Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Automata complete computation with Hodgkin–Huxley neural networks composed of synfire rings[☆]

Jérémie Cabessa^{a,b,*,1}, Aubin Tchaptchet^{c,1}

^a Laboratory of Mathematical Economics and Applied Microeconomics (LEMMA), Université Paris 2, Panthéon-Assas, 75005 Paris, France ^b Institute of Computer Science of the Czech Academy of Sciences, P. O. Box 5, 18207 Prague 8, Czech Republic

^c Institute of Physiology, Philipps University of Marburg, 35037 Marburg, Germany

ARTICLE INFO

Article history: Received 21 July 2019 Revised and accepted 23 March 2020 Available online 28 March 2020

Keywords: Neural computation Recurrent neural networks Cell assemblies Synfire rings Hodgkin–Huxley equations Finite state automata

ABSTRACT

Synfire rings are neural circuits capable of conveying synchronous, temporally precise and selfsustained activities in a robust manner. We propose a cell assembly based paradigm for abstract neural computation centered on the concept of synfire rings. More precisely, we empirically show that Hodgkin–Huxley neural networks modularly composed of synfire rings are automata complete. We provide an algorithmic construction which, starting from any given finite state automaton, builds a corresponding Hodgkin–Huxley neural network modularly composed of synfire rings and capable of simulating it. We illustrate the correctness of the construction on two specific examples. We further analyze the stability and robustness of the construction as a function of changes in the ring topologies as well as with respect to cell death and synaptic failure mechanisms, respectively. These results establish the possibility of achieving abstract computation with bio-inspired neural networks. They might constitute a theoretical ground for the realization of biological neural computers.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

The computational aspects of neural information processing is an issue of central importance. From a theoretical perspective, it has been shown that various models of abstract computation could be simulated by different kinds neural networks. In particular, Boolean recurrent neural networks are computationally equivalent to finite state automata (Kleene, 1956; McCulloch & Pitts, 1943; Minsky, 1967). Sigmoidal neural networks with rational synaptic weights are Turing-complete (Hartley & Szu, 1987; Hyötyniemi, 1996; Kilian & Siegelmann, 1996; Neto, Siegelmann, Costa, & Araujo, 1997; Pollack, 1987; Siegelmann & Sontag, 1995; Turing, 1948). And notably, sigmoidal neural networks with either real or evolving synaptic weights are super-Turing (Balcázar, Gavaldà, & Siegelmann, 1997; Cabessa & Siegelmann, 2011, 2014; Siegelmann, 2003; Siegelmann & Sontag, 1994). These studies have been generalized to alternative paradigms of computation (Cabessa & Duparc, 2016; Cabessa & Finkel, 2019; Cabessa

E-mail addresses: jeremie.cabessa@u-paris2.fr (J. Cabessa), aubin.tchaptchet@gmail.com (A. Tchaptchet).

¹ Both authors contributed equally to the work.

https://doi.org/10.1016/j.neunet.2020.03.019 0893-6080/© 2020 Elsevier Ltd. All rights reserved. & Siegelmann, 2012; Cabessa & Villa, 2012, 2014, 2015, 2016). The computational power of spiking (instead of sigmoidal) neural networks has also been extensively studied (Maass, 1999; Maass & Bishop, 1999). More recently, the study of P systems—parallel abstract models of computation inspired from the membrane structure of biological cells—, and in particular, of spiking neural P systems, has become a highly active field of research (G., 2000; Neary, 2015; Păun, 2002).

In biology, the computational capabilities of the cortex are rather investigated from the perspective of *cell assembly theory* (Braitenberg, 1978; Hebb, 1949; Palm, 1982; Palm, Knoblauch, Hauser, & Schüz, 2014). This approach understands and models the brain in terms of distributed neuronal activity. The general assumptions are that entities of the outside world as well as internal states are encoded into groups of neurons – the *neural assemblies* – rather than into single "grandmother" cells. In addition, the assemblies are involved in the implementation of two kinds of associative memories: an auto-associative memory, which corresponds to the storage and stabilization of local activity patterns; and a hetero-associative memory, which refers to long range interconnections between local assemblies. The assemblies would be generated via Hebbian coincidence or correlation learning mechanisms. The present study fits within this research direction.

In this context, the implementation of finite state automata has been achieved in neurobiologically inspired networks that are modularly composed of Hebbian cell assemblies (Fay, Kaufmann, Knoblauch, Markert, & Palm, 2005; Garagnani, Wennekers,







 $[\]stackrel{\mbox{\tiny theta}}{\rightarrow}$ This article is an extended version of the proceeding paper Cabessa and Tchaptchet (2018).

^{*} Corresponding author at: Laboratory of Mathematical Economics and Applied Microeconomics (LEMMA), Université Paris 2, Panthéon-Assas, 75005 Paris, France.

& Pulvermüller, 2009; Markert, Knoblauch, & Palm, 2005, 2007; Wennekers, 2006, 2007, 2009; Wennekers, Garagnani, & Pulvermüller, 2006; Wennekers & Palm, 2009) (the relationship between this and our approaches is discussed in Section 7). Besides, the implementations of associative memory tasks, logical gates, or abstract devices have been achieved on diverse types of oscillator networks (Hoppensteadt & Izhikevich, 2000; Malagarriga et al., 2015; Xu, Principe, & Harris, 2004; Zanin, De. Pozo, & Boccaletti, 2011).

In neural computation, several types of neural circuits have been argued to be crucially involved in the processing and coding of information. Amongst these, synfire chains are of particular interest. Synfire chains are feedforward neural circuits whose every layer is connected to the next by means of excitatory convergent/divergent synaptic patterns (Abeles, 1991, 2004; Diesmann, Gewaltig, & Aertsen, 1999; Ikegaya et al., 2004; Mainen & Seinowski, 1995). According to this architecture, the neurons of each layer tend to fire simultaneously, and the firing activity can propagate throughout the successive layers in a synchronized manner. In this way, synfire chains are capable of conveying repeated complex spatio-temporal patterns of discharges in a robust and highly temporally precise way. Moreover, synfire rings consist of looping synfire chains (Horn, Levy, Meilijson, & Ruppin, 1999; Levy, Horn, Meilijson, & Ruppin, 2001; Zheng & Triesch, 2014). As an additional dynamical feature, the ring shape naturally gives rise to self-sustained activities, which correspond to attractor dynamics.

Synfire chains and rings have been shown to spontaneously emerge in neural networks subjected to various kinds of synaptic plasticity. In networks of integrate-and-fire neurons, the consideration of a rate-based Hebbian learning rule together with repeated input stimulations gives rise to small synfire chains (Hertz & Prügel-Bennett, 1996). When provided with a synaptic scaling learning rule, the networks also unveil the organization of synfire-like patterns of activity (Buonomano, 2005). The incorporation of a spike-timing dependent synaptic plasticity (STDP) rule leads to the formation of sub-assemblies of cells firing in cyclic manner – synfire ring-like structures – referred to as distributed synchrony (DS) cycles (Horn et al., 1999; Levy et al., 2001). Notably, the consideration of STDP together with axon remodeling yields the development of long synfire chains (Jun & Jin, 2007). In more physiologically realistic neural networks composed of Hodgkin-Huxley cells, STDP also entails the self-organization of chain of cell assemblies dependent on the plasticity timing windows (Kitano, Câteau, & Fukai, 2002). In pacemaker-triggered oscillator networks, STDP with asymmetric learning windows induces formation of feedforward circuits - synfire ring-like structures starting from the pacemaker (Masuda & Kori, 2007). Furthermore, an abundance of synfire rings has been shown to emerge in self-organizing neural networks subjected to STDP, structural plasticity as well as homeostatic forms of plasticity (Zheng & Triesch, 2014). For a thorough review concerning the relationship between weight dynamics and emergence of various neuronal structures, see Gilson, Burkitt, and Van Hemmen (2010) (and the references therein). On the other hand, it has also been shown that locally connected random networks do not naturally sustain stable propagation of synfire activity, but rather lead to "synfire explosions", unless specific parametric regimes are considered (Mehring, Hehl, Kubo, Diesmann, & Aertsen, 2003).

Here, we propose a cell assembly based paradigm for abstract neural computation, centered on the concept of synfire rings. We consider a modified version of the highly accurate Hodgkin–Huxley model for capturing the dynamics of individual cells (Hodgkin & Huxley, 1952). We empirically show that Hodgkin–Huxley (HH) recurrent neural networks modularly composed of synfire rings are automata complete. More precisely,

we provide an algorithmic construction which, starting from any finite state automaton, builds a corresponding Hodgkin-Huxley neural network modularly composed of synfire rings and capable of simulating it. We illustrate the correctness of the construction on two specific examples. We further analyze the stability and robustness of the construction as a function of changes in the ring topologies as well as with respect to cell death and synaptic failure mechanisms, respectively. These results generalize those of Cabessa and Masulli (2017) and Cabessa, Horcholle-Bossavit, and Quenet (2017) to the more biological context of Hodgkin-Huxley neural networks. In this framework, the inhibitory system ensuring the transition between the rings can be significantly simplified.² These results further extend those of Cabessa and Tchaptchet (2018) by providing a rigorous formulation of the algorithmic construction, as well as a thorough analysis of the stability and robustness of the simulation process.

The proposed neuro-computational paradigm finds its relevance at many levels: (i) The successive computational states are encoded into temporally robust cyclic attractor dynamics, induced by the self-sustained activities of the synfire rings. (ii) The transitions between such attractors are perfectly controlled, in an input-driven way. (iii) The global computational process is robust to various kinds of architectural failures and synaptic noises. (iv) The proposed model fits within the general theory of cell assemblies. These considerations establish the possibility of achieving abstract computation with bio-inspired neural networks. They support the idea that biological neural networks are (at least) automata complete. They might also constitute a theoretical ground for the realization of biological neural computers.

2. Boolean recurrent neural networks and finite state automata

A Boolean recurrent neural network (BRNN) \mathcal{N} is a synchronous network of Boolean neurons, i.e., McCulloch and Pitts cells (Mc-Culloch & Pitts, 1943), related together in a general architecture. It is composed of M Boolean input neurons $(u_i)_{i=1}^M$ and N Boolean internal neurons $(x_i)_{j=1}^N$, among which P are considered to be the output neurons $(x_i)_{k=1}^N$ (if P = 0, then no output is considered). At each time step, the activation value of every cell is either firing (equal to 1) or quiet (equal to 0). The dynamics of network \mathcal{N} is computed as follows: given the activation values of the input neurons $(u_j)_{j=1}^M$ and the internal neurons $(x_j)_{i=1}^N$ at time t, the activation values of the internal neurons $(x_i)_{i=1}^N$ at time t + 1 are given by the following equations:

$$\begin{aligned} x_i(t+1) &= \theta \left(\sum_{j=1}^N a_{ij} \cdot x_j(t) + \sum_{j=1}^M b_{ij} \cdot u_j(t) + c_i \right), \\ \text{for } i &= 1, \dots, N \end{aligned}$$
(1)

where a_{ij} and b_{ij} are the weights of the synaptic connections from x_j to x_i and from u_j to x_i , respectively, c_i is the bias of cell x_i , and θ is the classical hard-threshold activation function defined by

$$\theta(x) = \begin{cases} 0 & \text{if } x < 1 \\ 1 & \text{if } x \ge 1. \end{cases}$$

Two Boolean recurrent neural networks are illustrated in Figures 3 and 4.

The *input state*, *internal state* and *output state* of network N at time *t* are the Boolean vectors

$$\mathbf{u}(t) = (u_1(t), \ldots, u_M(t))^I \in \mathbb{B}^M$$

² The so-called "triangular structures" of Cabessa et al. (2017), Cabessa and Masulli (2017), which were the Achille's heel of the construction's robustness, are no more needed here.



Fig. 1. A finite state automaton. Finite state automaton recognizing the language $\Sigma^* 0110 \Sigma^*$, i.e., all sequences of bits containing the pattern 0110. The nodes and edges of the graph represent the computational states and transitions of the automaton, respectively. Nodes q_0 and q_4 are the initial and final states, respectively. A transition from q_i to q_j labeled by u means that if the automaton is in state q_i and receives input symbol u, then it will move to state q_i .

$$\mathbf{x}(t) = (x_1(t), \dots, x_N(t))^T \in \mathbb{B}^N$$
$$\mathbf{o}(t) = (x_{i_1}(t), \dots, x_{i_P}(t))^T \in \mathbb{B}^P.$$

For any input stream $u = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2)\cdots$, the *computation* and *output* of \mathcal{N} working over input u are the sequences of internal states and output states at successive time steps

$$\mathcal{N}(u) = \mathbf{x}(0)\mathbf{x}(1)\mathbf{x}(2)\dots$$

 $o(u) = \mathbf{0}(0)\mathbf{0}(1)\mathbf{0}(2)\dots$

where $\mathbf{x}(0) = \mathbf{0}$ and the components of $\mathbf{x}(t)$ and $\mathbf{o}(t)$ are given by Eq. (1), for each t > 0.

Finite state automata constitute an abstract model of computation working as language recognizers. Formally, a *deterministic finite state automaton (DFSA)* is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where:

- Q is a finite set of *states*;
- Σ is an *alphabet* for the input symbols;
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function;
- $q_0 \in Q$ is the initial state;
- $F \subseteq Q$ is the set of final states.

A finite state automaton is generally represented as a directed graph, as illustrated in Fig. 1. The nodes and edges of the graph represent the states and transitions of the automaton. Each transition $\delta(q, a) = q'$ signifies that if the automaton is in state $q \in Q$ and receives input symbol $a \in \Sigma$, then it will move to state $q' \in Q$. For any input (stream) $w = a_0 a_1 \cdots a_n \in \Sigma^*$, the *computation* of \mathcal{A} over w is the sequence

$$\mathcal{A}(w) = ((q_{i_0}, a_0, q_{i_1}), (q_{i_1}, a_1, q_{i_2}), \dots, (q_{i_n}, a_n, q_{i_{n+1}}))$$

such that $q_{i_0} = q_0$ and $\delta(q_{i_k}, a_k) = q_{i_{k+1}}$, for all k = 0, ..., n. Such a computation is usually denoted as

$$\mathcal{A}(w): q_0 \xrightarrow{a_0} q_{i_1} \xrightarrow{a_1} q_{i_2} \cdots q_{i_n} \xrightarrow{a_n} q_{i_{n+1}}.$$

Input *w* is said to be *accepted* (resp. *rejected*) by automaton A if the last state $q_{i_{n+1}}$ of computation A(w) belongs (resp. does not belong) to the set of final states *F*. The set of all inputs accepted by A is the *language* recognized by A, denoted as L(A).

