IOPscience iopscience.iop.org

Home Search Collections Journals About Contact us My IOPscience

Dilution in Boolean perceptrons that learn from noisy examples

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

1996 J. Phys. A: Math. Gen. 29 7003

(http://iopscience.iop.org/0305-4470/29/22/006)

View the table of contents for this issue, or go to the journal homepage for more

Download details:
IP Address: 171.66.16.70
The article was downloaded on 02/06/2010 at 04:04

Please note that terms and conditions apply.

J. Phys. A: Math. Gen. **29** (1996) 7003–7012. Printed in the UK

Dilution in Boolean perceptrons that learn from noisy examples

D M L Barbato and J F Fontanari

Instituto de Física de São Carlos, Universidade de São Paulo, Caixa Postal 369, 13560-970 São Carlos SP, Brazil

Received 29 July 1996

Abstract. We investigate the effect of dilution after learning on the generalization ability of single-layer Boolean perceptrons that learn from noisy examples. We present a thorough comparison between the relative performances of several well known learning rules. In particular, we show that the effect of dilution is always deleterious, and that the Bayes algorithm always gives the best generalization performance.

0. Introduction

Most neural networks researchers today agree that the generalization error, rather than the training error, is a more appropriate measure for the performance of feed-forward neural networks that learn an input/output mapping from a limited number of examples. Although it had long been known that the Bayes rule gives the optimal generalization performance for this type of task (Duda and Hart 1973), it was the statistical mechanics approach initiated by Gardner (1988) that made possible the analytical calculation of the optimal generalization error for the linearly separable binary classification problem (Opper and Haussler 1991a, b). More recently, Watkin (1993) has shown that the Bayes algorithm can be implemented by a single-layer perceptron, while Kinouchi and Caticha (1996) have actually presented an algorithm to determine the weights of this Bayesian perceptron. An issue that has remained unaddressed in this search for the optimal performance is the robustness of the resulting network against the deletion of a fraction of its weights. In practical applications, this may become a major selection criterion for learning algorithms.

In this paper we investigate analytically the effect of the elimination of weights (dilution) on the generalization performance of single-layer Boolean perceptrons that learn from noisy examples. More specifically, *after* the learning process has finished, we set to zero the $(1-\delta)N$ smaller weights, where N is the total number of weights and $0 \leqslant \delta \leqslant 1$ is a parameter that controls the degree of dilution. We think that the deletion of the smaller weights, rather than random deletion, is a somewhat more realistic model of damage in neural networks, since the smaller weights are more likely to be affected by damage than the larger ones. Our main conclusions, however, do not depend on the sizes of the deleted weights or on the randomness of the dilution process.

Since we are interested in probing the robustness of the learning rules against the cutting of weights, we have focused only on the problem of dilution after learning. The problem of dilution during the learning process in single-layer Boolean perceptrons has been addressed by Bouten *et al* (1990) for the random mapping problem and by Kuhlmann and Müller

0305-4470/96/227003+10$19.50 © 1996 IOP Publishing Ltd

(1994) for the problem of learning from examples. The analysis of this type of dilution aims at determining the best training performance that a diluted network can achieve: the particular weights to be deleted are determined by the learning process in order to minimize the training energy. Since in this case the dilution process is correlated with the learning process, it is clearly not appropriate to model damage in neural networks and so it has little relevance to our investigation of the robustness of learning rules against the deletion of weights.

In the following, we evaluate and compare the performances of diluted perceptrons that learn from noisy examples using the following learning rules or algorithms: the pseudo-inverse rule, the Hebb rule, the Gibbs algorithm, the optimal stability algorithm and the Bayes algorithm. We mention that, to the best of our knowledge, we are not aware of any extensive comparison between the performances of these well known learning rules for the problem of learning from noisy examples.

