Journal of Computer and System Sciences 82 (2016) 1232–1250

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Expressive power of first-order recurrent neural networks determined by their attractor dynamics

Jérémie Cabessa [a,*], Alessandro E.P. Villa [b]

[a] Laboratory of Mathematical Economics (LEMMA), University Panthéon-Assas – Paris 2, 4 Rue Blaise Desgoffe, 75006 Paris, France
[b] Laboratory of Neuroheuristics (NHRG), Faculty of Business and Economics (HEC), University of Lausanne, CH-1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 19 October 2015
Received in revised form 31 March 2016
Accepted 19 April 2016
Available online 27 June 2016

Keywords:
Recurrent neural networks
Neural computation
Analog computation
Evolving systems
Learning
Attractors
Spatiotemporal patterns
Turing machines
Expressive power
ω-languages

ABSTRACT

We provide a characterization of the expressive powers of several models of deterministic and nondeterministic first-order recurrent neural networks according to their attractor dynamics. The expressive power of neural nets is expressed as the topological complexity of their underlying neural ω-languages, and refers to the ability of the networks to perform more or less complicated classification tasks via the manifestation of specific attractor dynamics. In this context, we prove that most neural models under consideration are strictly more powerful than Muller Turing machines. These results provide new insights into the computational capabilities of recurrent neural networks.

1. Introduction

In the central nervous system, one neuron may receive and send projections from and to thousands of other neurons. The huge number of connections established by a single neuron and the slow integration time of neurons, operating in the milliseconds range (billion times slower than presently available supercomputers), suggest that information in the nervous system might be transmitted by simultaneous discharges of large sets of neurons. In addition, the presence of recurrent connections within large neural circuits indicates that re-entrant activity through chains of neurons should represent a major hallmark of brain circuits [5]. Also, developmental and/or learning processes are likely to potentiate or weaken certain pathways through the network by affecting the number or efficacy of synaptic interactions between the neurons [19,30]. Hence, the activation of functional cell assemblies in distributed networks might be induced by transmissions of complex patterns of activity [1]. In fact, various experimental studies suggest that specific attractor dynamics [20,21,56] as well as spatiotemporal pattern of discharges (i.e., ordered and precise interspike interval relationships) [2,49,51,53,54,57] are likely to be significantly involved in the processing and coding of information in the brain. Moreover, the association between attractor dynamics and repeating firing patterns has been demonstrated in nonlinear dynamical systems [3,4] and in simulations of large scale neuronal networks [28,29].

* Corresponding author.
E-mail address: jeremie.cabessa@u-paris2.fr (J. Cabessa).

http://dx.doi.org/10.1016/j.jcss.2016.04.006

On the other hand, since the late 1940's, the issue of the computational and dynamical capabilities of neural models has also been approached from a very theoretical standpoint [34]. In this context, neural networks are considered as abstract computing systems and their computational power is investigated from a theoretical computer scientist perspective [33–36, 39–42,48]. As a consequence, the computational power of neural networks has been shown to be intimately related to the nature of their synaptic weights and activation functions, and able to range from the finite state automata level [33–35] up to Turing [41] and super-Turing capabilities [8,10,39,40]. More recently, the Turing and super-Turing capabilities of various models of recurrent neural networks have been extended to the contexts of alternative bio-inspired paradigms of computation, like reactive-system-based computation [11–13,16] (i.e., abstract devices working over infinite input streams [38,46]) or interactive computation [6,9,15,17] (i.e., infinite sequential exchange of information between the system and its environment [27,50,60]).

Based on these theoretical and experimental considerations, we initiated the theoretical study of the expressive power of recurrent neural networks from the perspective of their attractor dynamics [16]. We proved that Boolean recurrent neural networks provided with two different kinds of attractors are computationally equivalent to Muller automata, and hence, recognize precisely the so-called ω-regular neural languages. Consequently, the most refined topological classification of ω-languages [59] can be transposed from the automaton to the neural network context, and yield to some transfinite hierarchical classification of Boolean neural networks according to their attractor dynamics [11,12]. This classification induces a novel attractor-based measure of complexity for Boolean recurrent neural networks, which notably refers to the ability of the networks to perform more or less complicated classification tasks, via the manifestation of meaningful or spurious attractor dynamics [16].

The present paper pursues this precise research direction and constitutes an extended version of the two proceedings papers [7,18]. Here, we provide a characterization of the expressive powers of various models of deterministic and nondeterministic sigmoidal (rather than Boolean) first-order recurrent neural networks, in terms of their attractor dynamics. This attractor-based expressive power also notably refers to the ability of the networks to perform more or less complicated classification tasks of their input streams via the manifestation of meaningful or spurious attractor dynamics, and hence, via the manifestation of meaningful or spurious spatiotemporal patterns of discharge.

The paper is organized as follows. In Section 2, we present the mathematical notions required for the development of our theory. In Section 3, we recall some basic definitions and facts concerning deterministic and nondeterministic Muller Turing machines.

In Section 4, the various models of deterministic and nondeterministic first-order recurrent neural networks under consideration are presented. These models are all based on classical first-order recurrent neural network composed with Boolean input and output cells as well as sigmoidal hidden units, along the very lines of [8,10,40,41]. The sigmoidal hidden neurons introduce the source of nonlinearity which is so important to neural computation. The Boolean input and output cells carry out the exchange of discrete information between the network and the environment. When subjected to some infinite binary input stream, the Boolean output cells necessarily exhibit some attractor dynamics, which is assumed to be of two possible kinds, namely either meaningful or spurious. The neural ω-language of a network then corresponds to the set of all input streams which induce a meaningful attractor dynamics. The expressive power of the networks is then measured via the topological complexity of their underlying neural ω-languages.

Section 5 provides the results of the paper, which are summarized in Subsection 5.1, and in particular, in Fig. 2 and Table 2. In short, the deterministic static rational-weighted neural networks are computationally equivalent to the deterministic Muller Turing machines, and every other model of deterministic or nondeterministic static or evolving neural nets is strictly more expressive than the Muller Turing machine model.

Subsection 5.2 concerns the expressive power of the deterministic neural networks. We show that the static rational-weighted and real-weighted neural networks are computationally equivalent to and strictly more powerful than the deterministic Muller Turing machines, respectively. Moreover, the evolving neural nets are equivalent to the static real-weighted ones, irrespective of whether their synaptic weights are modeled by rational or real numbers. They recognize precisely the set of all $BC(\mathbf{\Pi}^0_2)$ neural ω-languages (the finite Boolean combinations of $\mathbf{\Pi}^0_2$ neural ω-languages).

Subsection 5.3 focuses on the expressive power of the nondeterministic neural networks. Here, we introduce a novel learning-based notion of nondeterminism which encompasses the classical one studied by Siegelmann and Sontag [40, 41]. More precisely, we consider a model of evolving neural networks, where each network might a priori follow various patterns of evolution for its synaptic weights. At the beginning of each computation, the network selects one possible evolving pattern – in a nondeterministic manner – and then sticks to it throughout its whole computational process. We prove that the two models of rational-weighted and real-weighted nondeterministic neural networks are computationally equivalent, and recognize precisely the set of all $\mathbf{\Sigma}^1_1$ neural ω-languages. They are therefore strictly more expressive than the nondeterministic Muller Turing machines.

Finally, Section 6 discusses the interpretation and relevance of the results and provides some general concluding remarks.

2. Preliminaries

A *topological space* is a pair $(S, \mathcal{T})$ where S is a set and $\mathcal{T}$ is a collection of subsets of S such that $\emptyset \in \mathcal{T}$, $S \in \mathcal{T}$, and $\mathcal{T}$ is closed under arbitrary unions and finite intersections. The collection $\mathcal{T}$ is called a *topology* on S, and its members are called *open sets*. Given some topological space $(S, \mathcal{T})$, the class of *Borel subsets* of S, denoted by $\mathbf{\Delta}^1_1$, is the σ-algebra generated by $\mathcal{T}$, i.e., the smallest collection of subsets of S containing all open sets and closed under countable union and complementation. For every ordinal $\alpha < \omega_1$ (where ω_1 is the first uncountable ordinal), one defines by transfinite induction the following *Borel classes*:

- $\mathbf{\Sigma}^0_1 = \{X \subseteq S : X \text{ is open}\}$,
- $\mathbf{\Pi}^0_\alpha = \{X \subseteq S : X^\complement \in \mathbf{\Sigma}^0_\alpha\}$,
- $\mathbf{\Sigma}^0_\alpha = \{X \subseteq S : X = \bigcup_{n\geq 0} X_n,\ X_n \in \mathbf{\Pi}^0_{\alpha_n},\ \alpha_n < \alpha,\ n \in \mathbb{N}\}$, for $\alpha > 1$,
- $\mathbf{\Delta}^0_\alpha = \mathbf{\Sigma}^0_\alpha \cap \mathbf{\Pi}^0_\alpha$.

The collection of all classes $\mathbf{\Sigma}^0_\alpha$, $\mathbf{\Pi}^0_\alpha$, and $\mathbf{\Delta}^0_\alpha$ provides a stratification of the whole class of Borel sets known as *the Borel hierarchy*. One has [31, Section 11.B]

$$\mathbf{\Delta}^1_1 = \bigcup_{\alpha<\omega_1} \mathbf{\Sigma}^0_\alpha = \bigcup_{\alpha<\omega_1} \mathbf{\Pi}^0_\alpha.$$

The *rank* of a Borel set $X \subseteq S$ is the smallest ordinal α such that $X \in \mathbf{\Sigma}^0_\alpha \cup \mathbf{\Pi}^0_\alpha \cup \mathbf{\Delta}^0_\alpha$, namely the minimal number of complementation and countable union operations that are needed in order to obtain X from an initial collection of open sets. It is commonly considered as a relevant measure of the topological complexity of Borel sets.

We recall that given two topological spaces $(S, \mathcal{T})$ and $(S', \mathcal{T}')$, a function $f : S \to S'$ is *continuous* if the preimage by f of any open set (i.e. $\mathbf{\Sigma}^0_1$-set) of S' is an open set (i.e. $\mathbf{\Sigma}^0_1$-set) of S. Consequently, the preimage by f of any $\mathbf{\Sigma}^0_\alpha$-set (resp. $\mathbf{\Pi}^0_\alpha$-set) is a $\mathbf{\Sigma}^0_\alpha$-set (resp. $\mathbf{\Pi}^0_\alpha$-set), for any ordinal $\alpha < \omega_1$. A function $f : S \to S'$ is of *Baire class 1* if the preimage by f of any $\mathbf{\Sigma}^0_1$-set of S' is a $\mathbf{\Sigma}^0_2$-set of S.

Besides, given any set A, we let A^*, A^+ and A^ω denote respectively the sets of finite sequences, non-empty finite sequences and infinite sequences of elements of A. For any $x \in A^* \cup A^\omega$, the *length* of x is denoted by $|x|$, the $(i+1)$-th element of x will be denoted by $x(i)$ for any $0 \leq i < |x|$, and the subsequence of the n-th first elements of x is denoted by $x[0{:}n]$, with the convention that $x[0{:}0] = \lambda$, the empty sequence. Hence, any $x \in A^+$ and $y \in A^\omega$ can be written as $x = x(0)x(1)\cdots x(|x|-1)$ and $y = y(0)y(1)y(2)\cdots$, respectively. The fact that x is a *prefix* (resp. *strict prefix*) of y will be denoted by $x \subseteq y$ (resp. $x \subsetneq y$). The concatenation of x and y is denoted $x \cdot y$ or simply xy, and for any $X \subseteq A^*$ and $Y \subseteq A^* \cup A^\omega$, one sets $X \cdot Y = \{z \in A^* \cup A^\omega : z = x \cdot y \text{ for some } x \in X \text{ and } y \in Y\}$. A set of the form $\{x\} \cdot A^\omega$ is generally denoted $x \cdot A^\omega$. Sometimes, a concatenated space of the form $A_0 \cdot A_1 \cdot \cdots$ will be naturally identified with the corresponding product space $A_0 \times A_1 \times \cdots$ via the identification $a_0 a_1 \cdots = (a_0, a_1, \ldots)$. Finally, a sequence of $A^* \cup A^\omega$ will be called a *word*, and a subset of A^ω is generally called an *ω-language*.

In the sequel, the spaces of N-dimensional Boolean, rational and real vectors will be denoted by $\mathbb{B}^N$, $\mathbb{Q}^N$ and $\mathbb{R}^N$, respectively. The space $(\mathbb{B}^N)^\omega$ is naturally assumed to be equipped with the product topology of the discrete topology on $\mathbb{B}^N$. Accordingly, the basic open sets are of the form $p \cdot (\mathbb{B}^N)^\omega$, for some $p \in (\mathbb{B}^N)^*$. The general open sets are countable unions of basic open sets. This space is Polish (i.e., separable and completely metrizable) [31]. Moreover, the spaces $(\mathbb{Q}^N)^\omega$ and $(\mathbb{R}^N)^\omega$ are assumed to be equipped with the product topologies of the usual topologies on $\mathbb{Q}^N$ and $\mathbb{R}^N$, respectively. Accordingly, the basic open sets are of the form $X_0 \cdot \ldots \cdot X_n \cdot (\mathbb{Q}^N)^\omega$ or $X_0 \cdot \ldots \cdot X_n \cdot (\mathbb{R}^N)^\omega$, for some $n \geq 0$, where each X_i is an open set of $\mathbb{Q}^N$ or $\mathbb{R}^N$ for their usual topologies, respectively. The general open sets are countable unions of basic open sets. These two spaces are also Polish [31].

In this paper, we will use the following two characterizations of *analytic sets* as first projections of either $\mathbf{\Pi}^0_2$-sets or general Borel sets [31]. First, an ω-language $L \subseteq (\mathbb{B}^N)^\omega$ is *analytic* iff there exists some $\mathbf{\Pi}^0_2$-set $X \subseteq (\mathbb{B}^N)^\omega \times \{0,1\}^\omega$ such that $L = \pi_1(X) = \{s \in (\mathbb{B}^N)^\omega : \exists\, e \in \{0,1\}^\omega \text{ s.t. } (s, e) \in X\}$ [31, Exercise 14.3]. This fact will be used in forthcoming Proposition 3. Equivalently, $L \subseteq (\mathbb{B}^N)^\omega$ is *analytic* iff there exists some Polish space E and some Borel set $X \subseteq (\mathbb{B}^N)^\omega \times E$ such that $L = \pi_1(X)$ [31, Exercise 14.3]. This fact will be used in forthcoming Proposition 4. The class of analytic sets, denoted by $\mathbf{\Sigma}^1_1$, strictly contains that of Borel sets, namely $\mathbf{\Delta}^1_1 \subsetneq \mathbf{\Sigma}^1_1$ [31, Theorem 14.2].

Finally, in the sequel, we will use the word *recursive* to denote a function, language or procedure that is computable by a Turing machine.