Finite state transducers are slightly modified versions of finite state automata working as output generators instead of language recognizers. Formally, a *deterministic finite state transducer (DFST)* is a tuple $T = (Q, \Sigma, \delta, q_0)$, where:

- Q is a finite set of *states*;
- Σ is an *alphabet* of the input and output symbols;
- $\delta : Q \times \Sigma \rightarrow Q \times \Sigma$ is the transition function;
- $q_0 \in Q$ is the *initial state*.

A finite state transducer is also generally represented as a directed graph, as illustrated in Fig. 2. Each transition $\delta(q, a) = (q', o)$ signifies that if the transducer is in state $q \in Q$ and receives input symbol $a \in \Sigma$, then it will move to state $q' \in Q$ and



Fig. 2. A finite state transducer. A finite state transducer implementing a serial binary adder. The nodes and edges of the graph represent the computational states and transitions of the transducer, respectively. A transition from q_i to q_j labeled by u/o means that if the transducer is in state q_i and receives input symbol u, then it will move to state q_j and output symbol o. The initial state is q_0 . The transducer computes the sum of two binary numbers as follows: it starts from initial state q_0 and takes as inputs the successive pairs of bits of the sum in the reverse order; if needed, it adds a last input $\binom{0}{0}$ so as to come back to its initial state q_0 . The successive output bits correspond to the result of the sum in the reverse order.

output symbol $o \in \Sigma$. For any input $w = a_0 a_1 \cdots a_n \in \Sigma^*$, the *computation* of \mathcal{T} over w is usually denoted as

$$\mathcal{T}(w): q_{i_0} \xrightarrow{a_0/o_0} q_{i_1} \xrightarrow{a_1/o_1} q_{i_2} \cdots q_{i_n} \xrightarrow{a_n/o_n} q_{i_{n+1}}$$

where $q_{i_0} = q_0$ and $q_{i_k}, a_k, o_k, q_{i_{k+1}}$ are such that $\delta(q_{i_k}, a_k) = (q_{i_{k+1}}, o_k)$, for all k = 0, ..., n. The *output* generated by \mathcal{T} working on input w is the sequence

 $o(w) = o_0 o_1 \cdots o_n.$

Boolean recurrent neural networks are known to be computationally equivalent to finite state automata (Kleene, 1956; McCulloch & Pitts, 1943; Minsky, 1967).

Theorem 1 (*Minsky 1967*). Any Boolean neural network can be simulated by some finite state automaton, and any finite state automaton or transducer can be simulated by some Boolean network.

The first part of this statement is straightforward. A Boolean network with N cells has at most 2^N spiking configurations. It can therefore be simulated by a finite state automaton containing at most 2^N states. In short, the states of the automaton are the spiking configurations (or states) of the network, and the edges of the automaton are the network's transitions between these configurations. In this way, the dynamics of the network corresponds precisely to the paths in the graph of the automaton.

The second part of the statement is more relevant, since it concerns the issue of the implementation of finite state machines on parallel hardwares (see for instance (Alon, Dewdney, & Ott, 1991; Elman, 1990; Horne & Hush, 1996; Indyk, 1995; Omlin & Giles, 1996; Siegelmann, 1996)). A algorithmic construction taking a given automaton or transducer as input and providing a recurrent neural network that simulates it as output is described in Algorithm 1 (adapted from Minsky, 1967).

In order to illustrate this construction, we consider the automaton and the transducer of Figures 1 and 2, respectively. The automaton is a pattern detector, recognizing the language $\Sigma^*0110\Sigma^*$, i.e., all sequences of bits containing the pattern 0110. The accepting computation of this automaton working over input 00101100 is illustrated in Table 1 (rows 2–3).

A Boolean neural network simulating this automaton is illustrated in Fig. 3. This network is obtained on the basis on Algorithm 1 ((Minsky, 1967)'s construction) which is not optimal in terms of number of cells and connections,³. The network is designed in such a way that, at each time step, at most one internal

 $^{^3}$ For optimality issues, see Horne and Hush (1996), Indyk (1995) as well as the discussion in Section 7.

Algorithm 1 Procedure which starts from a given automaton (transducer) and builds a E	Boolean recurrent neural network that simulates it.
Require: DFSA $\mathcal{A} = (Q, \Sigma, \delta_{\mathcal{A}}, q_0, F)$ (resp. DFST $\mathcal{T} = (Q, \Sigma, \delta_{\mathcal{T}}, q_0)$)	
1:	// ***cells***
2: set K Boolean input cells $(u_a)_{a \in \Sigma}$, where $K = \Sigma $	// input cells
3: set $K \times N$ Boolean internal cells $(C_{q,a})_{q \in Q, a \in \Sigma}$, where $N = Q $	// internal cells
4: set K Boolean output cells $(C_{out,a})_{a \in \Sigma}$	<pre>// output cells - transducer's case</pre>
5:	//***connections***
6: for all transition $(q, a, q') \in graph(\delta_{\mathcal{A}})$ (resp. $(q, a, q', o) \in graph(\delta_{\mathcal{T}})$) do	
7: add an input connections from u_a to $C_{q,a}$ of weight $\frac{1}{2}$	// input connections
8: for all input symbol $a' \in \Sigma$ do	
9: add a connection from $C_{a,a}$ to $C_{a',a'}$ of weight $\frac{1}{2}$	// internal connections
10: end for	
11: add a connection from $C_{q,a}$ to $C_{out,o}$ of weight 1	<pre>// output connections - transducer's case</pre>
12: end for	-

cell is spiking. More precisely, if at time step t, the network has its cell $C_{q,a}$ being spiking while receiving the encoding of input a', then at time t + 1, cell $C_{q',a'}$ will be spiking, where q' is given by the automaton's transition $\delta(q, a) = q'$. The computation of this network over the encoding of input 00101100 is illustrated in Table 1 (rows 4–7). Thanks to Algorithm 1, the indices of the successive spiking cells of the network ($C_{q_0,0}$, $C_{q_1,0}$, $C_{q_1,1}$, $C_{q_2,0}$, $C_{q_1,1}$, $C_{q_2,1}$, $C_{q_3,0}$, $C_{q_4,0}$) correspond precisely to the successive states visited by the automaton (q_0 , q_1 , q_1 , q_2 , q_1 , q_2 , q_3 , q_4). In this precise sense, the automaton is correctly simulated by the network.

The transducer of Fig. 2 implements a serial binary adder. It is composed of two states, corresponding to the two situations of either being currently carrying a 1 in the addition process or not. The computation of this transducer over the binary sum s

	1	1 ¹	1 ¹	1	0 ¹	0 ¹	1
+		1	0	1	0	1	1
	1	1	0	0	1	0	0

is illustrated in Table 1 (rows 8–10, see legend of Fig. 2 for a description of the transducer's computation).

The Boolean neural network simulating this transducer obtained via Algorithm 1 is illustrated in Fig. 4. The network dynamics satisfies the following property: if at time step *t*, the network has its cell $C_{q,a}$ being spiking while receiving the encoding of input *a'*, then at time t + 1, cells $C_{q',a'}$ and $C_{out,o'}$ will be spiking, where *q'* and *o'* are given by the transducer's transition $\delta(q, a) =$ (o', q'). The computation of this Boolean network over the encoding of the binary sum *s* is illustrated in Table 1 (rows 11–17). Note that the indices of the successive spiking cells of the network $(C_{q_0, (\frac{1}{1})}, C_{q_1, (\frac{0}{1})}, C_{q_0, (\frac{1}{1})}, C_{q_1, (\frac{1}{0})}, C_{q_1, (\frac{0}{0})})$ correspond precisely to the successive states visited by the transducer $(q_0,$ $q_1, q_1, q_0, q_1, q_1, q_1)$. Moreover, the successive spiking output cells of the network ($C_{out,0}, C_{out,0}, C_{out,1}, C_{out,0}, C_{out,1}, C_{out,1})$ correspond precisely to the successive outputs the transducer (0, 0, 1, 0, 0, 1, 1). In this very sense, the transducer is correctly simulated by the network.

These considerations show that the computations of the automaton and transducer of Figures 1 and 2 are perfectly reflected by the dynamics of their corresponding Boolean neural networks of Figures 3 and 4, respectively, with a time delay of 1 or 2 time steps.⁴. More precisely, the automaton (resp. the transducer) is in state q and receives input a at time t if and only if the neural network has its internal cell $C_{q,a}$ spiking at time t + 1 (resp. and its output cell $C_{out,o}$ spiking at time t + 2). These features can be verified in Table 1.

Finally, note that the above construction is *generic*: it can be applied to any finite state automaton or transducer. The formal

proof of the correctness of this simulation process goes back to Minsky (1967).

3. Modified Hodgkin-Huxley model

The pioneering Hodgkin–Huxley⁵ (HH) model is considered amongst the most accurate for the simulation of biological neurons (Hodgkin & Huxley, 1952). The parameters of the original conductance-based HH-equations are highly precise for the modeling of action potentials. But they remain difficult to be adjusted when simulating experimental data.

Here, we consider an improved HH-model which is closer the biological reality (cf. System (2)) (Tchaptchet et al., 2013). The rate constants of the original HH-equations are replaced by Boltzmann functions fitted with biological parameters. Moreover, the power functions of the original equations are discarded. This model is easier to handle, since the equations are simplified and all parameters have biological meanings. The equations of this improved model have been used successfully in various contexts, showing the relations to the original HH-model (Postnova, Finke, Huber, Voigt, & Braun, 2012; Tchaptchet, 2018, 2019; Tchaptchet et al., 2013).

In this biological context, each synapse is associated with a corresponding synaptic current. The synaptic currents are modeled by the so-called alpha function $\alpha(t) = a \cdot t \cdot e^{-b \cdot t}$, where a and b are different parameters depending on the kinds of connections that we consider (cf. Table 2). In the sequel, the maximal amplitudes of these synaptic currents will be referred to as the synaptic weights.

For any given neuron, let *V* be its membrane potential and *C* its membrane capacitance. Let also I_L be a leakage current, I_{Na} and I_K be the sodium and the potassium fast currents responsible for spike generation, respectively, I_C be the set of synaptic currents coming from the neighboring neurons, and I_{input} be a pulse-like input current. The following system of ordinary differential equations (ODE) characterizing the membrane potential *V* as a function of the currents I_L , I_{Na} , I_K , I_C and I_{input} constitutes the modified version of the Hodgkin–Huxley model considered here:

$$C \cdot \frac{dV}{dt} = -I_L - I_{Na} - I_K - I_C - I_{input}$$

$$I_L = g_L \cdot (V - V_L)$$

$$I_{Na} = g_{Na} \cdot m \cdot h \cdot (V - V_{Na}) \text{ where}$$

$$\frac{dm}{dt} = \frac{m_{\infty} - m}{\tau_m} \quad m_{\infty} = \frac{1}{1 + e^{-s_m \cdot (V - V_{hm})}}$$

$$\frac{dh}{dt} = \frac{h_{\infty} - h}{\tau_h} \quad h_{\infty} = 1 - \frac{1}{1 + e^{-s_h \cdot (V - V_{hh})}}$$

 $^{^{4}}$ A formal definition of the concept of *simulation* of an automaton or transducer by a corresponding neural network is provided in Section 4.3

 $^{^5}$ Alan Llyod Hodgkin and Andrew Fielding Huxley were awarded of the Medicine's Nobel Prize in 1963 for this model.



Fig. 3. Boolean recurrent neural network computationally equivalent to the finite automaton of Fig. 1. The network has 2 input cells, u_0 and u_1 (blue), used to encode the 2 possible automaton input symbols 0 and 1, respectively. The "start" cell spikes only at time t = 0 in order to initiate the dynamics. The 10 internal cells $C_{q,a}$ (black) represent the 10 possible events of the automaton, namely, "being is state q and receiving input a", for all $q \in Q = \{q_0, q_1, q_2, q_3, q_4\}$ and $a \in \Sigma = \{0, 1\}$. The orange and black synaptic connections have weights 1/2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 4. Boolean recurrent neural network computationally equivalent to the finite transducer of Fig. 2. The network has 4 input cells $u_{\binom{0}{6}}$, $u_{\binom{1}{1}}$, $u_{\binom{1}{6}}$, $u_{\binom{1}{1}}$, $u_{$

$$I_{K} = g_{K} \cdot n \cdot (V - V_{K})$$

$$\frac{dn}{dt} = \frac{n_{\infty} - n}{\tau_{n}} \qquad n_{\infty} = \frac{1}{1 + e^{-s_{n} \cdot (V - V_{hn})}}$$

$$I_{C} = w_{intra}^{exc} + w_{inter}^{exc} + w_{inter}^{inh} + w_{output}^{exc} + w_{output}^{inh}$$

$$w_{intra}^{exc} = a_{intra}^{exc} \cdot t \cdot e^{b_{intra}^{exc} \cdot t}$$

$$w_{inter}^{exc} = a_{inter}^{exc} \cdot t \cdot e^{b_{inter}^{exc} \cdot t}$$

$$w_{inter}^{exc} = a_{inter}^{inh} \cdot t \cdot e^{b_{inter}^{int} \cdot t}$$

$$w_{output}^{exc} = a_{output}^{exc} \cdot t \cdot e^{b_{output}^{inter} \cdot t}$$

$$w_{output}^{exc} = a_{output}^{exc} \cdot t \cdot e^{b_{output}^{inter} \cdot t}$$

$$w_{output}^{exc} = a_{output}^{exc} \cdot t \cdot e^{b_{output}^{inter} \cdot t}$$

$$I_{input} = a_{input}^{enh} \cdot \chi_{t_{input}}(t)$$

$$(2)$$

In the last equation, $\chi_{t_{input}}(t) = 1$ if $t \in [0, t_{input}]$ and $\chi_{t_{input}}(t) = 0$ otherwise, meaning that the input current I_{input} has a constant amplitude of a_{input}^{exc} and a duration of t_{input} ms.

A detailed explanation of the parameters of Eqs. (2) can be found in Postnova et al. (2012), Tchaptchet (2019), Tchaptchet et al. (2013). Throughout this work (Sections 4–6), the considered cells are the same as the "standard neuron" of the virtual laboratory "SimNeuron" (http://www.virtual-physiology.com/). They are all identical in the so-called steady-state mode. Their corresponding parameters are given in Table 2. In Section 6, variations of the parameters a_j^i and b_j^i characterizing the synaptic currents will be considered and specified explicitly (where $i \in \{exc, inh\}$ and $j \in \{intra, inter\}$).