The neural network we consider consists of N binary input units $s_i = \pm 1$, N synaptic weights w_i and a single Boolean output unit

$$\sigma = \operatorname{sign}\left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i s_i\right). \tag{1}$$

The task of this perceptron (student) is to realize the mapping between the 2^N possible input configurations $\{\boldsymbol{s}\}$ and their respective outputs $\{t\}$ generated by another perceptron (teacher),

$$t = \operatorname{sign}\left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i^{\mathbf{0}} s_i\right) \tag{2}$$

where the weights $w_i^{\mathbf{0}}$ are statistically independent, Gaussian distributed random variables of zero mean and unit variance. To achieve this task, the student perceptron is trained with $P = \alpha N$ noisy input/output pairs $\{\boldsymbol{s}^l, \zeta^l\}$ $(l = 1, \ldots, P)$ where ζ^l is drawn from the conditional probability distribution

$$P(\zeta^l | t^l) = (1-\chi)\delta(\zeta^l - t^l) + \chi\delta(\zeta^l + t^l) \tag{3}$$

and t^l is the teacher's output to input $\boldsymbol{s}^l$. The components of the input vectors are chosen randomly as ± 1 with equal probabilities. The parameter $0 \leqslant \chi \leqslant \frac{1}{2}$ measures the noise intensity: the larger χ, the larger the noise.

In order to characterize the performance of the diluted student perceptron in classifying correctly a randomly chosen input vector for a *typical* realization of the training set, we introduce the average generalization error, which in the Boolean case is given by (Györgyi and Tishby 1989, Seung *et al* 1992)

$$\epsilon_g = \frac{1}{\pi} \arccos\left(\frac{R_\delta}{\sqrt{Q_\delta}}\right) \tag{4}$$

where

$$Q_\delta = \left\langle \frac{1}{N} \sum_i w_i^2 \Theta(|w_i| - a) \right\rangle \tag{5}$$

and

$$R_\delta = \left\langle \frac{1}{N} \sum_i w_i^{\mathbf{0}} w_i \Theta(|w_i| - a) \right\rangle. \tag{6}$$

Here $\Theta(x) = 1$ if $x \geqslant 0$ and 0, otherwise. The relation between the cut-off a and the degree of dilution δ is simply

$$\delta = \left\langle \frac{1}{N} \sum_i \Theta(|w_i| - a) \right\rangle. \tag{7}$$

The noise-free generalization error measures the probability of the diluted student perceptron failing in outputting $t = \mathrm{sign}(\sum_j w_j^0 s_j)$ for the randomly chosen input vector $\boldsymbol{s}$. The notation $\langle \cdots \rangle$ stands for the averages over ζ^l, s_i^l and $w_i^{\mathbf{0}}$. We note that in the case of the Gibbs algorithm the set of weights generated by the algorithm is not unique, so this notation also stands for an average over the ensemble of weights that minimize the corresponding free energy. For a thorough discussion of the problem of learning from examples in neural networks we refer the reader to Watkin *et al* (1993).

The equilibrium properties of the perceptrons generated by the learning rules we consider in this paper can be obtained by minimizing a training energy function $E(\boldsymbol{w})$, which depends on the weights $\boldsymbol{w}$ only through the stabilities

$$\Delta^l = \frac{\zeta^l}{\sqrt{N}} \sum_i w_i s_i^l. \tag{8}$$

The specific form of the function $E\left(\{\Delta^l\}\right)$ depends of course on the learning rule considered. We have found that the effect of dilution for all such rules is a simple rescaling of the non-diluted order parameters Q_1 and R_1. More specifically, $Q_\delta = \Lambda_\delta Q_1$ and $R_\delta = \Lambda_\delta R_1$ where

$$\Lambda_\delta = \delta + \sqrt{\frac{2}{\pi}} \lambda \mathrm{e}^{-\lambda^2/2} \tag{9}$$

and λ is the unique solution of

$$\delta = 2H(\lambda). \tag{10}$$

Here $H(x) = \int_x^\infty \mathrm{D}t$ and $\mathrm{D}t = \mathrm{d}t/\sqrt{2\pi}\mathrm{e}^{-t^2/2}$ is the Gaussian measure. We note that $\Lambda_\delta \in [0, 1]$ is a monotonically increasing function of $\delta \in [0, 1]$. Hence the average generalization error (4) becomes

$$\epsilon_g = \frac{1}{\pi} \arccos\left(\sqrt{\Lambda_\delta} \frac{R_1}{\sqrt{Q_1}} \right). \tag{11}$$

