ON-LINE LEARNING IN THE COMMITTEE MACHINE

Mauro Copelli and Nestor Caticha
Instituto de Física, Universidade de São Paulo,
CP 20516, 01498 São Paulo, SP, Brazil
e-mail : copelli@if.usp.br and nestor@if.usp.br

October 11, 1994

Submitted to Journal of Physics A

Abstract

The dynamics of learning from examples in the $K = 3$ nonoverlapping committee machine with single presentation of examples is studied. The optimal algorithm, in the sense of mean generalization, is obtained from a variational analysis of the differential equations which describe the dynamics. The agreement of the theoretical predictions and the results of numerical simulations is excellent. The optimized dynamics has the extra advantage with respect to the nonoptimized cases in that it uncouples the differential equations which describe the evolution of the relevant parameters, *i.e.* the student-teacher overlap and the norm of the student synaptic vector. This in turn translates into the possibility of constructing useful practical optimized on-line algorithms which work optimally even in the absence of knowledge of the probability distribution of examples. For the optimal algorithm the generalization error decays as $\sim 0.88\,\alpha^{-1}$, the same nominal error as for the simple perceptron with optimized dynamics.

PACS : 87.10.e+10, 75.10 Hk, 64.60 Cn, 89.70.+c

1 Introduction

The application of the methods of Statistical Mechanics to the study of learning and generalization in perceptrons with no hidden units has yielded a wealth of results which lead in a natural way to the study of more realistic and complex nets. The feedforward nets with a layer of hidden units connected through fixed weights to the output layer have been the natural candidates to continue the study of Neural Networks.

The supervised learning process can typically be casted as a problem of minimizing some energy function constructed from a set of examples. This minimization leads naturally to a learning dynamics. If this minimization is performed in the presence of noise then the problem is that of Statistical Mechanics at finite temperature. Thus the learning algorithms describe a sequence of steps which tend to minimize a free energy. In the iterated form, these algorithms extract all the information contained in the *a priori* given energy function. From a practical point of view, this might be a computationally expensive operation. For the perceptron with no hidden units it has been shown that if the dynamics is properly chosen, then the first step of the minimization leads to an asymptotic decay of the generalization errors that is almost as good as the exhaustive iterated learning. For example, the generalization error in the case of the boolean perceptron with real weights decays as $0.88\alpha^{-1}$, compared to the bayesian bound of Opper and Haussler [11] of $0.44\alpha^{-1}$. In the case of learning by queries, by judiciously choosing the energy function, the error can be made to decay exponentially fast [6].

The main point of the above discussion is that by correctly choosing the dynamics, the computational effort of learning can be drastically reduced. The on-line learning by single presentation of examples, which has been lately studied by several authors [9]-[7]-[4]-[1], can thus lead to very efficient learning algorithms. In the case of nets which are not very useful in real life applications this might not be very important. This is not the case for nets with hidden units, where any reduction in the training times or improvement of performance will certainly be welcomed, and which are applicable in a large number of real problems.

The object of this paper is to study the single presentation of examples algorithms that lead to the best possible generalization (in the mean) in the committee machine with nonoverlapping architecture. The method generalizes to nets with hidden units that of reference [7].

In section 2 the model is presented, an expression for the generalization error as a function of the overlaps is obtained and the equations governing the evolution of the order parameters for a general algorithm are deduced. In section 3 we obtain the optimal weight function through a variational argument. This leads to an algorithm which belongs to the class of the "expected stability " algorithms [7]. In section 4 the theoretical results, obtained by numerical integration of the evolution equations are compared to the results of numerical simulations. Section 5 contains some concluding remarks.

2 The Generalization Error and the Learning Dynamics

Since we are interested in the problem of generalization or rule extraction within the framework of supervised learning, we build a learning set with the help of a teacher network [2]-[3], which for simplicity we take to have the same architecture as the student net.

The K nonoverlapping committee we deal with is a set of K independent boolean perceptrons or branches with N/K inputs units each. The notation we use is such that every N-dimensional vector $\vec{V}$ can be thought of as K branch-vectors $\vec{V} = (\mathbf{V}_1, \ldots, \mathbf{V}_K)$. We will restrict ourselves to the $K = 3$ case.

The learning set is a set of P pairs $\{(\vec{S}, \sigma_B)\}$ where $\vec{S} = (\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3)$ with $S_{kj} = \pm 1$, $k = 1, 2, 3$, $j = 1, \ldots, N/3$ and σ_B is the teacher output (see equation (1) below). The synaptic weights of the teacher are denoted by $\vec{B} = (\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3)$ and the normalization $\sum_{j=1}^{N/3} B_{kj}^2 \equiv B_k^2 = 1$ can be imposed without loss of generality.