In this work, the differential Equations (2) are implemented using the classical Euler's discretization method with a time step of 0.01. Simulations are in milliseconds, meaning that each millisecond of simulation is discretized in 100 time steps. The simulations are performed in C# for Sections 4 and 5, and in Python for Section 6.

Table 1

Row 1. Time steps. **Rows 2–3.** Computation of the automaton of Fig. 1 over input 00101100. **Rows 4–7.** Dynamics of the Boolean neural network of Fig. 3 over the encoding of the same input. Note that the indices *s* of the successive activated cells $C_{s,i}$ of the network match the successive states *s* of the automaton. **Rows 8–10.** Computation of the transducer of Fig. 2 over the input sum +101001. **Rows 11–17.** Dynamics of the Boolean neural network of Fig. 4 over the encoding of the same input. The indices *s* and *j* of the successive activated cells $C_{s,i}$ of the network match the successive states *s* and outputs *j* of the transducer.

-		-								
Time steps	0	1	2	3	4	5	6	7	8	9
Inputs	0	0	1	0	1	1	0	0	_	-
				◀ · · · · · ·	· pattern 0110	••••••				
States	q_0	q_1	q_1	<i>q</i> ₂	q_1	<i>q</i> ₂	<i>q</i> ₃	q_4 detected	q_4	-
Cell start	1	0	0	0	0	0	0	0	0	0
Cell u ₀	1	1	0	1	0	0	1	1	0	0
Cell u ₁	0	0	1	0	1	1	0	0	0	0
				◄ ·· encod	ing of the patte	ern 0110 ·· 🕨				
Cells $C_{s,i}$	-	$C_{q_0,0}$	$C_{q_{1},0}$	$C_{q_{1},1}$	$C_{q_2,0}$	$C_{q_{1},1}$	$C_{q_2,1}$	$C_{q_{3},0}$	$C_{q_4,0}$ detected	-
Inputs	$\binom{1}{1}$	$\binom{0}{1}$	$\begin{pmatrix} 0\\ 0 \end{pmatrix}$	$\binom{1}{1}$	$\begin{pmatrix} 1\\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0\\ 0 \end{pmatrix}$	_	-	-
States	q_0	q_1	q_1	q_0	q_1	q_1	q_1	q_0	-	-
Outputs	0	0	1	0	0	1	1	-	-	-
		· · · · · · resul	t of the sum s	in the reverse o	order · · · · · · · ·	•••••				
Cell start	1	0	0	0	0	0	0	0	0	0
Cell $u_{(0)}$	0	0	1	0	0	0	1	0	0	0
Cell $u_{(1)}^{(0)}$	0	1	0	0	0	0	0	0	0	0
Cell $u_{(1)}^{(1)}$	0	0	0	0	1	0	0	0	0	0
Cell $u_{(1)}$	1	0	0	1	0	1	0	0	0	0
Cells $C_{s,i}$	-	$C_{q_0, \binom{1}{1}}$	$C_{q_1, \binom{0}{1}}$	$C_{q_1, \begin{pmatrix} 0 \\ 0 \end{pmatrix}}$	$C_{q_0, \binom{1}{1}}$	$C_{q_1, \binom{1}{0}}$	$C_{q_1, \binom{1}{1}}$	$C_{q_1, \begin{pmatrix} 0 \\ 0 \end{pmatrix}}$	-	-
Cells Cout,j	-	-	C _{out,0}	C _{out,0}	Cout, 1	Cout,0	C _{out,0}	Cout, 1	$C_{out,1}$	-
			∢ · · · · · e	ncoding of the	result of the su	m s in the reve	rse order · · · · ·	• ►		

Table 2

Values of the parameters of Eqs. (2). These parameters are those of a "standard neuron" of the virtual laboratory "SimNeuron" (www.virtual-physiology.com).

1 5 65 7			
Membrane capacitance [pF]	C = 0.1		
Currents: non-synaptic	Leakage	Sodium	Potassium
Max. conductances $[\mu S]$	$g_L = 0.1$	$g_{Na} = 4.0$	$g_{K} = 2.0$
Potentials [mV]	$V_{L} = -60$	$V_{Na} = 50$	$V_K = -90$
Half potentials [mV]		$V_{hm}=-22 \qquad V_{hh}=-50$	$V_{hn}=-30$
Slope of		$s_m = 0.14 \qquad s_h = 0.12$	$s_n = 0.14$
Time constants [ms]		$\tau_m = 0.05 \qquad \tau_h = 1.5$	$\tau_n = 1.8$
Currents: synaptic		Excitatory	Inhibitory
Input		$a_{input}^{exc} = 1.9$ $t_{input} = 0.4$	
Intra-ring		$a_{intra}^{exc} = 25.0$ $b_{intra}^{exc} = 2.0$	
Inter-ring		$a_{inter}^{exc} = 3.0$ $b_{inter}^{exc} = 0.7$	$a_{inter}^{inh} = 15.0$ $b_{inter}^{inh} = 1.5$
Output		$a_{output}^{exc} = 6.0$ $b_{output}^{exc} = 0.7$	$a_{output}^{inh} = 15.0$ $b_{output}^{inh} = 1.5$

4. Finite state automata and Hodgkin–Huxley neural networks composed of synfire rings

We now generalize the equivalence between automata and recurrent neural networks presented in Section 2 to the biological context of Hodgkin–Huxley (HH) recurrent neural networks composed of synfire rings. More precisely, we show that any finite state automaton or transducer can be correctly simulated by a recurrent neural network modularly composed of synfire rings, and whose cells' dynamics is governed by the modified HH-Equations (2). Note that a formal proof of this feature requires an analytical resolution of the systems of differential Equations (2) modeling each cell of the networks. The multiplicity of the ODE involved makes this a complex task. Hence, the correctness of the simulation process will be verified empirically rather than analytically. In this sense, our analysis remains at the level of an empirical evidence. The result can be formalized as follows:

Result 1. Any finite state automaton or transducer can be correctly simulated by a Hodgkin–Huxley based recurrent neural network modularly composed of synfire rings.

The rest of the section is devoted to the achievement of this result. First, we properly recall the concept of a synfire ring. Next, we describe an algorithmic construction which takes a finite state automaton or transducer as input and provides a corresponding HH-based neural network modularly composed of synfire rings Algorithm 2 Procedure which starts from a given automaton (transducer) and builds a HH-neural network composed of synfire rings that simulates

it (generalization of Minsky (1967)'s construction).	
Require: DFSA $\mathcal{A} = (Q, \Sigma, \delta_{\mathcal{A}}, q_0, F)$ (resp. DFST $\mathcal{T} = (Q, \Sigma, \delta_{\mathcal{T}}, q_0)$)	
1: set K input cells $(u_a)_{a \in \Sigma}$, where $K = \Sigma $	// input cells
2: set $K \times N$ synfire rings $(R_{q,a})_{q \in Q, a \in \Sigma}$, where $N = Q $	// internal rings
3: set K synfire rings $(R_{out,a})_{a \in \Sigma}$	<pre>// output rings - transducer's case</pre>
4: for all transition $(q, a, q') \in graph(\delta_{\mathcal{A}})$ (resp. $(q, a, q', o) \in graph(\delta_{\mathcal{T}})$) do	
5: add a bundle of input excitatory connections from u_a to $R_{q,a}$	// input connections
6: for all input symbol $a' \in \Sigma$ do	
7: add a bundle of inter-ring excitatory connections from $R_{q,a}$ to $R_{q',a'}$	<pre>// inter-ring excitatory connections</pre>
8: add a bundle of inter-ring inhibitory connections from $R_{q',a'}$ to $R_{q,a}$	<pre>// inter-ring inhibitory connections</pre>
9: end for	
10: add a bundle of output excitatory connections from $R_{q,a}$ to $R_{out,o}$	<pre>// output excitatory connections - transducer's case</pre>
11: for all output symbol $o' \in \Sigma$ do	
12: add a bundle of output inhibitory connections from $R_{out,o}$ to $R_{out,o'}$	<pre>// output inhibitory connections - transducer's case</pre>
13: end for	
14: end for	
15: set w_{input}^{exc} , w_{inter}^{exc} , w_{output}^{exc} such that the following conditions are satisfied:	// excitatory connections
 one bundle of input excitatory connections is not sufficient to activate a rin 	g onto which it projects
• one bundle of inter-ring excitatory connections is not sufficient to activate a	a ring onto which it projects
• the synchronized activations of one hundle of input excitatory connections a	and one hundle of inter ring excitatory connections is sufficient

ed activations of one bundle of input excitatory connections and one bundle of inter-ring excitatory to activate a ring onto which it projects

16: set w_{inter}^{inh} and w_{output}^{inh} such that the following condition is satisfied: // inhibitory connections • each single synapse of a bundle of inhibitory connections is sufficiently strong to knock out the activation the cell onto which it projects

as output (Algorithm 2). The construction is generic, in the sense that it can be applied to any finite state automaton or transducer. Afterwards, we formally define the concept of correct simulation of a finite state machine by a corresponding HH-based neural network. Finally, we empirically show that any automaton or transducer is correctly simulated by the corresponding HH-based neural network constructed by Algorithm 2.

4.1. Synfire rings

A synfire chain consists of a sequence of layers of neurons that are fully connected from one stratum to the next by means of excitatory synaptic connections (Abeles, 1982, 1991, 2004). A synfire ring is a synfire chain that loops back in on itself, i.e., where the last layer is connected to the first (Zheng & Triesch, 2014). A synfire chain and a synfire ring are illustrated in Fig. 5. The width of a synfire chain or ring refers to the number of cells composing each of its layer. The *length* is the number of layers composing it. For instance, the synfire chain and ring of Fig. 5 have width 3 and length 8. In a synfire chain or ring, the weights of the connections are assumed to be strong enough to ensure that a spiking activity can propagate from one layer to the next in a robust manner. The internal excitatory connections of a synfire rings are referred to as the intra-ring connections. The synaptic weight associated to these internal connections, denoted by w_{intra}^{exc} refers to the sum of the synaptic currents transmitted by each. The parameters of w_{intra}^{exc} are given in Table 2. Synfire rings will always be activated via one of their specific layers called the activation layer (dark blue cells in Fig. 5, 7 and 8). We assume that the rings are wired in such a way that the information propagates in the clockwise direction of rotation (gray or red little arrows in Fig. 7 and 8).

4.2. General construction

We describe the construction which starts from a given finite state automaton and builds a HH-based neural network modularly composed of synfire rings capable of simulating it. The construction is a generalization of Algorithm 1 to the context of synfire rings. The main idea consists first in replacing the cells $C_{q,a}$ and $C_{out,a}$ of Algorithm 1 (Minsky's construction) by corresponding synfire rings $R_{q,a}$ and $R_{out,a}$, respectively. The input cells of Algorithm 1 remain however unchanged. In addition, every synaptic connection of Minsky's construction is replaced by a bundle of excitatory connections, and in some cases also, by an additional reverse bundle of inhibitory connections. The whole process is described in detail in Algorithm 2. The next paragraphs are devoted to the presentation of the connectivity patterns and successive steps composing this construction.

First of all, we focus on the various connectivity patterns between synfire rings. Let *u* be an input cell and *R* be a synfire ring. A bundle of input excitatory connections (Fig. 6, top panel) is an ensemble of excitatory synapses projecting from *u* onto each cell of a (target) layer of R (one-to-all connections). The synaptic weight associated to this bundle, denoted by w_{input}^{exc} , refers to the sum of the synaptic currents transmitted by u. Moreover, let R_1 and R_2 be two synfire rings. A bundle of excitatory or inhibitory connections (Fig. 6, bottom panel) denotes an ensemble of excitatory or inhibitory synapses connecting each cell of a (source) layer of R_1 to each cell of a (target) layer of R_2 (all-to-all connections). The synaptic weights associated to these bundles, denoted by w^{exc} or w_{-}^{inh} , refer to the sum of the synaptic currents transmitted by each cell of the source layer. Hence, the intensity of the whole bundle is equal to $w^{exc} \cdot k$ or $w^{inh} \cdot k$, where k is the number of cells in the source layer. In the sequel, these weights will be denoted by w_{inter}^{exc} , w_{inter}^{inh} or w_{output}^{exc} , w_{output}^{inh} depending on whether R_1 is an output synfire ring or not (this will be specified in the next paragraphs). The values of these weights are determined by the formula and parameters specified in Eqs. (2) and Table 2.

We now describe the successive steps of the construction. First of all, each input cell u of Minsky's construction is kept unchanged, and each internal or output cell C of Minsky's construction is replaced by a corresponding internal or output synfire ring, respectively.

Moreover, each input connection of Minsky's construction between two cells u_i and C_i (orange arrows of Figs. 3 and 4) is replaced by a bundle of input excitatory connections projecting from the input cell u_i onto the activation layer of the targeted ring R_i (orange double arrows of Fig. 10 and 11). This bundle of connections is illustrated in Fig. 7. Its associated weights w_{input}^{exc} are chosen such that the connections from u_i to R_i are not sufficient to activate the activation layer of R_i . More precise conditions for the setting of these weights are provided in Algorithm 2. Note also that in our simulation process, an input signal from u_i to R_i



Fig. 5. A synfire chain and a synfire ring. Layers of neurons fully connected from one to the next by means of excitatory synaptic connections. The filled-in cells represent the activation layer of the ring.



Fig. 6. A bundle of input excitatory connections (top panel), and two bundles of inter-ring excitatory (bottom left panel) and inter-ring inhibitory (bottom right panel) connections. For each panel, the top figure illustrates the detailed situation, and the bottom one its schematic representation (used in the subsequent figures). Columns of cells represent rings' layers. Red or blue arrows represent excitatory or inhibitory connections between these layers, respectively. The synaptic weights w_{input}^{exc} , w_{in}^{inh} associated to these bundles are also represented. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

is rather simulated by the transmission of a synaptic current I_{input} to each cell of the activation layer of R_i .

Furthermore, each internal connection of Minsky's construction between two cells C_i and C_i (black arrows of Figs. 3 and 4) is replaced by two reverse bundles of inter-ring connections. The first one consists of excitatory connections projecting from the activation layer of ring R_i onto that of ring R_i . The second one is composed of reverse inhibitory connections projecting from a layer of R_i located "after" its activation layer onto a layer of R_i located "before" its activation layer, where "before" and "after" are relative to the clockwise direction of rotation (black double arrows of Fig. 10 and 11). These bundles of inter-ring excitatory and inhibitory connections are illustrated in Fig. 7. Their respective weights w_{inter}^{exc} and w_{inter}^{inh} are chosen such that the two following conditions are satisfied: First, the bundle of inter-ring excitatory connections from R_i to R_j is not sufficient to activate the activation layer of R_j . Second, if ring R_j becomes active via its activation layer, then it will send back to R_i an inhibition that is sufficiently strong to knock out the activation that the latter ring might have received recently. In this way, the activation of a subsequent ring always triggers the inhibition of the previous one, ensuring that except during the transition phases, at most one internal synfire ring is always active. Further conditions for the setting of these weights are provided in Algorithm 2.

