Optimal generalization in perceptions

J. Phys. A: Math. Gen. 25 (1992) 6243–6250. Printed in the UK

Optimal generalization in perceptrons

Osame Kinouchi† and Nestor Caticha‡

† IFQSC, Universidade de São Paulo, CP 369, 13560 São Carlos, SP, Brazil
‡ Instituto de Física, Universidade de São Paulo, CP 20516, 01498 São Paulo, SP, Brazil

Received 16 January 1992

Abstract. A new learning algorithm for the one-layer perceptron is presented. It aims to maximize the generalization gain per example. Analytical results are obtained for the case of single presentation of each example. The weight attached to a Hebbian term is a function of the expected stability of the example in the teacher perceptron. This leads to the obtention of upper bounds for the generalization ability.

This scheme can be iterated and the results of numerical simulations show that it converges, within errors, to the theoretical optimal generalization ability of the Bayes algorithm.

Analytical and numerical results for an algorithm with maximized generalization in the learning strategy with selection of examples are obtained and it is proved that, as expected, orthogonal selection is optimal. Exponential decay of the generalization error is obtained for the single presentation of selected examples.

In the statistical mechanics approach to learning from examples and generalization by neural nets [1–4], the single-layer perceptron has been the preferred laboratory [5–11]. This is certainly due to its simplicity which affords relevant results from simple calculations and simulations. Despite its simplicity it has revealed a variety of interesting properties and, despite all the efforts, not all of them have been totally understood.

The perceptron generalization problem most studied is that of learning a linearly separable Boolean function

$$\mathcal{B}(\boldsymbol{S}) \equiv \sigma_{\mathrm{B}} = \operatorname{sign}(\boldsymbol{B} \cdot \boldsymbol{S}) \tag{1}$$

where $\boldsymbol{S}$ is an input vector with N Ising components and $\boldsymbol{B}$ is a vector in $\mathbb{R}^N$. The Boolean function is equivalent to the output of a 'teacher perceptron' with synaptic coupling vector equal to $\boldsymbol{B}$, which can be taken to be normalized to one. The task of the 'student perceptron' $\boldsymbol{J}$ is to approximate this function by using only the information contained in a 'learning set' $\mathcal{L}$ of P $(= \alpha N)$ examples. An example is a pair $(\boldsymbol{S}_\mu, \sigma_{\mathrm{B}}^\mu)$ of input vector $\boldsymbol{S}_\mu$ and correct output σ_{B}^μ.

Two learning strategies, as defined by Valiant [12], will be studied. In the first one, examples are randomly drawn with a fixed probability distribution, here uniform in $\mathbb{R}^N$. In the second, which has been called learning from an 'oracle', or with selection of examples [9], the teacher gives the correct answer to questions $\boldsymbol{S}$ appropriately chosen by the student during the learning process. We stress our use of the word strategy as referring to the actual probability distribution used to obtain the examples.

0305-4470/92/236243+08$07.50

The quantity of interest is the generalization ability $G(\alpha)$, defined as the probability that a new random input $\boldsymbol{S}_\mu$, statistically independent of the learning set, be well classified by the student perceptron. It depends only on α, in the thermodynamic limit $N \to \infty$ [4, 7]

$$G(\alpha) = 1 - \frac{1}{\pi} \cos^{-1}(\rho(\alpha)) \tag{2}$$

where ρ is the average overlap of the teacher and the student, $\rho = R/J$, $R = \boldsymbol{B} \cdot \boldsymbol{J}$ and $J = \sqrt{\boldsymbol{J} \cdot \boldsymbol{J}}$. The error of generalization is $e_g = 1 - G(\alpha)$. It is also useful to define the learning error e_l, which is the probability of misclassifying a vector belonging to the learning set.