In the appendix we present the proof of the general result that the effect of any *deterministic* procedure for deleting weights after learning is a rescaling of the non-diluted order parameters Q_1 and R_1. Furthermore, as pointed out there, a similar rescaling also holds for the random deletion of weights. Thus, the effect of dilution after learning in single-layer Boolean perceptrons is always deleterious, even in the case of noisy examples. This contrasts with the findings for the linear perceptron, for which the deletion of weights can actually improve the generalization performance in the case of learning from noisy examples (Barbato and Fontanari 1995). The reason for this discrepancy is as follows. As pointed out in the appendix, the rescaling of the non-diluted order parameters holds for the linear perceptron as well, so the average generalization error for the diluted linear perceptron becomes $\epsilon_g = 1 - \Lambda_\delta(2R_1 - Q_1)$ (Seung *et al* 1992, Barbato and Fontanari 1995). For noiseless examples, $\chi = 0$, one has $R_1 \geqslant Q_1/2$ for all α, so that the generalization error decreases with increasing δ. Hence the effect of dilution is deleterious. In the case of noisy examples, for any fixed $\chi > 0$ there is a range of α for which $R_1 < Q_1/2$ so that the generalization error decreases with decreasing δ; the effect of dilution is beneficial in this case.

We now proceed with the evaluation of Q_1 and R_1 for the several learning rules mentioned above. As these calculations are standard (Gardner and Derrida 1988, Seung *et al* 1992) we will present the final results only.

1. The pseudo-inverse rule

In the case of the pseudo-inverse rule, the weights are obtained by minimizing the training energy (Opper *et al* 1990)

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_l \left(1 - \frac{\zeta^l}{\sqrt{N}}\sum_i w_i s_i^l\right)^2. \tag{12}$$

For $\alpha \leqslant 1$, the minimum is not unique, so we must choose the one with the minimal norm. It yields

$$Q_1 = \frac{\alpha}{\pi}\frac{\pi - 2\alpha(1-2\chi)^2}{1-\alpha} \tag{13}$$

and

$$R_1 = \sqrt{\frac{2}{\pi}}\alpha(1-2\chi). \tag{14}$$

For $\alpha > 1$, we find

$$Q_1 = \frac{2(\alpha-2)(1-2\chi)^2 + \pi}{\pi(\alpha-1)} \tag{15}$$

and

$$R_1 = \sqrt{\frac{2}{\pi}}(1-2\chi). \tag{16}$$

These results agree with those obtained by Opper *et al* (1990) for the noiseless case. For large α we find

$$\epsilon_g^P \approx \frac{1}{\sqrt{2\pi\alpha(1-2\chi)^2}}\left[1 - \frac{2}{\pi}(1-2\chi)^2\right]^{1/2} \tag{17}$$

for $\delta = 1$, and

$$\epsilon_g^P \approx \frac{1}{\pi}\arccos\sqrt{\Lambda_\delta} + \left(\frac{\Lambda_\delta}{1-\Lambda_\delta}\right)^{1/2}\frac{1}{4\alpha(1-2\chi)^2}\left[1 - \frac{2}{\pi}(1-2\chi)^2\right] \tag{18}$$

for $\delta < 1$.

2. The Hebb rule

The problem of learning a linearly separable mapping using the Hebb rule was investigated by Vallet (1989). Here we generalize that analysis by including noise. Although the Hebbian prescription for writing the weights w_i in terms of the training set $\{\boldsymbol{s}^l, \zeta^l\}$ is exceedingly simple, and the analysis of the effects of dilution becomes much easier if we note that the Hebbian weight vector minimizes the training energy (Griniasty and Gutfreund 1991)

$$E(\boldsymbol{w}) = -\sum_l \frac{\zeta^l}{\sqrt{N}}\sum_i w_i s_i^l \tag{19}$$

with $Q_1 = 1$ fixed. The result for R_1 is simply

$$R_1 = \sqrt{\frac{2\alpha(1-2\chi)^2}{\pi + 2\alpha(1-2\chi)^2}} \tag{20}$$

which coincides with the result obtained by Vallet (1989) for $\chi = 0$. In the limit of large α we find

$$\epsilon_g^H \approx \frac{1}{\sqrt{2\pi\alpha(1-2\chi)^2}} \tag{21}$$

for $\delta = 1$, and

$$\epsilon_g^H \approx \frac{1}{\pi} \arccos\sqrt{\Lambda_\delta} + \left(\frac{\Lambda_\delta}{1-\Lambda_\delta}\right)^{1/2} \frac{1}{4\alpha(1-2\chi)^2} \tag{22}$$

for $\delta < 1$. Thus, in this limit we have $\epsilon_g^H > \epsilon_g^P$ for all χ.

