

RESEARCH ARTICLE

Computation in Dynamically Bounded Asymmetric Systems

Ueli Rutishauser^{1,2,3*}, Jean-Jacques Slotine⁴, Rodney Douglas⁵

1 Computation and Neural Systems, California Institute of Technology, Pasadena, California, United States of America, **2** Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, California, United States of America, **3** Departments of Neurosurgery, Neurology and Biomedical Sciences, Cedars-Sinai Medical Center, Los Angeles, California, United States of America, **4** Nonlinear Systems Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America, **5** Institute of Neuroinformatics, University and ETH Zurich, Zurich, Switzerland

* urut@caltech.edu



Abstract

Previous explanations of computations performed by recurrent networks have focused on symmetrically connected saturating neurons and their convergence toward attractors. Here we analyze the behavior of asymmetrical connected networks of linear threshold neurons, whose positive response is unbounded. We show that, for a wide range of parameters, this asymmetry brings interesting and computationally useful dynamical properties. When driven by input, the network explores potential solutions through highly unstable ‘expansion’ dynamics. This expansion is steered and constrained by negative divergence of the dynamics, which ensures that the dimensionality of the solution space continues to reduce until an acceptable solution manifold is reached. Then the system contracts stably on this manifold towards its final solution trajectory. The unstable positive feedback and cross inhibition that underlie expansion and divergence are common motifs in molecular and neuronal networks. Therefore we propose that very simple organizational constraints that combine these motifs can lead to spontaneous computation and so to the spontaneous modification of entropy that is characteristic of living systems.

OPEN ACCESS

Citation: Rutishauser U, Slotine J-J, Douglas R (2015) Computation in Dynamically Bounded Asymmetric Systems. *PLoS Comput Biol* 11(1): e1004039. doi:10.1371/journal.pcbi.1004039

Editor: Olaf Sporns, Indiana University, UNITED STATES

Received: September 1, 2014

Accepted: November 12, 2014

Published: January 24, 2015

Copyright: © 2015 Rutishauser et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All data in this manuscript is simulated. The MATLAB source code of the underlying models is available on the webpage <http://www.ini.uzh.ch/~urut/DFAWTA/>

Funding: Funded by EU grants DAISY (FP6-2005-015803) and SECO (FP7-2009-216593) awarded to RJD. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Author Summary

Biological systems are obviously able to process abstract information on the states of neuronal and molecular networks. However, the concepts and principles of such biological computation are poorly understood by comparison with technological computing. A key concept in models of biological computation has been the attractor of dynamical systems, and much progress has been made in describing the conditions under which attractors exist, and their stability. Instead, we show here for a broad class of asymmetrically connected networks that it is the unstable dynamics of the system that drive its computation, and we develop new analytical tools to describe the kinds of unstable dynamics that support this computation in our model. In particular we explore the conditions under which

networks will exhibit unstable expansion of their dynamics, and how these can be steered and constrained so that the trajectory implements a specific computation. Importantly, the underlying computational elements of the network are not themselves stable. Instead, the overall boundedness of the system is provided by the asymmetrical coupling between excitatory and inhibitory elements commonly observed in neuronal and molecular networks. This inherent boundedness permits the network to operate with the unstably high gain necessary to continually switch its states as it searches for a solution. We propose that very simple organizational constraints can lead to spontaneous computation, and thereby to the spontaneous modification of entropy that is characteristic of living systems.

Introduction

The principles of biological computation are not well understood. Although the Turing Machine and related concepts [1–3] have provided powerful models for understanding and developing technological computing, they have provided less insight for biological computation because they generally assume that the machines themselves, as well as their initial program and data are granted as input. In contrast, the organization of states and transitions of the biological process arise out of phylogenetic and ontogenetic configuration processes and execute autonomously without the intervention of an intelligent external programmer and controller being necessary to supply already encoded organizationally relevant information. Our goal here is to make steps towards understanding biological computation [4–6], by considering the behavior of a simple non-linear dynamical system composed of asymmetrically interconnected linear-threshold neurons. We suppose that such computations entail a mapping from some input towards a limited (low entropy) region of phase space, which is the solution [7]. We do not suppose that the computational goal is known—only that computation must conform to this basic entropy reducing process. Here we describe the organizational constraints that make such spontaneous computation possible.

Previous authors have explained neural network computation in terms of the convergence of special dynamical systems, and emphasized the attractors to which they converge [8–13]. For example, Hopfield [9, 10] has shown how and why the dynamics of symmetrically connected neurons with saturating outputs converge to attractor states; and others have offered similar insights for symmetrically connected linear threshold neurons [14–16]. However, interactions between inhibitory and excitatory neurons are clearly asymmetric, making these studies ill suited to study biological computation. To the extent that asymmetrical networks have been considered at all, this has been through assumptions that reduce asymmetrical networks to approximate symmetry. By contrast, we consider here the dynamics of fully asymmetrical networks, and discover that asymmetry contributes strongly to computational behavior.

The scope of our work is restricted to recurrent neural networks with asymmetric coupling that express an important and ubiquitous behavior: soft winner-take-all (sWTA) dynamics. We present a formal account of the response of these networks to exogenous perturbations and use a form of non-linear stability analysis (contraction analysis [17]) to characterize the itinerant transients that ensue, and which have useful interpretations in terms of neuronal computation and information theory. Contraction Theory offers a more flexible framework than the conventional Lyapunov approach to non-linear stability (See [Methods](#) for details). This is particularly the case for non-autonomous systems such as our network, in which external inputs can vary with time.

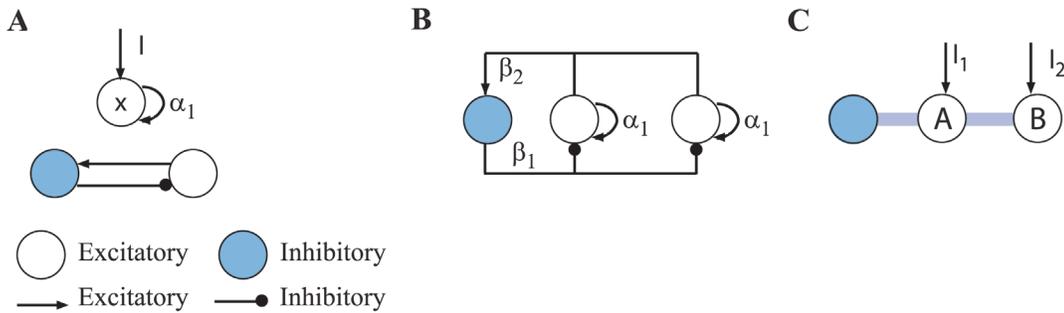


Figure 1. Circuit motifs and simple circuit composed of motifs. (A) Circuit motifs are excitatory self-recurrence (top) and shared inhibition (bottom). I_i denotes an external input. (B) Connectivity of a simple WTA circuit, consisting of two excitatory units that compete through shared inhibition. (C) More compact notation to denote the circuit shown in (B). Each excitatory element receives an external input.

doi:10.1371/journal.pcbi.1004039.g001

We explore particularly the behavior of network computation during the non-equilibrium phase, when the network is traversing its state-space seeking for a solution. We show that the ability of the network to explore potential solutions depends on highly unstable 'expansion' dynamics driven by recurrent excitation. This expansion is steered and constrained by negative divergence of the dynamics, which ensures that the dimensionality of the solution space continues to reduce until an acceptable solution manifold is reached. The system then 'contracts' stably on this manifold [17] towards its final solution trajectory, which is not necessarily converging to a fixed point. We argue that the simple principle of unstable expansion constrained by negative divergence provides the central organizing drive for more general autonomous biological systems from molecular networks, through neurons, to society.

Consider a simple network of non-linear neuron-like elements whose task it is to compute the solution to some problem. The states of the computation are encoded in the activations (firing rates) of the neurons, and the computational transitions between these states arise out of their synaptic interactions. The overall trajectory resulting from the successive transitions through its state space express its computation [18–20]. In current technological systems the hardware states of the system are encoded on binary nodes whose discrete states are imposed by signal restoring [21] circuitry [19]. This signal restoration is achieved by extremely high gain, so that a small input bias will drive the node into saturation at one of its two voltage limits. Biology rarely commands such sharply demarcated states and transitions. Instead, molecular and electrophysiological activation functions are often approximately sigmoidal (eg Hill functions, voltage dependent conductance, neuronal current-discharge curves, etc). However, neuronal systems do not typically run in saturation. The typical activation of a neuron is thresholded below, and above this threshold it makes use of only the lower part of its dynamic range. It very rarely enters saturation at the upper end of its activation range. Therefore, a suitable model for neuronal activation is a thresholded linear one. That is, their activity is bounded from below, but their positive activity is essentially unbounded (over any practical range of discharge). This is a very well studied model [14–16, 22].

As in our previous work, the neuronal network model is composed of thresholded linear neuron-like units coupled through positive (excitatory) and negative (inhibitory) connections (see Fig. 1a). The unbounded positive range of neuron activation implies that the global stability of networks of these neurons must arise out of their collective interactions rather than from saturation of their individual activation functions as assumed by for example [9, 10]. The key interaction here is the inhibitory feedback, which must at least ensure that not all neurons can simultaneously increase their activation [16]. Previous studies of such models have focused on the mathematically more tractable case in which the connections between neurons are

symmetrical, and have no transmission delays. Our networks, by contrast, need not be symmetrical and may have transmission delays. Indeed, the asymmetry of connections will be used to computational advantage, not offered by symmetrical networks.

Results

Network model

The network contains two fundamental circuit motifs (Fig. 1A): excitatory neurons that project both onto themselves and others, and inhibitory neurons which receive excitatory input from the same neurons that they inhibit. Several instances of these motifs together compose the general recurrent circuit that we investigate. In its simplest form, this recurrent network has the following form. There are N neurons, $N-1$ of which are excitatory and one (index N) is inhibitory (Fig. 1B,C). The excitatory neurons $x_i \neq N$ receive an optional external input I_i , and excitatory feedback α from themselves and nearby excitatory neurons (α_1 and α_2 , respectively). The single inhibitory neuron x_N sums the β_2 weighted input from all the excitatory neurons, and sends a common β_1 inhibitory signal to each of its excitatory neurons. Each neuron has a resistive constant leak term G_i .

The dynamics of this simple network are:

$$\tau \dot{x}_i + G_i x_i = f(I_i(t) + \alpha_1 x_i + \alpha_2 x_{i-1} + \alpha_2 x_{i+1} - \beta_1 x_N) \tag{1}$$

$$\tau \dot{x}_N + G_N x_N = f\left(\beta_2 \sum_{j=1}^{N-1} x_j\right) \tag{2}$$

where $f(x)$ is a non-saturating rectification non-linearity. Here, we take $f(x) = \max(x, 0)$, making our neurons linear threshold neurons (LTNs).

