Distributed dynamics in neural networks

Andreas V.M. Herz
Physics of Computation Laboratory, Division of Chemistry, California Institute of Technology 139-74, Pasadena, California 91125

Charles M. Marcus*
Division of Applied Sciences and Department of Physics, Harvard University, Cambridge, Massachusetts 02138

(Received 5 October 1992)

We analyze the dynamics and statistical mechanics of attractor neural networks with "distributed" updating rules in which groups of one or more neurons are updated simultaneously. Such partially parallel updating schemes are a central feature of neural-network architectures that use many processors, implemented either on special multiprocessor hardware, or among many computers linked over a network. Several updating rules are classified and discussed; these rules generalize the parallel dynamics of the Little model and the one-at-a-time dynamics of the Hopfield model. Analytic results presented herein include a stability criterion that specifies sufficient conditions under which distributed dynamics lead to fixed-point attractors. For binary neurons with block-sequential updating and a Hebbian learning rule, the storage capacity is found as a function of the number of update groups. Several open problems are also discussed.

PACS number(s): 87.10.+e, 64.60.Cn, 89.80.+h, 06.50.Mk

I. INTRODUCTION

Neural networks provide a systematic approach to massively parallel computation. For this reason they offer great promise for real-time applications where serial computers are too slow to be useful. Interestingly, to date the vast majority of neural-network applications have been implemented on single-processor computers, with (artificial) parallelism created in software. It is generally believed that this situation is temporary, and that the ultimate advantage of massively parallel computation will require true parallelism in which the computational task is distributed among a large number of processors, for instance, by using special purpose hardware or computer networks.

A concern in designing a parallel, distributed neural network is the *global stability* of the collective network state when many parts of the system are free to change their state simultaneously. As an example, attractor networks configured to store fixed patterns may be globally stable when the states of single "neurons" are updated one at a time — foregoing parallelism altogether — but may possess a large number of spurious oscillatory modes when all neurons are updated in parallel. In particular, all of the strong global stability results for sequential updating are constrained by the requirement that no set of two or more elements change their state at the same time. This constraint becomes severe when communication delays or the time required to perform local computations become comparable to the updating period. In this regime, standard updating rules break down, and little is known about the global dynamics.

There is a large body of literature within computer science that deals with iterative solutions for systems of linear and nonlinear equations using distributed processors. Global stability has been established for a number of distributed iteration schemes. Unfortunately, all of these results require that the system of equations possess a unique solution. For this reason, these general results are of limited applicability to neural networks. For attractor neural networks, the presence of frustration resulting from competing interactions typically ensures that the dynamics take place on a complex landscape with many local minima, corresponding to multiple solutions. In some sense, this situation is the defining characteristic of the sort of computational complexity that neural networks are designed to cope with.

Ultimately, the goal of this research is to be able to guarantee stability against oscillations or chaos in a neural network that has no controlling clock or other form of central timing (e.g., token passing). In such a distributed system, each processor would compute and update as rapidly as possible, and would broadcast its output as soon as it is available. The results we obtain are not as general as the results for systems that possess a unique attractor, and, for instance, do not cover dynamics with a range of delays and update rates that can lead to *overlapping*, illustrated in Fig. 1. Global stability analysis in this case remains an open problem.

In the present work, we focus on a discrete-time description of computational events. Such an approach applies directly to clocked networks of digital processors. However, our results will also shed light on continuous-time systems: As long as the time required for an individual neuron to change its state is much shorter than communication delays and the overall network relaxation time, these individual transitions can be considered quasi-instantaneous.

The paper is organized as follows. In Sec. II we develop a general framework to distinguish various discrete-time updatirg schemes in neural networks. In Sec. III we introduce a global stability analysis for certain classes of network dynamics by extending the Lyapunov function approach to these situations. Quantitative results on the emergent network properties are presented in Sec. IV. Finally, conclusions and open problems are discussed in Sec. V.

II. UPDATING STRATEGIES

A. Single-neuron dynamics

The state or "activity" of a single neuron, say neuron i, $1 \leq i \leq N$, is characterized by a scalar variable $x_i(t)$ which may be continuous (i.e., analog) or discrete. In the latter case, two-state neurons are of particular interest. They were first introduced by McCulloch and Pitts [1] and will be denoted by $S_i = \pm 1$ to emphasize their resemblance with Ising spins.

The time evolution of neurons is governed by their local fields $h_i(t)$, weighted sums of incoming signals from other elements and external sources I_i,

$$h_i(t) = \sum_{j=1}^{N} J_{ij} x_j(t) + I_i \ . \qquad (1)$$

Here, the contribution of other neurons to $h_i(t)$ is assumed to be linear in their activity level and is mediated through the interconnection matrix $\mathbf{J}$ which we require to be symmetric, i.e., $J_{ij} = J_{ji}$.

