Thermodynamic Neural Network

T.L. Hylton[1]

Abstract

This work describes a thermodynamically motivated neural network model that self-organizes to transport charge associated with internal and external potentials while in contact with a thermal bath. Isolated networks show multiscale dynamics and evidence of phase transitions, and externally driven networks evolve to efficiently connect external positive and negative potentials. The model implements techniques for rapid, global, reversible, conservative equilibration of node states followed by slow, local, irreversible, dissipative adaptation of the edge states as a means to create multiscale order. All interactions in the network are local and the network structures can be generic and recurrent. The model integrates concepts of fluctuation, dissipation, adaptation and equilibration and offers an illustration of the thermodynamic evolution of organization in open systems. The key conclusion of the work is that the dissipation of conserved physical quantities can drive the self-organization of open thermodynamic systems.

Introduction

Applying concepts from thermodynamics and statistical physics to neural network models has a relatively long history, much of it centered on models of interacting spins on a lattice – Ising models. The dynamics of these systems near critical points (Glauber, 1963) (Suzuki & Kubo, 1968) have been studied to derive, for example, magnetic response functions using mean field approximations. The statistical properties of randomly disordered Ising models - spin glasses – are also well studied and understood in limiting cases including infinite range interactions (Kirkpatrick & Sherrington, 1978). These ideas were extended to networks capable of storing memories as attracting states recalled using only a portion of the initial memory – Hopfield Networks (Hopfield, 1982). The statistical mechanics of these networks have been developed in detail, particularly regarding their capacity to store and recall memories (Amit, Gutfreund, & Sompolinsky, 1985) (Bruce, 1987) (Sompolinsky, 1988) (Gutfreund, 1990). Further development of these ideas to include “hidden” spin states not directly determined by a set of training vectors and thermodynamically inspired techniques for training these networks were captured in a class of models called Boltzmann Machines (Ackley, Hinton, & Sejnowski, 1985) (Hinton & Sejnowski, 1986). Statistical physics also found application in non-Ising neural network models including layered networks (Levin, Tishby, & Solla, 1990), complex network model spaces (Reka & Barabasi, 2002) (Watkin, Rao, & Biehl, 1993), and directed, Markovian neuronal networks (Clark, 1988). The motivations and techniques employed in this work borrow heavily from this history and the work on Ising models in particular.

[1] University of California, San Diego, Department of Electrical and Computer Engineering. The author can be contacted at thylton@ucsd.edu or toddhylton1@gmail.com

As compared to these earlier works, the Thermodynamic Neural Network (TNN) model presented here is distinguished by its electric circuit inspiration and the incorporation of physical concepts such as charge conservation, potential diffusion, reaction kinetics and dissipation-driven adaptation. In particular, the transport of a conserved quantity does not appear as a primary concern in these earlier works. Although learning is the primary objective of the model, the motivation is not the generalization of a training set, the replication of a function, or the storage of a memory; rather, learning is viewed as adaptation to improve equilibration with external potentials and a thermal reservoir.

The larger inspiration for this work is the long standing hypothesis that evolution of the natural world, including life, is driven by thermodynamics and constrained by the laws of physics (Schrodinger, 1943) (Schneider & Kay, 1994). In particular the model addresses the case of external boundary potentials that vary slowly as compared to the characteristic equilibration time of the network and the resulting evolution of the network to minimize internal entropy production (Glansdorff & Prigogine, 1964) while creating entropy at the boundaries via the transport of conserved quantities (Schlogl, 1967). These are the essential physics that drive the self-organization of the network. A particular contribution of this work is the recognition that selective internal dissipation of conserved quantities as the means by which the system "learns" without "forgetting" earlier configurations that were effective under other boundary potentials. In particular, this work was inspired by experiments on collections of metallic balls in oil that self-assemble to create electrical connections when subject to external potentials (Jun & Hubler, 2005). Although the primary context for the work is physics and thermodynamics, we were also inspired by ideas from complex systems, chemistry, neurobiology, computation and cognitive science.

Approach

Model Concepts

The model comprises a collection of internal network nodes connected by symmetrically weighted edges and driven by a collection of external biasing nodes through which external potentials may be applied and charge may be injected. Each network node optimizes the transport of charge among its inputs while separately conserving positive and negative charges. While sharing some similarity with the Ising model, this conservation requirement creates very different formulations of the node energy and the resulting network model (Fig.1).

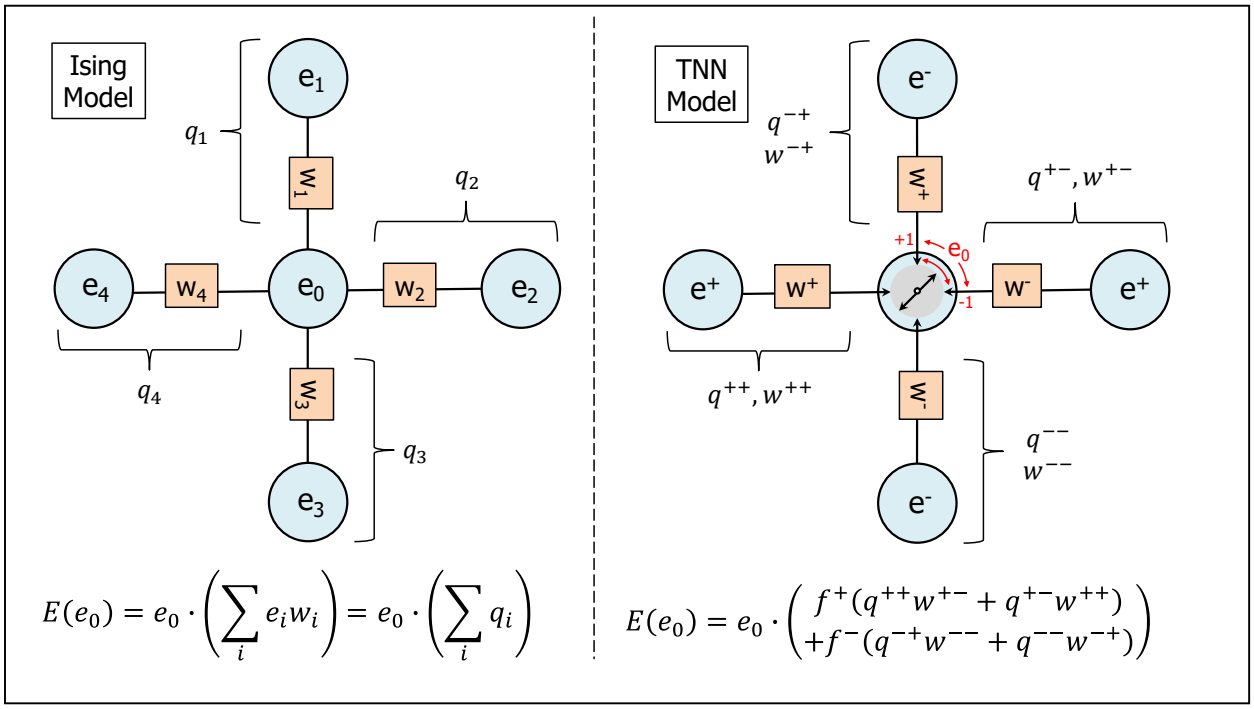


Figure 1: The Ising model (left) is a network of symmetrically weighted interacting spins on a lattice. Node energy $E(e_0)$ is a product of the node state e_0 and the weighted sum over the states of its connected nodes. Low energy states are those that align the node state with the "net" interaction of the connected nodes. The TNN model (right) is also a network of symmetrically weighted interactions on a lattice, but nodes interact via exchange of charge and low energy node states are those that effectively transport charge among its inputs. Charge conservation in the model prevents the cancelling of charges of opposite sign and requires an accounting of charge inputs by the signs of their input states and weights (see text). Node state selection is a competition to connect two different groups of input charge. Kinetic factors f are included in the node energy to focus weight updates on the subset of weights that dominate the selection of the decision state (see text).

The following list of criteria are the assumptions on which the TNN model is based.

1) Network nodes may assume states e representing a "potential", which are communicated to connected nodes through shared edges. For example, binary nodes might assume a state from {-1, 1}. In general, the model supports any number of values of the node state on the interval [-1, 1].

2) Each edge is characterized by a "weight" w describing a capacity to transport "charge" between the nodes it connects. Edge weights are symmetric – they receive charge from and provide charge to both of the nodes that they connect. In general, weights are continuous, real numbers but they may also be bounded on an interval.

3) Node potential e applied to an edge of weight w generates an edge charge $q = e \cdot w$ that is the input to its connected node along the edge.

4) Network nodes optimize the transfer of charge from input to output through selection of state - lower energy states are those that better match the charge collected on a node's inputs with the charge delivered on its outputs. Node state selection is determined by relaxation to a thermal bath using Boltzmann statistics. In general, charge transfer is imperfect and residual charge remains on the node after state selection.

5) External nodes may impose potentials and inject charge into the network and, thereby, create a "problem" that the network must "solve". In particular, because the injection of charge into the network raises the node energies, effective relaxation requires that the network self-organize to connect the external sources and sinks of charge or, equivalently, external positive and negative potentials. In the examples described below these external nodes are periodically varying potentials connected to a subset of the network nodes through large, fixed weights.

6) A critical characteristic of the model is that the positive and negative charges *never sum to cancel* each other[2], but instead communicate paths along which charge should flow. Correspondingly, external sources of positive charge are also sinks of negative charge and vice-versa. By these means, externally applied potentials are able to diffuse through the network and connect to complementary potentials[3]. Network node energies are low when positive charge flow counters the direction of negative charge flow. Correspondingly, low network energy favors simultaneous positive and negative charge flow in opposite directions through the edges[4], and the network evolves to favor asymmetric ("antiferromagnetic") order among the nodes. That is, if two nodes are connected by an edge, then configurations in which the nodes have opposite potentials will be favored. Hence, if the nodes are arranged on a regular grid and connected locally, then domains of anti-ferromagnetic order typically emerge. This requirement generates a complexity of charge accounting in the node energy formulation and weight updating that is not present in perceptron-based neural network models nor in Ising models.

7) Residual charge remaining on the node after state selection may be either accumulated (and, thereby, influence future state selection) or dissipated as updates to edge weights. A single state decision will in general result in both dissipation and accumulation of some of the charge. Qualitatively, if the selection of node state results in effective transfer of complementary positive and negative charge among a subset of edges then that subset of

[2] In physical systems (e.g. a semiconductor), positive and negative charges generally also do not annihilate each other, even though their net charge may cancel.

[3] In physical systems like metals and semiconductors, external potentials induce shift in the fermi level such that the net ionic-electronic charge is slightly positive or slightly negative depending on the sign of the potential.