As long as the key parameters (α_i, β_i) satisfy rather broad constraints [23], this network behaves as a soft winner-take-all by allowing only a small number of active neurons to emerge, and that solution depends on the particular input pattern $I(t)$. After convergence to the solution, the active neurons (winners) will express an amplified version of the input $I(t)$, that is they remain sensitive to the input even if it changes.

After convergence to steady state in response to constant input $I_i > 0$, the activation of the winner i is:

$$x_i = \frac{I_i}{1 - \alpha_1 + \beta_1 \beta_2} = g I_i \tag{3}$$

where $g = \frac{1}{1 - \alpha_1 + \beta_1 \beta_2}$ is the gain of the network. Importantly, this gain can be $g \gg 1$ due to excitatory recurrence, which amplifies the signal in a manner controlled by the feedback loop. While above assumes $\alpha_2 = 0$, similar arguments hold without this assumption [23].

The dynamics of the non-linear system are $\tau \dot{\mathbf{x}} = \mathbf{f}(\mathbf{W}\mathbf{x} + \mathbf{I}(t)) - \mathbf{G}\mathbf{x}$ (where \mathbf{f} applies the scalar function $f(x)$ component-wise). For example, a minimal WTA with two excitatory and one inhibitory units, $\mathbf{x} = [x_1, x_2, x_3]$ (Fig. 1B,C), the weight matrix \mathbf{W} is

$$\mathbf{W} = \begin{bmatrix} \alpha_1 & 0 & -\beta_1 \\ 0 & \alpha_1 & -\beta_1 \\ \beta_2 & \beta_2 & 0 \end{bmatrix} \tag{4}$$

where $\mathbf{G} = \text{diag}(G_1, \dots, G_n)$ is a diagonal matrix containing the dissipative leak terms for each unit. The time constant of the system is $\frac{\tau}{G}$. Now we consider the conditions under which the non-linear system is guaranteed to be stable but at the same time powerful, i.e. permitting high gain.

Instability drives computation

Contraction Theory assesses the stability of non-linear systems $\dot{x} = f(x, t)$ using virtual displacements of its state at any point with respect to a chosen uniformly positive definite metric $M(x, t)$, obtained by a transformation $\Theta(x, t)$ (see [Methods](#)). The key Theorem 2 of [17] asserts that if the temporal evolution of *any* virtual displacement in this metric space tends to zero, then *all* other displacement trajectories in this space will also contract (shrink) to the same (common) trajectory.

For our present case, this theorem implies (see [Methods](#)) that the trajectories of a system of neurons with Jacobian \mathbf{J} are exponentially contracting if

$$\Theta \mathbf{J} \Theta^{-1} < \mathbf{0} \tag{5}$$

The Jacobian \mathbf{J} has dimension N and describes the connection matrix of the network and Θ is a transformation matrix that provides a suitable choice of coordinates.

For a WTA with weight matrix \mathbf{W} , the Jacobian is

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \Sigma \mathbf{W} - \mathbf{G} \tag{6}$$

where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ is a switching matrix, whose diagonal elements $\sigma_i \in \{0, 1\}$ indicate whether unit i is currently active or not.

Following [16] we will call the subset of N neurons that are currently above threshold the active set. Those that are inactive cannot contribute to the dynamics of the network. Thus, we may distinguish between the static anatomical connectivity W of the network, and the dynamic functional connectivity that involves only the subset of currently active neurons. This active set is described by the switch matrix Σ . The Jacobian expresses which units contribute to the dynamics at any point of time, and is thus a reflection of functional rather than anatomical connectivity. Each possible effective weight matrix has a corresponding effective Jacobian. Thus, $\Sigma W - \mathbf{G}$ is the effective (functional) Jacobian of the network, in which only active units have an influence on the dynamics.

The active sets may be either stable or unstable. In previous work on the hybrid analog and digital nature of processing by symmetric networks, Hahnloser et al. refer to such sets as active and forbidden, respectively [16]. They further show that the eigenvectors associated with positive eigenvalues of forbidden sets are mixed. That is, the eigenvector contains at least one component whose sign is opposite to the remainder of the components. This property ensures that a neuron will finally fall beneath threshold, and that the composition of the active set must change. We now generalize this concept to our asymmetrical networks, and refer to permitted and forbidden subspaces rather than sets, to emphasize the changes in the space in which the computational dynamics play out.

It is the instability (exponential divergence of neighboring trajectories) of the forbidden subspaces rather than stability that drives the computational process. This instability can be approached also through Theorem 2 of [17], which notes that if the minimal eigenvalue of the symmetric part of the Jacobian is strictly positive, then it follows that two neighboring trajectories will diverge exponentially. We will use 'expanding' to refer to this unstable, exponentially diverging behavior of a set of neurons to avoid confusion with Gaussian divergence, which we will need to invoke in a different context, below.

Thus, we will say that the dynamics of a set of active neurons is expanding if

$$\Theta \mathbf{V} \mathbf{J} \mathbf{V}^T \Theta^{-1} > \mathbf{0} \tag{7}$$

where \mathbf{V} is a projection matrix which describes subspaces that are unstable. For example, for a

circuit with two excitatory units that cannot both be simultaneously active, \mathbf{V} is

$$\mathbf{V} = \begin{bmatrix} \alpha & 0 & -\beta_1 \\ 0 & \alpha & -\beta_1 \end{bmatrix} \quad (8)$$

and Θ is a metric. The constraint (7) asserts that the system escapes the unstable subspaces where $\mathbf{V}\mathbf{x}$ is constant. This guarantees that For \mathbf{V} as defined above $\mathbf{V}\mathbf{x} = \begin{bmatrix} \alpha x_1 - \beta_1 x_3 \\ \alpha x_2 - \beta_1 x_3 \end{bmatrix}$.

Each row represents one excitatory unit. Guaranteeing that $\mathbf{V}\mathbf{x}$ cannot remain constant for a particular subset implements the requirement that for a subspace to be forbidden, it cannot be a steady state because if it were $\mathbf{V}\mathbf{x}$ would remain constant after convergence.

The parameter conditions under which Eqn 7 holds, are given in detail in [24]. Importantly, these conditions guarantee that when the dynamics of our asymmetric network occupies an unstable subspace, all eigenvectors are mixed (see Methods for proof). Consequently, as in the symmetric case of [16, 25], this unstable subspace will be left (it is forbidden), because one unit will fall beneath its threshold exponentially quickly and so become inactive.

Divergence quenches computational instability

The dynamics of our asymmetric networks can now be explained in terms of the contraction theory framework outlined above. Consider a simple network consisting of $N = 5$ neurons, one of which is inhibitory and enforces competition through shared inhibition (Fig. 2A). For suitable parameters that satisfy the contraction constraints (see [24]), this network will contract towards a steady state for any input. The steady state will be such that the network amplifies one of the inputs while suppressing all others (for example, Fig. 2B). During this process of output selection and amplification, the network passes through a sequence of transformations, at each of which a different subset of units becomes active whereas the remainder are driven beneath threshold and so are inactive. These transformations continue while the network is in its expansion phase and cease when the network contracts. The network is in the expansion phase while the effective Jacobian is positive definite, and contracts when the Jacobian becomes negative definite (Fig. 2C).

The computational process of selecting a solution, conditional on inputs and synaptic constraints, involves testing successive subspaces, expressed as changing patterns of neuronal activation (Fig. 2B). Subspaces do not offer a solution (are forbidden) if for that subspace $\mathbf{V}\mathbf{J}\mathbf{V}^T$ is positive definite. In this case its dynamics are expanding, and because all eigenvectors in this subspace are mixed, the subspace will finally switch. Subspaces that offer solutions (are permitted) will have an effective Jacobian that is negative definite in some metric (they are contracting), and so the system will not leave such a subspace provided that the input remains fixed. Note that there can be subspaces whose Jacobian is neither positive nor negative definite, which are then neither permitted or forbidden. However, by definition, the networks we describe assure that each possible subspace is either permitted or forbidden.

Consider the case in which the computational process begins in a forbidden subspace (Fig. 2F). The process then passes successively through several forbidden subspaces before reaching a permitted subspace. Two properties ensure this remarkable orderly progression towards a permitted subspace. Firstly, the process is driven by the instability that ensures that forbidden subspaces are left. However the trajectory is associated with a progressive reduction in the maximum positive eigenvalue of the active set (Fig. 2C).