The overlap ρ is assumed to have self-averaging properties, and thus is independent of the particular learning set in the thermodynamic limit. Starting from a *tabula rasa* $J_1 = 0$, learning is achieved through a generalized Hebbian prescription [3]

$$\boldsymbol{J}_{\mu+1} = \boldsymbol{J}_\mu + \frac{1}{N} W_\mu \sigma_{\mathrm{B}}^\mu \boldsymbol{S}_\mu . \tag{3}$$

The Hebbian term is weighted by the function W_μ, up to now unspecified, which may depend on the previous states of the synaptic couplings. It may be called the 'attention' paid to that particular example μ. It follows that

$$R_{\mu+1} = R_\mu + \frac{1}{N} W_\mu \sigma_{\mathrm{B}}^\mu b_\mu \tag{4}$$

$$J_{\mu+1} = J_\mu \left[1 + \frac{1}{N} \left(\frac{W_\mu \sigma_{\mathrm{B}}^\mu h_\mu}{J_\mu} + \frac{W_\mu^2}{2J_\mu^2} \right) \right] \tag{5}$$

where only terms up to order $1/N$ have been kept, and where

$$b_\mu = \boldsymbol{B} \cdot \boldsymbol{S}_\mu \qquad \text{and} \qquad h_\mu = \frac{\boldsymbol{J}_\mu \cdot \boldsymbol{S}_\mu}{J_\mu} .$$

In the case of single presentation of the examples, b_μ and h_μ are Gaussian correlated variables with joint probability distribution

$$\begin{aligned} P(b_\mu, h_\mu) &= P(h_\mu) P(b_\mu | h_\mu) \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{h_\mu^2}{2} \right) \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp\left(-\frac{(b_\mu - \rho h_\mu)^2}{2(1-\rho^2)} \right) \end{aligned} \tag{6}$$

and $\rho = \rho_\mu$. The overlap evolution is given by

$$\rho_{\mu+1} = \rho_\mu + \frac{1}{N J_\mu} \left[(b_\mu - \rho_\mu h_\mu) \sigma_{\mathrm{B}}^\mu W_\mu - \frac{\rho_\mu W_\mu^2}{2J_\mu} \right] . \tag{7}$$

At this point we notice that if the normalization of $\boldsymbol{J}_\mu$ had been chosen to be spherical, equation (3) would have extra terms to account for the constraint, but equation (7) would be unchanged. After averaging over the possible choices of $\boldsymbol{S}_\mu$, and taking the thermodynamic limit a differential equation is obtained for the evolution of ρ

$$\frac{\mathrm{d}\rho}{\mathrm{d}\alpha'} = \frac{1}{J}\int_{-\infty}^{\infty} \mathrm{d}h_\mu\,\mathrm{d}b_\mu\, P(b_\mu, h_\mu)\left[(b_\mu - \rho h_\mu)\sigma_{\mathrm{B}}^{\mu} W_\mu - \frac{\rho W_\mu^2}{2J}\right] \tag{8}$$

where $\alpha' = (\mu/P)\alpha$, refers to the fraction of examples already presented. This equation describes the 'rule extraction speed' of the learning algorithm, and is a funtional of W. Since maximizing $\mathrm{d}\rho/\mathrm{d}\alpha'$ maximizes the gain in generalization ability per example, the problem of determining W turns into a simple variational problem. Its solution is

$$W_\mu^* = J(\kappa_\mu - \Delta_\mu) \tag{9}$$

where

$$\kappa_\mu = \sigma_{\mathrm{B}}^{\mu} b_\mu/\rho \qquad \text{and} \qquad \Delta_\mu = \sigma_{\mathrm{B}}^{\mu} h_\mu \tag{10}$$

are Gardner-like parameters and the local stability of example μ respectively. The parameters κ_μ are the desired stabilities of the examples (divided by ρ). They have a Gaussian distribution truncated at zero. It can be seen that forcing large stabilities, as in the random mapping case, will lead to overfitting of the examples, and it is thus not a good learning procedure if generalization ability is to be stressed.

The solution W_μ^* can only be used by the linear perceptron or any other with an invertible activation function, since it requires knowledge of b_μ. For the perceptron with activation function given by (1), this is not possible and the best thing that can be done is to use the expected value of $|b_\mu|$ given the local field h_μ and the teacher output $\sigma_{\mathrm{B}}^{\mu}$. Since W^* is linear in $|b|$ then

$$\overline{W}(\rho_\mu, J_\mu, \Delta_\mu) = \frac{\int \mathrm{d}|b|\, P(b,h) W_\mu^*}{\int \mathrm{d}|b|\, P(b,h)}. \tag{11}$$