[4] The "current" from the positive and negative charge flows is the same because they flow in opposite directions.

edges should be adapted via dissipation. If for another subset of edges, the node state selection fails to transfer complementary charge, then the input charges associated with those edges should be accumulated. To make this allocation precise, a "kinetic" factor is introduced into competing terms in the node energy. This kinetic factor is critical to the nodes ability to update its weights to improve charge transport associated with one set of set of "selected" inputs while preserving weight values associated with the "unselected" inputs. In effect, the kinetic factors are the means of addressing the "forgetting problem" in the network by having it "attend" selectively to a subset of weights as it organizes itself.

8) When residual charge is allocated to an edge weight for dissipation, that weight is updated by relaxation to a thermal bath according to Boltzmann statistics using the same node energy equations that determined the node states. In the method described below, residual charge for dissipation is distributed equally among a node's edge weights, which is the least biased choice given the information available to the node.

9) A range of network topologies is possible including multi-dimensional grid networks with near-neighbor connectivity, probabilistically connected networks for nodes on grids with a metric (or "distance") that determines the probability of connection and random networks. In general, there is no imposition of hierarchy or "layers" upon the network as is common in neural network models, but these kinds of networks can also be supported.

10) Because connected nodes are driven to orient asymmetrically most network configurations are inherently "frustrated" in that the nodes cannot find a way to satisfy asymmetric orientation with all their connected nodes. For a special class of networks that are partitioned into two groups (bipartite networks) in which nodes of the first partition can connect only to nodes in the second partition, this frustration can be avoided. Nearest neighbor grid networks are inherently bipartite and are attractive to study because they are easy to visualize.

11) The node and edge states in the network are highly interdependent. A key challenge in their simulation is effectively finding low energy configurations of the entire network. In the methodology described here, the network node states are first updated to sample a network-scale low energy representation via an iterative Gibbs sampling technique followed by residual charge dissipation / edge weight updating. This technique repeatedly connects and refines large spatial scale / short temporal scale (i.e. the collection of network node states) and small spatial scale / long temporal scale (i.e. edge weights) features of the network as it evolves.

Network Model

Term	Range	Description
$e_{ij} = e_i$	[-1,1]	State / potential of node *i* (input to node *j*)
$w_{ij} = w_{ji}$	$\mathbb{R}$	Symmetric edge weight connecting nodes *i* and *j*
q_{ij}	$\mathbb{R}$	Edge charge delivered from node *i* to node *j*
H_j	$\mathbb{R}^+$	Energy of node *j*

The energy of network is the sum of the node energies

$$H_N(\boldsymbol{e},\boldsymbol{w},\boldsymbol{q}) = \sum_j H_j\big(e_j, w_j, q_j\big)$$

where $\boldsymbol{e}$ refers to the set of nodes potentials $\{e_j\}$, $\boldsymbol{w}$ refers to the set of edge weights $\{w_{ij}\}$, $\boldsymbol{q}$ refers to a corresponding set of edge charges $\{q_{ij}\}$, w_j and q_j refer to the subset of edge weights $\{w_{ij}\}$ and edge charges $\{q_{ij}\}$ connected to node j. The network is assumed to be in contact with a thermal reservoir of inverse temperature β, and the probability of a network state is assumed to be described according to Boltzmann statistics.

$$P_N(\boldsymbol{e},\boldsymbol{w},\boldsymbol{q}) = \frac{exp\big(-\beta H_N(\boldsymbol{e},\boldsymbol{w},\boldsymbol{q})\big)}{\sum_{\boldsymbol{e},\boldsymbol{w},\boldsymbol{q}} exp\big(-\beta H_N(\boldsymbol{e},\boldsymbol{w},\boldsymbol{q})\big)} \tag{1}$$

Node Model

Charge conservation (assumption 6 above) requires the segregation of edge charges according to the polarities of both the voltages and weights that generated them. We define these polarity-segregated[5] input edge charges as

$$q_{ij}^{\pm\pm} = \int\limits_{\substack{e_{ij}\gtrless 0 \\ w_{ij}\gtrless 0}} e_{ij} w_{ij} dt \;\cong e_{ij}^{\pm} \cdot w_{ij}^{\pm\pm} \tag{2}$$

In general, input edge charge may accumulate over multiple state decision times, as represented by the integration over time. The approximate equality in Eqn.2 assumes that the edge weights are changing slowly in time as compared to the node voltages – as such we define accumulated edge voltages and segregate them as

$$e_{ij}^{+} = \int e_{ij} dt \;\text{ for } e_{ij} > 0\,, \quad e_{ij}^{+} \in [0,1]$$

$$e_{ij}^{-} = \int e_{ij} dt \;\text{ for } e_{ij} < 0, \quad e_{ij}^{-} \in [0,-1]$$

The integration over time is limited with a maximum accumulation of $e_{ij}^{\pm} = \pm 1$ on any edge reflecting the limited domain of the node potentials. Because charge can be accumulated over multiple time steps, a single edge may accumulate separate values for e_{ij}^{+} and e_{ij}^{-} that exist simultaneously and thereby store edge charges of both polarities simultaneously. The edge weights $w_{ij}^{\pm\pm}$ are segregated in the same way as the edge charges $q_{ij}^{\pm\pm}$.

[5] In what follows, the convention in the use of superscripts is that the first superscript refers to the polarity of the input voltage and the second refers to the polarity of the edge weight. The sign of input charges $q_j^{\pm\pm}$ is the product of the superscripts. The sign of $w_j^{\pm\pm}$ is the sign of the second superscript.

At the node level, we further aggregate input charge, potential and weights by summing over the node's edges

$$q_j^{\pm\pm} = \sum_i q_{ij}^{\pm\pm}$$

$$\epsilon_j^{\pm} = \sum_i e_{ij}^{\pm}$$

$$w_j^{\pm\pm} = \sum_i w_{ij}^{\pm\pm}$$

We define output edge charge as the product of the node state and edge weight,

$$p_{ij} = e_j w_{ij}$$

and aggregated output charges as

$$p_j^{\pm\pm} = e_j \cdot w_j^{\pm\pm}$$

Note that these aggregated output charges are *associated with aggregated input charges*[6] through the superscript weight designations $w_j^{\pm\pm}$ as in Eqn.2. These aggregations and segregations of charge, weight and voltage are needed to address the model concepts of the prior section, as will become clear with the form of the node energy below.

The choice of the node energy equation is driven by four considerations

1) minimizing residual charge on the node
2) maximizing charge transport through the node
3) avoidance of attractors to the node states $e_j = 0$
4) kinetic factors to sharpen state decisions and direct residual charge dissipation and accumulation processes

After much experimentation[7], the following expression for the node energy was selected

$$\begin{aligned} H_j(e_j) = \left(f_j^{+}\right)^2 \left[\left(q_j^{++} - p_j^{+-}\right)^2 - \left(q_j^{++} + p_j^{+-}\right)^2 + \left(q_j^{+-} - p_j^{++}\right)^2 - \left(q_j^{+-} + p_j^{++}\right)^2\right] \\ + \left(f_j^{-}\right)^2 \left[\left(q_j^{--} - p_j^{-+}\right)^2 - \left(q_j^{--} + p_j^{-+}\right)^2 + \left(q_j^{-+} - p_j^{--}\right)^2 - \left(q_j^{-+} + p_j^{--}\right)^2\right] \end{aligned} \quad (3)$$

The first, third, fifth and seventh terms are associated with minimizing residual charge on the node. The second, fourth, sixth and eighth terms are associated with maximizing charge transport. The avoidance of attractors at $e_j = 0$ is addressed by the pairs of positive and

[6] Unlike input charges $q_j^{\pm\pm}$, the sign of the output charges $p_j^{\pm\pm}$ is not the product of the superscripts.

[7] We note that alternative formulations are possible and effective if they also support the four considerations.

negative terms that cancel terms in $\left(q_j^{\pm\pm}\right)^2$ and $\left(p_j^{\pm\pm}\right)^2$. $f_j^{\pm} = f_j^{\pm}(e_j)$ are the kinetic factors that we describe below. Expanding and collecting terms yields

$$\frac{H_j(e_j)}{4} = -\left(f_j^{+}\right)^2 \cdot \left(q_j^{++}p_j^{+-} + q_j^{+-}p_j^{++}\right) - \left(f_j^{-}\right)^2 \cdot \left(q_j^{--}p_j^{-+} + q_j^{-+}p_j^{--}\right) \tag{4}$$

Ignoring the kinetic factors, the terms in Eqn.4 are suggestive complementary pairs of "forces" $q_j^{\pm\pm}$ and "fluxes" $p_j^{\pm\mp}$ that are familiar from thermodynamics. We rewrite Eqn.4 as

$$\frac{H_j(e_j)}{4} = -\left(f_j^{+}\right)^2 \cdot \left(q_j^{++}w_j^{+-} + q_j^{+-}w_j^{++}\right) \cdot e_j - \left(f_j^{-}\right)^2 \cdot \left(q_j^{--}w_j^{-+} + q_j^{-+}w_j^{--}\right) \cdot e_j \tag{5}$$

and recognize that the node energy is a competition between two groups of aggregated input charges. For the first group the input potentials are positive and the favored node state is negative, while for the second group the input potentials are negative and the favored node state is positive. In general, the selection of the node state is a stochastic competition between the two groups. The terms within the groups are products of complementary positive and negative charge aggregations, favoring the transfer of aggregated positive input charge (e.g. q_j^{++}) to aggregated output charge (e.g. p_j^{+-}) associated with aggregated negative input charge (e.g. q_j^{+-}) and vice versa: to say it differently, these terms enable the nodes to propagate potentials through the network. Nodes without complementary inputs ($H_j(e_j) = 0$) randomly select state, thereby helping the network search for low energy configurations. Network evolution can often be accelerated (particularly in the case of time varying potentials applied to the network - see below), through the addition of recurrent connections providing each node its prior state information through an adapting, weighted edge. These recurrent connections bias the node to change states when inputs are ambiguous, small, or not complemented – in colloquial terms, when the node doesn't know what to do it learns to do something different.

