Self-Organizing Topology Evolution of Turing Neural Networks

Christof Teuscher and Eduardo Sanchez

Logic Systems Laboratory, Swiss Federal Institute of Technology,
CH-1015 Lausanne, Switzerland.
E-mail: {name.surname}@epfl.ch, URL: http://lslwww.epfl.ch.

Abstract We present Turing's neural-network-like structures (unorganized machines) and compare them to Kauffman's random boolean networks (RBN). Some characteristics of attractors are briefly presented. We then apply a self-organizing topology evolving algorithm to Turing's networks and show that the network evolves towards an average connectivity of $K_C = 2$ for large systems ($N \to \infty$).

1 Introduction

In a little-known paper entitled "Intelligent Machinery" [7, 13], Turing had already investigated connectionist networks at the end of the forties, almost simultaneously with McCulloch and Pitts [9]. Copeland and Proudfoot write: "Turing was probably the first person to consider building computer machines out of simple, neuron-like elements connected together in to networks in a largely random manner" [5].

Since Turing, many researcher has been interested in randomly connected networks of simple elements. Allanson [1] investigated the properties of randomly connected networks of simple, biologically plausible neurons. Rozonoér [10] analyzed the properties of networks consisting of elements whose properties depend on parameters chosen at random. The connections among the elements were also established at random. He suggested that "[...] objects of this type may present a certain interest in connection with physiological models and, possibly, will have direct technical applications in the future" [10]. In 1971, Amari [2] published a paper on the characteristics of randomly connected threshold-element networks with the intention of understanding some aspects of information processing in nervous systems. He showed that two statistical parameters are sufficient to determine the characteristics of such networks. Random boolean networks (RBN) have been seriously investigated many years after Turing by Weisbuch [14] and Kauffman [8]. Asynchronous RBN have been investigated in [6].

Above all, Turing's neural-network-like machines are interesting since they are build up from all and the same very simple element (node) with the same logical function and since the interconnections might be configured (opened or closed) by means of an internal or external agent. The simplicity of the networks is essential for an efficient simulation of large networks and for hardware implementations. Furthermore, they offer many interesting characteristics of complex dynamical systems. The goal of this paper is to present a self-organizing topology evolving algorithm that evolves the average connectivity of the network towards a critical value.

In section 2, we present Turing's connectionist machines, an extension made by the authors, and some basic properties. Section 3 presents Kauffman's RBN and their properties. They are compared to Turing's unorganized machines. We briefly discuss the characteristics of attractors in Turing networks in section 4. Section 5 presents a self-organizing topology evolving algorithm for Turing networks that organizes the connectivity of a network towards a critical value. Section 6 concludes the paper.

2 Unorganized Machines

In this section, we shall present the neuron-like machines Turing described in his 1948 paper [13]. For a more detailed description, see also [5, 11, 12].

The term *unorganized machines* has been defined by Turing in a rather inaccurate and informal way. An *unorganized machine* is a machine that is built up "[...] in a comparatively unsystematic and random way from some kind of standard components" [13, p. 9]. In terms of a digital system, the basic unit which Turing describes can be straightforwardly defined as an edge-triggered D flip-flop preceded by a two-input NAND gate. A positive-edge-triggered D flip-flop samples its D input and changes its Q output only at the rising edge of the controlling clock (CLK) signal. A global clock generator is used to synchronize the units.

An A-*type unorganized machine* is a machine built up from N units (neurons). Each unit receives inputs from exactly two other units. A second type of unorganized machine is called B-*type unorganized machine*. A B-type machine is an A-type machine where each connection between two nodes has been replaced by a small A-type machine, as shown in Figure 1(a). This small A-type machine functions as a sort of switch (modifier) that is either *opened* or *closed* (b). Turing simply wanted to create a switch based on the same basic element as the rest of the machine. The B-type link can be in three different states of operation: (1) it may invert the incoming signal (closed/enabled connection), (2) it may interrupt the incoming signal and put a constant 1 on its output (opened/disabled connection), or (3) it may act as in (1) and (2) alternately.

