Pseudoinverse techniques, information theory, and the training of feedforward networks

L. Diambra,* J. Fernández,† and A. Plastino‡
Departamento de Física, Universidad Nacional de La Plata, Casilla de Correo 67, 1900 La Plata, Argentina

(Received 2 March 1995)

An information theory based method for the training of perceptrons is presented. Our technique guarantees an errorless learning process for learnable mappings with just a minimum amount of examples. The only requirement is that the transfer function must possess an inverse. Some illustrative results are presented. The method can be considered to yield another tool for feedforward training.

PACS number(s): 87.10.+e, 89.70.+c, 05.20.−y, 02.70.−c

I. INTRODUCTION

Over the past years a great deal of effort has been invested in the development of training algorithms for feedforward neural networks [1,2]. Neural networks have exhibited remarkable properties for the storage of patterns and for data processing, having found use in a wide variety of environments. Of particular interest is the application of statistical mechanics techniques in the analysis of the process of learning a rule (on the basis of selected examples), the case of a student perceptron trained by a teacher perceptron having been studied in great detail. The associated learning curves have been calculated on the basis of several (distinct) training schemes [3–5].

Most trained networks are able to *predict*, i.e., to produce outputs corresponding to *new* inputs (that are not included in the training set) on the basis of an adequately selected working *hypothesis*. This hypothesis is, of course, represented by a set of synaptic weights W_i that, when appropriately implemented, yields good results for the examples of the training set. Much effort has consequently been devoted to the task of developing suitable training algorithms that are able to adjust the synaptic weights so as to enable the network to *infer* the correct answer when presented with a new input. Of course, one wishes for algorithms that accomplish such a goal within a reasonable (CPU) time and with a not too large number of examples. The most popular learning methods involve minimization of an energy (or cost) function that depends upon the set of training patterns. Diverse approaches to this end include simulated annealing [6], genetic algorithms [7], and gradient methods [1,8,9]. A cost function is minimized by recourse to an algorithm that incorporates a degree of randomness, as represented by a "temperature" or by "mutations." In order to improve upon the learning process, diverse energy forms have been proposed [10].

In the present effort we also wish to introduce improvements upon the learning process. However, we shall concentrate our efforts on the *selection of the working hypothesis.* This is to be accomplished according to Ockham's razor, i.e., with the miminum number of assumptions compatible with the available input. Our tools will be those pertaining to the information theory (IT) approach to statistical mechanics as embedded in the maximum entropy principle [11–13]. We shall take advantage of a rather recent advance in this field: the pseudoinverse technique [14]. A learning protocol will be developed in this fashion and applied to the simplest layered network: the *perceptron.* The paper is organized as follows. A brief review of basic IT concepts is given in Sec. II. The present formalism is developed in Sec. III and illustrated with reference to some simple examples in Sec. IV. Our results are discussed in Sec. V.

II. BRIEF REVIEW OF ELEMENTARY INFORMATION THEORY CONCEPTS

A. Generalities

For the sake of completeness we briefly review here some elementary IT concepts. The reader acquainted with them is advised to skip this section. Information theory dates from the pioneering work of Shannon *et al.* [11,15] and is by now an established branch of mathematics, with multiple applications in most areas of scientific endeavor. Its main purpose is that of providing one with the best possible inference method, that is, the one that uses all available data while explicitly avoiding the introduction of any unnecessary hypothesis. In this sense, it can be asserted that it is philosophically founded on Ockham's razor, to which it gives an explicit implementation algorithm. It is convenient to be aware of the *inductive* character of any process of gaining knowledge by recourse to IT tools. One is always proceeding from particular instances (in most cases, specific pieces of data) toward more general recipes encompassing a variety of situations. This inductive aspect of IT procedures is to

*Electronic address: diambra@venus.fisica.unlp.edu.ar
†Electronic address: frenande@venus.fisica.unlp.edu.ar
‡Electronic address: plastino@venus.fisica.unlp.edu.ar