The node energy expression can be made non-negative through the addition of terms that are independent of e_j. Eqn.5a is helpful for visualizing plots of the network energy evolution in the Results section below.

$$\frac{H_j(e_j)}{4} = -\left(\left(f_j^{+}\right)^2 e_j + 1\right) \cdot \left(q_j^{++}w_j^{+-} + q_j^{+-}w_j^{++}\right) - \left(\left(f_j^{-}\right)^2 e_j - 1\right) \cdot \left(q_j^{--}w_j^{-+} + q_j^{-+}w_j^{--}\right) \tag{5a}$$

We have experimented with different kinetic factors[8], but have found the following rectifying linear function to be effective

$$f_j^{\pm}(e_j) = \frac{1}{2}\left(|e_j| \mp e_j\right) \tag{6}$$

[8] $f_j^{\pm}(e_j) = \frac{1}{2}\left(1 \mp e_j\right)$ is also effective

As will be made clearer below, these factors "select" or "focus" the node energy on one of the input groups after the state selection, communicates potentials and updates the weights associated with the selected group, and stores residual charge but does not communicate potentials or update the weights associated with the unselected group. The physical interpretation of this focus is that the "kinetics" of the state selection, while enabling the thermalization of the selected group also inhibits the thermalization of the unselected of group. A more colloquial interpretation is that the node can only do "one thing at a time" or that it has the ability to "attend to the matter at hand" while "saving other matters for a later time".

We note each term in the node energy (Eqn.5) is the product of 3 factors – 2 factors that are determined by the aggregated inputs ($q_j^{\pm\pm}$ and $w_j^{\pm\pm}$) and one that is determined by the node state (e_j). This is a critical distinction from Ising and perceptron models, where the node energies are a competition among terms that are the product of 2 factors – an input and a node state. This difference is crucial as it enables the TNN to connect complementary potentials and update weights without reliance on carefully engineered network architectures and *post hoc* error assignment techniques like back-propagation.

Edge Model

The edge model drives the adaptation of the edge weights and the accumulation of edge charge associated with residual charges on the node after state selection. Because the edge model concerns only the residual charges we consider only the residual charge terms from Eqn.3 and write

$$\begin{aligned} H_j(q_j, w_j|e_j) &= \left(f_j^{+}\right)^2\left[\left(q_j^{++} - e_j w_j^{+-}\right)^2 + \left(q_j^{\pm} - e_j w_j^{++}\right)^2\right] \\ &\quad + \left(f_j^{-}\right)^2\left[\left(q_j^{--} - e_j w_j^{-+}\right)^2 + \left(q_j^{\mp} - e_j w_j^{--}\right)^2\right] \\ &= \left(-\delta_j^{++}\right)^2 + \left(-\delta_j^{+-}\right)^2 + \left(-\delta_j^{--}\right)^2 + \left(-\delta_j^{-+}\right)^2 \end{aligned} \tag{7}$$

where

$$-\delta_j^{\pm\pm} = f_j^{\pm} \cdot \left(q_j^{\pm\pm} - e_j w_j^{\pm\mp}\right) \tag{8}$$

are the residual charges that we wish to minimize through weight updates $w_{ij} \to w_{ij} + \Delta w_{ij}$. We capture this objective by rewriting Eqn.7 as

$$H_j(\Delta w_j) = \left(\sum_{i^{++}} e_i^+ \Delta w_{ij}^{++} - \sum_{i^{+-}} e_j \Delta w_{ij}^{+-} - \delta_j^{++}\right)^2 + \left(\sum_{i^{+-}} e_i^+ \Delta w_{ij}^{+-} - \sum_{i^{++}} e_j \Delta w_{ij}^{++} - \delta_j^{+-}\right)^2 + \left(\sum_{i^{--}} e_i^- \Delta w_{ij}^{--} - \sum_{i^{-+}} e_j \Delta w_{ij}^{-+} - \delta_j^{--}\right)^2 + \left(\sum_{i^{-+}} e_i^- \Delta w_{ij}^{-+} - \sum_{i^{--}} e_j \Delta w_{ij}^{--} - \delta_j^{-+}\right)^2 \quad (9)$$

As explained below, we elect to distribute the residual charges to the weight updates as

$$H_j(\Delta w_j) = \left(\sum_{i^{++}} \left(e_i^+ \Delta w_{ij}^{++} - \eta_{ij}^{++}\right) - \sum_{i^{+-}} \left(e_j \Delta w_{ij}^{+-} + \mu_{ij}^{+-}\right)\right)^2 + \left(\sum_{i^{+-}} \left(e_i^+ \Delta w_{ij}^{+-} - \eta_{ij}^{+-}\right) - \sum_{i^{++}} \left(e_j \Delta w_{ij}^{++} + \mu_{ij}^{++}\right)\right)^2 + \left(\sum_{i^{--}} \left(e_i^- \Delta w_{ij}^{--} - \eta_{ij}^{--}\right) + \sum_{i^{-+}} \left(e_j \Delta w_{ij}^{-+} + \mu_{ij}^{-+}\right)\right)^2 + \left(\sum_{i^{-+}} \left(e_i^- \Delta w_{ij}^{-+} - \eta_{ij}^{-+}\right) + \sum_{i^{--}} \left(e_j \Delta w_{ij}^{--} + \mu_{ij}^{--}\right)\right)^2 \quad (10)$$

where

$$\eta_{ij}^{\pm\pm} = \frac{|e_i|}{\Delta_j^{\pm\pm}} \cdot \delta_j^{\pm\pm}$$
$$\mu_{ij}^{\pm\mp} = \frac{|e_j|}{\Delta_j^{\pm\pm}} \cdot \delta_j^{\pm\pm} \quad (11)$$
$$\Delta_j^{\pm\pm} = \sum_{i^{\pm\pm}} |e_i| + \sum_{i^{\pm\mp}} |e_j|$$

In the implementation described below, the weight updates are sampled independently such that the time average of the cross terms in Eqn.10 vanish. We rewrite Eqn.10 as

$$H_j(\Delta w_j) \cong \sum_{i^{++}} \left(e_i^+ \Delta w_{ij}^{++} - \eta_{ij}^{++}\right)^2 + \sum_{i^{++}} \left(e_j \Delta w_{ij}^{++} + \mu_{ij}^{++}\right)^2 + \sum_{i^{\pm}} \left(e_j \Delta w_{ij}^{\pm} + \mu_{ij}^{\pm}\right)^2 + \sum_{i^{\pm}} \left(e_i^+ \Delta w_{ij}^{\pm} - \eta_{ij}^{\pm}\right)^2 + \sum_{i^{--}} \left(e_i^- \Delta w_{ij}^{--} - \eta_{ij}^{--}\right)^2 + \sum_{i^{--}} \left(e_j \Delta w_{ij}^{--} + \mu_{ij}^{--}\right)^2 + \sum_{i^{\mp}} \left(e_j \Delta w_{ij}^{\mp} + \mu_{ij}^{\mp}\right)^2 + \sum_{i^{\mp}} \left(e_i^- \Delta w_{ij}^{\mp} - \eta_{ij}^{\mp}\right)^2 \quad (12)$$

Collecting terms, completing squares, dropping terms that are independent of Δw_{ij}, and substituting the definitions from Eqn.11 yields

$$\begin{aligned} H_j(\Delta w_j) \cong & \sum_{i^{++}} \left(e_i{}^2 + e_j{}^2\right)\left[\Delta w_{ij}^{++} - \left(\frac{1}{e_i{}^2 + e_j{}^2}\right)\left(\frac{e_i}{|e_i|}\frac{\delta_j^{++}}{\Delta_j^{++}}e_i{}^2 - \frac{e_j}{|e_j|}\frac{\delta_j^{+-}}{\Delta_j^{+-}}e_j{}^2\right)\right]^2 \\ & + \sum_{i^{+-}} \left(e_i{}^2 + e_j{}^2\right)\left[\Delta w_{ij}^{+-} - \left(\frac{1}{e_i{}^2 + e_j{}^2}\right)\left(\frac{e_i}{|e_i|}\frac{\delta_j^{+-}}{\Delta_j^{+-}}e_i{}^2 - \frac{e_j}{|e_j|}\frac{\delta_j^{++}}{\Delta_j^{++}}e_j{}^2\right)\right]^2 \\ & + \sum_{i^{--}} \left(e_i{}^2 + e_j{}^2\right)\left[\Delta w_{ij}^{--} - \left(\frac{1}{e_i{}^2 + e_j{}^2}\right)\left(\frac{e_i}{|e_i|}\frac{\delta_j^{--}}{\Delta_j^{--}}e_i{}^2 - \frac{e_j}{|e_j|}\frac{\delta_j^{-+}}{\Delta_j^{-+}}e_j{}^2\right)\right]^2 \\ & + \sum_{i^{-+}} \left(e_i{}^2 + e_j{}^2\right)\left[\Delta w_{ij}^{-+} - \left(\frac{1}{e_i{}^2 + e_j{}^2}\right)\left(\frac{e_i}{|e_i|}\frac{\delta_j^{-+}}{\Delta_j^{-+}}e_i{}^2 - \frac{e_j}{|e_j|}\frac{\delta_j^{--}}{\Delta_j^{--}}e_j{}^2\right)\right]^2 \end{aligned} \tag{13}$$

The form of Eqn.13 is the motivation for the choice of distributing residual charge among the weights in Eqn.11. The sum of quadratic terms, when exponentiated according to Boltzmann statistics, results in independent Gaussian distributions with simple offsets. If the node states are constrained to be binary $e_i = \pm 1$, then the weight updates are identical for every weight in an input group, which is perhaps the least biased of the possible choices for distributing the residual charge.

Because the residual charge terms in Eqn.8 are proportional to the kinetic focusing terms, in general only a fraction of the residual charge will be dissipated as weight updates. Charge conservation requires that the undissipated fraction be retained as edge charge (which will influence future state decisions and weight updates). Hence, as the edge weight updates are sampled according to Eqn.13, the edge charges are updated as

$$q_{ij}^{\pm\pm} \to \left(1 - f_j^{\pm}\right) \cdot q_{ij}^{\pm\pm} \tag{14}$$

Because each weight is connected to two nodes, it receives an update as specified in Eqn.13 from each node. In our implementation these two updates are separate sampling events, each providing only ½ of the total update (i.e. the Gaussian offset is divided by two) in Eqn.13.

The update of weights according the technique just described is convenient because the updates can be made independently, which greatly simplifies the computational model. It does, however, introduce uncontrolled weight growth as an undesirable artifact. If we consider the collection of weights associated with node j as a vector $\vec{w}_j$ then the effect of the this independent weight updating is to on average increase the magnitude $\left|\vec{w}_j\right|$, which can be seen from

$$\langle\left(\vec{w}_j + \overrightarrow{\Delta w_j}\right)^2\rangle = \vec{w}_j{}^2 + \langle\overrightarrow{\Delta w_j}{}^2\rangle = \vec{w}_j{}^2 + \frac{n_j}{2\beta} > \vec{w}_j{}^2$$

where n_j is the number edges connected to node j and β is the inverse temperature. One solution to this problem that still allows the weight updates to performed independently is to the reduce the size of the weight update to account for this artificial weight growth. In our implementation, we reduce the size of each weight on every update by a factor

$$\vec{w}_j \to \vec{w}_j \cdot \left(\frac{1}{1 + \langle \overrightarrow{\Delta w_j}^2 \rangle / \vec{w}_j^{\,2}} \right)^{1/2} = \vec{w}_j \cdot \left(\frac{1}{1 + n_j / \left(2 \cdot \beta \cdot \vec{w}_j^{\,2} \right)} \right)^{1/2} \tag{15}$$

Note that this correction becomes small as the $\vec{w}_j^{\,2}$ becomes large, but has large effect at small weight sizes.

