Learning dynamics of simple perceptrons with non-extensive cost functions

S A Cannas†§, D Stariolo‡ and F A Tamarit†‡

† Facultad de Matemática, Astronomía y Física, Universidad Nacional de Córdoba, Haya de la Torre y Medina Allende S/N, Ciudad Universitaria, 5000 Córdoba, Argentina

‡ Centro Brasileiro de Pesquisas Físicas, Rua Xavier Sigaud 150, 22290-180, Rio de Janeiro, Brazil

Abstract. A Tsallis-statistics-based generalization of the gradient descent dynamics (using non-extensive cost functions), recently introduced by one of us, is proposed as a learning rule in a simple perceptron. The resulting Langevin equations are solved numerically for different values of an index q ($q = 1$ and $q \neq 1$ respectively correspond to the extensive and non-extensive cases) and for different cost functions. The results are compared with the learning curve (mean error versus time) obtained from a learning experiment carried out with human beings, showing an excellent agreement for values of q slightly above unity. This fact illustrates the possible importance of including some degree of non-locality (non-extensivity) in computational learning procedures, whenever one wants to mimic human behaviour.

1. Introduction

Learning from examples is one of the topics of greatest current interest in the field of neural networks [1]. Learning procedures (or *learning rules*) are, in general, synaptic modification algorithms that allow an arbitrarily connected network to develop an internal structure appropriate for a particular task. This goal can be achieved on the basis of direct comparison of the output of the network with known correct answers (*examples*). The synaptic couplings are then modified in order to reproduce the examples as well as possible. This is sometimes called *supervised learning*. Learning rules can also be interpreted as a dynamical search of global minima of some *ad hoc* introduced *cost function*, through the space of synaptic couplings.

The constraints to be imposed over the cost function are, in general, very weak, allowing enormous freedom of choice. One of the most widely used constraints is that the cost function should induce a *local* learning rule. This means that the variation of the synapse between two neurons at a given time should depend only on the instantaneous post-synaptic potentials (PSP) received by them, and not on the PSP received by the rest of the neurons. Such a requirement has a heuristic character and, although quite plausible from a biological point of view, it is not supported by concrete empirical evidence. Therefore, it is of interest to investigate the effects of introducing *non-local* learning rules in a neural network.

In this paper we analyse the dynamical effects of some kinds of cost functions that induce non-local rules, in a simple perceptron. The associated dynamics is a Tsallis-statistics-based

§ Member of the National Research Council (CONICET–Argentina).

generalization of the gradient descent dynamics, recently introduced by one of us [2]. The effects of different choices of the cost function, which generate both local and non-local rules, are analysed in the problem of '*memorization*', i.e. the perceptron learns a single random pattern, uncorrelated with p previously learnt ones. The results are compared with the *learning curve* (mean error versus time) obtained from a learning experiment carried out with human beings [3]. In section 2 we briefly describe the experiment. In section 3 we analyse the proposed learning dynamics. In section 4 we present the numerical results and compare them with the experiment; in section 5 we draw our conclusions.

2. The experiment

The 'memorization' experiment consists of a series of steps, during which the individual (the subject of the experiment) has to 'learn' a visual pattern. The pattern is a 5 × 5 grid (or checkerboard) filled by circles and crosses (randomly chosen once for ever).

At every step the grid is shown to the individual for a period of eight seconds and then hidden. Then, the individual is asked to reproduce the pattern in an empty grid; when he (she) finishes, the reproduced picture is removed and the individual is left to rest for ten seconds before a new step starts. The procedure is repeated until the visual pattern is reproduced *exactly*, i.e. the Hamming distance (number of squares in which the patterns are different) $H = 0$ in two successive steps, or after 10 steps even if $H \neq 0$, in order to avoid fatigue effects (see [3] for details). By plottting the Hamming distance H versus the number of times (steps) the picture has been shown, a *learning curve* is obtained for every individual.