3. The Gibbs algorithm

The Gibbs algorithm chooses a weight vector $\boldsymbol{w}$ at random according to the Gibbs probability distribution

$$Pr(\boldsymbol{w}) = \frac{1}{Z} \exp[-\beta E(\boldsymbol{w})] \tag{23}$$

where

$$E(\boldsymbol{w}) = \sum_l \Theta\left(\kappa - \frac{\zeta^l}{\sqrt{N}} \sum_i w_i s_i^l\right) \tag{24}$$

is the training energy, β is the inverse temperature, and the normalization factor Z is the partition function. Here $\kappa \geqslant 0$ is the margin parameter. The normalization of the weights is not relevant (it gives the scale of κ), so we take $Q_1 = 1$, as usual. Within the replica-symmetric framework, the ensemble of weights generated by (23) is characterized by the order parameters R_1 and q that extremize the free energy density

$$\begin{aligned} -\beta f = {} & \frac{1}{2}\left[\frac{q - R_1^2}{1-q} + \ln(1-q)\right] \\ & + \alpha \int \mathrm{D}t\, [\chi + (1-2\chi)H(\xi_1)] \ln[\mathrm{e}^{-\beta} + (1-\mathrm{e}^{-\beta})H(\xi_2)] \end{aligned} \tag{25}$$

where

$$\xi_1 = \frac{R_1 t}{\sqrt{q - R_1^2}} \tag{26}$$

and

$$\xi_2 = \frac{\kappa + \sqrt{q}t}{\sqrt{1-q}}. \tag{27}$$

We restrict our analysis to the limit $\beta \to \infty$ with $\kappa \leqslant \kappa_c(\alpha, \chi)$, so that the algorithm is consistent, i.e. the training set is learned perfectly (zero average training error). Using standard techniques (de Almeida and Thouless 1978), we have verified that only in this regime is the replica-symmetric saddle point locally stable. In figure 1 we present κ_c as a function of α for several values of χ. Note that the curve for $\chi = 0$ never intersects the α-axis. Although a careful choice of the margin parameter can considerably improve the generalization performance of the Gibbs algorithm (Meir and Fontanari 1992), we will only consider the standard choice $\kappa = 0$ in this paper.

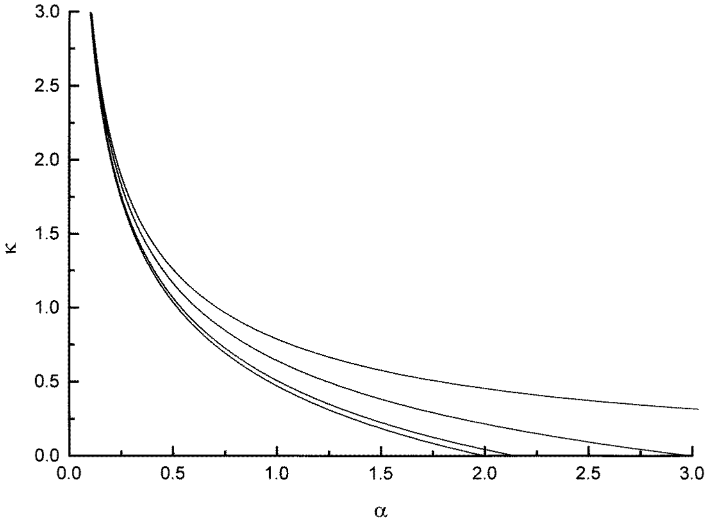

Figure 1. The largest value of the margin parameter κ for which the zero-temperature Gibbs algorithm is consistent as a function of the training set size α for, from top to bottom, $\chi = 0$, 0.1, 0.3 and 0.5.