Upon presentation of an input vector each branch perceptron gives a partial output

$$\sigma_{Bk} = \mathrm{sign}(b_k) \;, k = 1, 2, 3$$

where $b_k = \mathbf{B}_k.\mathbf{S}_k$. The set $\{\sigma_{Bk}\}$ is the unaccessible internal representation of $\vec{S}$ in the teacher net. The teacher's final output is made up from the internal representation

$$\sigma_B = \mathrm{sign}(\mathcal{B}) \;, \tag{1}$$

where $\mathcal{B} \equiv \sum_{k=1}^{3} \sigma_{Bk}$. The student net is defined by a vector of real connections $\vec{J}$ and an output

$$\sigma_J = \mathrm{sign}(\mathcal{H}) \;, \tag{2}$$

where $\mathcal{H} = \sum_{k=1}^{3} \sigma_{Jk}$, $\sigma_{Jk} = \mathrm{sign}(h_k)$ and $h_k = \tilde{h}_k / J_k \equiv \mathbf{J}_k.\mathbf{S}_k / |\mathbf{J}_k|$. The aim of training the net is to obtain $\vec{J}$, using information from the learning set, such that σ_J is near in some sense to σ_B. The meaning of "near" here is in the sense of generalization, that is, we should find $\vec{J}$ which maximizes the probability of $\sigma_B = \sigma_J$ upon the presentation of a random input vector, uncorrelated with the learning set. That probability is called the generalization ability (g) and its complement is called the generalization error (e_g).

We deduce now the generalization error as a function of the overlaps $\rho_k = \mathbf{J}_k.\mathbf{B}_k / J_k B_k$. For given $\vec{B}$ and $\vec{J}$, the generalization error is the average, over random examples, of

$$e_g^\mu = \frac{1 - \mathrm{sign}(\mathcal{H}^\mu).\mathrm{sign}(\mathcal{B}^\mu)}{2} \;,$$

which is the error of classification of a single example (labeled by the μ index) $(\vec{S}_k^\mu, \sigma_B^\mu)$. The average over the examples can be written as

$$e_g = \int_{-\infty}^{+\infty} d\mathcal{H}^\mu \int_{-\infty}^{+\infty} d\mathcal{B}^\mu P(\mathcal{H}^\mu, \mathcal{B}^\mu) e_g^\mu = \int\int_{\mathcal{H}^\mu \mathcal{B}^\mu < 0} P(\mathcal{H}^\mu, \mathcal{B}^\mu) d\mathcal{H}^\mu d\mathcal{B}^\mu \;,$$

where, supressing the μ index for simplicity,

$$P(\mathcal{H},\mathcal{B}) = \int_{-\infty}^{+\infty} \frac{dx}{2\pi} e^{ix\mathcal{H}} \int_{-\infty}^{+\infty} \frac{dy}{2\pi} e^{iy\mathcal{B}} \prod_{k=1}^{K} \left(e_k \cos\left(x-y\right) + g_k \cos\left(x+y\right)\right) \tag{3}$$

and

$$g_k = 1 - e_k = 1 - \frac{\arccos \rho_k}{\pi} , \tag{4}$$

which comes from the Gaussian correlated distribution of the internal fields of both the student and teacher branch perceptrons, as shown by Mato and Parga [10].

Defining $C_\pm = \cos(x \pm y)$, we have

$$\begin{aligned} P(\mathcal{H},\mathcal{B}) = \; & \textstyle\int_{-\infty}^{+\infty} \frac{dx}{2\pi} e^{ix\mathcal{H}} \int_{-\infty}^{+\infty} \frac{dy}{2\pi} e^{iy\mathcal{B}} \Big[(e_1e_2e_3)C_-^3 + (e_1e_2g_3 + e_1g_2e_3 + g_1e_2e_3)C_-^2C_+ + \\ & (e_1g_2g_3 + g_1e_2g_3 + g_1g_2e_3)C_+^2C_- + (g_1g_2g_3)C_+^3\Big] \; . \end{aligned}$$

Collecting terms with $\mathcal{H}\mathcal{B} < 0$ we get

$$e_g = (e_1e_2e_3) + \frac{1}{2}(e_1 + e_2 + e_3 - e_1e_2 - e_1e_3 - e_2e_3) \; . \tag{5}$$

This result expresses the geometrical relation between the generalization error and the overlaps. The evolution of the overlaps during the learning process is obtained from the learning algorithm prescription of how the synaptic weights are modified upon presentation of a new example, which we take as the most general linear and synchronous dynamics:

$$J_{ki}(\mu+1) = J_{ki}(\mu)\left(1 - \frac{\Omega_k(\mu)}{N}\right) + \frac{F_k(\mu)S_{ki}(\mu)}{N} , \tag{6}$$

where $F_k(\mu)$ defines the algorithm and $\Omega_k(\mu)$ just rescales the norm of $\mathbf{J}_k$. In the $N \to \infty$ limit we have the following equations

$$\frac{\Delta\rho_k(\mu)}{\Delta\alpha} = \frac{\rho_k(\mu)F_k(\mu)}{J_k(\mu)}\left(\frac{b_k(\mu)}{\rho_k(\mu)} - h_k(\mu) - \frac{F_k(\mu)}{2KJ_k(\mu)}\right)$$

$$\frac{\Delta J_k(\mu)}{\Delta\alpha} = J_k(\mu)\left(\frac{F_k^2(\mu)}{2KJ_k^2(\mu)} + \frac{F_k(\mu)h_k(\mu)}{J_k(\mu)} - \Omega_k(\mu)\right) ,$$