Bias Nodes

Network models may also include nodes that are sources and sinks of charge for the larger network. In our implementation, these nodes have predetermined node states (e.g. $e_j = \pm 1$ with some temporal pattern of changing state) and (large) fixed output weights. A single bias node is able to polarize the network around it (i.e. create some antiferromagnetic order in the connections around it) and, thereby, diffuse potential into the network. When paired with another bias node of opposite polarity, these polarized regions can evolve to form a conducting "bridge" with large weights that transport large amounts of charge through the network. See Results below for examples of these effects.

Network Effects

Unlike most neural network models, which critically depend on the network topology in order to update edge weights, the node and edge models described above are agnostic to the network topology. For most network topologies the nodes are "frustrated" because there is no way to achieve perfect antiferromagnetic order. This frustration does not prevent the network from effectively conducting charge, however, as it will partition itself as needed in order to build conducting paths. The edge charge accumulation and weight updating methods described allow the network to create "domains" of antiferromagnetic order with "domain walls" separating domains of one polarity from those of a different polarity. The node energy of Eqn.4 reflects this tendency as competition among the input groups and the domain walls are recognized as stored edge charge that the node retains on the unselected input groups. All of these effects are a consequence of the physically motivated ideas that permeate the model – charge conservation, potential propagation, dissipation driven adaptation, and kinetic focusing.

For a special class of networks that are partitioned into two groups ("bipartite" networks) in which nodes of the first partition can connect only to nodes in the second partition, the geometric frustration just described can be avoided. Nearest neighbor grid networks are inherently bipartite – the nodes fall naturally into two groups on a "checkboard" pattern. These networks are particularly attractive to study because they are easy to visualize. As externally biased nodes are introduced into bipartite networks, however, geometric frustration

can once again emerge as the node biases may conflict depending on their placement in the partitions and their relative polarity. For example, if two nodes of opposite polarity bias are placed in different partitions of a bipartite network then frustration between them is avoided and the network can evolve to transport charge between them. If on the other hand the bias nodes are placed in the same partition, then they are geometrically frustrated and the network will evolve a domain wall to separate them. These effects are elaborated in the Results section below.

The net result of frustration effects is that the network must segment itself into domains in order to minimize energy and effectively transport charge. As we will show below, these networks can rapidly and efficiently reorganize in response to changing inputs. The creation and destruction of domains in these frustrated networks is perhaps analogous the creation of "virtual networks" among collections of excitatory and inhibitory neurons in biological systems (Yufik, 2002).

Temperature

For unbiased networks, network behavior is independent of temperature because it is the only energy scale in the problem (and so the weights evolve to the scale of the temperature). As a variant in which there are two energy scales in unbiased networks, the node and weight updates can be made using different temperatures. In unbiased networks lower (higher) node temperature relative to edge temperature results in more (less) ordered networks. Ordering in the unbiased networks is also very sensitive to the connectivity of the nodes – more connectivity yields greater order – and changing the relative temperature of the node and weight updates can be used to modify the degree of ordering in unbiased networks. Externally biased nodes introduce other energy scales into the network. In general, the charge injected into the network by these bias nodes must compete with thermal fluctuations in the network nodes.

Network Simulation

The task of network simulation is to evolve the network toward global low-energy states in the presence of time varying inputs and thermal fluctuations. The technique employed here is a two-stage optimization process that first (reversibly) relaxes the node states of the entire network using a Gibb's sampling technique and then (irreversibly) relaxes the edge weight and charge states. The intuition here is that this combination of rapid, global relaxation of the network node states combined with slower, local relaxation of the edge states will connect large and small scale dynamics while using only local interactions to make the computations.

Node State Relaxation

Gibbs' sampling is a Markov Chain Monte Carlo method for sampling a multivariate distribution using conditional distributions for the variables of interest. In the work presented here, each node state is sampled in a round robin for several cycles. When the node j is reached in the round robin, its conditional state is sampled according to a Boltzmann distribution as

$$P_j\left(e_j|\boldsymbol{e} \neq e_j, \boldsymbol{w}, \boldsymbol{q}\right) = \frac{exp\left(-\beta H_j\left(e_j\right)\right)}{Z_j} \tag{16}$$

$$Z_j = \sum_{\boldsymbol{e_j}} exp\left(-\beta H_j\left(e_j\right)\right)$$

After a state e_j is selected, the nodes connected to node j are updated with new inputs, which temporarily adjust their edge input charges. After several round robin cycles, the node states are assumed to be a representative sample of the probability distribution of the entire network given by Eqn.1. Because the inputs to the network are changing, this distribution is also changing. Roughly speaking, in the work presented here, the number of cycles in the round robin was chosen to approach with the characteristic size of the network, the separation between bias nodes, or was selected through a process of trial and error. From a computational perspective, the node state relaxation process is a kind of "search" informed by prior "experience" captured in the edge states. This search is "reversible" in the sense that the edge states are not updated in the process – that is, the probability distribution in Eqn.1 is unchanged during the round robin search. From a pragmatic perspective we would like this search to be rapid as possible, and, as previously described, we have sometimes placed recurrent edges on the nodes, which bias the nodes to change state when the state choice is poorly defined by its other connections. Not surprisingly, given the objective of finding a representative, low-energy sample from a very large number of potential states, this node relaxation is the most challenging part of the model implementation.

Edge State Relaxation

After the node state relaxation process is complete the edge weights and charges are updated as described in Eqn.13. Edge state relaxation is irreversible in that residual charge is dissipated into the thermal bath and the conditional node state distributions are modified in subsequent node state relaxation cycles. Edge states update more slowly than the node states in the sense that (1) the node state relaxation process takes many cycles while the edge relaxation is a single cycle and (2) edge weight updates become relatively small as the network self-organizes.

Results

The following images are snapshots of the evolution of the network captured as images of the node states at a particular simulation step. These images are frames from a video that can be viewed online by following the links in the image description. In the images that follow, each node is one square and its state is indicated on a grayscale with black=-1 and white=+1. The examples below focus on 2-dimensional (bipartite) networks with nearest neighbor (NN) connectivity and periodic boundary conditions because these networks allow easy visualization of the network organization. The ideas presented below also apply to higher dimensional networks and networks with more complex connectivity such as randomly connected networks.

In every case, the network is initialized with random node and edge state values and allowed to evolve according to the methodology of the previous section.

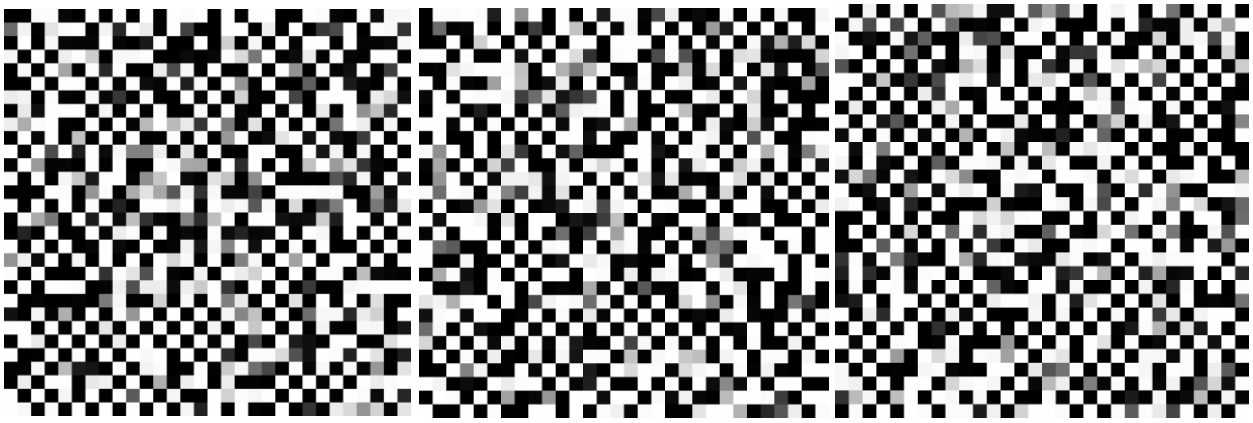
These results are the product of simulation on a laptop computer. The implementation has not been refined for efficiency of execution or speed. There is a very high degree of parallelism in the model that very likely could be effectively executed in modern architecture like multi-core CPUs, GPUs, and emerging asynchronous and neuromorphic computing systems (DeBole, et al., 2019). The code is available at https://github.com/toddhylton/Thermodynamic-Neural-Network---Public.

Isolated Networks

Figs.1-4 are sample results from isolated, undriven networks interacting with a thermal bath. Figs.1-2 are different simulations of identical networks with 4 nearest neighbor (4NN) connectivity illustrating the propensity of the network to organize. Fig.2 inverts the display polarity of one the network partitions in order to show the order more clearly (this convention is adopted in all figures except Fig.1 for bipartitioned networks) – in this way antiferromagnetic order can be visualized as ferromagnetic order. At the node and edge temperatures in Figs.1-2, ordering is local and transient. In general, as the node temperature decreases as compared to the edge temperature, ordering extends over larger spatial and temporal scales.

Fig.3 shows a larger bipartitioned network with 16 nearest neighbor (16NN) connections per node and node and edge temperatures chosen to foster slow evolution of the network from a disordered multidomain state to a single domain state with complex thermal excitations. Fig.4 shows plots of selected network statistics capturing the ordering of the network of Fig.3.