©1995 The American Physical Society

be emphasized in any physical discussion because a substantial portion of the concomitant theoretical research endeavor is of a *deductive* character and one proceeds thereby from a few principles (i.e., Maxwell's equations) to develop involved applications to specific (and mostly complex) environments. The essential IT idea is that of quantifying our ignorance in a given situation in such a way that one can afterward measure it and formally deal with it in mathematical fashion [15]. In applications to the physical sciences there is an intimate connection between IT and probabilistic environments, the essential IT result being that a definite amount of ignorance is to be associated to any given probability distribution $\{p_i\}$. This ignorance is measured by Shannon's entropy [11]

$$S = -\sum_i p_i \ln [p_i] , \tag{1}$$

where, if logarithms are expressed in base 2, *S is given in bits.* From a historical point of view, the first application of IT ideas to physics consisted in the elegant reformulation of statistical mechanics achieved by Jaynes [12].

B. IT implementation of Ockham's razor

We now describe the orthodox IT algorithm [15–19]. Visualize the following scenario. We are dealing with a system X with ν internal states labeled by an index i $(i = 1, ..., \nu)$. X can be found in the state k with probability p_k. Let A_α $(\alpha = 1, ..., N)$ be a set of random variables that characterize the system. Of course, these variables adopt specific (and known) numerical values $A_{\alpha,i}$ with probabilities p_i. We assume that our knowledge concerning X is limited to the set of expectation values

$$\langle A_\alpha \rangle \equiv \sum_{i=1}^{\nu} p_i A_{\alpha,i}, \qquad \alpha = 1, \ldots, M, \qquad M \ll N. \tag{2}$$

It is obviously desirable to be in a position to ascertain which is the probability distribution $\{p_i\}$ since this would be tantamount to knowing everything that is to be known concerning X (one could predict the result of any measurement of the A_α). But, regrettably, this distribution is unknown. All our *a priori* information is that of the M figures $\langle A_\alpha \rangle$.

The main question is, consequently, the following: What can we assert with respect to the probability distribution $\{p_i\}$?

Of course, many such distributions are compatible with the amount of information provided by (2). Information theory claims (following Ockham) that the "best" (or the least-biased) one is that which maximizes Shannon's entropy (1). We are thus led to an extremalization problem

$$\delta_{\{p_i\}} \left[-\sum_i p_i \ln p_i - \lambda_0 \left\{ \sum_i p_i - 1 \right\} - \sum_{\alpha=1}^{M} \lambda_\alpha \left\{ \sum_i p_i A_{\alpha,i} - \langle A_\alpha \rangle \right\} \right] = 0 \tag{3}$$

in which the Lagrange multiplier λ_0 guarantees normalizations and the remaining λ_α's ensure compliance with the set of relations (2) (the input information).

Luckily, the variational problem (3) can be solved in an analytical fashion and easily yields the recipe for constructing the "one and only" $\{p_i\}$,

$$p_i = \exp \left(-(1 + \lambda_0) - \sum_{\alpha=1}^{M} \lambda_\alpha A_{\alpha,i} \right) . \tag{4}$$

From the normalization condition we now immediately cast λ_0 in the form

$$\lambda_0 = \ln \sum_{i=1}^{\nu} \exp \left[-\sum_{\alpha=1}^{M} \lambda_\alpha A_{\alpha,i} \right] \equiv \ln Z(\lambda_1, ..., \lambda_M) , \tag{5}$$

where Z is the (generalized) "partition" function and, introducing (4) and (5) into (2), we are led to

$$\frac{\partial \ln Z(\lambda_1, ..., \lambda_M)}{\partial \lambda_\alpha} = -\langle A_\alpha \rangle, \qquad \alpha = 1, ..., M, \tag{6}$$

which provides us with a set of coupled equations for the Lagrange multipliers $\lambda_1, ..., \lambda_M$. By solving the system (6) we find the "canonical" IT probability distribution (4).

It can be proved that p_i always exists and is uniquely determined by (5) and (6) [15] provided the input information on the right-hand side of (6) is not of a self-contradictory character. Moreover, with a little additional work (see, for instance, [15]) one is easily convinced that S is actually maximized and not merely extremalized. A standard well-known algorithm is available that yields the Lagrange multipliers [20].