3. Muller Turing machines

The study of the behavior of reactive systems has led to the emergence of a theory of automata working on infinite objects [38,46].

In this context, a *Muller Turing machine* can be defined as a pair $(\mathcal{M}, \mathcal{T})$, where $\mathcal{M}$ is a classical multitape Turing machine whose input tape is associated with a read-only head that only moves to the right, and $\mathcal{T}$ is a finite collection of sets of states of $\mathcal{M}$ (i.e., $\mathcal{T} = \{T_1, \ldots, T_k\}$ and each T_i is a set of states of $\mathcal{M}$). At the beginning of the computation, an infinite input s (usually binary) is written on the input tape. A computation of $\mathcal{M}$ on s is defined in the classical way. An infinite sequence of successive states visited by $\mathcal{M}$ during the processing of s is called an infinite run of $\mathcal{M}$ on s, denoted by ρ_s. The set of states appearing infinitely often in ρ_s is denoted by $\inf(\rho_s)$

If $\mathcal{M}$ is a deterministic Turing machine, an infinite input stream s is said to be accepted by $(\mathcal{M}, \mathcal{T})$ if the unique infinite run ρ_s satisfies $\inf(\rho_s) \in \mathcal{T}$, or in other words, if the unique infinite run of $\mathcal{M}$ on s induces a set of states that are visited infinitely often which belongs to the collection $\mathcal{T}$; the infinite input s is said to be rejected otherwise. If $\mathcal{M}$ is nondeterministic, s is said to be accepted by $(\mathcal{M}, \mathcal{T})$ if there exists an infinite run ρ_s such that $\inf(\rho_s) \in \mathcal{T}$; s is rejected otherwise. The set of all words that are accepted by $(\mathcal{M}, \mathcal{T})$ is the *ω-language* recognized by $(\mathcal{M}, \mathcal{T})$.

Every ω-language recognized by some deterministic Muller Turing machine belongs to the topological class $BC(\mathbf{\Pi}^0_2)$ – the finite Boolean combinations of $\mathbf{\Pi}^0_2$-sets[1] [43, Corollary 3.4]. However, a simple cardinality argument shows that not all $BC(\mathbf{\Pi}^0_2)$-sets can be recognized by some deterministic Muller Turing machine: indeed, there are $\aleph_0$ Muller Turing machines and $2^{\aleph_0}$ sets in $BC(\mathbf{\Pi}^0_2)$. Besides, the nondeterministic Muller Turing machines are strictly more powerful than the deterministic ones. The ω-languages recognized by nondeterministic Muller Turing machines correspond precisely the class of *effectively analytic* ω-languages, denoted by Σ^1_1 (lightface) [43, Theorem 3.5]. We recall that the relation $\Sigma^1_1 \subsetneq \mathbf{\Sigma}^1_1$ holds [31].

The Muller acceptance condition is the most powerful one amongst those usually investigated in ω-automata theory (Büchi, Rabin, Streett, parity) [43, Corollaries 3.4, 3.5 and Theorem 3.5].

4. First-order recurrent neural networks

4.1. Generalities

We consider a general model of *first-order recurrent neural networks* consisting of three groups of neurons: a layer of "input" cells, which is connected to a set of recurrently interconnected "hidden" neurons, which is itself connected to a layer of "output" cells. At each time step, the activation values of each neuron is given by an affine combination of the other cells' activation values and inputs.

The sigmoidal hidden neurons introduce the biological source of nonlinearity which is so important to neural computation. They provide the possibility to surpass the capabilities of finite state automata, or even of Turing machines. The sigmoidal activation functions are particularly appropriate for the implementation of various learning algorithms. In neurobiology, they are usually considered as a representation of the rate of action potential firing in the cell. The Boolean input and output cells carry out the exchange of discrete information between the network and the environment. It will be noticed that, if some infinite input stream is supplied, the output cells necessarily enter into some attractor dynamics. The Boolean nature of the input and output cells provides the possibility to consider recurrent neural networks as computing systems working on discrete inputs and outputs streams, and consequently, to compare their computational capabilities to those of classical abstract machines, along the lines of [8,10,40,41]. The expressive power of the networks will be related to the attractor dynamics of their Boolean output cells.

4.2. Deterministic recurrent neural networks

A *deterministic (first-order) recurrent neural network* (simply denoted by DRNN) consists of a synchronous network of neurons related together in a general architecture. The network contains M Boolean input cells $(u_i)_{i=1}^M$, N sigmoidal hidden neurons $(x_i)_{i=1}^N$, and P Boolean output cells $(y_i)_{i=1}^P$. The dynamics of the network is computed as follows: given the activation values of the input and hidden neurons $(u_j)_{j=1}^M$ and $(x_j)_{j=1}^N$ at time t, the activation values of each sigmoidal hidden and Boolean output neuron x_i and y_i at time $t+1$ are updated by the following equations, respectively:

$$x_i(t+1) = \sigma\left(\sum_{j=1}^N a_{ij}(t)\cdot x_j(t) + \sum_{j=1}^M b_{ij}(t)\cdot u_j(t) + c_i(t)\right) \text{ for } i = 1, \ldots, N \tag{1}$$

$$y_i(t+1) = \theta\left(\sum_{j=1}^N a_{ij}(t)\cdot x_j(t) + \sum_{j=1}^M b_{ij}(t)\cdot u_j(t) + c_i(t)\right) \text{ for } i = 1, \ldots, P \tag{2}$$

where $a_{ij}(t)$, $b_{ij}(t)$, and $c_i(t)$ are the weighted synaptic connections and bias of the network at time t, and σ and θ are the linear sigmoidal[2] and the Heaviside step activation functions respectively defined by:

$$\sigma(x) = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{if } 0 \le x \le 1, \\ 1 & \text{if } x > 1 \end{cases} \quad \text{and} \quad \theta(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \ge 1. \end{cases}$$

We assume that the modification of the synaptic weights is constrained by the biological constitution of the synapses, i.e., that $a_{ij}(t), b_{ij}(t), c_i(t) \in [S, S']$ for all $t \ge 0$, $i = 1, \ldots, N$ and $j = 1, \ldots, M$, where $S, S' \in \mathbb{R}$, and $S \le 0 \le 1 \le S'$. In the

[1] $BC(\mathbf{\Pi}^0_2)$ is the collection of sets obtained by finite unions, intersections and complementations of $\mathbf{\Pi}^0_2$-sets.

[2] The linear sigmoidal activation function is a simple piecewise linear approximation of a general sigmoidal activation function. However, the results of this paper remain valid for any other kind of sigmoidal activation function satisfying the properties mentioned in [32, Section 4].

sequel, a synaptic weight w will be called *static* if $w(t) = C$, for all $t \geq 0$. It is *bi-valued evolving* if $w(t) \in \{0, 1\}$, for all $t \geq 0$, and it is *(general) evolving* if $w(t) \in [S, S']$, for all $t \geq 0$.

The dynamics of any DRNN $\mathcal{N}$ is therefore given by the function $f_{\mathcal{N}} : \mathbb{B}^M \times \mathbb{B}^N \to \mathbb{B}^N \times \mathbb{B}^P$ defined by

$$f_{\mathcal{N}}(\mathbf{u}(t), \mathbf{x}(t)) = (\mathbf{x}(t+1), \mathbf{y}(t+1)) \tag{3}$$

where the components of $\mathbf{x}(t+1)$ and $\mathbf{y}(t+1)$ are given by Equations (1) and (2), respectively.

Consider some DRNN $\mathcal{N}$ provided with M Boolean input cells, N sigmoidal hidden cells, and P Boolean output cells. For each time step $t \geq 0$, the Boolean vector

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

describing the activation values of the M input units of $\mathcal{N}$ at time t is the *input* submitted to $\mathcal{N}$ at time t. The pair

$$\langle \mathbf{x}(t), \mathbf{y}(t) \rangle \in [0, 1]^N \times \mathbb{B}^P$$

describing the activation values of the hidden and output cells at time t is the *state* of $\mathcal{N}$ at time t. The second element of this pair, namely $\mathbf{y}(t)$, is the *output state* of $\mathcal{N}$ at time t.

Assuming the initial state of the network to be $\langle \mathbf{x}(0), \mathbf{y}(0) \rangle = \langle \mathbf{0}, \mathbf{0} \rangle$, any infinite input stream

$$s = (\mathbf{u}(t))_{t \in \mathbb{N}} = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2) \cdots \in (\mathbb{B}^M)^\omega$$

induces via Equations (1) and (2) an infinite sequence of consecutive states

$$c_s = (\langle \mathbf{x}(t), \mathbf{y}(t) \rangle)_{t \in \mathbb{N}} = \langle \mathbf{x}(0), \mathbf{y}(0) \rangle \langle \mathbf{x}(1), \mathbf{y}(1) \rangle \cdots \in ([0, 1]^N \times \mathbb{B}^P)^\omega$$

called the *computation* of $\mathcal{N}$ induced by s. The corresponding infinite sequence of output states

$$c'_s = (\mathbf{y}(t))_{t \in \mathbb{N}} = \mathbf{y}(0)\mathbf{y}(1)\mathbf{y}(2) \cdots \in (\mathbb{B}^P)^\omega$$

is the *Boolean computation* of $\mathcal{N}$ induced by s.

Note that any DRNN $\mathcal{N}$ (with P Boolean output cells) can only have 2^P – i.e., finitely many – possible distinct output states. Consequently, for any infinite Boolean computation c'_s, there necessarily exists at least one output state that recurs infinitely often in c'_s. In fact, any Boolean computation c'_s necessarily consists of a finite prefix of output states followed by an infinite suffix of output states that repeat infinitely often – yet not necessarily in a periodic manner. The non-empty set of all the output states that repeat infinitely often in c'_s will be denoted by $\inf(c'_s)$. According to these considerations, a set of states of the form $\inf(c'_s)$ for some computation c'_s will be called an *attractor* for $\mathcal{N}$. A precise definition can be given as follows [16]:

Definition 1. Let $\mathcal{N}$ be some DRNN. A set $A = \{\mathbf{y_0}, \ldots, \mathbf{y}_k\} \subseteq \mathbb{B}^P$ is an *attractor* for $\mathcal{N}$ if there exists some infinite input stream s such that the corresponding Boolean computation c'_s satisfies $\inf(c'_s) = A$.

In other words, an attractor of $\mathcal{N}$ is a set of output states into which the computation of the network could become forever trapped – yet not necessarily in a periodic manner – for some infinite input stream s.

In this work, we suppose that the attractors can be of two distinct types, namely either *meaningful* or *spurious*. The type of each attractor could be determined by its neurophysiological significance with respect to measurable observations, e.g. associated with certain behaviors or sensory discriminations. The classification of these attractors into meaningful or spurious types is not the subject of this paper. For a referenced discussion about meaningful and spurious attractors in biological neural networks, see [16, Section "Neurophysiological Meaningfulness"]. Hence, from this point onwards, we assume any DRNN to be equipped with a corresponding classification of all of its attractors into meaningful and spurious types.

According to these considerations, given some DRNN $\mathcal{N}$, an infinite input stream $s \in (\mathbb{B}^M)^\omega$ of $\mathcal{N}$ is called *meaningful* if $\inf(c'_s)$ is a meaningful attractor, and it is called *spurious* if $\inf(c'_s)$ is a spurious attractor. The set of all meaningful input streams of $\mathcal{N}$ is called the *neural ω-language recognized by* $\mathcal{N}$ and is denoted by $L(\mathcal{N})$. A set $L \subseteq (\mathbb{B}^M)^\omega$ is said to be *recognizable* by some DRNN if there exists a network $\mathcal{N}$ such that $L(\mathcal{N}) = L$.

Finally, six different models of DRNNs will be considered according to the nature of their synaptic weights:

1. The *static rational DRNNs* (DRNN[$\mathbb{Q}$]s) refers to the class of all DRNNs whose every weights are static and modeled by rational values.
2. The *static real (or analog) DRNNs* (DRNN[$\mathbb{R}$]s) refers to the class of all DRNNs whose every weights are static and modeled by real values.
3. The *bi-valued evolving rational DRNNs* (Ev$_2$-DRNN[$\mathbb{Q}$]s) refers to the class of all DRNNs whose every evolving weights are bi-valued and every static weights are rational.
4. The *bi-valued evolving real DRNNs* (Ev$_2$-DRNN[$\mathbb{R}$]s) refers to the class of all DRNNs whose every evolving weights are bi-valued and every static weights are real.

Table 1
The six models of DRNNs according to the nature of their the synaptic weights.

	Static	Bi-valued Evolving	Evolving
$\mathbb{Q}$	DRNN[$\mathbb{Q}$]	Ev_2-DRNN[$\mathbb{Q}$]	Ev-DRNN[$\mathbb{Q}$]
$\mathbb{R}$	DRNN[$\mathbb{R}$]	Ev_2-DRNN[$\mathbb{R}$]	Ev-DRNN[$\mathbb{R}$]

5. The *(general) evolving rational DRNNs* (Ev-DRNN[$\mathbb{Q}$]s) refers to the class of all DRNNs whose every evolving and static weights are rational.
6. The *(general) evolving real DRNNs* (Ev-DRNN[$\mathbb{R}$]s) refers to the class of all DRNNs whose every evolving and static weights are real.

These models are summarized in Table 1. Note that the bi-valued evolving and the evolving models capture the key feature of *learning*, i.e., the possibility of an adaptation of the synaptic weights. Note also that in the above definitions, bi-valued or general evolving DRNNs are not necessarily required to contain at least one evolving weight. Moreover, by definition, any rational weight is obviously a specific real weight, and any bi-valued evolving weight is a specific general evolving weight. Hence, the following relationship between the expressive powers of the six classes of DRNNs holds (where "$\mathcal{C}_1 \leq \mathcal{C}_2$" signifies "the class $\mathcal{C}_1$ is less expressive than or equally expressive to the class $\mathcal{C}_2$"). This relationship is represented by directed arrows in Fig. 2.

$$\begin{array}{ccccc} \text{DRNN}[\mathbb{Q}]\text{s} & \leq & \text{Ev}_2\text{-DRNN}[\mathbb{Q}]\text{s} & \leq & \text{Ev-DRNN}[\mathbb{Q}]\text{s} \\ \text{\rotatebox{90}{$\geq$}} & & \text{\rotatebox{90}{$\geq$}} & & \text{\rotatebox{90}{$\geq$}} \\ \text{DRNN}[\mathbb{R}]\text{s} & \leq & \text{Ev}_2\text{-DRNN}[\mathbb{R}]\text{s} & \leq & \text{Ev-DRNN}[\mathbb{R}]\text{s} \end{array}$$

4.3. Nondeterministic recurrent neural networks

In their seminal work, Siegelmann and Sontag introduced a notion of nondeterministic recurrent neural network, where the nondeterminism is expressed by means of some extra Boolean stream – a so-called "guess stream" – transmitted to the network [40,41]. Formally, in their framework, a *nondeterministic recurrent neural network* $\mathcal{N}$ consists of a classical recurrent neural networks provided with an additional Boolean "guess cell". Such a network $\mathcal{N}$ would *accept* the input stream s if there exists some Boolean guess stream $g \in \mathbb{B}^\omega$ such that the network $\mathcal{N}$, when receiving input s and guess g, provides an accepting output. It would *reject* the input stream s if for all Boolean guess stream $g \in \mathbb{B}^\omega$, the network $\mathcal{N}$, when receiving input s and guess g, provides a rejecting output. The set of all input streams accepted by the network is the language *recognized* by the network, denoted by $L(\mathcal{N})$.

