Asynchronous Dynamics of Random Boolean Networks (Extended Abstract)

Craig Gotsman, Eli Shamir and Daniel Lehmann

Dept. of Computer Science,
The Hebrew University,
Jerusalem, Israel

Abstract

A generalization of neural networks, *boolean networks* have attracted much attention in the past few years. Random (n, k)-networks consist of n processors, each connected randomly to k others, computing random k-input boolean functions. The dynamic behaviour of these networks has been studied extensively. We examine the *asynchronous* dynamics of these networks, and prove a basic result: Convergence to fixpoints is assured for almost all random (n, k)-networks, at the limit $n \longrightarrow \infty$, provided $k > \log n$. The proof of this result lies in random graph theory. We believe it is the first time this branch of mathematics has been used to analyze dynamic behaviour of networks.

1 Introduction

Consider a network of n processors where each processor is connected to exactly k others. Each processor has an internal binary state and is capable of computing a specific k-input boolean function. Temporal dynamics of these networks can be observed by initializing the processor's internal states to binary values, and updating them by applying each processor's function to the k values of its neighbors. Dynamics *regimes* can be either *synchronous* or *asynchronous*. In a synchronous regime, the values of all the processors are updated simultaneously, as a function of the previous state of the system. In an asynchronous regime, each processor updates its internal value independently of the others, subject to the single constraint that all processors update at the same average rate. Obviously, any dynamics in system lacking a central clocking mechanism must be asynchronous.
Synchronous dynamics define a deterministic mapping of the finite state space $\{0,1\}^n$ into itself. Dynamic trajectories in the state space are characterized by transient and cycle sequences, not necessarily short. The dynamic properties of synchronous boolean networks have been studied extensively ([10], [7], [4], [5]). Asynchronous dynamics differ fundamentaly from synchronous dynamics, since the associated mapping of the state space is one to many, and the dynamics probabilistic. In this paper we prove a basic result on the dynamic behaviour of asynchronous networks.

2 The Model

2.1 Network Structure

Let n and k be two natural numbers such that $k \leq n$. Let $G =< V, E >$ be an random directed graph such that $|V| = n$ and $\forall v \in V$, $in - degree(v) = k$. V is a random $k - regular$ digraph. Denote $V = \{v_1, .., v_n\}$. The graph G defines the *topology* of the boolean network; vertices correspond to processors and edges to wired connections. (v_i, v_j) is a directed edge of G iff v_i is an input of v_j. From now on, the graph will be called the *network topology* and the vertices *processors*. Let $L : V \longrightarrow \{0, 1\}$ be a binary *labelling* of the processors. These are the internal values of the processors. The current *state* of the network is defined as the vector $x = (x_1, .., x_n) = (L(v_1), .., L(v_n))$. The set of 2^n possible states is called the *state space*. In general, the variables $x_1, .., x_n$ will denote the arbitrary position of the network in the state space.
Associate with each processor v_i a random $k - input$ boolean function $f_i : \{0,1\}^k \longrightarrow \{0,1\}$. The k inputs to f_i are $x_{i_1}, .., x_{i_k}$, determined by the network topology. Call the resulting network a *random* $(n, k) - network$. A random (n, n)-network will be abbreviated to a random $n - network$. Denote by $\mathcal{G}_{n,k}$ $(\mathcal{G}_n)$ the class of all random (n, k)-networks (n-networks), and by $G_{n,k}$ (G_n) an element of $\mathcal{G}_{n,k}$ $(\mathcal{G}_n)$.
Given n and k, a $G_{n,k}$ may be constructed by connecting each one of n processors to k randomly chosen processors and initializing the processor's truth tables with random k-input boolean functions.

2.2 Network Dynamics

Given $G_{n,k}$, define the *transition table* of the network $F : \{0,1\}^n \longrightarrow \{0,1\}^n$ by $F_i(x_1, .., x_n) = f_i(x_{i_1}, .., x_{i_k})$ where $x_{i_1}, .., x_{i_k}$ represent the internal values of the neighbors of v_i. If $k = n$, the transition table is just the concatenation of the truth tables of the individual processors, i.e. a random $n \times 2^n$ binary matrix and F is just a random mapping from $\{0,1\}^n$ into itself. If $k < n$, the transition table is obtained by duplicating the truth tables of the individual processors 2^{n-k} times, once for each combination of binary states for the processors which are not its neighbors. The individual columns of F are still independent, but within each column dependencies exist. Therefore, F is no longer a random mapping.
The transition table defines a unique *successor* $F(x)$ for each vector x in state space. This is a possible *synchronous transition* of the network. *Synchronous dynamics* of the network is the trajectory in state space $x(0), x(1), x(2), ...$ traversed by the system in time. It is obtained by initiating at some state $x(0)$ and iterating through the transition table indefinitely ($x(1) = F(x(0))$, $x(2) = F(x(1))$ etc.). Since there are a finite number (2^n) of possible states, network dynamics from any initial state must reduce to a cycle in state space after long enough. The sequence of states traversed before entering the cycle is called the *transient sequence*, and the sequence traversed in the cycle the *cycle sequence* or *limit cycle*. If the cycle is of length 1 (consists of one solitary state), the system has converged to a fixpoint. In network dynamics, the interesting phenomena, for obvious reasons, are the cycle sequences, especially the fixpoints.
In *asynchronous dynamics*, only one randomly chosen processor updates its value at each time step. The effect of this mode is that dynamics trajectories are now random walks in state space.

