The Expressive Power of Analog Recurrent Neural Networks on Infinite Input Streams

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

[a] *Grenoble Institute of Neuroscience, Inserm UMRS 836, University Joseph Fourier, FR-38041 Grenoble, France*

Abstract

We consider analog recurrent neural networks working on infinite input streams, provide a complete topological characterization of their expressive power, and compare it to the expressive power of classical infinite word reading abstract machines. More precisely, we consider analog recurrent neural networks as language recognizers over the Cantor space, and prove that the classes of ω-languages recognized by deterministic and non-deterministic analog networks correspond precisely to the respective classes of $\mathbf{\Pi_2^0}$-sets and $\mathbf{\Sigma_1^1}$-sets of the Cantor space. Furthermore, we show that the result can be generalized to more expressive analog networks equipped with any kind of Borel accepting condition. Therefore, in the deterministic case, the expressive power of analog neural nets turns out to be comparable to the expressive power of any kind of Büchi abstract machine, whereas in the non-deterministic case, analog recurrent networks turn out to be strictly more expressive than any other kind of Büchi or Muller abstract machine, including the main cases of classical automata, 1-counter automata, k-counter automata, pushdown automata, and Turing machines.

Keywords: Analog neural networks, Analog computation, Topology, Borel sets, Analytic sets, ω-Automata, Turing machines.
2000 MSC: 92B20, 68Q05, 68Q10, 68Q45

1. Introduction

Understanding the dynamical and computational capabilities of neural networks is an issue of central importance in neural computation. It is related to the fields of artificial intelligence, machine learning, and bio-inspired computing, and from a purely theoretical point of view, it directly contributes to a better global comprehension of biological intelligence. In this context, an interesting comparative approach that has been pursued consists in trying to understand the fundamental differences and similarities that exist between the processes of

transfer of information in biological systems on the one side and in artificial devices on the other. Towards this purpose, much interest has been focused on comparing the computational capabilities of diverse theoretical neural models and abstract computing devices. Two main distinct neural computational approaches have been considered in this respect: the digital neural computation, and the continuous-valued neural computation [1].

On the one hand, the field of digital neural computation assumes that the computational capabilities of the brain rely mainly on the discrete spiking nature of neurons. The approach was controversially initiated by McCulloch and Pitts, who proposed a modelization of the nervous system as a finite interconnection of logical devices [2]. In this context, neural networks are commonly considered as discrete abstract machines, and the issue of computational capabilities of neural models is commonly investigated from the automata-theoretic perspective. For instance, Kleene and Minsky early proved that the class of rational-weighted first-order recurrent neural networks equipped with boolean activation functions admits equivalent computational capabilities as classical finite state automata [3, 4]. Later, Siegelmann and Sontag showed that, still assuming rational synaptic weights, but considering a saturated-linear sigmoidal instead of a boolean hard-threshold activation function drastically increases the computational power of the networks from finite state automata up to Turing capabilities [5, 6]. Kilian and Siegelmann then extended the Turing universality of neural networks to a more general class of sigmoidal activation functions [7].

On the other hand, the field of continuous-valued neural computation – or analog neural computation – assumes that the continuous variables appearing in the underlying chemical and physical phenomena are essentially involved in the computational capabilities of the brain. In this context, Siegelmann and Sontag introduced the concept of an *analog recurrent neural network* as a neural net equipped with real synaptic weights and linear-sigmoid activation functions [8]. They further showed that the computational capabilities of such networks turn out to strictly surpass the Turing limits [9, 10, 8]. More precisely, the class of analog recurrent neural networks was proved to disclose unbounded power if exponential time of computation is allowed, and to recognize in polynomial time the same class of languages as those recognized by Turing machines that consult sparse oracles in polynomial time (the class P/Poly) [8]. These considerations led Siegelmann and Sontag to propose the concept of analog recurrent neural networks as standard in the field of analog computation [9].

