Generation of unpredictable time series by a neural network

Richard Metzler and Wolfgang Kinzel
Institut für Theoretische Physik, Universität Würzburg, Am Hubland, D-97074 Würzburg, Germany

Liat Ein-Dor and Ido Kanter
Minerva Center and Department of Physics, Bar-Ilan University, Ramat Gan, 52900 Israel

(Received 17 November 2000; revised manuscript received 12 January 2001; published 26 April 2001)

A perceptron that ''learns'' the opposite of its own output is used to generate a time series. We analyze properties of the weight vector and the generated sequence, such as the cycle length and the probability distribution of generated sequences. A remarkable suppression of the autocorrelation function is explained, and connections to the Bernasconi model are discussed. If a continuous transfer function is used, the system displays chaotic and intermittent behavior, with the product of the learning rate and amplification as a control parameter.

DOI: 10.1103/PhysRevE.63.056126 PACS number(s): 84.35.+i, 05.45.Tp

For every prediction algorithm that maps a binary time series onto a binary output, there is a sequence for which it gives 100% wrong predictions [1]. This sequence can be constructed easily by having the algorithm predict the next bit in the time series and continuing the series with the opposite of the prediction. Of course, this sequence will only make one given algorithm with one given set of initial parameters fail completely. However, it is still interesting to compare the properties of such an antipredictable sequence with one that can be predicted with good success by the same algorithm. We will study the statistical properties of the time series generated by one particular prediction machine, namely, a perceptron using the Hebb learning rule.

The perceptron is the simplest type of feed-forward neural network [2]. It consists of N input units that are connected to one output unit by N synaptic weights w_i, $i=1,\ldots,N$. An input vector $\mathbf{x}=(x_1,\ldots,x_N)$ is mapped onto an output σ by a sigmoidal function of the scalar product of $\mathbf{x}$ and $\mathbf{w}$: $\sigma=f(\Sigma_i^N x_i w_i)$, where $f(x)=\mathrm{sgn}(x)$ is used for the so-called simple perceptron, and the error functon $f(x)=\mathrm{erf}(\beta x)$ or the hyberbolic tangent $\tanh(\beta x)$ with an adjustable amplification β are common choices for the continuous perceptron. In Sec. I a sequence generated by a simple perceptron that ''learns'' the opposite of its own output is examined, whereas in Sec. II a continuous perceptron is used; the differences between the two cases are highlighted.

I. CONFUSED BIT GENERATOR

Perceptrons have been used for generating binary time series in a simple iteration that was named bit generator (BG) [3–5]: the pattern $\mathbf{x}^t$ at time t is an N-bit window of a binary time series $\mathbf{S}$, $\mathbf{x}^t=(S^t,\ldots,S^{t-N+1})$, $S^t\in\{-1,1\}$. The series is generated by the output of the perceptron: $S^{t+1}=\mathrm{sgn}(\mathbf{x}^t\cdot\mathbf{w})$. For a fixed $\mathbf{w}$, the sequence relaxes into a limit cycle whose average length increases more slowly than exponentially with N. Short cycles with a length $l<2N$ are more likely than longer ones, and the Fourier spectrum of the sequence is dominated by one frequency which is also prominent in the weights [3,4]. The cycles can be calculated analytically if the weights have only one Fourier component [5].

In Ref. [1], a variation of the BG was introduced in which the next bit of the sequence is the opposite of the perceptron's output, and the network learns the sequence according to the Hebb rule, with a learning rate η:

$$S^{t+1}=-\mathrm{sgn}(\mathbf{x}^t\cdot\mathbf{w}^t); \tag{1}$$

$$\mathbf{w}^{t+1}=\mathbf{w}^t+(\eta/N)S^{t+1}\mathbf{x}^t. \tag{2}$$

We call this system confused bit generator (CBG), because the perceptron is told that its output was wrong no matter what it predicted.

A. Dynamics of the weights

Geometrically, $\mathbf{w}$ makes a directed random walk on an N-dimensional cubic lattice: each component of the learning step is $\pm\eta/N$. Thus, while the values of the weight components w_i are real numbers, they can only take discrete values $w_i^0\pm n\eta/N$, with $n=0,1,2,\ldots$ once the initial values w_i^0 are chosen.

Furthermore, each learning step has a negative overlap with the current $\mathbf{w}$, which prevents a boundless growth of the vector. The norm of the weight vector fluctuates around an equilibrium value that can be estimated by replacing $\mathbf{x}$ with a random vector whose components have a variance of 1, taking the square of Eq. (2) and applying the usual formalism for online learning [6]:

$$\langle\mathbf{w}^{t+1}\cdot\mathbf{w}^{t+1}-\mathbf{w}^t\cdot\mathbf{w}^t\rangle=-\frac{2\eta}{N}\langle\mathbf{x}^t\cdot\mathbf{w}^t\mathrm{sgn}(\mathbf{x}^t\cdot\mathbf{w}^t)\rangle+\frac{\eta^2}{N^2}\langle\mathbf{x}^t\cdot\mathbf{x}^t\rangle. \tag{3}$$

Introducing a time scale α with $d\alpha=1/N$ and averaging over $\mathbf{x}$, this becomes a deterministic differential equation for the norm w of $\mathbf{w}$ in the thermodynamic limit $N\to\infty$:

$$\frac{dw}{d\alpha} = -\sqrt{\frac{2}{\pi}}\eta + \frac{\eta^2}{2w}. \tag{4}$$

The attractive fixed point of this equation is $w = \sqrt{\pi/8}\,\eta \cong 0.6267\,\eta$. However, using the time series generated by the perceptron as patterns, simulations give a slightly different value of $w \approx 0.566\,\eta$, independent of N (this was already observed in Ref. [1]). Two possible violations of the assumptions for which the analysis in Ref. [7] guarantees agreement with analytical predictions must be considered: first, the time series patterns generated by the CBG do now follow a uniform distribution (see Sec. I E). Second, they are not drawn independently from the weight vector and previous patterns. Simulations in which a perceptron was given patterns drawn randomly from a distribution as described in Sec. I E yield a norm w that is compatible with the analytical value of $0.6267\,\eta$. This indicates that temporal correlations are responsible for the deviations. The learning rate η only sets a length scale, but does not influence the long-term behavior of the system.

In a similar fashion, the autocorrelation of the weight vector can be calculated using the assumption of random patterns:

$$\langle \mathbf{w}^t \cdot \mathbf{w}^{t+\tau} \rangle = w^2 \exp\left(-\frac{4}{\pi}\frac{\tau}{N} \right). \tag{5}$$

In some cases (see Sec. I C), it is useful to assign an individual learning rate η_i to each weight component w_i. A short calculation shows that the mean square norm of each weight component is proportional to its learning rate:

$$\langle w_i^2 \rangle = \sqrt{\frac{\pi}{8}}\frac{\eta_i}{N}\sqrt{\sum_j w_j^2}. \tag{6}$$

A component with a higher learning rate thus has a stronger influence on the output. This also leads to a faster decay of the autocorrelation:

$$\langle w_i^t w_i^{t+\tau} \rangle = \frac{\sum_j \eta_j}{\eta_i}\frac{\pi}{4}\exp\left(-\frac{\eta_i}{\sum_j \eta_j}\frac{4}{\pi}\tau \right). \tag{7}$$

The dynamics of the weights can be linked to the the autocorrelation function C_j^t of the sequence, defined by

$$C_j^t = \sum_{i=1}^{t} S^i S^{i-j}, \tag{8}$$

where t is the number of patterns summed over. Simply add t update steps according to Eq. (2):

$$w_j^t = w_j^0 + \sum_{i=1}^{t} (\eta/N) S^i S^{i-j} = w_j^0 + (\eta/N) C_j^t. \tag{9}$$

Each value C_j^t for $1 \leqslant j \leqslant N$ corresponds to the distance of the weight vector from its starting point along one axis in the N-dimensional weight space, measured in units of η/N. This point is important, and will be exploited in the following paragraphs.