4. The optimal stability algorithm

The equilibrium properties of the unique weight vector generated by the optimal stability algorithm (Krauth and Mezard 1987) is obtained by extremizing the zero temperature free energy (25) at $\kappa = \kappa_c(\alpha, \chi)$, i.e. at the largest value of the margin parameter for which the zero temperature Gibbs algorithm is consistent (Opper *et al* 1990).

5. The Bayes optimal classification algorithm

Given a randomly chosen input vector s, we can divide the ensemble of weight vectors generated according to the Gibbs distribution (23) into two disjoint sets, such that the weight vectors belonging to the first set will classify s as $+1$, while the ones belonging to the second will classify s as -1. The weighted majority algorithm tells us to classify that input vector according to the classification of the largest set. The Bayes optimal classification algorithm is identical to this majority algorithm, except that the temperature of the Gibbs distribution depends on the noise parameter (Opper and Haussler 1991b)

$$\beta = \ln \frac{1-\chi}{\chi}. \tag{28}$$

We note that for this particular temperature the extrema of the free energy (25) are $q = R_1$, where R_1 is the solution of the equation

$$\frac{R_1}{\sqrt{1-R_1}} = \frac{\alpha}{\pi}(1-2\chi)^2 \int \mathrm{D}t \, \frac{\mathrm{e}^{-R_1 t^2/2}}{\chi + (1-2\chi) H\left(\sqrt{R_1} t\right)}. \tag{29}$$

In view of the results of Watkin (1993) and Kinouchi and Caticha (1996) mentioned before, we can write the noise-free generalization error for the diluted Bayes perceptron as (Opper and Haussler 1991a, b)

$$\epsilon_g^B = \frac{1}{\pi} \arccos\left(\sqrt{\Lambda_\delta R_1}\right) \tag{30}$$

with R_1 given by the solution of equation (29). In the limit of large α the optimal generalization error is given by

$$\epsilon_g^B \approx \frac{1}{\pi\alpha\,\Xi(1-2\chi)^2} \tag{31}$$

for $\delta = 1$, and

$$\epsilon_g^B \approx \frac{1}{\pi}\arccos\sqrt{\Lambda_\delta} + \left(\frac{\Lambda_\delta}{1-\Lambda_\delta}\right)^{1/2} \frac{1}{2\pi\alpha^2\,\Xi^2(1-2\chi)^4} \tag{32}$$

for $\delta < 1$. Here

$$\Xi = \frac{1}{\pi}\int \mathrm{D}t\,\frac{\mathrm{e}^{-t^2/2}}{\chi + (1-2\chi)H(t)}. \tag{33}$$

We now turn to the analysis of our results. For $\delta = 1$ and $\chi = 0$, we present in figure 2 the generalization error as a function of α for the five learning algorithms discussed above. This is essentially the figure presented by Opper *et al* (1990), except for the inclusion of the Gibbs and Bayes learning curves. In figure 3 we present the generalization error for non-diluted networks ($\delta = 1$) which learn from noisy examples ($\chi = 0.1$). Note the surprisingly good performance of the pseudo-inverse and Hebb rules. The learning curves for the optimal stability and the Gibbs algorithm are presented only for $\alpha \leqslant 2.992$, since beyond this value the replica-symmetric solution is locally unstable. Although the dilution does not alter the rank of the learning rules, it does decrease the difference between their relative performances. In fact, for small δ the generalization performances of all algorithms tend to that of the random guessing ($\epsilon_g \approx 0.5$). We note that, at least for single-layer Boolean perceptrons, the strategy of pruning the smaller weights to avoid overfitting in the case of learning from noisy examples actually worsens the generalization performance of the network.