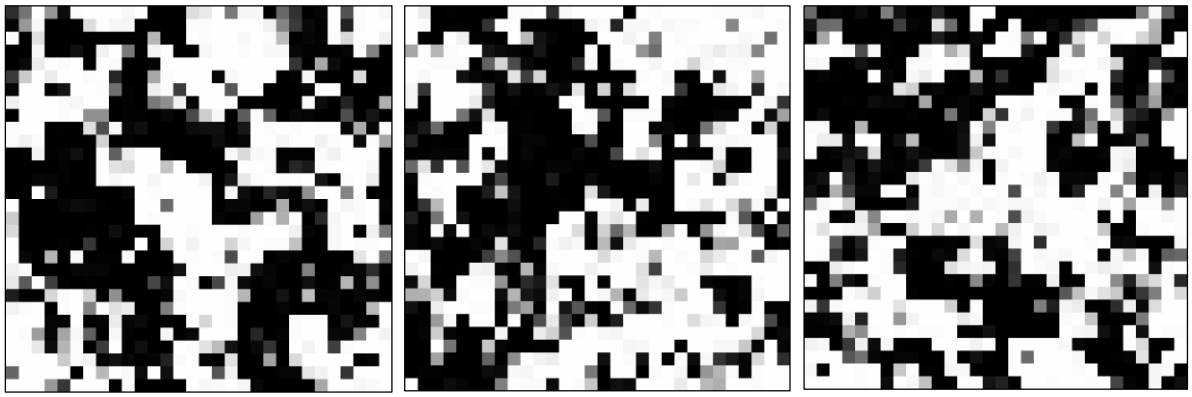
In general, the more highly connected the network, the more sensitive it becomes to the relative node and edge temperatures, with abrupt transitions from disordered to highly ordered network states occurring for small changes in temperature. Recursive edges in these models create complex temporal dynamics. In the examples that follow, the number of recursive edges was chosen to enhance network relaxation (i.e. escape local minima) but not overwhelm the dynamics.



Network: 2D, 4NN, 2 partition
Network nodes: 900
Network node states: 101
Recurrent edges per node: 1

Network relaxation steps: 10
Biasing nodes: 0
Temperature: T_{edge} / T_{node} = 1
Video: https://youtu.be/Fqq2pa5YFJo

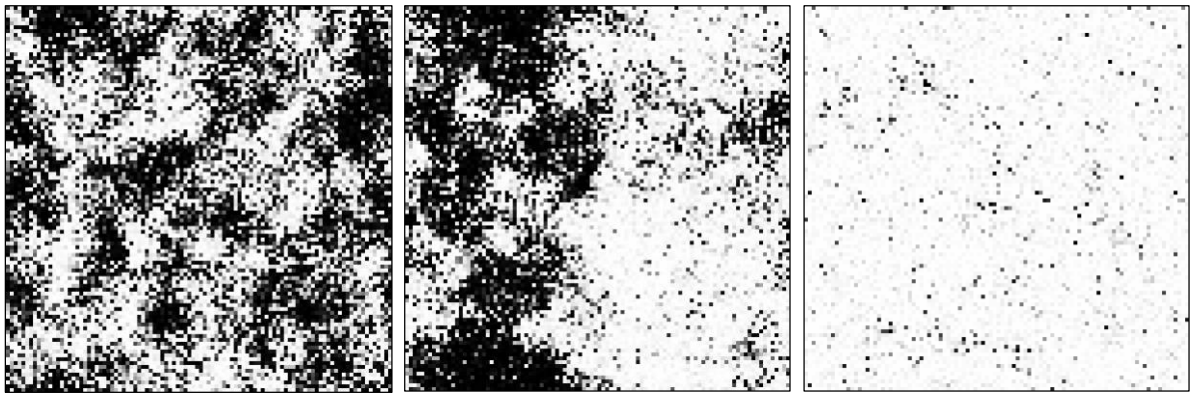
Figure 1: Representative nodes states for un-driven networks with 4 nearest neighbor connectivity. Networks are dynamic and "noisy" owing to contact with the thermal bath. The propensity for the nodes to organize anti-symmetrically is evident in the checkboard appearance of the various regions of the network. Domains are separated by less ordered domain walls.



Network: 2D, 4NN, 2 partition
Network nodes: 900
Network node states: 101
Recurrent edges per node: 1

Network relaxation steps: 10
Biasing nodes: 0
Temperature: T_{edge} / T_{node} = 1
Video: https://youtu.be/L315T9w4p6c

Figure 2: Because antisymmetric order is challenging to visualize, in these images and those that follow the order is displayed by reversing the sign of the node state in one of the partitions (i.e. on every other square on the checkboard). This change is applied only to the image display: the underlying order is still antiferromagnetic. When displaying images this way domains appear as preferentially "white" or "black". The images in this panel are from an identical network as those used in Figure 1.



Network: 2D, 16NN, 2 partition
Network nodes: 10,000
Network node states: 101
Recurrent edges per node: 2

Network relaxation steps: 10
Biasing nodes: 0
Temperature: T_{edge} / T_{node} = 1 / 67
Video: https://youtu.be/fcK4odnle_Y

Figure 3: A bipartitioned network with each node connected its 16 nearest neighbors in the opposite partition evolves from a complex, disordered dynamic state (left image) to single domain with local, spatiotemporal excitations (right image) through intermediate states of multiple, competing domains (middle image). Slight increases / decreases in the T_{node} relative to T_{edge} result in no ordering / rapid ordering. The temperatures in this simulation were chosen to show the transition from disorder to order. A similar video of a network with 40k nodes is available at https://youtu.be/3qAopGIrHoY.

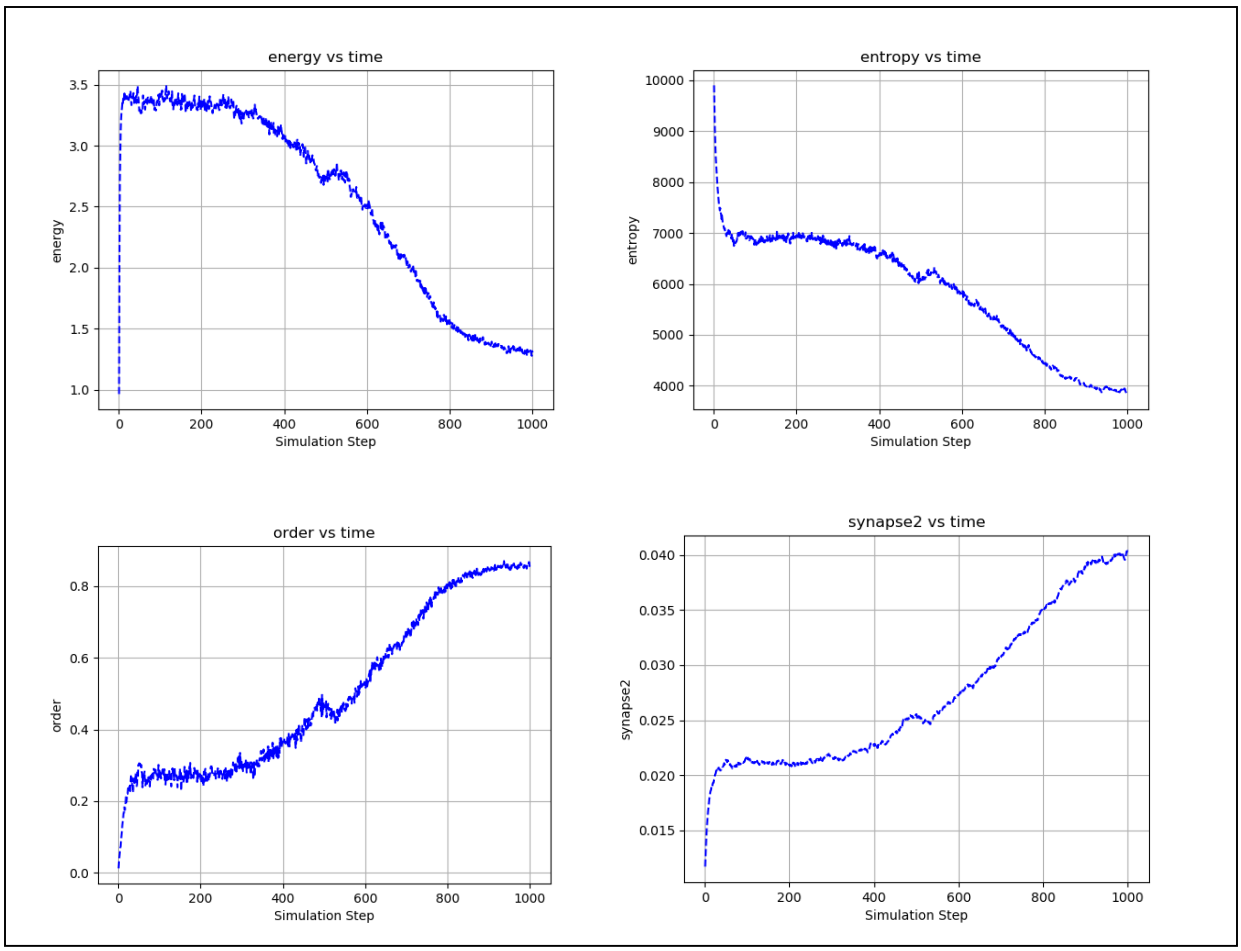
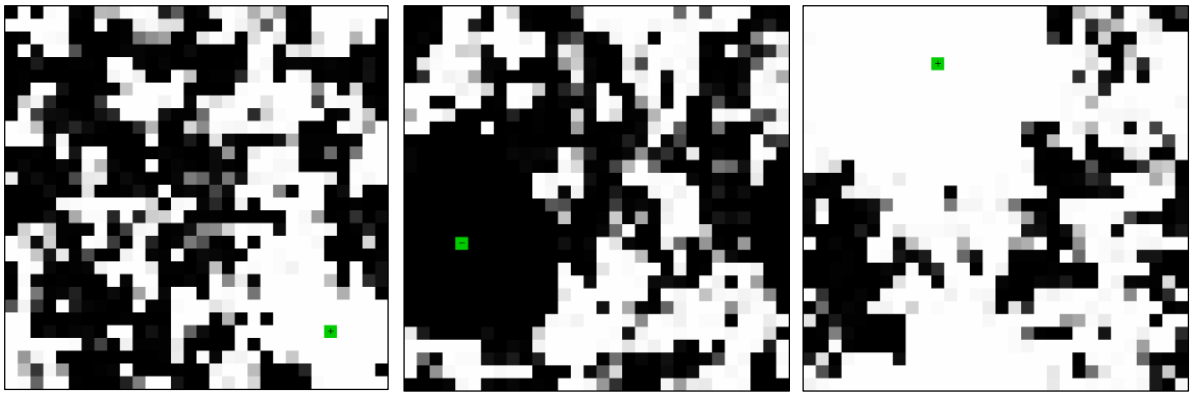


Figure 4: Temporal evolution of selected network averages for the simulation of Fig.3 – average node energy, sum of node entropy (normalized so that maximum is 10000), average edge order, and average square of the edge weights vs simulation time. Node energy from Eqn.5a. Node entropy computed from Eqn.13. Edge order is (the negative of) the product of its connected node states.

Biased Networks

Figs.5-11 are sample results from networks biased with external potentials while interacting with a thermal bath. Figs.5-7 illustrate the polarization of the network by external potentials, connection of nodes with complementary potentials, and creation of domain walls separating competing domains. Figs.8-11 show these same effects in larger networks and the ability of the networks to reconfigure to transport charge in response to time-varying external potentials. Figs.5-9 illustrate these ideas with networks having 4 nearest neighbor connectivity, while Fig. 10-11 show similar results in randomly connected, single partition networks.