We have previously studied a related algorithm [11] where the expected value of b_μ was used. A smaller generalization is achieved since not all the available information was used. Using (5) the weight function

$$\overline{W}(\rho_\mu, J_\mu, \Delta_\mu) = \frac{1}{\sqrt{2\pi}} J_\mu \lambda_\mu \exp\left(-\frac{\Delta_\mu^2}{2\lambda_\mu^2}\right)\frac{1}{H(-\Delta_\mu/\lambda_\mu)} \tag{12}$$

is obtained, where

$$\lambda = \frac{\sqrt{1-\rho^2}}{\rho} = \tan(\pi e_{\mathrm{g}}) \tag{13}$$

and

$$H(x) = \int_x^{\infty} \frac{\mathrm{d}t}{\sqrt{2\pi}} \mathrm{e}^{-t^2/2} = \frac{1}{2}\mathrm{erfc}\left(\frac{x}{\sqrt{2}}\right). \tag{14}$$

This weight function still depends on ρ. By introducing it into the differential equation ((7)) governing its evolution, it follows that

$$\frac{\mathrm{d}\rho}{\mathrm{d}\alpha'} = \frac{1-\rho^2}{2\pi\rho}\int_{-\infty}^{\infty} \mathrm{D}h \frac{\exp(-h^2/\lambda^2)}{H(h/\lambda)} \tag{15}$$

where $\mathrm{D}h$ is the Gaussian measure $(2\pi)^{-1/2}\mathrm{e}^{-h^2/2}\,\mathrm{d}h$. Numerical integration leads to the value of $\rho(\alpha')$ which is used in equation (12) to define the actual algorithm used to perform the simulations. Although it still depends on J this presents no problem, since from equation (5) a differential equation for the evolution of $J(\alpha)$ can be obtained and it leads to $J(\alpha) = \rho(\alpha)$.

In figure 1 the resulting weight function is shown, together with the corresponding weight functions for the pure Hebbian prescription, the perceptron, adaline and the relaxation algorithms. For each of these methods, the better its weight function approximates the weight function W the better its performance will be. It is reasonable to call this learning procedure the 'expected stability' algorithm. In figure 2 the theoretical prediction for the generalization ability is compared with a numerical simulation. The pure Hebbian result is presented (lower curve) for comparison. Also the learning curves for the Hebbian (upper full) and 'expected stability' (squares) are shown. Even though the algorithm stresses the generalization properties, it also leads to an improvement in the rote memorization. The generalization error decays as $0.88/\alpha$ for large α, whereas the pure Hebbian is only proportional to $\alpha^{-1/2}$.

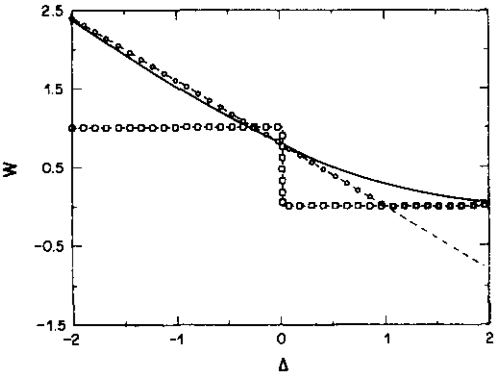

Figure 1. Examples of weight functions of the 'expected stability' (full curve), simple Hebbian (dotted), perceptron (squares-dotted), adaline (broken curve) and the relaxation (circles-broken) algorithms for fixed α.

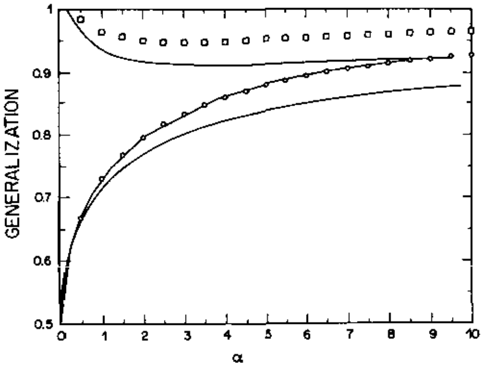

Figure 2. Generalization ability $G(\alpha)$ plotted against number of examples α $(= P/N)$. The full curve is obtained from a numerical integration of equation (15). The circles (generalization) and squares (memorization) are the result of a simulation with $N = 119$ averaged over 100 runs for the 'expected stabilty' algorithm. The lower curve is the pure Hebb [6].