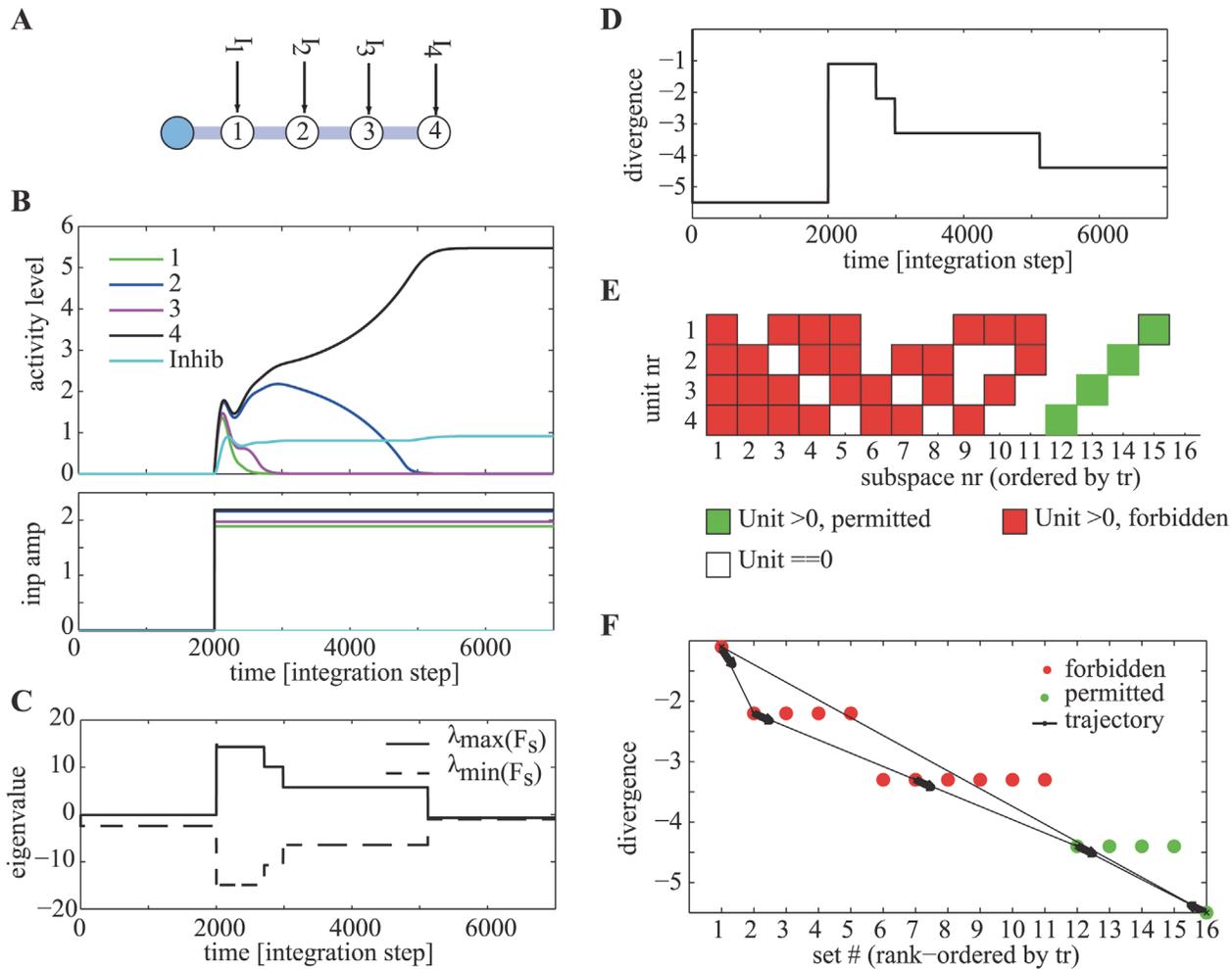


Figure 2. Illustration of key concepts to describe biological computation. (A) Connectivity of the circuit used in this figure. (B–F) Simulation results, with parameters $\alpha_1 = 1.2$, $\beta_1 = 3$, $\beta_2 = 0.25$, $G_5 = 1.5$, $G_{1..4} = 1.1$. (B) Activity levels (top) as a function of time for the units shown in (A) and external inputs provided (bottom). After stimulus onset at $t = 2000$, the network selectively amplifies the unit with the maximal input while suppressing all others. (C) Maximal and minimal eigenvalue of the functional connectivity F_S active at every point of time. At $t = 5000$ the system enters a permitted subspace, as indicated by the max eigenvalue becoming negative. (D) Divergence as a function of time. The divergence decreases with every subspace transition. (E) Illustration of all possible subspaces (sets) of the network. Subspaces are ordered by their divergence. For these parameters, only subspaces with at most one unit above threshold are permitted (green) whereas the others are forbidden (red). (F) Divergence as a function of subspace number (red and green circles) as well as trajectory through set space for the simulation illustrated in (B–D). Note how the divergence decreases with each transition, except when the input changes (set 16 is the off state).

doi:10.1371/journal.pcbi.1004039.g002

Secondly, an orderly progression through forbidden subspaces is ensured by systematic reduction of the state space through Gaussian divergence. Gauss’ theorem

$$\frac{d}{dt} \delta V = \text{div} \left(\frac{d}{dt} \delta z \right) \delta V \tag{9}$$

asserts that, in the absence of random fluctuations, any volume element δV shrinks exponentially to zero for uniformly negative definite $\text{div} \left(\frac{d}{dt} \delta z \right)$. This implies convergence to an $(n-1)$ -dimensional manifold rather than to a single trajectory. For our system, the Gaussian divergence is the trace of the Jacobian J or equivalently the sum of all of its eigenvalues. Note that this is a much weaker requirement than full contraction. In particular, we are concerned about the

trace of the effective Jacobian, which is the Jacobian of only the active elements, because the inactive elements (those below threshold) do not contribute to the dynamics.

The divergence quantifies the rate at which the volume shrinks (exponential) in its n dimensional manifold towards a $(n-1)$ dimensional manifold. The system will enter the $(n-1)$ dimensional and continue its evolution in this new active subset, and so on, until the reduced dimensional system finally enters a permitted subspace (which is contracting). Note that here we refer to the dimensionality of the system dynamics. This dimensionality is not necessarily equal to the number of active units (i.e. at steady state or when one of the units is constant).

Note that negative Gaussian divergence does not follow automatically from expansion. In fact most expanding sets will not have negative Gaussian divergence, and so will not be forbidden

For example consider the linear system $\dot{\mathbf{x}} = \mathbf{W}\mathbf{x}$ with $W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. This system is

expanding but it does not have negative divergence.

For orderly computation to proceed, such sets must not exist. We require that all forbidden subspaces are expanding as defined by contraction theory as well as have negative Gaussian divergence. Indeed, for our LTN networks all forbidden subspaces are guaranteed to satisfy both these properties.

Computation is steered by the rotational dynamics induced by asymmetric connections

Because their positive output is unbounded, linear thresholded neurons are essentially insensitive to the range of their positive inputs. Networks of these neurons amplify their inputs in a scale-free manner, according to the slope of their activation functions and the network gain induced by their connections. As explained above, this high gain drives the exploration and selection dynamics of the computational process. The network harnesses its high gain to steer the computation through its asymmetric connections. Indeed, the asymmetric nature of the connectivity in our network is central to its operation, and not a minor deviation from symmetry (approximate symmetry, [26]).

The interplay between high-gain and steering can be appreciated by considering the behavior of the system within one of the subspaces of its computation.

Consider a linear system of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$, which can be written as $\dot{\mathbf{x}} = \mathbf{M}\mathbf{x} + \mathbf{u}$ where \mathbf{M} is a matrix of connection weights and \mathbf{u} a vector of constant inputs. For the example of a WTA with 2 excitatory units and 1 inhibitory unit,

$$\mathbf{M} = \begin{bmatrix} l_1\alpha - G & 0 & -l_1\beta_1 \\ 0 & l_2\alpha - G & -l_2\beta_1 \\ l_3\beta_2 & l_3\beta_2 & -G \end{bmatrix} \tag{10}$$

Setting $\mathbf{M} = \mathbf{M}_1 + \mathbf{M}_2$ and defining

$$\mathbf{M}_1 = \begin{bmatrix} l_1\alpha - G & 0 & 0 \\ 0 & l_2\alpha - G & 0 \\ 0 & 0 & -G \end{bmatrix} \tag{11}$$

$$\mathbf{M}_2 = \begin{bmatrix} 0 & 0 & -l_1\beta_1 \\ 0 & 0 & -l_2\beta_1 \\ l_3\beta_2 & l_3\beta_2 & 0 \end{bmatrix} \quad (12)$$

provides a decomposition into a component with negative divergence and zero divergence (see [methods](#) for an example).

Any vector field $f(\mathbf{x})$ can be written as the sum of a gradient field $\nabla V(x)$ and a rotational vector field $\rho(x)$, i.e. a vector field whose divergence is zero. In analogy, $f(\mathbf{x}) = \dot{\mathbf{x}} = \mathbf{M}_1\mathbf{x} + \mathbf{u} + \mathbf{M}_2\mathbf{x}$ where $\nabla V(x) = \mathbf{M}_1\mathbf{x} + \mathbf{u}$ and $\rho(x) = \mathbf{M}_2\mathbf{x}$.

For our network \mathbf{M}_1 is the expansion/contraction component and \mathbf{M}_2 is the rotational component. This is an application of the Helmholtz decomposition theorem to our network [27, 28]. These two matrices relate to two functional components in the network architecture: The excitatory recurrence plus input, and the inhibitory recurrence. The first component provides the negative divergence that defines the gradient of computation, while the second steers its direction as follows.

Since \mathbf{M}_2 has a divergence of zero, this component is rotational. If, in addition, \mathbf{M}_2 is skew-symmetric so that $-\mathbf{M}_2 = \mathbf{M}_2^T$, the system is rotating, and the eigenvalues of \mathbf{M}_2 will be imaginary only. In general, $\beta_2 \neq -\beta_1$ and \mathbf{M}_2 is thus not skew-symmetric. However, note that a transform of the form $\Phi f(\mathbf{x}) \Phi^{-1}$ can be found that makes \mathbf{M}_2 skew-symmetric. Because such transform does not change the eigenvalues of \mathbf{M}_1 , it will not change its divergence either. For above example,

$$\Phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sqrt{\frac{\beta_1}{\beta_2}} \end{bmatrix} \quad (13)$$

which will result in a version of \mathbf{M}_2 that is skew-symmetric

$$\Phi \mathbf{M}_2 \Phi^{-1} = \begin{bmatrix} 0 & 0 & -\sqrt{\beta_1\beta_2} \\ 0 & 0 & -\sqrt{\beta_1\beta_2} \\ \sqrt{\beta_1\beta_2} & \sqrt{\beta_1\beta_2} & 0 \end{bmatrix} \quad (14)$$

The same transformation to a different metric has to be applied to \mathbf{M}_1 as well, but as both \mathbf{M}_1 and Φ are diagonal this will leave \mathbf{M}_1 unchanged, $\mathbf{M}_1 = \Phi \mathbf{M}_1 \Phi^{-1}$.

The orderly progression through the forbidden subspaces can be understood in this framework: The negative divergence provided by \mathbf{M}_1 enforces an exponential reduction of any volume of the state space while the rotational component \mathbf{M}_2 enforces exploration of the state space by directing (steering) the dynamics.

The stability of the permitted subspaces can also be understood in this framework. Permitted subspaces are contracting, despite the strong self-recurrence of the excitatory elements that results in positive on-diagonal elements. This high gain, which is necessary for computation, is kept under control by a fast enough rotational component \mathbf{M}_2 which ensures stability. This can be seen directly by considering one of the constraints imposed by contraction analysis on valid parameters affecting an individual neuron: keeping $\alpha < 2\sqrt{\beta_1\beta_2}$ guarantees that the system rotates sufficiently fast to remain stable.

Note that both M_1 and M_2 change dynamically as a function of the currently active subspace. Thus, the direction and strength of the divergence and rotation change continuously as a function of both the currently active set as well as the input.

Changes of entropy during computation

While the network is in the expansion phase, the volume of the state space in which the dynamics of the network evolves is shrinking. This process depends on initial conditions and external inputs. Consequently, the sequence by which dimensions are removed from state space is sensitive to initial conditions. To quantify the behavior of a network for arbitrary initial conditions it is useful to compute its information entropy $H(t)$ as a function of time (see [methods](#)). The smaller $H(t)$, the smaller the uncertainty about which subspace the network occupies at time t . Once $H(t)$ ceases to change, the network has converged. To reduce state uncertainty, and so to reduce entropy, is fundamentally what it means to compute [7].

The entropy remains 0 immediately after the application of external inputs to the network, because the same “all on” subspace is reached regardless of initial conditions. Thereafter the network begins to transition through the hierarchy of forbidden subspaces ([Fig. 2F](#)). This process initially increases the entropy as the network explores different subspaces. Eventually, after attaining peak entropy, the network reduces entropy as it converges towards one of its few permitted subspaces. [Fig. 3](#) illustrates this process for the network shown in [Fig. 2A](#).

Increasing the gain by increasing the value of the self-recurrence α increases the speed by which entropy is changed but not its asymptotic value. This means that all permitted subspaces are equally likely to be the solution but that solutions are found more quickly with higher gain. Adding additional constraints through excitatory connections makes some permitted subspaces more likely to be the solution, and so the asymptotic entropy is lower (see [Fig. 3B](#), where a connection $\alpha_2 = 0.2$ from unit 1 to 2 and $\alpha_3 = 0.2$ from unit 4 to 2 was added).