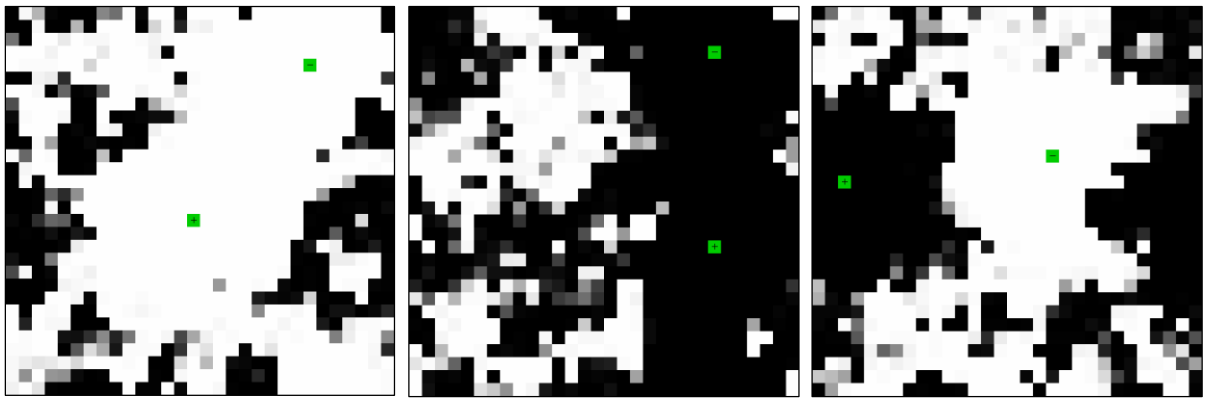


Network: 2D, 4N, 2 partition
Network nodes: 899
Network node states: 101
Recurrent edges per node: 1

Network relaxation steps: 10
Biasing nodes: 1 with
Bias node states: ±1 with periodic state change
Temperature: T_{edge} / T_{node} = 1

Video: https://youtu.be/EwXenHIXVIw | https://youtu.be/cs8ESDJooRs | https://youtu.be/7b9w_xx9tb8

Figure 5: A single positively biased node polarizes the region in its vicinity. The sequence of images is from three different simulations with increasing bias strength (increasing size of fixed edge weights connecting the bias node to its neighbors). Larger bias creates a larger region of polarization. Domain polarization changes as the biasing node changes sign. The ability of a bias node to polarize nodes in its vicinity enables it to communicate potential into the network.

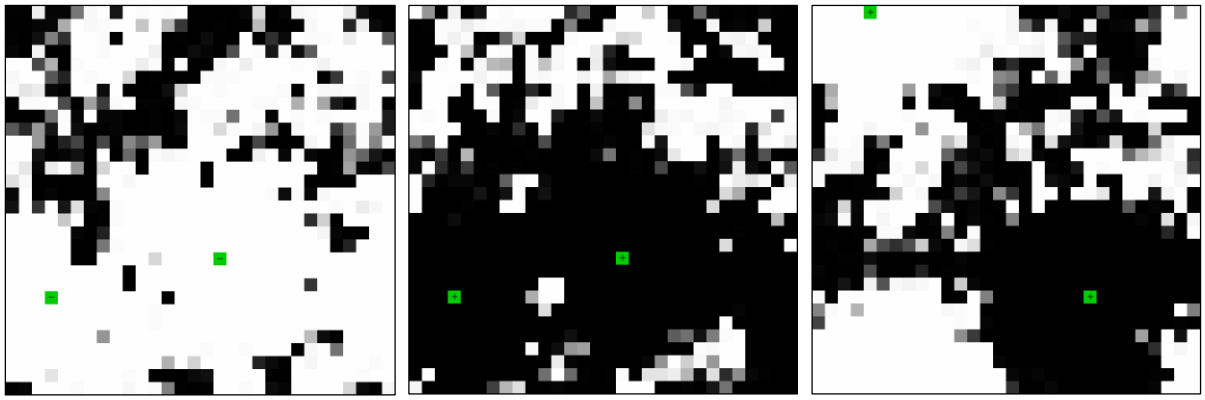


Network: 2D, 4NN, 2 partition
Network nodes: 898
Network node states: 101
Recurrent edges per node: 1

Network relaxation steps: 10
Biasing nodes: 2 with opposite polarity
Bias node states: ±1 with periodic state change
Temperature: T_{edge} / T_{node} = 1

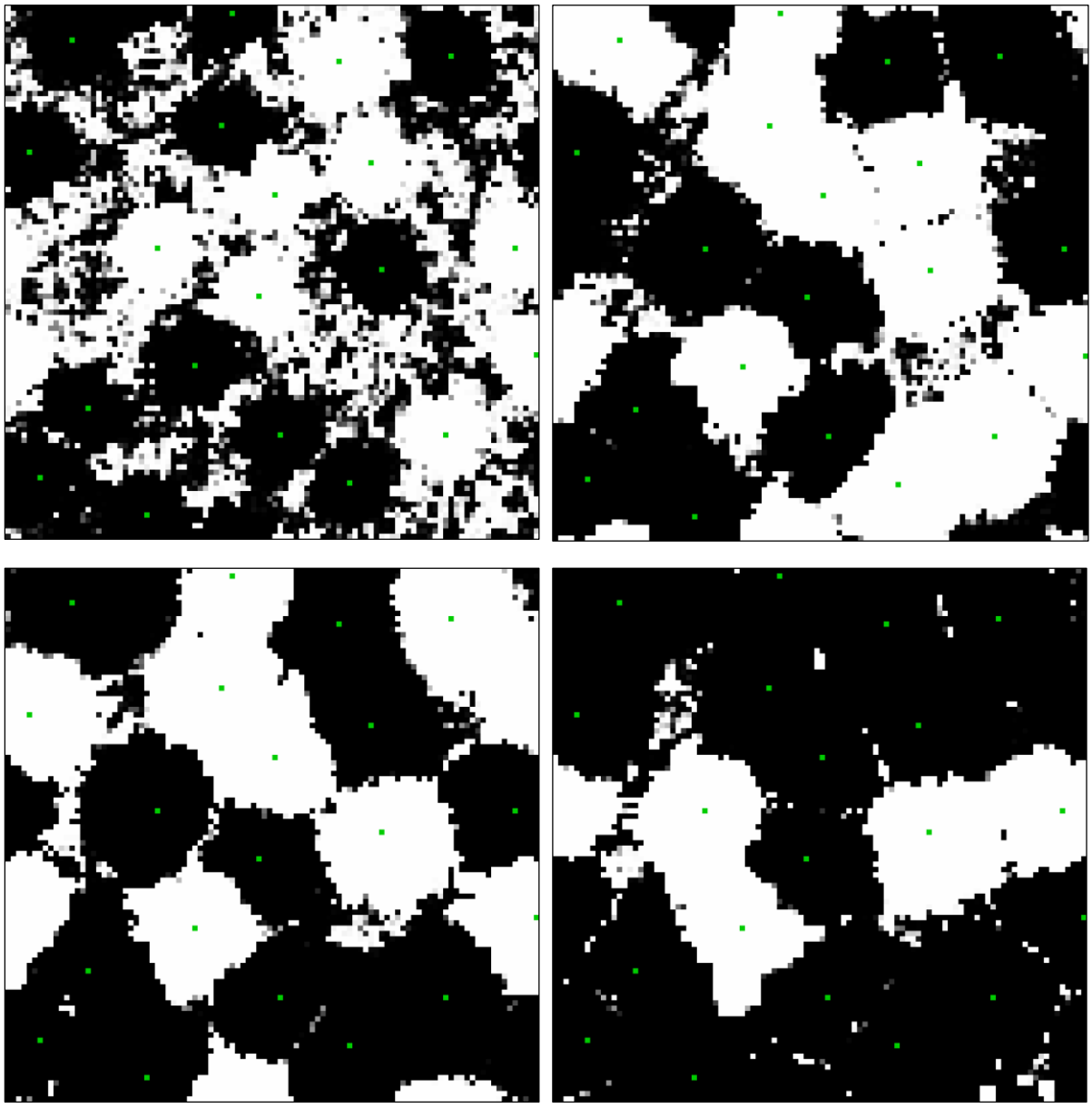
Video: https://youtu.be/yyqXsHtqfpE | https://youtu.be/xJt_dWNnlLQ | https://youtu.be/YSfSwVSOtj0

Figure 6: Two biased nodes of opposite polarity interact through the network. If the nodes are in opposite partitions, the network evolves a connection (left and center images). If the nodes are in the same partition, however, the network evolves domain walls separating the regions that they polarize (right image).



Network: 2D, 4NN, 2 partition
Network nodes: 898
Network node states: 101
Recurrent edges per node: 1
Network relaxation steps: 10
Biasing nodes: 2 with same polarity
Bias node states: ±1 with periodic state change
Temperature: T_{edge} / T_{node} = 1
Video: https://youtu.be/aQEHN4LsRxA | https://youtu.be/xHTDOVnWBCw

Figure 7: Two biased nodes of the same polarity interact through the network. If the nodes are in the same partition, they can jointly polarize the surrounding region, but do not grow strong weights between them (left and center images). If the nodes are in opposite partitions, they polarize domains of opposite polarity and create a domain wall where they intersect (right image).



Network: 2D, 4NN, 2 partition
Network nodes: 9980
Network node states: 101
Recurrent edges per node: 1
Video: https://youtu.be/amJ8_FtfP9g

Network relaxation steps: 14
Biasing nodes: 20 as 10 pairs with opposite polarity
Bias node states: ±1 with periodic state change
Temperature: T_{edge} / T_{node} = 1

Figure 8: 10 pairs of bias nodes interact through a 4 nearest neighbor network. Each pair is composed of opposite polarity (±1) nodes that change periodically in time, each of the 10 pairs with different periods. These four images show different configurations of the network as it adapts to changes in its inputs from early to late in the network evolution (left to right / top to bottom). As the edge weights grow, the domains become larger and better connected. As the input nodes change polarity, the network rapidly adapts by creating and destroying domain walls. In general, the network is challenged to connect and

separate nodes into black and white domain according to their polarity and partition (per the examples in Figs.6-7). The 2D network also creates geometric challenges in connecting nodes as two nodes that might be otherwise connected can be blocked by a competing domain. The video link shows the evolution of the network including the 14 Gibbs sampling cycles in which the network relaxes the node states – this relaxation appears as a "propagation" of nodes states stimulated by the changing polarity of the bias nodes.