The dynamics of a single neuron updated at time t may either be deterministic or stochastic. In the deterministic case, we may write

$$x_i(t+1) = F_i(h_i(t)) \qquad (2)$$

with piecewise constant function F_i for networks with discrete neurons or continuous transfer function for analog systems. We will call an input-output characteristic F_i "sigmoidal," if it is increasing, differentiable, and grows in magnitude more slowly than linear for large positive or negative arguments. The maximum slope of F_i will be referred to as the gain β_i of neuron i.

In the case of stochastic dynamics, we introduce a thermal noise which acts to improve computational performance. Also, this noise may be thought of as roughly incorporating the probabilistic nature of synaptic transmission in biological systems. For binary neurons, the favorite choice of physicists has been Glauber dynamics,

$$\text{Prob}\left[S_i(t+1) = \pm 1\right] = \frac{1}{2}\{1 \pm \tanh[T^{-1} h_i(t)]\} \ , \qquad (3)$$

since it allows for a connection with statistical mechanics [2, 3]. In (3), the "temperature" T characterizes the level of synaptic noise. In the noiseless limit, $T \rightarrow 0$, one recovers the deterministic dynamics $S_i(t+1) = \text{sgn}[h_i(t)]$.

B. Network dynamics

We now turn to the dynamics of the overall network. First, we consider how many processors may change their state at a given time. Theoretical investigations of the global behavior of feedback networks have almost exclusively focused on two extreme cases: parallel updating of all neurons [2] and one-at-a-time updating [4]. Notice that both of these scenarios require a global clock — in the case of parallel updating, the clock is needed to synchronize all neurons; in the case of one-at-a-time updating, the clock prevents the simultaneous updating of two or more neurons. Few results have been reported about the convergence and stability of more general models where *groups* of neurons are updated in parallel.

The only analytic study along these lines known to the authors was performed by Chacc, Fogelman-Soulie, and Pellegrin [5], who focused on block-sequential iterations. Using a global Lyapunov function, these authors investigated the asymptotic behavior and convergence time of systems with deterministic threshold units and a fixed partition into clusters of simultaneously updating neurons. Their stability results will be discussed in Sec. III.

Next, there is the question of how groups (of one or more neurons) are chosen at each time step. One may have a fixed partition of the network into groups of neurons [5] — this approach reduces to serial dynamics when each "group" consists of a single neuron. Alternatively, one may choose random samples at each time step. Or, one may use a selective mechanism to optimize network performance, as has been proposed by Horner, who investigated a "maximum-field" or "greedy" dynamics [6]. Here, the neuron with the largest local field opposite to its own activity is picked, allowing for a rapid descent in the energy landscape. The network dynamics of integrate-and-fire systems may similarly be viewed as a selective updating algorithm.

Throughout what follows, updating schemes are said to be *fair sampling* if on an intermediate time scale no neuron is skipped for the updating process on average. The terminology emphasizes the similarity with the idea of "fairness" used in the computer science literature; see, e.g., Ref. [7]. On a conceptual level, fair sampling guarantees that all neurons have a chance to explore the part of phase space accessible to them through their single-neuron dynamics (2) or (3). Obviously most computationally useful iteration schemes will be of this type. All updating schemes with a fixed partition or random selection process are fair sampling. However, exceptions may occur in pathological situations within selective algorithms. On a technical level, the property that the update rule be fair sampling is important for analyzing the asymptotic behavior of networks with a distributed dynamics.

Finally, there is the problem of communication delays between individual elements. The time evolution of both real neurons and artificial processors may be strongly influenced by signal delays due to the finite speed of the transmission of local information. Neurons will in general respond to activity states of other units, some of which are already outdated, some not, as is illustrated in Fig. 1. Problems of this kind have been successfully addressed in the computer science community under such terms as "asynchronous algorithms" or "chaotic relaxation" [8]. We emphasize, however, that analytical results obtained for these algorithms *do not* apply to neural networks where multiple basins of attraction are vital for use as an autoassociative memory. On the other hand, for multibasin systems, encouraging experimental results have been obtained by Sawitzki and co-workers [9] using a large array of interconnected personal computers simulating a Hopfield-like neural network. They have demonstrated that even in a highly complex network with competing interaction and overlapping delays, the system still settles to a stationary state.

Summarizing the above discussion, updating strategies for distributed dynamics may be categorized according to the following four criteria:

(1) Single-neuron dynamics: (a) deterministic; (b) stochastic.