Often, in addition to (2), some additional piece of information is available. One frequently knows beforehand that p_i is of the form

$$p_i = g_{1,i}\, g_{2,i}, \tag{7}$$

with $g_{1,i}$ known and $g_{2,i}$ unknown. The way to go in such a case is to maximize [instead of (1)] the so-called *relative entropy*

$$S' = -\sum_i p_i \ln [p_i / g_{1,i}] , \tag{8}$$

which produces no essential change in (4)–(6), except for the fact that one should place the unknown function $g_{2,i}$ on the left-hand side of (4).

III. PRESENT FORMALISM

Consider a student perceptron (SP) with N input units S_i connected to an output unit ζ whose state is determined according to $\zeta = g(h)$, where $g(x)$ is the transfer function and $h = \mathbf{S} \cdot \mathbf{W}$ is membrane potential of the output neuron. For each set of weights $\mathbf{W}$ the SP maps $\mathbf{S}$ on ζ. We train the SP with a set of P inputs $\mathbf{S}^{\mu}$, with $\mu = 1, ..., P$, and the corresponding appropriate outputs $\zeta_0(\mathbf{S})$, as provided by a teacher perceptron (TP) with weights $\mathbf{W}_0$. Of course, the SP and the TP share an identical architecture. It is obvious that

$$g^{-1}(\zeta_0^{\mu}) = \mathbf{S}^{\mu} \cdot \mathbf{W}, \tag{9}$$

where $\mathbf{S}^{\mu}$ is an input patterns matrix and $g^{-1}(\zeta_0^{\mu})$ is a vector of components $[g^{-1}(\zeta_0^{1}), ..., g^{-1}(\zeta_0^{P})]$, given by the output patterns, which constitute our available information. The idea is now to employ here the IT approach described in Sec. II [11,12,16] in order to determine the weights $\mathbf{W}$ on the basis of an *incomplete* information supply [in the present situation, the range of $\mathbf{S}^{\mu}$ is less than N, in general]. In order to infer weights consistent with Eq. (9) we shall assume that *each set of weights* $\mathbf{W}$ *is realized with probability* $P(\mathbf{W})$ (the esential IT ingredient). In other words, we introduce a (normalized) probability distribution over the collection of *conceivable* (possible) sets $\mathbf{W}$. This is quite reasonable on the basis that indeed *many* such sets are compatible with our incomplete information. Of course,

$$\int P(\mathbf{W}) d\mathbf{W} = 1, \tag{10}$$

where $d\mathbf{W} = dW_1 dW_2 \cdots dW_N$. Expectation values $\langle W_i \rangle$ are defined in the fashion

$$\langle W_i \rangle = \int P(\mathbf{W}) W_i d\mathbf{W} \tag{11}$$

and a *relative* entropy is, in the usual way [11–13], associated with the probability distribution, namely,

$$S = -\int P(\mathbf{W}) \ln\left(\frac{P(\mathbf{W})}{P_0(\mathbf{W})}\right) d\mathbf{W}, \tag{12}$$

where $P_0(\mathbf{W})$ is an appropriately chosen *a priori* distribution [see (7) and (8)] [11–13]. Following the central tenets of information theory, as reinterpreted by Jaynes [12], who embedded its procedural aspects within the maximum entropy (ME) principle [12], the entropy (12) is to be maximized subject to the constraints (9). Our *central* idea is to be introduced at this point. We will look upon Eq. (9) in the light of

$$g^{-1}(\zeta_0^{\mu}) = \mathbf{S}^{\mu} \cdot \langle \mathbf{W} \rangle, \tag{13}$$

where explicit account is taken of the fact that we are assumed to be dealing with *many* sets of weights, each one being realized with a given probability, and we borrow from statistical mechanics the idea that measured data are to be reproduced by theoretical averages [15].

