Learning and generation of long-range correlated sequences

A. Priel* and I. Kanter
Minerva Center and Department of Physics, Bar-Ilan University, 52900 Ramat-Gan, Israel
(Received 11 January 2000)

We study the capability to learn and to generate long-range, power-law correlated sequences by a fully connected asymmetric network. The focus is set on the ability of neural networks to extract statistical features from a sequence. We demonstrate that the average power-law behavior is learnable, namely, the sequence generated by the trained network obeys the same statistical behavior. The interplay between a correlated weight matrix and the sequence generated by such a network is explored. A weight matrix with a power-law correlation function along the vertical direction, gives rise to a sequence with a similar statistical behavior.

PACS number(s): 84.35.+i, 05.10.−a

I. INTRODUCTION

Real-life (temporal) sequences are characterized by a certain degree of correlation. It is known that a wide range of systems in nature displays long-range correlations, e.g., biological-DNA sequences and heartbeat intervals, natural languages, etc., see Refs. [1,2]. Since long-range correlations can appear in many forms, we restrict the analysis to the case of power-law correlations in a random sequence, e.g., the correlation function for a 1D sequence x_i is given by

$$C(l)=\langle x_i x_{i+l}\rangle \propto l^{-\gamma} \quad (l\to\infty,\ \gamma>0), \tag{1}$$

where the angular brackets denote an average over the randomness. This type of random sequence is also termed ''colored'' or correlated-noise. The general form of description allows us to investigate the capability of the network to capture statistical properties of a sequence.

The theory of learning from examples by a neural network, and in particular online learning, has been developed almost exclusively for uncorrelated patterns, see Refs. [3,4]. Though some particular cases of correlated patterns were treated, they were limited to simple spatial correlations within each pattern, or to temporal correlations of each input unit, e.g., Ref. [5]. The case of long-range correlations is absent. Clearly, the problem of extracting a feature from a correlated sequence whose length is much larger than the network's size, cannot be treated under the same assumptions. Rather than dealing with the question of the generalization error (average over a distribution of patterns), we focus on the capability of the network to asymptotically capture the correlations within the sequence and its ability to generate a sequence with similar properties.

As shown previously, the sequence generator (SGen), a continuous-valued feed-forward network in which the next state vector is determined from past output values, exhibits (quasi) periodic attractors in the stable regime, regardless of the complexity of the weights, both in the case of a perceptron as well as multilayer SGen's, e.g., Refs. [6–8]; the unstable, chaotic regime is studied in Ref. [9]. Therefore, it is obvious that the perceptron-SGen, or its extension to multilayer networks, are not suitable candidates for learning and generating correlated noise. The natural way to overcome this limitation is to increase the complexity of the feedback in the architecture. In this paper we study a fully connected, asymmetric network. The updating rule for the network's state is either sequential or parallel, namely, each unit is updated on its turn or all units are updated simultaneously. Unit i $(i=1,\ldots,N)$ is updated as follows:

$$S_i^{t+1}=\tanh\left(\beta\left[\sum_{j=1}^{i-1} W_{ij}S_j^{t+1}+\sum_{j=i}^{N} W_{ij}S_j^{t}\right]\right), \tag{2}$$

$$S_i^{t+1}=\tanh\left(\beta\sum_{j=1}^{N} W_{ij}S_j^{t}\right), \tag{3}$$

where Eqs. (2),(3) refers to the sequential (parallel) rule; $\mathbf{W}$ is an $[N\times N]$ weight matrix and β is a gain parameter. The network generates iteratively an infinite sequence $\{\sigma_m\}$ starting from an initial state $\mathbf{S}^0$ as follows:

$$\sigma_m=S_i^{t+1},\quad m=tN+i=1,2,\ldots, \tag{4}$$

where $i=1,\ldots,N,\ t=0,1,2\ldots$.

Two complementary issues are discussed in this paper: (a) Given a training sequence characterized by long-range correlations, can we train a network in an online scheme to generate a sequence with the same asymptotic statistical properties? (b) The inverse problem, is there an interplay between a network whose weight matrix follows a power-law correlation function, and the sequence it generates? It is important to stress that the model we investigate is not proposed as a practical method for generating long-range correlated sequences, rather, it is motivated by the issues raised above.

