UNIVERSIDADE DE SÃO PAULO

INSTITUTO DE FÍSICA
CAIXA POSTAL 20516
01498 SÃO PAULO - SP
BRASIL

PUBLICAÇÕES

IFUSP/P-962

OPTIMAL GENERALIZATION IN PERCEPTRONS

Osame Kinouchi
IFQSC, Universidade de São Paulo
Caixa Postal 369, 13560 São Carlos, SP, Brazil

Nestor Caticha
Instituto de Física, Universidade de São Paulo

Janeiro/1992

OPTIMAL GENERALIZATION IN PERCEPTRONS

Osame Kinouchi

IFQSC, Universidade de São Paulo,

CP 369, 13560 São Carlos, SP, Brazil

Nestor Caticha

Instituto de Física, Universidade de São Paulo,

CP 20516, 01498 São Paulo, SP, Brazil

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 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 situation with selection of examples are obtained and it is proved that, as expected, orthogonal selection is optimal.

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

In the statistical mechanics approach to learning from examples and generalization by neural nets [1–4], the single layer perceptron has been the prefered 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 mostly studied is that of learning a linearly separable boolean function

$$\mathcal{B}(\vec{S}) \equiv \sigma_B = \mathrm{sign}(\vec{B}\cdot\vec{S}) \tag{1}$$

where $\vec{S}$ is an input vector with N Ising components and $\vec{B}$ 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 $\vec{B}$, which can be taken to be normalized to one. The task of the "student perceptron" $\vec{J}$ 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 of input vector $\vec{S}_\mu$ and correct output σ_B^μ.

Two learning situations, 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 $\vec{S}$ appropriately chosen by the student during the learning process.

The quantity of interest is the generalization ability $G(\alpha)$, defined as the probability that a new random input $\vec{S}_\mu$, 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]

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

where ρ is the average overlap of the teacher and the student, $\rho = R/J$, $R = \vec{B}\cdot\vec{J}$ and $J = \sqrt{\vec{J}\cdot\vec{J}}$. 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 to misclassify 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 $\vec{J}_1 = 0$, the strategy of learning is a generalized Hebbian prescription [3]

$$\vec{J}_{\mu+1} = \vec{J}_\mu + \frac{1}{N} W_\mu \sigma_B^\mu \vec{S}_\mu \quad . \tag{3}$$

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

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

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

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

$$b_\mu = \vec{B}\cdot\vec{S}_\mu \quad \text{and} \quad h_\mu = \frac{\vec{J}_\mu\cdot\vec{S}_\mu}{J_\mu}$$

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

$$P(b_\mu, h_\mu) = P(h_\mu)\,P(b_\mu|h_\mu) =$$

$$= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{h_\mu^2}{2}\right) \cdot \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp\left(-\frac{(b_\mu - \rho h_\mu)^2}{2(1-\rho^2)}\right) \tag{6}$$

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

$$\rho_{\mu+1} = \rho_\mu + \frac{1}{NJ_\mu}\left[(b_\mu - \rho_\mu h_\mu)\,\sigma_B^\mu W_\mu - \frac{\rho_\mu W_\mu^2}{2J_\mu}\right] \quad . \tag{7}$$

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

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

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 $d\rho/d\alpha'$ 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

$$\kappa_\mu = \sigma_B^\mu b_\mu/\rho \quad \text{and} \quad \Delta_\mu = \sigma_B^\mu h_\mu \tag{10}$$

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 strategy 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 eq. (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 σ^μ_B

$$\bar{W}(\rho_\mu, J_\mu, \Delta_\mu) \;=\; \frac{\int d|b|\, P(b,h)\, W^*_\mu}{\int d|b|\, P(b,h)} \quad . \tag{11}$$

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 eq. (5) the weight function

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

is obtained, where

$$\lambda \;=\; \frac{\sqrt{1-\rho^2}}{\rho} \;=\; \mathrm{tg}(\pi\, e_g) \tag{13}$$

and

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

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

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

where Dh is the gaussian measure $(2\pi)^{-1/2}\, \mathrm{e}^{-h^2/2}\, dh$. 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 fig. 1 the resulting weight function is shown, together with the corresponding weight functions for 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 fig. 2 the theoretical prediction for the generalization ability is compared to a numerical simulation.

