Complex Systems 8 (1994) 337–346

Computational Properties of Boolean Networks

Sergey A. Shumsky
Lebedev Physics Institute, Leninsky pr.53, Moscow, Russia

Abstract. This paper considers computational characteristics of boolean automata networks with random interconnections. The subjects of interest are: the number of stationary points, the convergence of the dynamic flow to these points, compactness of the attraction basins, number of logical switchings during the run, and informational content of the phase portrait. For a special class of *uniform* diluted nets, these quantities are found to depend basically on two integrated characteristics of network elements.

1. Introduction

Recent developments in the fields of neural [1] and boolean networks [2, 3] have made it clear that the underlying mathematics of such systems provides the basis for programming the latest generation of highly parallel computers. In network architectures, traditional programming gives way to *learning* procedure, that is, the selection of phase portraits with the desired features by fine-tuning the parameters of the network elements. This raises the following fundamental question: are there *order parameters*, that is, integrated characteristics of network elements, which are crucial for computational behavior?

Compelling evidence for the existence of such parameters is found in recent studies of the dynamical properties of random boolean nets [4–9]. Results have been obtained for the Kauffman NK-model [10], constructed from N randomly chosen boolean functions of K inputs, receiving their values from K randomly chosen network elements. A dramatic change from stochastic dynamics to orderly behavior of the network was observed at some critical value of K. The order parameter for this "phase transition" was proposed by Derrida and Pomeau [5].

This paper extends these considerations to a wider class of boolean networks, including automata networks, where the next state of the automaton depends on its current state. Using the approach of previous work [11], we show that in such networks there exist two phase transitions, namely, the loss of stability of the trajectories and of the fixed points. In the NK-model these phase transitions coincide.

Another purpose of this paper is to interpret the observed dynamics, treating the latter as a calculation process, with the fixed points representing the possible results.

After formulation of the model in section 2, we speculate on what dynamical properties allow "learning" in section 3, and what differentiates memory and universal data processing systems in section 4. Section 5 summarizes and concludes the paper.

2. Model description

The networks under consideration consist of N two-state, coupled automata that update their states in parallel according to the mapping

$$x_i \Rightarrow \phi_i(x_i; x_{j_{i1}}, \ldots, x_{j_{iK}}), \qquad 1 \le i \le N.$$

In vector notation, we have

$$\mathbf{x} \Rightarrow \boldsymbol{\phi}(\mathbf{x}).$$

To avoid boolean algebra, we will use the Ising notation $\mathbf{x}, \boldsymbol{\phi} \in \{\pm 1\}^N$.

We are interested in the statistical properties of an ensemble of such networks, where each configuration $\boldsymbol{\phi}(\mathbf{x})$ is characterized by its statistical weight $\mu(\boldsymbol{\phi}(\mathbf{x}))$. Our ensemble consists of all networks constructed from a given infinite set of binary automata, in which the automaton ϕ has probability $\mu(\phi)$. The automata are chosen independently

$$\mu(\boldsymbol{\phi}(\mathbf{x})) = \prod_{i=1}^{N} \mu(\phi_i(\mathbf{x})) \tag{1}$$

and all possible interconnections between them are equiprobable, that is,

$$\mu(\phi_i(\mathbf{x})) = \mu(\phi_i(\pi\mathbf{x})), \qquad (\pi x_j = x_{\pi j}, \pi \in \{S_N \mid \pi i = i\}). \tag{2}$$

Compare (2) with the Kauffman NK-model, where all inputs are chosen at random, that is, $\pi \in S_N$.

The dynamics of a single system taken from our ensemble is deterministic, as is the dynamical flow. The latter is governed by the Liouville equation, which is, however, too complicated to deal with. A typical approach is to replace the deterministic dynamics with stochastic dynamics, in which the phase trajectories are generated in random fashion. Some statistical properties of the stochastic trajectories correspond to those of the real trajectories in our ensemble.