As is customary [12], one is then led to freely maximizing the quantity

$$S' = -\int \left\{ P(\mathbf{W}) \ln\left(\frac{P(\mathbf{W})}{P_0(\mathbf{W})}\right) + \lambda_0 P(\mathbf{W}) + (\mathbf{S}^{\mu})^t \vec{\lambda} \mathbf{W} P(\mathbf{W}) \right\} d\mathbf{W}, \tag{14}$$

where λ_0 and $\overrightarrow{\lambda}$ are Lagrange multipliers associated, respectively, with the normalization condition (10) and the constraints (9). Variation of S' with respect to $P(\mathbf{W})$ immediately gives [see (4) and (7)]

$$P(\mathbf{W}) = \exp[-(1 + \lambda_0)] \exp(-\mathbf{\Gamma} \cdot \mathbf{W}) P_0(\mathbf{W}), \tag{15}$$

where $\mathbf{\Gamma} = (\mathbf{S}^{\mu})^t \overrightarrow{\lambda}$. As explained in Sec. II, one conveniently defines the partition function Z [see (5)]

$$Z = \int d\mathbf{W} \exp(-\mathbf{\Gamma} \cdot \mathbf{W}) P_0. \tag{16}$$

A choice is now to be made concerning the *a priori* probability distribution P_0 [11–13]. Following many authors [19], we select here a Gaussian P_0, i.e., choose it to be proportional to $\exp(-\mathbf{W} \cdot \mathbf{W}/2a^2)$, with a (formally) free parameter a. The results, however, do not depend upon the value of a.

It is now an easy matter to explicitly evaluate the partition function. We find

$$Z = \prod_{i=1}^{N} (2a^2\pi)^{1/2} \exp\left(\frac{a^2\Gamma_i^2}{2}\right), \tag{17}$$

so that with (11) and the distribution (15) one has, for the $\langle W_i \rangle$, the convenient expression

$$\langle W_i \rangle = -2a^2\Gamma_i. \tag{18}$$

Notice that the present (pseudoinverse) IT approach [14] entirely bypasses consideration of the set of equations (6), which constitutes its main virtue. Both the definition of $\mathbf{\Gamma}$ and the constraints (9) allow for the elimination of the

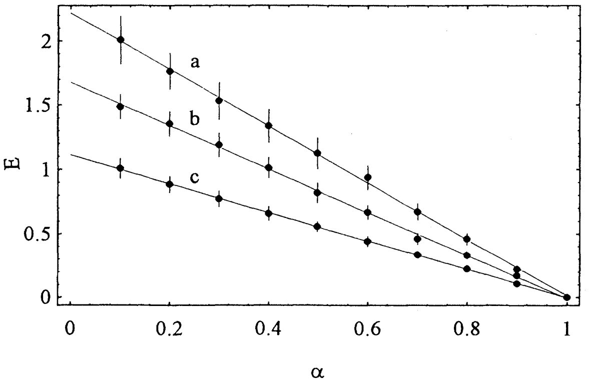

FIG. 1. Generalization error computed with 3000 new inputs and averaged over 200 networks with $g(x) = \tanh(x)$. (a) $N = 40$, (b) $N = 30$, and (c) $N = 20$.

TABLE I. $g(x) = x$. Generalization error E (averaged over 3000 cases). P stands for the number of training examples and R denotes the interval in which the testing procedure took place. Numbers in square brackets indicate powers of 10.

R	$P=1$	$P=2$	$P=5$	$P=8$	$P=9$	$P=10$
$[-1,1]$	0.165	1.02[−6]	1.85[−8]	2.64[−9]	1.30[−10]	5.08[−30]
$[-5,5]$	4.08	140[−6]	69.3[−8]	40[−9]	94.3[−10]	152[−30]
$[-50,50]$	422	1561[−6]	1284[−8]	105[−9]	527[−10]	17434[−30]

Lagrange multipliers $\overrightarrow{\lambda}$. One can thus express $\langle W_i\rangle$ solely in terms of the training examples

$$\langle \mathbf{W}\rangle = (\mathbf{S}^\mu)^t \left[\mathbf{S}^\mu (\mathbf{S}^\mu)^t\right]^{-1} g^{-1}(\zeta_0^\mu) . \tag{19}$$