However, in both digital or continuous approaches, the computational behavior of the networks has generally been approached from the point of view of classical finite computation theory [11]: a network is as an abstract machine that receives a finite input stream from its environment, processes this input, and then provides a corresponding finite output stream as answer, without any consideration to the internal or external changes that might happen during previous computations. But this classical computational approach is inherently restrictive, especially when it refers to bio-inspired complex information processing systems. Indeed, in the brain (or in organic life in general), previous experience must affect the perception of future inputs, and older memories themselves may

change with response to new inputs. Neural networks should thus rather be conceived as provided with memory that remains active throughout the whole computational process, rather than proceeding in a closed-box amnesic classical fashion. Hence, in order to take into account this persistence of memory, a possible approach consists in investigating the computational behavior of recurrent neural networks from the growing perspective of infinite computational abstract machines, as for instance presented in [12, 13, 14]. A first step in this direction has already been initiated by Cabessa and Villa who proposed a hierarchical classification of recurrent boolean networks on infinite input streams based on their attractive properties [15].

The present paper pursues this research direction by providing a characterization of the computational power of analog recurrent neural networks working on infinite input streams. More precisely, we consider analog recurrent neural networks as language recognizers over the Cantor space, and prove that the classes of languages recognized by deterministic and non-deterministic such networks exhaust the respective classes of $\mathbf{\Pi_2^0}$-sets and $\mathbf{\Sigma_1^1}$-sets of the Cantor space. Furthermore, we show that the result can be generalized to more expressive networks, by giving an upper topological bound on the expressive power of analog neural networks equipped with any kind of Borel accepting condition. Therefore, in the deterministic case, the expressive power of analog neural nets turns out to be comparable to the expressive power of any kind of $\mathbf{X}$-automata (i.e. finite automata equipped with a storage type $\mathbf{X}$, see [16]) equipped with Büchi acceptance condition, whereas in the non-deterministic case, analog recurrent networks turn out to be strictly more expressive than any other kind of Büchi or Muller $\mathbf{X}$-automata, including the main cases of classical automata, 1-counter automata, k-counter automata, pushdown automata, and Turing machines [16, 17, 18]. Hence, the present work provides an extension to the context of infinite computation of the study on the computational capabilities of analog neural networks pursued by Siegelmann and Sontag [8].

2. Preliminaries

All definitions and facts presented in this section can be found in [19, 20, 21]. First of all, as usual, we let $\{0,1\}^*$, $\{0,1\}^+$, and $\{0,1\}^\omega$ denote respectively the set of finite words, non-empty finite words, and infinite words, all of them over the alphabet $\{0,1\}$. Then, for any $x \in \{0,1\}^*$, the length of x corresponds to the number of letters contained in x and will be denoted by $|x|$. The empty word is denoted λ and has length 0, and every infinite word has length ∞. Moreover, if x is a non-empty word, for any $0 \leq i \leq |x| - 1$, the $(i+1)$-th letter of x will be denoted by $x(i)$. Hence, any $x \in \{0,1\}^+$ and $y \in \{0,1\}^\omega$ can be written as $x = x(0)x(1)\cdots x(|x|-1)$ and $y = y(0)y(1)y(2)\cdots$, respectively. Moreover, the concatenation of x and y will be denoted by xy, and if X and Y are subsets of $\{0,1\}^*$, the concatenation of X and Y is defined by $XY = \{xy : x \in X \text{ and } y \in Y\}$. The fact that x is a prefix (resp. strict prefix) of y will be denoted by $x \subseteq y$ (resp. $x \subsetneq y$). Then, for any $p \in \{0,1\}^*$, we set

$p\{0,1\}^\omega = \{x \in \{0,1\}^\omega : p \subsetneq x\}$. Finally, a subset of $\{0,1\}^\omega$ is generally called an ω-language.

The space $\{0,1\}^\omega$ can naturally be equipped with the product topology of the discrete topology on $\{0,1\}$. The obtained topological space is commonly called the *Cantor space*. The topology on $\{0,1\}^\omega$ is actually given by the metric $d : \{0,1\}^\omega \times \{0,1\}^\omega \longrightarrow [0,1]$ defined by $d(u,v) = 2^{-r}$, where $r = \min\{n : u(n) \neq v(n)\}$, and with the usual conventions $\min \emptyset = \infty$ and $2^{-\infty} = 0$. Accordingly, the basic open sets of $\{0,1\}^\omega$ are of the form $p\{0,1\}^\omega$, for some prefix $p \in \{0,1\}^+$, and the general open sets of $\{0,1\}^\omega$ are of the form $\bigcup_{i\in I} p_i\{0,1\}^\omega$, where $I \subseteq \mathbb{N}$ and each $p_i \in \{0,1\}^+$.