In the next section we investigate the first question. An ''online,'' gradient-based learning rule is applied where each example is presented to the network only once. The sequences that constitute the basis for the training patterns and the correlated weight matrices, are generated using an algorithm for re-shaping the power spectrum of an uncorrelated sequence; the method has been developed for investigating various stochastic processes, see Ref. [10]. In Sec. III the inverse problem is analyzed. A method for constructing the

*URL: http://faculty.biu.ac.il/~priel

weight matrix is presented based on the findings obtained from Sec. II regarding the correlation properties of the weights in trained networks. A simple analytical derivation support these findings.

II. GENERALIZING THE RULE OF A COLORED SEQUENCE

Suppose a source generating sequences that obey Eq. (1); the question we address in this section focuses on the possibility of learning the statistical properties of the source, and in particular the exponent γ of the power-law correlation function. For a network defined by its weights $\mathbf{W}$, a gain β and a nonlinear function f, the response of the ith unit, given the current state ξ^t, is

$$S_i^{t+1}=f(\xi^t,\mathbf{W}_i),$$

where $\mathbf{W}_i$ denotes the vector of weights connected to the ith unit. The online learning algorithm minimizes a quadratic error function

$$\epsilon_i(\xi^t,\mathbf{W}_i)=[S_i^{t+1}-\tau_i^{t+1}]^2/2, \tag{5}$$

where τ_i^{t+1} is the desired response of the ith unit given the state ξ^t. The weights are updated according to

$$\mathbf{W}_i^{t+1}=\mathbf{W}_i^t-\frac{\eta}{N}\nabla_W\epsilon_i(\xi^t,\mathbf{W}_i^t), \tag{6}$$

i.e., a gradient descent rule with a learning rate η (similar results were obtained using the Hebbian learning rule).

The training patterns are defined as follows. Let $D_L=\{x_1,x_2,\ldots,x_L\}$ be a 1D sequence obeying Eq. (1). A training pattern [the pair (ξ^m,τ_i^m) $0<m\leqslant L-2N+1$, $i=1,\ldots,N$] is defined by

$$\begin{aligned}\xi^m&=(x_m,x_{m+1},\ldots,x_{m+N-1}),\\ \tau_i^m&=x_{m+N+i-1},\end{aligned} \tag{7}$$

where each weight vector $\mathbf{W}_i$ is updated with the corresponding desired output τ_i^m, and the same vector ξ^m. Updating all N weight vectors for a given pattern $(\xi^m,\tau_i^m$ $i=1,\ldots,N)$, accounts for a single training cycle. The patterns for consecutive training cycles are achieved via sliding a window of size N by one site along the sequence D_L; e.g., given a current pattern starting at site m along the training sequence, ξ^m [Eq. (7)], the next pattern is $\xi^{m+1}=(x_{m+1},x_{m+2},\ldots,x_{m+N})$. The training patterns can be obtained in a different scheme by sliding the window N sites (nonoverlapping windows); i.e.,

$$\begin{aligned}\xi^{m+1}&=(x_{mN+1},x_{mN+2},\ldots,x_{mN+N}),\\ \tau_i^{m+1}&=x_{(m+1)N+i},\quad m=0,1,\ldots.\end{aligned} \tag{8}$$

Results obtained in both schemes are similar, however, the length of the sequence (D_L) used in the second scheme (to obtain the same results) is about N times larger.