To this end, consider a chain of stochastic processes which generates the trajectories with increasing accuracy:

$$\begin{aligned}
&0\text{: } P\{\mathbf{x}_0\mathbf{x}_1\ldots\} = P(\mathbf{x}_0)P(\mathbf{x}_1)\ldots \\
&1\text{: } P\{\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2\ldots\} = P(\mathbf{x}_0)P(\mathbf{x}_1 \mid \mathbf{x}_0)P(\mathbf{x}_2 \mid \mathbf{x}_1)\ldots \\
&2\text{: } P\{\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3\ldots\} = P(\mathbf{x}_0\mathbf{x}_1)P(\mathbf{x}_2 \mid \mathbf{x}_0\mathbf{x}_1)P(\mathbf{x}_3 \mid \mathbf{x}_1\mathbf{x}_2)\ldots
\end{aligned}$$

and so on. The conditional probabilities on the right-hand side of the kth-order approximation are defined in the usual way:

$$P(\mathbf{x}_k \mid \mathbf{x}_0\ldots\mathbf{x}_{k-1}) = P(\mathbf{x}_0\ldots\mathbf{x}_k)/P(\mathbf{x}_0\ldots\mathbf{x}_{k-1}).$$

The 0th-order approximation is trivial. Since there is a unique phase vector originating in each phase state, we have $P(\mathbf{x}_0) = \Omega^{-1}$, where $\Omega = 2^N$ is the phase volume of the networks. The first-order approximation, called the "mean field theory," corresponds to a reconfiguration of the network after each step. This prevents the stochastic trajectory from being captured by some fixed point. These fixed points appear only in the second-order approximation, since $P(\mathbf{x}_2 \mid \mathbf{x}_1\mathbf{x}_1) = \delta_{\mathbf{x}_2\mathbf{x}_1}$ for deterministic systems. Thus, we will limit ourselves to the latter approximation, which allows us to analyze the relaxation dynamics of the networks.

The second-order approximation requires knowledge of the moments $\langle \phi_i(\mathbf{x}_0) \rangle$ and $\langle \phi_i(\mathbf{x}_0)\phi_i(\mathbf{x}_1) \rangle$, where the brackets $\langle \cdots \rangle$ represent the ensemble average. This follows from the expressions

$$\begin{aligned} P(\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2) &\equiv \langle \delta_{\mathbf{x}_1,\boldsymbol{\phi}(\mathbf{x}_0)} \delta_{\mathbf{x}_2,\boldsymbol{\phi}(\mathbf{x}_1)} \rangle \\ &= \langle 2^{-2N} \prod_{i=1}^{N} [1 + x_{1i}\phi_i(\mathbf{x}_0)][1 + x_{2i}\phi_i(\mathbf{x}_1)] \rangle \\ &= 2^{-2N} \prod_{i=1}^{N} [1 + x_{1i}\langle \phi_i(\mathbf{x}_0) \rangle + x_{2i}\langle \phi_i(\mathbf{x}_1) \rangle \\ &\qquad + x_{1i}x_{2i}\langle \phi_i(\mathbf{x}_0)\phi_i(\mathbf{x}_1) \rangle] \end{aligned} \tag{3}$$

$$P(\mathbf{x}_0\mathbf{x}_1) \equiv \langle \delta_{\mathbf{x}_1,\boldsymbol{\phi}(\mathbf{x}_0)} \rangle = 2^{-N} \prod_{i=1}^{N} [1 + x_{1i}\langle \phi_i(\mathbf{x}_0) \rangle], \tag{4}$$

where we have used the representation $\delta_{x,y} = (1 + xy)/2$ for the δ-function with Ising arguments, and property (1) of the measure μ. (The first-order approximation, according to (4), may be expressed by $\langle \phi_i(\mathbf{x}_0) \rangle$.)