So far, the two unorganized machines we have described are, once initialized, no longer modifiable. It would clearly be interesting to modify the machine's interconnection switches (synapses) at runtime. Such a possibility would allow the use of an online learning algorithm that modifies the network's interconnections in order to train a net. Turing proposed to replace a B-type link by a connection as shown in Figure 1(c). Each interconnection has two additional *interfering inputs* I_A and I_B, i.e., inputs that affect the internal state of the link. By supplying appropriate signals at I_A and I_B we can set the interconnection into state (1) or (2). We will call this type of link a BI-type link (the I stands for "interference").

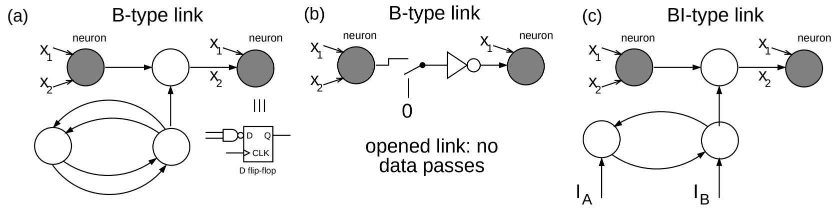

Figure 1. Turing's B-type links. Each B-type link is a small 3-node A-type machine. Part (b) shows a functional diagram of the switch incorporated in each node-to-node link: the example shows a disabled link). Part (c) shows a link with two interfering inputs that affect the internal state of the link.

By means of this type of link, an external or internal agent can organize an initially random BI-type machine by disabling and enabling connections.

It is easy to see that—when abstracting from the one-clock delay of the D flip-flop—the principal computing element (one node) of an A-type network is the NAND operation since each node contains one 2-input NAND gate. It thus directly follows that every logical function can be computed by an A-type network since NAND units form a logical basis. On the other hand, one can see that a B-type node together with its two associated input links is nothing more that a simple OR gate—again, when abstracting from the one-clock delay of the D flip-flops—and thus not all boolean functions can be computed by a B-type machine. However, it is generally desirable to work with networks that offer universal computability. We therefore propose the TB-type link shown in Figure 2(a)—which is functionally equivalent to Copeland's and Proudfoot's proposal [5], but simpler. The additional node simply inverts the input signal. Figure 2(b) shows how two interfering inputs may be added to the introverted pair of the TB-type link. We call the resulting link TBI-*type link*.

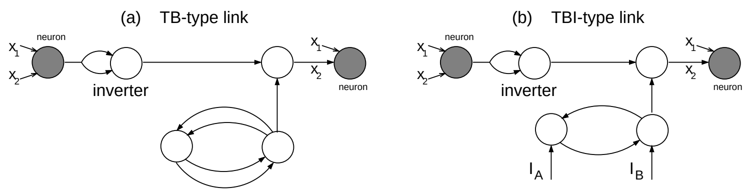

Figure 2. The TB-type (a) and TBI-type (b) link proposed by Teuscher. A simple node is used as an inverter in series with a normal introverted pair.

Modern neural networks are often organized in layers. For unorganized machines, the layered structure does not really exist since the machines are normally constructed at random and recurrent connections are allowed with no constraints.

3 Random Boolean Networks

Random boolean networks form a class of networks in which the links between nodes and the boolean functions are specified at random. They are often specified by two parameters: N, the number of nodes and K, the number of incoming links per node (sometimes, K indicates the average number of links). Synchronous RBN have been seriously investigated by Weisbuch [14] and Kauffman [8] and have been used as models for biological phenomena such as genetic regulatory networks. Turing's unorganized machines might in fact be considered as a very particular subset of synchronous RBN ($K = 2$, NAND functions only, special and fixed interconnections for B-, BI-, and TBI-types). A Turing unorganized machine can always be regarded as a RBN. The inverse statement is not true since it is very unlikely that one gets a Turing unorganized machine when randomly constructing a boolean network. Furthermore, Kauffman's RBN do not allow an external agent to configure the network directly.