This result can be thought of as being an upper bound to the generalization ability fo the 'single presentation of examples' case. The upper bound for the iterated presentations, where all information of each example is extracted has been calculated by Opper and Haussler [8]. In this case the error decays as $0.44/\alpha$, exactly (!) half the single presentation error.

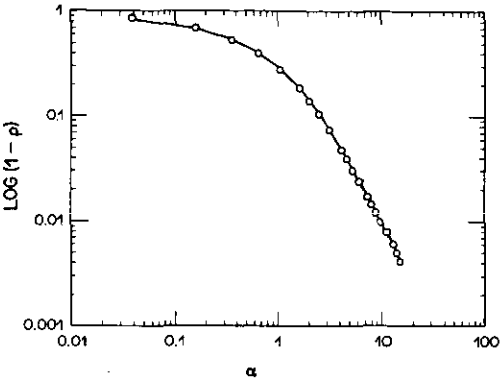

Figure 3. Comparison of average overlap $\rho(\alpha)$ of the iterated 'expected stability' algorithm (circles) averaged over 20 runs, $N = 149$ after 50 iterations, with the Bayes algorithm $\rho_{\mathrm{B}}(\alpha)$ (full curve), [8].

It is interesting and natural to examine how this algorithm will fare under an iterative scheme. We are not able to say whether the generalization gain per example will be maximized by using the same weight function (12) as before. We have only proved this optimal result for the first step of this sequential dynamics. The generalization is not totally straightforward. Notice that the ρ and J dependence on α and on the iteration stage n_{iter} are not known. The questions that are raised are what values are most appropriate for them for this problem. We have used the following recipe. We have set the J parameter equal to one. Its value is very near ρ and it does not affect the simulations. It is clear that during a numerical simulation we have access to the value of ρ. We have tested the numerical behaviour of the method in a simulation with the measured value of the overlap $\rho(\alpha)$ substituting the ρ parameter in the weight function. This is not very realistic and by this we are just judging the potential of the algorithm if ρ were known. After numerical convergence the performance was found to be very close to the value of the Bayes algorithm of Opper and Haussler [8], which cannot be implemented on a one-layer net, but gives a theoretical upper bound. Its performance is obtained from [8]

$$\frac{\rho_{\mathrm{B}}^2}{\sqrt{1-\rho_{\mathrm{B}}^2}} = \frac{\alpha}{\pi}\int_{-\infty}^{\infty} \mathrm{D}t\,\frac{\exp(-t^2\rho_{\mathrm{B}}^2/2)}{H(t\rho_{\mathrm{B}})} \tag{16}$$

which follows from a self-consistent replica symmetric calculation. This suggests an approximation which actually consists in using the known Bayes value $\rho_{\mathrm{B}}(\alpha)$ for $\rho(\alpha, n_{\mathrm{iter}})$ in the weight function. The result of a numerical simulation is shown in figure 3. The actual performance of the perceptron is seen to converge to that of the Bayes algorithm. We do not claim to have other than quite strong numerical evidence for this algebraically fast convergence in the number of the iterations. The difference between ρ_{B} and ρ is smaller than 10^{-3}, with the simulated result being the larger due to finite-size effects. It is interesting to note that the generalization error e_{g} and the learning error e_{l} converge at approximately the same rate (figure 4). Thus the measurement of e_{l} can be used in practice to decide when to stop the learning (iterative) phase. The learning error has been found to be zero up to a value $\alpha_{\mathrm{c}} \approx 0.8$, and is smaller than 2×10^{-3} for any α.