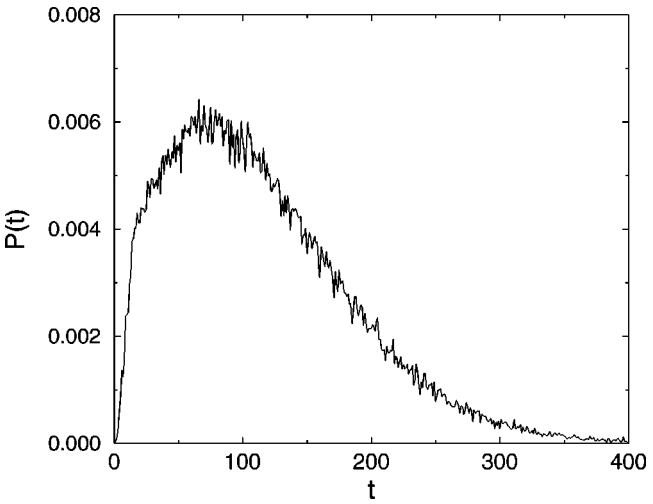

FIG. 1. Distribution of transient lengths t of a CBG with $N=8$. The small probability of short t indicates that only a small fraction of state space is part of a cycle.

B. Cycles and transients

The CBG is a deterministic map with a discrete, finite state space. This means that it eventually falls into a cycle of some length l: both sequence and weights repeat after l steps, i.e., $\mathbf{w}^t = \mathbf{w}^{t+l}$, or alternatively $C_j^l = 0$ for $1 \leqslant j \leqslant N$ after l steps. This means that l must be divisible by 4, since only sequences with $l \mod 4 = 0$ can have an autocorrrelation of 0. Also, a lower bound for l can be given: for the lth autocorrelation value, one obtains $C_l^l = l \neq 0$; therefore, $l > N$. By renaming indices, one finds $C_j^l = C_{l-j}^l$ for periodic sequences. If $l \leqslant 2N$, one thus obtains $C_j^l \equiv 0$ for all $j < l$. In Ref. [8], it was conjectured that such a sequence does not exist except for any l except $l=4$. If this is true, $l > 2N$ must hold for $N > 3$.

An upper bound on the cycle length can be found by estimating how many states in weight space the weight vector can take. Assuming that it stays inside an N-dimensional hypersphere of radius $w_f = 0.566\,\eta$ and volume $V = w_f^N \pi^{N/2}/\Gamma(N/2+1)$, we can divide that volume by the volume of a unit cell, $(\eta/N)^N$, and expand using Stirling's equation. We find that the number of possible states in weight space scales approximately like $5.45^N/\sqrt{N}$. Combining this with 2^N possible sequences gives $10.9^N/\sqrt{N}$ possible states of the system.

Simulations show that not all of these states are part of a cycle: starting from random initial conditions, there is a transient whose median length scales approximately like 2.04^N. The transient distribution (Fig. 1) shows that not all states have the same probability of being part of a cycle: the probability for a very short transient is smaller than that for a longer one, which implies that some sort of annealing occurs during the first steps. Simulations were done with random

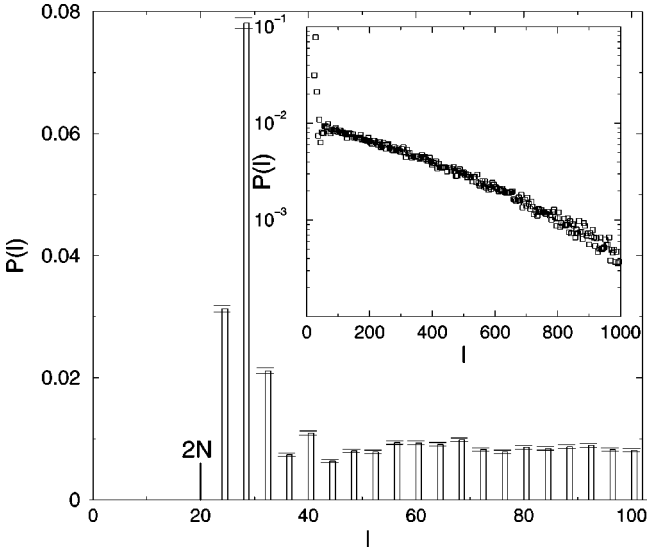

FIG. 2. Distribution of period lengths of a CBG with $N=10$ on linear and logarithmic (inset) scales. The initial weights and sequence were random. All l's are divisible by 4, and $l>2N$. Error bars denote the standard error.

initial sequences and random initial vectors normalized to $w=0.566\eta$.

The distribution of cycle lengths l found in simulations shows the expected features (see Fig. 2): a minimum cycle length $l>2N$ and no cycle lengths l that are not divisible by 4. There is a distinct maximum near the minimum cycle length and a broad distribution that falls off slightly faster than exponentially for large l. The average of l scales approximately like 2.2^N, as seen in Fig. 3. The fact that the largest l that is found scales exponentially with N suggests that there is an exponential number of different cycles.

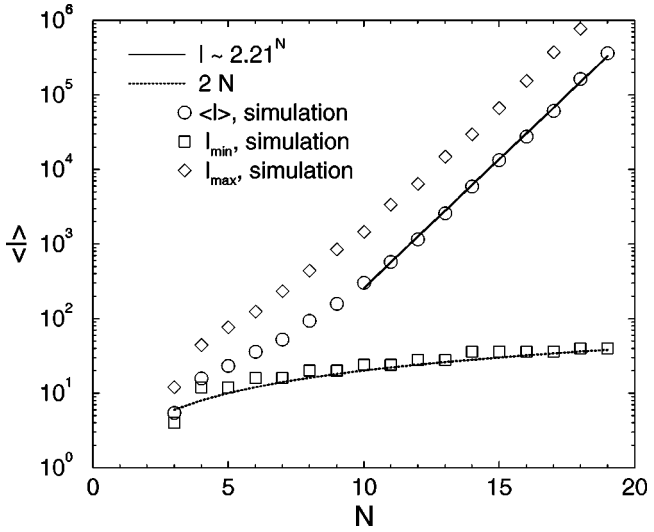

FIG. 3. Average, smallest, and largest cycles l found in simulations with 1000 random initial conditions for each value of N. The full line is an exponential fit to the data for $N>9$, and the dotted line denotes the theoretical lower bound of $l>2N$ for $N>3$.

