Analog neural networks with local competition. I. Dynamics and stability

F. R. Waugh and R. M. Westervelt
Division of Applied Sciences and Department of Physics, Harvard University, Cambridge, Massachusetts 02138
(Received 25 June 1992; revised manuscript received 25 January 1993)

We introduce a neural-network architecture in which analog-valued neurons compete through a constraint on a sum of neuron outputs within localized clusters. Local competition is useful in feature-extraction and pattern-classification applications. We show that with continuous-time updating, these networks converge only to fixed points, while with discrete-time, parallel updating, they converge to either fixed points or period-two limit cycles. We derive a stability criterion guaranteeing that discrete-time networks converge only to fixed points when their cluster gains, which are related to the slopes of the neuron input-output transfer functions, are sufficiently small. Numerical tests are presented showing that image-processing networks incorporating local competition operate reliably and in agreement with the stability criterion. We also describe a simple competitive analog electronic circuit that demonstrates that these networks are easily implementable.

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

I. INTRODUCTION

The relationship between the architecture of a neural network and its computational abilities is still poorly understood. Now that the properties of fully connected associative memories are well established [1–3], there is a trend toward studying network architectures with novel neuron interactions with the hope that they will exhibit new computational abilities. In these networks, neurons may communicate only with neurons in a layer [4,5], a cluster [6,7], or a restricted range [8–10], and the communication may involve inhibition [11], competition [12–17] or transmission delays [18,19]. The advantage in computational abilities may arise because the neuron interactions reflect the underlying structure of the problem being solved. Thus, for example, two-dimensional restricted-range interactions are useful for image processing [8–10], while transmission delays enable storage and retrieval of temporal pattern sequences [18,19].

In this paper, we introduce a network architecture in which analog-valued neurons interact not only through synaptic interconnections but also through a competitive mechanism. These competitive analog networks are related to the standard analog networks that have been studied previously [20–28]. A comparison of standard and competitive analog networks is made in Fig. 1. As indicated in the figure, neurons in both network architectures have analog input-output transfer functions and communicate through synaptic interconnections. Benefits of analog processing include discrete-time, parallel updating of neurons without oscillation [22–25, 29] and improved network performance through suppression of spurious attractors [25–27]. The important difference is that neurons in competitive analog networks are grouped into localized clusters, within which they compete through the constraint that their outputs sum to a constant at all times. Competition makes the output of a neuron depend on the inputs of *all* the neurons in its cluster, rather than just on its own input. As a result, clusters of neurons in competitive networks are capable of performing more complicated calculations—including the winner-take-all function, or more generally, the k-winner function—than are possible in standard analog networks.

The computational abilities arising from competition are especially useful for solving feature-extraction and pattern-classification problems. In these applications, each neuron in a cluster represents one feature or pattern, and competition among them determines which features or patterns are present. Localized competitive interactions have been used to detect elementary image features in image-processing networks such as the neocognitron

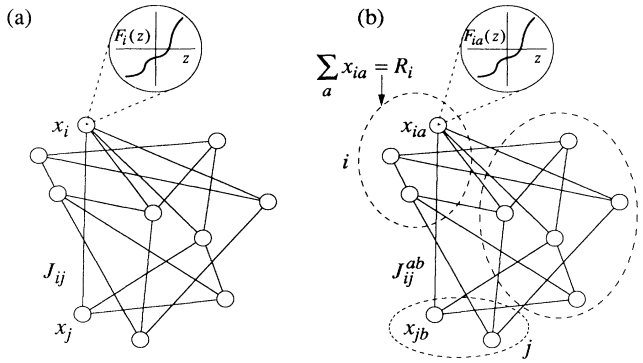

FIG. 1. Comparison of standard and competitive analog neural networks. Circles denote neurons, lines denote symmetric neuron interconnections. (a) Schematic diagram of a standard analog network of ten neurons. Nonlinear input-output transfer function $F_i(z)$ of neuron i and interconnection J_{ij} between neurons i and j are shown. (b) Same network with competitive interactions, denoted by dashed ellipses, among clusters of neurons. The ten neurons are divided into three competitive clusters containing two, three, and five neurons. Nonlinear input-output transfer function $F_{ia}(z)$ of neuron a in cluster i and interconnection J_{ij}^{ab} between neuron a in cluster i and neuron b in cluster j are shown. Competition is enforced by requiring neuron outputs in cluster i to sum to a constant R_i.

[30], and competition appears in other networks that classify patterns, learn regularities in inputs, and perform vector quantization [12–15]. More recently, Potts spins [31]—Q-state generalizations of Ising spins whose state is determined using a competitive rule—have been used to endow associative memories [32–35] and perceptrons [36,37] with localized feature-extracting abilities. The competitive networks we study here are closely related to, but considerably more general than, networks of Potts spins.

The particular competitive mechanism we study, which constrains the sum of neuron outputs in a cluster, has several attractive features. It requires only one connection for each neuron in a cluster; it is free of the stability problems often associated with competition [38]; and it can be implemented simply in a variety of physical systems by exploiting conservation laws [39–42]. Other competitive mechanisms involving all-to-all inhibitory interconnections [38,43], auxiliary neurons [38,43], or multineuron interactions [32] can be more cumbersome to implement.

This paper is organized as follows. In Sec. II, we show how to implement competition in clusters of analog neurons and describe a simple competitive electronic circuit we have built. Section III discusses how competitive clusters are connected to form networks and how these networks are related to networks of Potts spins. In Sec. IV, we show that, with continuous-time updating, competitive networks with symmetric interconnections converge only to fixed points, while with discrete-time, parallel updating, they converge to either fixed points or period-two limit cycles. We show furthermore that the limit cycles can be eliminated from discrete-time, parallel-update networks by reducing the slope of the neuron transfer functions sufficiently [22–25]. We then briefly discuss in Sec. V how local competition can be used to build two-dimensional feature-classifying networks for image processing. In the following paper (Ref. [44], hereinafter referred to as II), we apply competitive networks to the problem of associative memory. A summary of results appears in Sec. VI.

II. COMPETITIVE CLUSTERS

In this section, we describe how the neuron outputs in a *single* cluster of competing neurons are determined from the neuron inputs at one instant of time in the continuous-time case and during one time-step in the discrete-time case. (We will describe the dynamics of *networks* of clusters of competing neurons in Sec. III.) In Sec. II A, we show that competition can be implemented among a group of neurons by constraining the sum of their outputs to equal a constant and give examples of analog winner-take-all and analog k-winner clusters. In Sec. II B, we present an electronic circuit in which Kirchoff's current law implements this competitive mechanism.

A. Competition among a group of neurons

In the networks we study, neurons are grouped into clusters of two or more [See Fig. 1(b)]. We concentrate here on a single cluster i. The cluster contains Q_i neurons labeled by a, $a=1,\ldots,Q_i$, each of which is characterized by an input-output transfer function $F_{ia}(z)$ that may be different for each neuron. All Q_i neurons are updated simultaneously according to a set of Q_i equations that map real-valued neuron inputs h_{ia} onto real-valued neuron outputs x_{ia}. These equations may be either the continuous-time differential equations

$$\frac{dx_{ia}(t)}{dt}=-x_{ia}(t)+F_{ia}(h_{ia}(t)+B_i(t)) \ , \quad a=1,\ldots,Q_i \ , \tag{1a}$$

or the discrete-time, parallel-update equations

$$x_{ia}(t+1)=F_{ia}(h_{ia}(t)+B_i(t)) \ , \quad a=1,\ldots,Q_i \ . \tag{1b}$$

These equations are similar to the update equations of standard analog networks [20–28]. The important difference is that, in addition to their inputs h_{ia}, neurons also experience a time-dependent bias $B_i(t)$ that is the same for each neuron in a cluster. The bias $B_i(t)$ is determined implicitly at time t by the requirement that the outputs of all neurons in cluster i sum to a constant R_i at the *same* time t in the continuous-time case,

$$\sum_{a=1}^{Q_i} x_{ia}(t)=R_i \ , \tag{2a}$$

and at the *next* time step $t+1$ in the discrete-time case,

$$\sum_{a=1}^{Q_i} x_{ia}(t+1)=R_i \ . \tag{2b}$$

We will refer to Eqs. (2) as the *competitive constraint* and to groups of neurons obeying Eqs. (1) and (2) as *competitive clusters.*

The competitive constraint distinguishes the networks considered here from other analog neural networks [20–28]. It makes the output $x_{ia}(t+1)$ of neuron a depend not only on its input $h_{ia}(t)$ but also on the inputs $h_{ib}(t)$, $b\neq a$, of the other neurons in cluster i. In doing so, it constrains the Q_i neuron outputs in cluster i to a (Q_i-1)-dimensional space. (Thus a cluster of $Q_i=2$ neurons is equivalent to a standard analog neuron, because its two neuron outputs, which are related by $x_{i1}=R_i-x_{i2}$, lie in a one-dimensional space.) It is sometimes useful to think of the quantity R_i as a limited resource for which the neurons in cluster i compete. While it is possible to set R_i equal to zero through the transformation

$$F_{ia}(z)\rightarrow F_{ia}(z)-R_{ia} \ , \quad \sum_{a=1}^{Q_i} R_{ia}=R_i \ , \tag{3}$$

we will continue to allow it to be nonzero, since in hardware implementations its value may not be adjustable.