Also, each output connection of Minsky's construction between an internal cell C_i and an output cell $C_{out,j}$ (red arrows of Fig. 4) is replaced by a bundle of excitatory connections projecting from the activation layer of ring R_i onto that of ring $R_{out,j}$ together with reverse bundles of inhibitory connections (of the kind described above) between $R_{out,j}$ and all other output rings (red double arrows of Fig. 11). These bundles of excitatory and inhibitory output connections are illustrated in Fig. 8. Their respective weights w_{output}^{exc} and w_{output}^{inh} are chosen such that the two following conditions are satisfied: First, the excitatory connections from ring R_i to $R_{out,j}$ are sufficient to activate $R_{out,j}$. Second, if ring $R_{out,j}$ becomes active, then it will send to all other output rings an inhibition that is sufficiently strong to knock them out.

The parameters of the HH-cells and the weights of the intraring connections are set such that any activated synfire ring will necessarily settle into a self-sustained activity, as long as it is not affected by any inter-ring inhibitory connection. The



Fig. 7. Connectivity pattern between an input cell and two synfire rings. The light blue circle and gray rings represent an input cell u and two internal synfire rings R_i and R_j , respectively. A bundle of input excitatory connections (red arrow) projects from cell u onto the activation layer of ring R_j . The two rings R_i and R_j are connected by bundles of inter-ring excitatory and inhibitory connections. A bundle of inter-ring excitatory connections (red arrow) projects from the activation layer of R_i onto that of R_j . A reverse bundle of inter-ring inhibitory connections (blue arrow) projects from a layer of R_j located "after" its activation layer onto a layer of R_i located "before" its activation layer. The *transition-ring mechanism* refers to the dynamical situation where the combined activations of u and R_i provoke the activation of R_j , which in turn leads to the inhibition of R_i . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 8. Connectivity pattern between an internal synfire ring and two output synfire rings. The gray and red rings represent an internal and two output rings R_i , $R_{out,j}$, $R_{out,k}$, respectively. A bundle of output excitatory connections (red arrow) projects from the activation layer of R_i onto that of $R_{out,j}$. Two reverse bundles of output inhibitory connections (blue arrows) link $R_{out,j}$. Two reverse one projects from a layer located "after" the activation layer of one ring onto a layer located "before" the activation layer of the other ring. The *output-ring mechanism* refers to the dynamical situation where the activation R_i provokes the activation of $R_{out,j}$, which in turn leads to the inhibition of $R_{out,k}$, supposing that the latter was already active. (For interpretation of the series to color in this figure legend, the reader is referred to the web version of this article.)

precise conditions that these weights need to satisfy are given in Algorithm 2. The values of the parameters characterizing all synaptic currents are given in Table 2.

The correctness of this construction relies on two key dynamical processes, referred to as the *transition-ring* and *output-ring mechanisms*, illustrated in Figs. 7 and 8, respectively. In the first

case (Fig. 7), an input cell u and a synfire ring R_i both project via bundles of input and inter-ring connections onto a second synfire ring R_j . The synaptic weights w_{input}^{exc} , w_{inter}^{exc} , w_{inter}^{inh} can be fine-tuned in such a way that the combined activations of u and R_i provoke the activation of R_i , which in turn leads to the inhibition of R_i. The synaptic parameters given in Table 2 ensure that this mechanism is correctly realized in the present context of HH cells. The correctness of this mechanism is illustrated by the raster plot of Fig. 9 (left panel), and the issue of its secure implementation analyzed in Section 6.5. In the second case (Fig. 8), an internal synfire ring R_i projects via a bundle of output connections onto an output synfire ring $R_{out,j}$, which is connected by a bundle of output inhibitory connections to another output synfire ring $R_{out,k}$. The synaptic weights w_{output}^{exc} and w_{output}^{inh} can finely-tuned such that the activation of R_i provokes the activation of $R_{out,j}$ which in turn leads to the inhibition of $R_{out,k}$, supposing that the latter was already active. Again, the synaptic parameters given in Table 2 ensure that this mechanism is correctly achieved. The correctness of this mechanism is illustrated by the raster plot of Fig. 9 (right panel).

The whole construction is formally described in Algorithm 2. According to this procedure, the HH-based neural networks composed of synfire rings simulating the automata and of Figs. 1 and 2 are schematically illustrated in Fig. 10 and 11, respectively.

4.3. The concept of "simulation"

We provide a formal definition of the concept of *simulation* of an automaton or a transducer by a corresponding neural network. Towards this purpose, we first come back to Minsky's construction. Algorithm 1 takes as input an automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ or a transducer $\mathcal{T} = (Q, \Sigma, \delta, q_0)$ and provides as output a corresponding Boolean neural network $\mathcal{N}_{\mathcal{A}}$ or $\mathcal{N}_{\mathcal{T}}$, respectively. We recall that for any input stream $w = a_0 a_1 \cdots a_n \in \Sigma^*$, the computations of \mathcal{A} and \mathcal{T} on w are denoted by

$$\mathcal{A}(w): q_0 \xrightarrow{a_0} q_{i_1} \xrightarrow{a_1} q_{i_2} \cdots \xrightarrow{a_n} q_{i_{n+1}}$$

$$\mathcal{T}(w): q_0 \xrightarrow{a_0/o_0} q_{i_1} \xrightarrow{a_1/o_1} q_{i_2} \cdots \xrightarrow{a_n/o_n} q_{i_{n+1}}.$$
(3)

In addition, for any sequence of symbols $w = a_0 a_1 \cdots a_n \in \Sigma^*$, let $\bar{w} = u_{a_0} u_{a_1} \cdots u_{a_n}$ denote the successive activations of the input cells $u_{a_0}, u_{a_1}, \ldots, u_{a_n}$ of $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$) at times 0, 1, ..., *n*. We say that \mathcal{A} (resp. \mathcal{T}) is correctly simulated by $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$), if and only if, when \mathcal{A} (resp. \mathcal{T}) and $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$) are run in parallel on input streams w and \bar{w} , respectively, then the successive states of \mathcal{A} (resp. states and outputs of \mathcal{T}) are perfectly reflected with a time delay of 1 time step (resp. 1 and 2 time steps) by the successive spiking internal cells of $\mathcal{N}_{\mathcal{A}}$ (resp. internal and output cells of $\mathcal{N}_{\mathcal{T}}$). Formally, \mathcal{A} (resp. \mathcal{T}) is correctly simulated by $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$), if and only if, for any $w = a_0 a_1 \cdots a_n$, the computation of $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$) on $\bar{w} = u_{a_0} u_{a_1} \cdots u_{a_n}$ yields the following sequence of spiking internal cells (resp. internal and output cells)

This definition ensures that the successive states of A (resp. successive states and outputs of T), given in Relation (3), correspond precisely to the indices of the successive spiking internal cells of N_A (resp. internal and output cells of N_T), given in Relation (4).

The proof that any automaton \mathcal{A} (resp. transducer \mathcal{T}) is correctly simulated by the Boolean neural network $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$) provided by Algorithm 1 is fairly straightforward and can be



Fig. 9. Activities of the transition-ring and the output-ring mechanisms. The cells are modeled by the HH-equations (2). In both rasters, each dot represents the activation of a whole layer of a synfire ring (instead of a single cell). **Left raster:** Two synfire rings R_0 and R_1 (composed of 12 layers of 3 cells each) are connected as described in Fig. 7. The combined activations of R_0 (lower raster trace) and of the input current (upper blue trace), provoke the activation of R_1 (upper raster trace), which in turn, leads to the inhibition of R_0 (lower raster no-trace). This dynamics corresponds precisely to the *transition-ring mechanism*. **Right raster:** One internal synfire ring $R_{q,1}$ and two output synfire rings $R_{out,0}$ and $R_{out,1}$ (composed of 12 layers of 3 cells each) are connected as described in Fig. 8. If $R_{out,0}$ is already active (middle raster trace), then the activation of $R_{q,1}$ (lower raster trace), induced by an input current in this case (upper blue trace), provokes the activation of $R_{out,1}$ (upper raster trace), which in turn, leads to the inhibition of $R_{out,0}$ (middle raster trace). This dynamics corresponds precisely to the output-ring mechanism. **Right raster:** One internal synfire rings $R_{u,1}$ and two output synfire rings $R_{out,0}$ and $R_{out,1}$ (composed of 12 layers of 3 cells each) are connected as described in Fig. 8. If $R_{out,0}$ is already active (middle raster trace), which in turn, leads to the inhibition of $R_{out,0}$ (middle raster no-trace). This dynamics corresponds precisely to the output-ring mechanism. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 10. Schematic representation of the HH-neural network composed of synfire rings simulating the automaton of Fig. 1. The blue circles are the input HH-cells. The black double circles represent the internal synfire rings made up of HH-cells. The orange and black arrows represent the bundles of input and inter-ring connections described in Fig. 7. Note that for each bundle of inter-ring excitatory connections projecting from a ring R_i to a ring R_j (black connection), there is a corresponding reverse bundle of inter-ring inhibitory connections from R_j to R_i that is not represented (cf. Fig. 7). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 11. Schematic representation of the HH-neural network composed of synfire rings simulating the transducer of Fig. 2. The blue circles are the input cells. The black and red double circles represent the internal and output synfire rings made up of HH-cells, respectively. The orange, black and red arrows represent bundles of input, inter-ring and output connections described in Figs. 7 and 8, respectively. The blue arrows between $R_{out,0}$ and $R_{out,1}$ are two reverse bundles of inhibitory output connections described in Figs. 8. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

found in Minsky (1967). The correct simulations of automaton \mathcal{A} and transducer \mathcal{T} of Figs. 1 and 2 by their corresponding Boolean neural networks $\mathcal{N}_{\mathcal{A}}$ and $\mathcal{N}_{\mathcal{T}}$ of Figs. 3 and 4 can be verified in Table 1: see rows 3 and 7 for the dynamics of \mathcal{A} and $\mathcal{N}_{\mathcal{A}}$, as well as rows 9–10 and 16–17 for those of \mathcal{T} and $\mathcal{N}_{\mathcal{T}}$.

Similarly, Algorithm 2 takes as input an automaton \mathcal{A} or a transducer \mathcal{T} and provides as output a corresponding HH-based neural network composed of synfire rings $\mathcal{N}_{\mathcal{A}}^{HH}$ or $\mathcal{N}_{\mathcal{T}}^{HH}$, respectively. For any sequence of symbols $w = a_0a_1 \cdots a_n \in \Sigma^*$, let $\hat{w} = u_{a_0}, u_{a_1}, \ldots, u_{a_n}$ denote the successive activations of the input cells $u_{a_0}, u_{a_1}, \ldots, u_{a_n}$ of $\mathcal{N}_{\mathcal{A}}^{HH}$ (resp. $\mathcal{N}_{\mathcal{T}}^{HH}$) at times t_0, t_1, \ldots, t_n by an input current of amplitude 1.9 nA and duration 4 ms (cf. Section 3), and where the t_i 's are sufficiently distant apart from one another in order for the network to be able to settle into stable synfire ring activations between t_i and t_{i+1} , for $0 \leq i \leq n-1$ (cf. Section 6 for further detail). We say that \mathcal{A} (resp. \mathcal{T}) is correctly simulated by $\mathcal{N}_{\mathcal{A}}^{HH}$ (resp. $\mathcal{N}_{\mathcal{T}}^{HH}$) on $\hat{w} = u_{a_0}, u_{a_1}, \ldots, u_{a_n}$ yields the following sequence of activated internal rings (resp. internal and output rings)

Here again, the definition ensures that the successive states of \mathcal{A} (resp. successive states and outputs of \mathcal{T}), given in Relation (3), correspond precisely to the indices of the successive activated internal rings of $\mathcal{N}_{\mathcal{A}}$ (resp. internal and output rings of $\mathcal{N}_{\mathcal{T}}$), given in Relation (5).

4.4. Justification of Result 1

We now show that the construction given in Algorithm 2 correctly fulfills the statement of Result 1. The idea consists in showing that the dynamics of the Boolean network $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$) is stably and correctly reflected by that of the HH-based network $\mathcal{N}_{\mathcal{A}}^{\text{HH}}$ (resp. $\mathcal{N}_{\mathcal{T}}^{\text{HH}}$), in the precise sense described below. Since \mathcal{A} (resp. \mathcal{T}) is correctly simulated by $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$), then so is it by $\mathcal{N}_{\mathcal{A}}^{\text{HH}}$ (resp. $\mathcal{N}_{\mathcal{T}}^{\text{HH}}$). This analysis relies on the empirical evidence that the two transition-ring and output-ring mechanisms (cf. Fig. 7 and 8) can be successfully and stably implemented in the present context (cf. Fig. 9). A thorough analysis of this feature is provided in Section 6.5. In this sense, our analysis remains at the level of an empirical evidence.

First of all, note that Algorithm 2 is a generalization of Algorithm 1. In line 1, we also consider k input cells $(u_a)_{a \in \Sigma}$, where $k = |\Sigma|$. In lines 2–3, instead of considering Boolean cells $(C_{q,a})_{q \in Q, a \in \Sigma}$ and $(C_{out,a})_{a \in \Sigma}$, we consider synfire rings $(R_{q,a})_{q \in Q, a \in \Sigma}$ and $(R_{out,a})_{a \in \Sigma}$. In lines 4–14, instead of adding single input, internal and output synaptic connections between Boolean cells, we add bundles of input, inter-ring and output connections between input cells and rings, between internal rings, and between internal and output rings, respectively. Finally instead of setting the weights of the input and internal connections to 1/2 and those of the output connections to 1, we tune the synaptic weights of the connections bundles so as to fulfill the conditions of lines 15–16. The synaptic parameters given in Table 2 guarantee that these properties are satisfied.

The patterns of connectivity and their corresponding synaptic weights ensure the successful implementations of the transitionring and output-ring mechanisms (cf. Fig. 7, 8, 9 and Section 6.5). More precisely, if an internal ring $R_{q,a}$ receives combined activations from an input cell u_a and another internal ring $R_{q',a'}$ (via bundles of input and inter-ring connections), then $R_{q,a}$ will become activated and $R_{q',a'}$ will in turn be inhibited. In addition, if an output ring $R_{out,o}$ receives activation from an internal ring $R_{q,a}$ (via output connections), then $R_{out,o}$ will become activated and every other output rings $R_{out,o'}$ will in turn be inhibited, for all $o' \in \Sigma$.