(2) Size of group $U(t)$ to be updated at each time step: (a) all neurons; (b) some neurons; (c) one neuron.

(3) How update group $U(t)$ is chosen at each time step: (a) fixed partition; (b) random sample; (c) selective.

(4) Handling of delays: (a) overlapping not allowed; (b) overlapping allowed.

Most network dynamics appearing in the literature can be classified by these four criteria. For instance, the Little model [2] is stochastic [rule (1b], with all neurons updated at the same time [rule (2a) and (3a)] using instantaneous interactions [rule (4a)]. The Hopfield model [4] uses either deterministic or stochastic updating on the single-neuron level [rule (1a) or (1b)], with one neuron updated at a time [rule (2c)] and no signal delays [rule (4a)]. In general, neurons may be updated in a random sequential manner [rule (3b)], but in simulations, the update order is often fixed in advance, corresponding to a quenched random selection [rule (3a)]. Block-sequential algorithms for solving coupled nonlinear equations use a deterministic dynamics with fixed clusters and nonoverlapping communication [rules (1a), (2b), (3a), and (4a)]. Updating schemes with overlapping delays [rule (4b)] are usually referred to as asynchronous algorithms.

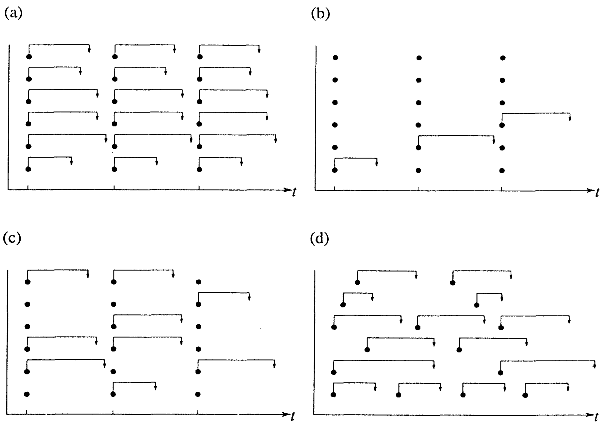

FIG. 1. Schematic representation of various updating schemes discussed in the text. Horizontal axis represents time. Delays due to transmission and computation times are indicated by the finite duration of the updating "event" for a given neuron. Clocked networks have ticks on time axis. (a) Parallel or synchronous dynamics; (b) sequential or one-at-a-time dynamics; (c) distributed dynamics, discussed in the present paper: still clocked, but with arbitrary update groups at each time step, (d) fully asynchronous dynamics, including *overlapping*. Global stability analysis in case (d) remains an open problem.

In the remainder of this paper we will consider an updating scheme that generalizes beyond the Hopfield and Little models. In particular, we allow both deterministic and Glauber dynamics on the single-neuron level, and all choices concerning rules (2) and (3) that are fair sampling. To some extent, we will follow the directions already investigated within the asynchronous computation literature, though with the added complication of multiple basins of attraction that are central to neural networks. This complication forces us at present to restrict our attention to rule (4a)—*nonoverlapping updating*—as illustrated in Fig. 1. For the deterministic case [rule (1a)], the network dynamics is thus defined by a set of discrete-time equations,

$$x_i(t+1) = \begin{cases} F_i\left(\sum_{j=1}^{N} J_{ij}x_j(t) + I_i\right) & \text{if } i \text{ is in } U(t), \\ x_i(t) \text{ otherwise.} \end{cases} \tag{4}$$

Our stability analysis can be applied to analog and discrete-valued neurons, where for the case of binary neurons, we recover earlier results of Goles-Chacc, Fogelman-Soulie, and Pellegrin [5]. The extension to stochastic Glauber dynamics will be analyzed in detail in Sec. IV, and will also help to obtain quantitative results about the emergent network properties for a system with deterministic two-state neurons.

In closing this section, let us mention that an important issue excluded in the present analysis — the dynamic effect and functional role of signal delays — has been addressed in specific contexts by various authors (see [10] for an overview). Results concerning the global dynamics and statistical mechanics for clocked time-delay networks have been obtained for systems storing both static patterns [11] and temporal associations [12, 13].

III. GLOBAL ANALYSIS OF THE DETERMINISTIC DYNAMICS

In this section we provide conditions sufficient to guarantee that the distributed dynamics as given by Eq. (4) converge to fixed points only.