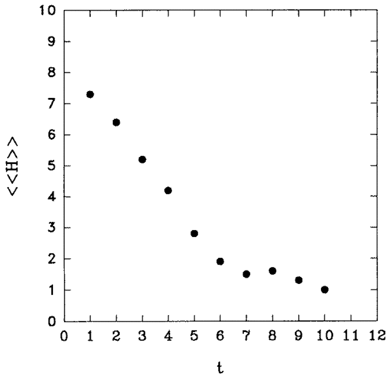

Figure 1. Mean Hamming distance between the pattern presented to and the pattern reproduced by the individuals versus the number of times the picture has been shown.

The experiment was performed on a 'human sample' of 92 individuals composed of students of humanistic disciplines at the Federal University of Rio de Janeiro (Brazil). The mean learning curve (i.e. the mean value of H computed over the sample at every step) versus the number of steps t is shown in figure 1.

3. The learning dynamics

In order to mimic the human behaviour in the memorization experiment, with learning in a neural network, we consider a simple perceptron [4], composed of an input layer of N *binary* neurons $S_i = \pm 1$, and an output layer of N *analogue* neurons (real variables)

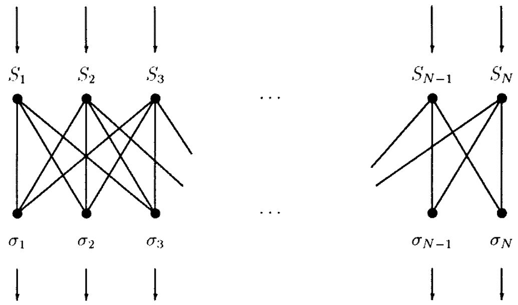

Figure 2. Simple perceptron with equal number N of input and output neurons. Every output neuron can, in principle, be connected to all the input neurons.

$\sigma_i \epsilon [-1, 1]$ (see figure 2). The activation law for the output neurons is given by

$$\sigma_i \left[\{S_j\}\right] = \tanh\left[\frac{g}{\sqrt{N}} \sum_{j=1}^{N} J_{ij} S_j\right] \tag{1}$$

where the *gain* $g > 0$ is an arbitrary real number and $\{J_{ij}\}$ are real-valued *synaptic efficacies*, whose values are restricted by the normalization [5]

$$\sum_{i=1}^{N} J_{ij}^2 = N\,. \tag{2}$$

Let us start by considering the simplest case of learning a single binary pattern $\{\xi_j\}$, with $j = 1, 2, \dots, N$, where $\xi_j = \pm 1$ are independent random variables with $\langle \xi_j \rangle = 0$. Starting from a random initial configuration of the synaptic couplings (subject to the constraint (2)) we look for a stochastic dynamics, such that these couplings evolve to a final configuration in which the network stores the input pattern associatively. In other words, it maps the pattern $\{\xi_j\}$, as well as any other state that is sufficiently close, into an analogue pattern $\{\sigma_j \approx \xi_j\}$, $j = 1, 2, \dots, N$, which is as similar as possible to $\{\xi_j\}$ within the present constraints. This task can be carried out by different dynamics, the most widely used being the *gradient descent* method, ruled by the following set of Langevin equations:

$$\frac{\mathrm{d}J_{ij}}{\mathrm{d}t} = -\frac{\partial V}{\partial J_{ij}} + \eta_{ij}(t) \tag{3}$$

where η_{ij} is white noise with $\langle \eta_{ij}(t) \rangle = 0$ and $\langle \eta_{ij}(t)\eta_{i'j'}(t')\rangle = 2T\delta_{ii'}\delta_{jj'}\delta(t-t')$. The cost function V is some measure of the deviation of the output of the network $\sigma_j(\{\xi_j\})$ from the desired output $\{\xi_j\}$. The cost function should be minimal whenever the two agree. The usual choices of V are *extensive* functions of the type $V = \sum_j V_j(J_{ij})$ where the sum runs over the output neurons, and V_j depends only on the synapses associated with the output neuron j.

This kind of dynamics generates a *local* learning rule, i.e. the updating of the coupling J_{ij} depends only on the local field at the output neuron j:

$$\frac{\mathrm{d}J_{ij}}{\mathrm{d}t} = -\frac{\partial V_j}{\partial J_{ij}} + \eta_{ij}(t)\,. \tag{4}$$

Hence one can work with one single output neuron.