Note how the number of constraints and the gain of the network are systematically related to both the increasing and decreasing components of $H(t)$. For example, increasing the gain leads to a more rapid increase of entropy, reaching peak earlier and decaying faster towards the asymptotic value ([Fig. 3A](#)). Also, adding constraints results in smaller peak entropy, indicating that the additional constraints limited the overall complexity of the computation throughout ([Fig. 3B](#)).

The network reaches maximal entropy when there exists the largest number of forbidden subspaces having the same divergence. This occurs always for an intermediate value of divergence, because then there occurs the largest number of subspaces having an equal number of units active and inactive. This can be seen by considering $\arg \max_k \binom{N}{k} = \frac{N}{2}$ (if N is even), i. e. the network will have maximal entropy when the number of active units is 50%. Numerically, this can be seen by comparing the time-course of the maximal positive eigenvalue or the divergence with that of the time-course of the entropy ([Fig. 3C,D](#)).

Overall, $H(t)$ demonstrates the dynamics of the computational process, which begins in the same state (at its extreme, all units on), and then proceeds to explore forbidden subspaces in a systematic fashion by first expanding and then contracting towards a permitted subspace.

Structure and steering of computation

We will now use the concepts introduced in the preceding sections to explain how our circuits compute and how this understanding can be utilized to systematically alter the computation through external inputs and wiring changes in the network.

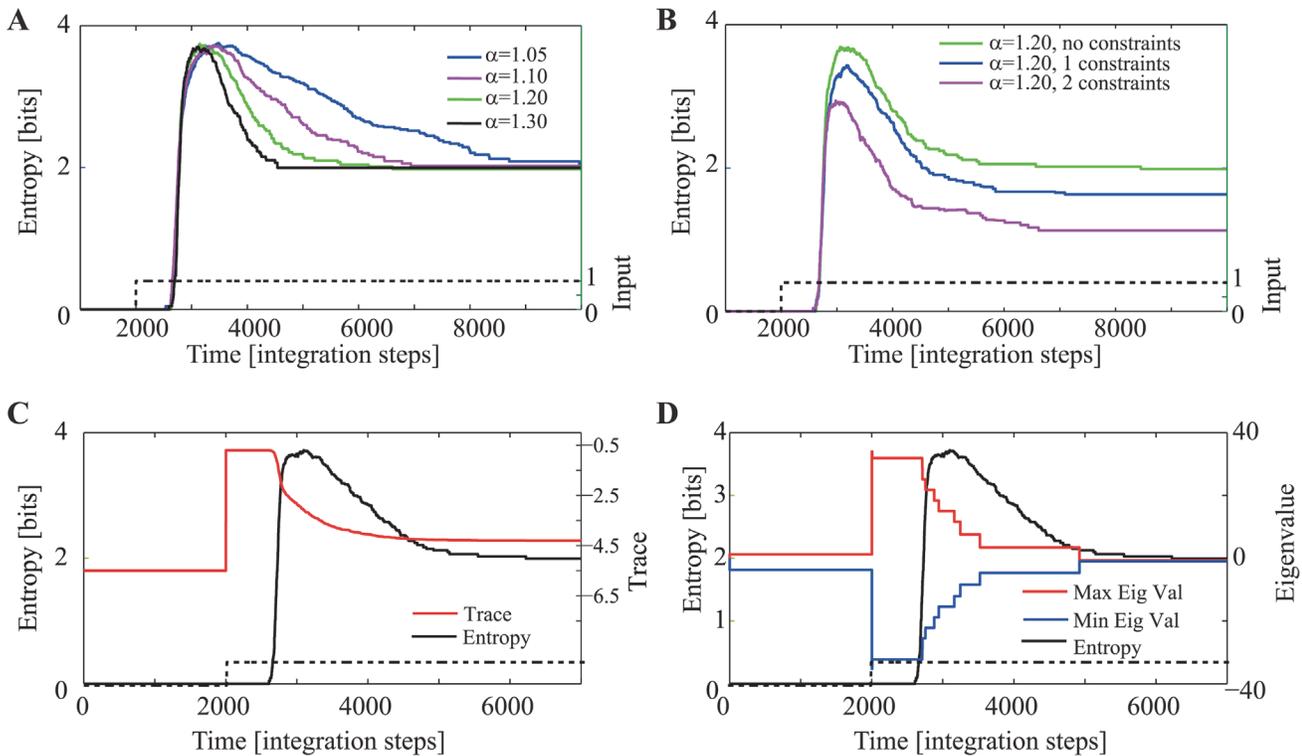


Figure 3. Spontaneous increase followed by reduction of state entropy during the expansion phase. Random inputs were provided to the 5-node network as shown in Fig. 2. Each input I_i was chosen i.i.d from a normal distribution with $\mu = 6$ and $\sigma = 0.25$. (A) Entropy as a function of time and gain of the network. Higher gains are associated with faster increases and reductions of entropy but converge to the same asymptotic entropy. This indicates that each permitted subspace is reached with equal probability. (B) Adding constraints reduces the peak and asymptotic entropy. The more constraints that are added, the larger is the reduction in entropy. (C) Comparison of time-course of average entropy and divergence (trace). (D) Comparison of time-course of average entropy and eigenvalues (min, max). Notice how both the divergence (C) and the eigenvalues (D) reach their maximal values well before the entropy reaches its maximum.

doi:10.1371/journal.pcbi.1004039.g003

Provided that the external input remains constant, the network proceeds in an orderly and directed fashion through a sequence of forbidden subspaces. This sequence of steps is guaranteed to not revisit subspaces already explored, because when in a forbidden subspace S_1 with divergence d_1 , the next subspace S_2 that the network enters must have more negative divergence $d_2 < d_1$. It thus follows that when the system has left a subspace with divergence d_1 that it can never return to any subspace with divergence $\geq d_1$. It also follows that the network can only ever enter one of the many subspaces S_i with equal divergence $d_i = X$ (Fig. 2F shows an example). Not all subspaces with lower d_i than the current subspace are reachable. This is because once a unit has become inactive by crossing its activation threshold, it will remain inactive. Together, this introduces a hierarchy of forbidden subspaces that the network traverses while in the exploration phase. Fig. 4A shows the hierarchy of the sets imposed by the network used in Fig. 2. This tree-like structure of subspaces constrains computation in such a way that at any point of time, only a limited number of choices can be made. As a consequence, once the network enters a certain forbidden subspace, a subset of other forbidden and permitted subspaces becomes unreachable (Fig. 4B). What those choices are depends on the input whereas the tree-like structure of the subspaces is given by the network connectivity.

Knowledge of how the network transitions through the hierarchy of forbidden subspaces can be used to systematically introduce biases into the computational process. Such additional steering of the computation can be achieved by adding connections in the network. Additional

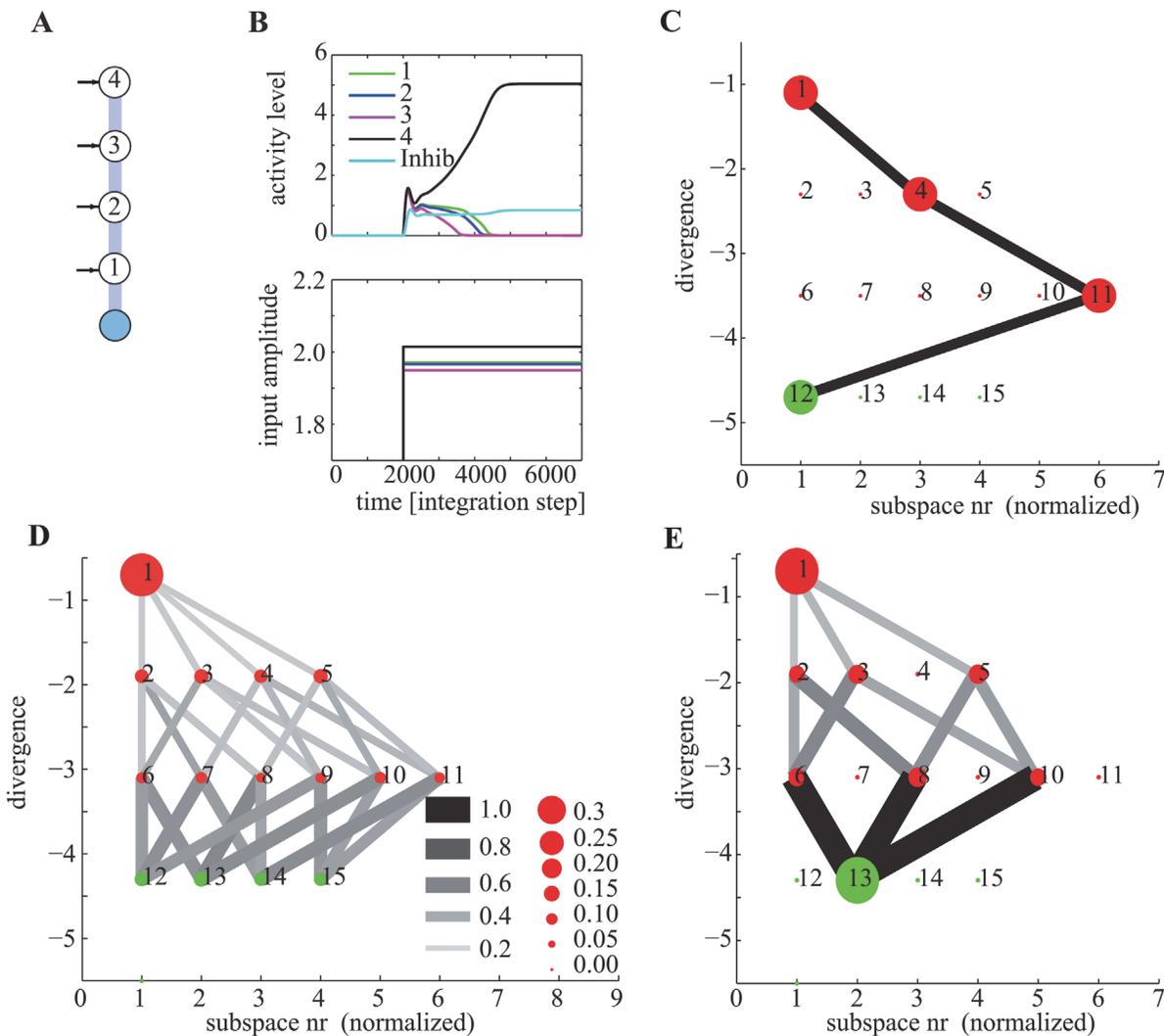


Figure 4. Hierarchy of subspaces and steering of computation. (A) 5-node network, using notation from Fig. 2. (B) Simulation of an individual run. Top shows the state and bottom the inputs to the network. The unit with the maximal input wins (unit 4). (C) Trajectory through state space for the simulation shown in (B). Each subspace is numbered and plotted as a function of its divergence. Red and green dots indicate forbidden and permitted subspaces respectively. Numbers inside the dots are the subspace (set) numbers (see Fig. 2E). (C–D) Transition probabilities for the same network, simulated with 1000 different random inputs. Connected subspaces are subsets between which the network can transition, in the direction that reduces divergence (gray lines). The size of dots and lines indicates the likelihood that a subset will be visited or a transition executed, respectively. The subset with most negative divergence is the zero set (all units off), which is not shown. (C) All transitions made by the network. Depending on the value of the input, the network will reach one of the permitted subspaces. Notice the strict hierarchy: all transitions were towards subspaces with lower divergence and only a subset of all possible transitions are possible. (D) All transitions the network made, conditional on that subspace 13 was the solution. Note that this subspace cannot be reached from some subspaces (such as nr 7).

doi:10.1371/journal.pcbi.1004039.g004

off-diagonal excitatory connections, for example, will make it more likely that a certain configuration is the eventual winner. An example of this effect is shown in Fig. 5, where adding two additional excitatory connections results in the network being more likely to arrive in a given permitted subspace than others. For identical inputs (compare Figs. 5B and 4B) the resulting permitted subspace can be different through such steering.