Up to this point only the "single presentation of examples" strategy has been discussed. It is possible however to extract further information from the same learning set by presenting again the examples already shown. We are not able to say whether the generalization gain per example will be maximized by using the weight function eq. (12) as before. We have only proved this optimal result for the first step of this sequential dynamics. Nevertheless, it seems quite natural to iterate the same algorithm. But now 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 for them are most appropriate for this problem. We have used the following recipe. First of all, the dependence of the performance on the step size, within some bounds, seems to be small. It only influences the convergence rate mildly. We have set the J parameter equal to one. 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 we are just judging the potential of the algorithm if ρ were known. After numerical convergence ρ 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]

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

which follows from a self consistent replica symmetric calculation. This suggests an approximation which actually consists in using the known Bayes value $\rho_B(\alpha)$ for $\rho(\alpha, n_{iter})$ in the weight function. The result of a numerical simulation is shown in fig. 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. Thus the measurement of e_l can be used in practice to decide when to stop the learning phase. The learning error has been found to be zero up to a value $\alpha_c \simeq .8$ and is smaller than 2×10^{-3} for any α.

Now the second learning situation 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

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

and the gain per example can be seen to be maximized if $P(h)$ is chosen to be a delta function centered at $h = 0$, $P(h) = \delta(h)$. That means that only examples orthogonal to $\vec{J}_\mu$, the accumulated knowledge, will be used during this learning

process. This justifies the heuristics of the selection criterion of Kinzel and Ruján, whom studied the case of selection 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 eq. (17) can be easily solved to yield

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

thus the weight function is $W = \sqrt{(2/\pi)}\, \exp\,(-\frac{\alpha}{\pi})$. Equation (19) shows that the selection of examples leads to expontential decrease of the generalization error, $e_g \simeq \frac{1}{\pi}\exp(-\alpha/\pi)$, whereas without selection of examples the error only decays algebraically as $e_g \simeq 0.44/\alpha$. Figure 4 shows results of a numerical simulation compared to the analytical prediction as well as the case where $W = 1$ [9]. Finite size effects account for the differences.

We now argue that the weight function W can be thought of as a measure of the "value of information" of a given example, for this particular problem. Althought 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 $\vec{B}$ and $\vec{J}$. 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 a 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 procedure has been presented which aims to maximize the generalization ability. The first step of the learning dynamical process has been studied analytically and numerical results of its 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 by Meir and Fontanari [14] where a relaxation algorithm with an α dependent κ parameter was studied. It seems, at least numerically, to also saturate the Bayes bound. Their choice of an optimal $\kappa(\alpha)$ in the relaxation algorithm leads to a weight function which approximates $\bar{W}(\rho, J, \Delta)$ of eq. (12), at least in the region where h_μ is close to zero.

Acknowledgment

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

References

[1] Denker J., Schwartz D., Wittner B., Solla S., Howard R., Jackel L. and Hopfield J.J., "Large Automatic Learning, Rule Extraction and Generalization", *Complex Systems* **1**, 877 (1987).

[2] Levin E., Tishby N. and Solla S.A., "A Statistical Approach to Learning and Generalization in Layered Natural Networks", in *Proceedings of Sec. Annual Workshop on Computational Learning Theory*, COLT–89, Eds. R. Rivest, D. Haussler and M.K. Warmuth (Morgan Kaufmann, San Mateo Ca, 1989).

[3] Abbott L. F. , "Learning in Neural Network Memories", *Network* **1** , 105 (1990).

[4] Györgyi G. and Tishby N., "Statistical Theory of Learning a Rule", in *Proceedings of STATPHYS 17 Workshop on Neural Nets and Spin Glasses*, Eds. W. Theumann and R. Köberle (World Scientific, 1989).

[5] Seung H. S., Sompolinsky H. and Tishby N. , "Statistical Mechanics of Learning from Examples", Preprint , 1991

[6] Vallet F., "The Hebb Rule for learning linearly separable Boolean Functions: learning and generalization" *Euro. Phys. Letts.* **8** , 747 (1989) Vallet F. and Cailton J.-G., "Recognition Rates of the Hebb Rule for Learning Boolean Functions", *Phys. Rev.* **A41**, 3059 (1990).

[7] Opper M., Kinzel W., Kleinz J. and Nehl R., "On the Ability of the Optimal Perceptron to Generalise", *J. Phys. A: Math. Gen.* **23**, L581 (1990).

[8] Opper M. and Haussler D., "Generalization Performance of Bayes Optimal Classification Algorithm for Learning a Perceptron", *Phys. Rev. Lett.* **66**, 2677 (1991).