To simplify our analysis, we restrict ourselves to the case of *uniform* ensembles, where the ensemble averages depend only on the relative distances between the state points: $P(\mathbf{x}_1 \mid \mathbf{x}_0) = P(|\mathbf{x}_1 - \mathbf{x}_0|)$, $P(\mathbf{x}_2 \mid \mathbf{x}_1\mathbf{x}_0) = P(|\mathbf{x}_2 - \mathbf{x}_1|, |\mathbf{x}_1 - \mathbf{x}_0|, |\mathbf{x}_2 - \mathbf{x}_0|)$. (The distance between two states in the Hamming sense $|\mathbf{x}_1 - \mathbf{x}_0|$ constitutes the number of automata that must change their states to transfer the system from state $\mathbf{x}_0$ to state $\mathbf{x}_1$.)

For uniform ensembles, the above moments should have the following dependence on $\mathbf{x}_0$ and $\mathbf{x}_1$:

$$\langle \phi_i(\mathbf{x}_0) \rangle = \nu_0 x_{0i} \tag{5}$$

$$\langle \phi_i(\mathbf{x}_0)\phi_i(\mathbf{x}_1) \rangle = w(p) + w_0(p)x_{0i}x_{1i} \tag{6}$$

where $p = |\mathbf{x}_1 - \mathbf{x}_0|/N$ is the normalized length of the phase vector emanating from $\mathbf{x}_0$ to $\mathbf{x}_1$. Reference [12] provides greater detail; for example, one can show that $\nu_0 = w_0(0)$. Thus the second-order approximation for uniform ensembles is determined by two functions $w(p)$ and $w_0(p)$ with $p \in [0, 1]$. The latter represent the statistical properties of the basic set of automata.

The uniform ensemble is a straightforward generalization of the Kauffman NK-model where "memory" effects are absent, that is, $w_0(p) = 0$. The ensemble of neural nets $\phi_i = \text{sgn}(\sum_j J_{ij}x_j + \theta_i)$, with a symmetric distribution of weights J_{ij} and thresholds θ_i, gives another example of the uniform ensemble.

In a uniform ensemble it is natural to analyze the stochastic dynamics in terms of the lengths of the trajectory vectors (that is, the number of automata switchings). The second-order approximation provides the conditional probability P_{ml} that a vector of length l follows a vector of length m. The spectrum of vector lengths in the phase portraits of our ensemble W_m, obtained by the first-order approximation, gives the initial condition for this Markov process.

Substitution of (5) and (6) in expressions (3) and (4) allows one to calculate P_{ml} and W_m. Compact expressions may thus be obtained for the corresponding generating functions [12]:

$$\widehat{W}(s) \equiv \sum_{m=0}^{N} W_m s^m = [1 + (s-1)(1-\nu_0)/2]^N \tag{7}$$

$$\hat{P}_m(s) \equiv \sum_{l=0}^{N} P_{ml} s^l = \left(\frac{A_+ + A_- s}{A_+ + A_-}\right)^{N-m} \left(\frac{B_+ + B_- s}{B_+ + B_-}\right)^m , \tag{8}$$

where

$$\begin{aligned} A_\pm(p) &\equiv 1 + \nu_0 \pm (\nu_0 + w + w_0) \\ B_\pm(p) &\equiv 1 - \nu_0 \pm (\nu_0 + w - w_0) \\ p &= m/N. \end{aligned}$$

From (8), we have $P_{0l} = \delta_{0l}$, since $A_-(0) = 0$ by definition.[1] Thus the above Markov process has one absorbing state, $m = 0$, representing all fixed points in the phase portrait. The mean number of fixed points is given by (7), namely, $\Omega_0 = \Omega W_0 = (1+\nu_0)^N$.

In the Kauffman NK-model, $\phi_i(\mathbf{x}_0)$ does not correlate with x_{0i}, so $\nu_0 = 0$ and the mean number of fixed points is unity. In our ensemble, $\nu_0 \geq 0$ is a measure of how much subsequent states depend on current states. Roughly speaking, in data processing systems, ν_0 constitutes the fraction of memory cells.

Questions of interest which we address in the next section are: when do existing fixed points attract the trajectories and when do nearby problems converge to the same answer?

