Maximum entropy, pseudoinverse techniques, and time series predictions with layered networks

L. Diambra* and A. Plastino†
Departamento de Física, Universidad Nacional de La Plata, Casilla de Correo 727, (1900) La Plata, Argentina
(Received 1 May 1995; revised manuscript received 5 June 1995)

A maximum-entropy-based method for the training of layered networks is presented. Our technique guarantees an errorless learning process for learnable mappings with just a minimum number of examples. The network is proposed for nonlinear systems prediction. Some numerical examples for chaotic time series are presented. The method can be considered to yield an alternative tool for feed-forward training.

PACS number(s): 87.10.+e, 05.20.−y, 05.45.+b

I. INTRODUCTION

During recent years a great deal of effort has been invested in the development of training algorithms for feed-forward 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 (SP) trained by a teacher perceptron (TP) 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 Occam's razor, i.e., with the miminum number of assumptions compatible with the available input. This is conveniently done by recourse to the information theory (IT) approach to statistical mechanics, as embedded in the maximum entropy principle [11–13]. A learning protocol will be developed in this fashion and applied to simple layered networks.

II. THE MAXIMUM ENTROPY PSEUDOINVERSE TRAINING TECHNIQUE

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

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

where ξ^μ is an input-patterns matrix and $g^{-1}(\zeta_0^\mu)$ is a vector of components $(g^{-1}(\zeta_0^1), g^{-1}(\zeta_0^2), ..., g^{-1}(\zeta_0^P))$, given by the output patterns, which constitute our available information. The idea is now to introduce a maximum entropy approach [11–13] in order to determine the weights $\mathbf{W}$ on the basis of an *incomplete* information supply [in the present situation, rank $(\xi^\mu) < N$, in general]. In order to infer weights consistent with Eq. (1) we shall assume that *each set of weights* $\mathbf{W}$ *is realized with probability* $P(W)$ (our essential IT ingredient). In other words, we introduce a (normalized) probability distribution over the possible sets $\mathbf{W}$. Of course,

$$\int P(\mathbf{W}) d\mathbf{W} = 1, \tag{2}$$

where $d\mathbf{W} = dW_1 dW_2 \cdots dW_N$. Expectation values $\langle W_i \rangle$

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

are defined in the fashion

$$\langle W_i \rangle = \int P(\mathbf{W}) W_i d\mathbf{W}, \tag{3}$$

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

$$S = -\int P(\mathbf{W}) \ln \left(\frac{P(\mathbf{W})}{P_0(\mathbf{W})} \right) d\mathbf{W}, \tag{4}$$

where $P_0(\mathbf{W})$ is an appropriately chosen *a priori* distribution [11–13]. This entropy is to be maximized, subject to the constraints (1). Our central idea is that we reinterpret these equations according to

$$g^{-1}(\zeta_0^\mu) = \xi^\mu \cdot \langle \mathbf{W} \rangle, \tag{5}$$

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.

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

$$S' = -\int \left\{ P(\mathbf{W}) \ln \left(\frac{P(\mathbf{W})}{P_0(\mathbf{W})} \right) + \alpha P\left(\mathbf{W}\right) + \left(\xi^\mu\right)^t \lambda \mathbf{W} P\left(\mathbf{W}\right) \right\} d\mathbf{W}, \tag{6}$$

where α and λ are Lagrange multipliers associated, respectively, with the normalization condition (2) and with the constraints (1). Variation of S' with respect to $P(\mathbf{W})$ immediately gives

$$P(\mathbf{W}) = \exp\left[-(1+\alpha)\right] \exp\left(-\mathbf{\Gamma} \cdot \mathbf{W}\right) P_0(\mathbf{W}), \tag{7}$$

where $\mathbf{\Gamma} = (\xi^\mu)^t \lambda$. As in statistical mechanics, one conveniently defines the partition function Z

$$Z = \int d\mathbf{W} \exp\left(-\mathbf{\Gamma} \cdot \mathbf{W}\right) P_0. \tag{8}$$