This concept of nondeterminism can be directly transposed to the current computational context. However, we chose to introduce some alternative notion of nondeterminism which is, on the one hand, more general than the notion of Siegelmann and Sontag (cf. Remark 1 below), and on the other hand, closer to the biological framework, for it is directly related to the key concept of *learning*. In the present case, the nondeterminism is expressed as a set of possible patterns of evolution that the synaptic connections of the network might follow over the successive time steps. At the beginning of a computation, the network selects one such possible evolving pattern – in a nondeterministic manner – and then sticks to it throughout its whole computational process.

Formally, a *nondeterministic (first-order) recurrent neural network* (denoted by NRNN) consists of a pair $(\mathcal{N}, E)$, where $\mathcal{N}$ is a recurrent neural networks of the form described in Section 4.2, namely with a dynamics governed by Equations (1) and (2), and $E \subseteq ([S, S']^K)^\omega$ is a set of infinite sequences of K-dimensional vectors describing the set of all possible evolutions for the K evolving synaptic connections of $\mathcal{N}$ (K is assumed to be smaller than the total number of synaptic connections of $\mathcal{N}$). According to this definition, any DRNN $\mathcal{N}$ is a particular case of a NRNN $(\mathcal{N}, E)$ where the evolution set E is reduced to a singleton.

Given some NRNN $(\mathcal{N}, E)$, every element

$$e = \mathbf{e}(0)\mathbf{e}(1)\mathbf{e}(2) \cdots \in E$$

is called a possible *evolution* for $(\mathcal{N}, E)$, where each vector $\mathbf{e}(t)$ describes the values of the K evolving synaptic weights of $\mathcal{N}$ at time step t. The set E is called the *evolution set* of $(\mathcal{N}, E)$. Assuming the initial state of the network to be $\langle \mathbf{x}(0), \mathbf{y}(0) \rangle = \langle \mathbf{0}, \mathbf{0} \rangle$, any infinite input stream $s = (\mathbf{u}(t))_{t \in \mathbb{N}} \in (\mathbb{B}^M)^\omega$ and evolution $e = (\mathbf{e}(t))_{t \in \mathbb{N}} \in E$ induce via Equations (1) and (2) an infinite sequence of consecutive states and output states

$$c_{(s,e)} = (\langle \mathbf{x}(t), \mathbf{y}(t) \rangle)_{t \in \mathbb{N}} \in ([0,1]^N \times \mathbb{B}^P)^\omega \text{ and } c'_{(s,e)} = (\mathbf{y}(t))_{t \in \mathbb{N}} \in (\mathbb{B}^P)^\omega$$

called the *computation* and *Boolean computation* of $(\mathcal{N}, E)$ induced by (s, e), respectively. Furthermore, the Definition 1 of an *attractor* remains unchanged in this case, and once again, we assume that any NRNN is equipped with a corresponding classification of all of its attractors into meaningful and spurious types.

Accordingly, an infinite input stream $s \in (\mathbb{B}^M)^\omega$ is called *meaningful* if *there exists* some evolution $e \in E$ such that $\inf(c'_{(s,e)})$ is a meaningful attractor, and it is called *spurious* otherwise, i.e., if *for all* evolution $e \in E$, the set $\inf(c'_{(s,e)})$ is a spurious attractor. The set of all meaningful input streams is called the *neural ω-language recognized by* $(\mathcal{N}, E)$ and is denoted by $L((\mathcal{N}, E))$. A set $L \subseteq (\mathbb{B}^M)^\omega$ is said to be *recognizable* by some nondeterministic recurrent neural network if there exists a NRNN $(\mathcal{N}, E)$ such that $L((\mathcal{N}, E)) = L$.

Two different models of NRNNs are considered according to the nature of their synaptic weights.

1. The *rational NRNNs* (NRNN[$\mathbb{Q}$]s) is the class of all NRNNs whose every weights are rational.
2. The *real (or analog) NRNNs* (NRNN[$\mathbb{R}$]s) is the class of all NRNNs whose every weights are real.

Since any rational weight is also a real weight, the following relationship between the expressive powers of the two models of NRNNs holds. It is represented by a directed arrow in Fig. 2.

$$\text{NRNN}[\mathbb{Q}]\text{s} \leq \text{NRNN}[\mathbb{R}]\text{s}$$

Finally, depending on whether $(\mathcal{N}, E)$ is either a NRNN[$\mathbb{Q}$] or a NRNN[$\mathbb{R}$], one has either $E \subseteq ((\mathbb{Q} \cap [S, S'])^K)^\omega$ or $E \subseteq ((\mathbb{R} \cap [S, S'])^K)^\omega = ([S, S']^K)^\omega$. Accordingly, we assume from now on that $(\mathbb{Q} \cap [S, S'])^K$ and $(\mathbb{R} \cap [S, S'])^K = [S, S']^K$ are equipped with the subspace topologies[3] of $\mathbb{Q}^K$ and $\mathbb{R}^K$, and that $((\mathbb{Q} \cap [S, S'])^K)^\omega$ and $((\mathbb{R} \cap [S, S'])^K)^\omega$ are equipped with the product topologies of these subspace topologies, respectively. These two spaces $((\mathbb{Q} \cap [S, S'])^K)^\omega$ and $((\mathbb{R} \cap [S, S'])^K)^\omega = ([S, S']^K)^\omega$, as a product of Polish spaces, are therefore Polish. We further assume that the evolution set E is a closed subset of these Polish subspaces, and hence is also Polish [31].[4]

Remark 1. Note that the "classical nondeterminism" a la Siegelmann and Sontag (described at the beginning of this section) is a particular case of our notion of nondeterminism. More precisely, for every "classical" nondeterministic version of some network $\mathcal{N}$ of Section 4.2, there exists some NRNN $(\tilde{\mathcal{N}}, E)$ of the kind described above such that $L(\mathcal{N}) = L((\tilde{\mathcal{N}}, E))$. The argument is the following. Suppose that $\mathcal{N}$ is an evolving real network (as described in Section 4.2), yet being considered in a "classical" nondeterministic way, i.e., equipped with an additional Boolean guess cell g as well as with the corresponding nondeterministic acceptance condition. Suppose also that $\mathcal{N}$ contains K evolving weights whose evolutions are given by the infinite sequence $e = \mathbf{e}(0)\mathbf{e}(1)\mathbf{e}(2) \cdots \in ((\mathbb{R} \cap [S, S'])^K)^\omega$. We define $\tilde{\mathcal{N}}$ as being the same network as $\mathcal{N}$, but with the guess cell g being considered as a hidden cell x with an associated evolving bias $c(t)$. We further define the closed evolution set of $\tilde{\mathcal{N}}$ associated with its $K+1$ evolving weights (the K same ones as $\mathcal{N}$ plus $c(t)$) as $E = \prod_{t \in \mathbb{N}} (\{\mathbf{e}(t)\} \times \{0, 1\}) \subseteq ((\mathbb{R} \cap [S, S'])^{K+1})^\omega$. Accordingly, $(\tilde{\mathcal{N}}, E)$ is a NRNN[$\mathbb{R}$] as described above. The accepting conditions of $\mathcal{N}$ and $(\tilde{\mathcal{N}}, E)$ ensure that $L(\mathcal{N}) = L((\tilde{\mathcal{N}}, E))$. Moreover, since any static rational, static real, bi-valued evolving rational, bi-valued evolving real, or general evolving rational network is a particular evolving real network (cf. Section 4.2), the argument can be generalized to these five other cases.

The current notion of nondeterminism actually refers to some learning paradigm that the networks might select in a nondeterministic manner at the beginning of a computational process. Accordingly, it would rather correspond to some nondeterministic learning rather than some nondeterministic computation. In this sense, it is in our opinion closer to the biological framework. Notably, this nondeterministic learning encompasses the notion of nondeterministic computation a la Siegelmann and Sontag.

4.4. Attractors and spatiotemporal patterns of discharge

Various experimental studies suggest that specific attractor dynamics [20,21,56] as well as spatiotemporal pattern of discharges, i.e., ordered and precise interspike interval relationships [2,49,51,53,54,57], are likely to be significantly involved in the processing and coding of information in the brain. Moreover, the association between attractor dynamics and spatiotemporal patterns has been demonstrated in nonlinear dynamical systems [3,4] and in simulations of large scale neuronal networks [28,29]. Spatiotemporal patterns are therefore assumed to be the witnesses of an underlying attractor dynamics – which itself would be a key feature of neural coding.

In our model, the *periodic attractor dynamics* of the neural networks are the precise phenomena that underly the arising of *spatiotemporal patterns of discharges* of the Boolean output cells. To illustrate this, suppose that some deterministic or nondeterministic neural network contains the three Boolean output cells y_0, y_1, y_2, and that some infinite input stream s induces a corresponding Boolean computation of the form

$$c'_s = \mathbf{y}(0) \cdots \mathbf{y}(n) \left[\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right]^\omega$$

[3] Given a topological space $(S, \mathcal{T})$ and a subset X of S, the subspace topology on X is defined as $\mathcal{T}_X = \{X \cap U : U \in \mathcal{T}\}$.

[4] The forthcoming results remain valid with E taken as $\mathbf{\Pi}^0_2$.

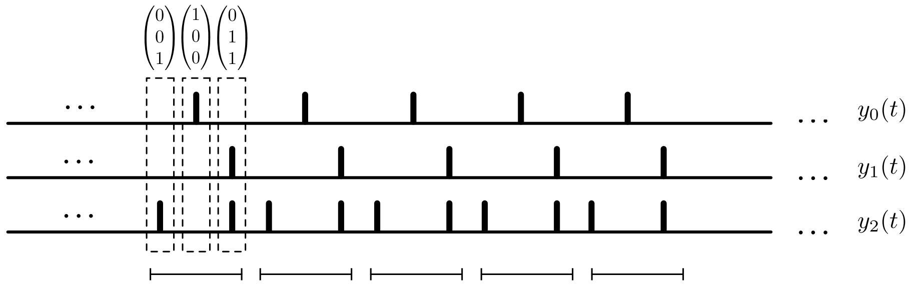

Fig. 1. The spatiotemporal pattern corresponding to periodic suffix of the Boolean computation c'_s. This spatiotemporal pattern corresponds to the periodic occurrences of the output states of the attractor $\inf(c'_s) = \{(0,0,1)^T, (1,0,0)^T, (0,1,1)^T\}$. The lines under the spike raster plots indicate the successive occurrences of the pattern.

Table 2
Expressive power of the eight models of RNNs.

	Deterministic			Nondeterministic
	Static	Bi-valued Evolving	Evolving	Evolving (by def.)
$\mathbb{Q}$	DRNN[$\mathbb{Q}$]s $\in BC(\mathbf{\Pi}^0_2)$	Ev$_2$-DRNN[$\mathbb{Q}$]s $= BC(\mathbf{\Pi}^0_2)$	Ev-DRNN[$\mathbb{Q}$]s $= BC(\mathbf{\Pi}^0_2)$	NRNN[$\mathbb{Q}$]s $= \mathbf{\Sigma}^1_1$
$\mathbb{R}$	DRNN[$\mathbb{R}$]s $= BC(\mathbf{\Pi}^0_2)$	Ev$_2$-DRNN[$\mathbb{R}$]s $= BC(\mathbf{\Pi}^0_2)$	Ev-DRNN[$\mathbb{R}$]s $= BC(\mathbf{\Pi}^0_2)$	NRNN[$\mathbb{R}$]s $= \mathbf{\Sigma}^1_1$

For this Boolean computation, the corresponding attractor is

$$\inf(c'_s) = \{(0,0,1)^T, (1,0,0)^T, (0,1,1)^T\}$$

and it is visited in a periodic way. The periodic visit of this attractor by the Boolean computation c'_s corresponds precisely to the spatiotemporal pattern illustrated in Fig. 1.

5. Expressive power of recurrent neural networks

5.1. Summary of the results

We provide a theoretical characterization of the expressive powers of the six models of DRNNs and the two models of NRNNs. These results (Theorems 1, 2 and 3) are summarized in Table 2 and Fig. 2.

5.2. The deterministic case

First, we show that the neural ω-languages recognized by DRNN[$\mathbb{Q}$]s are equivalent to those recognized by deterministic Muller Turing machines, and thus belong to the class $BC(\mathbf{\Pi}^0_2)$ of $(\mathbb{B}^M)^\omega$ (Theorem 1). Next, we prove that the five other models of Ev$_2$-DRNN[$\mathbb{Q}$]s, Ev-DRNN[$\mathbb{Q}$]s, DRNN[$\mathbb{R}$]s, Ev$_2$-DRNN[$\mathbb{R}$]s, and Ev-DRNN[$\mathbb{R}$]s are equivalent to each other, recognize the whole class of $BC(\mathbf{\Pi}^0_2)$ of $(\mathbb{B}^M)^\omega$, and therefore, are all strictly more powerful than the deterministic Muller Turing machines (Theorem 2).