Let us now describe our numerical investigation. An ensemble of long sequences, D_L (of size $L\gg N$), that obey Eq. (1) with a given γ is generated. A randomly chosen network is trained using a part of the sequence. Taking the last pattern from the training process as an initial state, the trained network is used to generate iteratively a long sequence $\{\sigma_m\}$, of size MN with $M=10$, following the sequential rule, Eqs. (2) and (4). The correlation function of this sequence can be calculated in the two following ways: spatial and temporal. The spatial correlation is obtained by averaging the correlation function, calculated on (M) sequences of size N after updating all (N) units, over the M iteration cycles, whereas the temporal correlation is simply the correlation function of the long sequence, i.e.,

$$\begin{aligned}C_{\text{spat}}(l)&=1/M\sum_{j=0}^{M-1}\left[1/N\sum_{i=1}^{N}\sigma_{jN+i}\sigma_{jN+(i+l \bmod N)}\right],\\ C_{\text{temp}}(l)&=(MN)^{-1}\sum_{i=1}^{MN}\sigma_i\sigma_{i+l},\end{aligned} \tag{9}$$

where we take periodic boundary conditions. Next, the same weight matrix is further trained and after each αN patterns $(\alpha=10)$, the same process of ''generating a sequence and calculating its correlation function,'' is repeated for a better statistical estimation. We found that both definitions of the correlation function yield similar results, therefore, in the sequel we omit the subscript from Eq. (9) and refer to the temporal correlation function

$$C(l)=C_{\text{temp}}(l). \tag{10}$$

Note that the range of correlations is bounded by the number of degrees of freedom $\{S_i\}_{i=1}^N$, as in Ref. [10]; therefore, the correlation function is calculated in the range $0<l<N/2$ (a symmetric function). The whole procedure is applied for all members of the ensemble. This extensive averaging is necessary since the patterns ξ^m taken from the long sequences D_L, exhibit large fluctuations (recall that $N\ll L$ and the variance decreases linearly with the size of the sequence).

Figure 1 depicts the results of the above procedure for $\gamma=0.4,0.6,0.8$ $(N=200)$, with $L\approx 10^5$ and an ensemble of

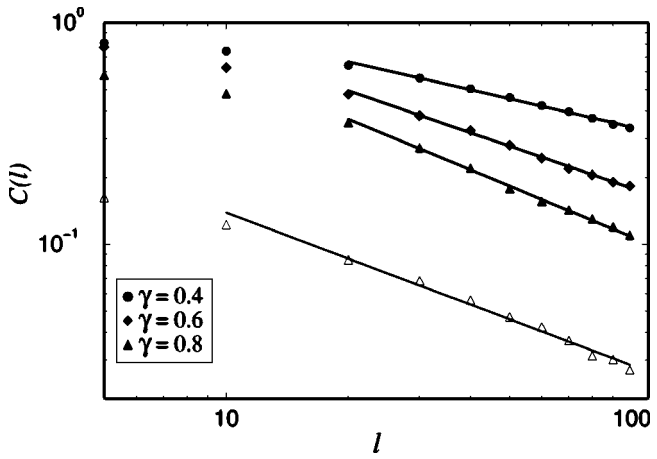

FIG. 1. The correlation function $C(l)$ [Eq. (10)] of the sequences generated by the trained networks with $N=200$. The training patterns are generated from correlated 1D sequences with $\gamma=0.4,0.6,0.8$. $C(l)$ is shown along with the power-law regression lines; the respective exponents are 0.42, 0.63, 0.76. The opaque triangle points correspond to training by sliding N sites each cycle, and the exponent of the regression line is 0.7 (for $\gamma=0.8$).

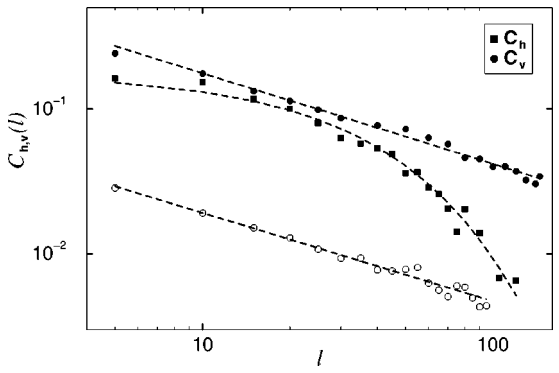

FIG. 2. The correlation functions of the weights **W** of trained networks with $N=300$. $C_{h(v)}$ is averaged over the rows (columns) of the weight matrix [Eq. (11)]. The dashed lines correspond to regression fits: a power-law $C_v(l)\sim l^{-0.625\pm0.016}$, and exponential $C_h(l)\sim\exp(-al)$ $a=-0.03\pm0.001$. The opaque circles represent C_v in the case of training by sliding N sites each cycle for a network with $N=200$. The power-law regression fit is $C_v(l)\sim l^{-0.61\pm0.025}$.

