{N}\sum_\mu g_B^2(h^\mu)x_\mu$$
$$= \alpha\int_{-\infty}^{\infty} \mathrm{d}x\, x P(x)\,. \tag{A4}$$

In the last equality, we have specialized on binary outputs only. Calculating the integral with the help of (A2) and comparing with (A3) leads to $\chi = 1-\alpha$.

Appendix B. Order parameters

The explicit results for the order parameters (19) are:

$$Q = -\frac{1}{\chi^2}\int_{-\infty}^{+\infty} \mathrm{D}h\,\mathrm{D}z\left(Rh - g_B(h) + z\sqrt{q-R^2}\right)^2$$

$$U = -\frac{1}{\chi}\int_{-\infty}^{+\infty} \mathrm{D}h\, h g_B(h)$$

$$\delta = 1 + \frac{\alpha}{\chi},$$

$$\gamma = -\frac{1}{\chi}\int_{-\infty}^{+\infty} \mathrm{D}h\,\mathrm{D}z\, f'\left(-\frac{Rh - g_B(h) + z\sqrt{q-R^2}}{\chi g_B(h)}\right)$$

$$F = \int_{-\infty}^{+\infty} \mathrm{D}h\,\mathrm{D}z\, g_B^2(h) f^2\left(-\frac{Rh - g_B(h) + z\sqrt{q-R^2}}{\chi g_B(h)}\right)$$

$$G = -\frac{1}{\chi}\int_{-\infty}^{+\infty} \mathrm{D}h\,\mathrm{D}z\, g_B(h)\,(Rh - g_B(h))\, f\left(-\frac{Rh - g_B(h) + z\sqrt{q-R^2}}{\chi g_B(h)}\right) - \frac{q-R^2}{\chi}\,\gamma$$

$$T = \int_{-\infty}^{+\infty} \mathrm{D}h\,\mathrm{D}z\, h g_B(h) f\left(-\frac{Rh - g_B(h) + z\sqrt{q-R^2}}{\chi g_B(h)}\right) - R\,\gamma\,.$$

The parameters Q and U depend only on the learning task and are independent of the choice of the weight function $f(x)$.

(i) Perceptron with threshold

$$Q = -\frac{q + 1 - 2R\sqrt{2/\pi}\exp\left(-\tau^2/2\right)}{\chi^2}$$
$$U = -\sqrt{\frac{2}{\pi}}\frac{1}{\chi}\exp\left(-\frac{\tau^2}{2}\right). \tag{B1}$$

(ii) Reversed-wedge problem

$$Q = -\frac{q + 1 - 2R\sqrt{2/\pi}\left[2\exp\left(-\tau^2/2\right) - 1\right]}{\chi^2}$$

$$U = -\sqrt{\frac{2}{\pi}}\frac{1}{\chi}\left[2\exp\left(-\frac{\tau^2}{2}\right) - 1\right].$$

For modified Hebbian learning $f(x) = \Theta(x)$ we get:

$$F = \int_{-\infty}^{+\infty} \mathrm{D}h\, g_B^2(h)\Phi\left(-\frac{g_B(h)\,(Rh - g_B(h))}{\sqrt{g_B^2(h)(q-R^2)}}\right)$$

$$\gamma = -\frac{1}{\sqrt{2\pi(q-R^2)}}\int_{-\infty}^{+\infty}\frac{\mathrm{d}h}{\sqrt{2\pi}}\sqrt{g_B^2(h)}\exp\left(-\frac{qh^2}{2(q-R^2)} + \frac{Rhg_B(h)}{q-R^2} - \frac{g_B^2(h)}{2(q-R^2)}\right)$$

$$G = -\frac{1}{\chi}\int_{-\infty}^{+\infty}\mathrm{D}h\, g_B(h)\,(Rh - g_B(h))\,\Phi\left(-\frac{g_B(h)\,(Rh-g_B(h))}{\sqrt{g_B^2(h)(q-R^2)}}\right) - \frac{q-R^2}{\chi}\gamma$$

$$T = \int_{-\infty}^{+\infty}\mathrm{D}h\, hg_B(h)\Phi\left(-\frac{g_B(h)\,(Rh-g_B(h))}{\sqrt{g_B^2(h)(q-R^2)}}\right) - R\gamma\,.$$