3. Computability and learning

The stochastic dynamics given in (8) depend on the entire functions $w(p)$ and $w_0(p)$. But a qualitative description requires only their behavior as $p \to 0$, which determines the stability of trajectories and fixed points to small perturbations.

[1] Note that $w(0) + w_0(0) = 1$ follows from (6).

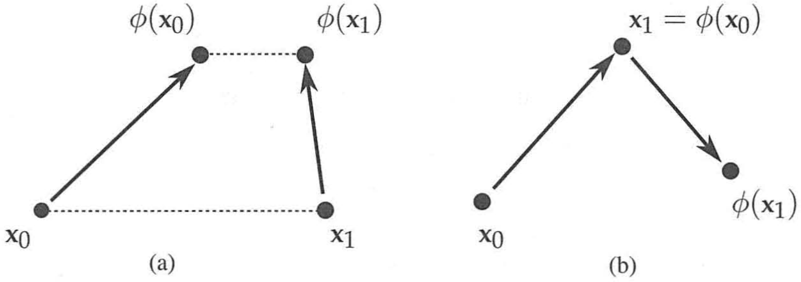

Figure 1: (a) Trajectory stability analysis and (b) convergence analysis.

To show this, consider first the distance between two nearby phase points in *different* trajectories $|\mathrm{x}_1 - \mathrm{x}_0|/N = p \ll 1$ (see Figure 1(a)). The mean distance between them at the next step, given by

$$\begin{aligned}\langle|\phi(\mathrm{x}_1) - \phi(\mathrm{x}_0)|\rangle &= (N - \langle\phi(\mathrm{x}_1)\phi(\mathrm{x}_0)\rangle)/2 \\ &= (N - Nw - \mathrm{x}_0\mathrm{x}_1 w_0)/2 \\ &= NA_-(p)/2 + |\mathrm{x}_0 - \mathrm{x}_1| w_0,\end{aligned}$$

will increase or decrease depending on the value of the characteristic multiplier κ_0 [5]:

$$\kappa_0 \equiv \lim_{\mathrm{x}_1 \to \mathrm{x}_0} \frac{\langle|\phi(\mathrm{x}_1) - \phi(\mathrm{x}_0)|\rangle}{|\mathrm{x}_1 - \mathrm{x}_0|} = A'_-(0)/2 + \nu_0. \qquad (9)$$

This result implies that $A_-(p)$ may be expanded in powers of p. This is valid, at least, for diluted networks with $K \ll N$, where the parameter of expansion is pK (see [11]).

Now consider the lengths of two consecutive vectors in the *same* trajectory in the vicinity of a fixed point: $p = |\mathrm{x}_1 - \mathrm{x}_0|/N \ll 1$, $\mathrm{x}_1 = \phi(\mathrm{x}_0)$ (see Figure 1(b)). The mean length of the next vector depends on the length of its predecessor, that is,

$$\langle|\phi(\mathrm{x}_1) - \phi(\mathrm{x}_0)|\rangle \equiv \sum_l P_{ml} l = \hat{P}'_m(1), \qquad (\mathrm{x}_1 = \phi(\mathrm{x}_0)).$$

Applying (8), in the limit $m \ll N$ one obtains the characteristic multiplier

$$\kappa \equiv \lim_{\mathrm{x}_1 \to \mathrm{x}_0} \frac{\langle|\phi(\mathrm{x}_1) - \phi(\mathrm{x}_0)|\rangle}{|\mathrm{x}_1 - \mathrm{x}_0|} = \frac{A'_-(0)}{2(1+\nu_0)}, \qquad (\mathrm{x}_1 = \phi(\mathrm{x}_0)), \qquad (10)$$

which determines the convergence of the trajectories to the fixed points, and differs from (9) in the case of nonzero ν_0: $\kappa_0 = (1 + \nu_0)\kappa + \nu_0$.