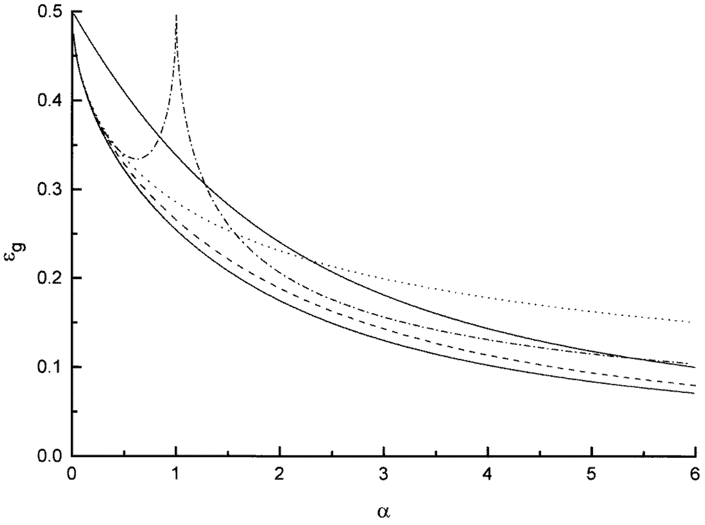

Figure 2. Average generalization error ϵ_g as a function of the training set size α for the pseudo-inverse rule (chain curve), the Hebb rule (dotted curve), the Gibbs algorithm (upper full curve), the optimal stability algorithm (broken curve) and the Bayes algorithm (lower full curve). The parameters are $\delta = 1$ and $\chi = 0$.

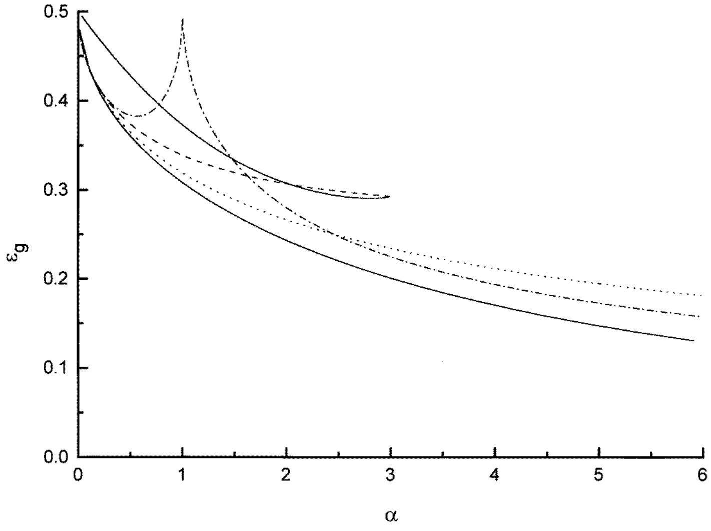

Figure 3. Same as figure 2 but for $\delta = 1$ and $\chi = 0.1$.

It is clear from the proof presented in the appendix that any *deterministic* procedure for deleting weights of a single-layer Boolean perceptron *after* the learning procedure has finished will not alter the performance rank of the learning algorithms. Thus the Bayes algorithm will always give the optimal generalization performance. This conclusion also seems to hold for the case of *random* dilution. The validity of this general result relies on the assumptions that the training energy depends on the weights only through the stabilities of the patterns (8) and that the replica-symmetric saddle point is stable. The former is clearly satisfied for all the learning rules considered in this paper, while the fulfilment of the latter was verified by a standard stability calculation (de Almeida and Thouless 1978).

We have also investigated the effect of noise acting on the input patterns instead of on the output bits. More specifically, we have considered the case in which each component s_i is flipped with a probability $(1-\gamma)/2$ where $0 \leqslant \gamma \leqslant 1$ (Györgyi and Tishby 1989). For the pseudo-inverse and Hebb rules we have found a simple relation between the two noise parameters, namely, $\gamma = 1 - 2\chi$. For the Gibbs learning rule, however, there is no such relation, although the behaviour pattern of the generalization error is qualitatively the same for both types of noise.

In summary, we have addressed the important problem of the robustness of the learning rules against cutting of weights after the learning process has finished. We have found that, under quite general conditions, the effect of dilution is a rescaling of the order parameters that measure the norm of the weights of the non-diluted student perceptron and its overlap with the teacher perceptron. As a result, the dilution will not alter the performance rank of the learning rules. Furthermore, we have presented a thorough comparison of the generalization performances of several well known learning rules for the problem of learning from noisy examples. In doing so, we have generalized previous analyses of the Hebb rule (Vallet 1989), the pseudo-inverse rule and the optimal stability algorithm (Opper *et al* 1990) by including the effects of noise in the training examples. Moreover, we have compared the performance of these learning rules with that of the Bayes optimal algorithm, which provides a natural standard for this kind of analysis.