The network remains continuously sensitive to changes in the external input. This is important and can be used to steer the computation without changing the structure of the network. In the absence of changes in the external input, the network is unable to make transitions other

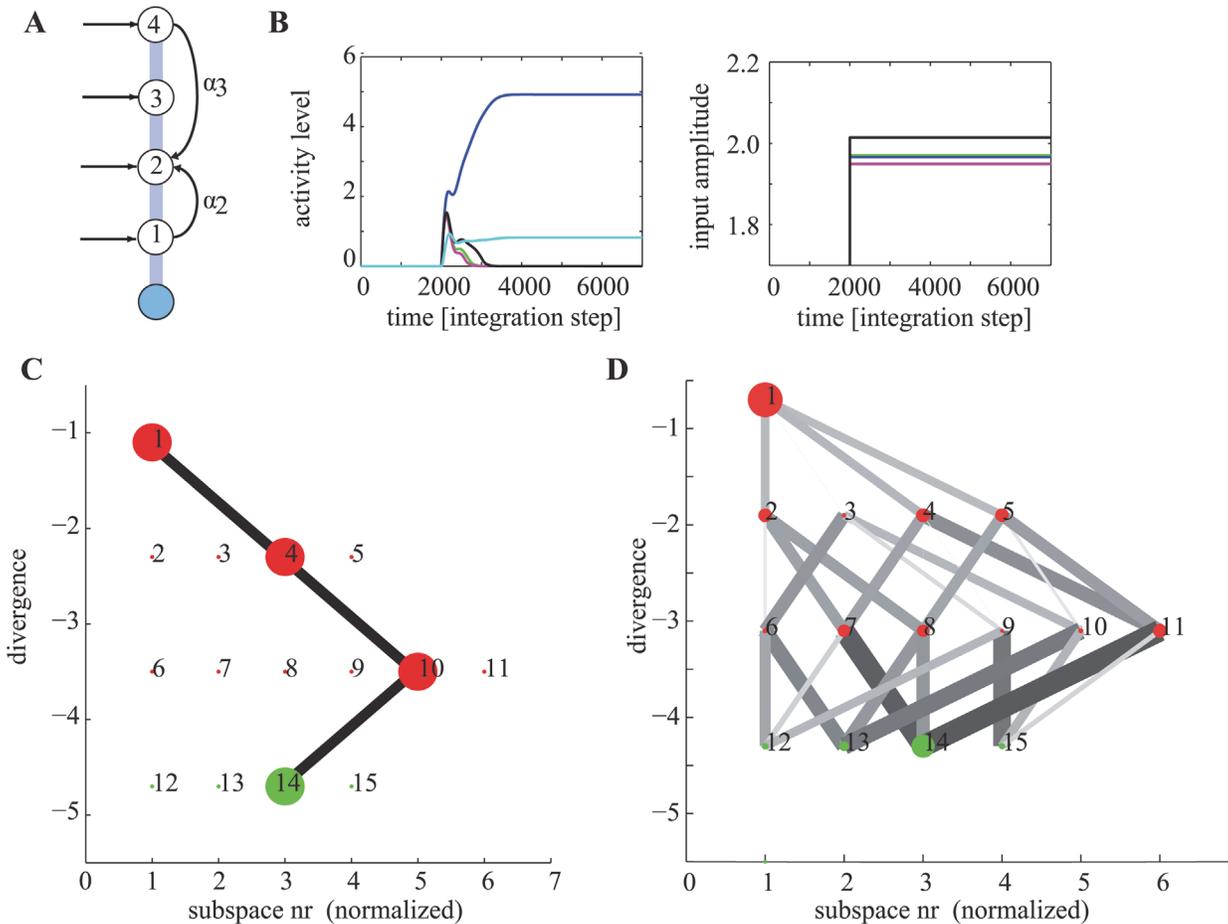


Figure 5. Steering of computation. (A) The same network as shown in Fig. 4A, with two additional excitatory connections $\alpha_2 = 0.2$ and $\alpha_3 = 0.2$. (B) Simulation of an individual run. Top shows the state and bottom the inputs to the network. Although the input is identical to Fig. 4B, the computation proceeds differently due to presence of the additional connections. (C) Trajectory through state space for the simulation shown in (C). The state space trajectory is initially identical to the simulation shown in (Fig. 4C), but then diverges. (D) All transitions made by the network. The transition probabilities are now biased and non-equal. Consequently, some permitted subspaces are reached more frequently than others.

doi:10.1371/journal.pcbi.1004039.g005

than those which lead to subspaces with lower divergence. When the input changes, on the other hand, the network can make such changes. For example, if the inputs change as a consequence of the network entering a certain forbidden subspace, the network can selectively avoid making certain transitions (Fig. 6A). This will steer the computation such that some permitted subspaces are reached with higher likelihood. Noisy inputs similarly can lead to transitions which make divergence less negative. Nevertheless, the large majority of transitions remains negative as long as noise levels are not too large. For example, repeating the same simulation but adding normally distributed noise with $\sigma = 1$ and $\mu = 0$ resulted in 26% of transitions being against the gradient (see Fig. 6B for an illustration).

So far, we have illustrated how negative divergence and expansion jointly drive the computational process in a simple circuit of multiple excitatory neurons that have common inhibition. While this circuit alone is already capable of performing sophisticated computation, many computations require that several circuits interact with one another [23, 29]. The concepts developed in this paper can also be applied to such compound circuits, because the circuit motifs and parameter bounds we describe guarantee that collections of these circuits will also possess forbidden and permitted subspaces and are thus also computing. The compound network is

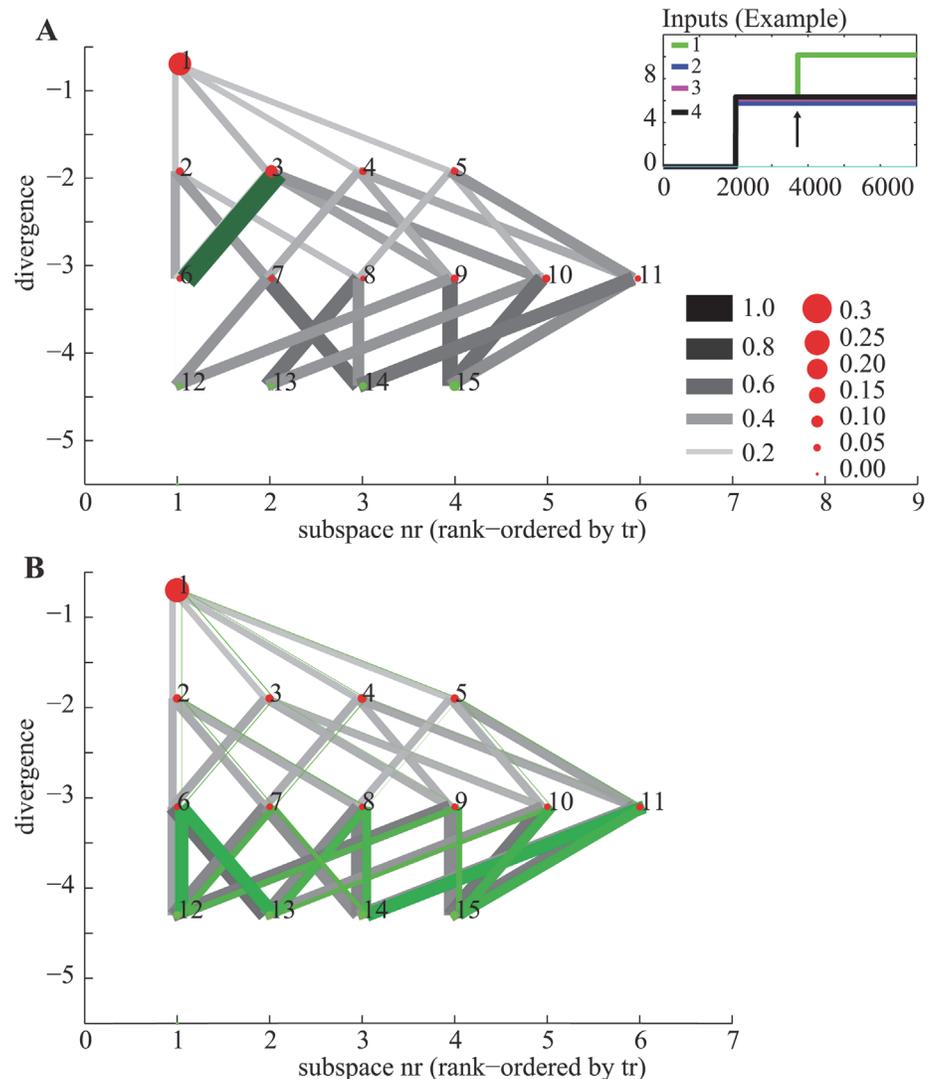


Figure 6. Steering of computation by external inputs. The network remains sensitive to changes in its input throughout the computation. Transitions are bi-directional: towards more negative and positive divergence are gray and green, respectively. (A) Example of a selective change in the input. Here, unit 4 receives additional external input only if the network enters forbidden subspace 6 (see inset). This change forces the network to make a transition to a subspace with less negative divergence (green). As a result, the eventual solution is more likely to be subspace 15. (B) Example of continuously changing inputs (input noise). Notice how noise introduces additional transitions towards less negative divergence at all stages. However, the general gradient towards more negative divergence persists: overall, 26% of transitions made the divergence more positive.

doi:10.1371/journal.pcbi.1004039.g006

guaranteed to be dynamically bounded, which means that no neuron’s activity can escape towards infinity. This property of the collective system relies on two key aspects: i) collections of individual circuits with negative divergence also have negative divergence, and ii) collective stability [24, 30]. Together, these properties guarantee that collections of the motifs will compute automatically.