The transfer functions $F_{ia}(z)$ determine how the neurons in cluster i compete for the quantity R_i. Three restrictions are sufficient to ensure that the competitive constraint (2) has a single, unique solution for the bias

$B_i(t)$: (i) all $F_{ia}(z)$ in a cluster must be continuous; (ii) all $F_{ia}(z)$ in a cluster must either increase or decrease monotonically (without loss of generality we assume that all $F_{ia}(z)$ increase monotonically); and (iii) for all possible values of the neuron inputs h_{ia}, R_i must lie within the range of the function

$$S_i(z) \equiv \sum_{a=1}^{Q_i} F_{ia}(h_{ia}+z) , \tag{4}$$

which, if (i) and (ii) are satisfied, is a continuous, monotonically increasing function over all real numbers z. In addition, to ensure boundedness of the solutions to Eqs. (1), all $F_{ia}(z)$ in a cluster must asymptotically increase less than linearly either for large negative values of their arguments, for large positive values of their arguments, or for both (see Sec. IV). These requirements are very general and are easily met by transfer functions that lead to useful competitive behavior.

As an example of competition, consider a cluster i that uses the discrete-time update equation (1b) to compute an analog winner-take-all function of its inputs, meaning that the neuron with the largest input has the largest output while the other neuron outputs are suppressed. Suppose that $R_i=1$, and suppose that each neuron a in the cluster has the same transfer function

$$F_{ia}(z)=\exp(\gamma z) , \quad a=1,\ldots,Q_i , \tag{5}$$

where the parameter γ, the *neuron gain,* controls the transfer function slope. This transfer function is a natural choice in that it arises in the statistical-mechanical treatment of the winner-take-all problem [45]. It is also similar in form to the threshold-linear functions used in the neocognitron [30] and in other networks with competitive or inhibitory behavior [11]. For this transfer function, the bias $B_i(t)$ can be expressed explicitly in terms of the neuron inputs $h_{ia}(t)$, so that the update equation (1b) reads [45–47]

$$x_{ia}(t+1)=\frac{\exp[\gamma h_{ia}(t)]}{\sum_{b=1}^{Q_i}\exp[\gamma h_{ib}(t)]} , \quad a=1,\ldots,Q_i . \tag{6}$$

Now consider how the neuron outputs $x_{ia}(t+1)$ vary with γ in the case that neuron 1 has the largest input $h_{i1}(t)$ at time t. In the limit $\gamma\to\infty$ of very steep transfer functions, $x_{i1}(t+1)\to 1$ for neuron 1, while $x_{ia}(t+1)\to 0$ for the other neurons $a>1$ in the cluster. As γ decreases, $x_{i1}(t+1)$ decreases from 1 but remains the largest output in the cluster, while the other neuron outputs increase from 0. Finally, in the limit $\gamma\to 0$ of nearly flat transfer functions, all neuron outputs approach the same value $x_{ia}(t+1)\to 1/Q_i$. Thus, in this example, the competitive constraint (2b) rations the quantity R_i among the Q_i neuron outputs in the cluster according the size of their inputs at each time step, with the neuron gain γ controlling how much is awarded to each neuron.

Competitive clusters are not limited to calculating the analog winner-take-all function; the broad class of possible transfer functions allows a great variety in cluster functionality. A cluster can be configured to calculate an analog k-winner function [44,48]—meaning that the neurons with the k largest inputs have large outputs while the other neurons are suppressed—by setting the constant $R_i=k$ and giving each neuron the same sigmoidal or S-shaped transfer function $F(z)=[1+\exp(-\gamma z)]^{-1}$. More complicated calculations are possible as well: for example, the number of winners can be made to depend on which neurons have the largest inputs by allowing some neurons to have exponential transfer functions and others to have sigmoids.

In standard analog networks, the neuron gain, defined as the steepest slope of a neuron transfer function, plays an important role [20–28]. In competitive analog networks, the analogous quantity is the *cluster gain,* which we will define precisely in Eqs. (35) and (36) of Sec. IV. The gain β_i of a competitive cluster i is a measure of the change in its neuron outputs in response to a change in their inputs. The cluster gain is thus related to the slopes of the neuron transfer functions; for example, rescaling all transfer functions in a cluster by $F_{ia}(z)\to F_{ia}(cz)$ changes the gain of that cluster by $\beta_i\to c\beta_i$.

B. Competition in an analog electronic circuit

An attractive feature of the competitive mechanism described above is that it can be easily implemented in a variety of physical systems by using conservation laws to enforce the competitive constraint (2). Competitive systems have been constructed using current conservation in an electronic circuit [39–41] and gain conservation in a laser resonator [42]. Here we describe a simple analog electronic circuit that uses Kirchoff's current law to implement a Q-neuron analog winner-take-all cluster.

A schematic diagram of the circuit appears in Fig. 2. Each neuron a, $a=1,\ldots,Q$, consists of an n-channel enhancement metal-oxide-semiconductor field-effect transistor (MOSFET) T_a. The input of neuron a is the gate voltage V_a of transistor T_a, and the output of neuron a is the current I_a flowing through transistor T_a. The

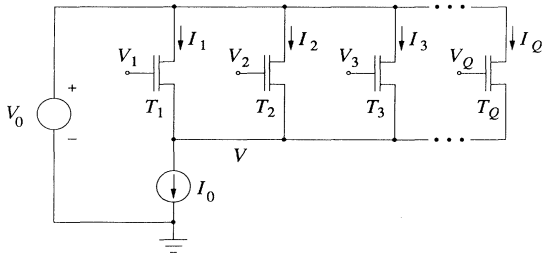

FIG. 2. Schematic diagram of analog electronic circuit implementing Q-neuron winner-take-all competitive cluster. Each neuron a, $a=1,\ldots,Q$, consists of an n-channel enhancement MOSFET T_a. Input of neuron a is gate voltage V_a of transistor T_a; output of neuron a is current I_a flowing through transistor T_a. Transistor sources are connected in parallel to a current source I_0, and transistor drains are connected in parallel to voltage source V_0. Kirchoff's current law enforces competition by constraining neuron outputs I_a to sum to I_0.

sources of the transistors are connected to a current source I_0 and their drains to a voltage source V_0. If the transistors are operated in the subthreshold regime, the current I_a through transistor T_a is [49]

$$I_a = I \exp(\gamma V_a)[\exp(-\gamma V) - \exp(-\gamma V_0)]$$
$$\cong I \exp[\gamma(V_a - V)] \ . \tag{7}$$

In Eq. (7), I is a device-dependent parameter typically 1 nA to 1 μA, V is the voltage of the common source line, V_0 is the power supply voltage, and $\gamma = e/k_B T \cong 40\ V^{-1}$, where e is the electron charge, k_B is Boltzmann's constant, and T is the temperature. The approximation holds in Eq. (7) when V_0 is much larger than V. The voltage V of the common source line adjusts to make the sum of the transistor currents equal the current through the current source, in accordance with Kirchoff's current law:

$$\sum_{a=1}^{Q} I_a = I_0 \ . \tag{8}$$

Comparing Eqs. (7) and (8) to Eqs. (1) and (2) shows that $-V$ plays the role of the bias term B_i, I_0 plays the role of the constant R_i, and Kirchoff's current law enforces the competitive constraint. Using Eq. (8) to eliminate V leads to

$$I_a \cong I_0 \frac{\exp(\gamma V_a)}{\sum_{b=1}^{Q} \exp(\gamma V_b)} \ , \tag{9}$$

which is identical to the analog winner-take-all transfer function (6).

We have constructed the circuit of Fig. 2 using Intersil Model VN86HF MOSFET transistors, a 15-V voltage source, and a standard op amp-based 75-mA current source. The transfer function for a typical transistor operating in the subthreshold regime is shown in Fig. 3(a). In Fig. 3(b), competitive behavior is shown in a cluster of $Q=3$ transistors. The gate voltages and currents (as measured by the voltage drop across a series 100-Ω resistor) of each transistor are shown in the case that the gates of transistors 1 and 2 are held at $V_1 = V_2 = 6$ V while the gate of transistor 3 is ramped between $V_3 = 5$ and 7 V. When $V_3 > 6$ V, transistor 3 has the maximum gate voltage and wins the competition. When $V_3 < 6$ V, transistors 1 and 2 both have the maximum gate voltage; a mismatch in transistor characteristics causes transistor 1 to win the competition. The circuit demonstrates that the competitive mechanism of Eqs. (1) and (2) can be implemented simply and robustly in standard analog electronics.