Theorem 1. *Let $L \subseteq (\mathbb{B}^M)^\omega$ be some ω-language. Then L is recognizable by some DRNN[$\mathbb{Q}$] if and only if L is recognizable by some deterministic Muller TM. In particular, if L is recognizable by some DRNN[$\mathbb{Q}$], then $L \in BC(\mathbf{\Pi}^0_2)$.*

Proof. Let $\mathcal{N}$ be some DRNN[$\mathbb{Q}$] recognizing the neural ω-language $L(\mathcal{N})$. Since the synaptic weights of $\mathcal{N}$ are rational and remain constant over time, Equations (1) and (2) are recursive, and hence, the function $f_\mathcal{N}$ described in Equation (3) is also clearly recursive. Consequently, there exists some TM $\mathcal{M}$ with $N+P$ work tapes which can simulate the behavior of $\mathcal{N}$ by writing on its tapes the successive rational and Boolean activations values of the N and P hidden and output cells of $\mathcal{N}$, respectively. We next provide $\mathcal{M}$ with 2^P additional designated states $q_1, \ldots, q_{2^P}$, and we modify its program in such a way that, after each simulation step, $\mathcal{M}$ enters state q_i iff $\mathcal{N}$ is in the i-th output state $\mathbf{b}_i \in \mathbb{B}^P$, according to the lexicographic order. In this way, each infinite input stream $s \in (\mathbb{B}^M)^\omega$ induces on the one side, in the network $\mathcal{N}$, a Boolean computation c'_s with an associated attractor $\inf(c'_s) \subseteq \mathbb{B}^P$, and, on the other side, in the machine $\mathcal{M}$, an infinite run ρ_s with an associated set of sates that are visited infinitely often $\inf(\rho_s)$ of the form $\inf(\rho_s) = Q \cup Q'$, with $Q' \subseteq \{q_1, \ldots, q_{2^P}\}$ and $Q' \neq \emptyset$. By construction, for any infinite input streams $s, s' \in (\mathbb{B}^M)^\omega$, the relation $\inf(c'_s) \neq \inf(c'_{s'}) \Rightarrow \inf(\rho_s) \neq \inf(\rho_{s'})$ holds. We can thus

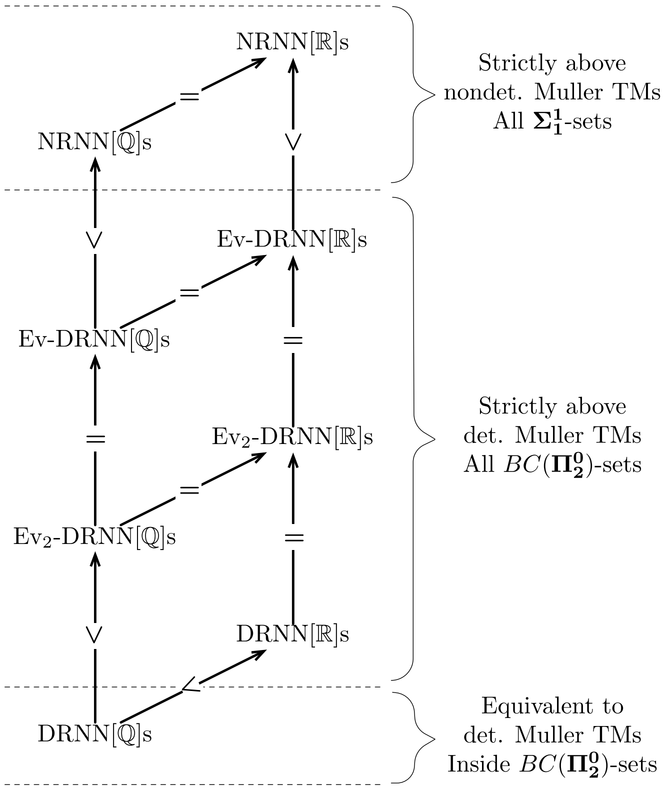

Fig. 2. Relationships between the expressive powers of the eight models of first-order RNNs. There is a directed arrow from one model to the other if the former is less expressive than or equally expressive to the latter. The relation represented by these arrows is clearly transitive. These relations are established in Subsections 4.2 and 4.3. In the sequel, we show that DRNN[ℚ]s are computationally equivalent to deterministic Muller Turing machines (Theorem 1). The five other models of Ev$_2$-DRNN[ℚ]s, Ev-DRNN[ℚ]s, DRNN[ℝ]s, Ev$_2$-DRNN[ℝ]s, Ev-DRNN[ℝ]s are all equivalent to each other, and strictly more powerful than deterministic Muller Turing machines (Theorem 2). Furthermore, the two models of NRNN[ℚ]s and NRNN[ℝ]s are computationally equivalent to each other, and strictly more powerful than nondeterministic Muller Turing machines as well as than all kinds of deterministic neural networks (Theorem 3). To illustrate these results, the arrows are labeled by a "$<$" or an "$=$" to designate if there is a strict inequality or an equality between the expressive powers of the respective models.

define the following Muller table of $\mathcal{M}$, namely $\mathcal{T} = \{\inf(\rho_s) : \inf(c'_s)$ is a meaningful attractor for $\mathcal{N}$, for any $s \in (\mathbb{B}^M)^\omega\}$.[5] According to this construction, one has $s \in L(\mathcal{N})$ iff $\inf(c'_s)$ is a meaningful attractor iff $\inf(\rho_s) \in \mathcal{T}$ iff $s \in L(\mathcal{M})$. Therefore, $L(\mathcal{N}) = L(\mathcal{M})$, showing that $L(\mathcal{N})$ is recognized by the deterministic Muller TM $\mathcal{M}$.

Conversely, let $\mathcal{M}$ be some deterministic Muller TM with the table $\mathcal{T} = \{T_1, \ldots, T_k\}$ and associated ω-language $L(\mathcal{M})$. By the construction given in [41], there exists some (static) rational-weighted RNN $\mathcal{N}$ which can simulate the behavior of $\mathcal{M}$. Now, if $\mathcal{M}$ contains n states $q_1, \ldots, q_n$, we provide $\mathcal{N}$ with P additional Boolean output cells $y_1, \ldots, y_P$, with P satisfying $2^P \geq n$, and we update the simulation process such that, during the processing of the input stream, the machine $\mathcal{M}$ visits the state q_k iff the network $\mathcal{N}$ activates the k-th output state $\mathbf{b}_k$, according to the lexicographic order, for $k = 1, \ldots, n$. Next, for each element $T_i = \{q_{i_1}, \ldots, q_{i_{k(i)}}\}$ of the Muller table $\mathcal{T}$ of $\mathcal{M}$, we set the meaningful attractor $A_i = \{\mathbf{b}_{i_1}, \ldots, \mathbf{b}_{i_{k(i)}}\}$ in the network $\mathcal{N}$. All other possible attractors of $\mathcal{N}$ are considered to be spurious. In this way, for any infinite input stream $s \in (\mathbb{B}^M)^\omega$, the infinite run ρ_s of $\mathcal{M}$ satisfies $\inf(\rho_s) \in \mathcal{T}$ iff the Boolean computation c'_s of $\mathcal{N}$ satisfies that $\inf(c'_s)$ is a meaningful attractor. In other words, $s \in L(\mathcal{M})$ iff $s \in L(\mathcal{N})$. Therefore, $L(\mathcal{M}) = L(\mathcal{N})$, showing that $L(\mathcal{M})$ is recognized by the DRNN[ℚ] $\mathcal{N}$.

The second part of the Theorem comes from the previous equivalence and the fact that any ω-language recognized by some deterministic Muller TM is in $BC(\boldsymbol{\Pi}_2^0)$ [38]. □

Theorem 2. *The five models of Ev$_2$-DRNN[ℚ]s, Ev-DRNN[ℚ]s, DRNN[ℝ]s, Ev$_2$-DRNN[ℝ]s, and Ev-DRNN[ℝ]s are equivalent to each other: they recognize the class of $BC(\boldsymbol{\Pi}_2^0)$ neural ω-languages. In other words, for any ω-language $L \subseteq (\mathbb{B}^M)^\omega$, the following conditions are equivalent:*

[5] Note that the relation $\inf(c'_s) \neq \inf(c'_{s'}) \Rightarrow \inf(\rho_s) \neq \inf(\rho_{s'})$ ensures that the table $\mathcal{T}$ is well defined, since it is impossible to have a situation where $\inf(c'_s)$ is meaningful, $\inf(c'_{s'})$ is spurious, and $\inf(\rho_s) = \inf(\rho_{s'})$, which would mean that $\inf(\rho_s) \in \mathcal{T}$, $\inf(\rho_{s'}) \notin \mathcal{T}$.

1. $L \in BC(\mathbf{\Pi}_2^0)$
2. *L is recognizable by some* Ev_2*-DRNN[*$\mathbb{Q}$*]*
3. *L is recognizable by some Ev-DRNN[*$\mathbb{Q}$*]*
4. *L is recognizable by some DRNN[*$\mathbb{R}$*]*
5. *L is recognizable by some* Ev_2*-DRNN[*$\mathbb{R}$*]*
6. *L is recognizable by some Ev-DRNN[*$\mathbb{R}$*]*

Proof. The proof is achieved via the two forthcoming Propositions 1 and 2.

First, let $L \subseteq (\mathbb{B}^M)^\omega$ such that $L \in BC(\mathbf{\Pi}_2^0)$. By Proposition 1, L is recognized by some Ev_2-DRNN[$\mathbb{Q}$] and by some DRNN[$\mathbb{R}$]. By definition, L is also recognizable by some Ev-DRNN[$\mathbb{Q}$], Ev_2-DRNN[$\mathbb{R}$], and Ev-DRNN[$\mathbb{R}$] (cf. arrows in Fig. 2). This proves the five implications from Point (1) to Points (2), (3), (4), (5), and (6).

Conversely, assume that L is recognizable by some Ev_2-DRNN[$\mathbb{Q}$], some Ev-DRNN[$\mathbb{Q}$], some DRNN[$\mathbb{R}$], or some Ev_2-DRNN[$\mathbb{R}$]. By definition, L is also recognizable by some Ev-DRNN[$\mathbb{R}$] (cf. arrows in Fig. 2). By Proposition 2, $L \in BC(\mathbf{\Pi}_2^0)$. This proves the five other implications from Points (2), (3), (4), (5), and (6) to Point (1). □

We now proceed to the proofs of Propositions 1 and 2.

Proposition 1. *Let* $L \subseteq (\mathbb{B}^M)^\omega$*. If* $L \in BC(\mathbf{\Pi}_2^0)$*, then L is recognizable by some DRNN[*$\mathbb{R}$*] and by some* Ev_2*-DRNN[*$\mathbb{Q}$*].*

Proof. First of all, let $L \subseteq (\mathbb{B}^M)^\omega$ such that $L \in \mathbf{\Pi}_2^0$. We will consider the case of $L \in BC(\mathbf{\Pi}_2^0)$ afterwards. Then L can be written as

$$L = \bigcap_{i \geq 0} \bigcup_{j \geq 0} p_{i,j} \cdot (\mathbb{B}^M)^\omega$$

where each $p_{i,j} \in (\mathbb{B}^M)^*$. Hence, a given infinite input $s \in (\mathbb{B}^M)^\omega$ belongs to L iff for all index $i \geq 0$ there exists an index $j \geq 0$ such that $s \in p_{i,j} \cdot (\mathbb{B}^M)^\omega$, or equivalently, iff for all $i \geq 0$ there exists $j \geq 0$ such that $p_{i,j} \subsetneq s$. Besides, as described in details in [13], one can show that the infinite sequence $(p_{i,j})_{i,j \in \mathbb{N}}$ can be encoded into some single real number r_L such that, for any pair of indices $(i, j) \in \mathbb{N} \times \mathbb{N}$, the decoding procedure of $(r_L, i, j) \mapsto p_{i,j}$ is actually recursive.

According to these considerations, the problem of determining whether some input $s \in (\mathbb{B}^M)^\omega$ supplied step by step belongs to L or not can be decided in infinite time by the Algorithm 1 given below. This algorithm consists of two subroutines performed in parallel. It uses the designated real number r_L (on line 12), and it is designed in such a precise way that, on every input $s \in (\mathbb{B}^M)^\omega$, it returns infinitely many 1's iff $s \in \bigcap_{i \geq 0} \bigcup_{j \geq 0} p_{i,j} \cdot (\mathbb{B}^M)^\omega = L$. Moreover, note that if the designated real number r_L is provided in advance, then every step of Algorithm 1 is actually recursive.

Algorithm 1 Procedure which uses the designated real number r_L.

Require: Input $s = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2) \cdots \in (\mathbb{B}^M)^\omega$ supplied step by step at successive time steps $t = 0, 1, 2, \ldots$

1: **SUBROUTINE 1**
2: $c \leftarrow 0$ // c counts the number of letters of s
3: **for all** time step $t \geq 0$ **do**
4: store each incoming Boolean vector $\mathbf{u}(t) \in \mathbb{B}^M$
5: $c \leftarrow c + 1$
6: **end for**
7: **END SUBROUTINE 1**

8: **SUBROUTINE 2**
9: $i \leftarrow 0$, $j \leftarrow 0$
10: **loop**
11: wait until $c \geq |p_{i,j}|$ // wait until $length(s) \geq length(p_{i,j})$
12: decode $p_{i,j}$ from r_L // recursive procedure if r_L is given in advance
13: **if** $p_{i,j} \subseteq s[0{:}c]$ **then** // $s \in p_{i,j} \cdot (\mathbb{B}^M)^\omega$
14: **return** 1 // $\exists$ j s.t. $s \in p_{i,j} \cdot (\mathbb{B}^M)^\omega$
15: $i \leftarrow i + 1$, $j \leftarrow 0$ // test if $s \in p_{i+1,0} \cdot (\mathbb{B}^M)^\omega$
16: **else** // $s \notin p_{i,j} \cdot (\mathbb{B}^M)^\omega$
17: **return** 0 // $\neg\exists$ $j' \leq j$ s.t. $s \in p_{i,j} \cdot (\mathbb{B}^M)^\omega$
18: $i \leftarrow i$, $j \leftarrow j + 1$ // test if $s \in p_{i,j+1} \cdot (\mathbb{B}^M)^\omega$
19: **end if**
20: **end loop**
21: **END SUBROUTINE 2**

Consequently, according to the real time computational equivalence between rational-weighted RNNs and TMs [41], there exists some RNN[$\mathbb{Q}$] $\mathcal{N}_1$ such that, if the real number r_L is given in advance as the activation value of one of its neuron x, then $\mathcal{N}_1$ is actually capable of simulating the behavior of Algorithm 1. In particular, to perform line 4, one uses M distinct cells in order to store the M Boolean components of $\mathbf{u}(t)$ (see [41] for further details). Consequently, if one adds to x a background synaptic connection of real intensity r_L, one obtains a RNN[$\mathbb{R}$] $\mathcal{N}_2$ capable of simulating Algorithm 1. Hence, if one provides $\mathcal{N}_2$ with an additional Boolean output cell y which is designed to take value 1 iff Algorithm 1 returns a 1, one obtains a DRNN[$\mathbb{R}$] $\mathcal{N}$ such that, on every input $s \in (\mathbb{B}^M)^\omega$, the Boolean output cell y will produce infinitely many 1's iff Algorithm 1 will return infinitely many 1's, namely iff $s \in \bigcap_{i\geq 0}\bigcup_{j\geq 0} p_{i,j} \cdot (\mathbb{B}^M)^\omega = L$. Consequently, by taking $\{(1)\}$ and $\{(0),(1)\}$ as the two meaningful attractors of $\mathcal{N}$, one has $L(\mathcal{N}) = L$, meaning that L is recognized by some DRNN[$\mathbb{R}$].