Consider the discrete-time evolution of the real scalar function $L(t)$,

$$L(t) = -\frac{1}{2}\sum_{i,j=1}^{N} J_{ij}x_i(t)x_j(t) + \sum_{i=1}^{N}[G_i(x_i(t)) - I_i x_i(t)] \ , \tag{5}$$

where the auxiliary quantities $G_i(x_i)$ are defined by

$$G_i(x_i) \equiv \int_0^{x_i} F_i^{-1}(z)dz \ . \tag{6}$$

For sigmoidal F_i, the G_i increase faster than quadratic for large absolute argument and guarantee that the function $L(t)$ is bounded below. Notice that L coincides with the Lyapunov function used by Marcus and Westervelt to study the time evolution of iterated-map networks [14] and with Hopfield's energy function for the special case of binary neurons [4].

The change of $L(t)$ in a single time step, defined as $\Delta L_i(t) \equiv L_i(t+1) - L_i(t)$ is given by

$$\begin{aligned}\Delta L(t) = &-\frac{1}{2}\sum_{i,j=1}^{N} J_{ij}[x_i(t+1)x_j(t+1) - x_i(t)x_j(t)] \\ &+\sum_{i=1}^{N}[G_i(x_i(t+1) - G_i(x_i(t)) \\ &-I_i(x_i(t+1) - x_i(t))] \ . \end{aligned}\tag{7}$$

The only neurons that may change their state at time t belong to the update group $U(t)$. Accordingly, $\Delta x_i(t) \equiv x_i(t+1) - x_i(t)$ vanishes for all other neurons. Using the symmetry of the connection matrix one obtains

$$\begin{aligned}\Delta L(t) = &-\frac{1}{2}\sum_{i\in U(t)}\sum_{j\in U(t)} J_{ij}\Delta x_i(t)\Delta x_j(t) \\ &-\sum_{j=1}^{N}\sum_{i\in U(t)} J_{ij}x_j(t)\Delta x_i(t) \\ &+\sum_{i\in U(t)}[G_i(x_i(t+1)) - G_i(x_i(t)) - I_i\Delta x_i(t)] \ . \end{aligned}\tag{8}$$

Since sigmoidal F_i are single valued and monotone increasing, the G_i are strictly convex. Expanding $G_i(x_i(t))$ in a Taylor series around $x_i(t+1)$ and replacing the coefficient of the quadratic term with the smallest possible value, i.e., β_i^{-1}, the following upper bound can be established:

$$\begin{aligned}&G_i(x_i(t+1)) - G_i(x_i(t)) \\ &\qquad \leq \Delta x_i(t)G_i'(x_i(t+1)) - [\Delta x_i(t)]^2\beta_i^{-1} \ . \end{aligned}\tag{9}$$

Notice that equality holds if and only if $x_i(t+1) = x_i(t)$. For a graphical illustration see [14]. Inserting the identity $G_i'(x_i(t+1)) = F_i^{-1}(x_i(t+1)) = \sum_{j=1}^{N} J_{ij}x_j(t) + I_i$ and the above inequality into (8), one arrives at the final expression

$$\Delta L(t) \leq -\frac{1}{2}\sum_{i\in U(t)}\sum_{j\in U(t)}[J_{ij} + \delta_{ij}\beta_i^{-1}]\Delta x_i(t)\Delta x_j(t). \tag{10}$$

To facilitate further discussion, let us define $V(t)$ as the number of neurons in the group $U(t)$ and symmetric $V(t) \times V(t)$ matrices $\mathbf{U}(t)$ as $V(t)$-dimensional submatrices of the connection matrix $\mathbf{T}$, determined by the interactions of those neurons that are updated at time t. For the Hopfield model [4], where updating is one-at-a-time, $V(t) = 1$ for all t, and $\mathbf{U}(t)$ reduces to the self-interaction term J_{ii}, where i denotes the neuron being updated at time t. For the Little model [2] or iterated-map analog networks [14], the matrix is identical to $\mathbf{J}$ itself. As is obvious from these limiting cases, the structure of the set of matrices $\mathbf{U}(t)$ encodes the global dynamics.