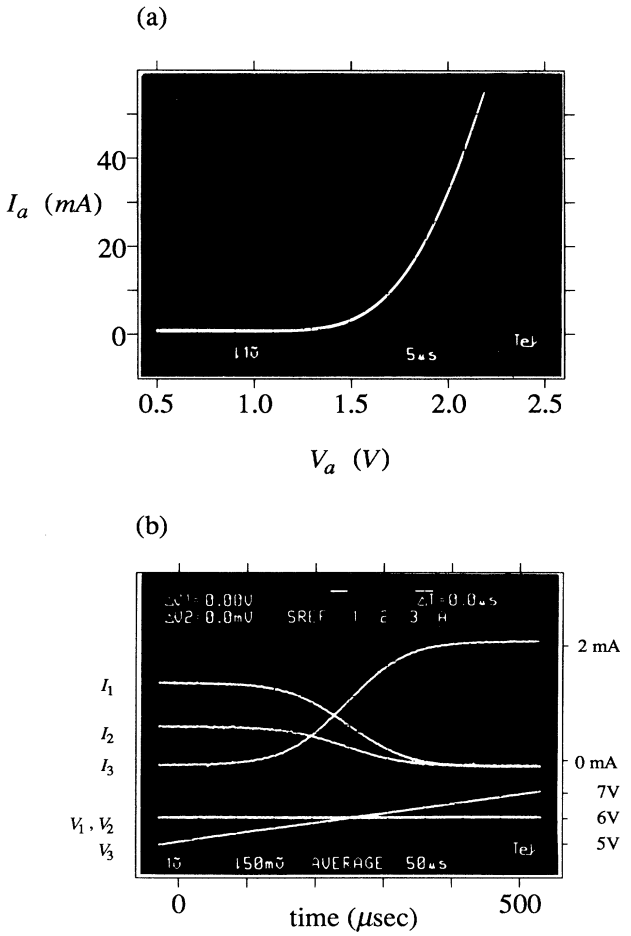

FIG. 3. Oscilloscope traces showing (a) transfer function of a single analog electronic neuron, (b) competition in an electronic circuit containing $Q=3$ neurons. In (b), input voltages (bottom) and output currents (top) of the three neurons are shown. Inputs of neurons 1 and 2 are held constant at 6 V while input to neuron 3 is ramped between 5 and 7 V. Competition leads to effectively sigmoidal response even though transfer function is exponential.

III. COMPETITIVE NETWORKS

In this section, we discuss the dynamics of networks of interacting competitive clusters. Section III A presents update equations for networks with continuous-time updating and discrete-time, parallel updating, and Sec. III B discusses how these networks are related to networks of Potts spins.

A. Network dynamics and interconnections

We consider networks of N interacting competitive clusters labeled by i, $i=1, \ldots, N$. The clusters may contain different numbers Q_i of neurons, so that the total number N_{tot} of neurons is

$$N_{\text{tot}} = \sum_{i=1}^{N} Q_i \ . \tag{10}$$

The neurons may have arbitrary transfer functions $F_{ia}(z)$ subject to the conditions stated above. An example of such a network is shown in Fig. 1(b).

The neurons are updated in parallel according to either the N_{tot} coupled nonlinear differential equations

$$\frac{dx_{ia}(t)}{dt}=-x_{ia}(t)+F_{ia}\left[\sum_{j=1}^{N}\sum_{b=1}^{Q_j}J_{ij}^{ab}x_{jb}(t)+B_i(t)+I_{ia}\right],$$
$$i=1,\ldots,N\ ,\quad a=1,\ldots,Q_i \qquad (11a)$$

or the N_{tot} discrete-time equations

$$x_{ia}(t+1)=F_{ia}\left[\sum_{j=1}^{N}\sum_{b=1}^{Q_j}J_{ij}^{ab}x_{jb}(t)+B_i(t)+I_{ia}\right],$$
$$i=1,\ldots,N\ ,\quad a=1,\ldots,Q_i \qquad (11b)$$

with parallel update. The interconnection matrix J_{ij}^{ab}, which couples the output of neuron b in cluster j to the input of neuron a in cluster i, is assumed to be real valued and to satisfy the symmetry condition

$$J_{ij}^{ab}=J_{ji}^{ba}\ . \qquad (12)$$

The N biases $B_i(t)$ enforce the competitive constraint (2) for each cluster. The N_{tot} external biases I_{ia} are time independent and may differ for each cluster i and neuron a. We will refer to networks that obey Eqs. (11) as *competitive networks.*

Because the competitive constraint restricts the Q_i neuron outputs in cluster i to a space of dimension Q_i-1, the dynamical systems (11) lie in a space of dimension $N_{\text{tot}}-N$. Thus, of the $N_{\text{tot}}(N_{\text{tot}}+1)/2$ independent elements of the symmetric matrix J_{ij}^{ab}, only $(N_{\text{tot}}-N)(N_{\text{tot}}-N+1)/2$ are necessary, and it is possible to impose up to $N_{\text{constr}}\equiv N(2N_{\text{tot}}-N+1)/2$ constraints on the interconnections without altering a network's trajectories in state space [36]. One useful choice of constraints is to require that, for $N(N+1)/2$ constants $J_{ij}=J_{ji}$,

$$\frac{1}{Q_i}\sum_{a=1}^{Q_i}J_{ij}^{ab}=\frac{1}{Q_j}\sum_{b=1}^{Q_j}J_{ij}^{ab}=J_{ij}\ . \qquad (13)$$

Equation (13) constrains the eigenvectors of the interconnection matrix either to lie in the same $(N_{\text{tot}}-N)$-dimensional space as the dynamical systems (11) or else to be orthogonal to this space. (The eigenvalues and eigenvectors of the interconnection matrix, which has four indices, are found by flattening it into an $N_{\text{tot}}\times N_{\text{tot}}$ matrix with two indices.) A simple counting argument shows that Eq. (13) imposes exactly N_{constr} constraints on the interconnections. For the Q_iQ_j interconnections J_{ij}^{ab} between two clusters $i\neq j$, Eq. (13) imposes Q_i+Q_j-1 constraints (the number of independent column and row sums of a $Q_i\times Q_j$ matrix). For the $Q_i(Q_i+1)/2$ independent interconnections J_{ii}^{ab} within a cluster i, Eq. (13) imposes Q_i constraints (the number of independent column and row sums of a $Q_i\times Q_i$ symmetric matrix). Thus the total number of constraints is

$$\sum_{i=1}^{N}\sum_{j>i=1}^{N}(Q_i+Q_j-1)+\sum_{i=1}^{N}Q_i=N_{\text{constr}}\ . \qquad (14)$$

An arbitrary, symmetric interconnection matrix J_{ij}^{ab} can be made to satisfy Eq. (13) by applying the symmetry-preserving transformation

$$J_{ij}^{ab}\rightarrow J_{ij}^{ab}+J_{ij}-\frac{1}{Q_i}\sum_{r=1}^{Q_i}J_{ij}^{rb}-\frac{1}{Q_j}\sum_{s=1}^{Q_j}J_{ij}^{as}+\frac{1}{Q_iQ_j}\sum_{r=1}^{Q_i}\sum_{s=1}^{Q_j}J_{ij}^{rs}\ . \qquad (15)$$

This transformation leaves the network dynamics (11) unchanged if in addition each neuron transfer function $F_{ia}(z)$ is shifted horizontally by an amount that depends on i and a that we now derive. Applying (15) transforms the neuron inputs $h_{ia}(t)$ by

$$h_{ia}(t)=\sum_{j=1}^{N}\sum_{b=1}^{Q_j}J_{ij}^{ab}x_{jb}(t)+I_{ia}$$
$$\rightarrow\sum_{j=1}^{N}\sum_{b=1}^{Q_j}\left[J_{ij}^{ab}-\frac{1}{Q_i}\sum_{r=1}^{Q_i}J_{ij}^{rb}\right]\left[x_{jb}(t)-\frac{R_j}{Q_j}\right]+\sum_{j=1}^{N}J_{ij}R_j+I_{ia}\ . \qquad (16)$$

Of the four new terms appearing on the right-hand side of Eq. (16) as a result of the transformation, three are independent of a and so contribute the same bias to the input of each neuron in cluster i, which is canceled by the bias $B_i(t)$. The other term does depend on a, and to cancel it the transfer function $F_{ia}(z)$ must be shifted:

$$F_{ia}(z)\rightarrow F_{ia}\left[z+\sum_{j=1}^{N}\sum_{b=1}^{Q_j}J_{ij}^{ab}\frac{R_j}{Q_j}\right]\ . \qquad (17)$$

The freedom to choose the constants J_{ij} can be useful in hardware applications, since a well-chosen set of J_{ij} can, for example, reduce the number of interconnections or make the interconnections all positive. In the rest of this paper, we will assume that the transformation (15) has been carried out with arbitrary values of J_{ij}.

B. Competition and Potts networks

When competitive clusters are configured as analog winner-take-all units, they can be viewed as a generalization of Potts spins [31]. A Potts spin has Q real-valued inputs and an output that takes on one of Q possible discrete values; at each time step, the output state is chosen to correspond with the largest input. In the limit $\gamma\rightarrow\infty$, the analog winner-take-all competitive cluster discussed in Sec. II reduces to a Potts spin. Moreover, for finite γ, the function appearing on the right-hand side of Eq. (6) describes the thermally averaged state of a Potts spin at temperature $T=1/\gamma$, just as the hyperbolic tangent function describes the thermally averaged state of an Ising spin.