where $\alpha = \mu/N$ is the number of presented patterns per adjustable network weight. Averaging over random examples and taking the mean values ρ_k and J_k as aproximations (which will turn out to be very good) for $\rho_k(\mu)$ and $J_k(\mu)$ we get the mean evolution equations

$$\frac{d\rho_k}{d\alpha} = \frac{\rho_k}{J_k}\int D\mu\, F_k(\mu)\left(\frac{b_k(\mu)}{\rho_k} - h_k(\mu) - \frac{F_k(\mu)}{2KJ_k}\right) \tag{7}$$

$$\frac{dJ_k}{d\alpha} = J_k \int D\mu \left(\frac{F_k^2(\mu)}{2KJ_k^2} + \frac{F_k(\mu)h_k(\mu)}{J_k} - \Omega_k(\mu) \right) , \tag{8}$$

where $D\mu$ stands for the measure over random patterns.

The dynamics is described by $2K = 6$ variables ρ_1, ρ_2, ρ_3, J_1, J_2, J_3 and by the initial conditions. The optimization procedure of the next section will uncouple the ρ set from the J set, leading not only to better performance but also to a simpler to analyze dynamics.

3 Dynamics Optimization

From equations (5) and (4) it can be seen that minimization of e_g is obtained by maximizing the increase in the overlaps ρ_k's for each newly presented example. A simple variational method applied to equation (7) leads to the optimal weight F_k^*, given by

$$F_k^* = KJ_k \left(\frac{b_k}{\rho_k} - h_k \right) , \tag{9}$$

where the μ index will be dropped for simplicity. However, the values of $\{b_j\}$ are not known. Define a new weight $\tilde{F}_k$ in which the unaccessible variable b_k is replaced for some unknown function f_k which may depend on anything but the b_j's, *i.e.*

$$\tilde{F}_k = KJ_k \left(\frac{f_k}{\rho_k} - h_k \right) .$$

Inserting $\tilde{F}_k$ into equation (7) we get

$$\frac{d\rho_k}{d\alpha} = \rho_k \int D\mu \left(\frac{f_k b_k}{\rho_k^2} - \frac{f_k^2}{2\rho_k^2} - \frac{h_k b_k}{\rho_k} + \frac{h_k^2}{2} \right) , \tag{10}$$

where we have used the short hand

$$\int D\mu \, (\ldots) = \sum_{\sigma_B = \pm 1} \int P(b_k | \sigma_B, \{h_j\}) P(\sigma_B, \{h_j\}) db_k \left[\prod_{j=1}^{3} dh_j \right] (\ldots) .$$

After integrating on b_k, variational optimization on the f_k function leads to

$$f_k^* = \langle b_k \rangle_k ,$$

where the brackets $\langle \cdots \rangle_k$ stand for the average of b_k given $\sigma_B, \{h_j\}$, *i.e.* average over the posterior distribution probability

$$f_k^* = \int_{-\infty}^{+\infty} db_k P(b_k | \sigma_B, \{h_j\}) b_k = \frac{\int_{-\infty}^{+\infty} db_k P(b_k, \sigma_B, \{h_j\}) b_k}{\int_{-\infty}^{+\infty} db_k P(b_k, \sigma_B, \{h_j\})} . \tag{11}$$

We have thus obtained the accessible optimal weight

$$\tilde{F}_k = KJ_k \left(\frac{\langle b_k \rangle_k}{\rho_k} - h_k \right) . \tag{12}$$

The meaning of this result is clear. The optimal weight that should be attached to a new example depends on b_k (equation (9)). This, however, doesn't comply with the rules of the game and the next best thing that can be done is to replace b_k by the expected value $\langle b_k \rangle_k$. Another point that might be raised is that the overlaps ρ_k's are not accessible either. There are several ways out of this problem. The first and most obvious one is to obtain the value of $\rho_k(\alpha)$ by integration of the evolution differential equations. This leads to mean field type lowerbounds on the generalization error. A second way out of the ρ-problem has been suggested for single layer perceptrons [8], where a self-adaptive algorithm estimates the value of ρ from a measure of the performance on the last few examples. We will now see that the optimization procedure will grant the possibility of a third solution, much more elegant and appealing than the previous two, specially from the point of view of applications. It is important to stress the fact that in equation (11) the dependence of the weight function on the distribution of examples cancels out, leaving us with an *optimal weight function independent of how the examples are chosen.* The effect of such choice will affect only the speed of learning but not the optimal learning algorithm.