3 State Space

3.1 The Hypercube

The state space of $G_{n,k}$ is the set of n-bit binary vectors, which are the vertices of the n-dimensional hypercube Q^n (or $n - cube$, for short). The directed n-cube $\vec{Q}^n$ may be constructed from the n-cube by connecting vertices to their n possible neighbors with directed edges. Altogether there are $n2^n$ directed edges. Since, in asynchronous dynamics, only one processor is updated at any given time, dynamic trajectories are random walks on a subgraph of $\vec{Q}^n$. The edges of the subgraph are determined from the transition function F, by the following rule:

$$(x_1, .., x_i, .., x_n) \longrightarrow (x_1, .., \bar{x}_i, .., x_n) \qquad iff \qquad F_i(x_1, .., x_i, .., x_n) = \bar{x}_i \tag{1}$$

In general x will have $d(x, F(x))$ out-edges, where $d(x, y)$ is the Hamming distance between x and y. During the random walk, all possible transitions (out-edges) from a state (vertex) x are equiprobable ($\frac{1}{n}$), and there is a probability $(1 - \frac{d(x,F(x))}{n})$ of staying in place.
Denote by Q_p^n the class of random subgraphs of Q^n, where each edge is chosen independently with probability p. Similarly, denote by $\vec{Q}_p^n$ the class of random directed subgraphs of $\vec{Q}^n$, where each directed edge is chosen independently with probability p. We say that *almost all* graphs in Q_p^n ($\vec{Q}_p^n$) have property P if the probability that any graph in Q_p^n ($\vec{Q}_p^n$) has P tends to 1 as $n \longrightarrow \infty$.
In this model, it is easily seen that there is a one-to-one correspondence between $\mathcal{G}_n$ and $\vec{Q}^n_{\frac{1}{2}}$. For the case $k < n$, the edges of $\vec{Q}^n$ are not chosen independently. Because of the dependencies in the columns of the transition table, the edges of $\vec{Q}^n$ are partitioned into groups of 2^{n-k} edges. All edges in each group appear (or disappear) together in the resulting subgraph. Examples of subgraphs of $\vec{Q}^n$ corresponding to specific networks appear in Figs. 1-3.

3.2 No Self-Connections

Consider the subset of $\mathcal{G}_{n,k}$ where the topology of $G_{n,k}$ is simple (has no loops). This can occur only if $k \leq n - 1$. In this case, each f_i is not directly dependent on the variable x_i, yielding:

$$F_i(x_1, .., x_i, .., x_n) \;=\; F_i(x_1, .., \bar{x}_i, .., x_n) \qquad i = 1, 2, .., n \tag{2}$$

The random subgraphs of $\vec{Q}^n$ formed by these $G_{n,k}$ have an interesting structure; between any pair of neighbors there exists one and only one directed edge. These graphs are analogous to *tournaments* on general directed graphs.

3.3 Fixpoints

In state space, fixpoints are points x for which $x = F(x)$. These fixpoints are also *attractors* of the network dynamics (synchronous and asynchronous), in the sense that many dynamic trajectories terminate in these states. For large n, the number of fixpoints of networks in $\mathcal{G}_n$ is approximately

Poisson distributed with $\lambda = 1$, (see [6] for a detailed proof), therefore

$$\lim_{n \longrightarrow \infty} Prob(\ G_n \ has\ at\ least\ one\ fixpoint\ ) = 1 - \frac{1}{e} \tag{3}$$

For $\mathcal{G}_{n,k}$, $k < n$, we can prove that the expected number of fixpoints is still exactly 1, but the distribution is no longer Poisson. However, approximate calculations and simulation results suggest that for $k > \log n$, the distribution is not much different from that of $k = n$. When the network topology is sparse ($k < \log n$), the distribution changes drastically, reaching a probability of 1 for exactly one fixpoint in the extreme case of $k = 0$, i.e. all processors have constant values.
In the hypercube formulation, a fixpoint is a vertex with no outgoing edges.