The most probable configuration of weights [compatible with the constraints (9)] is thus given in terms of a Moore-Penrose pseudoinverse matrix (that of $\mathbf{S}^\mu$). This technique resembles (but is in fact distinct from) the Personnaz *et al.* [21,22] projection rule for memorizing (without errors) correlated patterns in the Hopfield model. Notice that with the choice (19) the training error vanishes. Additionally, the set of "inverse" examples $\{-\mathbf{S}^\mu, -\zeta_0(\mathbf{S}^\mu)\}$ possesses an associated distribution identical to that given by (15). Consequently, $-\zeta_0(\mathbf{S}^\mu)$ is that ouput produced by the network for the input $-\mathbf{S}^\mu$.

The above methodology cannot be applied to the Boolean perceptron, as the transfer function $g(x) = \operatorname{sgn}(x)$ does not possess an inverse. The following considerations are in order, however. Since $g^{-1}(\zeta_0) = h$, knowledge of the membrane potential h is required in order to determine the weights [cf. Eq. (19)]. Given the examples, this becomes an impossible task whenever $g(x) = \operatorname{sgn}(x)$. An approximate treatment may perhaps be available. If one takes into account *just the sign of* h, then $\operatorname{sgn}(h) = \zeta_0$, which leads to the pseudoinverse rule of Opper *et al.* [23]. For this rule an overfitting obtains in the region surrounding $\alpha = 1$ ($\alpha = P/N$), as noted by Vallet *et al.* [24]. This overfitting might be attributed to the approximation made as, from the examples, one is learning only the sign of h. For an invertible transfer function this difficulty does not arise: Exact knowledge of the membrane potential ($P = N$ examples) allows for an exact weight inference, no matter *which* transfer function we use.

A different way of approximately reconstructing the membrane potential would take $h = \mathbf{S}\cdot\mathbf{W}^*$, with $\mathbf{W}^*$ weights given by some alternative algorithm. This leads to

$$\mathbf{W} = (\mathbf{S}^\mu)^t \left[\mathbf{S}^\mu (\mathbf{S}^\mu)^t\right]^{-1} \mathbf{S}^\mu \cdot \mathbf{W}^* = \mathbf{W}^*$$

and no overfitting ensues.

Our results immediately generalize to networks with several output neurons. The appropriate map is given by $\sigma_j = g(\mathbf{S}\cdot\mathbf{W}_j)$ and the weights become

$$\mathbf{W}_j = (\mathbf{S}^\mu)^t \left[\mathbf{S}^\mu (\mathbf{S}^\mu)^t\right]^{-1} g^{-1}(\zeta_j^\mu) , \tag{20}$$

which is the ME recipe for the learning process.

IV. SIMPLE APPLICATIONS

Let us study now the performance of a perceptron trained according to the present formalism. We define the generalization error E as the average value (computed over a set of new questions) of the quantity

$$E = \frac{1}{2}\left[g(\mathbf{S}\cdot\mathbf{W}) - g(\mathbf{S}\cdot\mathbf{W}_0)\right]^2 . \tag{21}$$

Our procedure can be illustrated with reference to a perceptron that has been trained with the ME algorithm by exposing it to a variety of examples. We do this in a number of instances, varying each time the number of training examples. These, in turn, are provided by a randomly generated perceptron teacher W_0. The generalization error (21) is averaged over 3000 new examples provided by each TP (we deal here with 200 randomly generated networks). In Fig. 1 one easily appreciates how the generalization error falls down (to zero) when the value of the charge parameter $\alpha = P/N$ approaches unity. (This behavior obtains for *any* invertible transfer function.) This behavior is invariant with respect to the number of neurons because of the lack of normalization of E. Another illustrative example refers to an extremely simple problem. The task to be learned is that of finding the coefficients of a straight line that "fits" ten experimental points. The input information in these examples is given by a set of coordinates (of ten points). The output information consists of the associated least-squares values. The examples are restricted to some fixed interval of the abscissa axis. Our task can be exactly learned with a linear tranfer function. For a nonlinear transfer function $g(x)$ the concomitant results cannot be exact. It is thus appropriate to study the network's performance in

TABLE II. $g(x) = \tanh(x)$. Additional details are as in Table I.