We now modify the proof in order to capture the case of an Ev$_2$-DRNN[$\mathbb{Q}$]. In this context, one can show that the infinite sequence $(p_{i,j})_{i,j\in\mathbb{N}}$ can be encoded into some infinite word $w_L \in \{0,1\}^\omega$ such that, for any pair of indices $(i,j) \in \mathbb{N}\times\mathbb{N}$, the decoding procedure of $(w_L, i, j) \mapsto p_{i,j}$ is actually recursive. According to these considerations, we modify Algorithm 1 by assuming that it receives the designated infinite word w_L bit by bit instead of having the designated real number r_L be provided in advance. One then replaces lines 11 and 12 by the following two ones:

11: wait until $p_{i,j}$ has been encoded in w_L and until $c \geq |p_{i,j}|$
12: decode $p_{i,j}$ from w_L

Algorithm 1 can be performed by some Ev$_2$-DRNN[$\mathbb{Q}$]. Indeed, in the DRNN[$\mathbb{R}$] $\mathcal{N}$ described above, one replaces the static background activity of neuron x of real intensity r_L by an evolving background activity of intensities $w_L = w_L(0)w_L(1)w_L(2)\cdots \in \{0,1\}^\omega$. In this way, one obtains a network $\mathcal{N}_1'$ whose all static synaptic weights are rational and whose only evolving synaptic weight is bi-valued. We next slightly modify this networks in order to perform correctly the recursive updated lines 11 and 12 of Algorithm 1. One obtains an Ev$_2$-DRNN[$\mathbb{Q}$] $\mathcal{N}'$ such that, on every input $s \in (\mathbb{B}^M)^\omega$, the Boolean output cell y of $\mathcal{N}'$ will produce infinitely many 1's iff the updated Algorithm 1 will return infinitely many 1's, namely iff $s \in \bigcap_{i\geq 0}\bigcup_{j\geq 0} p_{i,j} \cdot (\mathbb{B}^M)^\omega = L$. Therefore, by taking $\{(1)\}$ and $\{(0),(1)\}$ as the two sole meaningful attractors of $\mathcal{N}'$, one obtains $L(\mathcal{N}') = L$, meaning that L is recognized by some Ev$_2$-DRNN[$\mathbb{Q}$].

This concludes the proof for the case of $L \in \mathbf{\Pi}^0_2$. We finally extend the proof for the case of $L \in BC(\mathbf{\Pi}^0_2)$. We thus show that any finite union and any complementation of a $\mathbf{\Pi}^0_2$-set can also be recognized by some DRNN[$\mathbb{R}$] and by some Ev$_2$-DRNN[$\mathbb{Q}$].

First, let $L = L_1 \cup L_2$ such that $L_i \in \mathbf{\Pi}^0_2$, for $i = 1,2$. By the previous arguments, there exist two DRNN[$\mathbb{R}$]s (or two Ev$_2$-DRNN[$\mathbb{Q}$]s) $\mathcal{N}_1$ and $\mathcal{N}_2$ which recognize L_1 and L_2, respectively. By suitably merging $\mathcal{N}_1$ and $\mathcal{N}_2$ into some new network $\mathcal{N}$, and by setting as meaningful attractors of $\mathcal{N}$ all those involving at least one output state for which at least one of the two Boolean output cells is spiking, one obtains a DRNN[$\mathbb{R}$] (or an Ev$_2$-DRNN[$\mathbb{Q}$]) $\mathcal{N}$ that recognizes $L_1 \cup L_2$. In other words, one has $L(\mathcal{N}) = L_1 \cup L_2 = L$.

Secondly, let $L \in \mathbf{\Sigma}^0_2$. Then by definition, $L^\complement \in \mathbf{\Pi}^0_2$. By the previous arguments, there exists a DRNN[$\mathbb{R}$] (or an Ev$_2$-DRNN[$\mathbb{Q}$]) $\mathcal{N}$ which recognizes $L^\complement$ via some relevant simulation of Algorithm 1. We now slightly update $\mathcal{N}$ in a way that its only Boolean output cell y takes value 1 iff Algorithm 1 returns a 0 (instead of 1). One thus obtains a DRNN[$\mathbb{R}$] (or an Ev$_2$-DRNN[$\mathbb{Q}$]) $\mathcal{N}'$ such that, on every input $s \in (\mathbb{B}^M)^\omega$, the Boolean output cell y of $\mathcal{N}'$ will produce infinitely many 1's as well as only finitely many 0's iff Algorithm 1 will return infinitely many 0's and finitely many 1's, i.e. iff $s \notin L^\complement$, i.e. iff $s \in L$. Consequently, by taking $\{(1)\}$ as the sole meaningful attractor of $\mathcal{N}'$, one obtains a DRNN[$\mathbb{R}$] (or an Ev$_2$-DRNN[$\mathbb{Q}$]) $\mathcal{N}'$ that recognizes L. In other terms, $L(\mathcal{N}') = L$.

According to these considerations, if $L \in BC(\mathbf{\Pi}^0_2)$, then L is recognizable by some DRNN[$\mathbb{R}$] and by some Ev$_2$-DRNN[$\mathbb{Q}$]. □

Proposition 2. *Let $L \subseteq (\mathbb{B}^M)^\omega$. If L is recognizable by some Ev-DRNN[$\mathbb{R}$], then $L \in BC(\mathbf{\Pi}^0_2)$.*

Proof. Let $L \subseteq (\mathbb{B}^M)^\omega$ be recognizable by some Ev-DRNN[$\mathbb{R}$] $\mathcal{N}$. Suppose that $\mathcal{N}$ contains the K meaningful attractors $A_i = \{\mathbf{b}_{i_1}, \ldots, \mathbf{b}_{i_{k(i)}}\}$, for $i = 1,\ldots,K$, where $1 \leq i_1 < \ldots < i_{k(i)} \leq 2^P$, and where $\mathbf{b}_n$ denotes the n-th Boolean vector of $\mathbb{B}^P$ according to the lexicographic order.

Note that the dynamics of $\mathcal{N}$ can naturally be associated with the function $g_{\mathcal{N}} : (\mathbb{B}^M)^\omega \to (\mathbb{B}^P)^\omega$ defined by $g_{\mathcal{N}}(s) = c_s'$, where $c_s' = \mathbf{y}(0)\mathbf{y}(1)\mathbf{y}(2)\cdots$ is the Boolean computation generated by $\mathcal{N}$ when the infinite input stream $s = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2)\cdots$ is received. The nature of our dynamics ensures that this function is sequential, i.e., for any time step $t \geq 0$, the Boolean vectors $\mathbf{u}(t)$ and $\mathbf{y}(t)$ are generated simultaneously. Hence, $g_{\mathcal{N}}$ is Lipschitz and thus continuous, cf. [13, Lemma 1].

Consequently, the neural ω-language $L(\mathcal{N})$ can be expressed as follows:

$$\begin{aligned}
L(\mathcal{N}) &= \left\{ s \in (\mathbb{B}^M)^\omega : \inf(c_s') \text{ is a meaningful attractor} \right\} \\
&= \left\{ s \in (\mathbb{B}^M)^\omega : \inf(c_s') = A_i \text{ for some } i = 1,\ldots,K \right\} \\
&= \bigcup_{i=1}^{K} \left\{ s \in (\mathbb{B}^M)^\omega : \inf(c_s') = A_i \right\}
\end{aligned}$$

$$= \bigcup_{i=1}^{K} \Big\{ s \in (\mathbb{B}^M)^\omega : \text{for all } j \in \{i_1, \ldots, i_{k(i)}\}, \; g_{\mathcal{N}}(s) \text{ contains infinitely many } \mathbf{b}_j\text{'s, and for all } j \in \{1, \ldots, 2^P\} \setminus \{i_1, \ldots, i_{k(i)}\}, \; g_{\mathcal{N}}(s) \text{ contains finitely many } \mathbf{b}_j\text{'s} \Big\}$$

$$= \bigcup_{i=1}^{K} \Big[\bigcap_{j \in \{i_1,\ldots,i_{k(i)}\}} \Big\{ s \in (\mathbb{B}^M)^\omega : g_{\mathcal{N}}(s) \text{ has infinitely many } \mathbf{b}_j\text{'s} \Big\} \cap \bigcap_{j \in \{1,\ldots,2^P\} \setminus \{i_1,\ldots,i_{k(i)}\}} \Big\{ s \in (\mathbb{B}^M)^\omega : g_{\mathcal{N}}(s) \text{ has finitely many } \mathbf{b}_j\text{'s} \Big\} \Big]$$

$$= \bigcup_{i=1}^{K} \Big[\bigcap_{j \in \{i_1,\ldots,i_{k(i)}\}} \Big\{ s \in (\mathbb{B}^M)^\omega : g_{\mathcal{N}}(s) \in \underbrace{\bigcap_{n \geq 0} \bigcup_{m \geq 0} (\mathbb{B}^P)^{n+m} \cdot \mathbf{b}_j \cdot (\mathbb{B}^P)^\omega}_{\substack{c'_s \text{ contains infinitely many } \mathbf{b}_j\text{'s, i.e.} \\ \forall n \geq 0 \; \exists m \geq n \; \mathbf{c}'_s(n+m) = \mathbf{b}_j \\ \text{thus in } \mathbf{\Pi}^0_2}} \Big\} \cap \bigcap_{j \in \{1,\ldots,2^P\} \setminus \{i_1,\ldots,i_{k(i)}\}} \Big\{ s \in (\mathbb{B}^M)^\omega : g_{\mathcal{N}}(s) \in \underbrace{\Big(\bigcap_{n \geq 0} \bigcup_{m \geq 0} (\mathbb{B}^P)^{n+m} \cdot \mathbf{b}_j \cdot (\mathbb{B}^P)^\omega \Big)^{\complement}}_{\substack{c'_s \text{ contains only finitely many } \mathbf{b}_j\text{'s} \\ \text{complement of a } \mathbf{\Pi}^0_2\text{-set, thus in } \mathbf{\Sigma}^0_2}} \Big\} \Big]$$

$$= \bigcup_{i=1}^{K} \Big[\bigcap_{j \in \{i_1,\ldots,i_{k(i)}\}} \underbrace{g_{\mathcal{N}}^{-1} \Big(\bigcap_{n \geq 0} \bigcup_{m \geq 0} (\mathbb{B}^P)^{n+m} \cdot \mathbf{b}_j \cdot (\mathbb{B}^P)^\omega \Big)}_{\text{preimage by a continuous function of a } \mathbf{\Pi}^0_2\text{-set, thus in } \mathbf{\Pi}^0_2} \cap \bigcap_{j \in \{1,\ldots,2^P\} \setminus \{i_1,\ldots,i_{k(i)}\}} \underbrace{g_{\mathcal{N}}^{-1} \Big(\Big(\bigcap_{n \geq 0} \bigcup_{m \geq 0} (\mathbb{B}^P)^{n+m} \cdot \mathbf{b}_j \cdot (\mathbb{B}^P)^\omega \Big)^{\complement} \Big)}_{\text{preimage by a continuous function of a } \mathbf{\Sigma}^0_2\text{-set, thus in } \mathbf{\Sigma}^0_2} \Big]$$

It follows that $L(\mathcal{N}) \in BC(\mathbf{\Pi}^0_2)$, since it consists of finite unions and finite intersections of $\mathbf{\Pi}^0_2$ and $\mathbf{\Sigma}^0_2$ sets. □

5.3. The nondeterministic case

We show that both models of NRNN[$\mathbb{Q}$]s and NRNN[$\mathbb{R}$]s are computationally equivalent to each other, and recognize precisely the class of all analytic sets ($\mathbf{\Sigma}^1_1$). Therefore, their expressive powers strictly encompass that of Muller Turing machines, which is restricted to the effectively analytic sets (Σ^1_1 lightface) (cf. Section 3).

Theorem 3. *Let $L \subseteq (\mathbb{B}^M)^\omega$ be some ω-language. The following conditions are equivalent:*

1. *$L \in \mathbf{\Sigma}^1_1$;*
2. *L is recognizable by some NRNN[$\mathbb{Q}$] $(\mathcal{N}, E)$;*
3. *L is recognizable by some NRNN[$\mathbb{R}$] $(\mathcal{N}, E)$.*

Proof. (1) → (2) is provided by the forthcoming Proposition 3. (2) → (3) holds by definition. (3) → (1) is provided by the forthcoming Proposition 4. □

First, we show that any analytic neural ω-language L can be recognized by some NRNN[$\mathbb{Q}$] $(\mathcal{N}, E)$. The idea of the proof is the following. To begin with, we note that the analytic set L can be written as the first projection π_1 of some $\mathbf{\Pi}^0_2$-set $X \subseteq (\mathbb{B}^M)^\omega \times \{0,1\}^\omega$ (cf. Section 2). Next, we consider some recursive encoding of X by an infinite word $w_X \in \{0,1\}^\omega$. Afterwards, we consider a NRNN[$\mathbb{Q}$] $(\mathcal{N}, E)$ equipped with only two possible evolving synaptic connections: one which might follow any possible binary evolution $e \in \{0,1\}^\omega$, and the other one which always follows the same binary evolution $w_X \in \{0,1\}^\omega$. We then design the static part of $\mathcal{N}$ such that the network visits a meaningful attractor iff the current input

s and evolving synaptic pattern $e \in \{0,1\}^\omega$ are such that (s,e) belongs the set encoded by w_X, namely X. In this way, $L((\mathcal{N}, E)) = \pi_1(X) = L$, and thus L is recognized by $(\mathcal{N}, E)$.

Proposition 3. *Let $L \subseteq (\mathbb{B}^M)^\omega$ such that $L \in \mathbf{\Sigma}^1_1$. Then there exists some NRNN[$\mathbb{Q}$] $(\mathcal{N}, E)$ such that $L((\mathcal{N}, E)) = L$.*

Proof. Since $L \in \mathbf{\Sigma}^1_1$, there exists some $X \subseteq (\mathbb{B}^M)^\omega \times \{0,1\}^\omega$ such that $X \in \mathbf{\Pi}^0_2$ and $L = \pi_1(X)$ (cf. Section 2). Since $X \in \mathbf{\Pi}^0_2$, it can be written as $X = \bigcap_{i \geq 0} \bigcup_{j \geq 0} (p_{i,j} \cdot (\mathbb{B}^M)^\omega \times q_{i,j} \cdot \{0,1\}^\omega)$, where each $(p_{i,j}, q_{i,j}) \in (\mathbb{B}^M)^* \times \{0,1\}^*$. Consequently, the set X (and hence also L) is completely determined by the countable sequence of pairs of finite prefixes $((p_{i,j}, q_{i,j}))_{i,j \geq 0}$. We consider some encoding $w_X \in \{0,1\}^\omega$ of the sequence $((p_{i,j}, q_{i,j}))_{i,j \geq 0}$ such that, for any pair of indices $(i,j) \in \mathbb{N} \times \mathbb{N}$, the decoding procedure $(w_X, i, j) \mapsto (p_{i,j}, q_{i,j})$ is actually recursive.