Networks built of many interacting units (nodes, neurons) are used to study complex dynamical systems in may areas (e.g., gene regulation networks, neural networks, economics, etc.). Kauffman's studies have revealed surprisingly ordered structures in randomly constructed networks. In particular, the most highly organized behavior appeared to occur in networks where each node—like in Turing's unorganized networks—receives inputs from two other nodes ($K = 2$). It turned out that the networks exhibit three major regimes of behavior: *ordered* (solid), *complex* (liquid), and *chaotic* (gas). The most complex and interesting dynamics correspond to the liquid interface, the boundary between order and chaos. In the ordered regime, little computation can occur. In the chaotic phase, dynamics are too disordered to be useful. The most important and dominant results of Kauffman's numerical simulations can be summarized as follows [8]: (1) The expected median state cycle length is about $\sqrt{N}$ (where N=number of network nodes). (2) Most networks have short state cycles, while a few have very long ones. (3) The number of state cycle attractors is about $\sqrt{N}$. (4) The most interesting dynamics appear with an average connectivity of $K = 2$ (the boundary between order and chaos).

Very few work has been done around asynchronous random boolean networks (ARBN). Harvey and Bossomaier [6] have shown that they behave radically different from the deterministic synchronous version. For many physical and biological phenomena, the assumption of asynchrony seems more plausible.

4 Attractors

An *attractor* in a dynamical system is an *equilibrium state*. Each attractor is encompassed by a *bassin (domain) of attraction*. A deterministic complex dynamical system with a finite number of states ultimately "settles down" in an attractor after a finite time. If the state vector comes to rest completely, it is called a *fixed point*. If the state vector settles into a periodic motion, it is called a *limited cycle*. There exist also *chaotic* or *strange* attractors. As presented above, Kauffman has shown that the expected median state cycle length in RBN is about $\sqrt{N}$ and that the number of state cycle attractors is about $\sqrt{N}$. The question is whether these values are also valid for Turing's RBN. Figure 3 shows two typical plots that arise when drawing the attractor length and the number of attractors of a TBI-type network in function of the number of network nodes. For comparison, $\sqrt{N}$ is also plotted on the drawing. One can see that the number of attractors as well as the attractor length of the networks are about $\sqrt{N}$.

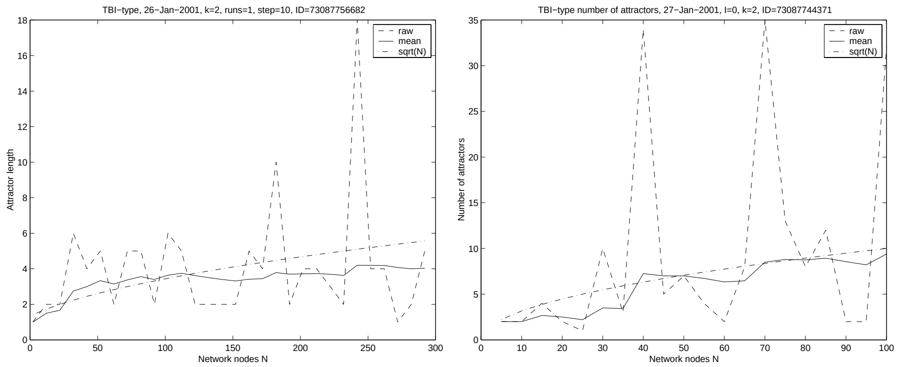

Figure 3. Attractor length (on the left) and number of attractors (on the right) of a TBI-type network. Both values are about $\sqrt{N}$.

5 Topological Evolution of TBI-type Networks

Topological evolution of dynamical networks has been studied in a recent paper by Bornholdt and Rohlf [4]. They evolved network topologies of an asymmetrically connected threshold network by a simple local rewiring rule: quiet nodes grow links, active nodes lose links. They have shown that this leads to an average connectivity of the networks towards the *critical* value $K_C = 2$ for large systems.

Kauffman postulated that gene regulatory networks may exhibit properties of complex dynamical networks near criticality [8], however, without providing an algorithm able to generate an topology near the critical point. Bornholdt and Rohlf asked "[...] whether connectivity may be driven towards a critical point by some dynamical mechanism" [4] and presented an approach that evolves the connectivity towards the critical point. In the remainder of this section we present a self-organizing topology evolving rule adapted for Turing's boolean networks that might also be applied to RBN.