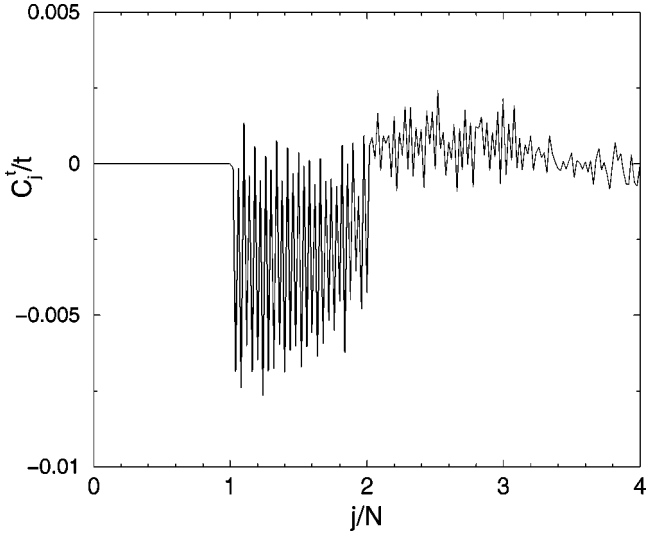

FIG. 4. Autocorrelation function C_j^t/t of a CBG with $N=50$, averaged over $t=2\times10^6$ patterns.

C. Autocorrelation function and the Bernasconi model

The autocorrelation of the sequence shows some peculiarities, as seen in Fig. 4: As explained, the first j values correspond to components of $\mathbf{w}$. Since w is finite, C_j^t is bounded for $1\leqslant j\leqslant N$, i.e., it does not grow like $\sqrt{t}$ as it would for a random sequence. The values for $N<j\leqslant 2N$ show negative correlations that grow linearly with t for even j, whereas they are compatible with a random sequence for odd j. Between $2N$ and $3N$, correlations are positive for even j and 0 for odd j. These effects appear for all N in both the transient and the cycle, as long as the cycle length is much larger than N.

Bit series with low autocorrelations are of interest in mathematics, and have applications in signal processing [9]. It is therefore interesting to know whether the CBG generates sequences with autocorrelations significantly lower than for random series. Two measures are commonly used in the literature: for periodic sequences of length l, an energy function (which is studied in the so-called Bernasconi model for periodic boundary conditions [10]) can be defined by

$$H_p=\sum_{j=1}^{l-1}(C_j^l)^2=\sum_{j=1}^{l-1}\left(\sum_{i=1}^{l}S^iS^{i+j}\right)^2. \tag{10}$$

Results on the ground states of this Hamiltonian can be found in Ref. [8]. By trial and error, initial conditions for the CBG can be found which yield cycles slightly larger than $2N$, for which all value of C_j^l except one are 0. However, even for the best sequences we found, H_p was larger than the known ground state energies by at least a factor of 2.

The original model does not use periodic boundary conditions: in a sequence of length p, only the sum over $p-j$ different terms with a lag of j can be calculated. The energy is therefore given by

$$H_{ap}=\sum_{j=1}^{p-1}(C_j^{p-j})^2 \tag{11}$$

(note the summation limits). The so-called merit factor F introduced by Golay [11] is defined by

$$F=\frac{p^2}{2H_{ap}}. \tag{12}$$

A merit factor of 1 is expected for a random sequence; lower autocorrelations yield higher F. The theoretical limit for large p is conjectured to be about $F=12$ [10], whereas optimization routines typically find sequences with $5<F<9$ (see Ref. [12], and references therein) and exact enumeration for small p suggests $\lim_{p\to\infty}F=9.3$ for the optimal sequence [13].

To estimate the merit factor of sequences generated by the CBG analytically, we solve Eq. (9) for C_j^{t2}, and use the autocorrelation of the weights given by Eq. (5):

$$\begin{aligned}\langle (C_j^{p-j})^2\rangle &= \frac{N^2}{\eta^2}\langle w_j^{02}+w_j^{t2}-2w_j^t w_j^0\rangle \\ &= \frac{\pi}{4}N\left[1-\exp\left(-\frac{4}{\pi}\frac{p-j}{N}\right)\right].\end{aligned} \tag{13}$$

The energy can be expressed as a sum or approximated by an integral in continuous variables $\alpha=p/N$ and $\beta=j/N$. Since Eq. (13) only holds for $1\leqslant j\leqslant N$, $C_j^{p-j^2}=p-j$ must be used for $j>N$. We obtain the expression

$$\begin{aligned}H_{ap} &= \sum_{j=1}^{p-1} N\frac{\pi}{4}\left[1-\exp\left(-\frac{4}{\pi}\frac{p-j}{N}\right)\right] \\ &\approx \int_0^{\alpha} N^2\frac{\pi}{4}\{1-\exp[-(4/\pi)(\alpha-\beta)]\}d\beta \\ &= N^2\frac{\pi}{4}\left\{\alpha-\frac{\pi}{4}\left[1-\exp\left(-\frac{4}{\pi}\alpha\right)\right]\right\} \quad \text{for } j\leqslant N\end{aligned} \tag{14}$$

and

$$\begin{aligned}H_{ap} = N^2\Bigg(&\frac{\pi}{4}\left\{1-\frac{\pi}{4}\left[\exp\left(\frac{4}{\pi}(1-\alpha)\right)-\exp\left(-\frac{4}{\pi}\alpha\right)\right]\right\} \\ &+\frac{1}{2}(\alpha-1)^2\Bigg) \quad \text{for } j>N.\end{aligned} \tag{15}$$

The corresponding merit factor is compared to simulations in Fig. 5: Eqs. (14) and (15) give qualitatively correct results, but differ from the observed values by roughly 10%. The feedback mechanisms of the CBG cause a faster decay of C_j^t than predicted for random patterns.

We observed in Sec. I A that individual learning rates can make C_j^t decay faster for some j and slower for others. The search for a minimal H_{ap} can be written as an optimization problem in the continuous function $\eta(\beta)$, where $\eta_j=\eta(j/N)$. Solving this problem with a variational approach, one finds that it is sensible to give the last 41% of the weights a learning rate and norm of zero, and increase the learning rate continuously toward components with smaller indices. Unfortunately, even this optimization does not improve the merit factor beyond $F=1.74$ in theory and $\langle F\rangle=1.86$ in simulations. This is still a lot worse than the results of other optimization methods [10,12], so the CBG is not a competitive generator of low-autocorrelation sequences. Nevertheless, it has some interesting possibilities:

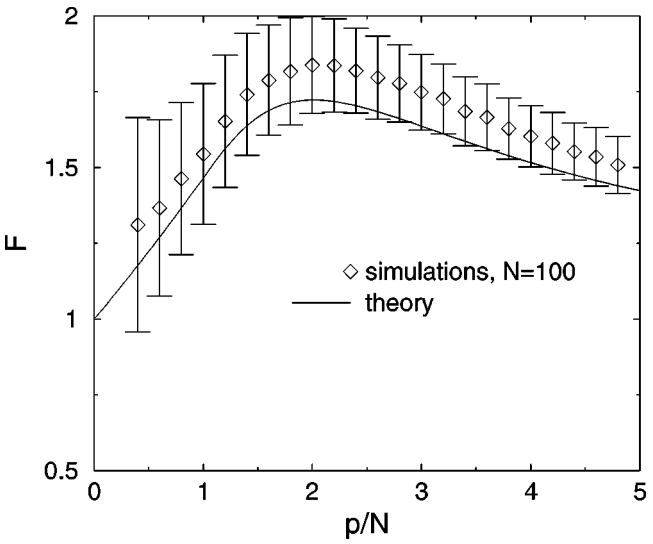

FIG. 5. Merit factor F as a function of scaled sequence length p/N, compared to Eq. (15). Error bars denote the standard deviation of F for $N=100$, not the standard error.