A choice is now to be made concerning the *a priori* probability distribution P_0 [11–13]. Here we select a Gaussian P_0, i.e., choose it to be proportional to $\exp\left(-\frac{\mathbf{W}\cdot\mathbf{W}}{2a^2}\right)$, 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

$$Z = \prod_{i=1}^{N} \left(2a^2\pi\right)^{1/2} \exp\left(\frac{a^2\Gamma_i^2}{2}\right), \tag{9}$$

so that with (3) and the distribution (7) one has, for the $\langle W_i \rangle$, the convenient expression

$$\langle W_i \rangle = -2a^2\Gamma_i. \tag{10}$$

Notice that the present (pseudoinverse) approach entirely bypasses consideration of the set of equations (1), which constitutes its main virtue. Both the definition of $\mathbf{\Gamma}$ and the constraints (1) allow for the elimination of the Lagrange multipliers λ. *One can thus express the* $\langle W_i \rangle$ *solely in terms of the training examples*

$$\langle \mathbf{W} \rangle = I_{\mathrm{MP}}\left[\xi^\mu\right] g^{-1}\left(\zeta_0^\mu\right), \tag{11}$$

where $I_{\mathrm{MP}}\left[\xi^\mu\right] = \left(\xi^\mu\right)^t \left[\xi^\mu \left(\xi^\mu\right)^t\right]^{-1}$ is the Moore-Penrose pseudoinverse. The most probable configuration of weights [compatible with the constraints (1)] is thus given in terms of a pseudoinverse matrix (that of ξ^μ). This result is intimately related to the Bös *et al.* [14] learning rule. Notice that with the choice (11) the training error vanishes. Additionally, the set of "inverse" examples $\{-\xi^\mu, -\zeta_0(\xi^\mu)\}$ possesses an associated distribution identical to that given by (7). Consequently, $-\zeta_0(\xi^\mu)$ is that ouput produced by the network for the input $-\xi^\mu$.

III. TIME SERIES APPLICATIONS

We aim to apply the above discussed methodology to time series predictions, with reference to chaotic systems. Good predictions of such a kind find application in diverse areas, specially in connection with signal processing. It goes without saying that chaotic systems are abundant and provide an excellent test for the predictive ability of neural networks. Of course, a deterministic algorithm for chaotic time series would not be lacking for interested users.

For our present purposes we make use of the standard trick of placing high order monomials in the inputs of a projector [15]. With these monomials as inputs, the state of the output neuron is given by

$$\zeta = g\left(\theta + \sum_i w_i \xi_i + \sum_{ij} w_{ij} \xi_i \xi_j + \sum_{ijk} w_{ijk} \xi_i \xi_j \xi_k + \cdots\right), \tag{12}$$

restricting our analysis, for simplicity's sake, to the first orders in (12), this being a good approximation in most instances. With the threshold θ and the weights w_i, w_{ij}, and w_{ijk} we build up that particular vector $\mathbf{W}$ which satisfies the relation $g^{-1}(\zeta_0^\mu) = \boldsymbol{\delta}^\mu \cdot \mathbf{W}$, where a vector $\boldsymbol{\delta}$ has been introduced whose components are zero order, first order, second order, etc., terms. The maximum entropy algorithm prescribes that the most probable configuration of weights compatible with the relevant constraints is given by

$$\mathbf{W} = I_{\mathrm{MP}}\left[\delta^\mu\right] g^{-1}\left(\zeta_0^\mu\right). \tag{13}$$

This is the maximum entropy recipe for the learning process to be invoked in the applications to be discussed here. The concomitant transfer function will be chosen to be of the linear form $g(x) = x$.

Our task is now that of starting with some data

$$\{x(t_i), x(t_i - \Delta), \ldots, x(t_i - m\Delta); x(t_i + \tau)\} ,$$

$$i = 1, \ldots, P, \quad (14)$$

where P is the number of patterns used in order to train the network with the present algorithm, and τ denotes a suitable temporal step. Now, given the new input $x(t), x(t-\Delta), \ldots, x(t-m\Delta)$ the network should correctly predict $x(t+\tau)$.