The above calculations give rise to the following qualitative classification of network behavior:

- For networks with κ in the range $0 < \kappa < (1 - \nu_0)/(1 + \nu_0)$, that is, $\kappa_0 < 1$, existing fixed points attract the phase trajectories, and their attracting basins are compact. In other words, nearby trajectories converge to the same fixed point. This case is desirable for various recognition systems based on neural network architectures. Basins with a relatively simple structure permit generalization during the learning process. Such networks are capable of learning.

- For networks satisfying $(1-\nu_0)/(1+\nu_0) < \kappa < 1$, the fixed points attract the trajectories, but their attracting basins are not compact, that is, points close to each other may belong to different basins. Due to the complexity of the basin structure, learning by examples is impossible in this case. The full prescription for each phase trajectory is required, which may be considered as traditional programming.

- For $\kappa > 1$, fixed points repel the trajectories, and such networks are not suited for data processing: a typical calculation never converges to any definite result.

4. Memory versus universal computer

So far we have discussed the qualitative properties of network dynamics. Now we extend our consideration to certain quantitative characteristics related to parallel computations. To this end, we associate an ensemble of networks, assembled from a given set of basic elements with a *computer.* Construction of a specific network configuration, that is, the process of embedding a given algorithm constitutes *programming.* (In this section we do not distinguish between traditional programming and learning.) The statistical properties of the basic set of automata give rise to the statistical characteristics of computations and programming in such parallel computers.

Consider first the *complexity of computations* given by the amount of information processed during the course of a typical computation. We define this quantity as the mean number of automata switchings per run, that is, averaged over all possible initial conditions.

For the sake of simplicity, consider networks where the mean vector length is relatively small, say, $\overline{m} \ll N$—that is, networks with a large fraction of memory cells $(1 - \nu_0) \ll 1$ (since $\overline{m} = \widehat{W}'(1) = (1 - \nu_0)N/2$). Thus one can use the approximation $m/N \ll 1$ globally, and the expression (8) simplifies to

$$\hat{P}_m(s) \approx \exp[m\kappa(s-1)] \equiv [\hat{f}(s)]^m. \tag{11}$$

In (11), the generating function for the vector following a vector of length m is the product of m generating functions, indicating the independence of m probabilistic processes. This fact has a simple physical meaning. Recall that a vector of length m represents m automata switchings at the corresponding time step. Thus the stochastic dynamics (11) may be taken to be the reproduction of independent automata switchings, with the mean number of offsprings given by $\hat{f}'(1) = \kappa$.

With this interpretation in mind, one immediately has the average number of offsprings of one initial switching in the nth generation κ^n. The total number of offsprings is $\sum_{n=0}^{\infty} \kappa^n = (1-\kappa)^{-1}$. Because of the statistical independence indicated by (11), m initial switchings leads to $m/(1-\kappa)$ switchings during the course of computation. Averaging over the initial conditions results in

$$\overline{I} = \sum_{m=0}^{N} W_m \frac{m}{1-\kappa} = \frac{\overline{m}}{1-\kappa}. \tag{12}$$

This quantity it taken to be the mean number of pieces of information processed by a typical system from the given ensemble during the course of a computation. It increases monotonically from $\overline{m}$ for systems that find an answer in a single step $\kappa \to 0$ and is thought of as *memories.* The complexity of computations tends to infinity for $\kappa \to 1$, the latter being the cut-off point between ordered and chaotic dynamics. This type of computation "at the edge of chaos" [13] is usually associated with a *universal computer* since such a network may embed an arbitrarily complex algorithm. We present here additional evidence why the computers with $\kappa \to 1$ are treated as universal.