We next define the maximum neuron gain in the update group $U(t)$ by $\beta(t)$ and the minimum eigenvalue of the matrix $\mathbf{U}(t)$ by $\lambda_{\min}[\mathbf{U}(t)]$. Since for arbitrary symmetric matrices $\mathbf{A}$ and $\mathbf{B}$, $\lambda_{\min}[\mathbf{A}+\mathbf{B}] \geq \lambda_{\min}[\mathbf{A}] + \lambda_{\min}[\mathbf{B}]$, a sufficient condition for $\Delta L(t) \leq 0$ is given by

$$\beta(t)^{-1} \geq -\lambda_{\min}[\mathbf{U}(t)] \ . \tag{11}$$

If the above condition holds for all t, $L(t)$ is strictly decreasing as long as $x_i(t+1) \neq x_i(t)$ for at least some i in the update group $U(t)$. The function L is bounded below. The dynamics relaxes therefore asymptotically to a state where L does not vary in time if all directions in the space spanned by the neural activities are explored, i.e., if the updating scheme is fair sampling. Since the equality in (9) and (10) holds only if $x_i(t+1) = x_i(t)$, the only solutions of (4) with time-independent L are fixed-point solutions. Earlier results on iterated-map networks [14] are recovered if $U(t)$ is constant in time and identical to the set of all neurons. The opposite case, where analog neurons are updated according to a one-at-a-time dynamics, has been discussed in Ref. [15].

Our results up to this point can be summarized as follows: Suppose the following three conditions hold: (a) the updating rule is fair sampling, (b) the neuron transfer functions are sigmoidal, and (c) the symmetric connection matrix satisfies (11) for all times. Then the distributed dynamics (4) converges to fixed points only.

Networks with discrete elements correspond to the limit $\beta_i \to \infty$, and (11) reduces to $\lambda_{\min}[\mathbf{U}(t)] \geq 0$. The convergence properties of the models of Hopfield and Little follow immediately: the Hopfield model has no time-dependent attractors if the self-interactions J_{ii} are all positive or zero; the Little model has no time-dependent attractors if the whole connection matrix is non-negative definite. The findings of Goles-Chacc, Fogelman-Soulie, and Pellegrin [5] on block-sequential algorithms correspond to a special intermediate case where the network is partitioned into fixed updating clusters $U_k, k = 0, 1, ..., K-1$ and $U(t) = U_{(t \bmod K)}$.

We mention a potentially confusing point concerning the discrete-neuron limit $\beta_i \to \infty$. Whereas for any finite β, the state space is continuous, it collapses for binary neurons to the corners of an N-dimensional hypercube, also known as Hamming space — and to generalizations thereof for multistate neurons. In such a discrete space, the smallest state change is a single spin flip. As a consequence, networks with discrete neurons may converge to fixed points that are not stable with respect to single-spin flips, in the sense that a single-bit change made to such a fixed point could actually lower L. For instance, consider a Hopfield-like model where for some neuron i, the self-interaction dominates possible contributions from other neurons, i.e., $J_{ii} > \sum_{j\neq i}|J_{ij}|$: the initial value of S_i will never be changed, independent of its sign. Our earlier results about network convergence continue to hold, that is, the system evolves towards fixed points only, but those may or may not be local minima of L in the discrete-space sense. This atypical behavior is only possible because the F_i are piecewise constant functions in models with discrete neurons. For the generic case of sigmoidal input-output characteristics, the network will always settle in a minimum as long as the initial conditions do not coincide with an unstable fixed-point of (4).

The comparison between the Hopfield and Little model demonstrates that the convergence criterion is less restrictive for smaller update groups than for larger ones, as is also obvious from (11) since

$$\lambda_{\min}[\mathbf{U}_1] \geq \lambda_{\min}[\mathbf{U}_2] \quad \text{if} \quad U_1 \subset U_2\ . \tag{12}$$

The case of equality in (12) holds if $\mathbf{U}_2$ is a tensor product of $\mathbf{U}_1$ and some other matrix $\mathbf{U}_3$ and $\lambda_{\min}[\mathbf{U}_3] \geq \lambda_{\min}[\mathbf{U}_2]$. Note that (12) implies in particular that the stability criterion for a fully parallel network $[V(t) = N]$, $\beta^{-1} \geq -\lambda_{\min}[\mathbf{J}]$, is a sufficient condition for (11), and thus sufficient to assure that the system (4) will converge to a fixed point for *any* fair sampling updating scheme.

Formula (12) has direct consequences for possible applications. Consider a high-dimensional optimization task such as the traveling-salesman problem. It may be mapped onto a neural network architecture—see e.g., [16]—which then defines a fixed connection matrix $\mathbf{J}$. The computational time needed to find a good solution can easily be reduced on a parallel computer by increasing the size of update groups. However, the bounds given by (11) have to be met in order to ensure convergence to fixed points, and will limit the maximal update group size. The goal of large updating groups will be achieved in an optimal way if one can form update groups of weakly or non-interacting neurons. All submatrices $\mathbf{U}(t)$ will have small off-diagonal elements in that case, and their eigenvalues will be close or identical to the diagonal elements, i.e., the bounds (11) are largely independent of the size of the update groups. In principle, the search for optimal partitions of the above kind is itself a difficult optimization problem, but many applications exhibit an intrinsic structure (e.g., predominantly short-range interactions) which naturally leads to good choices for the updating groups. An example of a network partitioning is illustrated in Fig. 2.