Takens [16] has shown that, for flows evolving to compact attracting manifolds of dimension d_a, a functional relation of the type

$$x(t+\tau) = f(x(t), x(t-\Delta), \ldots, x(t-m\Delta)) \quad (15)$$

exists, and that m lies in the range $d_a < m+1 < 2d_a+1$. Takens's theorem provides no information either on how to choose Δ or on the form of f. Obviously, due to the nature of chaotic systems the exactitude of the $x(t+\tau)$ prediction decreases with τ. In the present effort our procedure will be illustrated with reference to the well-trodden logistic map and forced pendulum.

A. Logistic map

We consider here the time series x_i generated by the logistic map

$$x_{n+1} = C\, x_n (1 - x_n) . \quad (16)$$

For $C = 4$, the series is a chaotic one for (almost) any initial value $\in (0,1)$. Data generated by the logistic map are now subdivided into two parts. P examples are employed in order to train the network $((x, x_{n-1}, x_{n-2}, x_{n-3},)^{\mu}, x_{n+1}^{\mu})$, $\mu = 1, \ldots, P$; while the other part tests its subsequent predictive power.

In a typical DX2 486 PC run, our maximum entropy network is trained in a few seconds with just 30 examples ($P = 30$). The associated mean square deviation between maximum entropy predictions and correct outputs, for 200 points x_n, is 1.6×10^{-30}. This is to be compared with the performance of an orthodox feed-forward neural network trained with error back propagation. If the training set has not 30 but 500 examples, after training it during several hours, we get a mean square error of about 2.7×10^{-4}. Figure 1 displays predictions by networks trained with (i) our maximum entropy technique (solid line) and (ii) the back-propagation method (dashed line).

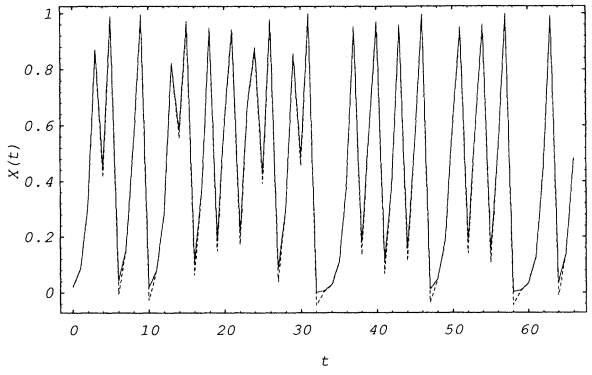

FIG. 1. Logistic map one-step-ahead predictions obtained with (i) maximum entropy neural networks (solid line) and (ii) networks trained with back propagation (dashed line). We take $C = 4$. The solid line fits the time series data within eight significant digits. The main difference between the two training approaches resides in the fact that ours is much quicker.

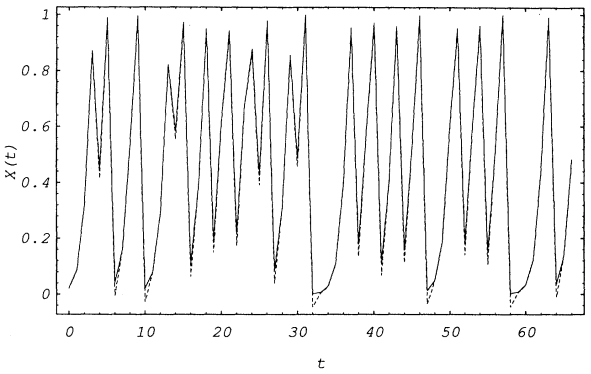

FIG. 2. Two-steps-ahead ($\tau = 2$) predictions corresponding to (i) maximum entropy networks (dashed line) and (ii) networks trained with back propagation (dotted line), for the forced pendulum time series (solid line). Additional details are as in Fig. 1.