50–100 samples. For comparison, we show the results of training with patterns obtained by sliding the window N sites each cycle [nonoverlapping windows, Eq. (8)]; in this case, the sequence is much larger, $L\approx10^7$. The data points are the average values with relative error-bars that vary from 5% for small l to 40% (less than 20% for nonoverlapping windows) of the data point for $l\approx N/2$, hence, we omit them to preserve the clarity of the figure. The learning rate used, $\eta=2$ [Eq. (6)], is not optimal; it is obvious that an optimization can reduce the fluctuations in the correlation function since it affects the relative change in the weights. We note that the fluctuations inspected in the sequences $\{\sigma_m\}$ generated by the trained networks are similar to those of the training patterns ξ, indicating a finite size effect. This has been confirmed for several network sizes.

It is interesting to examine how the network (learning algorithm) has embedded the relevant information associated with the correlations. At the end of the training process described above, we measured the correlation function of the weight vectors (averaged over realizations of D_L) in two directions, horizontal (over rows) and vertical (over columns), as follows:

$$C_h(l)=\left\langle \sum_{i,j=1}^{N} W_{i,j}W_{i,j+l \bmod N}\right\rangle, \quad \text{horizontal},$$
$$C_v(l)=\left\langle \sum_{i,j=1}^{N} W_{i,j}W_{i+l \bmod N,j}\right\rangle, \quad \text{vertical}. \tag{11}$$

Results are presented in Fig. 2 for a training sequence obeying a power-law correlation function with exponent $\gamma=0.6$ and a network of size $N=300$. Clearly, the vertical correlations follow a rule similar to that of the sequence, $C_v(l)\sim l^{-0.625}$, while the horizontal correlations decay much faster, with an exponential fit $C_h(l)\sim\exp(-0.03\,l)$. The case of training with patterns obtained from nonoverlapping windows is presented for comparison by the opaque circles. In this case, the vertical correlations are similar, $C_v(l)\sim l^{-0.61}$, however, there are no horizontal correlations. We conclude that sliding one site each cycle induces horizontal correlations, however they decay exponentially fast.

The issue of learning the rule from a teacher is not treated in this framework, however, we add the following (numerical) observation: The overlap between two (initially random) networks, $R=\mathbf{W}^1\cdot\mathbf{W}^2$, learning from the same rule (training sequence) decreases with the size of the network (N) and remains very low even for $L\gg N^2$, although the two networks generate sequences with similar correlation functions, asymptotically. The same holds when each network learns from a different sequence. We conclude that the network indeed learns the statistical properties of the sequence, but not its values. Clearly, in batch learning one expects that the network would learn the rule when $L=N^2$, since the number of free parameters (weights) equals the number of examples.

III. GENERATING COLORED SEQUENCES BY A COLORED NETWORK

So far, we demonstrated the capability of the network to capture statistical properties from the training sequence. Let us now consider the inverse problem, i.e., that of constructing a network that is capable of generating correlated sequences. The information obtained from the trained networks in the preceding section regarding the structure of the weight matrix, suggests that the significant correlation is present between the elements of a column, i.e., vertical direction. Therefore, we would like to compare the correlation function of sequences generated by networks with the same vertical correlation function (power law), and various horizontal decay forms, i.e., power law with an increasing exponent γ. The weights are constructed as follows. Start by generating a random matrix of normally distributed elements. Each column is treated as a 1D sequence and is ''colored'' following the process described above for generating a 1D correlated sequence. After this stage, the rows are still uncorrelated. To achieve a different power-law function for the rows, we treat each row independently as a 1D sequence and follow the same procedure as above, this time with (possibly) a different exponent. This process generates a weight matrix with pronounced correlations in the vertical and horizontal directions only. We normalize the weights $\Sigma_{i,j=1}^{N}W_{(i,j)}^2=N$, such that $\beta=\mathcal{O}(1)$ (independent of N). The value of β in the dynamic equations, Eqs. (2),(3), is taken well above bifurcation to increase the probability of nonperiodic attractors [11] (we carefully avoid the periodic attractors in our measurements). In the analysis described below, each sample network (colored **W**) is initialized at random ($\mathbf{S}^0$) and the correlation function of the sequence generated is calculated at long times.