Acknowledgment

This work was supported in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Appendix

In this appendix we present the proof that the effect of any *deterministic* procedure for deleting weights in a single-layer perceptron is a simple rescaling of the non-diluted physical order parameters Q_1 and R_1. In the following we focus on the calculation of Q_δ, which is given by

$$Q_\delta = -\frac{1}{\beta N}\frac{\partial}{\partial h}\langle \ln Z\rangle|_{h=0} \tag{34}$$

where $Z = \sum_{\boldsymbol{w}} \exp(-\beta \hat{E})$ and we have introduced the auxiliary energy

$$\hat{E}(\boldsymbol{w}) = E(\boldsymbol{w}) + h\sum_i w_i^2 \mathcal{F}^2(w_i, \delta). \tag{35}$$

The dependence on the learning rule appears through the training energy $E(\boldsymbol{w}) = E(\{\Delta^l\})$. The deterministic procedure for cutting $(1-\delta)N$ weights is modelled by the function $\mathcal{F}(w_i, \delta)$ that must satisfy $\mathcal{F}(w_i, 0) = 0$ and $\mathcal{F}(w_i, 1) = 1$. In the case of the deletion of the smaller weights we have $\mathcal{F}(w_i, \delta) = \Theta(|w_i| - a)$ with a given by equation (7). As usual, we evaluate the averages over the statistically independent random variables s_i^l, ζ^l and $w_i^{\mathbf{0}}$ through the replica method which consists of using the identity

$$\langle \ln Z\rangle = \lim_{n\to 0}\frac{1}{n}\ln\langle Z^n\rangle \tag{36}$$

evaluating $\langle Z^n\rangle$ for integer n and then analytically continuing to $n = 0$. The calculation of $\langle \ln Z\rangle$ within the replica-symmetric framework in the limit $N \to \infty$ is standard (Gardner and Derrida 1988, Seung *et al* 1992) and yields

$$\frac{1}{N}\langle \ln Z\rangle = R_1\hat{R}_1 - \tfrac{1}{2}q_1\hat{q}_1 + Q_1\hat{Q}_1 + \alpha[(1-\chi)G_1^+ + \chi G_1^-] + G_2 \tag{37}$$

where

$$G_1^\pm = \int \mathrm{D}z \int \frac{\mathrm{d}y\,\mathrm{d}\hat{y}}{2\pi}\mathrm{e}^{-\hat{y}^2/2 + \mathrm{i}y\hat{y}} \ln \int \frac{\mathrm{d}x\,\mathrm{d}\hat{x}}{2\pi}\mathrm{e}^{\mathrm{i}x\hat{x}}$$
$$\times \exp[-\tfrac{1}{2}(Q_1 - q_1)\hat{x}^2 + \mathrm{i}\sqrt{q_1}z\hat{x} - R_1\hat{x}\hat{y} - \beta E(\pm x\,\mathrm{sign}\,y)] \tag{38}$$

and

$$G_2 = \int \mathrm{D}z \int \mathrm{D}w^0 \ln \int \mathrm{d}w\ \exp[\mathcal{H}(w, w^0, z) - \beta h w^2 \mathcal{F}^2(w, \delta)]. \tag{39}$$

Here,

$$\mathcal{H}(w, w^0, z) = -(\hat{Q}_1 - \hat{q}_1/2)w^2 + \mathrm{i}\sqrt{\hat{q}_1}wz - \hat{R}_1 w w^0 \tag{40}$$

is an effective Hamiltonian whose form is independent of $E(\boldsymbol{w})$ and so of the learning rule. The saddle-point parameters $(\hat{Q}_1, \hat{R}_1, \hat{q}_1, Q_1, R_1, q_1)$ are determined by extremizing (37) for $h = 0$. Hence equation (34) yields

$$Q_\delta = \int \mathrm{D}z \int \mathrm{D}w^0 \frac{\int \mathrm{d}w\, w^2\mathcal{F}^2(w, \delta)\exp[\mathcal{H}(w, w^0, z)]}{\int \mathrm{d}w\ \exp[\mathcal{H}(w, w^0, z)]}. \tag{41}$$