In this work we propose a generalization of this method in which the cost function V in (3) is replaced by a *non-extensive* function $\overline{V}$ defined by the map [2]

$$\overline{V} = \frac{1}{\beta(q-1)} \ln\left[1 + \beta(q-1)V\right] \tag{5}$$

where $\beta \equiv 1/T$ and the index q is an arbitrary real number such that $q \geqslant 1$. Consequently, equation (3) is replaced by

$$\frac{\mathrm{d}J_{ij}}{\mathrm{d}t} = -\frac{1}{1+\beta(q-1)V}\frac{\partial V}{\partial J_{ij}} + \eta_{ij}(t)\,. \tag{6}$$

Note that this new dynamics induces *non-local* learning rules (compare equations (4) and (6)). Consequently, one has to consider the full set of output neurons in the updating of every coupling, as can be infered from the non-linear structure of (6).

This dynamics leads, for long times, to a generalized equilibrium probability distribution for the couplings J_{ij} of the form [2]:

$$p(\{J_{ij}\}) = \frac{\left[1 - \beta(1-q)V\right]^{1/(1-q)}}{Z_q} \tag{7}$$

with

$$Z_q = \int \mathrm{d}\mu(\{J_{ij}\})\ \left[1 - \beta(1-q)V\right]^{1/(1-q)} \tag{8}$$

where $\mathrm{d}\mu(\{J_{ij}\})$ is a normalized measure in the coupling space that takes into account the constraint (2). The probability distribution (7) can be derived by optimizing the Tsallis entropy [6]:

$$S_q\left[p(\{J_{ij}\})\right] = \frac{1}{q-1}\left\{1 - \int \mathrm{d}\mu(\{J_{ij}\})\ \left[p(\{J_{ij}\})\right]^q\right\} \tag{9}$$

with the constraint [7]

$$\langle V\rangle_q \equiv \int \mathrm{d}\mu(\{J_{ij}\})\ \left[p(\{J_{ij}\})\right]^q V(\{J_{ij}\}) = \text{constant}\,. \tag{10}$$

Probability distributions derived from this entropy have been applied recently to generate very efficient optimization algorithms [8]. Physical applications of this entropy formalism can be found in [9, 10].

In the limit $q = 1$ the standard gradient descent equation (3) is recovered from (6) and the equilibrium distribution is the canonical Boltzmann–Gibbs one [5].

We now introduce the *quadratic error function*

$$\varepsilon \equiv \frac{1}{4N}\sum_{j=1}^{N}\left(\sigma_j(\boldsymbol{\xi}) - \xi_j\right)^2 \tag{11}$$

$$= \frac{1}{4N}\sum_{j=1}^{N}\left[1 + \tanh^2\left(\frac{g}{\sqrt{N}}\boldsymbol{J}_j\cdot\boldsymbol{\xi}\right) - 2\ \tanh\left(\frac{g}{\sqrt{N}}\boldsymbol{J}_j\cdot\boldsymbol{\xi}\,\xi_j\right)\right] \tag{12}$$

with

$$\boldsymbol{J}_j \equiv \begin{pmatrix} J_{1j} \\ J_{2j} \\ \vdots \\ J_{Nj} \end{pmatrix} \qquad \boldsymbol{\xi} \equiv \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_N \end{pmatrix}. \tag{13}$$

In the limit $g \to \infty$, equation (1) reduces to

$$\sigma_j(\boldsymbol{\xi}) = \operatorname{sgn}\left[\frac{g}{\sqrt{N}} \boldsymbol{J}_j \cdot \boldsymbol{\xi}\right]$$

and equation (12) gives the Hamming distance between the input and output patterns, $\boldsymbol{\xi}$ and $\boldsymbol{\sigma}(\boldsymbol{\xi})$, respectively.