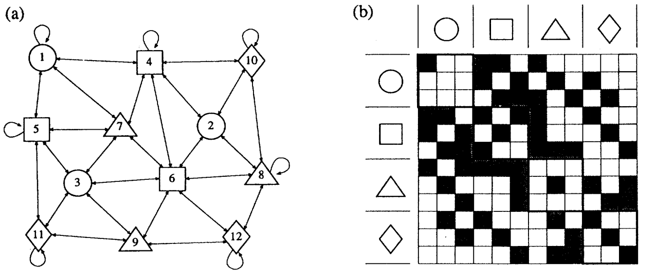

FIG. 2. (a) Example neural-network architecture with 12 neurons belonging to four update groups, represented by four shapes. Symmetric connections between neurons are represented by double-arrowed lines, self-connections by looped arrows. (b) 12 × 12 connection matrix for network in (a), with filled squares indicating nonzero connection strengths. The stability criterion (11) places a bound on the minimum eigenvalues of submatrices corresponding to update groups, shown as bold 3 × 3 blocks along the diagonal. For many structured architectures like the planar network in (a), neurons can be judiciously assigned to groups so as to greatly reduce or eliminate connections within a group.

IV. STATISTICAL MECHANICS AND NETWORK PERFORMANCE

In this section, we investigate the associative capabilities of neural networks with distributed dynamics. The networks are designed to store a set of p memory patterns $\xi^\mu \in \{-1,+1\}^N$, $1 \leq \mu \leq p$ as fixed points of the dynamics (4), and we want to compare the emergent network properties with those of conventional models with a random sequential or fully parallel dynamics. A statistical mechanical analysis of performance measures such as storage capacity and retrieval quality can be carried out most readily if we restrict our attention to networks that can be partitioned into n fixed update blocks of equal size V such that there are *no interactions within a group*. As emphasized before, such a situation can be arranged for many applications that map onto diluted or geometrically structured networks.

To further simplify the analysis, we focus on a system with zero external input and two-state neurons of Ising type. We now label neurons by a double index S_{ia}. The first index $1 \leq i \leq V$ refers to the position within an update group, the second $1 \leq a \leq n$ labels the update group. As before, the total number of neurons is $N = nV$. The connection structure we consider is a Hebb rule with connections within an update group set to zero:

$$J_{ij}^{ab} = \begin{cases} \frac{1}{V(n-1)}\sum_{\mu=1}^{p}\xi_{ia}^{\mu}\xi_{jb}^{\mu} & \text{if } a \neq b \\ 0 & \text{if } a = b\ . \end{cases} \tag{13}$$

The normalization factor in (13) guarantees the correct scaling behavior of L in the thermodynamic limit $N \to \infty$. In what follows, we will study the case of large update group size $V \to \infty$ with the number of update groups n kept finite. Also, we require that there be at least two update groups $n \geq 2$, otherwise all neurons would be disconnected according to (13).

Statistical mechanics may be used to analyze the emergent properties of this block-sequential dynamics once we have shown that under a stochastic dynamics such as (3), the network relaxes to a Gibbsian equilibrium distribution generated by L. This need not be true in general — although L is identical to the Hamiltonian of a Hopfield network of size N — because the distributed updating scheme differs from both the Hopfield or Little dynamics where proofs using the principle of detailed balance are well established [3]. However, in the special case of vanishing connection strength within all update groups (13), neurons "do not know" about the state of other neurons in the same group and thus there is no formal difference between the block-sequential rule considered here and a serial updating, where neurons change their state in consecutive order: every set of V successive updates of the latter dynamics is identical to one time step in the former case.

In what follows, we work with unbiased random patterns where $\xi^{\mu}_{ia} = \pm 1$ with equal probability and study our network at a finite storage level $\alpha = \lim_{N\to\infty}(\frac{p}{N})$. Following the replica-symmetric theory of Amit, Gutfreund, and Sompolinsky [17], we single out s patterns, with s finite, and assume that the network is in a state highly correlated with these memories. The remaining, extensively many patterns are described collectively by a noise term. Notice that for coupling matrices of the form (13), both the overlaps m and spin-glass parameters q have to be defined as order parameters *on the level of the update groups.* In light of this requirement, and considering the *retrieval solutions*, we make the ansatz

$$m^{\mu}_{\sigma a} \equiv V^{-1}\sum_{i=1}^{V} \xi^{\mu}_{ia} S^{\sigma}_{ia} = m\delta_{\mu,1} \ , \tag{14}$$