However, the analogy between gain in deterministic, analog systems and temperature in stochastic, discrete-state systems is not exact. The mean-field treatment of finite-temperature Potts systems [50,51], like that of finite-temperature Ising systems [52], yields a term known as the reaction field acting on each spin. By design, no reaction field appears in the update equations (11) for competitive networks. These issues have been discussed elsewhere [25,27,28,53].

The computational abilities of networks of Potts spins have been studied by a number of authors [32–37]. Competitive clusters are considerably more general than Potts spins because they are analog rather than digital and because they can calculate other functions of their inputs besides the winner-take-all function.

IV. GLOBAL STABILITY ANALYSIS OF COMPETITIVE NETWORKS

In this section, we use a Liapunov function approach to analyze the dynamics of competitive networks. We show that, for symmetric interconnection matrices and the broad class of transfer functions described in Sec. II, (i) the attractors of competitive networks with continuous-time updating are fixed points; (ii) the attractors of competitive networks with discrete-time, parallel updating are either fixed points or period-two limit cycles; and (iii) period-two limit cycles are eliminated from discrete-time networks, leaving only fixed-point attractors, when the cluster gains are sufficiently reduced as described below. The analysis of discrete-time, parallel-update networks follows that of Ref. [23].

A. Continuous-time updating

We first consider continuous-time competitive networks, whose time evolution is described by Eq. (13a), and show that they have only fixed-point attractors. We construct the function

$$L(t)=-\tfrac{1}{2}\sum_{i,j=1}^{N}\sum_{a=1}^{Q_i}\sum_{b=1}^{Q_j}J_{ij}^{ab}x_{ia}(t)x_{jb}(t) -\sum_{i=1}^{N}\sum_{a=1}^{Q_i}I_{ia}x_{ia}(t)+\sum_{i=1}^{N}\sum_{a=1}^{Q_i}G_{ia}(x_{ia}(t)) , \tag{18}$$

where $G_{ia}(x)$ is the integral of the transfer function inverse:

$$G_{ia}(x)\equiv\int_{x_0}^{x}F_{ia}^{-1}(z)dz . \tag{19}$$

The quantity x_0 is an arbitrary constant. The function $L(t)$ and similar functions have been used by a number of authors to study the dynamics and stability of analog networks [20–24]. We prove that $L(t)$ is a Liapunov function for continuous-time competitive networks by showing (i) that the derivative of $L(t)$ with respect to time is always less than or equal to zero, and (ii) that $L(t)$ is bounded below. Using the symmetry of the interconnection matrix, the time derivative is

$$\frac{dL(t)}{dt}=\sum_{i=1}^{N}\sum_{a=1}^{Q_i}\frac{dx_{ia}(t)}{dt}\left[-\sum_{j=1}^{N}\sum_{b=1}^{Q_j}J_{ij}^{ab}x_{jb}(t) -I_{ia}+G'_{ia}(x_{ia}(t))\right] , \tag{20}$$

where $G'_{ia}(x)=F_{ia}^{-1}(x)$ is the derivative of $G_{ia}(x)$. The first two terms in square brackets can be rewritten using Eq. (13a), with the result

$$\frac{dL(t)}{dt}=\sum_{i=1}^{N}\sum_{a=1}^{Q_i}\frac{dx_{ia}(t)}{dt}\left[-F_{ia}^{-1}\left[x_{ia}(t)+\frac{dx_{ia}(t)}{dt}\right] +F_{ia}^{-1}(x_{ia}(t))+B_i(t)\right] . \tag{21}$$

The sum over the third term in square brackets in Eq. (21) vanishes, since

$$\sum_{a=1}^{Q_i}\frac{dx_{ia}(t)}{dt}=\frac{d}{dt}\sum_{a=1}^{Q_i}x_{ia}(t)=\frac{dR_i}{dt}=0 . \tag{22}$$

Equation (21) may therefore be written

$$\frac{dL(t)}{dt}=\sum_{i=1}^{N}\sum_{a=1}^{Q_i}\frac{dx_{ia}(t)}{dt}\left[-F_{ia}^{-1}\left[x_{ia}(t)+\frac{dx_{ia}(t)}{dt}\right] +F_{ia}^{-1}(x_{ia}(t))\right] . \tag{23}$$

As long as each transfer function increases monotonically, the quantity in square brackets in Eq. (23) always has the opposite sign of $dx_{ia}(t)/dt$. Thus

$$\frac{dL(t)}{dt}\le 0 , \tag{24}$$

with equality holding only when $dx_{ia}(t)/dt=0$ for all clusters i and neurons a, implying that the network has reached a fixed-point attractor. Furthermore, because all transfer functions in a cluster asymptotically increase less than linearly either for large negative values of their arguments or for large positive values of their arguments, or for both, the third sum in Eq. (18) increases more than quadratically when the neuron outputs are large in magnitude. The third sum therefore dominates the other two sums in $L(t)$, which are quadratic and linear in the neuron outputs, causing $L(t)$ to be bounded below. The result (24) and the boundedness of $L(t)$ imply that $L(t)$ is a Liapunov function of continuous-time competitive networks: the networks seek out the local minima of $L(t)$ as they evolve in time. Continuous-time competitive networks therefore have only fixed-point attractors.

B. Discrete-time, parallel updating

Competitive networks with discrete-time, parallel updating, whose time evolution is given by Eq. (13b), may have both fixed points and period-two limit cycles. To prove this, we construct the function [20–24]

$$E(t)=-\sum_{i,j=1}^{N}\sum_{a=1}^{Q_i}\sum_{b=1}^{Q_j}J_{ij}^{ab}x_{ia}(t)x_{jb}(t-1) -\sum_{i=1}^{N}\sum_{a=1}^{Q_i}I_{ia}[x_{ia}(t)+x_{ia}(t-1)] +\sum_{i=1}^{N}\sum_{a=1}^{Q_i}[G_{ia}(x_{ia}(t))+G_{ia}(x_{ia}(t-1))] , \tag{25}$$

where $G_{ia}(x)$ is defined as in Eq. (19) and the time t is now discrete. We prove that $E(t)$ is a Liapunov function for discrete-time, parallel-update competitive networks

by showing (i) that the change in $E(t)$ between successive time steps, $\Delta E(t) \equiv E(t+1) - E(t)$, is a nonincreasing function of time, and (ii) that $E(t)$ is bounded below. Using the update equation (13b) and the symmetry condition (12), $\Delta E(t)$ can be written as

$$\Delta E(t) = -\sum_{i=1}^{N} \sum_{a=1}^{Q_i} F_{ia}^{-1}(x_{ia}(t+1)) \Delta_2 x_{ia}(t) - \sum_{i=1}^{N} B_i(t) \sum_{a=1}^{Q_i} \Delta_2 x_{ia}(t) + \sum_{i=1}^{N} \sum_{a=1}^{Q_i} [G_{ia}(x_{ia}(t+1)) - G_{ia}(x_{ia}(t-1))] , \tag{26}$$

where $\Delta_2 x_{ia}(t) \equiv x_{ia}(t+1) - x_{ia}(t-1)$ is the change in $x_{ia}(t)$ between two time steps. The second term in Eq. (26) is identically zero, since

$$\sum_{a=1}^{Q_i} \Delta_2 x_{ia}(t) = \sum_{a=1}^{Q_i} x_{ia}(t+1) - \sum_{a=1}^{Q_i} x_{ia}(t-1) = R_i - R_i = 0 . \tag{27}$$

Thus Eq. (26) becomes

$$\Delta E(t) = -\sum_{i=1}^{N} \sum_{a=1}^{Q_i} F_{ia}^{-1}(x_{ia}(t+1)) \Delta_2 x_{ia}(t) + \sum_{i=1}^{N} \sum_{a=1}^{Q_i} [G_{ia}(x_{ia}(t+1)) - G_{ia}(x_{ia}(t-1))] . \tag{28}$$

We now construct an upper bound for the contribution from each cluster to the last term in Eq. (28). The upper bound is

$$\sum_{a=1}^{Q_i} [G_{ia}(x_{ia}(t+1)) - G_{ia}(x_{ia}(t-1))] \leq \sum_{a=1}^{Q_i} G'_{ia}(x_{ia}(t+1)) \Delta_2 x_{ia}(t) , \tag{29}$$

where $G'_{ia}(x)$ is the derivative of $G_{ia}(x)$. Equation (29), which states that the Q_i-dimensional surface $\sum_a G_{ia}(x_{ia})$ lies everywhere on or above its tangent planes, holds because the neuron transfer functions increase monotonically. Equation (29) is illustrated in Fig. 4 for a cluster of $Q_i = 2$ neurons with transfer functions $F_{i1}(z) = F_{i2}(z) = \exp(\gamma z)$ and with $R_i = 1$. Figure 4(a) depicts the surface $G_{i1} + G_{i2}$. The competitive constraint $x_{i1} + x_{i2} = 1$ restricts the possible values of $G_{i1} + G_{i2}$ to the dashed curve lying on the surface. Also shown is the tangent plane and a tangent paraboloid (to be used below) at a particular point (x_{i1}, x_{i2}) such that $x_{i1} + x_{i2} = 1$. Figure 4(b) is a detailed view of the tangent plane and the dashed curve; the left-hand side of Eq. (29) appears as the quantity A and the right-hand side as the quantity B.