In the special case where the examples are independent identical uniformly distributed random variables, and for the nonoverlapping architecture which allows for branch factorization, we have

$$P(\{b_k, h_k\}) = \prod_{k=1}^{3} P_0(b_k, h_k) ,$$

where

$$P_0(b_k, h_k) \equiv P_0(h_k) \times P_0(b_k|h_k) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{h_k^2}{2}\right) \times \frac{1}{\sqrt{2\pi(1-\rho_k^2)}} \exp\left(-\frac{(b_k - \rho_k h_k)^2}{2(1-\rho_k^2)}\right). \tag{13}$$

Taking advantage of the symmetry of the problem, we calculate now $P(\sigma_B, b_1, \{h_j\})$.

$$P(\sigma_B, b_1, \{h_j\}) = P_0(b_1, h_1) \prod_{j\neq 1} P_0(h_j) \int \delta \left(\sigma_B - \mathrm{sign}\left(\sum_{i=1}^{3} \mathrm{sign}(b_i)\right)\right) \prod_{j \neq 1} P_0(b_j|h_j) db_j ,$$

where δ is the Kronecker delta function. Introducing $\lambda_k \equiv \rho_k^{-1}\sqrt{1-\rho_k^2}$ it can be easily shown from equation (13) that

$$P(\sigma_{Bk}|h_k) = H\left(-\frac{\sigma_{Bk} h_k}{\lambda_k}\right) , \tag{14}$$

where the H function is

$$H(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt .$$

Introducing the usual notation for the stabilities $\Delta_k \equiv \sigma_B h_k$ it follows that

$$
\begin{aligned}
P(\sigma_B, b_1, \{h_j\}) = & \quad P_0(b_1, h_1) P_0(h_2) P_0(h_3) \times \\
& \times \left[H\left(-\tfrac{\Delta_2}{\lambda_2}\right) H\left(-\tfrac{\Delta_3}{\lambda_3}\right) + \Theta(\sigma_B b_1) D\left(\tfrac{\Delta_2}{\lambda_2}, \tfrac{\Delta_3}{\lambda_3}\right)\right] ,
\end{aligned} \tag{15}
$$

where Θ is the Heaviside function and

$$
D(y_2, y_3) \equiv H(-y_2)H(y_3) + H(y_2)H(-y_3) . \tag{16}
$$

Integrating over b_1 we obtain

$$
\begin{aligned}
P(\sigma_B, \{h_j\}) = & \quad P_0(h_1) P_0(h_2) P_0(h_3) \times \\
& \times \left[H\left(-\tfrac{\Delta_2}{\lambda_2}\right) H\left(-\tfrac{\Delta_3}{\lambda_3}\right) + H\left(-\tfrac{\Delta_1}{\lambda_1}\right) D\left(\tfrac{\Delta_2}{\lambda_2}, \tfrac{\Delta_3}{\lambda_3}\right)\right] .
\end{aligned} \tag{17}
$$

Introducing further simplifications in the notation, $y_k \equiv \Delta_k/\lambda_k$, we finally arrive at the result

$$
P(b_1|\sigma_B, \{h_j\}) = P_0(b_1|h_1) \times \left[\frac{H(-y_2)H(-y_3) + \Theta(\sigma_B b_1) D(y_2, y_3)}{H(-y_2)H(-y_3) + H(-y_1) D(y_2, y_3)} \right] . \tag{18}
$$

We are able now to evaluate the optimal weight, according to equation (12). Straightforward calculations using equations (13), (16) and (18) yield

$$
\tilde{F}_k = \frac{3}{\sqrt{2\pi}} J_k \lambda_k \sigma_B e^{-\left(\Delta_k^2/2\lambda_k^2\right)} g\left(\frac{\Delta_1}{\lambda_1}, \frac{\Delta_2}{\lambda_2}, \frac{\Delta_3}{\lambda_3}\right) , \tag{19}
$$

where

$$
g(y_1, y_2, y_3) \equiv \frac{D(y_2, y_3)}{H(-y_2)H(-y_3) + H(-y_1) D(y_2, y_3)} . \tag{20}
$$

Inserting result (19) in the mean evolution equation (7), we see that the optimization procedure left the ρ_1 equation uncoupled from the $\{J_k\}$ equations. Integrating over b_1 and summing over σ_B we obtain

$$
\frac{d\rho_1}{d\alpha} = \frac{3}{4\pi} \rho_1 \lambda_1^2 \int \left(\prod_{j=1}^{3} Dh_j \right) e^{-h_1^2/\lambda_1^2} \frac{D^2\left(\frac{h_2}{\lambda_2}, \frac{h_3}{\lambda_3}\right)}{r\left(\frac{h_1}{\lambda_1}, \frac{h_2}{\lambda_2}, \frac{h_3}{\lambda_3}\right)\left[1 - r\left(\frac{h_1}{\lambda_1}, \frac{h_2}{\lambda_2}, \frac{h_3}{\lambda_3}\right)\right]} , \tag{21}
$$

where Dx is the Gaussian measure

$$
Dx = dx \, \frac{e^{-x^2/2}}{\sqrt{2\pi}}
$$

and

$$
r\left(\frac{h_1}{\lambda_1}, \frac{h_2}{\lambda_2}, \frac{h_3}{\lambda_3}\right) \equiv P(\sigma_B = -1|\{h_j\}) = H\left(\frac{h_2}{\lambda_2}\right) H\left(\frac{h_3}{\lambda_3}\right) + H\left(\frac{h_1}{\lambda_1}\right) D\left(\frac{h_2}{\lambda_2}, \frac{h_3}{\lambda_3}\right) . \tag{22}
$$