$$q^{\rho\sigma}_{ab} \equiv V^{-1}\sum_{i=1}^{V} S^{\rho}_{ia} S^{\sigma}_{ib} = \delta_{ab}[\delta_{\rho\sigma}(1-q)+q] \tag{15}$$

for a k-fold replicated network, $1 \le \rho, \sigma \le k$, and arrive at the fixed-point equations

$$m = \langle\langle \tanh[T^{-1}\{m+\sqrt{\alpha r}z\}]\rangle\rangle \tag{16}$$

and

$$q = \langle\langle \tanh^2[T^{-1}\{m+\sqrt{\alpha r}z\}]\rangle\rangle \tag{17}$$

where

$$r \equiv \frac{q}{[1-T^{-1}(1-q)]^2} - \frac{q(n-1)}{[n-1+T^{-1}(1-q)]^2} \ . \tag{18}$$

Double angular brackets represent an average with respect to both the condensed patterns and the normalized Gaussian random variable z.

The preceding equations closely resemble their counterparts for the Hopfield model [17], and become identical to them in the limit of large n, i.e., many update groups. On a formal level, the same holds for $n = 1$, but as explained before, this case corresponds to an unphysical situation. For a general number of update groups, there exists a first-order phase transition at $T = 0$ between the retrieval state and a spin-glass phase as α is varied. The critical storage level α_c and the corresponding overlap m_c are shown below. In passing, let us remark that there is also a high degree of similarity between the fixed-point equations (16) – (18) and results obtained for time-delay networks [12]. This is not surprising since the latter class of model can be mapped onto networks with block-sequential updating and no time delays.

We now introduce a third performance measure, the information content $I_R(n)$, measured *per synapse* and *relative* to that of a Hopfield model,

$$I_R(n) \equiv \frac{I_n(\text{block-sequential})}{I(\text{random-sequential})} = \frac{n\,\alpha_c(n)}{(n-1)\alpha_c(\text{Hopfield})} \tag{19}$$

where as before n counts the number of updating groups.

A numerical solution of the saddle-point equations at $T = 0$ leads to the following results:

n	α_c	m_c	I_R
2	0.100	0.93	1.45
3	0.110	0.95	1.20
4	0.116	0.96	1.12
5	0.120	0.96	1.09

Compared to the Hopfield or Little model where $\alpha_c = 0.138$, and $m_c = 0.97$, our block-sequential updating scheme exhibits quantitatively similar performance: the capability to retrieve stored random patterns is slightly lower when measured in terms of patterns per neuron — see the second column — and slightly higher when measured in terms of patterns per synapse — see the last column. Notice in particular, that the information content increases with decreasing network connectivity, i.e., for small n.

The above findings clearly indicate that the associative capabilities of networks with a distributed dynamics are rather similar to those of systems with either fully parallel or one-at-a-time updating rules. Unlike these conventional schemes, they offer a potentially large advantage in terms of computational costs when implemented on a parallel computer, allowing for a speed-up that may be as large as the number of processors, without sacrificing network stability.

V. CONCLUSIONS AND OPEN PROBLEMS

We have studied the global dynamics of neural networks with a distributed updating scheme. We have shown that dynamics with a deterministic single-neuron time evolution and nonoverlapping communication delays converge to a fixed point attractor if three conditions hold: First, the process of selecting update groups has to be *fair sampling*, that is, no neurons may be neglected (on average) by the updating rule. Second, the neuron transfer functions must be sigmoidal, as is the case for most implementations in analog hardware. Third, the connection matrix has to be symmetric and in addition satisfy the stability criterion (11), a constraint that can be naturally met in many applications, particularly where limited-range interactions play a dominant role. For a discussion of biological aspects of the present model, see Ref. [18].

We have also investigated the stochastic Glauber dynamics of two-state neurons in a distributed updating scheme and discussed the equilibrium statistical mechanics for this case. Within a replica-symmetric theory, quantitative measures of emergent network properties were obtained. They showed that both the storage capacity and information content are similar to those of the Hopfield model. Given the significant computational advantage of block-sequential updating schemes, their inherent ability to be run in parallel, we concluded that distributed dynamics offer a promising perspective for many applications.

The statistical mechanical analysis in Sec. III focused on two-state neurons and a block-sequential update scheme. It would be interesting to investigate the general conditions on the coupling matrix under which the stochastic dynamics will relax to an invariant probability distribution determined by L. Similarily, one might want to study different thermodynamic limits, where, for instance, the number of neurons per update group stays fixed while the number of clusters grows to infinity. Finally, systems with continuous neurons could be analyzed using techniques developed by Kühn and co-workers [15].