As shown in section 2, each link between two nodes of a TBI-type network is equipped with a small A-type machine that functions as a switch (see figure 2). When the switch is opened, no information is passed, when it is closed, all information is passed and inverted. So far, all networks had an connectivity of exactly

$K = 2$ connections per node. Algorithm 1 presents one of several possible iterating rules that evolves the average connectivity of TBI-type networks towards the critical point. Instead of creating and removing interconnections, the algorithm only operates the switches incorporated in each link and thus *configures* the network topology. The connectivity $K(i)$ of a given node i is equivalent to the number of incoming links that have closed switches. At the end of a successful topology evolution, links with opened switches might simply be removed.

Algorithm 1 Topology evolution of TBI-type networks

```
Choose a fully connected network (K = N)
Choose a random switch configuration with an average connectivity K_ini
for t = 1 to T time steps do
  Choose a random initial state vector s(0)
  Find a dynamical attractor and determine its length L
  for l = 1 to L do
    Run the network and record the activity A(i) of each node
  end for
  for all nodes i do
    if A(i) = 0 then
      A switch on link l_ji (where node j is chosen a random) is enabled/closed.
    else
      A switch on link l_ji (where node j is chosen a random) is disabled/opened.
    end if
  end for
end for
```

The activity $A(i)$ of a node i is defined as the number of changes it makes in the time interval $T_2 - T_1$. When algorithm 1 is applied to a TBI-type network, a typical picture as shown in figure 4 (to the left) arises. The network self-organizes towards an average connectivity of about $K = 3.7$ ($N = 30$ nodes, 10^4 time steps), independent of the initial connectivity (switch configuration). For the simulations we ran, the average connectivity approximatively obeys the scaling law presented by Bornholdt and Rohlf [4]: $K_{ev}(N) - 2 = cN^{-\delta}$ where $c = 12.4$ and $\delta = 0.47$. Thus, when $N \rightarrow \infty$, the network evolves towards the critical connectivity $K_C = 2$. Figure 4 (to the right, $N = 20$ nodes, 10^3 time steps) shows the evolution of the average attractor length and the average number of attractors when algorithm 1 is applied to a network. It can be seen that both values essentially remain constant.

The above topology evolution algorithm (algorithm 1) presents a new and different type of mechanism compared to the phenomenon of self-organized criticality [3]. The networks self-organize towards criticality and "[...] exhibit considerable robustness against noise in the system" [4]. Bornholdt and Rohlf have shown that the mechanism is based on a topological phase transition in the networks.

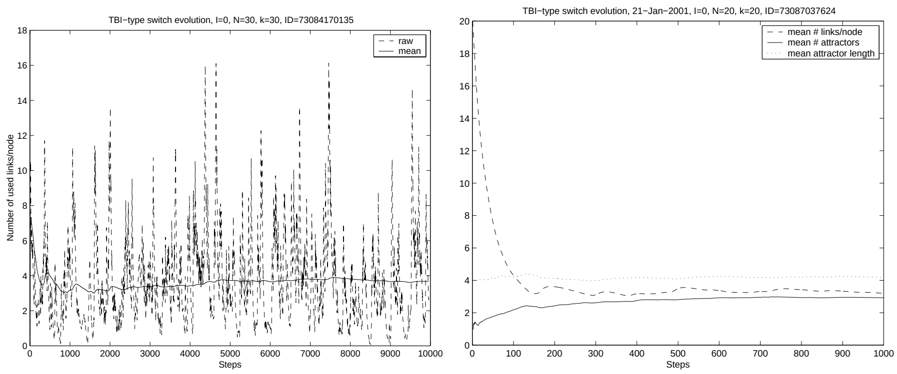

Figure 4. Evolution of the average connectivity of a TBI-type network (on the left, $N = 30$ nodes). Typical average attractor length and average number of attractors during TBI-type topology evolution (on the right, $N = 20$ nodes). Self-organization started at $K = 20$ (all connection switches closed).