In this paper we consider two different choices for the cost function V:

(a) $V = N\varepsilon$ [5]

(b) $V = \sum_j \left(\kappa - \lambda_j\right)^2 \Theta\left(\kappa - \lambda_j\right)$

where κ is a positive constant, the *stability parameters* λ_j are defined as $\lambda_j \equiv \xi_j \boldsymbol{J}_j \cdot \boldsymbol{\xi}/\sqrt{N}$ [11] and $\Theta(x)$ is the Heaviside function. The error function (12) can be expressed as

$$\varepsilon = \frac{1}{4N} \sum_{j=1}^{N} \left[1 + \tanh^2\left(g\lambda_j\right) - 2\tanh\left(g\lambda_j\right)\right] . \tag{14}$$

Our aim is to calculate the time evolution of $\langle\langle\varepsilon\rangle\rangle$, where $\langle\langle\cdots\rangle\rangle$ denotes a double average over the initial conditions and the realizations of the noise. Note that, for both definitions, the cost function depends on the J_{ij} through the parameters λ_j, whose time evolution is obtained from (6) as follows:

$$\frac{\mathrm{d}\lambda_j}{\mathrm{d}t} = -\frac{1}{1+\beta(q-1)V}\frac{\partial V_j}{\partial \lambda_j} + \sqrt{T}\eta'_j(t) \tag{15}$$

with

(a) $V_j(\lambda_j) = \frac{1}{4N}\left[1 + \tanh^2\left(g\lambda_j\right) - 2\tanh\left(g\lambda_j\right)\right]$

(b) $V_j(\lambda_j) = \left(\kappa - \lambda_j\right)^2 \Theta\left(\kappa - \lambda_j\right)$

where

$$\eta'_j(t) \equiv \frac{1}{\sqrt{TN}} \sum_i \xi_i \eta_{ij}(t)$$

is white noise with $\langle\eta'_j(t)\rangle = 0$ and $\langle\eta'_j(t)\eta'_{j'}(t')\rangle = 2\delta_{jj'}\delta(t-t')$.

Starting from different initial configurations for λ_j we calculate $\langle\langle\varepsilon\rangle\rangle$ by solving equation (15) numerically. Since the initial values of the $\boldsymbol{J}_j$ are chosen from a uniform distribution in the N-dimensional hypersphere of radius $\sqrt{N}$, it is easy to see that the initial values of the λ_j's follow a Gaussian distribution, with mean value 0 and variance 1.

Before we present our results, let us briefly discuss the problem of learning one single pattern, once the network has already learnt p previous ones. In this case, for any pattern μ one must introduce a set of stability parameters $\{\lambda_j^\mu\}$, with $j = 1, \ldots, N$ and $\mu = 1, \ldots, p+1$. Instead of the set of N equations given by (15) we will have a set of $(p+1)\ N$ coupled Langevin equations. However, for uncorrelated patterns, it can be shown that for $N \gg p$ these equations decouple and the stability parameter for each pattern evolves independently of the others. So, in such a limit we recover equation (15).

4. Results

In this section we present the numerical calculations of $\langle\langle\varepsilon\rangle\rangle$ for different values of the parameters q, T and g (we verified that varying κ does not introduce qualitative new effects, so we kept it fixed at $\kappa = 1$) and for both definitions of the cost function introduced in the previous section. The Langevin equations are solved by standard methods [12] for $N = 25$ (the number of bits corresponding to the 5×5 checkerboard referred to previously).