We now consider the infinite procedure given by Algorithm 2 below. This procedure requires as input and auxiliary items the following three infinite sequences delivered step by step: an infinite input stream $s \in (\mathbb{B}^M)^\omega$, an infinite word $e \in \{0,1\}^\omega$ chosen arbitrarily, and the precise infinite word $w_X \in \{0,1\}^\omega$. Note that provided that these three items are correctly supplied by some external source, every instruction of the procedure is actually recursive. Farther note that, by construction, the Algorithm 2 returns infinitely many 1's on the pair of infinite sequences (s,e) iff (s,e) belongs to X.

Algorithm 2 Infinite procedure.

Require:
1. Input $s = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2) \cdots \in (\mathbb{B}^M)^\omega$ supplied step by step at successive time steps $t = 0, 1, 2, \ldots$
2. some auxiliary infinite word $e = e(0)e(1)e(2) \cdots \in \{0,1\}^\omega$ supplied step by step at successive time steps $t = 0, 1, 2, \ldots$
3. the specific auxiliary infinite word $w_X = w_X(0)w_X(1)w_X(2) \cdots \in \{0,1\}^\omega$ supplied step by step at successive time steps $t = 0, 1, 2, \ldots$

SUBROUTINE 1
$c \leftarrow 0$ // c counts the number of letters provided so far
for all time step $t \geq 0$ **do**
  store each incoming Boolean vector $\mathbf{u}(t) \in \mathbb{B}^M$
  store each incoming bit $e(t) \in \{0,1\}$
  store each incoming bit $w_X(t) \in \{0,1\}$
  $c \leftarrow c + 1$
end for
END SUBROUTINE 1

SUBROUTINE 2
$i \leftarrow 0,\ j \leftarrow 0$
loop
  wait until $w_X[0{:}c]$ becomes long enough to contain the encoding of $(p_{i,j}, q_{i,j})$
  wait until $c \geq \max\{|p_{i,j}|, |q_{i,j}|\}$
  decode $(p_{i,j}, q_{i,j})$ from $w_X[0{:}c]$ // recursive procedure
  if $p_{i,j} \subseteq s[0{:}c]$ and $q_{i,j} \subseteq e[0{:}c]$ **then** // $(s,e) \in p_{i,j} \cdot (\mathbb{B}^M)^\omega \times q_{i,j} \cdot \{0,1\}^\omega$
    return 1 // $\exists\ j$ s.t. $(s,e) \in p_{i,j} \cdot (\mathbb{B}^M)^\omega \times q_{i,j} \cdot \{0,1\}^\omega$
    $i \leftarrow i + 1,\ j \leftarrow 0$ // test if $(s,e) \in p_{i+1,0} \cdot (\mathbb{B}^M)^\omega \times q_{i+1,0} \cdot \{0,1\}^\omega$
  else // $(s,e) \notin p_{i,j} \cdot (\mathbb{B}^M)^\omega \times q_{i,j} \cdot \{0,1\}^\omega$
    return 0 // $\neg \exists j' \leq j$ s.t. $(s,e) \in p_{i,j'} \cdot (\mathbb{B}^M)^\omega \times q_{i,j'} \cdot \{0,1\}^\omega$
    $i \leftarrow i,\ j \leftarrow j + 1$ // test if $(s,e) \in p_{i,j+1} \cdot (\mathbb{B}^M)^\omega \times q_{i,j+1} \cdot \{0,1\}^\omega$
  end if
end loop
END SUBROUTINE 2

Based on the infinite procedure, we provide the description of a NRNN[$\mathbb{Q}$] $(\mathcal{N}, E)$ such that $L((\mathcal{N}, E)) = L$. The network $(\mathcal{N}, E)$ contains only two evolving synaptic weights $w_1(t)$ and $w_2(t)$ which evolve among only two possible values, 0 or 1. All other synaptic weights are static and rational. The weight $w_1(t)$ might follow every possible evolution in $\{0,1\}^\omega$, while $w_2(t)$ always follows the same evolution, which are the successive letters of w_X. Formally, one has the following *closed* set of possible evolutions:

$$E = \{\tilde{e} \in (\{0,1\}^2)^\omega : (\tilde{\mathbf{e}}(t))_0 \in \{0,1\} \text{ and } (\tilde{\mathbf{e}}(t))_1 = w_X(t), \text{ for any } t \geq 0\}.$$

We then consider a neural circuit which stores the incoming values of the input stream $s \in (\mathbb{B}^M)^\omega$ into M designated neurons, as well as two neural circuits which store the successive bits of $w_1(t)$ and $w_2(t)$ into two designated neurons (see [41] for further technical details). Afterwards, according to the real time computational equivalence between (static) rational RNNs and TMs [41], we consider a (static) rational RNN which is suitably designed and connected to the above mentioned circuits in order to simulate all the recursive instructions of Algorithm 2. We then add a single Boolean output neuron y and update the whole construction in order that y takes an activation value of 1 iff the simulation of Algorithm 2 by our network enters the instruction "return 1". Finally, the Boolean output cell y leads to the existence of only three possible attractors, namely $\{(0)\}$, $\{(0),(1)\}$, and $\{(1)\}$. We set $\{(0)\}$ as spurious, and $\{(0),(1)\}$ and $\{(1)\}$ as meaningful.

In this way, one has the description of a NRNN[$\mathbb{Q}$] $(\mathcal{N}, E)$ which suitably simulates the behavior of Algorithm 2. By construction, given any infinite input $s \in \{0,1\}^\omega$ and infinite evolution $\tilde{e} = (e(t), w_X(t))_{t \in \mathbb{N}} \in E$, the Boolean computation

$c'_{(s,\tilde{e})}$ visits a meaningful attractor iff Algorithm 2 returns infinitely many 1's on the pair of infinite sequences (s, e) (where $e = e(0)e(1)e(2)\cdots$ is the sequence of first coordinates of $\tilde{e}$).

Therefore, one has that $s \in L((\mathcal{N}, E))$ iff, by definition, there exists some $\tilde{e} \in E$ such that $\inf(c'_{(s,\tilde{e})})$ is meaningful, iff there exists $e \in \{0,1\}^\omega$ such that $\inf(c'_{(s,\tilde{e})})$ is meaningful (since the second coordinates of $\tilde{e}$ are fixed), iff, by construction, there exists $e \in \{0,1\}^\omega$ such that Algorithm 2 returns infinitely many 1's on the pair of infinite sequences (s, e), iff there exists $e \in \{0,1\}^\omega$ such that the pair $(s, e) \in X$, iff, by definition, $s \in \pi_1(X) = L$. Therefore, $L((\mathcal{N}, E)) = L$, showing that L is recognized by the NRNN[$\mathbb{Q}$] $(\mathcal{N}, E)$. □

Conversely, we show that every ω-language recognized by some NRNN is analytic. Towards this purpose, we need a preliminary Lemma. This result is based on the important result [10, Lemma 9] which states that, for every $t \geq 0$, any evolving real-weighted neural network $\mathcal{N}$ can be perfectly simulated by some other evolving rational-weighted neural network $\mathcal{N}_t$, up to time step t.

Lemma 1. *Let $(\mathcal{N}, E)$ be some NRNN[$\mathbb{R}$] and let $f_{(\mathcal{N},E)} : (\mathbb{B}^M)^\omega \times E \to (\mathbb{B}^P)^\omega$ be the function defined by $f_{(\mathcal{N},E)}(s, e) = c'_{(s,e)}$, where $c'_{(s,e)} = \mathbf{y}(0)\mathbf{y}(1)\mathbf{y}(2)\cdots$ is the Boolean computation produced by $(\mathcal{N}, E)$ when it receives the input stream $s = \mathbf{u}(0)\mathbf{u}(1)\mathbf{u}(2)\cdots$ and evolution $e = \mathbf{e}(0)\mathbf{e}(1)\mathbf{e}(2)\cdots$. Then $f_{(\mathcal{N},E)}$ is of Baire class 1.*

Proof. Note that the nature of our dynamics ensures that the function $f_{(\mathcal{N},E)}$ is sequential, i.e., for any time step $t \geq 0$, the vectors $\mathbf{u}(t)$, $\mathbf{e}(t)$ and $\mathbf{y}(t)$ are generated simultaneously. Therefore, for any $t \geq 0$, the t first Boolean vectors $\mathbf{y}(0)\mathbf{y}(1)\cdots\mathbf{y}(t-1)$ only depend on the t first input and evolution vectors $\mathbf{u}(0)\mathbf{u}(1)\cdots\mathbf{u}(t-1)$ and $\mathbf{e}(0)\mathbf{e}(1)\cdots\mathbf{e}(t-1)$. Formally, given any basic open set $y \cdot (\mathbb{B}^P)^\omega$ with $y \in (\mathbb{B}^P)^*$, one has that $f^{-1}_{(\mathcal{N},E)}(y \cdot (\mathbb{B}^P)^\omega)$ is of the form

$$\Theta_y = \bigcup_{i \in I} \Big[u_i \cdot (\mathbb{B}^M)^\omega \times (e_{\mathbb{R},i} \cdot ([S, S']^K)^\omega \cap E) \Big]$$

where each $u_i \in (\mathbb{B}^M)^{|y|}$ and $e_{\mathbb{R},i} \in ([S, S']^K)^{|y|}$ (i.e., u_i and $e_{\mathbb{R},i}$ are sequences of length $|y|$).

Now, the result [10, Lemma 9] ensures that, for any $(s, e) \in (\mathbb{B}^M)^\omega \times E$, the Boolean computation produced by $(\mathcal{N}, E)$ subjected to (s, e) is the very same – up to time step $|y|$ – as that produced by $(\mathcal{N}, E)$ subjected to (s, e'), where e' is obtained by truncating all the components of all the vectors of e after $t(|y|)$ bits,[6] for some function $t : \mathbb{N} \to \mathbb{N}$. Formally, by [10, Lemma 9], given any u_i and $e_{\mathbb{R},i}$ as above, there exists $I_{\mathbb{Q},i} = \left(\prod_{k=1}^{K} [a_{j,k}, b_{j,k}] \right)_{j<|y|}$, where each $a_{j,k}, b_{j,k} \in \mathbb{Q}$ and $e_{\mathbb{R},i} \in I_{\mathbb{Q},i}$, and such that

$$f_{(\mathcal{N},E)} \left(u_i \cdot (\mathbb{B}^M)^\omega \times (I_{\mathbb{Q},i} \cdot ([S, S']^K)^\omega \cap E) \right) \subseteq y \cdot (\mathbb{B}^P)^\omega.$$

Hence, if we let $u_i \cdot (\mathbb{B}^M)^\omega \times (I_{\mathbb{Q},i} \cdot ([S, S']^K)^\omega \cap E)$ be denoted by C_i, the above relation is equivalent to $C_i \subseteq f^{-1}_{(\mathcal{N},E)}(y \cdot (\mathbb{B}^P)^\omega) = \Theta_y$, for each $i \in I$, and thus $\bigcup_{i \in I} C_i \subseteq \Theta_y$. On the other hand, by construction of the closed hyper-intervals $I_{\mathbb{Q},i}$, one has $\Theta_y \subseteq \bigcup_{i \in I} C_i$. Therefore, $\Theta_y = \bigcup_{i \in I} C_i$, i.e.,

$$\Theta_y = \bigcup_{i \in I} \Big[u_i \cdot (\mathbb{B}^M)^\omega \times (I_{\mathbb{Q},i} \cdot ([S, S']^K)^\omega \cap E) \Big].$$

Since the bounds of the hyper-intervals $I_{\mathbb{Q},i}$'s are rational numbers, there exist only countably many distinct u_i and $I_{\mathbb{Q},i}$, and thus, Θ_y can be rewritten as

$$\Theta_y = \bigcup_{i \in J} \Big[u_i \cdot (\mathbb{B}^M)^\omega \times (I_{\mathbb{Q},i} \cdot ([S, S']^K)^\omega \cap E) \Big]$$

where the index set J is countable.

Now, notice that for each $i \in J$, the sets $u_i \cdot (\mathbb{B}^M)^\omega$ and $I_{\mathbb{Q},i} \cdot ([S, S']^K)^\omega \cap E$ are cl-open and closed sets of $(\mathbb{B}^M)^\omega$ and E, respectively. Consequently, $u_i \cdot (\mathbb{B}^M)^\omega \times (e_{\mathbb{R},i} \cdot ([S, S']^K)^\omega \cap E)$ is a closed set of $(\mathbb{B}^M)^\omega \times E$, as a product of closed sets, and Θ_y is a $\boldsymbol{\Sigma}^0_2$-set of $(\mathbb{B}^M)^\omega \times E$, as a countable union of closed sets. Therefore, $f_{(\mathcal{N},E)}$ of Baire class 1. □

Proposition 4. *Let $(\mathcal{N}, E)$ be some NRNN[$\mathbb{R}$]. Then $L((\mathcal{N}, E)) \in \boldsymbol{\Sigma}^1_1$.*

[6] Note that for any $r \in \mathbb{R}$, the set of $r' \in \mathbb{R}$ such that the truncations of (the binary representations of) r and r' after n bits are the same is a closed interval of the form $[a, b]$, where $a, b \in \mathbb{Q}$ and $r \in [a, b]$. Formally, a and b are the rational numbers whose binary representations are $r|_n \cdot 0^\omega$ and $r|_n \cdot 1^\omega$, respectively, where $r|_n$ denotes the truncation of (the binary representations of) r after n bits.