To this end, consider the number of algorithms present in a given ensemble, that is, those which may be programmed on a given computer. The logarithm of this quantity constitutes the entropy of the phase portraits of a given ensemble,

$$H \equiv -\sum_{\mu=1}^{\Omega}\sum_{\lambda=1}^{\Omega} p_{\mu\lambda} \ln(p_{\mu\lambda}), \tag{13}$$

where $p_{\mu\lambda}$ is the probability of a vector from state μ to state λ. The maximum entropy $H_{max} = \Omega \ln \Omega$ corresponds to random phase portraits, but the constraints posed by the choice of the basic automata set decrease this quantity. In the present framework, these constraints are represented by spectrum W_m and matrix P_{ml}. Let μ and λ represent the states with vectors of lengths m and l, respectively. Thus $p_{\mu\lambda}$ is proportional to W_m and P_{ml}. The probability that the vector terminates at a particular state is inversely proportional to the number of available states $V_m W_l$, with $V_m = \binom{N}{m}$ being the number of states at a distance m from a given phase point. Finally, one obtains the second-order approximation $p_{\mu\lambda} = W_m P_{ml}/V_m W_l$. Collecting in (13) the states with vectors of equal lengths one obtains:

$$H_2 = -\Omega \sum_{m=0}^{N}\sum_{l=0}^{N} W_m P_{ml} \ln(W_m P_{ml}/V_m W_l) = H_1 - \Delta H$$

with

$$H_1 = -\Omega N \left(\frac{1+\nu_0}{2} \ln \frac{1+\nu_0}{2} + \frac{1-\nu_0}{2} \ln \frac{1-\nu_0}{2} \right), \tag{14}$$

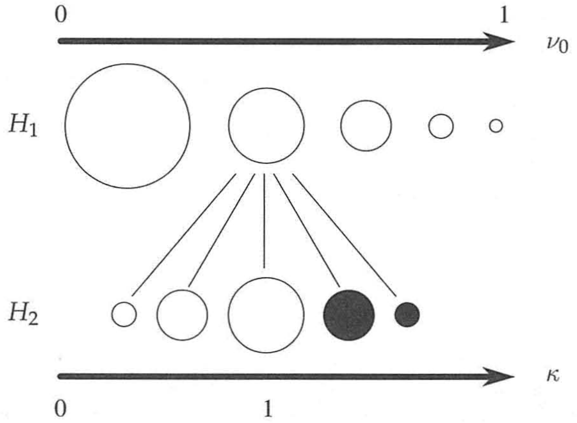

Figure 2: Representation of the ensemble entropy in the first two orders of approximation. The solid circles denote systems with $\kappa > 1$, which are not computers.

the entropy of the first-order approximation (when $P_{ml} = W_l$), and

$$\Delta H = \Omega \overline{m}(1 - \kappa + \kappa \ln \kappa), \qquad (\overline{m} \ll N), \tag{15}$$

the additional entropy decrease in the second-order approximation.

The first-order result (14) indicates that the number of algorithms increases monotonically as the mean number of automata switchings $\overline{m} = N(1-\nu_0)/2$ decreases (i.e., with an increasing degree of parallelism of computations). The mean number of stationary points $\Omega_0 = (1+\nu_0)^N$, in contrast, decreases. Thus, parallel computers provide a greater number of algorithms for a smaller number of solutions.

For a given degree of parallelism, that is, given ν_0, the number of available algorithms depends on κ. According to (15), H_2 increases monotonically with increasing $\kappa \in [0, 1]$, reaching its maximum $H_2 = H_1$ at $\kappa = 1$ (see Figure 2). Thus all the algorithms with a given degree of parallelism may be programmed on a computer with $\kappa \to 1$. That is why such computers are referred to as universal computers.

5. Conclusions

This paper discusses the features of boolean network dynamics in relation to its computational properties. The novelty of our approach is that in general there exist three dynamical regimes of network behavior, separated by two "phase transitions" in parameter space, where the trajectories and fixed points lose their stability. Networks with very complicated dynamical behavior, nevertheless converging to some fixed point, may be interpreted as following the instructions of some program. We note that simple and quickly