Consider a network assembled by randomly placing instances of the two circuit motifs on a 2D plane and connecting them to each other probabilistically (Fig. 7A,B and methods). This random configuration results in some excitatory elements sharing inhibition via only one

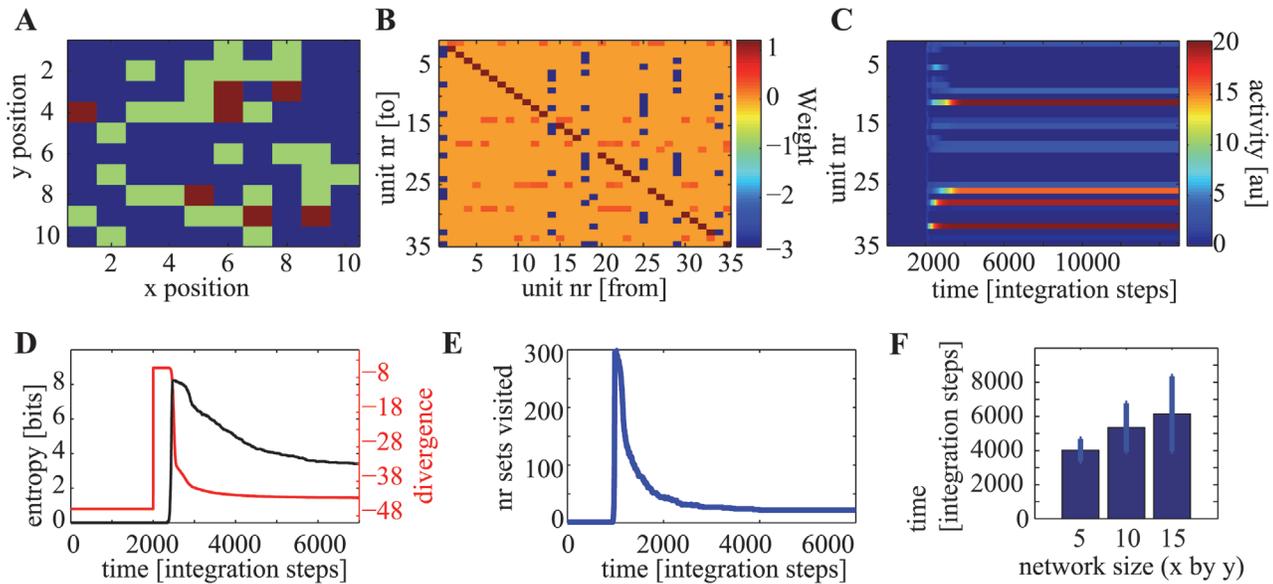


Figure 7. Large random networks spontaneously compute. (A) Each site in a 10×10 grid is either empty or occupied by an excitatory (green) or inhibitory (red) neurons (probabilistically). (B) Connectivity matrix. (C) Example run. (D) Entropy and divergence as a function of time for random initial conditions. (E) The number of different subspaces visited reduces as a function of time similarly to the entropy. (F) Time \pm s.d. till the network first enters a permitted subspace as a function of network size, specified as grid with. 100 runs of randomly assembled networks were simulated for each size. The larger the network, the longer the network spent going through the forbidden subspaces. Panels A–E show an example of grid size 10. See [Methods](#) for simulation details.

doi:10.1371/journal.pcbi.1004039.g007

inhibitory motif, whereas others take part in many inhibitory feedback loops (Fig. 7B). This random circuit will compute spontaneously (Fig. 7C,D). It is not known a priori how many forbidden and permitted subspaces the network has, nor how many possible solutions it can reach. Nevertheless, it is guaranteed that the network will reduce entropy and eventually reach a permitted subspace (Fig. 7E). The more connections (constraints) that are added to the network the smaller the number of permitted subspaces, and generally the harder the computation will become. How long the computation will take to reach a permitted subspace depends on both the network size, and the number of connections (constraints). Generally, the smaller the number of permitted subspaces the harder the computation will be. The important point is that random instances of such circuits will always compute, which means they will always reach a permitted subspace (Fig. 7F).

Discussion

The contribution of this paper has been to explore the fundamental role of instability in driving computation in networks of linear threshold units. Previous studies of computation in neural networks have focused on networks of sigmoidal units with symmetrical connectivity. Our networks of asymmetrically connected LTNs draw attention to important features of computation that were not apparent in these previous models. The conditional selective behavior crucial for computation depends on the threshold nonlinearity of the LTN. However, in order to make use of these non-linearities the network must express substantial gain. Because the activation of LTNs is unbounded for positive inputs, the network can in principle produce very high activations through unstably high gain. In these networks, computation is expressed as passage through a sequence of unstable states. It is this dynamical trajectory by which the network computes [1, 2, 31]. Despite this essential instability, the system does not escape, but remains

bounded in its behavior. In this paper we have analyzed why this is so. We find that the instabilities are self limiting, and that the overall process of computation is systematically quenched by Gaussian divergence. Contraction analysis provides explicit tools to quantify both instantaneous rates of exponential convergence to limiting states or trajectories, and divergence rates from specific subspaces. Here, we use these tools to analyze the unstable phase of the dynamics. This phase is crucial, because computation is inseparable from instability. Here we have made steps towards characterizing and explaining these phenomena.

The type of dynamical system we consider can implement soft-WTA type behavior, amongst others. This makes our framework applicable to the extensive body of literature on this type of network [32–38]. While simple, the soft-WTA is a powerful computational primitive that offers the same computational power than a multi-layer perceptron [32]. Key aspects of what makes WTA-networks powerful are high network gain, which allows computations that require sparsification, and also provides stability. While the computational power of WTAs has long been recognized and exploited to implement simple cognitive behaviors [23, 29, 39], it has remained unclear what it means to compute in such networks. Here, we provide such understanding in terms of a dynamical system. This system is physically realizable by realistic neurons and their connections. Other work in this direction has focused on abstract mathematical models [36, 40–43], and less on physically realizable dynamical computation. More recently, others [44–46] have offered useful models for understanding the principles whereby the brain may attain cognition, but these approaches do not offer methods for implementing such algorithms as physical computation in neuronal circuits.

The advances of this paper can be seen in contrast with classical assumptions concerning the form of activation functions, continuous sensitivity to input, and symmetry of connections. For example, the behavior of our LTN networks can be contrasted with networks of the kind originally proposed by Hopfield [20] that allow no self-connections ($w_{ii} = 0, \forall i$), have symmetric connectivity ($w_{ij} = w_{ji}$), and their activation function is bounded on both sides. This guarantees bounded dynamics by construction, allowing such networks to express high gain by virtue of a steep activation function rather than through connections of the network. However, a consequence of this is that when it operates with high gain the network operates in saturation and thus becomes insensitive to input apart from initial conditions. Such networks have neither negative divergence nor rotational dynamics, which together with insensitivity to external input severely restricts their computational abilities as well as systematic design.

Importantly, our networks are continuously sensitive to their inputs. These external inputs are a combination of signal and noise and can transfer the network from one subspace to another at any point of time and this transfer can be against the gradient imposed by negative divergence. Non-autonomous systems continuously interact with their environment, for which continuous sensitivity to input is crucial. Systems of asymmetrically interacting linear threshold units are well suited for this situation. This is because their non-saturating units make the system adaptive to the input amplitudes and sensitivity to inputs is conditional on the current state, i.e. only the inputs contributing to the dynamics of the currently active state influence the dynamics.

Although there has been a considerable amount of work on symmetric networks, biological neuronal networks are always asymmetric because of inhibitory neurons. Also, the inhibitory inputs to an excitatory neuron can be substantially stronger than the excitatory inputs. This results in particularly strong asymmetry, a property with many implications for computation in such networks [47]. The theoretical study of networks with defined cell types (excitatory or inhibitory) thus requires asymmetric connectivity. Previous studies have used infinitely fast all-to-all inhibition to circumvent this problem, which results in symmetric connectivity but lacks defined cell types. Such networks allow dynamically bounded activity for linear threshold units

[16, 48]. Apart from being biologically unrealistic, such networks can only express limited gain and are thus computationally very limited [47]. By contrast, our networks express high gain and dynamic instabilities during the exploration phase. Their asymmetric connections provide the rotational dynamics that keep their activity bounded despite this high gain. It is worth noting that many powerful algorithms, such as e.g. the Kalman filter [49, 50] also rely on negative feedback and strongly asymmetric connectivity.

The dynamics of the exploration phase are highly structured because the different forbidden subspaces are systematically related to one another. Indeed, the subspaces are ordered in a hierarchy through which the dynamics proceed. At any point in this hierarchy only a limited and known set of subspaces can be entered next (unless the external input changes). The systematic understanding of the unstable dynamics driving exploration can be used to steer and modify the computational trajectory while it is in process, rather than only when a solution has been found. The network can influence its environment continuously as a function of the forbidden subspaces it traverses, for example by executing a specific action whenever a particular subspace is entered. This feature can be used to make the computations of several networks dependent on each other. For example, to enforce dependencies between several ongoing computations such as, “all solutions must be different”.

The connections of the network are the constraints imposed on the computation. The more connections per neuron, the fewer possible solutions exist and the harder (slower) the computation is. From this point of view, the networks we describe perform constraint satisfaction, which is a hard computational problem and which has been proposed as an abstract model of computation [51, 52]. Connections can be inserted systematically to forward program specific algorithms and behaviors [23, 24, 29], randomly or a combination thereof. Either way, the system will compute [53], but in the former case will execute specific algorithms while in the later the algorithm is unknown.

The constraints active at any point of time depend on the state of the network as expressed by the effective connectivity of the network expressed by the switching matrix. Every time the network changes state, the switching matrix changes. Dynamically, the same concept can be applied: the effective Jacobian jointly expresses all the currently activity constraints for a given state. Only if the possible state(s) of a network are known is it possible to determine the effective Jacobian. An important implication is that to understand the underlying algorithm that drives the computation performed by a group of neurons knowledge of the structural connectivity is not sufficient [54–56]. This is because connectivity alone does not determine the possible states of the network.

The circuit motifs and parameter bounds we describe guarantee that collections of these circuits will also possess forbidden and permitted subspaces and thus are computing. By collections we mean multiple copies of the same motifs that are in addition connected to each other, as for example in the random network discussed in the results. This is important because collections of these motifs will compute automatically, a property we refer to as collective computation. This makes it practical to design large-scale computing systems without having to perform global analysis to guarantee both the type of instability required for computation as well as stability of the solutions.

It is important to note that one need not commit to a particular circuit motif beyond guaranteeing that both forbidden and permitted subspaces exist in the way we define them. While a network composed of such motifs but otherwise connected randomly will always compute, the individual states do not have meaning nor is the algorithm that the network computes known. However, the states that the network proceeds through while it computes are systematically related to each other. Consequently, assigning a meaningful interpretation to a few key states will make all states meaningful. A similar approach is used in reservoir computing, where states are first created and only later assigned with meaning by learning mechanisms [57]. A key next

step will be to discover how linking specific forbidden subspaces with motor actions that in turn change the input to the system allow a computation to remain continuously sensitive to the environment while it proceeds. An other next step is to discover how several interacting systems can bias each others computations systematically to reach a solution that is agreeable to all while satisfying the local constraints of each computation.