Figure 3 depicts two cases for which the vertical correlation function of the weights decays polynomially with exponents $\gamma_v=0.4$ and $\gamma_v=0.6$. For each case, the horizontal correlation function takes one of the following three values: $\gamma_h=\gamma_v$, $\gamma_h=2\gamma_v$ or uncorrelated. The results were obtained for a network of size $N=2048$ and averaged over 50 realizations of the weight matrix. Additional averaging is done by starting from several initial conditions for each matrix. It is apparent that the symmetric case, $\gamma_h=\gamma_v$, gives rise to a relatively poor long-range correlations in the generated sequence. The other two cases exhibit much longer-range cor-

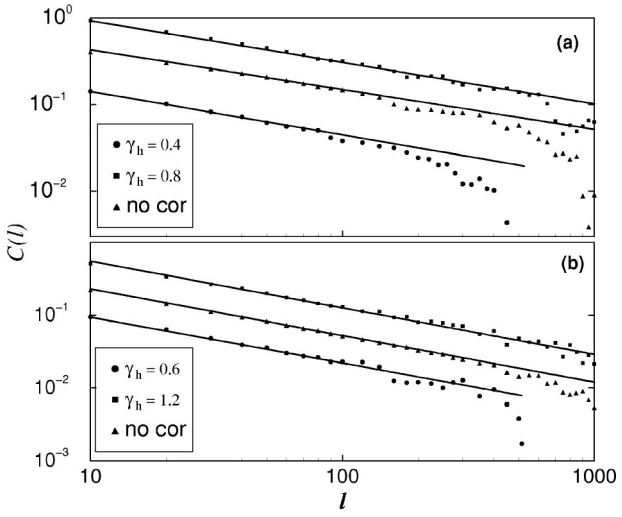

FIG. 3. The correlation function of sequences generated by a constructed colored network, $N=2048$. (a) $\gamma_v=0.4$, (b) $\gamma_v=0.6$. γ_h is given in the figure (''no cor'' stands for no horizontal correlations). The solid lines are the power-law regression fits with exponents: (a) 0.47 (top line), 0.46 (middle), and 0.5 (bottom); (b) 0.64 (top), 0.64 (middle), and 0.63 (bottom)

relations. Although we are not trying to determine the optimal correlation function, it seems that weak correlations are better than lack of horizontal correlations. This is in agreement with our findings regarding the trained networks, see Fig. 2.

To conclude, we propose a naive calculation which should serve as a starting point to the analytical investigation of the model. The quantity of interest in our calculation is the asymptotic correlation function

$$C(l)=Z\langle S_i^t S_{i+l}^t\rangle_{t,W}, \tag{12}$$

where Z is a normalization factor $[C(0)=1]$. The average is taken over the time t and the realizations of the weights, expecting $C(l)$ to be independent of the site i. For simplicity we use the parallel updating rule, Eq. (3), hence, the stationary state of the correlation function may be given by

$$C(l)=Z\left\langle \tanh\left(\beta\sum_{j=1}^{N} S_j^t W_{i,j}\right)\tanh\left(\beta\sum_{j=1}^{N} S_j^t W_{i+l,j}\right)\right\rangle_{t,W}. \tag{13}$$

The approximation of Eq. (13) consists of linearizing the right-hand side and assuming **S** independent of the realization of **W**, leading to

$$C(l)\approx\sum_{j,k=1}^{N}\langle S_j^t S_k^t\rangle_t\langle W_{i,j}W_{i+l,k}\rangle_W. \tag{14}$$