According to the parallelism between Algorithms 2 and 1 as well as to the successful implementations of the transitionring and output-ring mechanisms, if $\mathcal{N}_{\mathcal{A}}^{HH}$ (resp. $\mathcal{N}_{\mathcal{T}}^{HH}$) receives the input pattern $\hat{w} = u_{a_0}, u_{a_1}, \ldots, u_{a_n}$, then it will mimic the behavior of $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$) over input stream $\bar{w} = u_{a_0}u_{a_1}\cdots u_{a_n}$, in the following sense: the network $\mathcal{N}_{\mathcal{A}}^{HH}$ (resp. $\mathcal{N}_{\mathcal{T}}^{HH}$) will have its internal ring $R_{q,a}$ (resp. its internal ring $R_{q,a}$ and output ring $R_{out,o}$) activated after receiving input signal u_{a_i} , if and only if, the network $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$) has its internal Boolean cell $C_{q,a}$ (resp. its internal Boolean cell $C_{q,a}$ and output Boolean cell $C_{out,o}$) spiking 1 time step (resp. 1 and 2 time steps) after receiving input u_{a_i} , for each $0 \leq i \leq n$. Hence, throughout the computational process, the sequence of activated internal rings (resp. and output rings) of $\mathcal{N}_{\mathcal{A}}^{HH}$ (resp. $\mathcal{N}_{\mathcal{T}}^{HH}$) is the exact counterpart to that of spiking internal cells (resp. and output cells) of $\mathcal{N}_{\mathcal{A}}$ (resp. $\mathcal{N}_{\mathcal{T}}$), given in Relation (4), namely:

Relations (3) and (6) show that the automaton \mathcal{A} (resp. the transducer \mathcal{T}) is *correctly simulated* by the HH-neural network $\mathcal{N}_{\mathcal{A}}^{\text{HH}}$ (resp. $\mathcal{N}_{\mathcal{T}}^{\text{HH}}$). The correct simulations of automaton \mathcal{A} and transducer \mathcal{T} of

The correct simulations of automaton \mathcal{A} and transducer \mathcal{T} of Figs. 1 and 2 by their corresponding HH-based neural networks $\mathcal{N}_{\mathcal{A}}^{HH}$ and $\mathcal{N}_{\mathcal{T}}^{HH}$ of Fig. 10 and 11 can be verified in Table 3 (Section 5): see rows 3 and 4 for the dynamics of \mathcal{A} and $\mathcal{N}_{\mathcal{A}}^{HH}$, as well as rows 7–8 and 9–10 for those of \mathcal{T} and $\mathcal{N}_{\mathcal{T}}^{HH}$.

5. Numerical simulations

In order to validate the correctness of our construction (Algorithm 2), we show by means of numerical simulations that the HH-based neural networks of Figs. 10 and 11 *correctly simulate* the automaton and transducers of Fig. 1 and 2, in the sense of Section 4.4. The governing Equations of the neurons' dynamics and their parameters are described in Section 3.

We consider synfire rings composed of 12 layers of 3 neurons each, which amounts to a total of 36 neurons. The number of 12 layers has been chosen in order to satisfy the requirements described in Section 6.3. Accordingly, each activated synfire ring will necessarily settle into a self-sustained activity, as long as it does not receive any other inhibition.

The HH-network simulating the automaton of Fig. 1 is subjected to 2 input signals (start signal is omitted) and is composed of 10 internal synfire rings (cf. Fig. 10). We simulated the activity of this network receiving input pattern 00101100. The raster plot of this simulation is presented in Fig. 12. The internal rings that are successively activated are reported in Table 3. The activation of ring $R_{q_{4,0}}$ shows that the pattern 0110 has been correctly detected in the input. This experiment shows that the automaton of Fig. 1 is correctly simulated by the HH-based neural network composed of synfire rings of Fig. 10. Indeed, the successive states of the automaton (Table 3, row 3)

$q_0, q_1, q_1, q_2, q_1, q_2, q_3, q_4$

are correctly reflected by the indices of the successive activated rings of the corresponding network (Table 3, row 4)

$R_{q_0,0}, R_{q_1,0}, R_{q_1,1}, R_{q_2,0}, R_{q_1,1}, R_{q_2,1}, R_{q_3,0}, R_{q_4,0}.$

Similarly, The HH-network simulating the automaton of Fig. 2 is subjected to 4 input signals (start signal is omitted) and is composed of 8 internal and 2 output synfire rings (cf. Fig. 11). We simulated the activity of the network during the computation of the sum $s = \frac{111001}{+101011}$. The raster plot of this simulation



Fig. 12. Raster plot of the activity of the HH-based neural network of Fig. 10 simulating the automaton of Fig. 1. The raster traces are composed of successive dots, each of which representing the activation of a whole layer of a synfire ring. Inputs (0, 1) and (1, 0) encode the automaton's inputs 0 and 1, respectively. Inputs are transmitted to the network in the form of constant currents of fixed duration (blue trace). In this case, the input stream corresponds to the encoding of the binary input 00101100. The raster shows the synfire rings that are successively activated. We see that the last activated ring that is $R_{q_4,0}$. This encodes the fact of the corresponding automaton being in final state q_4 , and therefore, that the input pattern is accepted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 13. Raster plot of the activity of the HH-based neural network of Fig. 11 **simulating the transducer of** Fig. 2. The raster traces are composed of successive dots, each of which representing the activation of a whole layer of a synfire ring. Inputs (0, 0), (0, 1), (1, 0), (1, 1) encode the transducer's inputs $\binom{0}{0}, \binom{0}{1}, \binom{1}{0}, \binom{1}{1}$, respectively. Inputs are transmitted to the network in the form of constant currents of fixed duration (blue trace). In this case, the input stream corresponds to the encoding of the binary sum $s = \frac{1}{1} \frac{1}{0} \frac{1}{1} \frac{0}{0} \frac{1}{1}$: the successive pairs of bits of *s* are given in the reverse order. The raster shows the successive internal and output rings that are successively activated. The sequence of activated output rings corresponds to the encoding of the result of *s* in the reverse order, namely, 0010011. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

is presented in Fig. 13. The inputs submitted to the network correspond to the encodings the successive pairs of bits of *s* in the reverse order. The internal and output rings that are successively activated are reported in Table 3. The activity of the output rings

(red patterns) corresponds to the encoding of the result of *s* in the reverse order. Here again, this experiment shows that the automaton of Fig. 2 is correctly simulated by the HH-based neural network of synfire rings of Fig. 11. Indeed, the successive states

and outputs of the transducer (Table 3, rows 7-8)

 $q_0, q_1, q_1, q_0, q_1, q_1, q_1$ and 0, 0, 1, 0, 0, 1, 1

are correctly reflected by the indices of the successive activated rings of its corresponding network (Table 3, rows 9–10)

 $\begin{array}{l} R_{q_{0},\binom{1}{1}}, R_{q_{1},\binom{0}{1}}, R_{q_{1},\binom{0}{0}}, R_{q_{0},\binom{1}{1}}, R_{q_{1},\binom{1}{0}}, R_{q_{1},\binom{1}{1}}, R_{q_{1},\binom{0}{0}} \text{ and } \\ R_{out,0}, R_{out,0}, R_{out,1}, R_{out,0}, R_{out,0}, R_{out,1}, R_{out,1}. \end{array}$

6. Stability, robustness and representational capacity

Sections 4 and 5 empirically show that any finite state automaton can be correctly simulated by a corresponding HH-based neural network modularly composed of synfire rings. The stability of the simulation process relies on the following four dynamical features: (i) the possibility of a Hodgkin–Huxley cell to be periodically reactivated; (ii) the ability of synfire rings to generate self-sustained activities; (iii) the ability of synfire rings to generate temporally synchronized activities; (iv) the possibility to activate and inhibit synfire rings in a controlled manner, in order to ensure the implementations of the transition-ring mechanism and the output-ring mechanism (cf. Fig. 7 and 8). Furthermore, the robustness of the simulation process derives from the resistance of those dynamical features to cell death and/or synaptic failure mechanisms.

In the following subsections, we first study the *self-sustained activities* of synfire rings. We then focus on the *stability* and *robustness* of the these dynamics as a function of the ring topologies and the presence of synaptic noises. We finally touch on issue of the *representational capacity* of the synfire ring neural architecture, by considering the possibility of overlapping rings.

6.1. Refractory period and depolarization block

The modified Hodgkin–Huxley Equations (2) capture the modeling of the *refractory period* of the cells, i.e., the period following a spike and during which any further stimulation of the cell has no effect in (cf. Fig. 14). The refractory period is affected by the intrinsic parameters of the cell (cf. Table 2): its membrane capacitance as well as the maximum conductances, potentials, half potentials, slope of (in-)activation and time constants, all of them related to the leaky, sodium and potassium currents (Tchaptchet, 2019).

The intrinsic parameters of the cells used in our model (cf. Table 2) induce a total spike duration – refractory period included – of about 8 ms. Consequently, if two activations of a cell occur within a delay of less than 8 ms, then only one action potential will be generated: due to the refractory period, the cell membrane will not be ready for responding to the second stimulus. We will show that the refractory period of the cells imposes a strong condition on the ring topology in order to ensure the possibility of generating self-sustained activities (cf. Section 6.3).

The modified Hodgkin–Huxley Equations (2) also capture the effect of *depolarization block*—the silent state occurring when the neuron receives excessive excitation (cf. Fig. 15) (Bianchi et al., 2012). In fact, input stimuli of larger intensities induce responding spikes of higher frequencies (Fig. 15, cases C,D,E), up to the level where the input current becomes too large. In that case, the cell to enter the depolarization block, and no more spiking response is possible (Fig. 15, case F). We will show that the depolarization block effect imposes restrictions on the intensities of the input and intra-ring synaptic current (cf. Sections 6.4 and 6.6).



Fig. 14. Refractory period. Different responses of the cell (blue trace) to different pairs of input stimuli of same intensities (red trace). **Cases A,B,C.** Due to the refractory period of the cell, the short time delay between the two inputs makes it impossible for it to generate a second action potential. **Case D.** The two inputs are sufficiently spaced in time for the cell to generate two action potentials. (For interpretation of the references to color in this figure legend, the reder is referred to the web version of this article.)



Fig. 15. Spike frequency and depolarization block. Different responses of the cell (blue trace) to different input currents of constant intensities (red trace). Cases C,D,E. The larger the input current, the higher the spiking frequency of the cell. Case F. When the input current becomes too high, the cell enters the depolarization block, where no action potential is possible. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.2. Self-sustained activity

The *self-sustained activity* of a synfire ring refers to the dynamics where the successive layers of the ring are spiking in a time-locked and persistent manner, as illustrated in Fig. 18 (middle panel for instance). A self-sustained activity is characterized by the following dynamical properties:

- the successive layers are activated in a synchronous way (vertical spiking patterns in Fig. 18)
- the successive layers are activated in a persistent way (periodicity of the vertical spiking patterns in Fig. 18)
- the time intervals between the successive layer activations remain constant over time (constant distance between vertical spiking patterns in Fig. 18)

As regards self-sustained activities, the synfire ring topology exhibits two important properties. First, it leads to the emergence of a *temporal structure*, which refers to the discrete-time steps at which the successive layers of the rings are activated.⁶ The time structure is discrete, regular and persistent over time. Second and

 $^{^{6}}$ Note that the modified HH-model Eq. (2), by its continuous-time nature, imposes no discrete-time dynamical feature to the neurons.

Table 3

Rows 1–3. Computation of the automaton of Fig. 1 over input 00101100. **Row 4.** Dynamics of the HH-based neural network of Fig. 10 over the encoding the same input 00101100 (cf. Raster plot of Fig. 12 for the timing of the inputs). **Rows 5–7.** Computation of the transducer of Fig. 2 over the sum $\begin{array}{c} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{array}$ **Rows 4–9.** Dynamics of the HH-based neural network of Fig. 11 over the encoding of the same input $\begin{array}{c} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{array}$ (cf. Raster plot of Fig. 13 for the timing of the inputs).

Time steps	0	1	2	3	4	5	6	7	8		
Inputs	0	0	1	0	1	1	0	0	-		
	◄ · · · · · · · pattern 0110 · · · · · · ►										
States	q_0	q_1	q_1	<i>q</i> ₂	q_1	<i>q</i> ₂	<i>q</i> ₃	q_4 detected	q_4		
Rings $R_{s,i}$	-	$R_{q_0,0}$	$R_{q_{1},0}$	$R_{q_{1},1}$	$R_{q_2,0}$	$R_{q_{1},1}$	$R_{q_{2},1}$	$R_{q_{3},0}$	$R_{q_4,0}$ detected		
Time steps	0	1	2	3	4	5	6	7	8		
Inputs	$\binom{1}{1}$	$\begin{pmatrix} 0\\1 \end{pmatrix}$	$\begin{pmatrix} 0\\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1\\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0\\ 0 \end{pmatrix}$	-			
States	q_0	q_1	q_1	q_0	q_1	q_1	q_1	q_0	-		
Outputs	0	0	1	0	0	1	1	-	-		
\blacktriangleleft result of the sum s in the reverse order											
Rings R _{s,i}	-	$R_{q_0, \binom{1}{1}}$	$R_{q_1, \binom{0}{1}}$	$R_{q_1, \begin{pmatrix} 0 \\ 0 \end{pmatrix}}$	$R_{q_0, (1)}$	$R_{q_1, \binom{1}{0}}$	$R_{q_1, \binom{1}{1}}$	$R_{q_1, \begin{pmatrix} 0 \\ 0 \end{pmatrix}}$	-		
Ring Rout,j	-	Rout,0	R _{out,0}	Rout, 1	Rout,0	Rout,0	Rout, 1	Rout, 1	-		
\triangleleft encoding of the result of the sum s in the reverse order											

most importantly, the synfire topology turns out to enhance and consolidate *synchronicity* of the successive layer activations. In other terms, even if not present at the start of the dynamics, the synfire topology forces the emergence of a synchronous dynamics. This property is explained by the full converging/diverging connectivity pattern between the successive layers of the rings. In fact, each cell of a subsequent layer l_{k+1} receives the same synaptic connections from all cells of the preceding layer l_k . Accordingly, all cells of l_{k+1} integrate the same synaptic inputs, and in turn, produce the very same – thus synchronous – spiking activity. The synchronizing capability of a ring is illustrated in Fig. 16. While the cells of the very first layer are activated with time delays of up to 1 ms, the subsequent layer sturn out to be perfectly synchronized, from the second layer already.