Combining Eqs. (28) and (29) and using the result $G'_{ia}(x) = F_{ia}^{-1}(x)$ leads to

$$\Delta E(t) \leq 0 , \tag{30}$$

where equality holds when $\Delta_2 x_{ia}(t) = 0$ for all i and a, which implies that the network has reached either a fixed-point or a period-two attractor. The result (30) and the fact that $E(t)$, like $L(t)$, is bounded below, together prove that $E(t)$ is a Liapunov function of competitive networks with discrete-time, parallel updating, and that all attractors of these networks are either fixed points or period-two limit cycles.

We now show that period-two limit cycle attractors can be eliminated from discrete-time, parallel-update competitive networks by reducing the cluster gains sufficiently. We again consider the function $L(t)$ of Eq. (18), only this time applied to the discrete-time, parallel-update competitive network, and show that it is a Liapunov function for this network when all cluster gains are sufficiently small. We consider the change in $L(t)$ between successive time steps, $\Delta L(t) \equiv L(t+1) - L(t)$. Using the update equation and the symmetry of the interconnection matrix, $\Delta L(t)$ can be written

$$\Delta L(t) = -\tfrac{1}{2} \sum_{i,j=1}^{N} \sum_{a=1}^{Q_i} \sum_{b=1}^{Q_j} J_{ij}^{ab} \Delta x_{ia}(t) \Delta x_{jb}(t) - \sum_{i=1}^{N} \sum_{a=1}^{Q_i} F_{ia}^{-1}(x_{ia}(t+1)) \Delta x_{ia}(t) - \sum_{i=1}^{N} B_i(t) \sum_{a=1}^{Q_i} \Delta x_{ia}(t) + \sum_{i=1}^{N} \sum_{a=1}^{Q_i} [G_{ia}(x_{ia}(t+1)) - G_{ia}(x_{ia}(t))] , \tag{31}$$

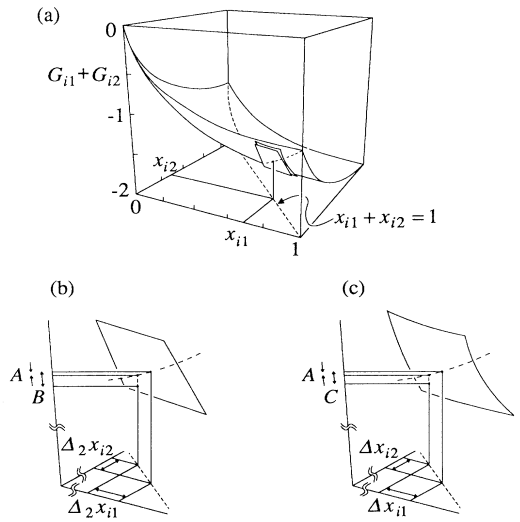

FIG. 4. Illustration of inequalities (29) and (33) for an analog winner-take-all cluster i containing $Q_i=2$ neurons with transfer functions $F_{i1}(z)=F_{i2}(z)=\exp(z)$ and $R_i=1$. (a) The surface $G_{i1}(x_{i1})+G_{i2}(x_{i2})$ as a function of neuron outputs x_{i1} and x_{i2}. Competition constrains the system to dashed curve lying on surface. Also shown are the tangent plane and a tangent paraboloid at a particular point (x_{i1},x_{i2}). Parabola curvature $1/\beta_i$ equals greatest curvature of dashed curve. (b) Detail of tangent plane. Equation (29) is represented by the inequality $A\le B$. (c) Detail of tangent parabola. Equation (33) is represented by the inequality $A\le C$.

where $\Delta x_{ia}(t)=x_{ia}(t+1)-x_{ia}(t)$ is the change in x_{ia} between consecutive time steps. As in Eq. (27), the third term on the right-hand side of (31) is zero, so that

$$\Delta L(t)=-\tfrac{1}{2}\sum_{i,j=1}^{N}\sum_{a=1}^{Q_i}\sum_{b=1}^{Q_j}J_{ij}^{ab}\Delta x_{ia}(t)\Delta x_{jb}(t)-\sum_{i=1}^{N}\sum_{a=1}^{Q_i}F_{ia}^{-1}(x_{ia}(t+1))\Delta x_{ia}(t)+\sum_{i=1}^{N}\sum_{a=1}^{Q_i}[G_{ia}(x_{ia}(t+1))-G_{ia}(x_{ia}(t))]\ . \tag{32}$$

We now construct an inequality relating the last term in Eq. (32) to $\Delta x_{ia}(t)$. This inequality is similar to (29) but includes a term quadratic in $\Delta x_{ia}(t)$:

$$\sum_{a=1}^{Q_i}[G_{ia}(x_{ia}(t+1))-G_{ia}(x_{ia}(t))]\le\sum_{a=1}^{Q_i}G_{ia}'(x_{ia}(t+1))\Delta x_{ia}(t)-\frac{1}{2\beta_i}\sum_{a=1}^{Q_i}(\Delta x_{ia}(t))^2\ . \tag{33}$$

The quantity β_i is the gain of cluster i. We define the cluster gain to be the smallest number β_i such that the inequality

$$\frac{1}{\beta_i}\sum_{a=1}^{Q_i}(\Delta x_{ia}(t))^2\le\sum_{a=1}^{Q_i}G_{ia}''(x_{ia})(\Delta x_{ia}(t))^2\ , \tag{34}$$

where $G_{ia}''(x)$ is the second derivative of $G_{ia}(x)$, is satisfied for any possible values of $x_{ia}(t)$ and $\Delta x_{ia}(t)$ subject to the competitive constraint (2b). Note that, if there were no constraints on $x_{ia}(t)$ and $\Delta x_{ia}(t)$, β_i would equal the steepest slope among all the transfer functions in cluster i. However, because competition constrains the sums of $x_{ia}(t)$ and $\Delta x_{ia}(t)$ in each cluster to equal R_i and zero, respectively, β_i takes on a value less than or equal to the steepest slope. This value is given by

$$\frac{1}{\beta_i}=\min_{\{z_a,\epsilon_a\}}\sum_{a=1}^{Q_i}\epsilon_a^2\left[\frac{dF_{ia}^{-1}(z)}{dz}\right]_{z=z_a}\ . \tag{35}$$

The minimization in (35) is over the dummy variables z_a and ϵ_a, $a=1,\ldots,Q_i$, subject to the constraints

$$\sum_{a=1}^{Q_i}z_a=R_i\ ,\quad\sum_{a=1}^{Q_i}\epsilon_a=0\ ,\quad\sum_{a=1}^{Q_i}\epsilon_a^2=1\ , \tag{36}$$

and to the restriction that each z_a lie in the range of the transfer function $F_{ia}(z)$. In cases for which the minimization (35) is difficulty to carry out, a useful upper bound for β_i is the steepest slope of any of the transfer functions $F_{ia}(z)$, $a=1,\ldots,Q_i$, over the range to which the competitive constraint (2b) restricts them. For clusters in which $R_i=1$ and all neurons have the exponential transfer function of Eq. (5), Eqs. (35) and (36) give the result $\beta=\gamma/2$ independent of Q_i.

The inequality (33) and the geometrical meaning of the cluster gain can be understood by referring again to Fig. 4. The tangent paraboloid in Fig. 4(a) has curvature $1/\beta_i$. The minimization procedure of Eqs. (35) and (36) sets β_i equal to the greatest curvature of the dashed curve lying on the surface $G_{i1}+G_{i2}$ (recall that competition constrains the system to the dashed curve). Thus the dashed curve lies everywhere on or above the paraboloid when the paraboloid is tangent to it. Figure 4(c) is a detailed view of the tangent paraboloid and the dashed curve; the left-hand side of Eq. (33) appears as the quantity A and the right-hand side as the quantity C. For a general cluster i, β_i is the greatest curvature of the (Q_i-1)-dimensional region of the surface $\sum_a G_{ia}(x_{ia})$ to which the neurons of the cluster are restricted. The vector z_a and the unit vector ϵ_a produced by the minimization (35) indicate, respectively, the location and the direction of the curvature β_i.