Finally, by first evaluating the integral over w in the denominator, the remaining Gaussian integrals over z and w^0 can easily be performed, yielding

$$Q_\delta = Q_1 \int \mathrm{D}w\, w^2 \mathcal{F}^2 \left(\sqrt{Q_1} w, \delta \right) . \tag{42}$$

To calculate R_δ we add the auxiliary term $h \sum_i w_i^{\mathbf{0}} w_i \mathcal{F}(w_i, \delta)$ to the training energy $E(\boldsymbol{w})$ and follow the procedure given above. The final result is

$$R_\delta = R_1 \int \mathrm{D}w\, w^2 \mathcal{F} \left(\sqrt{Q_1} w, \delta \right) . \tag{43}$$

In particular, for the deletion of the $(1-\delta)N$ smaller weights we obtain $Q_\delta = \Lambda_\delta Q_1$ and $R_\delta = \Lambda_\delta R_1$ with Λ_δ given in (9). It is interesting to note that the above results are valid for the linear perceptron as well, since the choice of the transfer function of the input neurons affects only the term $G_1^\pm$ in the equation (37).

We have also considered a stochastic dilution procedure in which each weight w_i is set to zero with a probability $1-\delta$, so that there will be on average δN non-vanishing weights. This can be achieved by calculating the order parameters

$$Q_\delta = \left\langle \overline{ \frac{1}{N} \sum_i w_i^2 c_i^2 } \right\rangle \tag{44}$$

and

$$R_\delta = \left\langle \overline{ \frac{1}{N} \sum_i w_i w_i^{\mathbf{0}} c_i } \right\rangle \tag{45}$$

where the bar indicates an average over the statistically independent random variables c_i which can assume the values 0 and 1 with a probability $1-\delta$ and δ, respectively. In this case the rescaling of the non-diluted order parameters is trivial: $Q_\delta = \delta Q_1$ and $R_\delta = \delta R_1$.

References

de Almeida J R and Thouless D J 1978 *J. Phys. A: Math. Gen.* **11** 983
Barbato D M L and Fontanari J F 1995 *Phys. Rev.* E **51** 6219
Bouten M, Engel A, Komoda A and Serneels R 1990 *J. Phys. A: Math. Gen.* **23** 4643
Duda R O and Hart P E 1973 *Pattern Classification and Scene Analysis* (New York: Wiley)
Gardner E 1988 *J. Phys. A: Math. Gen.* **21** 257.
Gardner E and Derrida B 1988 *J. Phys. A: Math. Gen.* **21** 271
Griniasty M and Gutfreund H 1991 *J. Phys. A: Math. Gen.* **24** 715
Györgyi G and Tishby N 1989 *Neural Networks and Spin Glasses* ed W K Theumann and R Köberle (Singapore: World Scientific)
Kinouchi O and Caticha N 1996 *Phys. Rev.* E **54** R54
Krauth W and Mezard M 1987 *J. Phys. A: Math. Gen.* **20** L745
Kuhlmann P and Müller K R 1994 *J. Phys. A: Math. Gen.* **27** 3759
Meir R and Fontanari J F 1992 *Phys. Rev.* A **45** 8874
Opper M, Kinzel W, Kleinz J and Nehl R 1990 *J. Phys. A: Math. Gen.* **23** L581
Opper M and Haussler D 1991a *Phys. Rev. Lett.* **66** 2677
——1991b *Proc. IVth Annual Workshop on Computational Learning Theory* (San Mateo, CA: Morgan Kaufman)
Seung S, Sompolinsky H and Tishby N 1992 *Phys. Rev.* A **45** 6056
Vallet F 1989 *Europhys. Lett.* **8** 747
Watkin T L H 1993 *Europhys. Lett.* **21** 871
Watkin T L H, Rau A and Biehl M 1993 *Rev. Mod. Phys.* **65** 499