Inserting the optimal weight in equation (8), the choice $\Omega_k = 0$ leads to the equation for the evolution of the branches norms:

$$\frac{dJ_1}{d\alpha} = \frac{3}{4\pi} J_1 \lambda_1^2 \int \left(\prod_{j=1}^{3} Dh_j \right) e^{-h_1^2/\lambda_1^2} \frac{D^2\left(\frac{h_2}{\lambda_2}, \frac{h_3}{\lambda_3}\right)}{r\left(\frac{h_1}{\lambda_1}, \frac{h_2}{\lambda_2}, \frac{h_3}{\lambda_3}\right)\left[1 - r\left(\frac{h_1}{\lambda_1}, \frac{h_2}{\lambda_2}, \frac{h_3}{\lambda_3}\right)\right]} , \tag{23}$$

so that, as can be seen from equations (21) and (23),

$$J_k(\alpha) = c \times \rho_k(\alpha) , \tag{24}$$

where c is a constant and we can self-consistently choose $c = 1$, since ρ_k enters the J_k dynamics (23). Such a choice leaves these two equations uncoupled and, what is even more interesting, identical. That means that, in the mean,

$$J_k(\alpha) = \rho_k(\alpha), \tag{25}$$

as long as $J_k(0) = \rho_k(0)$.

4 Theoretical results and simulations

The branch permutation symmetry displayed by the set of differential equations (7) or (21) allows for a branch symmetric solution whenever the initial conditions are themselves symmetric, *i.e.* $\rho_k(\alpha = 0)$ is independent of k. The generalization error (5) can be obtained once the overlaps have been obtained by numerically integrating equation (21). The result is shown in Figure 1 (lower solid curve). The symbols represent the results of two different numerical simulations which were performed as follows. The first type of simulation (represented in Figure 1 by diamonds) uses the value of $\rho(\alpha)$ obtained from the numerical integration of equation (21), in evaluating the weight function $\tilde{F}_k$:

$$\tilde{F}_1(\sigma_B, \tilde{h}_1, \tilde{h}_2, \tilde{h}_3, \alpha) = \frac{3}{\sqrt{2\pi}} \mathcal{D}(\alpha) \sigma_B e^{-\left(\tilde{h}_1^2/2\mathcal{D}^2(\alpha)\right)} g\left(\frac{\sigma_B \tilde{h}_1}{\mathcal{D}(\alpha)}, \frac{\sigma_B \tilde{h}_2}{\mathcal{D}(\alpha)}, \frac{\sigma_B \tilde{h}_3}{\mathcal{D}(\alpha)} \right) , \tag{26}$$

where $\mathcal{D}(\alpha) \equiv \sqrt{1 - \rho^2(\alpha)}$. The second type of simulation (represented by circles) is done using the decoupling property of the optimized dynamics and instead of using the *a priori* known value of $\rho(\alpha)$ it uses the accessible values of the norms $\{J_k\}$ as measured during run time. That means that the weight function is

$$\tilde{F}_1(\sigma_B, \tilde{h}_1, \tilde{h}_2, \tilde{h}_3, J_1, J_2, J_3) = \frac{3}{\sqrt{2\pi}} \mathcal{D}_1 \sigma_B e^{-\left(\tilde{h}_1^2/2\mathcal{D}_1^2\right)} g\left(\frac{\sigma_B \tilde{h}_1}{\mathcal{D}_1}, \frac{\sigma_B \tilde{h}_2}{\mathcal{D}_2}, \frac{\sigma_B \tilde{h}_3}{\mathcal{D}_3} \right) , \tag{27}$$

where $\mathcal{D}_k \equiv \sqrt{1 - J_k^2}$ and in this case the value of J_k in each step is measured and used. The fact that both simulations give rise to identical results, within error bars, shows that the last algorithm is immune to the fluctuations in the values of J_k. In this form we have a perfectly well defined and applicable algorithm.