Proof. Since $\mathcal{N}$ contains finitely many Boolean output cells, it can exhibit finitely many possible output states, and thus also finitely many possible attractors. This feature is independent from the nondeterministic behavior associated with the set of possible evolutions E. Hence, suppose that $\mathcal{N}$ contains the I meaningful attractors $A_i = \{\mathbf{b}_{i_1}, \dots, \mathbf{b}_{i_{k(i)}}\}$, for $i = 1, \dots, I$, where $1 \le i_1 < \dots < i_{k(i)} \le 2^P$, and where $\mathbf{b}_n$ denotes the n-th Boolean vector of $\mathbb{B}^P$ according to the lexicographic order. Then, the ω-language $L((\mathcal{N}, E))$ can be expressed by the following sequence of equalities (where we use the fact that $f_{(\mathcal{N},E)}$ is of Baire class 1, established in Lemma 1):

$$\begin{aligned}
L((\mathcal{N}, E)) &= \left\{s \in (\mathbb{B}^M)^\omega : \text{there exists } e \in E \text{ s.t. } \mathit{inf}\,(c'_{(s,e)}) \text{ is a meaningful attractor}\right\} \\
&= \left\{s \in (\mathbb{B}^M)^\omega : \text{there exists } e \in E \text{ s.t. } \inf(c'_{(s,e)}) = A_i, \text{ for some } i = 1, \dots, I\right\} \\
&= \pi_1\Big(\left\{(s, e) \in (\mathbb{B}^M)^\omega \times E : \inf(c'_{(s,e)}) = A_i, \text{for some } i = 1, \dots, I\right\}\Big) \\
&= \pi_1\Big(\bigcup_{i=1}^{I}\left\{(s, e) \in (\mathbb{B}^M)^\omega \times E : \inf(c'_{(s,e)}) = A_i\right\}\Big) \\
&= \pi_1\Big(\bigcup_{i=1}^{I}\big\{(s, e) \in (\mathbb{B}^M)^\omega \times E : \\
&\qquad \forall j \in \{i_1, \dots, i_{k(i)}\},\ f_{(\mathcal{N},E)}(s, e) \text{ contains infinitely many } \mathbf{b}_j\text{'s and} \\
&\qquad \forall j \in \{1, \dots, 2^P\} \setminus \{i_1, \dots, i_{k(i)}\},\ f_{(\mathcal{N},E)}(s, e) \text{ contains finitely many } \mathbf{b}_j\text{'s}\big\}\Big) \\
&= \pi_1\Big(\bigcup_{i=1}^{I}\Big[\bigcap_{j\in\{i_1,\dots,i_{k(i)}\}} \Big\{(s, e) \in (\mathbb{B}^M)^\omega \times E : f_{(\mathcal{N},E)}(s, e) \in \underbrace{\bigcap_{n\ge 0}\bigcup_{m\ge 0}(\mathbb{B}^P)^{n+m} \cdot \mathbf{b}_j \cdot (\mathbb{B}^P)^\omega}_{\substack{c'_{(s,e)} \text{ contains infinitely many } \mathbf{b}_j\text{'s, i.e.} \\ \forall n \ge 0\ \exists m \ge n\ \mathbf{y}(n+m) = \mathbf{b}_j, \text{ thus in } \mathbf{\Pi}^0_2}}\Big\} \cap \\
&\qquad \bigcap_{j\in\{1,\dots,2^P\}\setminus\{i_1,\dots,i_{k(i)}\}} \Big\{(s, e) \in (\mathbb{B}^M)^\omega \times E : f_{(\mathcal{N},E)}(s, e) \in \underbrace{\Big(\bigcap_{n\ge 0}\bigcup_{m\ge 0}(\mathbb{B}^P)^{n+m} \cdot \mathbf{b}_j \cdot (\mathbb{B}^P)^\omega\Big)^{\complement}}_{\substack{c'_{(s,e)} \text{ contains only finitely many } \mathbf{b}_j\text{'s, i.e.} \\ \text{complement of a } \mathbf{\Pi}^0_2\text{-set, thus in } \mathbf{\Sigma}^0_2}}\Big\}\Big]\Big) \\
&= \pi_1\Big(\bigcup_{i=1}^{I}\Big[\bigcap_{j\in\{i_1,\dots,i_{k(i)}\}} \underbrace{f^{-1}_{(\mathcal{N},E)}\Big(\bigcap_{n\ge 0}\bigcup_{m\ge 0}(\mathbb{B}^P)^{n+m} \cdot \mathbf{b}_j \cdot (\mathbb{B}^P)^\omega\Big)}_{\substack{\text{preimage by a Baire class 1 function of a } \mathbf{\Pi}^0_2\text{-set} \\ \text{thus in } \mathbf{\Pi}^0_3\ [31]}} \cap \\
&\qquad \bigcap_{j\in\{1,\dots,2^P\}\setminus\{i_1,\dots,i_{k(i)}\}} \underbrace{f^{-1}_{(\mathcal{N},E)}\Big(\Big(\bigcap_{n\ge 0}\bigcup_{m\ge 0}(\mathbb{B}^P)^{n+m} \cdot \mathbf{b}_j \cdot (\mathbb{B}^P)^\omega\Big)^{\complement}\Big)}_{\substack{\text{preimage by a Baire class 1 function of a } \mathbf{\Sigma}^0_2\text{-set} \\ \text{thus in } \mathbf{\Sigma}^0_3\ [31]}}\Big]\Big)
\end{aligned}$$

It follows that $L((\mathcal{N}, E))$ is a projection of a finite union and intersection of $\mathbf{\Pi}^0_3$ and $\mathbf{\Sigma}^0_3$ subsets of the Polish space $(\mathbb{B}^M)^\omega \times E$, and therefore, $L((\mathcal{N}, E)) \in \mathbf{\Sigma}^1_1$. □

6. Discussion

6.1. Beyond the Turing limits

We have characterized the expressive powers of several models of deterministic and nondeterministic first-order sigmoidal recurrent neural networks in relation with their attractor dynamics.

In the deterministic context, we have considered six different models of RNNs according to whether the synaptic weights are modeled by rational or real numbers, and according to whether the these synaptic weights are either of a static nature, or able to evolve over time among only two possible values, or able to evolve over time among any possible values between two designated bounds. We have shown that DRNN[$\mathbb{Q}$]s are computationally equivalent to deterministic Muller Turing machines (Theorem 1). The five other models of Ev$_2$-DRNN[$\mathbb{Q}$]s, Ev-DRNN[$\mathbb{Q}$]s, DRNN[$\mathbb{R}$]s, Ev$_2$-DRNN[$\mathbb{R}$]s, Ev-DRNN[$\mathbb{R}$]s are computationally equivalent to each other and strictly more powerful than deterministic Muller Turing machines – with a class of ω-languages equal to $BC(\mathbf{\Pi}^0_2)$ (Theorem 2). These results show that the incorporation of bi-valued evolving capabilities into some basic model of rational-weighted recurrent neural networks provides the possibility to break the Turing barrier of computation. The additional consideration of real synaptic weights or of any more general mechanism of architectural evolvability would actually not further increase the capabilities of the neural networks. These considerations constitute a precise generalization to the current computational context of those obtained for the cases of classical as well as interactive computations [6,8–10,15,17].

In the nondeterministic context, we have proven that the two models of NRNN[$\mathbb{Q}$]s and NRNN[$\mathbb{R}$]s are computationally equivalent to each other and strictly more expressive than the nondeterministic Muller Turing machines – with a class of ω-languages equal to $\mathbf{\Sigma}^1_1$ (Theorem 3). Consequently, the nondeterminism does inject an extensive amount of computational power to the neural systems, from $BC(\mathbf{\Pi}^0_2)$ to $\mathbf{\Sigma}^1_1$. The consideration of real synaptic weights does in this case not provide any further expressive power to the neural networks. The added value of the power of the continuum is somehow absorbed by the nondeterminism.

Overall, these results show that bi-valued evolving capabilities of neural networks represent sufficient conditions to transcend the classical Turing barrier of computation and achieve maximal computational capabilities. Any further kind of analog or general evolving assumptions can be dropped out without compromizing the achievement of this maximal computational power. This feature is of specific interest, since discrete architectural evolving phenomena are indeed observable in biological neural networks – as opposed to the power of the continuum, which remains at a conceptual level. These considerations also provide novel theoretical insights into the correlation that would exist between the learning paradigms (achieved via evolving synaptic weights) and the computational capabilities of neural networks.

Our achievements also show that recurrent neural networks constitute a natural model of computation beyond the scope classical Turing machines [14]. The computational processes that cannot be encompassed by the classical Turing machine model (in whatever computational paradigm) have sometimes be referred to as *super-Turing* [39]. In our context, the emergence of such super-Turing capabilities necessarily requires the consideration of non-recursive real numbers or non-computable patterns of evolution in the model. Otherwise, the whole process could be simulated by some Turing machine. The question of the possible achievement of such super-Turing capabilities in biological neural networks lies beyond the scope of this work. For deeper philosophical considerations about hypercomputation, see for instance [23,24,37,44,45].

6.2. Attractor dynamics and spatiotemporal patterns

The key element that we want to emphasize is the association between dynamical systems, attractor dynamics, and spatiotemporal patterns. In fact, attractor dynamics and spatiotemporal patterns of discharges are likely to be significantly involved in the processing and coding of information in the brain [1,51]. For instance, experimental evidence exist of attractor dynamics in the time series of neuronal spikes recorded in the brain of rats and primates performing some specific behavioral task [20,21,56]. Spatiotemporal patterns of discharges have been observed in several brain areas and in relation with specific sensory stimulations and behavioral paradigms [2,49,51–54,57]. They have also been observed in neural network simulations at various scales [28,55], in particular as the outcome of evolvable changes in synaptic weights determined by simple learning rules [29]. Furthermore, the association between attractor dynamics and repeating firing patterns has been demonstrated in nonlinear dynamical systems [3,4] and in simulations of large scale neuronal networks [28,29]. Hence, it is important to consider the spatiotemporal patterns not as kind of Morse-code messages, but rather as witnesses of an underlying dynamics – the attractor dynamics – which is assumed to be a key feature of neural coding. A real spatiotemporal pattern observed in the context of an experimental paradigm is illustrated in Fig. 3.

Our attractor-based approach to the computational capabilities of recurrent neural networks is justified by the fact that, in our model, the periodic attractor dynamics of the neural networks are the precise phenomena that underly the arising of spatiotemporal patterns of discharges, as described in Section 4.4. Our results suggest that the computational power of recurrent neural networks would be related to the topological complexity of their underlying neural ω-language, i.e., the set of input streams which induce a meaningful attractors dynamics. In other words, some aspect of the computational capabilities of recurrent neural networks would be precisely determined by the ability of these networks to perform more or less complicated *classification tasks* of their input streams via the manifestation of meaningful or spurious attractor dynamics.

To illustrate this feature, consider some recurrent neural network $\mathcal{N}$ whose associated ω-language would be the (very) basic $\mathbf{\Sigma}^0_1$-set $L(\mathcal{N}) = \mathbf{0} \cdot \mathbf{0} \cdot \mathbf{1} \cdot (\mathbb{B}^M)^\omega$ (where $\mathbf{0}$ and $\mathbf{1}$ denote the M-dimensional vectors with only 0's and 1's, respectively). If some input stream s is supplied to the network $\mathcal{N}$, then it will simply look at the three first elements of s, and, if they correspond to $\mathbf{0}$, $\mathbf{0}$ and $\mathbf{1}$, it will accept the whole input by entering into some meaningful attractor (since $s \in \mathbf{0} \cdot \mathbf{0} \cdot \mathbf{1} \cdot (\mathbb{B}^M)^\omega$); otherwise, it will reject the input by entering into some spurious attractor (since $s \notin \mathbf{0} \cdot \mathbf{0} \cdot \mathbf{1} \cdot (\mathbb{B}^M)^\omega$). Consequently, in this basic case, the classification task of $\mathcal{N}$ reduces to the simple scanning procedure of the three first elements of s in order to accept or reject the input

However, if the ω-language $L(\mathcal{N})$ consists of a more topologically complicated set, then the classification task performed by $\mathcal{N}$ to accept or reject the input will amount to a much more complicated "procedure". For instance, Algorithm 1 describes the classification task associated with a $\mathbf{\Pi}^0_2$ ω-language. Algorithm 2 describes the classification task associated with a $\mathbf{\Sigma}^1_1$-set, i.e., a set which can be of any of the ω_1 (transfinitely many) Borel ranks, and even higher. In this case, the power of the nondeterminism is crucially involved: it provides – through the evolvability of the network – an encoded additional information which allows the network to reduce its classification task to that of a $\mathbf{\Pi}^0_2$ ω-language only (this is precisely the

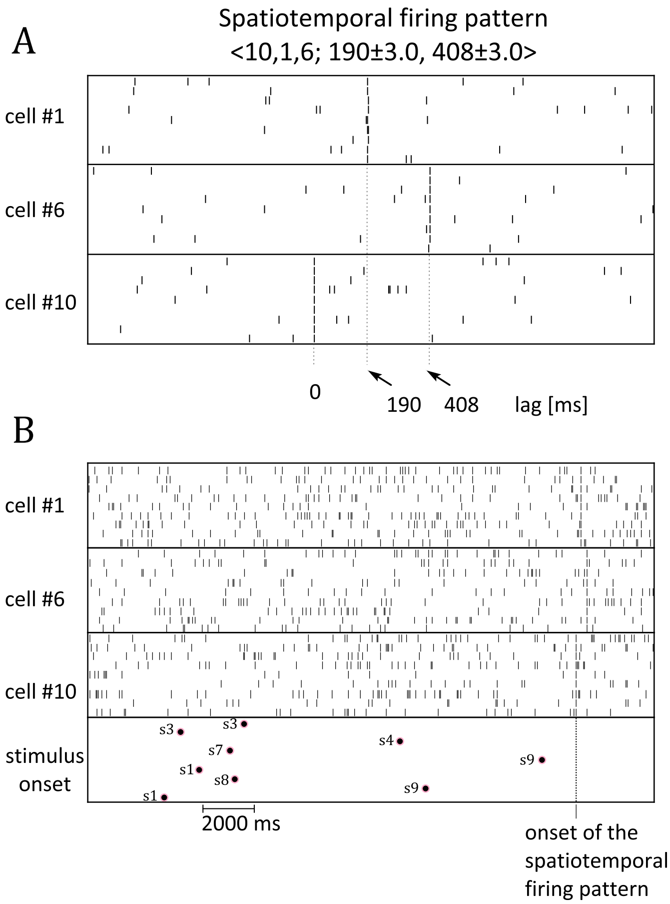

Fig. 3. In an experimental paradigm, freely-moving behaving rats were trained to discriminate two classes of human vowels [26,52]. If the discrimination was correct, the rats had to go to a specific area of the labyrinth and received a food reward. After reaching a steady-state performance, usually above 90%, the rats underwent a brain surgery and multiple electrodes were fit in an area of the cerebral cortex connected to the ascending auditory pathway and in the inferolimbic cortex, that is an area of the cerebral cortex associated with higher cerebral functions (processing of cross-contingencies, time relations, rehearsal of memory traces). The figure shows a spatiotemporal pattern of discharges associated to the class of stimuli that were priming the food reward. Note that the firing pattern was associated with the equivalence class of the vowel, irrespective of the pitch or any other simple acoustical feature. **A.** Raster display of the activities of three cortical neurons: Cells #1 and #6 were recorded in the auditory area Te1 from the left hemisphere from two different electrodes, and cell #10 in the auditory area Te3 from the right hemisphere. The rasters are aligned by displaying the first spike in the pattern at lag 0 ms. The pattern occurred 9 times during the session, starting with a spike of cell #10, then after 190 ms a spike of cell #1 with a jitter ± 3 ms and then after 218 ± 3 ms a spike of cell #6. This pattern is denoted $\langle 10, 1, 6; 190 \pm 3, 408 \pm 3 \rangle$. **B.** The same triplet is plotted at a different time scale such to show the corresponding stimuli onset times. Notice that the pattern occurred always after stimuli of the class 's', but that the lag after specific stimuli, i.e. 's9', is much shorter than after other specific stimuli, i.e. 's1'.

idea of the proof of Proposition 3). In all cases, the output of the classification task is provided by an attractor dynamics: a meaningful attractor signifies the acceptance of the input, and a spurious one signifies rejection.