6.3. Length of the rings

The possibility for a ring to generate self-sustained activities is directly related to its length. There is a minimal ring length – determined by the refractory period of the cells – below which self-sustained activities are not possible.

More precisely, in a synfire ring, the spiking of two successive layers l_{k-1} and l_k is possible only if the activations sent by l_{k-1} are received by l_k at a time when the cells of l_k are not in their refractory periods anymore. Otherwise, the spike propagation would be interrupted (Fig. 14, cases A,B,C). In order to ensure the possibility of a self-sustained activity, the *period* of the ring – the time delay between two activations of a same layer – must therefore be larger than the refractory period of the cells. Clearly, the period of the ring is related to its length: the larger the length, the larger the period. Whenever the ring length is too small, the spiking activity will not be able to propagate throughout the successive layers in a persistent way. On the other hand, there is no theoretical upper bound on the ring's length that would prevent from a time-locked inter-layer spike propagation.

In the present case, according to the parameters of our cells (cf. Table 2), the ring's length must be larger than 5 to ensure the possibility of a self-sustained activity. Fig. 17 (first graph) illustrates the spiking activity of a synfire ring of length 3 (and width 3). In this case, the spikes cannot propagate back from the third to the first layer: the reactivation of the cells of the first layer (blue trace) occurs during their refractory periods, preventing them from spiking again. A self-sustained activity is thus not possible. A synfire ring of length 4 would also not permit a self-sustained activity. Fig. 17 (second graph) illustrates the spiking activity of a synfire ring of length 5 (and width 3). A self-sustained activity is now generated. The color repetition highlights the periodicity of

the ring's dynamics. The fact that the second blue spike is slightly shorter than the first indicates that cells of the first layer have not fully recovered their resting states when receiving their second activations (Fig. 14, case D). This "anomaly" is recovered in the next periods (the next blue spikes recover full amplitude). Fig. 17 (third graph) illustrates the spiking activity of a synfire ring of length 25 (and width 3). Here again, a self-sustained activity is generated. In this case, the spiking pattern is highly stable both over time and intensity. In all simulations, the parameters of the intra-ring synaptic currents are set to $a_{intra}^{exc} = 5.0$ and $b_{intra}^{exc} = 1.0$.

Note that the period of the ring is not only influenced by the ring length, but also by the intensities of the excitatory intraring synaptic currents. Larger synaptic currents produce faster openings and closings of the ionic channels, which in turn induce faster spike generations (Fig. 15, cases C,D,E), thus leading to a smaller ring period. This feature is emphasized in more detail in Section 6.6.

6.4. Width of the rings

Contrary to the ring's length, the width of rings does not influence the possibility to generate self-sustained activities. Rings composed of reliable neurons and synapses can theoretically give rise to self-sustained activities, irrespective of their width (assuming that they are long enough, cf. Section 6.3). However, in the presence of cell-based or synaptic-based failure mechanisms, wider rings tend to be more robust than thinner ones.

The dynamics of three synfire rings of length 10 and widths 2, 5 and 25 are illustrated in Fig. 18 respectively. In all cases, a proper self-sustained activity is obtained. But in order to ensure a regular and persistent inter-layer spike propagation, the synaptic strengths of the intra-ring synapses must be adjusted with respect to the ring width. In fact, the wider the ring, the larger the number of intra-ring synapses projecting onto each cell, and, hence, the larger the synaptic currents received by each layer cell. Consequently, the intra-ring synaptic currents must be adapted in such a way that the sum of input currents received by each cell permits its activation. Currents of too low intensities will not be able to activate the layer cells, while too high currents will drive them into their depolarization block, preventing them from firing (Fig. 15, case F). In our context, the parameters of the intra-ring synaptic currents in the first and second rasters of Fig. 18 are set to $a_{intra}^{exc} = 5.0$ and $b_{intra}^{exc} = 1.0$, whereas those of the third raster are $a_{intra}^{exc} = 0.5$ and $b_{intra}^{exc} = 1.0$. In the latter case, a wider ring imposes the consideration of lower intra-ring currents, in order to avoid the depolarization block effect.



Fig. 16. Synchronizing capability of the synfire ring topology. The cells of the first layer are activated with time delays of up to 1 ms (ill-aligned first vertical spiking pattern). From the second layer already, the successive layer activations become perfectly synchronized (well-aligned vertical spiking patterns). Bottom plot. An input current of 5 nA and 0.4 ms is injected to each cell of their first layer to initiate the rings' dynamics.



Fig. 17. Spiking activities of 3 synfire rings. The successive spikes represent the activities of cells belonging to successive layers of the rings. They show the propagation of activity through the successive layers. **First graph.** Synfire ring of length 3 (and width 3). The spike propagation is interrupted at the third layer. **Second graph.** Synfire ring of length 5 (and width 3). The ring generates a self-sustained activity, as shown by the periodicity of the colored spikes. **Third graph.** Synfire ring of length 25 (and width 3). Only the activities of the first 5 cells are displayed in different colors. Here again, the ring generates a self-sustained activity, as shown by the periodicity of the colored spikes. **Bottom plot.** Input current of 5 nA and 0.4 ms injected to each cell of the rings' first layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 18. First raster. Activity of a synfire ring of length 10 and width 2. The raster shows the activations of the successive layers of the ring. After each activation of the (upper) 10-th layer, the (downer) first layer is being reactivated. Hence, a self-sustained activity is generated. **Second raster.** Self-sustained activity of a synfire ring of length 10 and width 5. **Third raster.** Self-sustained activity of a synfire ring of length 10 and width 25. In this case, the ring is wider than long. **Bottom plot.** Input current of 5 nA and 0.4 ms injected to each cell of the first layer of each ring.

6.5. Transition between rings' activities

The correctness of our construction resides in the controlled activations and deactivations of synfire rings by means of combined triggers coming from inputs cells and other rings. In fact, the transition-ring and output-ring mechanisms (Figures 7 and 8) rely on the possibility for a ring R_j , when activated, to trigger the inhibition of another ring R_i . We show that this mechanism can be securely implemented in our framework. For this purpose, the inhibition currents must be sufficiently strong (cf. Table 2 for the parameters characterizing these currents). But most importantly, the activation and inhibited layers of R_i (cf. Fig. 7, dotted dark blue and dotted white layers of the left-hand ring) need to be neither too close nor too far apart from each other. We focus on the explanation of this feature.

In Fig. 7, let the activation and inhibited layers of R_i (dotted dark blue and dotted white layers) be denoted by l_1^i and l_k^i , respectively; let also the activation and inhibition layers of R_j (dotted dark blue and dotted white layers) be denoted by l_1^i and l_2^j , respectively; let the input cell (upper isolated cell) be denote by u. Now, suppose that the activation layer l_1^i and the input cell u are both spiking within a specific short time interval.

Their combined current activate layer l_1^l , which initiates the selfsustained activity of ring R_j . When layer l_2^j is activated, it sends inhibitions to layer l_k^i . Meanwhile, the self-sustained activity of R_i continues to propagate, and at some point, the cells of layer l_k^i will integrate the temporal summation of the inhibitory and excitatory postsynaptic potentials (IPSPs and EPSPs) generated by the cells of layers l_2^i and l_{k-1}^i , respectively. The l_k^i cells will then either stay silent or produce action potentials, depending on whether this temporal summation is below or above their firing threshold. Note that the intensity of the temporal summation depends on the position of l_k^i in R_i . In Fig. 19, $l_k^i = l_4^i$. In this case, the temporal summation is below the threshold's cells, the self-sustained activity of ring R_i is blocked, and the ring transition process is correctly achieved. In Fig. 20, $l_k^i = l_8^i$. In this case, the temporal summation is above the threshold's cells, the selfsustained activity of ring R_i is pursued, and the ring transition process is defective. In our case, taking l_k^i as the fourth layer of R_i ensures a secured implementation of the transition-ring mechanism.



Fig. 19. Transition-ring mechanism (cf. Fig. 7). Activity of two synfire rings R_i (blue trace, cells 0 to 29) and R_j (red trace, cells 30 to 59) of lengths 10 and widths 3. The first input stimulus (bottom red trace) activates the first layer of R_i (first red arrow), which then enters into a self sustained activity (blue pattern). A second input stimulus from an external cell u (bottom turquoise trace) occurs during a time period that overlaps with the reactivation of the first layer l_1^i of R_i . The combined activations of u and l_1^i activate the first layer l_1^j of R_j (two red arrows), which in turn enters into a self sustained activity (red pattern). The second layer l_2^j of R_j inhibit the fourth layer l_4^i of R_i (blue round arrow). The spiking activity of R_i is blocked by this inhibition. Therefore, the transition of activities from R_i to R_j is correctly achieved. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 20. Transition-ring mechanism (cf. Fig. 7). The situation is similar to that of Fig. 19, except that the second layer l_2^i of R_j inhibit the eighth layer l_8^i (instead of the fourth l_4^i) of R_i (blue round arrow). By the time the spiking activity of R_i reaches layer l_8^i , the inhibition sent by l_j^2 has weakened, and l_8^i can thus be activated. Consequently, the self-sustained activity of R_i is not blocked, and the transition of activities from R_i to R_j is unsuccessful. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.6. Robustness

Transmission at individual synaptic contacts can be highly unreliable, with presynaptic nerve impulses failing to evoke a postsynaptic response (Allen & Stevens, 1994). Due to the redundancy of connections projecting onto each layer cell, the synfire ring topology provides the networks with important robustness capabilities. The self-sustained activities of the rings can be preserved in cases of synaptic unreliability.

To illustrate this feature, we first study the robustness of the self-sustained activity of synfire rings where for each cell c, a given percentage α of its incoming synaptic connections are randomly deactivated. The failure rate α is common to all cells. A proper self-sustained activity can be generated, as long as each cell receives synaptic currents of sufficiently high intensities. Figures 21 and 22 illustrate the dynamics of a synfire ring of length 10 and width 5 subjected to different failure rates. The parameters of the intra-ring synaptic currents are set to a_{intra}^{exc} = 2.0 and $b_{intra}^{exc} = 1.0$. In Fig. 21, 20% (black pattern) and 50% (blue pattern) of the synaptic connections projecting onto each cell are randomly deactivated, respectively. In both cases, a proper self-sustained activity is obtained. The two situations lead to different inter-layer spike propagation frequencies. This feature is explained as follows: the lower the failure rate, the higher the intensities of the intra-ring synaptic currents received by the cells, hence the faster their spiking responses (cf. Fig. 15, cases C,D,E), and thus the higher the ring period. Fig. 22 illustrates the case where 80% of the connections projecting onto each cell are



Fig. 21. Raster plot. Activities of a synfire ring of length 10 and width 5 where 20% (black pattern) and 50% (blue pattern) of the synaptic connections projecting onto each cell are randomly deactivated. For the sake of conciseness, both patterns are represented on the same graph. In both cases, a self-sustained activity is generated. The lower the failure rate, the higher the inter-layer spike propagation frequency, thus the higher the ring period (illustrated by the superimposition of the two patterns). **Bottom plot**. Input current of 5 nA and 0.4 ms injected to each cell of the first layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 22. Top graph. Activity of a synfire ring of length 10 and width 5 where 80% of the incoming synaptic connections to each cell are randomly deactivated. The spiking activity vanishes at the second layer already. From the third layer onwards, the activities of the cells remain completely flat. Bottom plot. Input current of 5 nA and 0.4 ms injected to each cell of the first layer.



Fig. 23. Raster plot. Activities of a synfire ring of length 10 and width 5 where each cell *c* has a certain percentage α_c of its incoming synaptic connections that are randomly deactivated. The cyclic attractor dynamics of the ring is preserved, but the temporal structure, given by the synchronicity successive layer activations, is lost. **Bottom plot.** Input current of 5 nA and 0.4 ms injected to each cell of the first layer.

randomly discarded. The self-sustained activity is not possible anymore. The spike train stops at the second layer already, due to insufficient synaptic currents transmitted by the first layer.

In the preceding simulations, the failure rate α is fixed for all cells. Now, we consider the case where each cell *c* has a specific percentage α_c of its incoming synaptic connections that

are randomly deactivated. The failure rates α_c are drawn from the uniform distribution $\mathcal{U}(0, 1)$. Hence, on average, 50% of the synaptic connections of the rings are discarded, but some cells receive their full synaptic currents (if $\alpha_c = 0$), some receive only a portion of them (if $\alpha_c \in [0, 1[)$), and some are dead (if $\alpha_c = 1$). In this case, the cyclic attractor dynamics of the ring is preserved, but synchronicity of the successive layer activations is lost. Fig. 23 illustrates the dynamics of a synfire ring of length 10 and width 5 subjected to this synaptic failure mechanism. The spiking activity is able to propagate throughout the successive layers without vanishing and in a periodic manner (since the α_c are fixed across time). The average rate of 50% of remaining synapses in the whole ring permits the continuous propagation of a spiking pattern. Hence, the cyclic attractor dynamics of the ring is preserved. However, the layer cells do not spike in a synchronous way anymore. In fact, since the layer cells have different percentages of their incoming synaptic connections that are randomly deactivated, they receive synaptic currents of different intensities, and in turn, show responses at different timings (cf. Fig. 15, cases C,D,E). The layer responses are therefore desynchronized, meaning that the time-locked inter-layer temporal structure is lost.

6.7. Overlapping rings

In the biological context of cell assemblies, same cells are likely to belong to various assemblies (Abeles, 1991; Izhikevich, 2006; Palm et al., 2014). This overlapping feature ensures that each cell might participate to different neural functionalities, thus enhancing the representational capacity of the networks. According to these considerations, the simulation of finite automata by neural networks composed of overlapping rings is an issue of specific interest. In this context, in order for the rings to preserve their full encoding capabilities, the overlapping pattern must satisfy the following property: at each time *t*, the dynamics of each ring should never be able to activate, inhibit or desynchronize the dynamics of another ring. For instance, two different rings could never possess any layer in common, since the activation of such shared layer would lead to an uncontrolled activation of both rings.