D. Shaping the autocorrelation function

Being able to suppress autocorrelations, the CBG is also capable of controlling the shape of the autocorrelation function in the long-time limit. Using Eqs. (6) and (9) in the limit where $\mathbf{w}^0\to 0$ and for non-negative learning rates, one can obtain the inverse relation between the square of the autocorrelation function $(C_j^t)^2$ and the corresponding learning rate η_j:

$$\langle |C_j^t|\rangle = \sqrt{\frac{\pi}{8}}\sqrt{\frac{\sum_{i=1}^{N}\eta_i}{\eta_j}}. \tag{16}$$

Thus any desired shape of the autocorrelation function is achievable by using the appropriate profile for η_j which can be extracted from Eq. (16). The high performance of the CBG as a producer of sequences with specific desired shapes of the autocorrelation function is observed in simulations. This feature of the CBG is demonstrated in Figs. 6 and 7, where both exponential and polynomial profiles of the autocorrelations are successfully generated. The slight deviations from the target profile are probably due to a violation of the assumption of random patterns. Simulations are done for a CBG with 30 input units. The autocorrelations are calculated for time windows of 100 000 bits, and are averaged over 1000 such successive windows. Checking a wide variety of shapes, the CBG exhibits a decent capability of achieving the

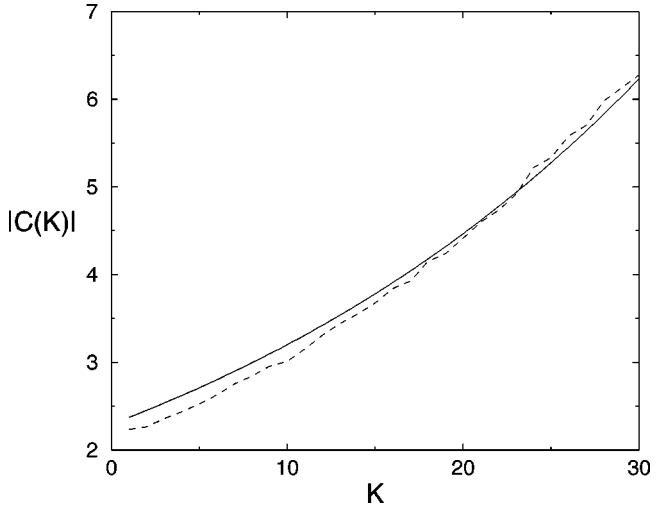

FIG. 6. Profile of the average absolute value of the autocorrelation function $|C_j|$ which was achieved by a CBG with $N=30$ input units using $\eta_j = \exp(-2j/N)$ (dashed curve). The solid curve stands for the desired profile, $|C_j| = A\exp(j/N)$, where $A = \sqrt{\pi/8}\sqrt{(e^{-2}-1)/(e^{-2/N}-1)}$.

expected profiles. It could be used as an alternative mechanism for generating colored binary sequences using local rules instead of nonlocal mechanisms auch as Fourier transforms.

The limited use of the CBG in generating sequences with a high merit factor may be related to phase space arguments: as seen in Sec. I B; the CBG can still generate exponentially many different time series depending on initial conditions, whereas there are very few sequences with the highest achievable high merit factors (see Ref. [14] for the density of states with cyclic boundary conditions). The mechanism of the CBG allows for manipulation of the autocorrelation function only if the constraints on the desired sequence are not too strong, such as suppressing all of the elements of C_j on a short time scale. On the other hand, choosing a given shape for long time averages of C_j still allows for many realizations of the sequence.

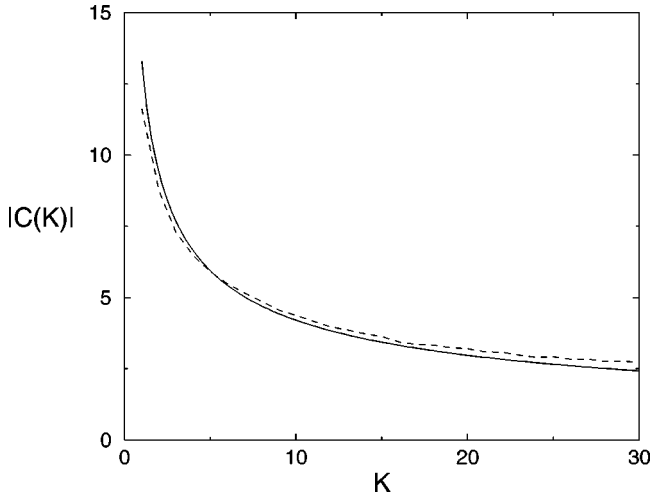

FIG. 7. Profile of the average absolute value of the autocorrelation function $|C_j|$ which was achieved by a CBG with $N=30$ input units using $\eta_j = j/N$ (dashed curve). The solid curve stands for the desired profile, $|C_j| = 2.427\sqrt{j/N}$.

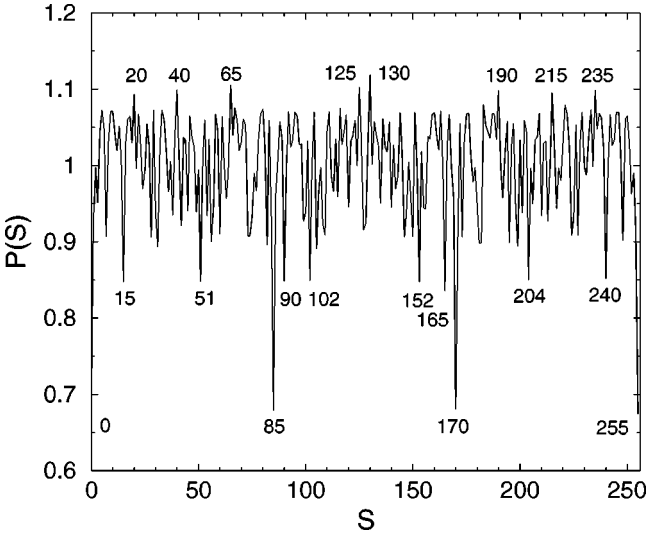

FIG. 8. Histogram of eight-bit subsequences generated by a CBG with $N=50$, averaged over 5×10^6 steps. The y axis gives the probability of a subsequence multiplied by 2^8 to give an average of 1.

E. Distribution of generated sequences

The structure in C_j shows that the CBG does not generate a random sequence. This becomes more obvious in a histogram of subsequences generated by the system. Figure 8 shows the probability distribution of eight-bit substrings from a run of a CBG with $N=50$, encoded as decimal integers. Some strings are strongly suppressed, most notably 0 (binary 00000000), 85 (01010101), 170 (10101010), and 255 (11111111). Other sequences with a below-average likelihood also correspond to ''simple'' sequences, like 15 (00001111) and 51 (00110011). Continued simple sequences give high values of some components of the autocorrelation function, which is unlikely as explained above.

The shape of the histogram is the same for all N in both the transient and the cycle. However, l must be much larger than the number of bins in the histogram. The amplitude of the deviations from uniform distribution again goes like $1/N$.

Ordering histograms in descending rank order often reveals insights into the underlying processes and phase space structure (see, e.g., Refs. [15,16]). In our case, the rank ordered histogram does not show a power law or other universal behavior, as seen in Fig. 9.

One way to explain this histogram is by finding the stationary distribution for a biased random walk on a DeBruijn graph, as done in Refs. [16,17]: a subsequence S is followed by 1 with probability p_S, and by 0 with probability $1-p_S$. It is possible to reproduce the histogram of the CBG accurately this way; however, one has to take the transition probabilities p_S for each subsequence from simulations of the CBG. There is no obvious way of calculating these analytically, and taking random transition probabilities does not reproduce the shape of Fig. 9.