Equations (32) and (33) and the result $G_i'(x)=F_i^{-1}(x)$ lead to

$$\Delta L(t)\le-\tfrac{1}{2}\sum_{i,j=1}^{N}\sum_{a=1}^{Q_i}\sum_{b=1}^{Q_i}\left[J_{ij}^{ab}+\delta_{ij}\delta_{ab}\frac{1}{\beta_i}\right]\times\Delta x_{ia}(t)\Delta x_{jb}(t)\ , \tag{37}$$

where $\delta_{ij}=1$ for $i=j$ and 0 otherwise. If the matrix $M_{ij}^{ab}=(J_{ij}^{ab}+\delta_{ij}\delta_{ab}\beta_i^{-1})$ is positive definite, then the inequality

$$\Delta L(t)\le 0 \tag{38}$$

is satisfied. Equality holds in (38) only when $\Delta x_{ia}(t)=0$,

implying that the network has reached a fixed-point attractor. The requirement that M_{ij}^{ab} be positive definite leads to the stability criterion

$$\frac{1}{\beta_i} > -\lambda_{\min} , \quad i=1,\ldots,N , \tag{39}$$

where $\lambda_{\min}$ is the smallest or most negative eigenvalue of the interconnection matrix J_{ij}^{ab}, and β_i is the gain of cluster i. Equation (38) and the boundedness of $L(t)$ prove that, when the cluster gains satisfy (39), $L(t)$ is a Liapunov function of discrete-time, parallel-update competitive networks and the only attractors are fixed points. Thus networks with interconnection matrices for which $\lambda_{\min} \geq 0$ do not have period-two limit cycles, while those with interconnection matrices for which $\lambda_{\min} < 0$ can have period-two limit cycles if anyone of the cluster gains exceeds $1/|\lambda_{\min}|$.

The stability results derived above serve as guidelines for designing competitive networks that compute by relaxing to fixed-point attractors. Specifically, continuous-time networks are guaranteed to converge only to fixed points, while discrete-time, parallel-update networks converge only to fixed points when the cluster gains are chosen to satisfy the stability criterion (39). In the next section, we will apply the stability criterion to discrete-time, parallel-update networks that use fixed-point computation to perform image processing tasks. We will see that the stability criterion provides a reliable upper limit on the cluster gain when convergence to fixed points is desired.

V. TWO-DIMENSIONAL COMPETITIVE NETWORKS FOR IMAGE PROCESSING

In this section, we demonstrate how competition can lead to desirable computational abilities in a specific application. The application we investigate, depicted schematically in Fig. 5, is a two-dimensional network in which competitive clusters detect and classify localized features in a visual scene. We show analytically how the stability criterion depends on interconnection matrix parameters such as local connectivity, and we present numerical data supporting these results. The main point is to demonstrate that competitive networks are well suited for solving feature-extraction and pattern-classification problems.

Two-dimensional neural networks are useful in artificial vision applications because they allow efficient mapping of an image onto a network and because they are well suited for implementation in very-large-scale integrated circuits. Competition can be employed in two-dimensional image-processing networks in several ways. A competitive cluster can be assigned to each image pixel to extract information that can be categorized into discrete classes, such as color, depth, or gray-scale intensity. Competitive clusters can also be assigned to pixel groups to extract and classify image features locally, as in the neocognitron [30], and to communicate information about these features, rather than about individual pixels, to nearby image regions. In this latter case, depicted in Fig. 5, each neuron represents a different feature, such as an edge, corner, texture, or other image primitive. Neurons receive as input a weighted average of the pixel values in a localized image region. The weights, which encode the different features, can be chosen by hand or learned through a supervised perceptron learning rule or an unsupervised competitive learning rule [12–15].

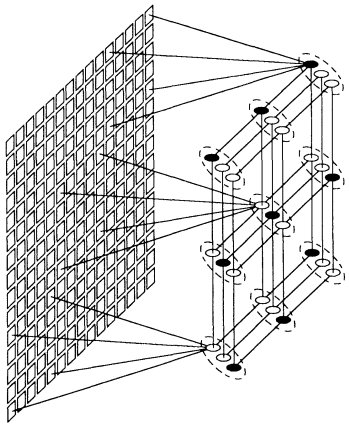

FIG. 5. Two-dimensional competitive network for image processing. Open circles denote neurons; lines denote symmetric neuron interconnections; dashed ellipses delineate clusters of competing neurons. Neurons receive input from a localized region of a two-dimensional pixel array and communicate through interconnection matrix (40) with neurons in neighboring clusters on a square lattice. For clarity only some of pixel array inputs are shown. All clusters have same number Q of neurons, same transfer functions $F_a(z)$, $a=1,\ldots,Q$, and same value R of the constant appearing in the competitive constraint (2).

We consider two-dimensional competitive networks configured to recognize image regions where similar features occur. In these networks, spatial domains of neuron activity form that correspond to regions of similar color, depth, or texture in the image. A competitive cluster is located at each of the N vertices of a square lattice. The clusters are identical in that they have the same number $Q_i=Q$ of neurons, the same neuron transfer functions $F_{ia}(z)=F_a(z)$, and the same value $R_i=R$ of the constant appearing in the competitive constraint. Thus each cluster has the same cluster gain β. The interconnection matrix is constructed so that neuron a of a given cluster communicates through an excitatory connection with neuron a in the z clusters nearest to it; the cases $z=4$, 8, and 12 are shown in Fig. 6. In addition, each neuron can have a self-coupling connection.

The interconnection matrix is written as

$$J_{ij}^{ab} = \begin{cases} \sigma\delta_{ab}/(|\sigma|+z) & \text{if } i=j \\ \delta_{ab}/(|\sigma|+z) & \text{if } i,j \text{ neighbors} \\ 0 & \text{otherwise} . \end{cases} \tag{40}$$

In (40), z is the connectivity, or number of neighboring clusters in a neighborhood, and σ determines the strength of the self-connection relative to the other connections. This interconnection scheme satisfies the con-

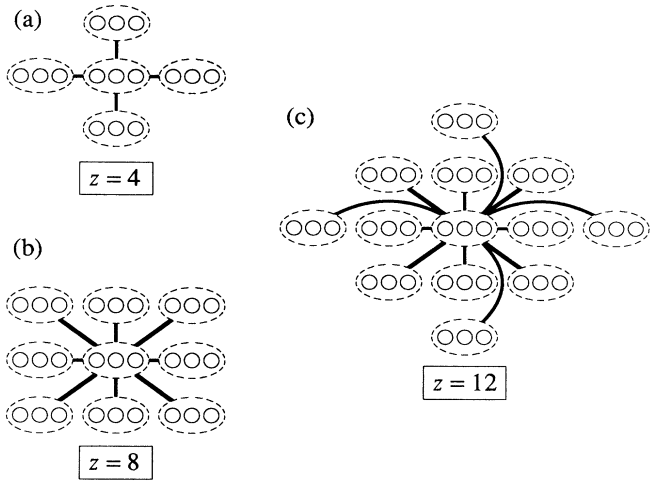

FIG. 6. Interconnection schemes on a two-dimensional square lattice, with (a) $z=4$, (b) $z=8$, and (c) $z=12$ neighboring clusters. Circles denote neurons, dashed ellipses denote clusters, and heavy lines schematically denote symmetric interconnections among all neurons within the clusters they connect. Each cluster has identical interconnections; for clarity, only those for one cluster are shown.

straints (13) with

$$J_{ij}=\begin{cases}\sigma/(|\sigma|+z) & \text{if } i=j \\ 1/(|\sigma|+z) & \text{if } i,j \text{ neighbors} \\ 0 & \text{otherwise .}\end{cases} \tag{41}$$

Note that allowing the constants J_{ij} to be nonzero reduces wiring requirements by eliminating connections between neurons with $a\neq b$, i.e., neurons representing different features in neighboring clusters.

As an example of how these networks perform, Fig. 7 shows a fixed-point attractor of a network containing $N=40\times 40=1600$ competitive clusters with connectivity $z=4$. All clusters implement the analog winner-take-all function with $Q=4$ neurons per cluster, $R=1$, and neuron transfer functions given by Eq. (5). The network has wrap-around boundary conditions. Figure 7 was generated by starting the network from random initial conditions $x_{ia}(0)=\delta_{a,b_i}$, where each b_i, $i=1,\ldots,N$, is an integer chosen randomly and without bias from the set $\{1,\ldots,Q\}$, and iterating Eq. (11b) until convergence to an attractor. The fixed point shows how neurons representing the same feature tend to form domains in a way that can be useful in image processing. Domain size can be regulated by the neuron gain γ; the average domain size shrinks as gain increases [8–10].