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

[10] Rau A. and Sherrington D., Preprint (1991).

[11] Kinouche O. and Caticha N., " Biased learning in Boolean perceptrons", Preprint (1991) submitted to Physica A

[12] Valiant G.L., "A Theory of the Learnable", *Comm. ACM* **27**, 1134 (1984).

[13] Brillouin L., " Science and Information Theory", 2^{nd} ed., Academic Press Inc. New York (1971); Volkenshtein M. V. " Information Theory and Evolution", in "Cybernetics of Living Matter", Makarov I. M., Ed., Mir Publishers Moscow (1987)

[14] Meir R. and Fontanari J.F., "On the Calculation of Learning Curves for Inconsistent Algorithms", Preprint (1991).

Figure Captions

[Fig.1] Examples of weight functions of the "expected stability" (solid), perceptron (squares-dotted), adaline (dashed) and the relaxation (circles-dashed) algorithms for fixed α.

[Fig.2] $G(\alpha)$: the solid line is obtained from a numerical integration of equation (15) . The squares are the result of a simulation with $N = 199$ averaged over 200 runs. The lower curve is the pure Hebb [6].

[Fig.3] $\rho(\alpha)$ of the iterated expected stability algorithm (circles) average over 20 runs $N = 149$ after 50 iterations and the Bayes algorithm $\rho_B(\alpha)$ (solid line)

[Fig.4] Convergence of the Iterated Expected Stability algorithm: $\Delta e_g = e_g(n_{iter}) - e_g(n_{iter} - 1)$ (circles) and Δe_l (squares) converge to 0 at approximately the same rate for fixed α (dot : $\alpha = 0.6$, dash : $\alpha = 2.5$ and solid : $\alpha = 10$)

[Fig.5] Selection of examples. Expected stability, circles $N = 149$, squares $N = 249$ both averaged over 30 runs, theoretical value from eq. [18] is the solid line, and dashed line for Hebbian $(W = 1)$ as in ref. [9]

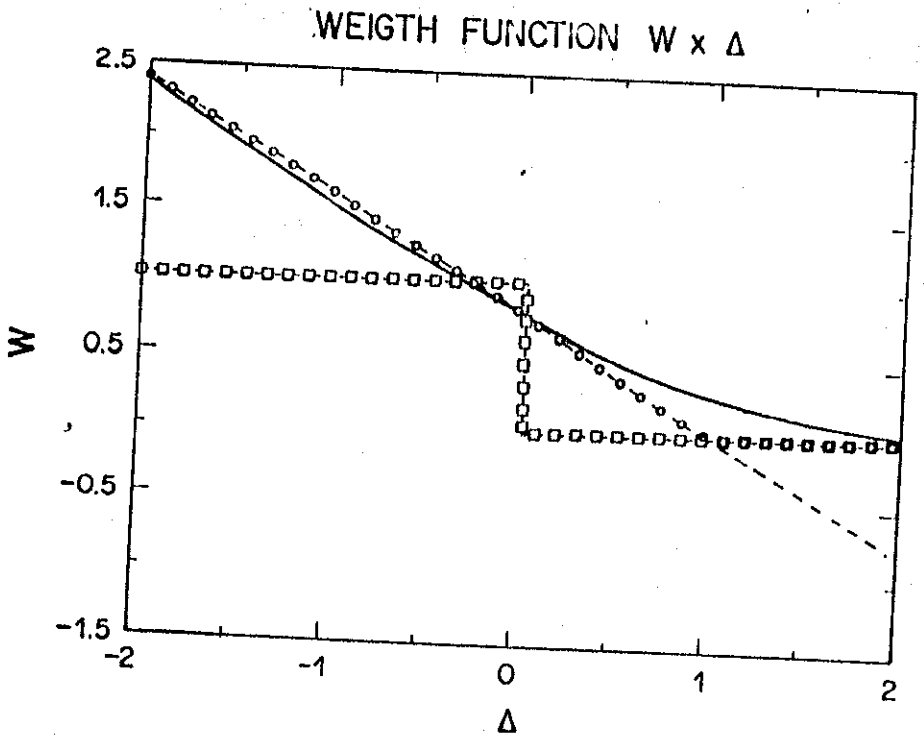
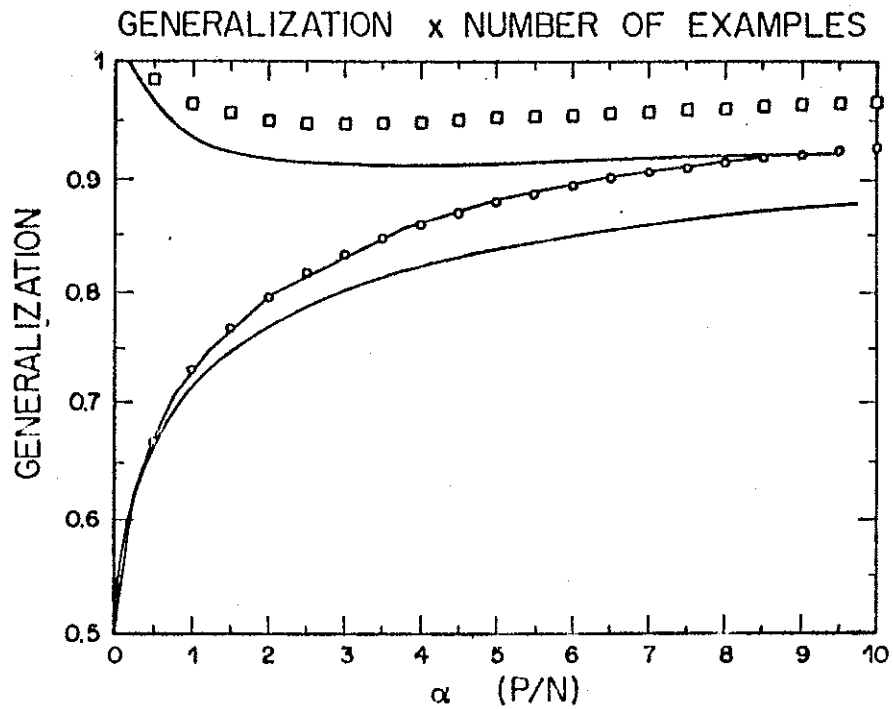