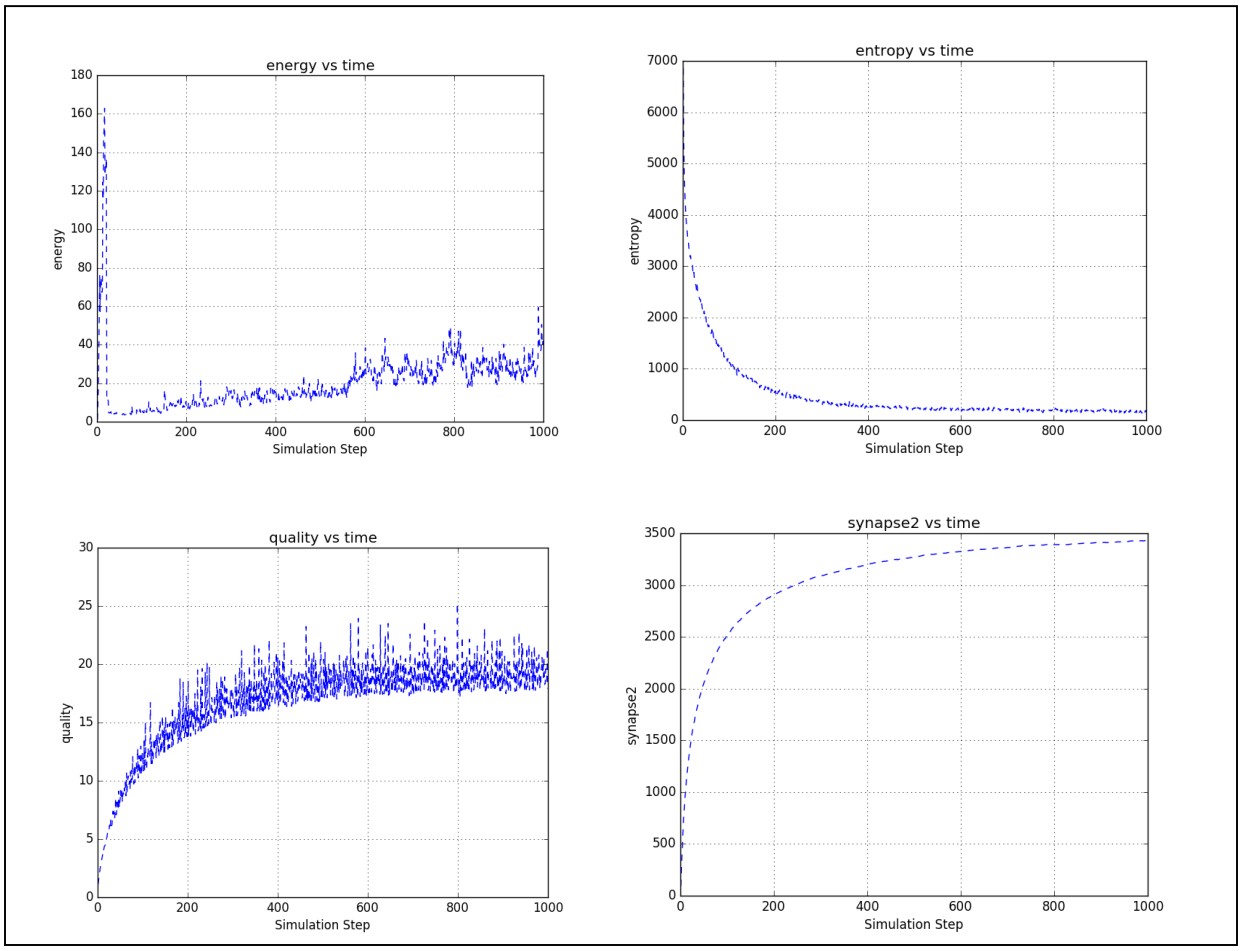


Figure 9: Temporal evolution of selected network averages for the simulation of Fig.8 – average node energy, sum of node entropy (normalized so that maximum is 10000), average node charge transport quality, and average square of the edge weights vs simulation time. Node energy from Eqn.5a. Node entropy computed from Eqn.13. Quality is defined as the sum of the square of charges transported by the nodes divided by the sum of the square of the charges dissipated by the nodes. As the network adapts to its inputs, its ability to transport charge improves. Average node energy increasing in time is an artifact of the offset applied in Eqn.5a but also reflects charge stored on the edges (particularly the recursive edges), which grows as the edge weights grow.

Network: 2D, 8 Random, 1 partition
Network nodes: 9980
Network node states: 101
Recurrent edges per node: 1
Video: https://youtu.be/14VXDNorRVY

Network relaxation steps: 10
Biasing nodes: 20 as 10 pairs with opposite polarity
Bias node states: ± 1 with periodic state change
Temperature: $T_{edge} / T_{node} = 1$

Figure 10: 10 pairs of bias nodes interact through a randomly connected network with 8 edges per node. Each pair is composed of opposite polarity (± 1) nodes that change periodically in time, each of the 10 pairs with different periods. As with the neighbor network of Fig.8, the network is challenged to connect and separate nodes into black and white domain according to their polarity and partition. Domain walls form to separate competing domains. Ordering is impossible to visualize in these networks, but the transitions in the network as the bias nodes change states are visible in the video.

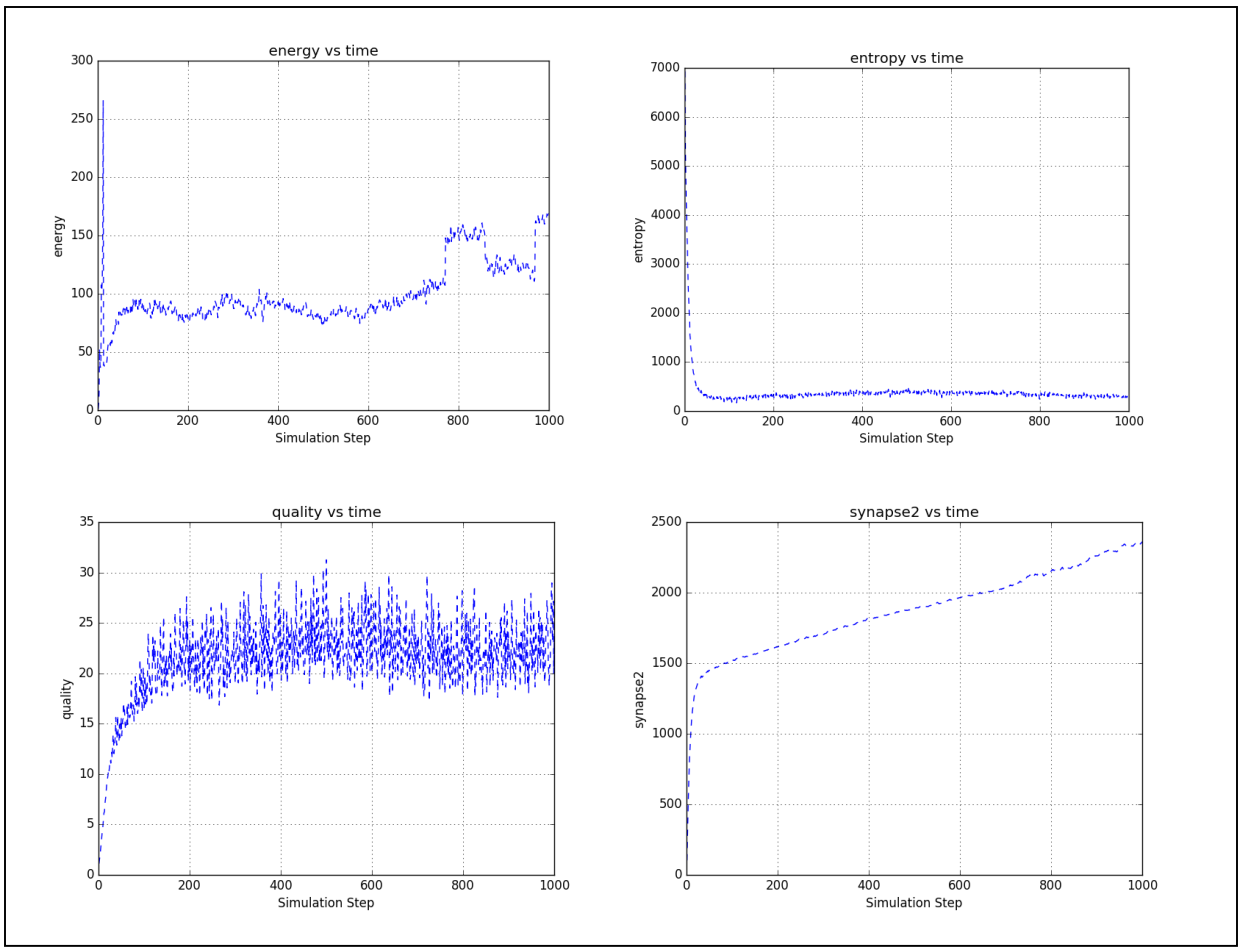


Figure 11: Temporal evolution of selected network averages for the simulation of Fig.10 – average node energy, sum of node entropy (normalized so that maximum is 10000), average node charge transport quality, and average square of the edge weights vs simulation time. Node energy from Eqn.5 modified to remove negative offsets. Node entropy computed from Eqn.13. Quality is defined as square of charge transported by the node divided by charge dissipated by the node. As the network adapts to its inputs, its ability to transport charge improves. Average node energy increasing in time reflects charge stored on the edges (particularly the recursive edges), which grows as the edge weights grow. The evolution of the random single partition network is similar to the nearest neighbor two partition network of Figure 9.

Discussion

Model Features

The following paragraphs highlight the key features of the TNN.

Conserved complementary quantities (positive and negative charge) interact but do not cancel. Accumulation of conserved quantities represent an energetic cost than can be reduced by transporting these charges through the network and connecting external sources and sinks. Local differences in these conserved quantities create the potentials that transport them. These ideas are consistent with relations among forces and fluxes in near-equilibrium thermodynamic systems (Onsager, 1931).

Dissipation of residual charge is coupled to fluctuations. Mismatch of input charges on a node creates residual charge that must be dissipated. The edge weight updates (Eqn.13) strive to eliminate this mismatch, but thermal noise in the updates means that the match is always imperfect - even in networks driven by constant external inputs evolved to a low energy steady state. Hence, dissipation in the network is inextricably linked to fluctuations in the network. Fluctuation-dissipation effects are well known from equilibrium statistical physics (Callen & Welton, 1951) and recently extended to non-equilibrium systems (Jarzynski, 1997) (Crooks, 1999).