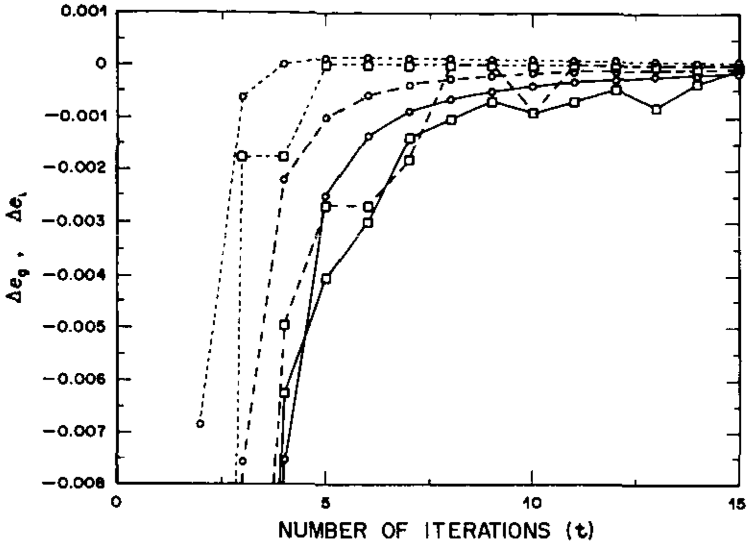

Figure 4. Convergence of the iterated 'expected stability' algorithm: under iteration e_g (full curves) and e_l (broken curves) converge at approximately the same rate for fixed α.

Now the second learning strategy is considered. Learning with selection of examples has been previously studied in [9, 10]. If the examples are chosen in any special way, then the distribution $P(h)$ is modified. The evolution is then governed by

$$\frac{\mathrm{d}\rho}{\mathrm{d}\alpha} = \frac{1-\rho^2}{4\pi\rho}\int_{-\infty}^{\infty}\mathrm{d}h\, P(h)\exp\left(\frac{-h^2}{\lambda^2}\right)\left[\frac{1}{H(h/\lambda)}+\frac{1}{H(-h/\lambda)}\right] \tag{17}$$

and the gain per example can be seen to be maximized if $P(h)$ is chosen to be a delta function centred at $h = 0$, $P(h) = \delta(h)$. That means that only examples orthogonal to $\boldsymbol{J}_\mu$, the accumulated knowledge, will be used during this learning process. This justifies the heuristics of the selection criterion of Kinzel and Ruján [9], who studied the case of selections of examples with a Hebbian weight rule $W = 1$. The weight function is obtained from equation (12)

$$\bar{W}(\rho_\mu, J_\mu, \Delta_\mu) = \sqrt{\frac{2}{\pi}}J_\mu\lambda_\mu . \tag{18}$$

In our case equation (17) can be easily solved to yield

$$\rho = \sqrt{1-\mathrm{e}^{-2\alpha/\pi}} \tag{19}$$

thus the weight function is $W = \sqrt{2/\pi}\exp(-\alpha/\pi)$. Equation (19) shows that the selection of examples with weight given by equation (18) leads, somewhat suprisingly to expontential decrease of the generalization error, $e_g \simeq 1/\pi\exp(-2\alpha/\pi)$, whereas it is bounded by $0.199/\alpha^{-1/2}$ when $W = 1$, and without selection of examples the error only decays algebraically as $e_g \simeq 0.44/\alpha$ at most. Figure 5 shows the results

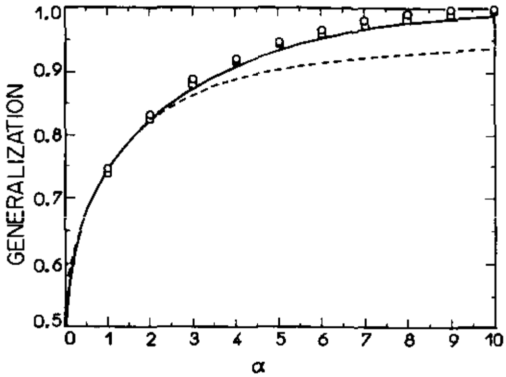

Figure 5. Selection of examples. Expected generalization stability for $N = 149$ (circles), $N = 249$ (squares), both averaged over 30 runs. The full curve is the theoretical value from equation (18) and the broken curve is the Hebbian ($W = 1$) as in [9].

of a numerical simulation compared with the analytical prediction as well as the case where $W = 1$ [9]. Finite-size effects account for the differences.

The learning dynamics of these algorithms can be understood as minimizing an 'energy function' E. For both strategies, with and without selection of examples, it is easy to see that

$$E = \lambda^2 \ln(P(\sigma|h)) \tag{20}$$

where the conditional probability $P(\sigma|h)$ of σ given h is

$$P(\sigma|h) = H(-\Delta/\lambda). \tag{21}$$

We now argue that the energy function E can be thought of as a measure of the 'value of information' of a given example, for this particular problem. Although this concept has not been quantitatively defined in general, it has been discussed in the literature [13] as related to the 'degree of non-redundancy' or 'independence' of each example's information content. The value of information is supposed to depend on the particular task to be implemented and on the state of the receptor, while the Shannon information content is an absolute quantity independent of task and receptor's previous experience. For instance, consider an example with high overlap h, which is well classified by the student perceptron. It will certainly be of very little value to modify any possible difference between B and J. On the other hand if a high-overlap example is misclassified, the weight will be very large and also its value of information. Note that a high overlap h means high *a priori* confidence (stability under addition of noise) in classifying the example and the misclassification of this putative easy example means that a high value of information should be attributed to it. The selection of examples works by choosing examples with a reasonable high value of information.