fig 1 Kinouchi & Caticha



Kinouchi & Caticha Fig 2

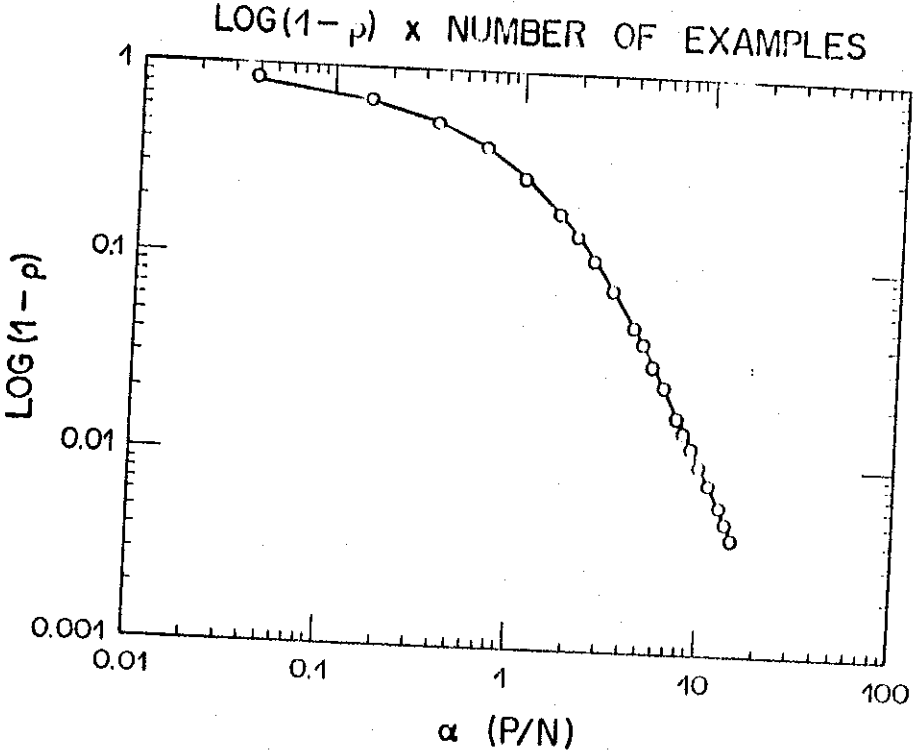


fig 3 Kinouchi & Caticha

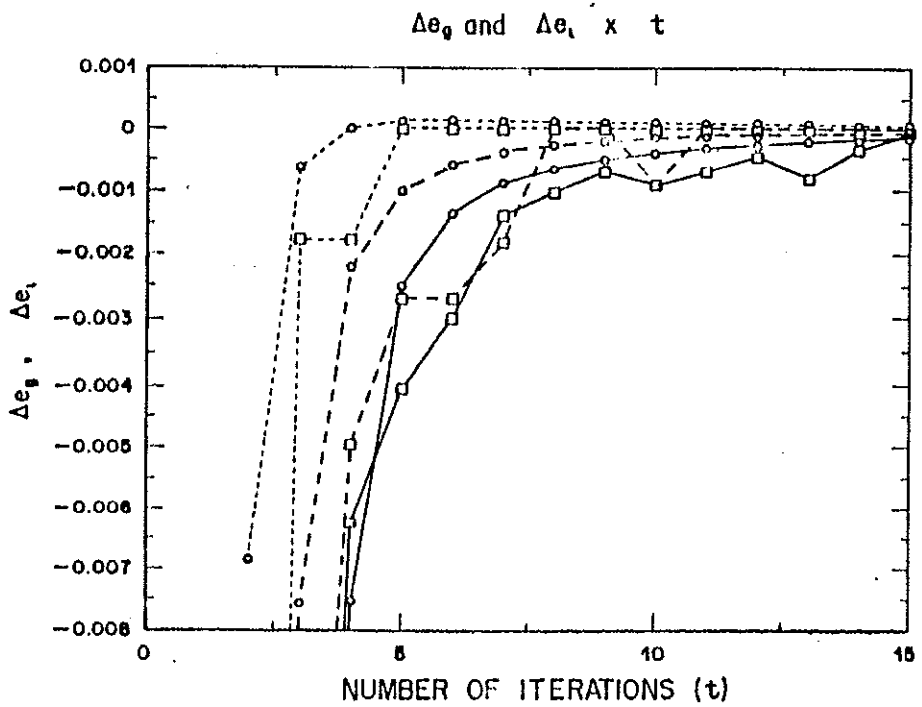


Fig 4