We consider two synfire rings R_1 and R_2 and show that, for some specific overlapping scheme, the dynamics of the rings remain controllable, and hence, their full encoding capabilities guaranteed. The overlapping pattern under consideration is illustrated in Fig. 24. Accordingly, the common cells of R_1 and R_2 (black cells) are part of a same layer of R_1 (blue ring), but are spread across successive layers of R_2 (red ring). The dynamics of two such overlapping R_1 and R_2 are illustrated in Figures 25 and 26. In the first case (Fig. 25), only one ring, say R_1 , is active. We see that the sustained activity of R_1 does not elicit undesired activation of R_2 other than some sporadic spikes. In the second case (Fig. 26), both R_1 and R_2 are active. We observe that their respective dynamics are not significantly disrupted by the spiking of the shared cells, since only additional sporadic spikes appear. In both cases, the dynamical encoding of the rings, represented by their self-sustained activities, is therefore guaranteed.

7. Conclusion

We introduced a paradigm for neural computation based on robust, temporally precise and self-sustained activities of cell assemblies—the *synfire rings*. The proposed paradigm is capable of abstract computation via the simulation of specific abstract machines. More specifically, we empirically showed that finite state automata can be simulated by Hodgkin–Huxley recurrent neural networks modularly composed of synfire rings. The proposed model of computation possesses the following characteristics:

- The successive computational states are encoded into temporally robust cyclic attractor dynamics, instead of discrete spiking configurations;
- The transitions between such attractors are perfectly determined by the connectivity pattern between the rings, and triggered by the input signals;



Fig. 24. Two overlapping synfire rings R_1 (**blue**) and R_2 (**red**). For the purpose of clarity, the synfire rings are represented as synfire chains. But the leftmost and rightmost layers of R_1 are assumed to coincide, and the lowermost and uppermost layers of R_2 also coincide. The rings R_1 and R_2 contain 4 cell in common (black bold cells). Note that the 3 leftmost and 3 rightmost black cells are the same. The lowermost black cell is also the same as the uppermost one. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- The self-sustained activity of synfire rings is highly stable with respect to variations in their topologies.
- The global computational process is robust to some architectural failures and synaptic noises, due to the redundancy of the connectivity patterns.

From a purely theoretical point of view, finite state automata are the simplest kinds of finite state machines. They recognize the class of *regular languages*, which corresponds to the lowest level of the Chomsky's hierarchy. Their memory is limited to their number of states. More complex finite state machines, like counter automata, pushdown automata or Turing machines, can all be seen as automata provided with different kinds of external memories. Nevertheless, these abstract machines are at the basis of the fundamental conception of computation, which consists in successive "mechanical-like" input driven transitions between different computational states. From a practical point of view, however, large enough automata capture the computational capabilities of any current digital machine. Our results state that the automaton-based conception of computation can be achieved by bio-inspired neural networks modularly composed of synfire rings.

As regards optimality, the conception of a synfire ring based neural network simulating a given automaton (Algorithm 2) is a generalization of Minsky (1967)'s construction (Algorithm 1). The idea consists in replacing the cells and connections of Minsky's network by corresponding synfire rings and bundles of connections, respectively. According to Algorithm 2, an automaton with *i* input symbols and *n* states is simulated by a corresponding network containing $\mathcal{O}(i \cdot n)$ rings, and hence, containing $\mathcal{O}(i \cdot n)$ cells also (note that the number of cells per ring in bounded, i.e., does not increase with the size of the network) (Minsky, 1967). However, Minsky (1967)'s construction is known to be sub-optimal. It has been shown that any deterministic automaton with *n* states can be implemented by a neural network of optimal size containing $\Theta(\sqrt{n})$ cells (Horne & Hush, 1996; Indyk, 1995). We claim that the optimal-size construction of Horne and Hush (1996) can also be generalized to the context of synfire rings, in a similar manner as (Minsky, 1967)'s construction. As a consequence, an automaton with *n* states could a priori be simulated



Fig. 25. Dynamics of two overlapping synfire rings. Two rings (blue and red) of lengths 10 and widths 5 overlap according to the pattern of Fig. 24. Accordingly, the cells of the blue ring's first layer numbered 1, 2, 3, 4, 5, and those of the red ring's layers 1 to 5 numbered 50, 55, 60, 65, 50, are pairwise equivalent (highlighted with black squared). For the sake of clarity, they are represented as distinct, but blue and red black squared columns of cells are the same. Only the blue ring is activated. Despite being overlapping, the self-sustained activity of the blue ring does not elicit any self-sustained activity of the ring; but only sporadic spikes of the shared cells. The encoding represented by the rings' dynamics is thus guaranteed. **Bottom plot.** Input current of 5 nA and 0.4 ms injected to each cell of the first layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Fig. 26. Dynamics of two overlapping synfire rings. Two rings (blue and red) of lengths 20 and widths 5 overlap according to the pattern of Fig. 24. Accordingly, the cells of the blue ring's 13th layer numbered 60, 61, 62, 63, 64, and those of the red ring's layers 3 to 7 numbered 110, 111, 112, 113, 114, are pairwise equivalent (highlighted with black squared). Both rings are activated. Despite being overlapping, the self-sustained activities of the rings are not significantly perturbed, with only additional sporadic spikes of the shared cells. The encoding represented by the rings' dynamics is thus guaranteed. **Bottom plot.** Input current of 5 nA and 0.4 ms injected to each cell of the first layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

by an optimal-size neural network containing $\Theta(\sqrt{n})$ rings and cells. This issue is expected to be studied in more detail in a future work.

Our work follows the general theory of cell assemblies (Braitenberg, 1978; Hebb, 1949; Palm, 1982; Palm et al., 2014), but more specifically, fits within the framework of operational cell assemblies, as developed for instance by Fay et al. (2005), Garagnani et al. (2009), Markert et al. (2005), Wennekers (2006, 2007, 2009), Wennekers et al. (2006), Wennekers and Palm (2009). In this context, Hebbian cell assemblies consist in generic fully (or densely) connected pools of neurons, and correspond to fixed point attractors. They "serve as representations for either static entities (attractors) or temporal memory contents (synfire chains)" (Wennekers et al., 2006). The assemblies are considered as "building blocks", and thus can be combined in a modular way to form rulelike state transition graphs-referred to as synfire graphs (Wennekers, 2006). The combination process is achieved by means of few associative principles in line with Hebb's original approach: auto-associative mappings; hetero-associative mappings; and sometimes also input-gated mappings (Wennekers et al., 2006). The dynamical transitions across assemblies are activated in a controlled manner, either by specific input sequences (Wennekers et al., 2006) or by unspecific trigger signals (Wennekers & Palm, 2009). In addition, operational cell assemblies can either be implemented constructively, during initialization phase, or can selforganize by means of ongoing synaptic plasticity mechanisms, e.g., incremental covariance learning rule or synaptic plasticity rule with fixed LTP/LTD thresholds (Wennekers et al., 2006). In the case of specific inputs and deterministic transitions, the obtained systems are computationally equivalent to *finite state automata*, while in the case of non-deterministic transitions, they implement *Markov chains*. As a consequence, operational cell assemblies can handle functionalities of a well-defined complexity. They have been applied in the contexts of language and complex behavior models (Garagnani et al., 2009; Markert et al., 2007; Wennekers, 2006, 2007, 2009; Wennekers et al., 2006)

In our context, the *synfire rings* play the role of the *Hebbian cell assemblies*. The modular combination of synfire rings into state transition graphs – the synfire graphs (Wennekers, 2006) – also requires few combining principles: the input connections, the intra-ring connections, the inter-ring connections and the transition- and output-ring mechanisms, respectively. The main difference of our approach resides in that the synfire graphs are

composed of synfire rings instead of Hebbian cell assemblies. Accordingly, our "building blocks" correspond to cyclic instead of fixed points attractors, and most of all, they do underlie "spatiotemporal sequences of activity patterns" (Wennekers et al., 2006). In fact, the self-sustained activities of the rings lead to the emergence of a "primary" temporal structure at a "micro-level". On top of this, the controlled transitions between attractors, along the edges of the synfire graphs, induce a "secondary" temporal structure at a "macro-level". In this sense, our model permits the development of nested temporal structures at different scales. These considerations are in line with the original reflections by Braitenberg (1978), who saw in modular cell assemblies the possibility to generate a "hierarchical temporal structure" composed of temporal structures at different levels (Palm et al., 2014). The second difference is that, as opposed to Hebbian cell assemblies, synfire rings give rise to periodic dynamics. The periods and phases differences of the rings could be thus exploited as additional encoding information. Indeed, our findings show that the ring periods depend directly on the intensity of the intra-ring synaptic currents (Fig. 21). Wennekers and Palm (1996) further showed that the wave velocities of synfire chains - and hence the periods of synfire rings - could be controllable by the intensity of external inputs. Accordingly, synfire rings might capture characteristics of neural oscillators (Hoppensteadt & Izhikevich, 2000; Malagarriga et al., 2015; Xu et al., 2004; Zanin et al., 2011), and as such, constitute a bridge between the cell assembly and the neural oscillator theories.

In biological networks, neural assemblies are likely to be overlapping with each other, rather than being disjoint (Abeles, 1991; Palm et al., 2014). In an associative memory model, where memories are either embedded into attractors via Hebbian cell assemblies (Braitenberg, 1978; Hebb, 1949; Hopfield, 1982) or encoded into spatio-temporal patterns via synfire chains (Abeles, 1991) or polychronous groups of neurons (Izhikevich, 2006), this overlapping feature ensures that memories are stored in a distributed manner. In the case of Hebbian cell assemblies, Boolean (Hopfield, 1982) networks can store a maximal number of uncorrelated memory patterns $P_{max} = 0.14N$, where N is the number of neurons (Amit, Gutfreund, & Sompolinsky, 1985). More biologically realistic spiking networks provide robust associative storage of sparse patterns at a capacity close to the one of technical networks (Sommer & Wennekers, 2001). In the case of synfire rings, Boolean and integrate-and-fire winners-take-all networks can embed a maximal number of pools of $P_{max} \cong 8N$, where N is the number of neurons (Bienenstock, 1995; Herrmann, Hertz, & Prügel-Bennett, 1995; Hertz, 1999). As for balanced networks of spiking neurons, upper limits on memory capacity for both cases of Hebbian assemblies and synfire chains can be obtained, and correspond to $P_{max} = 0.1N_E$ and $P_{max} = 0.065N_E$, respectively, where N_E is the number of excitatory neurons (Aviel, Horn, & Abeles, 2005). In the case of polychronous groups, the memory capacity of the system turns out to exceed the number of units composing it (Izhikevich, 2006; Izhikevich, Gally, & Edelman, 2004; Szatmáry & Izhikevich, 2010). Along these lines, the investigation of neural networks composed of overlapping rings, initiated in Section 6.7, is expected to be pursued.

Furthermore, as for the context of general cell assemblies (Palm et al., 2014), the central question of learning within the specific synfire ring neural architecture is expected to be studied. As a first step, the auto-associations (i.e., the intra-ring connections) could be frozen, and only the hetero-associations (i.e., the intra-ring connections) would be subjected to spike-timing dependent plasticity (STDP) mechanisms. Accordingly, Hebbian-like learning mechanisms shall lead to the self-organization of designated hetero-associative memory models composed of synfire rings .

As a third research direction, the proposed paradigm is intended to be generalized towards the achievement of Turing-like

computation. First (unpublished) results in this direction show that fixed-space Turing machines can be simulated by synfire ring based neural networks composed of Boolean cells. Even if fixedspace Turing machines are (only) computationally equivalent to finite state automata, these results are of conceptual relevance: the synfire-ring based neural networks could be considered – and thus programmed - like Turing machines up to their maximal representational capabilities.⁷ In this context, the program of the machine is represented by a suitably interconnected pool of rings. The tape content, head position and symbol currently read by the machine are represented by overlying sequences of rings. The computation of the machine is then simulated by updating, in a controlled manner, the dynamics of the pool and overlying sequences of rings, every time a new input is received. The generalization of the construction to the case of synfire ring based neural networks composed of Hodgkin-Huxley cells is envisioned for future work.

Finally, with these achievements, we do not intend to argue that brain computational processes really proceed via simulations of finite state automata in the very way described here. Rather, our intention is to show that a paradigm for abstract neural computation, based on sustained activities of cell assemblies, is possible and potentially exploitable. As a consequence, biological neural networks should in principle be capable of simulating finite state automata, whether via the proposed paradigm, or via some other one. Finally, should the proposed networks be implemented in patterns of *in vitro* cultures of neurons (Feinerman, Rotem, & Moses, 2008; Wolf & Geisel, 2008), it would lead to the realization of biological neural abstract computers.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Supports from the Defense Advanced Research Projects Agency (DARPA), USA, Microsystems Technology Office (MTO), USA, Lifelong Learning Machines program (L2M), USA, cooperative agreement number HR0011-18-2-0023, as well as from the Czech Science Foundation, project "FoNeCo", grant number GA19-05704S are gratefully acknowledged.

References

Abeles, M. (1982). Local cortical circuits. an electrophysiological study. In *Studies of brain function (Vol. 6)*. Berlin, Heidelberg, New York: Springer-Verlag.

- Abeles, M. (1991). Corticonics: Neuronal circuits of the cerebral cortex (1st ed.). Cambridge University Press.
- Abeles, M. (2004). Time is precious. Science, 304, 523-524.
- Allen, C., & Stevens, C. F. (1994). An evaluation of causes for unreliability of synaptic transmission. Proceedings of the National Academy of Sciences of the United States of America, 91, 10380–10383.
- Alon, N., Dewdney, A. K., & Ott, T. J. (1991). Efficient simulation of finite automata by neural nets. *Journal of the ACM*, *38*, 495–514.
- Amit, D. J., Gutfreund, H., & Sompolinsky, H. (1985). Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, 55, 1530–1533.

⁷ Similarly, digital computers, as finite binary machines, are computationally equivalent to finite state automata (FSA). However, they benefit from being generally conceptualized as (fixed-space) Turing machines (TM). For instance, the TM-based conception of computers lead programmers to always define counters in their programs, knowing full well that these counters cannot count indefinitely, but only up to some maximal integer imposed by the memory capacity of the computers. Such practices would not occur if computers were generally conceived as FSA instead of TM, since FSA, unlike TM, cannot count.