In a similar way we are able to obtain the parameters for modified Adaline learning, i.e. $f(x) = x\Theta(x)$:

$$\begin{aligned}F = {} & \int_{-\infty}^{+\infty}\mathrm{D}h\,\frac{(Rh-g_B(h))^2 + q - R^2}{\chi^2}\Phi\left(-\frac{g_B(h)(Rh-g_B(h))}{\sqrt{g_B^2(h)(q-R^2)}}\right)\\ & -\frac{\sqrt{q-R^2}}{\sqrt{2\pi}\,\chi^2}\int_{-\infty}^{+\infty}\frac{\mathrm{d}h}{\sqrt{2\pi}}\frac{g_B(h)(Rh-g_B(h))}{\sqrt{g_B^2(h)}}\\ & \times\exp\left(-\frac{qh^2}{2(q-R^2)} + \frac{Rhg_B(h)}{q-R^2} - \frac{g_B^2(h)}{2(q-R^2)}\right)\end{aligned}$$

$$\gamma = -\frac{1}{\chi}\int_{-\infty}^{+\infty}\mathrm{D}h\,\Phi\left(-\frac{g_B(h)(Rh-g_B(h))}{\sqrt{g_B^2(h)(q-R^2)}}\right)$$

$$\begin{aligned}T = {} & \int_{-\infty}^{+\infty}\mathrm{D}h\,\frac{hg_B(h) - R(h^2-1)}{\chi}\Phi\left(-\frac{g_B(h)(Rh-g_B(h))}{\sqrt{g_B^2(h)(q-R^2)}}\right)\\ & +\frac{\sqrt{q-R^2}}{\chi}\int_{-\infty}^{+\infty}\frac{\mathrm{d}h}{2\pi}\frac{hg_B(h)}{\sqrt{g_B^2(h)}}\exp\left(-\frac{qh^2}{2(q-R^2)} + \frac{Rhg_B(h)}{q-R^2} - \frac{g_B^2(h)}{2(q-R^2)}\right).\end{aligned}$$

References

[1] Gardner E 1988 *J. Phys. A: Math. Gen.* **21** 257

[2] Gardner E and Derrida B 1988 *J. Phys. A: Math. Gen.* **21** 271

[3] Opper M, Kinzel W, Kleinz J and Nehl R 1990 *J. Phys. A: Math. Gen.* **23** L581

[4] Hebb D O 1949 *The Organization of Behaviour* (New York: Wiley)

[5] Anlauf J and Biehl M 1989 *Europhys. Lett.* **10** 687

[6] Seung H S, Sompolinsky H and Tishby N 1992 *Phys. Rev.* A **45** 6056

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

[8] Opper M and Kinzel W 1996 Statistical mechanics of generalization *Physics of Neural Networks III* ed J S van Hemmen, E Domany and K Schulten (Berlin: Springer) to be published
[9] Vallet F 1989 *Europhys. Lett.* **8** 747
[10] Boes S, Kinzel W and Opper M 1993 *Phys. Rev.* E **47** 1384
[11] Kohonen T 1988 *Self-Organisation and Associative Memory* (Berlin: Springer)
[12] Diederich S and Opper M 1987 *Phys. Rev. Lett.* **59**, 949
[13] Widrow B and Hoff M E 1960 *WESCON Convention Report* IV 96
[14] Kinzel W and Ruján P 1990 *Europhys. Lett.* **13** 473
[15] Watkin T L H and Rau A 1992 *J. Phys. A: Math. Gen.* **25** 113
[16] Seung H S, Opper M and Sompolinsky H 1992 *Vth Ann. Workshop on Computational Learning Theory (COLT92) (Pittsburgh)* pp 287–94 (New York: Ass. for Computing Machinery)
[17] Opper M 1988 *Phys. Rev.* A **38** 3824
[18] Garces R 1993 Scheme to improve the generalization error *Proc. 1993 Connectionist Models Summer School* ed M Mozer, P Smolensky, D Touretzky, J Elman and A Weigend pp 358–63
[19] Watkin T L H and Rau A 1992 *Phys. Rev.* A **45** 4102
[20] Garces R, Kuhlmann P and Eissfeller H 1992 *J. Phys. A: Math. Gen.* **25** L1335
[21] Kinzel W and Opper M 1991 Dynamics of learning *Physics of Neural Networks* ed J L van Hemmen, E Domany and K Schulten (Berlin: Springer) p 149
[22] Krogh A and Hertz J 1991 *Advances in Neural Information Processing Systems III* (San Mateo, CA: Morgan Kaufmann)
[23] LeCun Y, Denker J S and Solla S 1990 Optimal brain damage *Advances in Neural Information Processing Systems II (Denver, 1989)* ed D Touretzky (San Mateo, CA: Morgan Kaufmann) p 598
[24] Bouten M, Engel A, Komoda A and Serneels R 1990 *J. Phys. A: Math. Gen.* **23** 4643
[25] Kuhlmann P and Müller K R 1994 *J. Phys. A: Math. Gen.* **27** 3759