There are a number of important questions that we have not addressed in the present work: For instance, given the generality of the results reported here, one might want to optimize the updating scheme to further decrease the computational time. We have discussed a first step into this direction at the end of Sec. III. One could go further, however, and investigate updating schemes where the group sizes are changed in a selective manner during the descent on the energy landscape so as to minimize the time to reach a deep minimum while avoiding shallow local minima. It would also be interesting to investigate the consequences of violating condition (11). It is known [5, 14] that for a parallel updating scheme, violating (11) generates only simple period-two attractors. One might expect, however, that a variety of complicated dynamic attractors could appear in a distributed updating scheme when (11) is violated. Furthermore, it is possible in this case that the resulting chaotic dynamics could be used as an effective noise source similar to that of stochastic dynamics. One may even try to control these violations in a supervised manner by changing the update group size, as has been proposed for Monte Carlo simulations [19].

ACKNOWLEDGMENTS

It is a great pleasure to thank John Hopfield, Reimer Kühn, Leo van Hemmen, Fred Waugh, and Bob Westervelt for stimulating discussions. This study was initiated during a stay of C.M.M. at Caltech and finished during a visit of A.V.M.H. at Harvard. Both authors are grateful for the hospitality at the inviting institutes. C.M.M. acknowledges financial support from the ONR under Grant No. N00014-89-J-1592 and the NSF under Grant No. DMR-91-19386; A.V.M.H. thanks the DFG for financial support.

* Present address: Department of Physics, Stanford University, Stanford, CA 94305-4060.

[1] W. C. McCulloch and W. Pitts, Bull. Math. Biophys. **5**, 115 (1943).

[2] W. A. Little, Math. Biosci. **19**, 101 (1974).

[3] P. Peretto, Biol. Cybern. **50**, 51 (1984).

[4] J. J. Hopfield, Proc. Natl. Acad. Sci. U.S.A. **79**, 2554 (1982).

[5] E. Goles-Chacc, F. Fogelman-Soulie, and D. Pellegrin, Disc. Appl. Math. **12,** 261 (1985).

[6] H. Horner, in *Computational Systems — Natural and Artificial*, edited by H. Haken (Springer-Verlag, Heidelberg, 1988).

[7] K. M. Chandy, Sci. Comp. Pr. **14**, 117 (1990).

[8] D. Chazan and W. Miranker, Lin. Alg. Appl. **2**, 199 (1969); G. M. Baudet, J. Assoc. Comp. Mach. **25**, 226 (1978); D. Mitra, SIAM J. Sci. Stat. Comput. **8**, 43 (1987); P. Tseng, D. P. Bertsekas, and J. N. Tsitsiklis, Siam J. Control **28**, 678 (1989).

[9] G. Sawitzki, The NetWork Project, StatLab, Universität Heidelberg, 1989. Republished on Apple Developer CD Series disk IV (1990); J. Lindenberg, G. Sawitzki, R. Kühn, and J. L. van Hemmen (private communication).

[10] R. Kühn and J. L. van Hemmen, in *Models of Neural Networks*, edited by E. Domany, J. L. van Hemmen, and K. Schulten (Springer-Verlag, Heidelberg, 1990).

[11] C. M. Marcus and R. M. Westervelt, Phys. Rev. A **42**, 2410 (1990).

[12] A. V. M. Herz, Z. Li, and J. L. van Hemmen, Phys. Rev. Lett. **66**, 1370 (1991); A. V. M. Herz, Phys. Rev. A **44**, 1415 (1991).

[13] K. Bauer and U. Krey, Z. Phys. B **84**, 131 (1991); G. Mato and N. Parga, *ibid.* **84**, 483 (1991).

[14] C. M. Marcus and R. M. Westervelt, Phys. Rev. A **40**, 501 (1989).

[15] R. Kühn, S. Bös, and J. L. van Hemmen, Phys. Rev. A **43**, 2084 (1991); R. Kühn and S. Bös (unpublished).

[16] J. J. Hopfield and D. W. Tank, Biol. Cybern. **52**, 141 (1985).

[17] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Phys. Rev. A **32**, 1007 (1985); Ann. Phys. (N.Y.) **173**, 30 (1987).

[18] A. V. M. Herz and C. M. Marcus, in *Neural Networks: From Biology to High Energy Physics*, edited by O. Benhar, C. Bosio, P. Del Guidice, and E. Tabet (World Scientific, Singapore, in press).

[19] R. Kühn (private communication).