4 The Convergence Theorem

We are now ready to state the central theorem of this paper. Let us begin with a definition:

Definition 1 *Let $G_{n,k}$ be an element of $\mathcal{G}_{n,k}$. Call $G_{n,k}$* **convergent** *if for all initial states $x \in Q^n$, all asynchronous dynamics trajectories initiating at x terminate at a fixpoint with probability* 1.

The probabilistic part of the definition takes into account the fact that trajectories may wander in cycles endlessly, even though there is a positive transition probability to a fixpoint. However, this occurs with probability 0.

Theorem 1 *Define*

$$P_{n,k} \ = \ Prob(\ G_{n,k} \ is\ convergent \mid G_{n,k} \ has\ at\ least\ one\ fixpoint\ ) \tag{4}$$

then

$$lim_{n \longrightarrow \infty} P_{n,k} = 1 \qquad k = \log n, .., n \tag{5}$$

This theorem implies that for almost all large $G_{n,k}$ with fixpoints, asynchronous dynamics convergence to a fixpoint is guaranteed for any initial state.

Proof: We will prove the theorem for $\mathcal{G}_n$. For $\mathcal{G}_{n,k}$, the proof is considerably more complex, relying on the theory of stochastic processes. The complete proof will be given fully in the final paper.
The proof for $\mathcal{G}_n$ relies on results from random graph theory. In what follows, we assume familiarity with the basics of this theory. For a complete exposition, we refer the reader to [2].
Call a directed graph *strongly connected* if for any two vertices v_1, v_2 in the graph, there is a directed path from v_1 to v_2. Call a vertex v in a directed graph *semi-isolated* if it has no incoming edges (a *source*) or no outgoing edges (a *sink*). Following [3], call of type A those graphs in $\vec{Q}^n_{\frac{1}{2}}$ which consist of a strongly connected subgraph of size $2^n - m$ and m semi-isolated points connected to the subgraph for some m.
We now prove the following lemma:

Lemma 1 *Denote by $P(A,n)$ the probability that a directed random graph in $\vec{Q}^n_{\frac{1}{2}}$ belongs to the type A. Then*

$$\lim_{n \longrightarrow \infty} P(A,n) = 1. \tag{6}$$

Proof of the lemma completes the proof of the theorem because, in this model, an asynchronous dynamics fixpoint is a sink, and the lemma guarantees that from any other vertex (initial state), there is a directed path (a positive transition probability) to all sinks.
This lemma is analogous to the one proved in [9] for $Q^n_{\frac{1}{2}}$, and in [3] for general directed random graphs.
Note that we have transformed a dynamic problem into a static combinatorial one. The theorem says nothing about the rate of convergence or the relative probabilities of convergence between the fixpoints.

Proof: Erdos and Spencer proved ([9]) that almost all random graphs in $Q^n_{\frac{1}{2}}$ consist only of a simply connected component of size at least 2^{n-1} and isolated vertices (vertices with no connecting edges). For $p > \frac{1}{2}$, almost all graphs in Q^n_p are connected. Since $\vec{Q}^n_{\frac{1}{2}}$ has a simple edge (ignoring direction) between two neighboring vertices with probability $\frac{3}{4}$, $\vec{Q}^n_{\frac{1}{2}}$ is almost always simply connected, eliminating the possibility of graphs with totally isolated vertices (no incoming or outgoing edges).
Identical calculations to those in [9] yield the following similar result for $\vec{Q}^n_{\frac{1}{2}}$:

For almost all graphs in $\vec{Q}^n_{\frac{1}{2}}$, any vertex with at least one outgoing edge has an outgoing directed path to at least half the other vertices in the graph. Similarily, any vertex with at least one incoming edge has an incoming path from at least half the other vertices.

Now consider any sink (fixpoint) x and any non-sink y of the graph. Since the graph is simply connected, x has at least one incoming edge. Therefore x has incoming paths from at least half the vertices in the graph. The vertex y has at outgoing edge, therefore there exist outgoing paths from y to at least half the other vertices. If this is the case, there must exist a vertex z (in the intersection of the two halves), such that there is a directed path from x to y via z. This completes the lemma. ∎

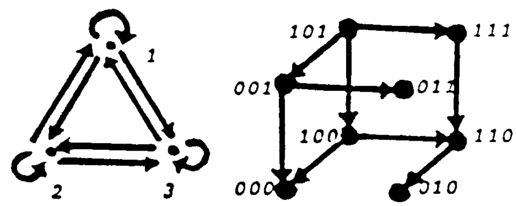
Note 1 *We conjecture that analogously to the case of the general directed random graph in [3], the probability of $\vec{Q}^n_{\frac{1}{2}}$ being strongly-connected (at the limit $n \longrightarrow \infty$) is precisely e^{-2}.*

5 Convergence Rates

Theorem 1 asserts that for large $G_{n,k}$ with fixpoints, there is a positive transition probability between any non-fixed point in the network state space and any fixed point. At this stage, we have not calculated these transition probabilities, which would lead to estimates on convergence rates. Full simulation and analytic results will be reported in the final paper.