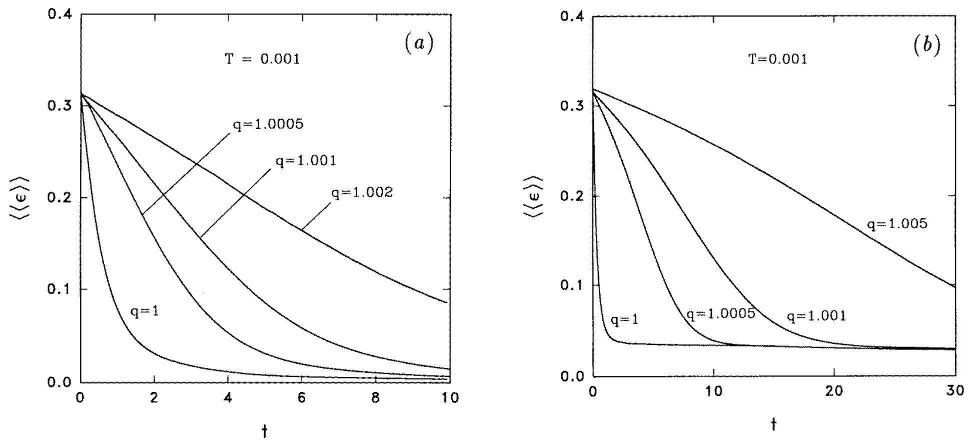

Figure 3. Mean error $\langle\langle\varepsilon\rangle\rangle$ versus time t (arbitrary units), for typical values of q and T, ($g = 0.69$ and $N = 25$). (a) Cost function $V = N\varepsilon$. (b) Cost function $V = (\kappa - \lambda)^2\,\Theta(\kappa - \lambda)$ with $\kappa = 1$.

In figure 3 we present the learning curves $\langle\langle\varepsilon\rangle\rangle$ versus time t for typical values of q at low T (g and κ fixed), for both choices of the cost function. For long times they both display for all values of q an exponential decay. For short times, the qualitative behaviour changes drastically in both cases when q departs from 1, showing a slow decay for $q > 1$. Moreover, for $q = 1$, the learning curves are convex functions (positive curvature) for all t while, for $q > 1$, they are concave at short t, changing their curvature at intermediate times. The last behaviour can be observed in figure 1. We finally observe that learning is slower when q increases above unity. These effects can be easily understood by looking at equation (6). For short times the mean value of the cost function V is relatively high, and the non-local factor $\left[1 + \beta(q-1)V\right]^{-1}$ diminishes (for $q \neq 1$) the driven effect of the gradient term. As the system evolves $V \to 0$ and $1 + \beta(q-1)V \to 1$; therefore, for long times, the dynamics becomes the gradient descent one and $\langle\langle\varepsilon\rangle\rangle$ presents the $q = 1$ exponential decay. Note that this property is quite general for this kind of dynamics and will also be present in multilayer neural networks.

We now try to fit the experimental results with the learning curves obtained with our model. Since the microscopic time scale of the experiment is not accesible, we have to rescale the time appropriately both for the experimental and perceptron data, in order to make them comparable. We define, for every learning curve, a characteristic time t_{m}, as the time for which the mean error decays to half of its maximum value, i.e.

$$\langle\langle\varepsilon\rangle\rangle(t_{\mathrm{m}}) = \tfrac{1}{2}\langle\langle\varepsilon\rangle\rangle(0)$$

and we use t_{m} as the time unit. The value of $\langle\langle\varepsilon\rangle\rangle(0)$ for the experimental curve is estimated by a quadratic extrapolation of the first data points.

Next, we note that the value of $\langle\langle\varepsilon\rangle\rangle(0)$ for the theoretical curves is independent of the choice of the cost function and of q, since it is always a Gaussian average of (14). Hence, it only depends on the gain parameter g, which can be fitted in order to reproduce the experimental result. A numerical calculation yields the value $g = 0.69$ (to reproduce $\langle\langle\varepsilon\rangle\rangle(0) = 0.316$).

Finally, the values of T can be bounded by noting that the learning curves decay monotonically with t. Hence, the minimum value of the experimental curve can be taken as an upper bound for the asymptotic value of $\langle\langle\varepsilon\rangle\rangle$ at $t \to \infty$. This value can be calculated numerically as a function of T, from the equilibrium distribution (7). Since the equlibrium value of $\langle\langle\varepsilon\rangle\rangle$ is an increasing function of T, we obtain the upper bound $T < 0.01$ (to guarantee $\langle\langle\varepsilon\rangle\rangle(\infty) < 0.04$).