- Aviel, Y., Horn, D., & Abeles, M. (2005). Memory capacity of balanced networks. Neural Computation, 17, 691–713.
- Balcázar, J. L., Gavaldà, R., & Siegelmann, H. T. (1997). Computational power of neural networks: a characterization in terms of kolmogorov complexity. *IEEE Transactions on Information Theory*, 43, 1175–1183.
- Bianchi, D., Marasco, A., Limongiello, A., Marchetti, C., Marie, H., Tirozzi, B., et al. (2012). On the mechanisms underlying the depolarization block in the spiking dynamics of CA1 pyramidal neurons. *Journal of Computational Neuroscience*, 33, 207–225.
- Bienenstock, E. (1995). A model of neocortex. Network. Computation in Neural Systems, 6, 179–224.
- Braitenberg, V. (1978). Cell assemblies in the cerebral cortex. In R. Heim, & G. Palm (Eds.), *Theoretical approaches to complex systems* (pp. 171–188). Berlin, Heidelberg: Springe.
- Buonomano, D. V. (2005). A learning rule for the emergence of stable dynamics and timing in recurrent networks. *Journal of Neurophysiology*, *94*, 2275–2283.
- Cabessa, J., & Duparc, J. (2016). Expressive power of nondeterministic recurrent neural networks in terms of their attractor dynamics. *IJUC*, *12*, 25–50.
- Cabessa, J., & Finkel, O. (2019). Computational capabilities of analog and evolving neural networks over infinite input streams. *Journal of Computer and Systems Sciences*, 101, 86–99.
- Cabessa, J., Horcholle-Bossavit, G., & Quenet, B. (2017). Neural computation with spiking neural networks composed of synfire rings. In A. Lintas, S. Rovetta, P. F. M. J. Verschure, & A. E. P. Villa (Eds.), Lecture notes in computer science: Vol. 10613, Artificial neural networks and machine learning - ICANN 2017 -26th international conference on artificial neural networks, Proceedings, Part I (pp. 245–253). Springer.
- Cabessa, J., & Masulli, P. (2017). Emulation of finite state automata with networks of synfire rings. In 2016 international joint conference on neural networks (pp. 4641–4648). IEEE.
- Cabessa, J., & Siegelmann, H. T. (2011). Evolving recurrent neural networks are super-turing. In *Proceedings of IJCNN 2011* (pp. 3200–3206). IEEE.
- Cabessa, J., & Siegelmann, H. T. (2012). The computational power of interactive recurrent neural networks. *Neural Computation*, 24, 996–1019.
- Cabessa, J., & Siegelmann, H. T. (2014). The super-turing computational power of plastic recurrent neural networks. *International Journal of Neural Systems*, 24.
- Cabessa, J., & Tchaptchet, A. (2018). Automata computation with hodgkin-huxley based neural networks composed of synfire rings. In 2018 international joint conference on neural networks (pp. 1–8). IEEE.
- Cabessa, J., & Villa, A. E. P. (2012). The expressive power of analog recurrent neural networks on infinite input streams. *Theoretical Computer Science*, 436, 23–34.
- Cabessa, J., & Villa, A. E. P. (2014). An attractor-based complexity measurement for boolean recurrent neural networks. PLoS One, 9(e94204+).
- Cabessa, J., & Villa, A. E. P. (2015). Recurrent neural networks and superturing interactive computation. In P. Koprinkova-Hristova, V. Mladenov, & K. N. Kasabov (Eds.), Artificial neural networks: Methods and applications in bio-/neuroinformatics (pp. 1–29). Springer.
- Cabessa, J., & Villa, A. E. P. (2016). Expressive power of first-order recurrent neural networks determined by their attractor dynamics. *Journal of Computer* and Systems Sciences, 82, 1232–1250.
- Diesmann, M., Gewaltig, M.-O., & Aertsen, A. (1999). Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402, 529–533.
- Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14, 179-211.
- Fay, R., Kaufmann, U., Knoblauch, A., Markert, H., & Palm, G. (2005). Combining visual attention, object recognition and associative information processing in a neurobotic system. In S. Wermter, G. Palm, & M. Elshaw (Eds.), *Biomimetic* neural learning for intelligent robots: intelligent systems, cognitive robotics, and neuroscience (pp. 118–143). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Feinerman, O., Rotem, A., & Moses, E. (2008). Reliable neuronal logic devices from patterned hippocampal cultures. *Nature Physics*, 4, 967–973.
- G., Păun (2000). Computing with membranes. Journal of Computer and Systems Sciences, 61, 108–143.
- Garagnani, M., Wennekers, T., & Pulvermüller, F. (2009). Recruitment and consolidation of cell assemblies for words by way of hebbian learning and competition in a multi-layer neural network. *Cognitive Computation*, 1, 160–176.
- Gilson, M., Burkitt, A., & Van Hemmen, L. (2010). Stdp in recurrent neuronal networks. Frontiers in Computational Neuroscience, 4, 1–15.
- Hartley, R., & Szu, H. (1987). A comparison of the computational power of neural network models. In C. Butler (Ed.), *Proceedings of the IEEE first international conference on neural networks* (pp. 17–22). IEEE.
- Hebb, D. O. (1949). The organization of behavior: A neuropsychological theory. John Wiley & Sons Inc.
- Herrmann, M., Hertz, J., & Prügel-Bennett, A. (1995). Analysis of synfire chains. Network. Computation in Neural Systems, 6, 403–414.
- Hertz, J. A. (1999). Modelling synfire networks. In O. P. G. Burdet, & P. Combe (Eds.), Neuronal information processing: From biological data to modelling and applications (pp. 27–46). Wold scientific.

- Hertz, J., & Prügel-Bennett, A. (1996). Learning synfire chains by selforganization. Network. Computation in Neural Systems, 7, 357–363.
- Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal* of *Physiology*, 117, 500–544.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 79, 2554–2558.
- Hoppensteadt, F. C., & Izhikevich, E. M. (2000). Synchronization of laser oscillators, associative memory, and optical neurocomputing. *Physical Review E*, 62, 4010–4013.
- Horn, D., Levy, N., Meilijson, I., & Ruppin, E. (1999). Distributed synchrony of spiking neurons in a hebbian cell assembly. In S. A. Solla, T. K. Leen, & K. Müller (Eds.), Advances in neural information processing systems (Vol. 12), NIPS conference (pp. 129–135). The MIT Press.
- Horne, B. G., & Hush, D. R. (1996). Bounds on the complexity of recurrent neural network implementations of finite state machines. *Neural Networks*, 9, 243–252.
- Hyötyniemi, H. (1996). Turing machines are recurrent neural networks. In J. Alander, T. Honkela, & J. M. (Eds.), STeP '96 - genes, nets and symbols; finnish artificial intelligence conference (pp. p13–24). Vaasa, Finland: University of Vaasa, Finnish Artificial Intelligence Society (FAIS), STeP '96 - Genes, Nets and Symbols; Finnish Artificial Intelligence Conference, Vaasa, Finland, 20–23 August 1996.
- Ikegaya, Y., Aaron, G., Cossart, R., Aronov, D., Lampl, I., Ferster, D., et al. (2004). Synfire chains and cortical songs: Temporal modules of cortical activity. *Science*, 304, 559–564.
- Indyk, P. (1995). Optimal simulation of automata by neural nets. In *STACS* (pp. 337–348).
- Izhikevich, E. M. (2006). Polychronization: computation with spikes. Neural Computation, 18, 245–282.
- Izhikevich, E. M., Gally, J. A., & Edelman, G. M. (2004). Spike-timing dynamics of neuronal groups. *Cerebral Cortex*, 14, 933–944.
- Jun, J. K., & Jin, D. Z. (2007). Development of neural circuitry for precise temporal sequences through spontaneous activity, axon remodeling, and synaptic plasticity. *PLoS One*, 2, 1–17.
- Kilian, J., & Siegelmann, H. T. (1996). The dynamic universality of sigmoidal neural networks. *Information and Computation*, 128, 48-56.
- Kitano, K., Câteau, H., & Fukai, T. (2002). Self-organization of memory activity through spike-timing-dependent plasticity. *Neuroreport*, 13, 795–798.
- Kleene, S. C. (1956). Representation of events in nerve nets and finite automata. In C. Shannon, & J. McCarthy (Eds.), *Automata studies* (pp. 3–41). Princeton, NJ: Princeton University Press.
- Levy, N., Horn, D., Meilijson, I., & Ruppin, E. (2001). Distributed synchrony in a cell assembly of spiking neurons. *Neural Networks*, 14, 815–824.
- Maass, W. (1999). Computing with spiking neurons. In W. Maass, & C. M. Bishop (Eds.), Pulsed neural networks (pp. 55–85). Cambridge, MA, USA: MIT Press.
- Maass, W., & Bishop, C. M. (Eds.), (1999). *Pulsed neural networks*. Cambridge, MA, USA: MIT Press.
- Mainen, Z., & Sejnowski, T. (1995). Reliability of spike timing in neocortical neurons. Science, 268, 1503–1506.
- Malagarriga, D., García-Vellisca, M. A., Villa, A. E. P., Buldú, J. M., García-Ojalvo, J., & Rivero, A. J. P. (2015). Synchronization-based computation through networks of coupled oscillators. *Frontiers in Computing Neuroscience*, 9.
- Markert, H., Knoblauch, A., & Palm, G. (2005). Detecting sequences and understanding language with neural associative memories and cell assemblies. In S. Wermter, G. Palm, & M. Elshaw (Eds.), *Biomimetic neural learning for intelligent robots: Intelligent systems, cognitive robotics, and neuroscience* (pp. 107–117). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Markert, H., Knoblauch, A., & Palm, G. (2007). Modelling of syntactical processing in the cortex. *Biosystems*, 89, 300–315.
- Masuda, N., & Kori, H. (2007). Formation of feedforward networks and frequency synchrony by spike-timing-dependent plasticity. *Journal of Computational Neuroscience*, 22, 327–345.
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysic, 5, 115–133.
- Mehring, C., Hehl, U., Kubo, M., Diesmann, M., & Aertsen, A. (2003). Activity dynamics and propagation of synchronous spiking in locally connected random networks. *Biological Cybernetics*, *88*, 395–408.
- Minsky, M. L. (1967). Computation: Finite and infinite machines. Englewood Cliffs, N. J.: Prentice-Hall, Inc.
- Neary, T. (2015). Three small universal spiking neural p systems. Theoretical Computer Science, 567, 2–20.
- Neto, J. P. G., Siegelmann, H. T., Costa, J. F., & Araujo, C. P. S. (1997). Turing universality of neural nets (revisited). In F. Pichler, & R. Moreno-Díaz (Eds.), Lecture notes in computer science: Vol. 1333, Computer aided systems theory - EUROCAST'97, a selection of papers from the 6th international workshop on computer aided systems theory, proceedings (pp. pp. 361–366). Springer.
- Omlin, C. W., & Giles, C. L. (1996). Constructing deterministic finite-state automata in recurrent neural networks. *Journal of the ACM*.

Palm, G. (1982). Neural assemblies: An alternative approach to artificial intelligence. Berlin, Heidelberg: Springer-Verlag.

- Palm, G., Knoblauch, A., Hauser, F., & Schüz, A. (2014). Cell assemblies in the cerebral cortex. *Biological Cybernetics*, 108, 559–572.
- Pollack, J. B. (1987). On connectionist models of natural language processing (Ph.D. thesis), Las Cruces, NM: Computing Research Laboratory, New Mexico State University.
- Postnova, S., Finke, C., Huber, M. T., Voigt, K., & Braun, H. A. (2012). Conductancebased models for the evaluation of brain functions, disorders, and drug effects. In E. Mosekilde, O. Sosnovtseva, & A. Rostami-Hodjegan (Eds.), *Biosimulation in biomedical research, health care and drug development* (pp. 97–132). Vienna: Springer Vienna.
- Păun, G. (2002). Membrane computing. An introduction. Berlin: Springer-Verlag.
- Siegelmann, H. T. (1996). Recurrent neural networks and finite automata. Computational Intelligence, 12, 567–574.
- Siegelmann, H. T. (2003). Neural and super-turing computing. *Minds and Machines*, 13, 103-114.
- Siegelmann, H. T., & Sontag, E. D. (1994). Analog computation via neural networks. Theoretical Computer Science, 131, 331–360.
- Siegelmann, H. T., & Sontag, E. D. (1995). On the computational power of neural nets. Journal of Computer and Systems Sciences, 50, 132–150.
- Sommer, F. T., & Wennekers, T. (2001). Associative memory in networks of spiking neurons. Neural Networks, 14, 825–834.
- Szatmáry, B., & Izhikevich, E. M. (2010). Spike-timing theory of working memory. PLoS Computational Biology, 6, e1000879.
- Tchaptchet, A. (2018). Activity patterns with silent states in a heterogeneous network of gap-junction coupled huber-braun model neurons. *Chaos. An Interdisciplinary Journal of Nonlinear Science*, 28, 106327.
- Tchaptchet, A. (2019). Numerical implementation of neuronal diversity with regard to a more realistic computer model for research and teaching (Ph.D. thesis), Institute of Physiology, Philipps University of Marburg.

- Tchaptchet, A., Postnova, S., Finke, C., Schneider, H., Huber, M. T., & Braun, H. A. (2013). Modeling neuronal activity in relation to experimental voltage/patch-clamp recordings. *Brain Research*, 1536, 159–167, Selected papers presented at the Tenth International Neural Coding Workshop, Prague, Czech Republic.
- Turing, A. M. (1948). Intelligent machinery: Technical report, UK: National Physical Laboratory Teddington.
- Wennekers, T. (2006). Operational cell assemblies as a paradigm for braininspired future computing architectures. *Neural Information Processing Letters Review*, 10, 135–145.
- Wennekers, T. (2007). A cell assembly model for complex behaviour. *Neurocomputing*, 70, 1988–1992.
- Wennekers, T. (2009). On the natural hierarchical composition of cliques in cell assemblies. Cognitive Computation, 1, 128–138.
- Wennekers, T., Garagnani, M., & Pulvermüller, F. (2006). Language models based on hebbian cell assemblies. Journal of Physiology-Paris, 100, 16–30.
- Wennekers, T., & Palm, G. (1996). Controlling the speed of synfire chains. In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, & B. Sendhoff (Eds.), Lecture notes in computer science: Vol. 1112, Artificial neural networks - ICANN 96, 1996 international conference, Proceedings (pp. 451–456). Springer.
- Wennekers, T., & Palm, G. (2009). Syntactic sequencing in hebbian cell assemblies. Cognitive Neurodynamics, 3, 429–441.
- Wolf, F., & Geisel, T. (2008). Neurophysics: Logic gates come to life. Nature Physics, 4, 905–906.
- Xu, D., Principe, J. C., & Harris, J. G. (2004). Logic computation using coupled neural oscillators. In Circuits and systems, 2004. ISCAS '04. Proceedings of the 2004 international symposium on (Vol. 5) (pp. V-788-V-791).
- Zanin, M., De. Pozo, F., & Boccaletti, S. (2011). Computation emerges from adaptive synchronization of networking neurons. *PLoS One*, 6, 1–6.
- Zheng, P., & Triesch, J. (2014). Robust development of synfire chains from multiple plasticity mechanisms. Frontiers in Computing Neuroscience, 8.