In conclusion a new learning algorithm, has been presented which aims to maximize the generalization ability. The first step of the learning dynamical process has been studied analytically and compared to numerical results, the agreement is excellent (figures 2 and 5) for both learning strategies with and without selection of examples. It gives an upper bound to generalization in the single presentation case.

The expected stability algorithm can be generalized so that the result of iterated presentation of examples can be studied. Numerical results of the asymptotical behaviour have been presented. These seem to saturate the theoretical bound of the Bayes algorithm. Whether this is true or not remains to be seen and it certainly deserves further study. The dynamical properties of this iterative scheme will be the subject of future work.

After this work was completed, we received a preprint from Meir and Fontanari [14] where a relaxation algorithm with an α-dependent κ parameter was studied. It also seems, at least numerically, to saturate the Bayes bound. Their choice of an optimal $\kappa(\alpha)$ in the relaxation algorithm leads to a weight function which approximates $\overline{W}(\rho, J, \Delta)$ of equation (12), at least in the region where h_μ is close to zero.

Acknowledgments

The authors thank J F Fontanari for a discussion on the convergence of the algorithm. OK has received financial support from a CAPES graduate fellowship, while the research of NC has been partially supported by CNPq.

References

[1] Denker J, Schwartz D, Wittner B, Solla S, Howard R, Jackel L and Hopfield J J 1987 Large automatic learning, rule extraction and generalization *Complex Systems* **1** 877

[2] Levin E, Tishby N and Solla S A 1989 A statistical approach to learning and generalization in layered natural networks *Proc. 2nd Ann. Workshop on Computational Learning Theory (COLT–89)* ed R Rivest, D Haussler and M K Warmuth (San Mateo, CA: Morgan Kaufmann)

[3] Abbott L F 1990 Learning in neural network memories *Network* **1** 105

[4] Györgyi G and Tishby N 1989 Statistical theory of learning a rule *Proc. STATPHYS 17 (Workshop on Neural Nets and Spin Glasses)* ed W Theumann and R Köberle (Singapore: World Scientific)

[5] Seung H S, Sompolinsky H and Tishby N 1992 Statistical mechanics of learning from examples *Phys. Rev.* A **45** 6056

[6] Vallet F 1989 The Hebb Rule for learning linearly separable Boolean Functions: learning and generalization *Euro. Phys. Lett.* **8** 747

Vallet F and Cailton J-G 1990 Recognition rates of the Hebb rule for learning Boolean functions *Phys. Rev.* A **41** 3059

[7] Opper M, Kinzel W, Kleinz J and Nehl R 1990 On the ability of the optimal perceptron to generalise *J. Phys. A: Math. Gen.* **23** L581

[8] Opper M and Haussler D 1991 Generalization performance of Bayes optimal classification algorithm for learning a perceptron *Phys. Rev. Lett.* **66** 2677

[9] Kinzel W and Rujàn P 1990 Improving a network generalization ability by selecting examples 1990 *Europhys. Lett.* **13** 473

[10] Watkin T L H and Rau A 1991 Selecting examples for perceptrons *J. Phys. A: Math. Gen.* **25** 113

[11] Kinouchi O and Caticha N 1992 Biased learning in Boolean perceptrons *Physica* **185A** 411

[12] Valiant G L 1984 A theory of the learnable *Comm. ACM* **27** 1134

[13] Brillouin L 1971 *Science and Information Theory* 2nd edn (New York: Academic)

Volkenshtein M V 1987 Information theory and evolution *Cybernetics of Living Matter* ed I M Makarov (Moscow: Mir)

[14] Meir R and Fontanari J F 1991 On the calculation of learning curves for inconsistent algorithms *Preprint* IFQSC, Universidade de São Paulo