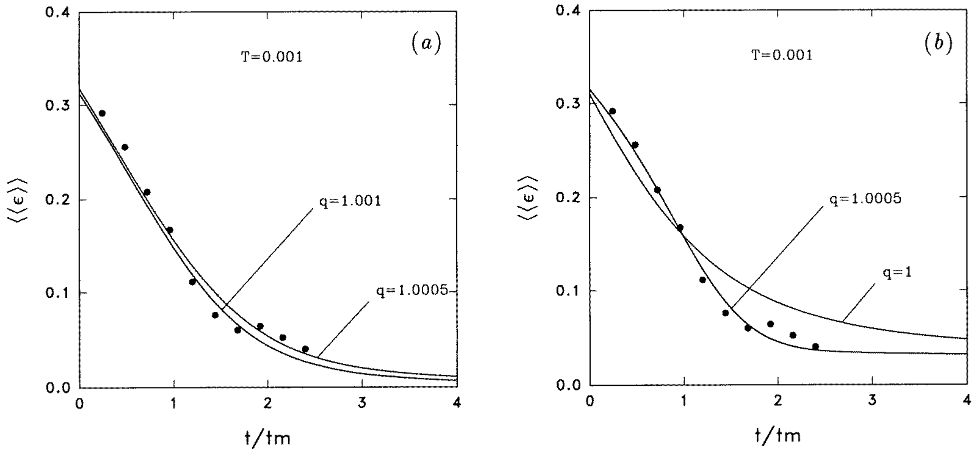

Figure 4. Mean error (normalized Hamming distance) versus rescaled time t_{m}. Full circles correspond to the learning experiment with human beings (see figure 1); full curves correspond to the perceptron model ($g = 0.69$). (a) Cost function $V = N\varepsilon$. (b) Cost function $V = (\kappa - \lambda)^2\,\Theta(\kappa - \lambda)$ with $\kappa = 1$.

In figure 4 we compare the rescaled experimental data with the theoretical learning curves $\langle\langle\varepsilon\rangle\rangle$ versus t/t_{m} for different values of q and T, for both choices of the cost function. The best fitting is obtained for the cost function (b) with $T = 0.001$ and $q = 1.0005$ ($\kappa = 1$). It is worth noting that, while for the cost function (a) the rescaled learning functions vary appreciably with q (at least for q near to one), for the cost function (b) the rescaled curves vary very little with q, for $1.005 < q < 1.01$.

5. Conclusions

We have analysed the time evolution of the mean error in the supervised learning dynamics of a perceptron, for the particular task of memorizing a single pattern uncorrelated with p previously learnt ones (with $p \ll N$). Our results suggest that, at short times, the dynamics induced by an extensive cost function (local learning rules) can be very different from that induced by a non-extensive cost function (non-local learning rules).

The agreement with the experimental results is quite impressive (at least at a phenomenological level), especially if we consider that they are reproduced with an extremely simplified model (a simple perceptron with one single pattern), far removed from a complex system such as the human brain. This fact suggests that some aspects of the learning dynamics in biological systems could be independent of the detailed microscopic structure of the neural network, depending only on some overall characteristics. In this sense, some kind of universality could exist in these processes, like that appearing in critical phenomena, where the asymptotic behaviour of most relevant variables can essentially be determined by a few macroscopic parameters [13]. Moreover, our results suggest that the non-locality of the learning rules (and therefore equilibrium statistics *other* than the usual Boltzmann–Gibbs one) could be one such universal aspect, and perhaps a very important one in understanding learning processing in real biological systems. By the way, let us note that the influence of the holistic (i.e. context-dependent [7]) nature of the dynamics used herein in the learning process is consistent with the information interpretation of the entropy (9) in the context of statistical inference [14]. This is, of course, only one possible non-local learning model of many, such as multilayer perceptrons. However, using a non-local dynamics in a simple perceptron has the computational advantage that we have to solve only N coupled stochastic differential equations (equations (15)) for the stability parameters $\{\lambda_j\}$, instead of the set of N^2 equations for the synapses $\{J_{ij}\}$ (equations (6)). On the other hand, using a noisy local dynamics in a multilayer perceptron with only one hidden layer for the present problem implies solving a set of $\mathrm{O}(N^4)$ coupled stochastic differential equations for the synapses $\{J_{ij}\}$, since they cannot be reduced as before.