$P=1$	$P=2$	$P=4$	$P=5$	$P=7$	$P=10$
0.164	5.31×10^{-3}	3.98×10^{-4}	5.81×10^{-4}	4.99×10^{-4}	1.74×10^{-3}

two cases, namely, for $g(x) = x$ and for $g(x) = \tanh(x)$.

Table I displays typical values of the generalization error (for several P values) when the transfer function is $g = x$. The errors are small not only within the training interval $[-1, 1]$ but also way beyond it. The net was tested with reference to intervals R as large as $[-5, 5], [-50, 50]$ (the task is an exactly learnable one).

In the case of a network within $g(x) = \tanh(x)$ the rule cannot be exactly learned: The SP and the TP have different architectures. The associated generalization error is displayed in Table II. The network's performance is good only within the training interval, as illustrated in Fig. 2, which depicts level curves for the generalization error as a function of the coefficients that define the straight line. A remarkable fact is to be emphasized: the rather *small* quantity of examples needed for the training process. This is certainly a notable facet of our approach, which differentiates it from other, more orthodox approaches (where a very slow convergence rate obtains if the number of examples is small enough).

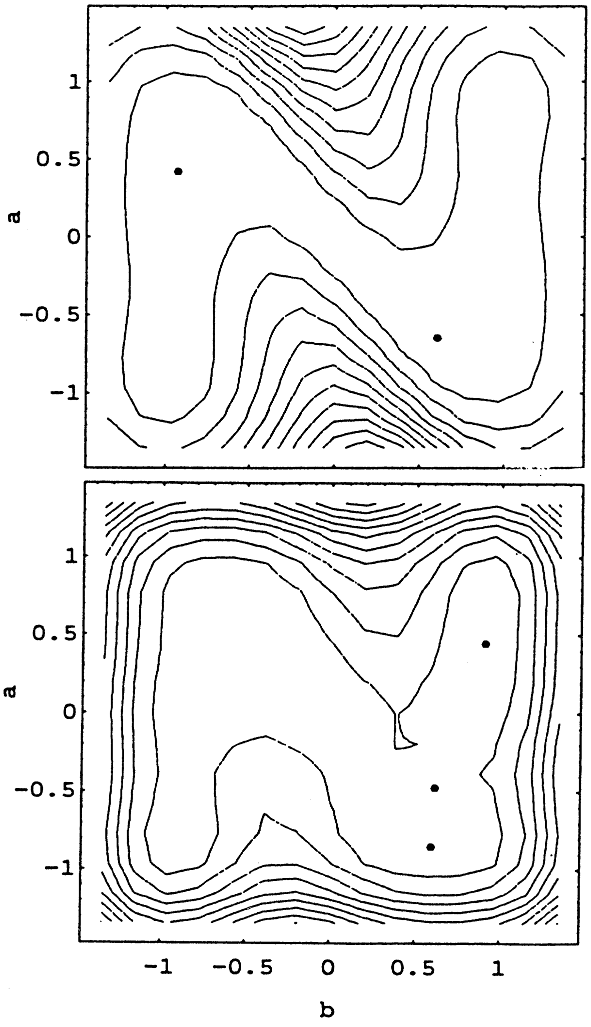

FIG. 2. Level curves for the generalization error as a function of the coefficients a and b of the straight line $ax + b$. The transfer function is $g(x) = \tanh(x)$. Black dots represent the examples (straight lines employed in the training process).