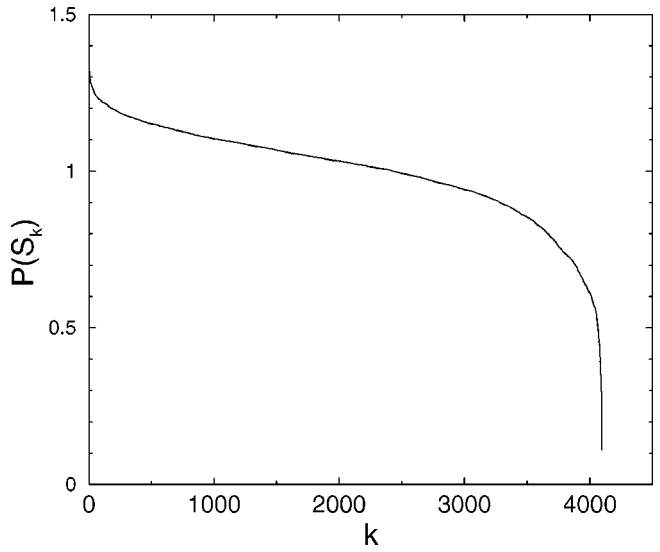

FIG. 9. Frequencies of 12-bit subsequences generated by a CBG with $N=50$, ordered by rank k and normalized to an average of 1 (solid line). The frequencies reproduced by a Markov process are also displayed (dotted line), but are indistinguishable from the first curve.

The CBG may be considered the simplest case of a sequence-generating perceptron that deterministically changes its direction. The sequence generated by it, while complex, has many properties that can be understood at least qualitatively, especially those that can be linked to the autocorrelation function. It is not by any standard a satisfying random bit sequence, and while results derived from the assumption of random patterns are usually qualitatively correct, the exact values have to be modified.

II. CONFUSED SEQUENCE GENERATOR

The simplest generalization of the CBG to a continuous perceptron replaces the sign function in Eq. (2) by a continuous sigmoidal function,

$$S^{t+1}=-\mathrm{erf}\left(\beta\sum_{j=1}^{N} w_j S_{t-j+1}\right)=-\mathrm{erf}(\beta\mathbf{w}^t\cdot\mathbf{x}^t)=-\mathrm{erf}(\beta h^t), \tag{17}$$

$$\mathbf{w}^{t+1}=\mathbf{w}^t+\frac{\eta}{N}S^{t+1}\mathbf{x}^t, \tag{18}$$

where the only new quantity is the amplification β, and h^t is an abbreviation of $\mathbf{w}^t\cdot\mathbf{x}^t$. The generation of a time series by a continuous perceptron with fixed weights was studied in a number of publications [18–20], in which the system was called the sequence generator (SGen). We will call the mapping defined by Eqs. (17) and (18) a confused sequence generator (CSG).

The SGen with fixed weights has a critical amplification β_c that depends on $\mathbf{w}$, below which $S=0$ is the attractive fixed point. Above β_c, the zero solution becomes repulsive, and the SGen generates a periodic or quasiperiodic time series with an attractor dimension of 0 or 1 for most choices of $\mathbf{w}$ and β [18]. This attractor is robust to noise [19]. The time series displays chaotic behavior only for very special choices of $\mathbf{w}$ and β (''fragile chaos'') if the transfer function is monotonic, and for generic initial conditions (''robust chaos'') only if it is nonmonotonic [20]. We will compare these properties to those of the CSG.

A. Mean-field solution for w

Similar to the CBG, the weight vector of the CSG preforms a directed random walk near the surface of a hypersphere of radius w. Unlike the CBG, the length of the learning steps depends on the magnitude of the output, which in turn depends on w and the outputs in previous time steps. To find an approximate solution to this self-consistency problem, we will first ignore correlations between patterns and weights, and treat the patterns as random and independent. In this approach, the inner field h is a Gaussian random variable of mean 0 and variance w^2S^2, where $S^2=\langle S^{t2}\rangle_t$ is the mean square output of the system.

The norm w is found by taking the square of Eq. (18),

$$w^{t+1^2}=w^{t^2}-\frac{2\eta}{N}\mathbf{w}^t\cdot\mathbf{x}^t\mathrm{erf}(\beta\mathbf{x}^t\cdot\mathbf{w}^t)+\frac{\eta^2}{N^2}S^{t^2}\mathbf{x}^t\cdot\mathbf{x}^t, \tag{19}$$

and averaging over the input patterns. The self-overlap $\mathbf{x}\cdot\mathbf{x}$ is on the average NS^2, so the fixed point of w is given by

$$2\langle h\ \ \mathrm{erf}(\beta h)\rangle=\eta S^4. \tag{20}$$

The average on the left hand side can be evaluated, and leads to

$$\frac{4}{\sqrt{\pi}}\frac{\beta w^2S^2}{\sqrt{1+2\beta^2w^2S^2}}=\eta S^4, \tag{21}$$

or

$$w^2=\frac{\pi\beta\eta^2S^6+\eta S^2\sqrt{\pi}\sqrt{16+\pi\beta^2\eta^2S^8}}{16\beta}. \tag{22}$$

Let us now turn to S^2. The probability distribution of S itself is rather awkward, since it involves inverse error functions, and its slope diverges at $S=\pm1$. However, S^2 can be easily calculated by using the distribution of h:

$$\langle S^2\rangle=\int_{-\infty}^{\infty}\mathrm{erf}^2(\beta h)(2\pi w^2S^2)^{-1/2}\exp\left(-\frac{h^2}{2w^2S^2}\right)dh$$

$$=\frac{2}{\pi}\arcsin\left(\frac{2\beta^2w^2S^2}{1+2\beta^2w^2S^2}\right). \tag{23}$$

Plugging $w^2(\eta,\beta,S^2)$ from Eq. (22) into (23), and solving numerically, one obtains a self-consistent solution for S^2. A closer look at the equations reveals that if a new quantity $\gamma=\eta\beta$ is introduced, only γ enters into the equation for S^2, and w^2 is of the form $w^2=\eta^2\hat{w}^2(\gamma)$, so only one curve must be considered. This is intuitive, since a higher η eventually

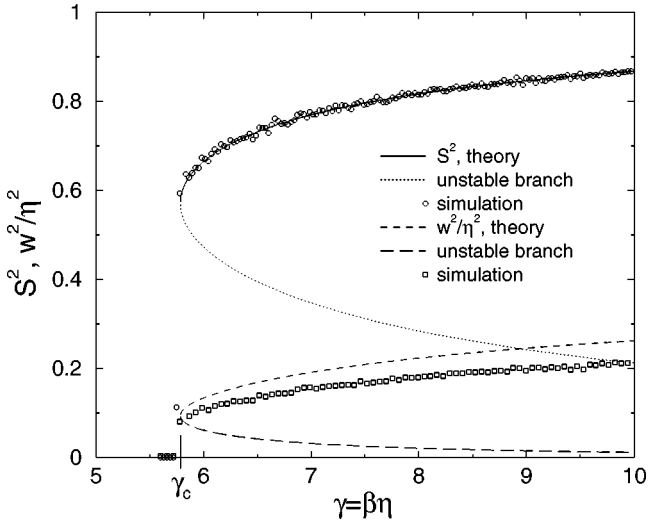

FIG. 10. Mean-field solution of the CBG, compared to simulations with $N=400$.