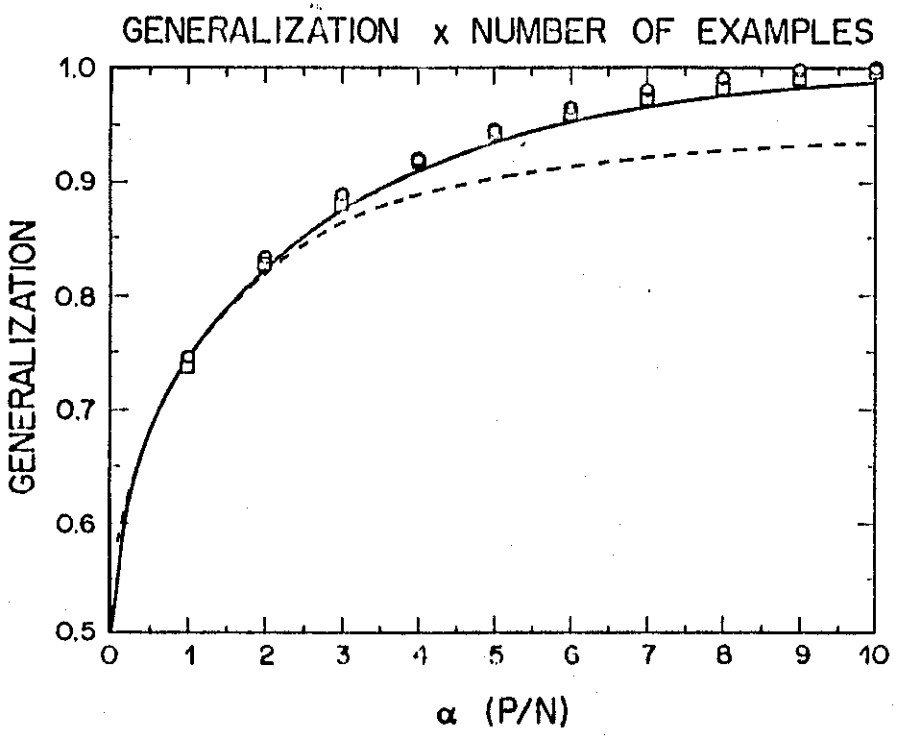

GENERALIZATION x NUMBER OF EXAMPLES
GENERALIZATION
1.0
0.9
0.8
0.7
0.6
0.5
0
1
2
3
4
5
6
7
8
9
10
α (P/N)

fig 5. [illegible]