The unstable positive feedback and cross inhibition that underly expansion and divergence are common motifs found in many molecular, cellular and neuronal networks [58]. Therefore all such systems follow the very simple organizational constraints that combine these motifs. This will lead such circuits to compute spontaneously and thereby to reduce their state entropy as is characteristic of living systems.

Methods

Numerical methods

All simulations were performed with Euler integration with $\delta = 0.01$, thus τ is equivalent to 100 integration steps. All times in the figures and text refer to numerical integration steps. Unless noted otherwise, the external inputs $I_i(t)$ were set to 0 at $t = 0$ and then to a constant non-zero value at $t = 2000$. The constant value I_i was drawn randomly and i.i.d. from $\mathcal{N}(6, 1\sigma)$. All simulations were implemented in MATLAB.

In cases where noise was added, the noise was supplied as an additional external input term $\mathcal{N}(0, \sigma)$ with $\sigma = 1$.

$$\tau \dot{x}_i + G_i x_i = f(I_i + \mathcal{N}(0, \sigma) + \alpha_1 x_i + \alpha_2 x_{i-1} + \alpha_2 x_{i+1} - \beta_1 x_N) \quad (15)$$

A new noise sample is drawn from the random variable every τ to avoid numerical integration problems.

Large network simulation

A 2D square of dimension 10 by 10 spots is created. Each spot is occupied by a neuron with $P = 0.4$. If a spot is occupied, it is excitatory with $P = 0.8$ and inhibitory otherwise. Connectivity between units is generated with the following constraints: All excitatory units receive excitation from themselves with α_1 . Further, each excitatory unit receives inhibition from and excites up to 8 randomly chosen inhibitory neurons with probability $P = 0.4$. It is very unlikely that an excitatory unit is not connected to any inhibitory neuron at all ($P = (0.6)^8$).

Entropy

The information entropy $H(t)$ is

$$H(t) = - \sum_i p_i(t) \log_2(p_i(t)) \quad (16)$$

The sum is over all subspaces i and $p_i(t)$ is the probability of the network being in subspace i at time t , over random initial conditions (we used 1000 runs). By definition, we take $0 \log_2(0) = 0$.

Analytical methods

The principal analytical tool used is contraction analysis [17, 59–61]. In this section, we briefly summarize the application of contraction analysis to analyzing asymmetric dynamically bounded networks [24]. Essentially, a nonlinear time-varying dynamic system will be called *contracting* if arbitrary initial conditions or temporary disturbances are forgotten exponentially

fast, i.e., if trajectories of the perturbed system return to their unperturbed behavior with an exponential convergence rate. Relatively simple algebraic conditions can be given for this stability-like property to be verified, and this property is preserved through basic system combinations and aggregations.

A nonlinear contracting system has the following properties [17, 59–61]

- global exponential convergence and stability are guaranteed
- convergence rates can be explicitly computed as eigenvalues of well-defined Hermitian matrices
- combinations and aggregations of contracting systems are also contracting
- robustness to variations in dynamics can be easily quantified

Consider now a general dynamical system in \mathbb{R}^n ,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \tag{17}$$

with \mathbf{f} a smooth non-linear function. The central result of Contraction Analysis, derived in [17] in both real and complex forms, can be stated as:

Theorem Denote by $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ the Jacobian matrix of \mathbf{f} with respect to \mathbf{x} . Assume that there exists a complex square matrix $\Theta(\mathbf{x}, t)$ such that the Hermitian matrix $\Theta(\mathbf{x}, t)^{*T} \Theta(\mathbf{x}, t)$ is uniformly positive definite, and the Hermitian part \mathbf{F}_H of the matrix

$$\mathbf{F} = \left(\dot{\Theta} + \Theta \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right) \Theta^{-1}$$

is uniformly negative definite. Then, all system trajectories converge exponentially to a single trajectory, with convergence rate $|\sup_{\mathbf{x}, t} \lambda_{\max}(\mathbf{F}_H)| > 0$. The system is said to be *contracting*, \mathbf{F} is called its *generalized Jacobian*, and $\Theta(\mathbf{x}, t)^{*T} \Theta(\mathbf{x}, t)$ its *contraction metric*. The contraction rate is the absolute value of the largest eigenvalue (closest to zero, although still negative) $\lambda = |\lambda_{\max}(\mathbf{F}_H)|$.

In the linear time-invariant case, a system is globally contracting if and only if it is strictly stable, and \mathbf{F} can be chosen as a normal Jordan form of the system, with Θ a real matrix defining the coordinate transformation to that form [17]. Alternatively, if the system is diagonalizable, \mathbf{F} can be chosen as the diagonal form of the system, with Θ a complex matrix diagonalizing the system. In that case, \mathbf{F}_H is a diagonal matrix composed of the real parts of the eigenvalues of the original system matrix. Here, we choose $\Theta = \mathbf{Q}^{-1}$ where \mathbf{Q} is defined based on the eigendecomposition $\mathbf{J} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$.

The methods of Contraction Analysis were crucial for our study for the following reasons: i) Contraction and divergence rates are exponential guarantees rather than asymptotic (note that the more familiar Lyapunov exponents can be viewed as the average over infinite time of the instantaneous contraction rates in an identity metric). ii) No energy function is required. Instead, the analysis depends on a metric Θ that can be identified for a large class of networks using the approach outlined. iii) The analysis is applicable to non-autonomous systems with constantly changing inputs.

The boundary conditions for a WTA-type network to be contracting as well as to move exponentially away from non-permitted configurations were derived in detail in [24]. They are:

$$\begin{aligned} 1 < \alpha < 2\sqrt{\beta_1\beta_2} \\ \frac{1}{4} < \beta_1\beta_2 < 1 \end{aligned} \tag{18}$$

Forbidden subspaces have mixed eigenvalues

When the system resides in a forbidden subspace, the maximal eigenvalue of the effective Jacobian is positive and its divergence is negative. A forbidden subspace thus has a mixed eigenvector \mathbf{v} associated with the maximal eigenvalue λ_{max} . A mixed eigenvector is one where at least one entry is strictly negative and at least one strictly positive.

Proof by contradiction: Assume all components $v_i \geq 0$. Consider a constant external input where the system starts along one of the eigenvectors \mathbf{v} . In this case, the state of the network is determined entirely by this eigenvector and would grow towards infinity without ever changing dimensionality. However, the conditions for a forbidden subspace do not permit this because $\mathbf{V}\mathbf{x} = 0$ ensures that the system escapes this subspace exponentially fast (see Eqs 7–8). Thus in a forbidden subspace, by definition, the state cannot grow towards infinity and the eigenvector \mathbf{v} must be mixed.

Rotation versus expansion—Example

This example illustrates the role of rotation.

$$\begin{bmatrix} 1 & a \\ -a & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} + a \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (19)$$

In the above, divergence is -1 . Whether the system is linearly stable depends on the value of a . If the determinant $-2 + a^2 > 0$, this system is stable. Thus the system is stable if $a^2 > 2$, i.e. if the rotation is “fast enough”.

Acknowledgments

We thank the participants of the Capo Caccia Cognitive Neuromorphic Engineering workshops of 2009–2014 for discussion.

Author Contributions

Conceived and designed the experiments: UR JJS RJD. Performed the experiments: UR. Analyzed the data: UR JJS RJD. Contributed reagents/materials/analysis tools: UR JJS RJD. Wrote the paper: UR JJS RJD.

References

1. Turing AM (1936) On computable numbers, with an application to the entscheidungsproblem. *J of Math* 58: 345–363.
2. Minsky L (1967) *Computation: finite and infinite machines*. Englewood Cliffs: Prentice Hall.
3. Gödel K, Kleene SC, Rosser JB (1934) On undecidable propositions of formal mathematical systems. Institute for Advanced Study.
4. Mitchell M (2011) Ubiquity symposium: Biological computation. Ubiquity 2011.
5. Rabinovich MI, Friston KJ, Varona P (2012) *Principles of brain dynamics: global state interactions*. MIT Press.
6. Simon HA (1991) *The architecture of complexity*. Springer.
7. Feynman RP, Hey J, Allen RW (1998) *Feynman lectures on computation*. Addison-Wesley Longman Publishing Co., Inc.
8. Cohen MA, Grossberg S (1983) Absolute stability of global pattern-formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems Man and Cybernetics* 13: 815–826. doi: [10.1109/TSMC.1983.6313075](https://doi.org/10.1109/TSMC.1983.6313075)