The eigenvalue spectra of the interconnection matrices in Eq. (40) can be calculated using standard techniques for finding classical phonon dispersion spectra [54]. In the limit of large N, the minimum eigenvalues are [55]

$$\lambda_{\min}^{(4)}=\frac{\sigma-4}{|\sigma|+4}\ ,\quad \lambda_{\min}^{(8)}=\frac{\sigma-4}{|\sigma|+8}\ ,\quad \lambda_{\min}^{(12)}=\frac{\sigma-\frac{13}{3}}{|\sigma|+12}\ , \tag{42}$$

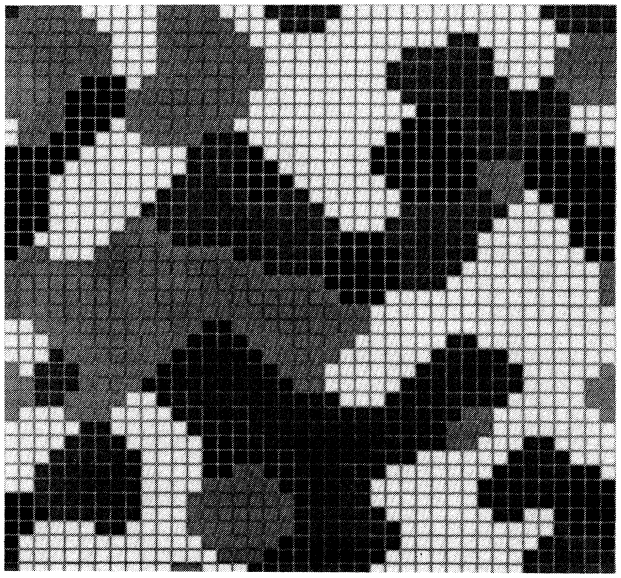

FIG. 7. Fixed point attractor of a two-dimensional competitive network, showing how neurons representing the same feature tend to form domains in a way that can be useful in image-processing applications. Each square represents one cluster; side length is proportional to output of winning neuron, and shading indicates which of the neurons in each cluster wins. Network has $N=40\times 40=1600$ analog winner-take-all clusters with $Q=4$ neurons per cluster, transfer functions $F(z)=\exp(\gamma z)$, and $R=1$. Cluster gain is $\beta=20$, self-coupling is $\sigma=1$, and connectivity is $z=8$. Boundary conditions are wrap-around.

where the superscript in parentheses indicates the number z of neighbors for the lattices of Fig. 6. Note that the minimum eigenvalues are independent of Q. These minimum eigenvalues are used in Fig. 8 to show how the stability criterion (39) depends on the cluster gain β and neuron self-connection σ for the cases $z=4$, 8, and 12.

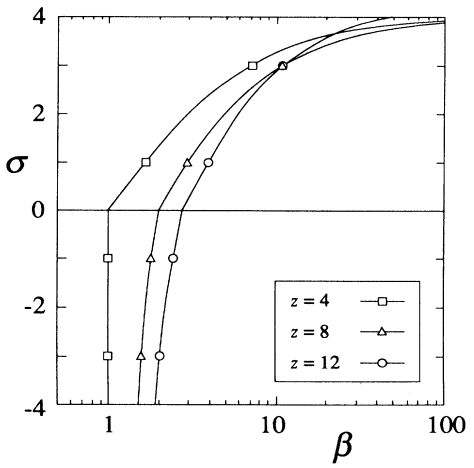

FIG. 8. Stability criterion (39) for two-dimensional competitive networks with self-coupling σ and cluster gain β for various values of connectivity z. Stability criterion is violated to right of each curve. Curves are independent of the number Q of neurons in each cluster.

For each value of cluster gain, the curves in Fig. 8 give the value of σ for which $1/\beta = -\lambda_{\min}$. The curves are independent of the number Q of neurons per cluster and of the particular form of the neuron transfer functions $F_a(z)$.

Numerical tests carried out on computer-generated two-dimensional competitive networks indicate that limit cycles often do not appear until the cluster gain is somewhat higher than the value given by the stability criterion. The reason is that Eqs. (35) and (36) define cluster gain as the greatest curvature at one point and in one direction on a many-dimensional surface, and the trajectories of a particular network may not pass through that point in that direction. This is in contrast to standard analog networks, in which neurons experience the high-gain regions of their transfer functions because the functions are one dimensional.

We illustrate this point by considering two-dimensional competitive networks with $z=4$ and with analog winner-take-all clusters at each lattice site. The clusters have $R=1$ and neuron transfer functions given by Eq. (5), so that the cluster gain is $\beta=\gamma/2$. The eigenvector associated with $\lambda_{\min}$, and hence the structure of the first period-two limit cycle to appear as cluster gain is increased, are known [55]. The limit cycle consists of two neurons (neurons 1 and 2, for example) in each cluster that alternately oscillate between outputs x_1 and x_2, while the $(Q-2)$ other neurons have constant outputs x_3. Spatially, the limit cycle has a checkerboard pattern, as shown in the inset of Fig. 9: when neurons 1 and 2 in cluster i have outputs x_1 and x_2, neurons 1 and 2 in the four nearest-neighbor clusters have outputs x_2 and x_1. If such a limit cycle exists at given values of the neuron gain γ and the self-connection σ, its values of x_1, x_2, and x_3 are

$$x_1=\frac{f_1}{f_1+f_2+(Q-2)f_3}\ ,$$

$$x_2=\frac{f_2}{f_1+f_2+(Q-2)f_3}\ , \tag{43}$$

$$x_3=\frac{f_3}{f_1+f_2+(Q-2)f_3}\ ,$$

where

$$f_1=\exp[\gamma(\sigma x_2+4x_1)/(|\sigma|+4)]\ ,$$

$$f_2=\exp[\gamma(\sigma x_1+4x_2)/(|\sigma|+4)]\ , \tag{44}$$

$$f_3=\exp[\gamma(\sigma x_3+4x_3)/(|\sigma|+4)]\ .$$

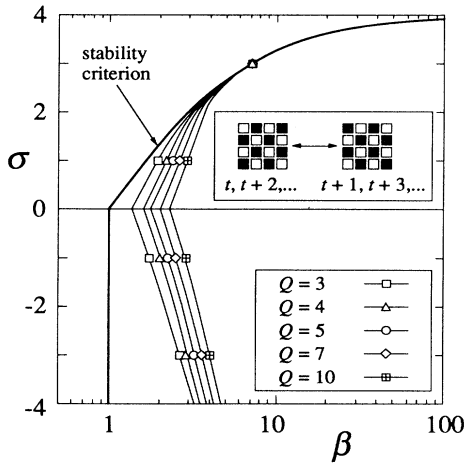

FIG. 9. Comparison of stability criterion with gain at which period-two limit cycles appear for two-dimensional competitive networks with connectivity $z=4$. Heavy curve shows the stability criterion (39). Light curves show, for various values of Q, gain at which period-two limit cycle given by Eqs. (43) and (44) first appears in networks in which $F(z)=\exp(\gamma z)$ and $R=1$, with cluster gain $\beta=\gamma/2$. Inset depicts limit cycle structure using same scheme as in Fig. 7. When neurons 1 and 2 in cluster i have outputs x_1 and x_2, neurons 1 and 2 in the four nearest-neighbor clusters have outputs x_2 and x_1.

In Fig. 9, we compare the stability criterion with the value of gain at which the period-two limit cycle given by Eqs. (43) and (44) first appears for various values of Q. For $\sigma \gtrsim 3$, the limit cycle appears as soon as the stability criterion is violated, while for $\sigma \lesssim 3$, it appears at Q-dependent values of gain that are larger than the gain at which the stability criterion is violated. We emphasize that this analysis improves upon the stability criterion at the expense of generality: Eqs. (43) and (44) depend on the particular form of the neuron transfer functions and on the structure of the limit cycle corresponding to $\lambda_{\min}$, while the stability criterion (39) depends only on the cluster gain and on $\lambda_{\min}$. In general, the limit cycle structure may not be known, and Eq. (39) provides a simple and reliable criterion guaranteeing convergence to a fixed point.

VI. SUMMARY

If neural-network implementations are to become prevalent, it is likely that network architectures will need to reflect the underlying structure of the problems they are to solve. In this paper, we have introduced a network architecture in which analog neurons compete within localized clusters. Local competition makes these networks particularly well suited for tasks such as feature extraction and pattern classification. Using a global stability analysis, we have shown that these networks converge only to fixed points with continuous-time updating and to either fixed points or period-two limit cycles with discrete-time, parallel updating. Furthermore, we have provided a stability criterion that guarantees convergence of discrete-time, parallel-update networks to fixed points

when the cluster gain is sufficiently reduced. We have discussed how clusters of competing neurons can be used in image processing applications, and we have shown through numerical tests that such networks operate reliably and in accordance with the stability criterion. We have also constructed a simple analog electronic circuit to demonstrate that these networks are easily implementable.

ACKNOWLEDGMENTS

We thank C. Marcus, A. Herz, R. Kühn, W. Yang, and A. Yuille for valuable discussions and correspondence. One of us (F.R.W.) acknowledges support from the U.S. Army Research Office. Research was supported in part by Contracts Nos. N00014-89-J-1023 and F49620-92-J-0466