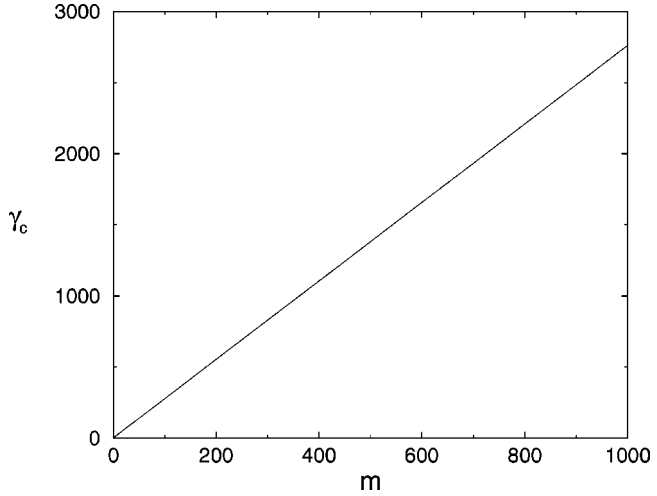

FIG. 11. The linear relation between γ_c and m derived from the numerical solution of Eqs. (26) and (27).

leads to a higher w, which has the same effect on S^2 as having a smaller w, but multiplying $\mathbf{w}\cdot\mathbf{x}$ by a higher factor β.

The map defined by Eqs. (17) and (18) always has a trivial solution $S=0$. Only for a sufficiently high $\gamma>\gamma_c$ are the outputs high enough to sustain a nonvanishing solution. Note that $S=0$ is always an attractive solution for all $\gamma<\infty$, but its basin of attraction becomes smaller for larger γ.

The numerical solution of Eqs. (22) and (23) shows that the system undergoes a saddle-node bifurcation at $\gamma_c \doteq 5.785$, which is in good agreement with simulations. Above γ_c, two new fixed points exist, only one of which is stable. While for $S^2(\gamma)$ excellent agreement is found between theory and simulation (see Fig. 10), $w^2(\gamma)$ shows quantitative differences which are caused by correlations between $\mathbf{x}$ and $\mathbf{w}$: the mean square overlap $\langle(\mathbf{x}\cdot\mathbf{w})^2\rangle$ turns out to be $(1.22\pm0.01)w^2S^2$ instead of w^2S^2, as expected for random patterns. This causes a factor of roughly 0.82 between the theoretical and observed value of w^2 seen in Fig. 10). The same factor is found in the CBG. For large γ, S^2 goes to 1 (as it should, since the system is identical to the CBG if $\gamma=\infty$), and the theoretical prediction for w goes to $\sqrt{\pi/8}\eta$, just like in the CBG.

B. CSG of the *m*th degree—CSG*m*

Multispin interactions were studied in fields like neural networks [21], low-autocorrelated sequences [10] and error-correcting codes [22]. The idea to include multispin interactions in our work originated as an attempt to improve the suppression of the autocorrelation function achieved by the CBG. The existence of four-spin interactions in the Bernasconi model implies that a CBG with multispin interaction might be useful in the construction of low-autocorrelated sequences. However, it turns out that a CBG with multispin interactions suppresses the corresponding multispin correlations instead.

In this section we apply multispin interactions to the CSG, and define the CSGm, namely, a CSG in which each weight component w_j connects the corresponding input unit x_j not only to the output but also to other $m-2$ additional input units.

Assigning $A_{j,i}$ $i=1,..,m-2$ to be the labels of the $m-2$ additional input units participating in the jth interaction, the dynamics of a CSGm is given by

$$S^{t+1}=-\mathrm{erf}\left(\beta\sum_{j=1}^{N} w_j S^{t+1-j}\prod_{i=1}^{m-2} S^{t+1-A_{j,i}}\right), \tag{24}$$

$$w_j^{t+1}=w_j^t+\frac{\eta}{N}S^{t+1}S^{t+1-j}\prod_{i=1}^{m-2} S^{t+1-A_{j,i}}. \tag{25}$$

A similar calculation under the same assumptions used to yield the solution of the original CSG gives the following general set of equations:

$$w^2=\frac{\pi\beta\eta^2S^{2(m+1)}+\eta S^2\sqrt{\pi}\sqrt{16+\pi\beta^2\eta^2S^{4m}}}{16\beta}, \tag{26}$$

$$\langle S^2\rangle=\frac{2}{\pi}\arcsin\left(\frac{2\beta^2w^2S^{2(m-1)}}{1+2\beta^2w^2S^{2(m-1)}}\right). \tag{27}$$

Numerically solving Eqs. (26) and (27) for a large range of m values, both the bifurcation point γ_c and the first nonzero values of S^2 and w^2 were found to increase with m. For $m\to\infty$, one can easily show that $\gamma_c\to\infty$, while for the nonvanishing solution $w^2\to1$ and $S^2\to1$. Aiming to study the asymptotic behavior of S^2 and γ_c in the large m limit, we set $S^2=1-\epsilon$, and find that ϵ must decay to zero at least as $1/m$ in order for a nonzero solution to exist. This inverse relation between ϵ and m derives S^m terms, since $(1-\epsilon)^m\to0$ unless $\epsilon<1/m$. Inserting $S^2=1-\epsilon$ into Eqs. (26) and (27), and expanding the resulting expression to a power series in ϵ, the inverse relation between ϵ and m leads to linear increment of γ_c as a function of m. The numerical solutions of the system in the large-m regime support the linear behavior of γ_c as derived from the aforementioned analysis (Fig. 11).

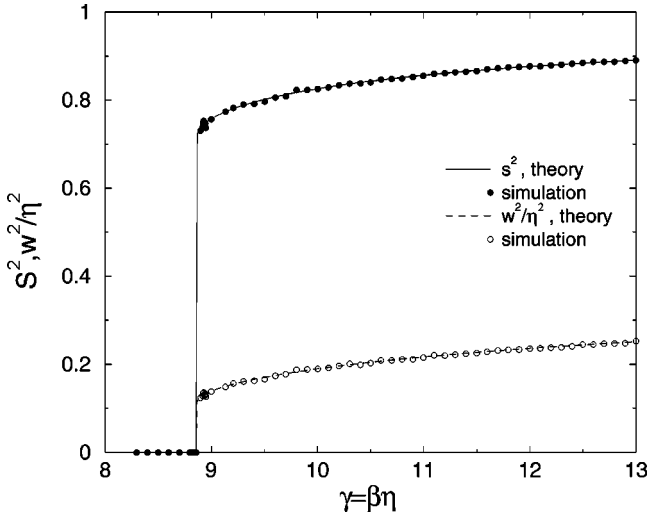

FIG. 12. Mean-field solution of the CSG3, compared to simulations with $N=2000$.

Figure 12 describes the numerical solution with respect to the simulation results for a system with $m=3$. This harmony between analytics and simulations is observed for larger m as well.

C. Autocorrelation function

Relation (9), that links the autocorrelation function to the weights, still holds for the CSG. Since the weight vector is bounded in the CSG as well, the same argument can be given for the first suppression of the first N values of the autocorrelation function. Correspondingly, C_j^p/p is almost indistinguishable from that of the CBG shown in Fig. 4.