9. Hopfield JJ (1984) Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences* 81: 3088–3092. doi: [10.1073/pnas.81.10.3088](https://doi.org/10.1073/pnas.81.10.3088) PMID: [6587342](https://pubmed.ncbi.nlm.nih.gov/6587342/)
10. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* 79: 2554–2558. doi: [10.1073/pnas.79.8.2554](https://doi.org/10.1073/pnas.79.8.2554) PMID: [6953413](https://pubmed.ncbi.nlm.nih.gov/6953413/)
11. Hertz J, Krogh A, Palmer R (1991) *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley.
12. Seung HS, Richardson TJ, Lagarias JC, Hopfield JJ (1998) Minimax and hamiltonian dynamics of excitatory-inhibitory networks. In: *Advances in Neural Information Processing Systems* 10. MIT Press, pp. 329–335.
13. Wang X (2008) Attractor network models. In: Squire L, editor, *Encyclopedia of Neuroscience*. Oxford: Academic Press, volume 1, pp. 667–679.
14. Ben-Yishai R, Bar-Or RL, Sompolinsky H (1995) Theory of orientation tuning in visual cortex. *Proceedings of the National Academy of Sciences* 92: 3844–3848. doi: [10.1073/pnas.92.9.3844](https://doi.org/10.1073/pnas.92.9.3844) PMID: [7731993](https://pubmed.ncbi.nlm.nih.gov/7731993/)
15. Dayan P, Abbott L (2001) *Theoretical Neuroscience*. Cambridge MA: MIT Press.
16. Hahnloser R, Sarpeshkar R, Mahowald M, Douglas R, Seung HS (2000) Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* 405: 947–51. doi: [10.1038/35016072](https://doi.org/10.1038/35016072) PMID: [10879535](https://pubmed.ncbi.nlm.nih.gov/10879535/)
17. Lohmiller W, Slotine J (1998) On contraction analysis for nonlinear systems. *Automatica* 34. doi: [10.1016/S0005-1098\(98\)00019-3](https://doi.org/10.1016/S0005-1098(98)00019-3)
18. McCulloch W, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology* 5: 115–133.
19. Mead C, Conway L (1980) *Introduction to VLSI systems*, volume 1080. Addison-Wesley Reading, MA.
20. Hopfield J (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America* 79: 2554–2558. doi: [10.1073/pnas.79.8.2554](https://doi.org/10.1073/pnas.79.8.2554) PMID: [6953413](https://pubmed.ncbi.nlm.nih.gov/6953413/)
21. von Neumann J (1958) *The Computer and the Brain*. New Haven: Yale University Press.
22. Amari SI (1972) Characteristics of random nets of analog neuron-like elements. *IEEE Transactions on Systems, Man and Cybernetics SMC-2*: 643–657.
23. Rutishauser U, Douglas R (2009) State-dependent computation using coupled recurrent networks. *Neural Computation* 21. doi: [10.1162/neco.2008.03-08-734](https://doi.org/10.1162/neco.2008.03-08-734) PMID: [19431267](https://pubmed.ncbi.nlm.nih.gov/19431267/)
24. Rutishauser U, Douglas R, Slotine J (2011) Collective stability of networks of winner-take-all circuits. *Neural computation* 23. doi: [10.1162/NECO_a_00091](https://doi.org/10.1162/NECO_a_00091) PMID: [21162667](https://pubmed.ncbi.nlm.nih.gov/21162667/)
25. Hahnloser R, Seung H, Slotine J (2003) Permitted and forbidden sets in symmetric threshold-linear networks. *Neural Comput* 15: 621–638. doi: [10.1162/089976603321192103](https://doi.org/10.1162/089976603321192103) PMID: [12620160](https://pubmed.ncbi.nlm.nih.gov/12620160/)
26. Hopfield JJ (1984) Neurons with graded response have collective computational properties like those of 2-state neurons. *Proceedings of the National Academy of Sciences of the United States of America-Biological Sciences* 81: 3088–3092. doi: [10.1073/pnas.81.10.3088](https://doi.org/10.1073/pnas.81.10.3088)
27. Ao P (2004) Potential in stochastic differential equations: novel construction. *Journal of physics A: mathematical and general* 37: L25. doi: [10.1088/0305-4470/37/3/L01](https://doi.org/10.1088/0305-4470/37/3/L01)
28. Joseph DD (2006) Helmholtz decomposition coupling rotational to irrotational flow of a viscous fluid. *Proceedings of the National Academy of Sciences* 103: 14272–14277. doi: [10.1073/pnas.0605792103](https://doi.org/10.1073/pnas.0605792103)
29. Neftci E, Binas J, Rutishauser U, Chicca E, Indiveri G, et al. (2013) Synthesizing cognition in neuromorphic electronic systems. *Proceedings of the National Academy of Sciences* 110: E3468–E3476. doi: [10.1073/pnas.1212083110](https://doi.org/10.1073/pnas.1212083110)
30. Slotine J, Lohmiller W (2001) Modularity, evolution, and the binding problem: a view from stability theory. *Neural Networks* 14: 137–145. doi: [10.1016/S0893-6080\(00\)00089-7](https://doi.org/10.1016/S0893-6080(00)00089-7) PMID: [11316230](https://pubmed.ncbi.nlm.nih.gov/11316230/)
31. Dijkstra EW, Dijkstra EW, Dijkstra EW, Dijkstra EW (1976) *A discipline of programming*, volume 4. prentice-hall Englewood Cliffs.
32. Maass W (2000) On the computational power of winner-take-all. *Neural Computation* 12: 2519–2536. doi: [10.1162/089976600300014827](https://doi.org/10.1162/089976600300014827) PMID: [11110125](https://pubmed.ncbi.nlm.nih.gov/11110125/)
33. Hahnloser R, Douglas R, Mahowald M, Hepp K (1999) Feedback interactions between neuronal pointers and maps for attentional processing. *Nat Neurosci* 2: 746–52. doi: [10.1038/11219](https://doi.org/10.1038/11219) PMID: [10412065](https://pubmed.ncbi.nlm.nih.gov/10412065/)

34. Douglas R, Martin K (2007) Recurrent neuronal circuits in the neocortex. *Curr Biol* 17: R496–R500. doi: [10.1016/j.cub.2007.04.024](https://doi.org/10.1016/j.cub.2007.04.024) PMID: [17610826](https://pubmed.ncbi.nlm.nih.gov/17610826/)
35. Yuille A, Geiger D (2003) Winner-take-all networks. In: Arbib M, editor, *The Handbook of Brain Theory and Neural Networks*, MIT Press. pp. 1228–1231.
36. Ermentrout B (1992) Complex dynamics in winner-take-all neural nets with slow inhibition. *Neural Networks* 5: 415–431. doi: [10.1016/0893-6080\(92\)90004-3](https://doi.org/10.1016/0893-6080(92)90004-3)
37. Wersing H, Steil JJ, Ritter H (2001) A competitive-layer model for feature binding and sensory segmentation. *Neural Computation* 13: 357–387. doi: [10.1162/089976601300014574](https://doi.org/10.1162/089976601300014574) PMID: [11177439](https://pubmed.ncbi.nlm.nih.gov/11177439/)
38. Abbott L (1994) Decoding neuronal firing and modelling neural networks. *Q Rev Biophys* 27: 291–331. doi: [10.1017/S0033583500003024](https://doi.org/10.1017/S0033583500003024) PMID: [7899551](https://pubmed.ncbi.nlm.nih.gov/7899551/)
39. Rutishauser U, Slotine JJ, Douglas RJ (2012) Competition through selective inhibitory synchrony. *Neural computation* 24: 2033–2052. doi: [10.1162/NECO_a_00304](https://doi.org/10.1162/NECO_a_00304) PMID: [22509969](https://pubmed.ncbi.nlm.nih.gov/22509969/)
40. Blum L (1998) *Complexity and real computation*. Springer.
41. Blum L, Shub M, Smale S (1989) On a theory of computation and complexity over the real numbers: ηp -completeness, recursive functions and universal machines. *Bulletin (New Series) of the American Mathematical Society* 21: 1–46. doi: [10.1090/S0273-0979-1989-15750-9](https://doi.org/10.1090/S0273-0979-1989-15750-9)
42. Siegelmann HT, Fishman S (1998) Analog computation with dynamical systems. *Physica D: Nonlinear Phenomena* 120: 214–235. doi: [10.1016/S0167-2789\(98\)00057-8](https://doi.org/10.1016/S0167-2789(98)00057-8)
43. Siegelmann HT, Sontag ED (1995) On the computational power of neural nets. *Journal of computer and system sciences* 50: 132–150. doi: [10.1006/jcss.1995.1013](https://doi.org/10.1006/jcss.1995.1013)
44. Deco G, Jirsa VK, Robinson PA, Breakspear M, Friston K (2008) The dynamic brain: from spiking neurons to neural masses and cortical fields. *PLoS computational biology* 4: e1000092. doi: [10.1371/journal.pcbi.1000092](https://doi.org/10.1371/journal.pcbi.1000092) PMID: [18769680](https://pubmed.ncbi.nlm.nih.gov/18769680/)
45. Dayan P (2008) Simple substrates for complex cognition. *Frontiers in neuroscience* 2: 255. doi: [10.3389/neuro.01.031.2008](https://doi.org/10.3389/neuro.01.031.2008) PMID: [19225599](https://pubmed.ncbi.nlm.nih.gov/19225599/)
46. Ma WJ, Beck JM, Latham PE, Pouget A (2006) Bayesian inference with probabilistic population codes. *Nature neuroscience* 9: 1432–1438. doi: [10.1038/nn1790](https://doi.org/10.1038/nn1790) PMID: [17057707](https://pubmed.ncbi.nlm.nih.gov/17057707/)
47. Li Z, Dayan P (1999) Computational differences between asymmetrical and symmetrical networks. In: *Advances in Neural Information Processing Systems*. pp. 274–280.
48. Hahnloser RH, Seung HS, Slotine JJ (2003) Permitted and forbidden sets in symmetric thresholdlinear networks. *Neural Computation* 15: 621–638. doi: [10.1162/089976603321192103](https://doi.org/10.1162/089976603321192103) PMID: [12620160](https://pubmed.ncbi.nlm.nih.gov/12620160/)
49. Haykin SS, Haykin SS, Haykin SS (2001) *Kalman filtering and neural networks*. Wiley Online Library.
50. Kalman RE, Bucy RS (1961) New results in linear filtering and prediction theory. *Journal of Fluids Engineering* 83: 95–108.
51. Sussman GJ, Steele Jr GL (1980) Constraints a language for expressing almost-hierarchical descriptions. *Artificial intelligence* 14: 1–39. doi: [10.1016/0004-3702\(80\)90032-6](https://doi.org/10.1016/0004-3702(80)90032-6)
52. Radul A, Sussman GJ (2009) The art of the propagator. In: *Proceedings of the 2009 international lisp conference*. pp. 1–10.
53. Stepney S (2012) Nonclassical computation: a dynamical systems perspective. In: *Handbook of Natural Computing*. Springer, pp. 1979–2025.
54. Marder E (1998) From biophysics to models of network function. *Annual Review of Neuroscience* 21: 25–45. doi: [10.1146/annurev.neuro.21.1.25](https://doi.org/10.1146/annurev.neuro.21.1.25) PMID: [9530490](https://pubmed.ncbi.nlm.nih.gov/9530490/)
55. Getting PA (1989) Emerging principles governing the operation of neural networks. *Annual Review of Neuroscience* 12: 185–204. doi: [10.1146/annurev.ne.12.030189.001153](https://doi.org/10.1146/annurev.ne.12.030189.001153) PMID: [2648949](https://pubmed.ncbi.nlm.nih.gov/2648949/)
56. Koch C, Laurent G (1999) Complexity and the Nervous System. *Science* 284: 96–98. doi: [10.1126/science.284.5411.96](https://doi.org/10.1126/science.284.5411.96) PMID: [10102826](https://pubmed.ncbi.nlm.nih.gov/10102826/)
57. Jaeger H, Maass W, Principe J (2007) Special issue on echo state networks and liquid state machines. *Neural Networks* 20: 287–289. doi: [10.1016/j.neunet.2007.04.001](https://doi.org/10.1016/j.neunet.2007.04.001)
58. Genot AJ, Fujii T, Rondelez Y (2012) Computing with competition in biochemical networks. *Physical review letters* 109: 208102. doi: [10.1103/PhysRevLett.109.208102](https://doi.org/10.1103/PhysRevLett.109.208102) PMID: [23215526](https://pubmed.ncbi.nlm.nih.gov/23215526/)
59. Lohmiller W, Slotine J (2000) Nonlinear process control using contraction theory. *AIChE Journal* 46: 588–596. doi: [10.1002/aic.690460317](https://doi.org/10.1002/aic.690460317)
60. Slotine J (2003) Modular stability tools for distributed computation and control. *International Journal of Adaptive Control and Signal Processing* 17: 397–416. doi: [10.1002/acs.754](https://doi.org/10.1002/acs.754)
61. Wang W, Slotine J (2005) On partial contraction analysis for coupled nonlinear oscillators. *Biological Cybernetics* 91.