Adaptation is coupled to residual charge dissipation. While updating weights to eliminate errors in an objective function is the foundation of most neural network models, in the TNN model it also has a physical interpretation. Namely, the dissipation of conserved quantities within an open physical system (e.g. the residual node charges for the TNN), when coupled to the system features responsible for their creation (e.g. the edge weights), can adapt the system to reduce dissipation under similar future conditions. If the environment in which this system is embedded has certain stable features, then through its interaction with that environment, the system may come to represent and predict those features and to thereby minimize internal dissipation (Still, Sivak, Bell, & Crooks, 2012). The intuition here is that the dissipation of a conserved physical quantity requires a physical structure to transport it out of the system. In the TNN model, the supposition is that the edge weight, which creates the charge imbalance, mediates the transport of the residual charge to a thermal/charge reservoir, and in the process is adapted by it. As an example from everyday life, consider a housing construction site in which certain raw materials (the conserved quantities) are cut as the house is built and residual scraps of material are produced that cannot be used. Those scraps, which must be transported away from the construction site (the dissipation), can be used to inform the acquisition of materials in future (the adaptation) and to improve the efficiency of the construction process up to the point that variances in materials and construction permit (the fluctuation). This process can become highly predictable and efficient if the same house is constructed many times, materials suppliers are reliable, and labor is consistent (a stable environment).

Edge states adapt with respect to the current state of their connected nodes without destroying state information associated with other node states. The kinetic factors in the model adapt edge states selectively depending on the node state (Eqn.6). Edge state updates associated with different configurations of the collective node states will reinforce to the degree that the nodes state configurations are similar but not cancel where they are different, thereby enforcing commonalities (generalization) while preserving differences (specialization). Referring again to the example of constructing a house, building a wall and building a floor may share similar tools and fasteners, but use different types of lumber. In adapting for these tasks we would like to generalize the adaptation for tools and fasteners while specializing the adaptation for the different types of lumber.

Rapid, global, reversible relaxation of the nodes states followed by slower, local, irreversible adaptation of the edge state creates a multiscale, complex adaptive system. The node state relaxation stage attempts to keep to keep the network in a constant state of equilibrium with respect to its time-varying inputs by adapting more quickly than the inputs change. The edge state relaxation stage attempts to refine the network structure to improve network efficiency. In the house building analogy, the large scale reversible stage might involve the delivery of the materials for the day's work (which could be returned at no cost) while the smaller scale irreversible stage might involve the many individual activities using those materials (such that they cannot be returned). A variety of natural, networked systems and models of such systems involve the idea of adaptation at different scales. Comparison of various systems related to physics, materials, ecology, biology and cognition indicate that "dual phase evolution" may explain common observations of modularity, network statistics and criticality (Paperin, Green, & Sadedin, 2011) in these diverse domains. As perhaps an indication of the modularity effect, the images and video of Fig.8 show the evolution of "modules" of nodes surrounding the external biasing nodes that connect to and disconnect from other modules of nodes as the external bias changes.

The TNN avoids many computational challenges found in other neural network models. Nodes can be connected as networks of any type without creating dynamic instabilities. Node and edge updates are continuous and online. There are few ad hoc meta-parameters: for example, there are no learning rates.

The TNN unifies concepts of fluctuation, dissipation, adaptation and equilibration under a common physical model to illustrate a self-organizing, complex, adaptive system. The model self-organizes with and without external inputs. Externally applied potentials propagate through the network by polarizing connected nodes. Self-organization is strongly modulated by network effects, the relative temperatures of the nodes and edges, and the strength of the external applied potentials.

Limitations

As mentioned previously, the most challenging part of the implementation of the model is the global search over the node state space to find a representative state. As is typical, the search for a global optimum is frustrated by local minima and there is no all-purpose algorithm to address this problem (Wolpert & Macready, 1997). In the implementation described here, this is typically recognized as a domain that fails to change state as the external potentials transition. There is little doubt that the methods used here might be improved to address this challenge, for instance by searching over more than single node state changes in the Gibbs sampling technique.

It is interesting to consider the source of this challenge in the context of the thermodynamic concepts that motivated the TNN. Every computing model is composed of a sequence of variable assignments. The ability to make these assignments effectively requires that the variables of the model be independent at the time of assignment. For example if we wish to perform the assignment $a \leftarrow b \cdot c$ then b and c must exist and be independent of a at the time of the assignment. In the model implementation described above, this limitation is recognized in the node-at-a-time round-robin search for a low energy configuration of the node states (Eqn.16) and in the approximations leading to the edge weight updates (Eqns.13 & 15). More generally, the challenge of creating the TNN can be seen as taking the relatively simple statement of Eqn.1 and the assumptions 1-11 and translating them into a sequence of variable assignments that effectively addresses the challenges of capturing the interdependencies of the state variables.

Speculation on Future Opportunities

While we can claim some success in our efforts to address the challenge just described and suppose even that there might be useful implementation of the TNN, there are certain ironies implicit in simulating complex thermodynamic systems on deterministic computing hardware. We must, for example, calculate probability distributions and generate pseudo-random numbers to sample fluctuations using computing hardware that, at great expense, is engineered, manufactured and operated to prevent fluctuations. We must, for example, at great expense, compute a representative sample of an equilibrium distribution, while every natural system does this at essentially zero cost. So, perhaps the most promising future implementations of models such as the one presented here would involve computational substrates in which the device electronics inherently perform the thermodynamic relaxation that drives the evolution of the network. For example, nodes might be constructed of multistate devices that are marginally stable at their operating temperature and that can be biased to favor transition to a particular state by the charge received from their inputs. Also, edges might be constructed of semi-stable, hysteretic resistive components (“memristors” or “memcapacitors”) that change impedance depending on the history of the current passing through them (Wang, et al., 2017). Such systems would have orders of magnitude higher energy efficiency, scalability and perhaps offer much more complex functionality than the

computational model described here. These future systems, in combination with conventional computing elements, might create the foundations for a "thermodynamic computer" that can both evolve "from below" according to the basic thermodynamics of its components and be constrained "from above" by human specified code. In such systems "thermodynamic evolution" might be an omnipresent capacity driving its self-organization toward a high-level, human specified goal.

Conclusions

We have described a neural network model comprising a collection of the nodes and edges that that organizes according to basic principles of physic and thermodynamics. Charge conservation laws and the hypothesis that nodes should evolve to transport charge effectively results in networks of nodes that organize to maximize charge transport efficiency. Node and edge state updates derive from relaxation of the network according to Boltzmann statistics. Node states relax globally and reversibly via a Gibbs' sampling methodology followed by irreversible and local relaxation of the edge states, resulting in a multiscale self-organizing, complex system. The dynamics are sensitive to network structure and temperature. Externally applied potentials diffuse into the network, establishing strong connections to complementary potentials and creating domain walls to separate competing potentials. The model integrates ideas of fluctuation, dissipation, adaptation and equilibration to illustrate the thermodynamic evolution of organization.

Acknowledgements

The inspiration for this work started roughly 12 years ago, shortly before I began working at DARPA. Since that time many people have influenced my thinking in the domains of thermodynamics, learning, neural networks, complex systems, computation, cognitive science, neuroscience and related domains that were key to the development of the Thermodynamic Neural Network. I would like to acknowledge these people in particular for their insights and support – Alex Nugent, Yan Yufik, Yaneer Bar Yam, Ben Mann, Dharmendra Modha, Narayan Srinivasa, Stan Williams, Jennifer Klamo, Filip Piekniewski, Patryk Laurent, Csaba Petre, Micah Richert, Dimitri Fisher, Tom Conte, Michael Hazoglou, Natesh Ganesh, and Ken Kreutz-Delgado.

References

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive science, 9*(1), 147-169.

Amit, D. J., Gutfreund, H., & Sompolinsky, H. (1985). Spin-glass Models of Neural Networks. *Phys. Rev. A, 32*(2), 1007-1018.

Bruce, A. (1987). Dynamics and statistical mechanics of the Hopfield model. *J. Phys. A: Math. Gen., 20*, 2909-2934.

Callen, H. B., & Welton, T. A. (1951). Irreversibility and Generalized Noise. *Physical Review, 83*(1), 34-40.

Clark, J. W. (1988). Statistical mechanics of neural networks. *Physics Reports, 158*(2), 91-157.

Crooks, G. E. (1999). Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences. *Physical Review E, 60*(3), 2721.

Glansdorff, P., & Prigogine, I. (1964). On a General Evolution Criterion in Macroscopic Physics. *Physica, 30*, 351-374.

Glauber, R. J. (1963). Time-dependent statistics of the Ising Model. *Journal of Mathematical Physics, 4*, 294-307.

Gutfreund, H. (1990). From Statistical Mechanics to Neural Networks and Back. *Physica A, 163*, 373-385.

Hinton, G., & Sejnowski, T. (1986). Learning and relearning in Boltzmann machines. In *Parallel distributed processing: Explorations in the microstructure of cognition* (pp. 282-317).

Hoffmann, H., & Payton, D. W. (2018). Optimization by Self-Organized Criticality. *Scientific reports, 8*(1), 2358. doi:10.1038/s41598-018-20275-7

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *PNAS, 79*, 2554-2558.

Jarzynski, C. (1997). Nonequilibrium Equality for Free Energy Differences. *Physical Review Letters, 78*(14), 2690-2693.

Jun, J. K., & Hubler, A. H. (2005). Formation and structure of ramified charge transportation networks in an electromechanical system. *PNAS, 102*(3), 536-540. doi:10.1073/pnas.0406025102

Kirkpatrick, S., & Sherrington, D. (1978). Infinite-ranged Models of Spin Glasses. *Phys. Rev. B, 17*(11), 4384-4403.

Levin, E., Tishby, N., & Solla, S. (1990). A Statistical Approach to Learning and Generalization in Layered Neural Networks. *Proceedings of the IEEE, 78*(10), 1568-1574.

Reka, A., & Barabasi, A.-L. (2002). Statistical mechanics of complex networks. *Rev. Mod. Phys., 74*, 47-97.

Schlogl, F. (1967). On the Statistical Foundation of the Thermodynamic Evolution Criterion of Glansdorf and Prigogine. *Annals of Physics, 45*, 155-163.

Schneider, E., & Kay, J. (1994). Life as a Manifestation of the Second Law of Thermodynamics. *Mathl. Comput. Modelling, 19*(6-8), 25-48.

Schrodinger, E. (1943). *What is Life?* Cambridge: University Press.

Sompolinsky, H. (1988). Statistical Mechanics of Neural Networks. *Physics Today*, 70-80.

Suzuki, M., & Kubo, R. (1968). Dynamics of the Ising Model near the Critical Point. I. *Journal of the Physical Society of Japan, 24*(1), 51-60.

Watkin, T. L., Rao, A., & Biehl, M. (1993). The statistical mechanics of a learning rule. *Rev. Mod. Phys., 65*(2), 499-556.

Yufik, Y. M. (2002). How the mind works: An exercise in pragmatism. *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02, 3*, 2265-2269.