D. Cycles and attractors

The CSG can be seen as a nonlinear mapping that maps the vector $\mathbf{x}^t \oplus \mathbf{w}^t$ onto $\mathbf{x}^{t+1} \oplus \mathbf{w}^{t+1}$. This is in contrast to previous work on the SGen [22], where the weights were fixed and could be considered parameters of the model rather than dynamic variables. The only real control parameter of this mapping is γ. Since both the sequence and the weights now live in a high-dimensional space of real numbers, the CBG can display a wide variety of behaviors, depending on N and γ:

For $\gamma < \gamma_c$, the zero solution is the only attractor, and the system will quickly reach $\mathbf{x}^t=0$ and stop developing. For γ slightly above γ_c, an irregular-looking time series with the statistical properties calculated in Sec. II A, and displayed in Fig. 10, is generated. However, the zero solution is still attractive, and after some time the system will drift close to it and stay there, i.e., the irregular behavior is due to a chaotic transient rather than a proper chaotic attractor.

The survival time on the transient increases dramatically with increasing N and γ. It is hard to decide from numerical results whether the average survival time $\langle t_s \rangle$ diverges with a power law ($\langle t_s \rangle \propto |\gamma - \gamma_d|^{-a}$), as one usually finds in scenarios where a chaotic transient becomes a chaotic attractor [23], or whether $\langle t_s \rangle$ increases exponentially with γ. In either case, the system shows chaotic behavior for sufficiently long times to obtain stable numerical results—for example, for $N=20$ and $\gamma=7.0$, the average survival time is of the order of 10^6 steps.

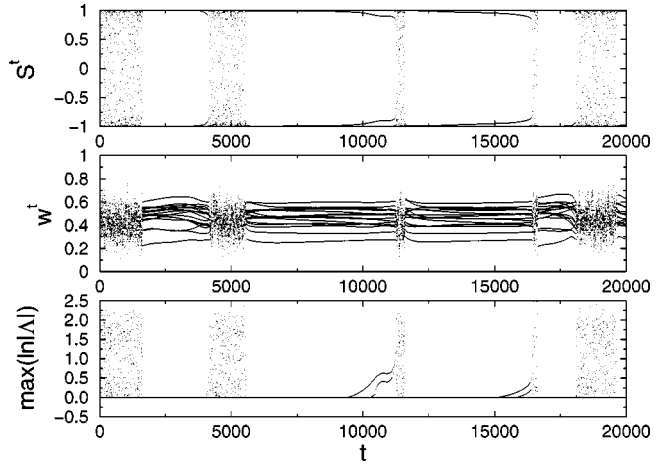

FIG. 13. Example of intermittent behavior for $N=6$ and $\gamma=8$. From top to bottom: output S^t, norm of weights w^t and largest ''one-step Lyapunov exponent'' $\max(\ln(|\Lambda|)$ (see Sec. II E).

If γ is larger than some critical value that depends on N, the chaotic transient can eventually end in a cycle that is related to a possible cycle of the discrete CBG. By ''related'' we mean that S^t in the CSG is very close to ± 1, and that clipping the sequence to the nearest value of ± 1 would give the equivalent attractor of the CBG. More different cycles become stable with higher γ; however, the cycle lengths are usually of order $2N$—short cycles are apparently more likely to become stable than cycles whose length is of order 2^N.

At amplifications γ slightly below the lowest γ for which the first cycle becomes stable for for a given N, intermittent behavior is observed: both S^t and w^t stay near a cycle for an extended number of steps (typically several thousand steps for $N=6$), before returning to chaotic behavior for a similar time. An example of this is given in Fig. 13.

E. Stability and Lyapunov exponents

The term ''chaotic'' was used in Sec. II D to describe the irregular time series generated by the CBG. We will now show that the system is in fact chaotic in the strict sense.

The sensitivity of trajectories of the map [Eqs. (17) and (18)] to small changes in the initial conditions can be tested by calculating the eigenvalues of the Jacobi matrix:

$$\mathbf{M}^t = \begin{pmatrix} \dfrac{\partial x_i^{t^1}}{\partial x_j^t} & \dfrac{\partial x_i^{t+1}}{\partial w_j^t} \\ \dfrac{\partial w_i^{t+1}}{\partial x_i^t} & \dfrac{\partial w_i^{t+1}}{\partial w_j^t} \end{pmatrix}. \tag{28}$$

This is to be understood as a $2N \times 2N$ matrix with indices i and j running from 1 to N. The entries of this matrix are of the following forms:

$$\frac{\partial x_1^{t+1}}{\partial x_j^t} = -\beta w_j^t \frac{2}{\sqrt{\pi}} \exp(-\beta^2 h^2),$$

$$\frac{\partial x_i^{t+1}}{\partial x_j^t} = \delta_{i-1,j} \quad \text{for } i=2,\ldots,N,$$

$$\frac{\partial x_1^{t+1}}{\partial w_j^t} = -\beta x_j^t \frac{2}{\sqrt{\pi}} \exp(-\beta^2 h^2), \tag{29}$$

$$\frac{\partial x_i^{t+1}}{\partial w_j^t} = 0 \quad \text{for } i=2,\ldots,N,$$

$$\frac{\partial w_i^{j+1}}{\partial x_j^{t+1}} = -\frac{\eta}{N} \mathrm{erf}(\beta h)\, \delta_{i,j} - \frac{\eta}{N} \beta w_j^t x_i^t \frac{2}{\sqrt{\pi}} \exp(-\beta^2 h^2),$$

$$\frac{\partial w_i^{t+1}}{\partial w_j^t} = \delta_{i,j} - \frac{\eta}{N} \beta x_j^t x_i^t \exp(-\beta^2 h^2).$$

If $|\beta h|$ is large and the transfer function is saturated, the exponential terms in Eq. (29) are negligible. In that case, the upper left section of $\mathbf{M}$ is occupied only on the first lower off-diagonal; the lower right section is the $N \times N$ unity matrix. Since the upper right section is identically 0, the lower left part does not enter into the calculation of the eigenvalues either.

This simplified matrix has N eigenvalues $\Lambda = 0$ and N eigenvalues $\Lambda = 1$. The eigenvectors of the latter span the space of weight vectors, where small changes to $\mathbf{w}^t$ are transferred unmodified to $\mathbf{w}^{t+1}$. The eigenvalues $\Lambda = 0$ all have the same eigenvector, whose only nonvanishing component is x_N, the component of the sequence vector that is rotated out at $t+1$. This means that the eigenvectors do not span the whole space, and that thus the eigenvalues are not a reliable measure of the propagation of a disturbance in the system.

If $|\beta h|$ is small enough for the exponential terms to have an appreciable effect, the effect on the eigenvalues is not easy to calculate. By using values of $\mathbf{x}$ and $\mathbf{w}$ taken from a run of the simulation, and numerically calculating the eigenvalues, we find that typically one of the $\Lambda = 0$ eigenvalues is changed drastically, and may have an absolute value $|\Lambda| > 1$. This corresponds to a strong susceptibility of the newly generated sequence component S_1 on small changes in $\mathbf{w}$ or $\mathbf{x}$. The other eigenvalues only undergo small corrections, corresponding to the feedback of the new component to the weights.

During the regular phases of intermittent behavior, the largest eigenvalues of the one-step matrix are significantly smaller than during the chaotic bursts (see Fig. 13)—corresponding to sequence values that are close to $S = \pm 1$, and thus a nearly saturated transfer function.