V. DISCUSSION

We have considered in this effort the learning of a rule with a neural network of continuous units and have been able to show that a pseudoinverse type of solution can be derived from the maximum entropy principle. We have illustrated our considerations with reference to simple examples and found that our procedure can be favorably compared to standard algorithms (SAs) that minimize an appropriate cost function (gradient descent with backpropagation). First of all, a delicate initial adjustment of the learning parameters is required in the case of the latter techniques (and is avoided in our case). Additionally, a careful "fine-tuning" process is needed in order to determine the parameters of the transfer function in order to attain "convergence" in networks that minimize a cost function. Such a process is entirely bypassed here. The SA solution (to which the network "converges") strongly depends upon the initial weights and, moreover, for some types of energy surface that are usually associated with "small" training sets, local minima of low generalization performance are to be regarded as inconvenient "traps" [25]. Our method is not bothered by such incoveniences. In the SA instance, a good training performance does not necessarily translate into a good generalization one. Our ME algorithm, on the other hand, *exactly learns* the training examples and provides one with an excellent generalization performance when different examples are to be confronted. We conclude then that pseudoinverse learning is a topic worth studying.

Summing up, an alternative approach to the learning process in a neural network has been added to the reservoir of learning techniques. It seems to offer promising perspectives.

ACKNOWLEDGMENTS

The authors acknowledge fruitful discussions with C. Mostaccio and M. T. Martin. L.D. would like to thank CICPBA (Argentina) for financial support. J.F. acknowledges support from UNLP (Argentina). A.P. is grateful for the support given by CONICET (Argentina).

[1] F. Rosemblatt, *Principles of Neurodynamics* (Spartan, New York, 1962).

[2] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing* (MIT Press, Cambridge, MA, 1986).

[3] H. S. Seung, H. Sompolinsky, and N. Tishby, Phys. Rev. A **45**, 6056 (1992).

[4] M. Opper and D. Haussler, Phys. Rev. Lett. **66**, 2677 (1991).

[5] T. Watkin, A. Rau, and M. Biehl, Rev. Mod. Phys. **65**, 499 (1993).

[6] S. Kirkpatrick, C. Gellat, and M. Vecci, Science **220**, 671 (1983).

[7] J. Holland, *Evolution, Learning and Cognition*, edited by Y. S. Lee (World Scientific, Singapore, 1988).

[8] D. B. Parker, MIT Technical Report No. TR-47, 1985 (unpublished).

[9] Y. Le Cun, *Disordered Systems and Biological Organization*, edited by E. Bienenstock, F. Fogelman, and G. Weisbuch (Springer, Berlin, 1986).

[10] E. Gardner, J. Phys. A **21**, 257 (1988).

[11] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication* (University of Illinois Press, Chicago, 1949).

[12] E. T. Jaynes, Phys. Rev. **108**, 171 (1957).

[13] R. D. Levine and M. Tribus, *The Maximum Entropy Principle* (MIT Press, Boston, 1978).

[14] J. Baker-Jarvis, J. Math. Phys. **30**, 302 (1989).

[15] A. Katz, *Principles of Statistical Mechanics* (Freeman, San Francisco, 1967).

[16] E. Duering, D. Otero, A. Plastino, and A. Proto, Phys. Rev. A **32**, 3681 (1985).

[17] R. Rossignoli and A. Plastino, Phys. Rev. A **42**, 2065 (1990).

[18] E. Duering, D. Otero, A. Plastino, and A. Proto, Phys. Rev. A **32**, 2455 (1985).

[19] J. Nunez, L. E. Rebollo-Neira, A. Plastino, R. D. Bonetto, D. M. A. Guerin, and A. G. Alvarez, X-Ray Spectrom. **17**, 47 (1985).

[20] N. Agmon, Y. Alhassid, and R. D. Levine, *The Maximum Entropy Formalism*, edited by R. D. Levine and M. Tribus (MIT Press, Cambridge, MA, 1979).

[21] L. Personnaz, I. Guyon, and G. Dreyfus, Phys. Rev. A **34**, 4217 (1985).

[22] T. Kohonen, *Self-organization and Associative Memory* (Springer-Verlag, Berlin, 1984).

[23] M. Opper, W. Kinzel, J. Kleinz, and R. Nehl, J. Phys. A **23**, L581 (1990).

[24] F. Vallet, J. Cailton, and P. Refregier, Europhys. Lett. **9**, 747 (1989).

[25] J. Freeman, *Simulating Neural Networks with Mathematica* (Addison-Wesley, Reading, MA, 1994).