[1] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Ann. Phys. (N.Y.) **173**, 30 (1987).
[2] I. Kanter and H. Sompolinsky, Phys. Rev. A **35**, 380 (1987).
[3] D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks* (Cambridge University Press, Cambridge, 1989).
[4] E. Domany, W. Kinzel, and R. Meir, J. Phys. A **22**, 2081 (1989).
[5] H.-U. Bauer and T. Geisel, Phys. Rev. A **42**, 2401 (1990).
[6] H. Sompolinsky, D. Golomb, and D. Kleinfeld, Phys. Rev. A **43**, 6990 (1991).
[7] M. A. Pires Idiart and A. Theumann, J. Phys. A **25**, 779 (1992).
[8] A. J. Noest, Phys. Rev. Lett. **63**, 1739 (1989).
[9] A. J. Noest, in *Statistical Mechanics of Neural Networks: Proceedings of the XIth Sitges Conference, Sitges, Barcelona, 1990,* edited by L. Garrido (Springer-Verlag, New York, 1990), p. 303.
[10] A. C. C. Coolen, in *Statistical Mechanics of Neural Networks: Proceedings of the XIth Sitges Conference, Sitges, Barcelona, 1990* (Ref. [9]), p. 381.
[11] A. Treves and D. J. Amit, J. Phys. A **21**, 3155 (1988); **22**, 2205 (1989); A. Treves, *ibid.* **23**, 2631 (1990).
[12] S. Grossberg, Biol. Cyber. **23**, 121 (1976); S. Grosberg, Cog. Sci. **11**, 23 (1987).
[13] T. Kohonen, *Self-Organization and Associative Memory* (Springer-Verlag, Berlin, 1988).
[14] D. E. Rumelhart and D. Zipser, in *Parallel Distributed Processing, Vol. 1,* edited by J. A. Feldman, P. J. Hayes, and D. E. Rumelhart (MIT Press, Cambridge, MA, 1986). p. 151.
[15] J. A. Hertz, A. S. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Reading, MA, 1991), Chap. 9.
[16] A. Bennett, J. Phys. A **22**, 2047 (1989).
[17] M. Schmutz and W. Banzhaf, Phys. Rev. Rev. A **45**, 4132 (1992).
[18] A. V. M. Herz, Phys. Rev. A **44**, 1415 (1991); A. V. M. Herz, Z. Li, and J. L. van Hemmen, Phys. Rev. Lett. **66**, 1370 (1991).
[19] P. C. Bressloff and J. G. Taylor, J. Phys. A **25**, 833 (1992).
[20] M. A. Cohen and S. Grossberg, IEEE Trans. SMC **13**, 815 (1983).
[21] J. J. Hopfield, Proc. Natl. Acad. Sci. U.S.A. **81**, 3008 (1984); J. J. Hopfield and D. W. Tank, Science **233**, 625 (1986).
[22] R. M. Golden, J. Math. Psych. **30**, 73 (1986).
[23] C. M. Marcus and R. M. Westervelt, Phys. Rev. A **40**, 501 (1989).
[24] F. Fogelman Soulie, C. Mejia, E. Goles, and S. Martinez, Complex Sys. **3**, 269 (1989).
[25] C. M. Marcus, F. R. Waugh, and R. M. Westervelt, Phys. Rev. A **41**, 3355 (1990).
[26] F. R. Waugh, C. M. Marcus, and R. M. Westervelt, Phys. Rev. Lett. **64**, 1986 (1990); Phys. Rev. A **43**, 3131 (1991).
[27] T. Fukai and M. Shiino, Phys. Rev. A **42**, 7459 (1990); M. Shiino and T. Fukai, J. Phys. A **23**, L1009 (1990).
[28] R. Kühn, S. Bös, and J. L. van Hemmen, Phys. Rev. A **43**, 2084 (1991).
[29] E. Goles-Chacc, F. Fogelman-Soulie, and D. Pellegrin, Disc. Appl. Math. **12**, 261 (1985); E. Goles and G. Y. Vichniac, in *Neural Networks for Computing,* AIP Conf. Proc. No. 151, edited by J. S. Denker (American Institute of Physics, New York, 1986), p. 165.
[30] K. Fukushima, Biol. Cyber. **36**, 193 (1980); Neural Networks **1**, 199 (1988).
[31] F. Y. Wu, Rev. Mod. Phys. **54**, 235 (1982).
[32] I. Kanter, Phys. Rev. A **37**, 2739 (1988).
[33] D. Bolle and P. Dupont, in *Statistical Mechanics of Neural Networks: Proceedings of the XIth Sitges Conference, Sitges, Barcelona, 1990* (Ref. [9]), p. 365; D. Bolle, P. Dupont, and J. van Mourik, J. Phys. A **24**, 1065 (1991); D. Bollé, P. Dupont, and B. Vinck, *ibid.* **25**, 2859 (1992); D. Bollé, P. Dupont, and J. Huyghebaert, Phys. Rev. A **45**, 4194 (1992); Physica A **185**, 363 (1992).
[34] P. A. Ferrari, S. Martinez, and P. Picco, J. Stat. Phys. **66**, 1643 (1992).
[35] H. Vogt and A. Zippelius, J. Phys. A **25**, 2209 (1992).
[36] J.-P. Nadal and A. Rau, J. Phys. I (Paris) **1**, 1109 (1991).
[37] G. M. Shim, D. Kim, and M. Y. Choi, Phys. Rev. A **45**, 1238 (1992).
[38] J. A. Feldman and D. H. Ballard, Cog. Sci. **6**, 205 (1982).
[39] J. Pankove, C. Radehaus, and K. Wagner, Electron. Lett. **26**, 349 (1990).
[40] R. Perfetti, IEE Proc. **137**, 353 (1990).
[41] L. G. Johnson and S. M. S. Jalaleddine, Electron. Lett. **27**, 957 (1991).
[42] D. Z. Anderson, in *Neural Networks for Computing* (Ref. [29]), 12; D. Z. Anderson, in *An Introduction to Neural and Electronic Networks,* edited by S. F. Zornetzer, J. L. Davis, and C. Lau (Academic, New York, 1990), p. 349; C. Bernkert and D. Z. Anderson, Phys. Rev. A **44**, 4633 (1991).
[43] R. P. Lippmann, IEEE Acous. Speech Signal Proc. **4**, 4 (1987).
[44] F. R. Waugh and R. M. Westervelt, following paper, Phys. Rev. E. **47**, 4537 (1993).
[45] D. Geiger and A. Yuille, Int. J. Comp. Vision **6**, 227 (1991).
[46] C. Peterson and B. Söderberg, Int. J. Neural Syst. **1**, 3 (1989).
[47] J. S. Bridle, in *Advances in Neural Information Processing Systems 3,* edited by D. S. Touretsky (Kaufmann, San Ma-

teo, CA, 1990), p. 211; J. S. Bridle and S. J. Cox, in *Advances in Neural Information Processing Systems 3,* edited by R. P. Lippmann, J. E. Moody, and D. S. Touretsky (Kaufmann, San Mateo, CA, 1991), p. 234.

[48] E. Majani, R. Erlanson, and Y. Abu-Mostafa, in *Advances in Neural Information Processing Systems 1,* edited by D. S. Touretzky (Kaufmann, San Mateo, CA, 1989), p. 634.

[49] S. M. Sze, *Physics of Semiconductor Devices,* 2nd ed. (Wiley, New York, 1981), p. 446; C. A. Mead, *Analog VLSI and Neural Systems* (Addison-Wesley, Reading, MA, 1989), p. 36.

[50] E. J. S. Lage and J. M. Nunes da Silva, J. Phys. C **17**, L593 (1984).

[51] D. J. Gross, I. Kanter, and H. Sompolinsky, Phys. Rev. Lett. **55**, 304 (1984).

[52] D. J. Thouless, P. W. Anderson, and R. G. Palmer, Philos. Mag. **35**, 593 (1977).

[53] H. Takayama and K. Nemoto, J. Phys.: Condens. Matter **2**, 1997 (1990); K. Nishimura, K. Nemoto, and H. Takayama, J. Phys. A **23**, 5915 (1990).

[54] O. Madelung, *Introduction to Solid State Theory* (Springer-Verlag, Berlin, 1981), p. 129.

[55] C. M. Marcus, F. R. Waugh, and R. M. Westervelt, in *Advances in Neural Information Processing Systems 3* (Ref. [47]), p. 331.

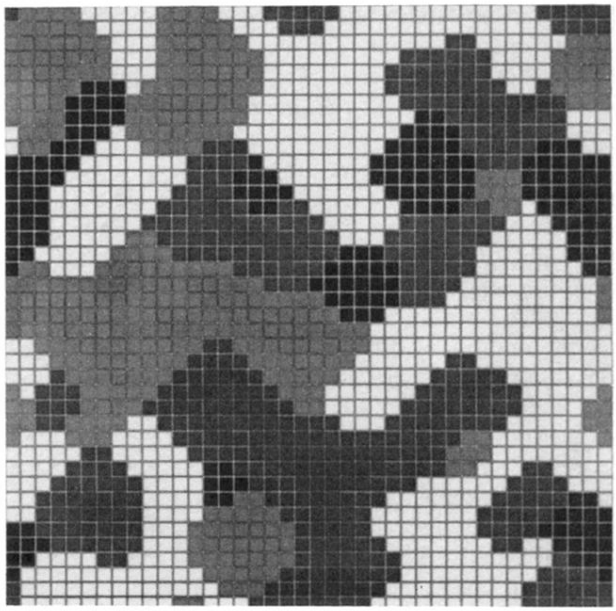

FIG. 7. Fixed point attractor of a two-dimensional competitive network, showing how neurons representing the same feature tend to form domains in a way that can be useful in image-processing applications. Each square represents one cluster; side length is proportional to output of winning neuron, and shading indicates which of the neurons in each cluster wins. Network has $N=40\times40=1600$ analog winner-take-all clusters with $Q=4$ neurons per cluster, transfer functions $F(z)=\exp(\gamma z)$, and $R=1$. Cluster gain is $\beta=20$, self-coupling is $\sigma=1$, and connectivity is $z=8$. Boundary conditions are wrap-around.