To find the Lyapunov exponents of the map (see, e.g., Ref. [24]), it is necessary to consider the development of a small perturbation over a long time, i.e., to calculate the eigenvalues Λ_i^T of $\Pi_{t=1}^T \mathbf{M}^t$ (of course, the trajectory is determined using the full nonlinear map). The Lyapunov exponents are then defined as

$$\lambda_i = \lim_{T\to\infty} (1/T) \ln |\Lambda_i^T|. \tag{30}$$

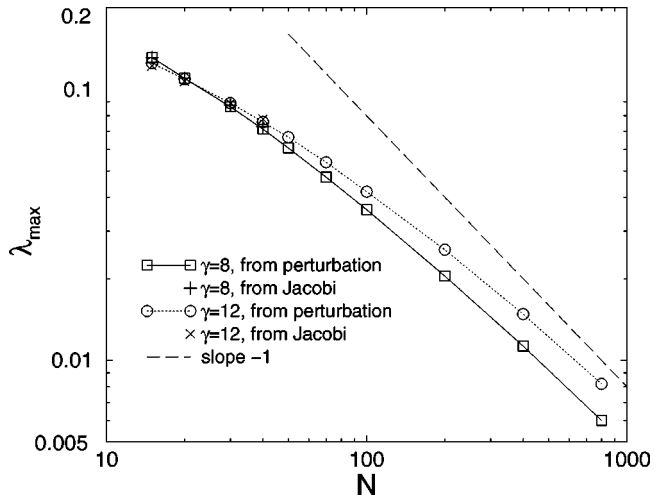

FIG. 14. Lyapunov exponents measured from the time development of perturbations, and from iteratively multiplying the Jacobi matrix [Eq. (30)], for $\gamma = 8$ and 12.

The straightforward calculation of the product of Jacobi matrices brings many numerical problems which can be eliminated by applying a Gram-Schmidt orthonormalization procedure to the columns of the product matrix in regular distances, as described in Ref. [25]. With this procedure, it is possible to average over $T > 100N$ and obtain numerically stable results. The largest Lyapunov exponent is displayed in Fig. 14. Typically, there are $N/2$ positive exponents.

The Kaplan-Yorke conjecture [26] states that there is a connection between the dimension D of a attractor of a map and the spectrum of Lyapunov exponents, which here are assumed to be ordered ($\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_{2N}$),

$$D_{KY} = k + \sum_{i=1}^{k} \lambda_i / |\lambda_{k+1}|, \tag{31}$$

where k is the value for which $\Sigma_{i=1}^k \lambda_i > 0$ and $\Sigma_{i=1}^{k+1} \lambda_i < 0$. Applying this to the spectrum of exponents derived from Eq. (30) gives an average attractor dimension between $1.1N$ and $1.2N$, depending slightly on γ.

An alternative method for measuring the largest Lyapunov exponent is to start two trajectories with infinitesimally different initial conditions, and propagate both of them using the nonlinear map. In regular intervals, we measure the distance between the trajectories, store it, and reset the distance to the initial value while keeping the direction of the distance vector. The advantage of this method is that it requires only $O(N)$ calculations per time step, rather than $O(N^2)$ as in the previous method, allowing one to go to a much higher value of N.

The results for λ_1 are also displayed in Fig. 14: the values gained by the two methods agree well within the numerical errors. For large N, λ_{max} decreases with $1/N$, i.e., perturbations grow on the α time scale of online learning.

III. SUMMARY

In this paper, we have studied the properties of a time sequence generated by a perceptron which learns the opposite of its own prediction. In the case of a simple perceptron, some properties are accessible analytically through the application of online learning techniques, and through the connection between the weights and the autocorrelation function of the sequence. The distribution of learning rates among the weight components has a decisive influence on the statistical properties of the generated sequence, and allows for sequences with a wide variety of autocorrelation shapes.

Due to the discrete nature of the sequence and the learning algorithm, cycles of the system are inevitable. We find that their typical length, as well as that of the transient, grow exponentially with the system size N.

A histogram of substrings of the generated sequence reveals that the sequence has significant deviations from randomness, although the deviations decrease with increasing N.

Replacing the sign function in the update rule by a continuous sigmoidal function changes many of these results. A vanishing solution now becomes possible; only for sufficiently large values of the rescaled amplification γ can nontrivial solutions survive. The critical γ_c can be calculated in a mean-field online learning calculation.

Since both sequence and weights are now continuous, cycles vanish for low values of γ, and the trajectory is a chaotic sequence. The largest Lyapunov exponent scales like $1/N$ for large N; the spectrum of Lyapunov exponents suggests high-dimensional chaos.

At least some cycles of the CBG reemerge as stable fixed points of the CGS above a critical γ that is different for each attractor. Slightly below the smallest critical γ for a given N, an intermittent behavior is observed.

Compared to the behavior of sequence-generating perceptrons with fixed weights, the sequence generated with changing weights shows more a complex behavior: longer cycles, more randomness, and chaotic as opposed to quasiperiodic behavior. It seems likely that this tendency also holds for other algorithms in which the weights keep changing.

ACKNOWLEDGMENTS

R. M., W. K., and I. K. are grateful for financial support by the German-Israeli Foundation. We thank Stephan Mertens, Avner Priel, and Andreas Engel for discussions, know-how, and ideas.

[1] H. Zhu and W. Kinzel, Neural Comput. **10**, 2219 (1998).
[2] J. Hertz, A. Krogh, and R. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood City, CA, 1991).
[3] E. Eisenstein, I. Kanter, D. A. Kessler, and W. Kinzel, Phys. Rev. Lett. **74**, 6 (1995).
[4] M. Schröder, Ph.D. thesis, Universität Würzburg, 1998.
[5] M. Schröder and W. Kinzel, J. Phys. A **31**, 9131 (1998).
[6] *On-Line Learning in Neural Networks*, edited by D. Saad (Cambridge University Press, Cambridge, 1998).
[7] G. Reents and R. Urbanczik, Phys. Rev. Lett. **80**, 5445 (1998).
[8] S. Mertens and C. Bessenrodt, J. Phys. A **31**, 3731 (1998).
[9] M. Schroeder, *Number Theory in Science and Communication* (Springer-Verlag, Berlin, 1984).
[10] J. Bernasconi, J. Phys. (France) **48**, 559 (1987).
[11] M. J. E. Golay, IEEE Trans. Inf. Theory **IT-28**, 543 (1982).
[12] C. de Groot, D. Würtz, and K. H. Hoffmann, Optimization **23**, 369 (1992).
[13] S. Mertens, J. Phys. A **29**, L473 (1996).
[14] http://itp.nat.uni-magdeburg.de/~mertens/bernasconi/cyclic.shtml
[15] C. Van den Broeck and R. Kawai, Phys. Rev. A **42**, 6210 (1990).
[16] I. Kanter and D. A. Kessler, Phys. Rev. Lett. **74**, 4559 (1995).
[17] D. Challet and M. Marsili, e-print, cond-mat/0004196.
[18] I. Kanter, D. A. Kessler, A. Priel, and E. Eisenstein, Phys. Rev. Lett. **75**, 2614 (1995).
[19] A. Priel, I. Kanter, and D. A. Kessler, J. Phys. A **31**, 1189 (1998).
[20] A. Priel and I. Kanter, Phys. Rev. E **59**, 3368 (1999).
[21] I. Kanter, Phys. Rev. A **38**, 5972 (1988).
[22] I. Kanter and D. Saad, Phys. Rev. Lett. **83**, 2660 (1999).
[23] E. Ott, *Chaos in Dynamical Systems* (Cambridge University Press, Cambridge, 1993).
[24] *Introduction to Nonlinear Physics*, edited by L. Lam (Springer, New York, 1997).
[25] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, Physica D **16**, 285 (1985).
[26] J. Kaplan and J. Yorke, in *Functional Differential Equations and Approximations of Fixed Points*, edited by H. Peitgen and H. Walther (Springer, Heidelberg, 1979).