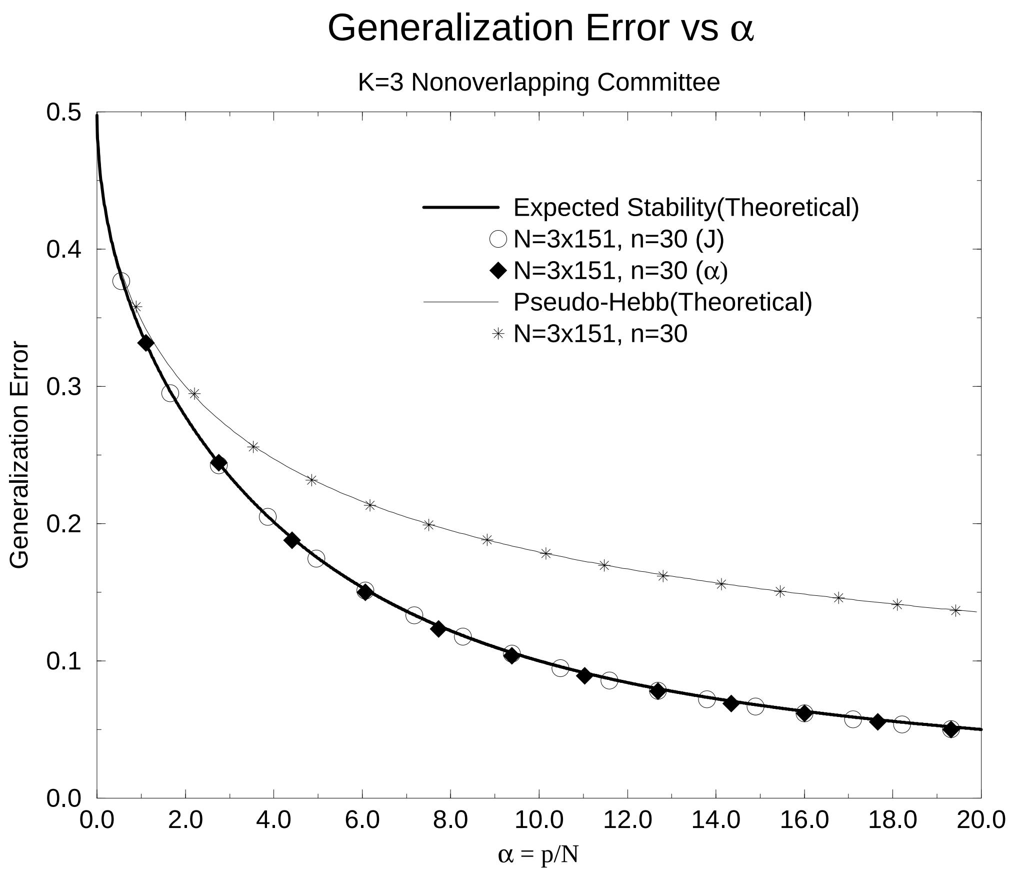

Figure 1: Theoretical results (solid lines) and simulations (symbols) for the pseudo-Hebb (upper curve) and Expected Stability (lower curve) algorithms. N is the number of boolean input units and n is the number of networks used for obtaining the mean ρ_k's. Standard error bars would be approximately the same size of the symbols. See text for details.

Still in Figure 1, and for the sake of comparison, we present the theoretical results accompanied by those of simulations, for what we dubbed the pseudo-Hebb algorithm. This is just a generalization of the Hebb rule to the committee machine. The hebbian rule uses a weight function independent of the pos-synaptic fields $\{h_k\}$. If we allow for a possibly varying step size, as a function of α but not of the stabilities, the most general weight function, within what might be called the pseudo-Hebb class, is given by

$$F_k^{pH} = \sigma_B W_k(\alpha) \ . \tag{28}$$

It is curious that optimization of W_k in the sense of mean generalization gives $W_k(\alpha) =$ constant, yielding an optimal weight independent of α, a result previously obtained in [5] for the boolean perceptron with no hidden layers. The overlaps' dynamics is given by

$$\frac{d\rho_k}{d\alpha} = \frac{3}{4\pi}\frac{(1-\rho_k^2)^2}{\rho_k} \tag{29}$$

and for the initial condition $\rho(0) = 0$ leads to

$$\rho_k(\alpha) = \left(1 + \frac{2\pi}{3\alpha}\right)^{-1/2} , \tag{30}$$

implying in an asymptotic error $e_g^{pH} \simeq \sqrt{3/2\pi}\alpha^{-1/2}$. This should be compared to the error decay of the optimized algorithm which gives, for large α,

$$e_g \simeq \frac{\kappa}{\alpha} , \tag{31}$$

where

$$\kappa = 2 \times \left[\int_{-\infty}^{+\infty} Dx\, \frac{e^{-x^2/2}}{H(x)}\right]^{-1} \simeq 0.88 \ . \tag{32}$$

This is *exactly* the same nominal error of the expected stability algorithm for the perceptron with no hidden layers.

5 Conclusions

The analysis of the optimal weight function (19) yields some very interesting results. This analysis is based on the interpretation of the weight function as a measure of the information value as judged from the previous experience acquired during the learning process. Two elements govern the behaviour of the weight function. In the first place, it depends on the the error of generalization through its dependence on the ρ_k's. This means that the synaptic couplings will be changed differently depending on whether the net is in an advanced stage of learning or just beginning. These remarks are the basis of the construction of a self adaptive algorithm. Figure 2 shows the weight function dependence on the branch stability Δ_1 in the special case where $\Delta_2 = -3$ and $\Delta_3 = 3$, for different learning stages as parametrized by the overlaps ρ_k. Note the evolution of F_1 from a flat, pure pseudo-Hebbian (*i.e.* constant) weight at the beginning of the process to a highly structured function of the stability Δ_1 at later times. The second element is what might be dubbed the element of surprise of a new example. Consider Figure 3, where the weight function (rescaled by $1/\sqrt{1-\rho_1^2}$) is shown as a function of $y_1 = \Delta_1/\lambda_1$ for various values of the pair y_2, y_3. By presenting the values in such rescaled form we can analyze the effect of the surprise factor independently of the stage of learning. Notice that if a given branch-variable y_k is positive that branch gives a contribution which agrees with the teacher output. In general if the internal representation agrees with the teacher then no major change in the synaptic couplings is performed. However, and specially when the other two branches disagree (either between themselves or with the teacher), the cases with $y_1 < 0$ lead to recognizing a surprise and thus performing a major synaptic change. However, if y_1 gets to be even more negative than y_2 and y_3, then the increased confidence on being correct, although