Next, we identify the averages as the correlation functions, defined above, that depend on the distance only, and rewrite Eq. (14) using $m\equiv k-j$ in the form

$$C(l)=\hat{Z}\sum_{m=1}^{N} C(m)C_W(l,m), \tag{15}$$

where $\hat{Z}$ is a normalization factor, $C_W(l,m)$ denotes the 2D correlation function of the weights, and $C(l)$ is the 1D function which is the quantity of interest. In the scenario described above, C_W is known *a priori* (independent of time) since the weights are constructed. If we assume a power-law vertical correlations for C_W and no horizontal correlations, the correlation function of the sequence $C(l)$, simply follows the vertical correlations of the weights, i.e., $C(l)=\hat{Z}\Sigma_m C_W(l,m)\,\delta_{m,0}=\hat{Z}C_W(l,0)$, supporting the above findings. We remark that this decomposition of C_W into independent vertical and horizontal functions is still a good approximation when C_h is not a delta function, as long as C_h decays much faster than C_v, which enables us to neglect other correlations. In this case Eq. (15) can be formulated as an ''eigenvalue problem'' of the matrix C_W. Few such cases have been solved numerically for which the assumption of decomposition was found consistent, see Ref. [12]. When this decomposition is no longer valid, we observe a breakdown of the long-range power-law behavior, see Fig. 3—the case $\gamma_h=\gamma_v$.

IV. DISCUSSION

In this paper we analyzed the capability of a neural network model to learn the rule of a long-range correlated sequence on the one hand, and a method for constructing a network that is able to generate such sequences on the other hand. We demonstrated that a simple online learning algorithm can be used to extract the rule, provided that the sequence is long enough. The fluctuations observed in the training patterns are manifested in the generated sequences as well. The investigation of the weights that were obtained during the learning process indicates that the vertical correlations [C_v, Eq. (11)] play the most important role in generating correlated sequences by the network. Employing this finding, we were able to construct networks (without training) that are capable of generating sequences with a predefined power-law correlation function. Indeed, we found that additional significant horizontal correlations in the constructed networks' weights corrupt this property. These observations were confirmed by a naive analytical treatment of the stationary correlation function.

The question of the optimal learning rate (η) was not treated although it seems to be an important parameter in the convergence of the training process. Another question which deserves further research regards the analytical derivation of the correlation function. Taking into account the nonlinearity of the transfer function is necessary to close the naive calculation self-consistently, and to obtain the corrections to the correlation function.

ACKNOWLEDGMENTS

We thank W. Kinzel for fruitful discussions and a critical reading of the manuscript. I.K. acknowledges the support of the Israel Ministry of Science.

[1] *Fractals in Science*, edited by A. Bunde and S. Havlin (Springer-Verlag, Berlin, 1994).
[2] I. Kanter and D.A. Kessler, Phys. Rev. Lett. **74**, 4559 (1995).
[3] P. Riegler and M. Biehl, J. Phys. A **28**, L507 (1995).
[4] M. Opper and W. Kinzel, in *Physics of Neural Networks* (Springer-Verlag, Berlin, 1995).
[5] W. Tarkowski and M. Lewenstein, J. Phys. A **26**, 3669 (1993).
[6] I. Kanter, D.A. Kessler, A. Priel, and E. Eisenstein, Phys. Rev. Lett. **75**, 2614 (1995).
[7] M. Schroder and W. Kinzel, J. Phys. A **31**, 2967 (1998).
[8] L. Ein-Dor and I. Kanter, Phys. Rev. E **57**, 6564 (1998).
[9] A. Priel and I. Kanter, Phys. Rev. E **59**, 3368 (1999).
[10] H.A. Makse, S. Havlin, M. Schwartz, and H.E. Stanley, Phys. Rev. E **53**, 5445 (1996).
[11] I. Kanter, Phys. Rev. Lett. **77**, 4844 (1996).
[12] A. Priel, Ph.D. thesis, Bar-Ilan University, 1999 (unpublished).