6 Conclusion

We first presented Turing's early ideas on connectionism and his different types of machines. The TBI-type network is then introduced as a machine with universal computational capabilities. We briefly studied the characteristics of attractors and observed that the networks obey the characteristics of Kauffman's RBN. The number and the length of the attractors are about $\sqrt{N}$. We then applied a simple topology evolving algorithm to TBI-type networks and showed that the network self-organizes towards an average connectivity of about $K = 2$. For large systems ($N \to \infty$), the connectivity will evolve towards the critical value $K_C = 2$. It is of a particular interest that, although Turing TBI-type networks are quite different compared to the asymmetrically connected threshold networks presented in [4], both models evolve towards the same critical connectivity $K_C = 2$ for large systems. Already Kauffman experimentally showed that the most interesting dynamics of RBN appear with an average connectivity of $K = 2$ (the boundary between order and chaos!) [8], however, he did not self-organize the topology of his networks.

The crucial question is "[...] whether a comparable mechanism may occur in natural complex systems" [4]. Bornholdt and Rohlf further state "[...] that this form of global evolution of a network structure towards criticality might be found in natural complex systems". So far, there are still a lot of open questions. Kauffman's gene regulation networks are a good example of a biologically plausible model. It is still one of the greatest mysteries how the genetic code influences the development and growth of organisms. The synaptic changes in the brain are another example where the connectivity of a neural systems is regulated. Turing himself rather naively suggested that an A-type unorganized machine might be the simplest model of a nervous system with a random arrangement of neurons [13, p. 10]. However, today it is clear that his neural networks were by far too simple to model biological neurons, nevertheless, they offer many interesting properties of dynamical complex systems (see also [11]).

Future work will concentrate on the development of a completely local self-organizing rule that not only evolves the network towards an average critical connectivity but also optimizes the network to perform a given task.

References

1. J. T. Allanson. Some properties of randomly connected neural nets. In C. Cherry, editor, *Proceedings of the 3rd London Symposium on Information Theory*, pages 303–313, Butterworths, London, 1956.
2. S.-I. Amari. Characteristics of Randomly Connected Threshold-Element Networks and Network Systems. *Proceedings of the IEEE*, 59(1):35–47, January 1971.
3. P. Bak and L. Chen. Self-organized criticality. *Scientific American*, 265:26–33, January 1991.
4. S. Bornholdt and T. Rohlf. Topological evolution of dynamical networks: Global criticality from local dynamics. *Physical Review Letters*, 84(26):6114–6117, June 2000.
5. B. J. Copeland and D. Proudfoot. On Alan Turing's anticipation of connectionism. *Synthese: An International Journal for Epistemology, Methodology and Philosophy of Science*, 108:361–377, 1996. Kluwer Academic Publishers.
6. I. Harvey and T. Bossomaier. Time out of joint: Attractors in asynchronous random boolean networks. In P. Husbands and I. Harvey, editors, *Proceedings of the Fourth European Conference on Articial Life*, pages 67–75. MIT Press, Cambridge, MA, 1997.
7. D. C. Ince, editor. *Mechanical Intelligence: Collected Works of A. M. Turing*, chapter Intelligent Machinery, pages 107–128. North-Holland, 1992.
8. S. A. Kauffman. *The Origins of Order: Self–Organization and Selection in Evolution*. Oxford University Press, New York, Oxford, 1993.
9. W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas immanent in neural nets. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
10. L. I. Rozonoér. Random logical nets I. *Automation and Remote Control*, 5:773–781, 1969. Translation of Avtomatika i Telemekhanika.
11. C. Teuscher. Study, Implementation, and Evolution of the Artificial Neural Networks Proposed by Alan M. Turing. A Revival of his "Schoolboy" Ideas. Master's thesis, Swiss Federal Institute of Technology Lausanne, Logic Systems Laboratory, EPFL-DI-LSL, CH-1015 Lausanne, February 2000.
12. C. Teuscher and E. Sanchez. A Revival of Turing's Forgotten Connectionist Ideas: Exploring Unorganized Machines. In *Proceedings of the 6th Neural Computation and Psychology Workshop, NCPW6*, University of Liège, Belgium, September 16–18 2000. Springer-Verlag. (To appear).
13. A. M. Turing. Intelligent Machinery. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 5 of *National Physical Laboratory Report*, pages 3–23. Edinburgh University Press, Edinburgh, 1969.
14. G. Weisbuch. *Complex Systems Dynamics: An Introduction to Automata Networks*, volume 2 of *Lecture Notes, Santa Fe Institute, Studies in the Sciences of Complexity*. Addison-Wesley, Redwood City, CA, 1991.