B. Forced pendulum

This application confronts the network with a more involved system. The pertinent differential equation reads

$$\ddot{y} + 1/q\dot{y} + \sin y = g\cos(\omega_D t) \quad (17)$$

where q is the damping factor and g stands for the in-

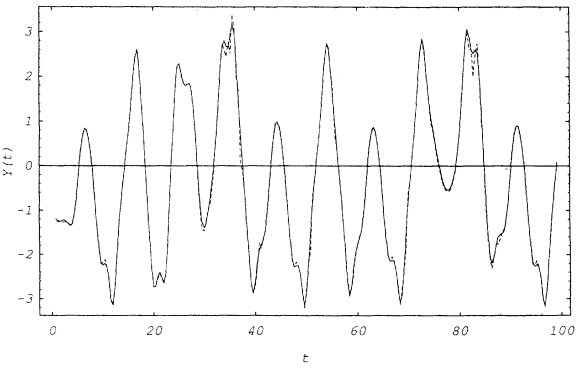

FIG. 3. Four-steps-ahead ($\tau = 4$) predictions by our neural networks (dashed line) and the time series (solid line) from the forced pendulum.

homogeneity amplitude and ω_D its frequency. In the present instance we take $q = 3.5$, $g = 1.5$, and $\omega_D = 2/3$. These values account for an attractor dimensionality of the order of 2.38, so that it suffices to consider [see Eq. (15)] $m = 4$.

Our differential equation (16) is tackled, as usual, by recourse to the Runge-Kutta method in order to acquire the data needed for use in the training process. The training data are given by $((y_n, y_{n-1}, y_{n-2}, y_{n-3})^\mu, y^\mu_{n+\tau})$, with $\mu = 1, \ldots, P$. For $\tau = 2$, and after a training session of a few seconds, our maximum entropy predictions ($P = 60$) are characterized by a mean square error (200 points) of 5×10^{-3}. For the back-propagation network we need several hours of training, with 500 examples, to obtain a mean square error of about 5.3×10^{-2}. Figures 2 and 3 display maximum entropy predictions for different τ values. As expected, the larger τ, the worse the performance.

IV. DISCUSSIONS AND CONCLUSIONS

A general method for time series prediction has been presented that makes use of a feed-forward network trained with a maximum entropy algorithm. Very good results are obtained. 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. By suitably increasing the number of components of the input vector [the matrix of the P inputs ξ^μ is associated, via the matrix δ^μ (see above), with the corresponding outputs ζ_0^μ] the input patterns are able to "capture" the essential correlations of the system in a rather natural fashion. This allows for the elimination of intermediate layers of more complex architectures, and thus reduces the training times. Additionally, alternative approaches [17,18] suffer from very slow convergence rates if the number of examples is small enough.

Summing up, 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 two examples, where it becomes apparent that pseudoinverse learning is a topic worth studying. Thus a maximum entropy approach to the learning process in a neural network has been added to the reservoir of learning techniques that seems to offer promising perspectives.

ACKNOWLEDGMENTS

L.D. would like to thank CICPBA and A.P. CONICET of Argentina for financial support.

[1] F. Rosemblatt, *Principles of Neurodynamics* (Spartan, New York, 1962).
[2] D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing* (MIT, 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, in *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, in *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, Cambridge, MA, 1978).
[14] S. Bös, W. Kinzel, and M. Opper, Phys. Rev. E **47**, 1384 (1993).
[15] I.J. Matus and P. Perez, Phys. Rev. A **43**, 5683 (1991).
[16] F. Takens, in *Dynamical Systems and Turbulence Warwick*, edited by D. Rand and L.-S. Young, Lecture Notes in Mathematics Vol. 898 (Springer, Berlin, 1981), p. 366.
[17] A. Lapedes and R. Farber, Los Alamos National Laboratory Report No. LA-UR-87-2662, 1987 (unpublished).
[18] A. Lapedes and R. Farber, in *Neural Information Processing Systems*, edited by D.Z. Anderson (AIP, New York, 1987), p. 442.