We believe the results obtained with this simple non-local model are sufficiently encouraging in this direction for it to be interesting to try studying the much more difficult problem of a multilayer perceptron with a noisy local dynamics and see whether it can lead to similar learning curves. Moreover, it would also be interesting to compare these results with those due to other effects, such as the superposition of different learning rates.

It is worth noting that the experimental learning curves are best fitted by a $q > 1$ model instead of the more efficient $q = 1$ one (see figure 3). Note, however, that after a transient period the learning curves decay exponentially, even for $q > 1$. In other words, the biological solution to the memorization problem seems not to be the best one, compared with the solutions that can be obtained by an artificial (externally designed) mechanism. It is a known fact that, owing to its evolutionary origin, a biological brain does not necessarily find the best solution, but only a good one, for a given problem (very interesting discussions about the consequences of the evolutionary nature of biological brains can be found in [15]). In fact, a similar phenomenon has been observed in a comparison between an experiment of human learning, an algorithm of symbolic learning (ID3) and neural network learning (perceptron) (see [16] and references therein).

Acknowledgments

Fruitful discussions with, as well as many suggestions by Constantino Tsallis are acknowledged. We are also indebted to A C Tsallis, A B Lima and A C N Magalhães for making their experimental data available to us prior to publication. This work was partially supported by grants B-11487/2B9210 from Fundação Vitae (Brazil), PID 3235/94 from Consejo Provincial de Investigaciones Científicas y Tecnológicas (Córdoba, Argentina), PID 3592 from Consejo Nacional de Investigaciones Científicas y Tecnológicas–CONICET (Argentina) and a grant from Secretaría de Ciencia y Tecnología de la Universidad Nacional de Córdoba (Argentina).

References

[1] Watkin T L H, Rau A and Biehl M 1993 *Rev. Mod. Phys.* **65** 499–556

[2] Stariolo D A 1994 *Phys. Lett.* **185A** 262–4

[3] Tsallis A C, Lima A B, Tsallis C, Magalhães A C N and Tamarit F A 1996 to be published

[4] Müller B and Reinhardt J 1991 *Neural Networks: An Introduction* (Berlin: Springer)

[5] Seung H S, Sompolinsky H and Tishby N 1992 *Phys. Rev.* A **45** 6056–90

[6] Tsallis C 1988 *J. Stat. Phys.* **52** 479

[7] Curado E M F and Tsallis C, *J. Phys. A: Math. Gen.* **24** L69 (Corrigenda 1991 *J. Phys. A: Math. Gen.* **24** 3187; 1992 *J. Phys. A: Math. Gen.* **25** 1019)

[8] Tsallis C and Stariolo D A *Preprint* Generalized simulated annealing

[9] Plastino A R and Plastino A 1993 *Phys. Lett.* **174A** 384
Aly J J 1993 *Proc. Meeting (Aussois, France, 21–25 March 1993)* ed F Combes and E Athanassoula (Paris: Publications de l'Observatoire de Paris) pp 19–23

[10] Alemany P A and Zanette D H 1994 *Phys. Rev.* E **49** 956

[11] Horner H 1992 *Z. Phys.* B **86** 291–308

[12] Risken H 1984 *The Fokker–Planck Equation* (Berlin: Springer)

[13] Binney J J, Dowrick N J, Fisher A J and Newman M E J 1993 *The Theory of Critical Phenomena* (Oxford: Oxford Science)

[14] Tsallis C, Deutscher G and Maynard R 1994 *Preprint*
de Souza A M C and Tsallis C 1994 *Preprint*

[15] Barlow H 1994 *Biology and Computation: A Physicist's Choice* (Advance Series in Neuroscience 3) ed H Guttfreund and G Toulouse (Singapore: World Scientific) pp 5–14
Jacob F 1994 *Biology and Computation: A Physicist's Choice* (Advance Series in Neuroscience 3) ed H Guttfreund and G Touluse (Singapore: World Scientific) pp 72–7

[16] Bernasconi J and Gustafson K 1994 *Network* **5** 203–27