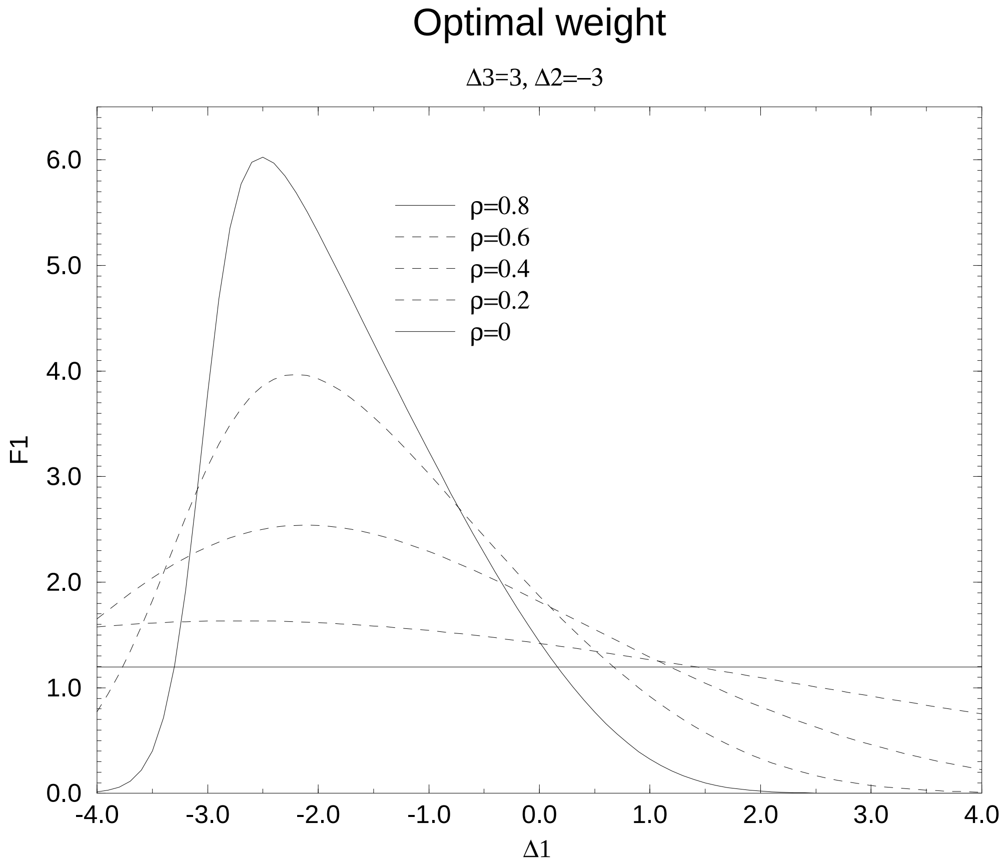

Figure 2: Optimal weight function for the first branch in the symmetrical situation where $\rho_1 = \rho_2 = \rho_3 = \rho$.

with $\sigma_{J1} = -\sigma_B$, leads to a decrease in the weight function as if this meant that the blame for a wrong classification lies somewhere on the other branches.

A final point that deserves to be mentioned is the fact that as in all the optimized on-line dynamics for spherical perceptrons, the energy function per example is related to the conditional probability

$$E = -\lambda^2 \ln P(\sigma_B|\{h_j\}) \ .$$

In conclusion the study of the learning dynamics of a net with three hidden units has led to algorithms that can be used in a practical learning task where this architecture is suited. The uncoupling of the ρ and J variables due to the optimization procedure rids the algorithm from unknown parameters and yields a weight function that can be evaluated with the available information.

Acknowledgements: The authors thank O. Kinouchi for several useful discussions. The work of M. Copelli was supported by FAPESP, while N. Caticha received support from CNPq.