$$f_1 = x_1 \cdot x_2 \cdot x_3$$
$$f_2 = x_1 + x_2 + x_3$$
$$f_3 = \bar{x}_1 \cdot x_2 \cdot x_3$$

x	$F(x)$
000	000
001	010
010	010
011	011
100	010
101	010
110	010
111	110

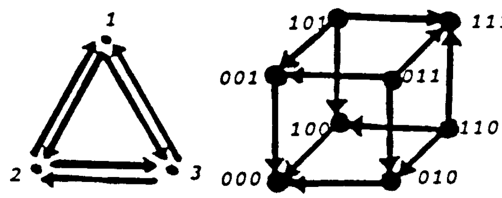


(a) (b) (c) (d)

Figure 1: A random 3-network.
(a) Processor functions. (b) Transition table; fixpoints are $\{000, 010, 011\}$. (c) Topology graph. (d) Corresponding subgraph of $\vec{Q}^3$.

$$f_1 = x_2 \cdot x_3$$
$$f_2 = x_1 \cdot x_3$$
$$f_3 = x_1 \cdot x_2$$

x	$F(x)$
000	000
001	000
010	000
011	100
100	000
101	010
110	001
111	111

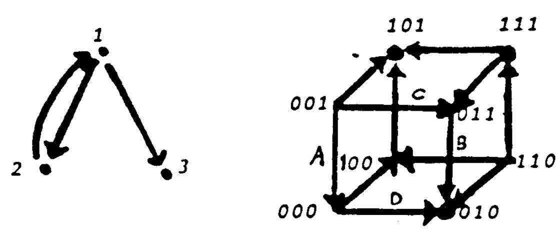


(a) (b) (c) (d)

Figure 2: A random (3,2)-network.
(a) Processor functions. (b) Transition table; fixpoints are $\{000, 111\}$. (c) Topology graph with no self-connections. (d) Corresponding subgraph of $\vec{Q}^3$. Note that there is one and only onedirected edge between any two neighbors.

$$f_1 = \bar{x}_2$$
$$f_2 = \bar{x}_1$$
$$f_3 = x_1$$

x	$F(x)$
000	110
001	110
010	010
011	010
100	101
101	101
110	001
111	001



(a) (b) (c) (d)

Figure 3: A random (3,1)-network.
(a) Processor functions. (b) Transition table; fixpoints are $\{010, 101\}$. (c) Topology graph. (d) Corresponding subgraph of $\vec{Q}^3$. Note that edges A and B appear together, as do edges C and D.

6 Conclusion

We have proven that for almost all large random boolean (n,k)-networks, asynchronous dynamics, the most realistic, guarantee convergence to fixpoints, which exist for most of these networks. We showed this by transforming the dynamic problem to a static one and then applying random-graph results. These techniques have proved to be extremely powerful, enlarging the set of mathematical tools now in use in network analysis.

Can similar results be proved for random neural networks ? We conjecture that similar results hold. Simulations carried out by Hopfield ([8]), and by Crisanti and Sompolinsky ([1]) suggest that large random fully-connected neural networks are also convergent. As of yet, this lacks formal proof and remains an open question.

References

[1] Crisanti A. and Sompolinsky H. Dynamics of spin systems with randomly asymmetric bonds: Ising spins and Glauber dynamics. *Manuscript*, 1988.

[2] Bollobas B. *Random Graphs.* Academic Press Inc., London, 1985.

[3] Palasti E. On the strong-connectedness of directed random graphs. *Studia Scientiarum Mathematicarum Hungarica*, (1):205–214, 1966.

[4] Robert F. Basic results for the behaviour of discrete iterations. In E. Bienenstock, editor, *Disordered Systems and Biological Organization*, Springer-Verlag, 1986.

[5] Weisbuch G. Networks of automata and biological organization - mini review. *Journal of Theoretical Biology*, (121):255–267, 1986.

[6] Rubin H. and Sitgreaves R. *Probability Distributions Related to Random Transformations of a Finite Set.* Technical Report 19A, Appl. Maths. and Stat. Lab, Stanford University.

[7] Coste J. and Henon M. Invariant cycles in the mapping of n integers onto themselves. Comparison with Kaufmann binary network. In E. Bienenstock, editor, *Disordered Systems and Biological Organization*, Springer-Verlag, 1986.

[8] Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sc. USA*, 79:2554–2558, 1982.

[9] Erdos P. and Spencer J. Evolution of the n-cube. *Comp. and Maths. with Applications*, (5):33–39, 1979.

[10] Kauffman S. Boolean systems, adaptive automata, evolution. In E. Bienenstock, editor, *Disordered Systems and Biological Organization*, Springer-Verlag, 1986.