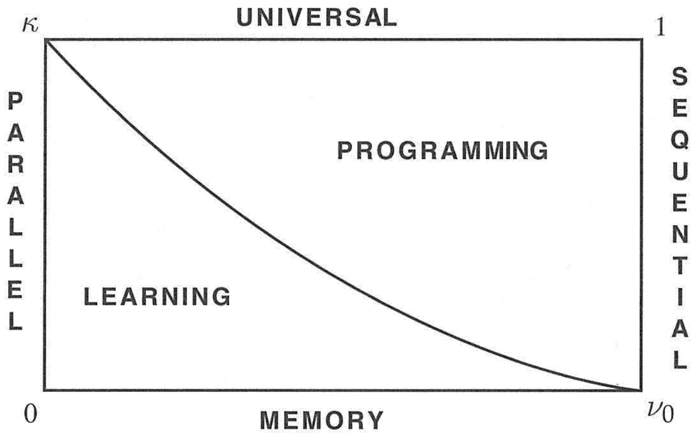

Figure 3: A two parameter classification of computers.

converging dynamics resemble the associative recall of neural networks. Figure 3 illustrates a two parameter classification of computational properties of boolean networks, emerging in the second-order approximation, which describes the convergence to the fixed points. It shows that the capability for *learning* requires a certain degree of parallelism and "nonuniversality" of computations.

The present study can easily be extended to cope with various types of networks. For example, *classifier systems* may be mapped onto boolean nets [14]. The structure of classifiers thus determines the dynamics of the entire system, and may be chosen in accordance with the above theory. Phase transitions similar to those found in boolean nets are also observed in *cellular automata* [15, 13] where unknown order parameters are anticipated [16]. The values of κ and ν_0 may turn out to be important parameters. Many other connectionist models [17] may also be analyzed using the present approach.

References

[1] D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks* (Cambridge: Cambridge University Press, 1989).

[2] I. Aleksander, editor, *Neural Computing Architectures* (London: Kogan Page, 1988).

[3] K. Y. M. Wong and D. Sherrington, "Theory of Associative Memory in Randomly Connected Boolean Neural Networks," *Journal of Physics A*, **22** (1989) 2233–2263.

[4] S. A. Kauffman, "Emergent Properties in Random Complex Automata," *Physica D*, **10** (1984) 145–156.

[5] B. Derrida and Y. Pomeau, "Random Networks of Automata: A Simple Annealed Approximation," *Europhysics Letters*, **1** (1986) 45–49.

[6] H. J. Hilhorst and M. Nijmeijer, "On the Approach of the Stationary State in Kauffman's Random Boolean Network," *Journal de Physique (France)*, **48** (1987) 185–191.

[7] G. Weisbuch and D. Stauffer, "Phase Transition in Cellular Random Boolean Nets," *Journal de Physique (France)*, **48** (1987) 11–18.

[8] D. Stauffer, "Random Boolean Networks: Analogy with Percolation," *Philosophical Magazine*, **56** (1987) 901–916.

[9] S. A. Kauffman, "Requirements for Evolvability in Complex Systems: Orderly Dynamics and Frozen Components," *Physica D*, **42** (1990) 135–152.

[10] S. A. Kauffman, "Metabolic Stability and Epigenesis in Randomly Connected Nets," *Journal of Theoretical Biology*, **22** (1969) 437–467.

[11] S. A. Shumsky, "Phase Portrait Characteristics of Random Logical Networks," *Journal of the Moscow Physical Society*, **2** (1992) 263–281.

[12] S. A. Shumsky, "Statistical Mechanics of Distributed Computations," *Math. Modelling* (to appear).

[13] C. G. Langton, "Computation at the Edge of Chaos: Phase Transitions and Emergent Computation," *Physica D*, **42** (1990) 12–37.

[14] S. Forrest and J. H. Miller, "Emergent Behavior in Classifier Systems," *Physica D*, **42** (1990) 213–227.

[15] B. Derrida and D. Stauffer, "Phase-Transitions in Two-Dimensional Kauffman Cellular Automata," *Europhysics Letters*, **2** (1986) 739–745.

[16] W. Li, N. H. Packard, and C. G. Langton, "Transition Phenomena in Cellular Automata Rule Space," *Physica D*, **45** (1990) 77–94.

[17] J. D. Farmer, "A Rosetta Stone for Connectionism," *Physica D*, **42** (1990) 153–187.