In this sense, a neural ω-language can be associated to some corresponding classification task. The topological complexity of that language would then correspond to the intricacy of the corresponding task. According to this correspondence, those classification tasks referring to non-Turing recognizable ω-languages can be argued as lying outside the scope of current algorithmic. They could be qualified as *super-Turing* or *hypercomputational* procedures. The real-weighted and evolving recurrent neural networks are capable to perform such hypercomputational tasks.

We can now illustrate these theoretical considerations in the experimental context described in Fig. 3. In this case, a biological neural network performs the discrimination of an equivalence class of a human vowels. The discrimination of the auditory input – leading to the classification of an equivalence class of human vowels – is achieved via some specific attractor dynamics of the neural network activity, which, in turn, evokes precise spatiotemporal patterns of discharges that can be detected experimentally. According to our model, *the neural ω-language* of the neural network would correspond to the set of auditory inputs which belong to the suitable equivalence class of a human vowels. The problem of determining whether some given auditory input belongs to its neural ω-language or not corresponds to the classification task performed by the neural network. The *output* of this classification process, expressed via some *meaningful or spurious attractor dynamics*, would evoke the precise spatiotemporal patterns of discharge that are detected experimentally. The *computational power* of the neural network, determined as the *topological complexity* of its underlying ω-language, would correspond to the intricacy of the classification task performed by the network. Note that the observed spatiotemporal patterns do not allow to conclude anything about the topological complexity of the underlying neural ω-language, or in other words, about the intricacy of the classification task of the biological network.

6.3. Conclusion

The present study provides a novel theoretical approach to the crucial role that attractors and spatiotemporal patterns of discharges play in the computational capabilities of neural networks, and more generally, in the processing and coding of information in the brain. It establishes a link between the attractor dynamics of the networks, their spatiotemporal patterns of discharge, and their ability to perform more or less intricate discrimination tasks.

Recently, Cicurel and Nicolelis argued that the brain cannot be simulated by any Turing machine [22]. They claim that the brain should rather be conceived as a *hybrid digital-analog computational engine* (HDACE):

> According to the relativistic brain theory, complex central nervous systems like ours generate, process, and store information through the recursive interaction of a hybrid digital-analog computation engine (HDACE). In the HDACE, the digital component is defined by the spikes produced by neural networks distributed all over the brain, whereas the analog component is represented by the superimposition of time-varying, neuronal electromagnetic fields (NEMFs), generated by the flow of neuronal electrical signals through the multitude of local and distributed loops of white matter that exist in the mammalian brain. [22, p. 27]

Our concept of recurrent neural networks – composed with Boolean and sigmoidal cells – follows such a hybrid digital-analog – but also evolvable or adaptale – conception of the brain. Accordingly, the present work provides a theoretical foundation towards the understanding of the computational capabilities of such hybrid digital-analog-evolving brain-like models of computation.

For future work, a refined classification of the expressive power of recurrent neural networks according to the nature of their real synaptic weights or patterns of evolution is expected to be studied. For instance, it would be particularly interesting to understand the conditions on the synaptic weights or patterns of evolution that would correspond to the various Borel classes, or even the Wadge classes[7] [58], of the networks' underlying ω-languages. Besides, a study of the relationship between the graph topology of the networks and the topological complexity of their underlying ω-languages would also be of specific interest. This would provide a characterization of the graph theoretical invariants involved in the attractor-based computational complexity of the networks. More generally, the study of the computational capabilities of more biologically-oriented neural models involved in more bio-inspired paradigms of computation is expected to be pursued.

Finally, we hope that such comparative studies between the computational capabilities of neural models and abstract machines might eventually bring further insight into the understanding of both biological and artificial intelligences. We believe that similarly to the foundational work from Turing [47], which played a crucial role in the practical realization of modern computers, further theoretical considerations about neural- and natural-based models of computation shall contribute to the emergence of novel computational technologies, and step by step, open the way to the next computational generation.

Acknowledgments

This study was partially supported by the Swiss National Science Foundation grant #IZK0Z2_161582 and paritally conducted as part of the project Labex MME-DII (ANR11-LBX-0023-01).

References

[1] M. Abeles, Local Cortical Circuits. An Electrophysiological Study, Studies of Brain Function, vol. 6, Springer-Verlag, Berlin Heidelberg New York, 1982.
[2] M. Abeles, G.L. Gerstein, Detecting spatiotemporal firing patterns among simultaneously recorded single neurons, J. Neurophysiol. 60 (3) (Sep 1988) 909–924.
[3] Y. Asai, A.E.P. Villa, Reconstruction of underlying nonlinear deterministic dynamics embedded in noisy spike trains, J. Biol. Phys. 34 (3–4) (2008) 325–340.
[4] Y. Asai, A.E.P. Villa, Integration and transmission of distributed deterministic neural activity in feed-forward networks, Brain Res. 1434 (2012) 17–33.
[5] V. Braitenberg, A. Schüz, Cortex: Statistics and Geometry of Neuronal Connectivity, 2nd thoroughly rev. ed. edition, Springer, Berlin, 1998.
[6] J. Cabessa, Interactive evolving recurrent neural networks are super-Turing, in: J. Filipe, A.L.N. Fred (Eds.), Proceedings of ICAART, SciTePress, 2012, pp. 328–333.
[7] J. Cabessa, J. Duparc, Expressive power of non-deterministic evolving recurrent neural networks in terms of their attractor dynamics, in: C.S. Calude, M.J. Dinneen (Eds.), Proceedings of UCNC 2015 The 14th International Conference on Unconventional Computation and Natural Computation, in: Lecture Notes in Computer Science, vol. 9252, Springer, 2015, pp. 144–156.
[8] J. Cabessa, H.T. Siegelmann, Evolving recurrent neural networks are super-Turing, in: Proceedings of IJCNN 2011, IEEE, 2011, pp. 3200–3206.

[7] The Wadge hierarchy is a considerable refinement of the Borel hierarchy [25,58].

[9] J. Cabessa, H.T. Siegelmann, The computational power of interactive recurrent neural networks, Neural Comput. 24 (4) (2012) 996–1019.
[10] J. Cabessa, H.T. Siegelmann, The super-Turing computational power of plastic recurrent neural networks, Int. J. Neural Syst. 24 (8) (2014).
[11] J. Cabessa, A.E.P. Villa, A Hierarchical classification of first-order recurrent neural networks, Chin. J. Phys. 53 (2010) 407–416.
[12] J. Cabessa, A.E.P. Villa, A hierarchical classification of first-order recurrent neural networks, in: Proceedings of LATA 2010, in: Lecture Notes in Computer Science, vol. 6031, Springer, 2010, pp. 142–153.
[13] J. Cabessa, A.E.P. Villa, The expressive power of analog recurrent neural networks on infinite input streams, Theor. Comput. Sci. 436 (2012) 23–34.
[14] J. Cabessa, A.E.P. Villa, Recurrent neural networks – a natural model of computation beyond the Turing limits, in: Proceedings of IJCCI 2012, SciTePress, 2012, pp. 594–599.
[15] J. Cabessa, A.E.P. Villa, The super-Turing computational power of interactive evolving recurrent neural networks, in: Proceedings of ICANN 2013, in: Lecture Notes in Computer Science, vol. 8131, Springer, 2013, pp. 58–65.
[16] J. Cabessa, A.E.P. Villa, An attractor-based complexity measurement for boolean recurrent neural networks, PLoS ONE 9 (4) (2014) e94204.
[17] J. Cabessa, A.E.P. Villa, Interactive evolving recurrent neural networks are super-Turing universal, in: Proceedings of ICANN 2014, in: Lecture Notes in Computer Science, vol. 8681, Springer, 2014, pp. 57–64.
[18] J. Cabessa, A.E.P. Villa, Computational capabilities of recurrent neural networks based on their attractor dynamics, in: Proceedings of IJCNN 2015, IEEE, 2015, pp. 1–8.
[19] N. Caporale, Y. Dan, Spike timing-dependent plasticity: a Hebbian learning rule, Annu. Rev. Neurosci. 31 (2008) 25–46.
[20] A. Celletti, A.E.P. Villa, Determination of chaotic attractors in the rat brain, J. Stat. Phys. 84 (5) (1996) 1379–1385.
[21] A. Celletti, A.E.P. Villa, Low-dimensional chaotic attractors in the rat brain, Biol. Cybern. 74 (5) (1996) 387–393.
[22] R. Cicurel, M.A.L. Nicolelis, The Relativistic Brain: How It Works and Why It Cannot Be Simulated by a Turing Machine, Kios Press, 2015.
[23] B.J. Copeland, Hypercomputation, Minds Mach. 12 (4) (2002) 461–502.
[24] B.J. Copeland, Hypercomputation: philosophical issues, Theor. Comput. Sci. 317 (1–3) (2004) 251–267.
[25] J. Duparc, Wadge hierarchy and Veblen hierarchy part I: Borel sets of finite rank, J. Symb. Log. 66 (1) (2001) 56–86.
[26] J.L. Eriksson, A.E.P. Villa, Learning of auditory equivalence classes for vowels by rats, Behav. Process. 73 (2006) 348–359.
[27] D. Goldin, S.A. Smolka, P.C. Attie, E.L. Sonderegger, Turing machines, transition systems, and interaction, Inf. Comput. 194 (2004) 101–128.
[28] J. Iglesias, O.K. Chibirova, A.E.P. Villa, Nonlinear dynamics emerging in large scale neural networks with ontogenetic and epigenetic processes, Lecture Notes in Computer Science 4668 (2007) 579–588.
[29] J. Iglesias, A.E.P. Villa, Recurrent spatiotemporal firing patterns in large spiking neural networks with ontogenetic and epigenetic processes, J. Physiol. (Paris) 104 (3–4) (2010) 137–146.
[30] G.M. Innocenti, Exuberant development of connections, and its possible permissive role in cortical evolution, Trends Neurosci. 18 (9) (1995) 397–402.
[31] A.S. Kechris, Classical Descriptive Set Theory, Graduate Texts in Mathematics, vol. 156, Springer-Verlag, New York, 1995.
[32] J. Kilian, H.T. Siegelmann, The dynamic universality of sigmoidal neural networks, Inf. Comput. 128 (1) (1996) 48–56.
[33] S.C. Kleene, Representation of events in nerve nets and finite automata, in: C. Shannon, J. McCarthy (Eds.), Automata Studies, Princeton University Press, Princeton, NJ, 1956, pp. 3–41.
[34] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (1943) 115–133.
[35] M.L. Minsky, Computation: Finite and Infinite Machines, Prentice–Hall, Inc., Englewood Cliffs, NJ, 1967.
[36] J.v. Neumann, The Computer and The Brain, Yale University Press, New Haven, CT, USA, 1958.
[37] T. Ord, The many forms of hypercomputation, Appl. Math. Comput. 178 (1) (2006) 143–153.
[38] D. Perrin, J.-E. Pin, Infinite Words – Automata, Semigroups, Logic and Games, Pure and Applied Mathematics, vol. 141, Elsevier, 2004.
[39] H.T. Siegelmann, Neural Networks and Analog Computation: Beyond the Turing Limit, Birkhauser Boston Inc., Cambridge, MA, USA, 1999.
[40] H.T. Siegelmann, E.D. Sontag, Analog computation via neural networks, Theor. Comput. Sci. 131 (2) (1994) 331–360.
[41] H.T. Siegelmann, E.D. Sontag, On the computational power of neural nets, J. Comput. Syst. Sci. 50 (1) (1995) 132–150.
[42] J. Síma, P. Orponen, General-purpose computation with neural networks: a survey of complexity theoretic results, Neural Comput. 15 (12) (2003) 2727–2778.
[43] L. Staiger, ω-languages, in: Handbook of Formal Languages, vol. 3: Beyond Words, Springer-Verlag New York, Inc., New York, NY, USA, 1997, pp. 339–387.
[44] M. Stannett, Computation and hypercomputation, Minds Mach. 13 (1) (2003) 115–153.
[45] M. Stannett, The case for hypercomputation, Appl. Math. Comput. 178 (1) (2006) 8–24.
[46] W. Thomas, Automata on infinite objects, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics, Elsevier and MIT Press, 1990, pp. 133–192.
[47] A.M. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proc. Lond. Math. Soc. 2 (42) (1936) 230–265.
[48] A.M. Turing, Intelligent machinery. Technical report, National Physical Laboratory, Teddington, UK, 1948.
[49] E. Vaadia, I. Haalman, M. Abeles, H. Bergman, Y. Prut, H. Slovin, A. Aertsen, Dynamics of neuronal interactions in monkey cortex in relation to behavioural events, Nature 373 (6514) (1995) 515–518.
[50] J. van Leeuwen, J. Wiedermann, On algorithms and interaction, in: M. Nielsen, B. Rovan (Eds.), Mathematical Foundations of Computer Science 2000, in: Lecture Notes in Computer Science, vol. 1893, Springer, Berlin–Heidelberg, 2000, pp. 99–113.
[51] A.E.P. Villa, Empirical evidence about temporal structure in multi-unit recordings, in: R. Miller (Ed.), Time and the Brain, in: Conceptual Advances in Brain Research, vol. 3, Harwood Academic, Amsterdam, The Netherlands, 2000, pp. 1–51.
[52] A.E.P. Villa, Spatio-temporal patterns of spike occurrences in feely-moving rats associated to perception of human vowels, in: R. König, P. Heil, E. Budinger, H. Scheich (Eds.), Auditory Cortex: Towards a Synthesis of Human and Animal Research, Lawrence Erlbaum, Mahwah, NJ, 2005, pp. 275–294.
[53] A.E.P. Villa, M. Abeles, Evidence for spatiotemporal firing patterns within the auditory thalamus of the cat, Brain Res. 509 (2) (1990) 325–327.
[54] A.E.P. Villa, J.M. Fuster, Temporal correlates of information processing during visual short-term memory, NeuroReport 3 (1) (1992) 113–116.
[55] A.E.P. Villa, I.V. Tetko, Spatio-temporal patterns of activity controlled by system parameters in a simulated thalamo-cortical neural network, in: H. Herrmann, D. Wolf, E. Poppel (Eds.), Supercomputing in Brain Research: From Tomography to Neural Networks, World Scientific, 1995, pp. 379–388.
[56] A.E.P. Villa, I.V. Tetko, A. Celletti, A. Riehle, Chaotic dynamics in the primate motor cortex depend on motor preparation in a reaction-time task, Current Psychology of Cognition 17 (1998) 763–780.
[57] A.E.P. Villa, I.V. Tetko, B. Hyland, A. Najem, Spatiotemporal activity patterns of rat cortical neurons predict responses in a conditioned task, Proc. Natl. Acad. Sci. USA 96 (3) (1999) 1106–1111.
[58] W.W. Wadge, Reducibility and determinateness on the Baire space, PhD thesis, University of California, Berkeley, 1983.
[59] K. Wagner, On ω-regular sets, Inf. Control 43 (2) (1979) 123–177.
[60] P. Wegner, Interactive foundations of computing, Theor. Comput. Sci. 192 (1998) 315–351.