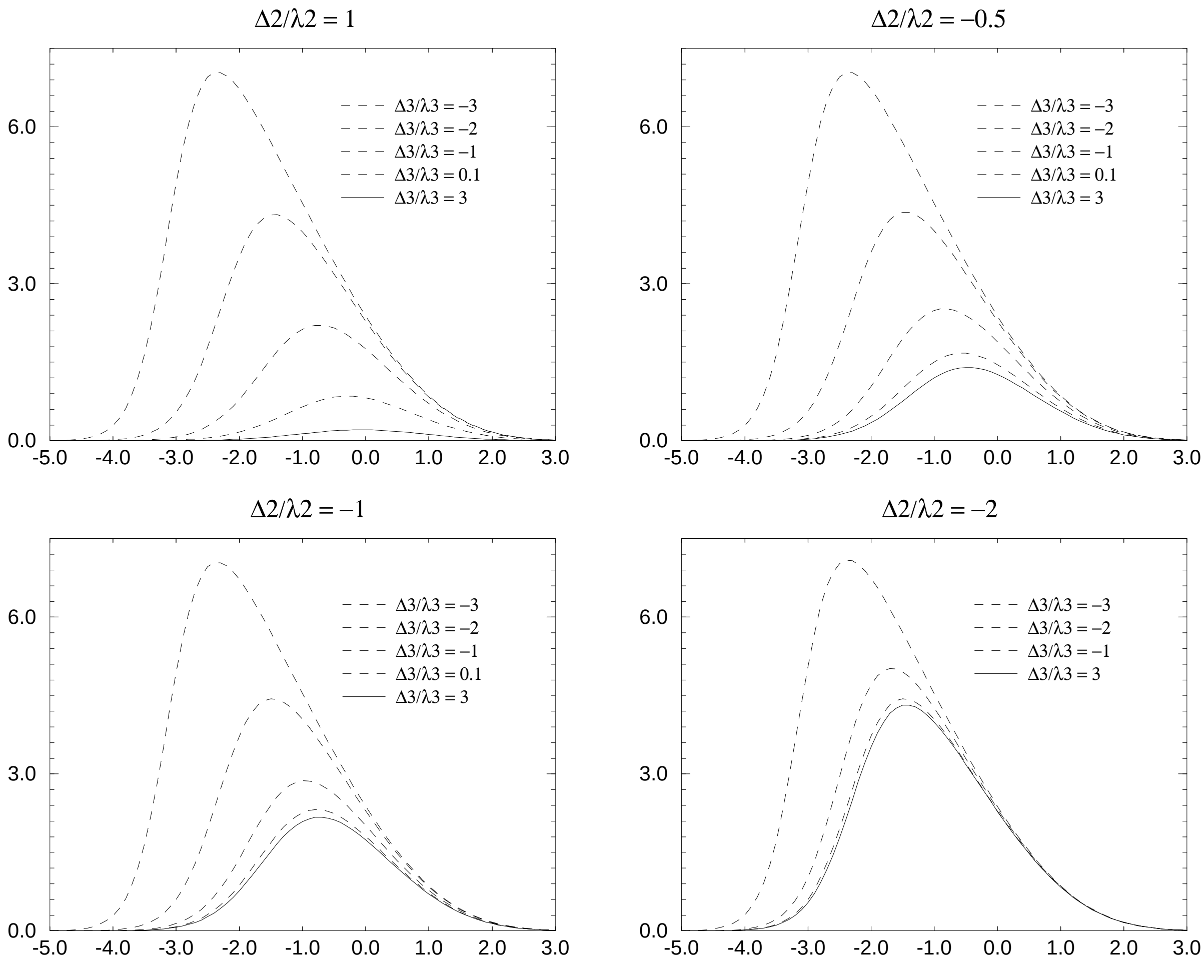

Figure 3: Rescaled optimal weight function $(1-\rho^2)^{-1/2}F_1$ against rescaled stability $y_1 = \Delta_1/\lambda_1$ for several values of Δ_2/λ_2 and Δ_3/λ_3. Solid lines represent the asymptotic behaviour for increasing Δ_3/λ_3 with fixed Δ_2/λ_2.

References

[1] Biehl, M., Schwarze, H., "Learning by Online Gradient Descent", Preprint, Institut für theoretische Physik, Universität Würzburg (1994)

[2] Gardner, E., Derrida, B., "Three unfinished works on the optimal storage capacity of networks", *J.Phys. A: Math. Gen.* **22** 1983 (1989)

[3] Györgyi, G., Tishby, N., "Statistical theory of learning a rule" *in* "Neural Networks and spin glasses", Theumann, W., Köberle, R., Eds. World Scientific, 1989

[4] Kabashima, Y., "Perfect loss of generalization due to noise in K=2 parity machines", *J.Phys. A: Math. Gen.* **27** 1917 (1994)

[5] Kinouchi, O., *Optimal Generalization in Perceptrons*, Master's thesis, IFQSC, Universidade de São Paulo, São Carlos/SP, (1992)

[6] Kinouchi O., Caticha N., "Biased learning in boolean perceptrons", *Physica* **185A** 411 (1991)

[7] Kinouchi O., Caticha N., "Optimal generalization in perceptrons", *J.Phys. A: Math. Gen.* **25** 6243 (1992)

[8] Kinouchi, O., Caticha, N., "Lower bounds on generalization errors for drifting rules", *J. Phys. A: Math. Gen.* **26** 6161 (1993)

[9] Kinzel, W., Rubán, P., "Improving a Network Generalization Ability by Selecting Examples", *Europhys. Lett.* **13** 473 (1990)

[10] Mato, G., Parga, N., "Generalization properties of multilayered neural networks", *J. Phys. A: Math. Gen.* **25** 5047 (1992)

[11] Opper M., Haussler D., "Generalization performance of Bayes optimal classification algorithm for learning a perceptron", *Phys. Rev. Lett.* **66** 2677 (1991)