The class *Borel* subsets of $\{0,1\}^\omega$, denoted by $\mathbf{\Delta^1_1}$, consists of the smallest collection of subsets of $\{0,1\}^\omega$ containing all open sets and closed under countable union and complementation. Now, if ω_1 denote the first uncountable ordinal, then for each $0 < \alpha < \omega_1$, one defines by transfinite induction the following classes of Borel sets: $\mathbf{\Sigma^0_1} = \{X \subseteq \{0,1\}^\omega : X \text{ is open}\}$, $\mathbf{\Pi^0_\alpha} = \{X \subseteq \{0,1\}^\omega : X^c \in \mathbf{\Sigma^0_\alpha}\}$, $\mathbf{\Delta^0_\alpha} = \mathbf{\Sigma^0_\alpha} \cap \mathbf{\Pi^0_\alpha}$, and for $\alpha > 1$, $\mathbf{\Sigma^0_\alpha} = \{X \subseteq \{0,1\}^\omega : X = \bigcup_{n\geq 0} X_n,\ X_n \in \mathbf{\Pi^0_{\alpha_n}},\ \alpha_n < \alpha,\ n \in \mathbb{N}\}$. For each $0 < \alpha < \omega_1$, it can be shown that the strict inclusion relations $\mathbf{\Delta^0_\alpha} \subsetneq \mathbf{\Sigma^0_\alpha} \subsetneq \mathbf{\Delta^0_{\alpha+1}}$ and $\mathbf{\Delta^0_\alpha} \subsetneq \mathbf{\Pi^0_\alpha} \subsetneq \mathbf{\Delta^0_{\alpha+1}}$ both hold. Moreover, one has

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

The collection of all classes $\mathbf{\Sigma^0_\alpha}$, $\mathbf{\Pi^0_\alpha}$, and $\mathbf{\Delta^0_\alpha}$ thus provides a stratification of the whole class of Borel sets into ω_1 distinct levels known as *the Borel hierarchy*. The *rank* of a Borel set $X \subseteq \{0,1\}^\omega$ then consists of the smallest ordinal α such that $X \in \mathbf{\Sigma^0_\alpha} \cup \mathbf{\Pi^0_\alpha} \cup \mathbf{\Delta^0_\alpha}$. The rank of X represents 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.

In the sequel, the set $F_\infty \subseteq \{0,1\}^\omega$ consisting of all infinite words that contain infinitely many occurrences of 1's will be of specific interest. Note that $F_\infty = \bigcap_{n\geq 0}\bigcup_{m\geq 0}\{0,1\}^{n+m}1\{0,1\}^\omega$ (with the convention that $\{0,1\}^0 = \lambda$), and therefore $F_\infty \in \mathbf{\Pi^0_2}$.

Now, the product space $\{0,1\}^\omega \times \{0,1\}^\omega$ can naturally be equipped with the product topology of the space $\{0,1\}^\omega$. The topology on $\{0,1\}^\omega \times \{0,1\}^\omega$ is actually given by the metric d' defined by $d'((x_1,x_2),(y_1,y_2)) = \frac{1}{2}\cdot d(x_1,y_1) + \frac{1}{2}\cdot d(x_2,y_2)$. The basic open sets of $\{0,1\}^\omega \times \{0,1\}^\omega$ are of the form $p_1\{0,1\}^\omega \times p_2\{0,1\}^\omega$, for some prefixes $p_1, p_2 \in \{0,1\}^+$, and the general open sets are thus of the form $\bigcup_{i\in I}(p_{i,1}\{0,1\}^\omega \times p_{i,2}\{0,1\}^\omega)$, where $I \subseteq \mathbb{N}$ and each $p_{i,j} \in \{0,1\}^+$. The definitions of Borel sets and Borel classes $\mathbf{\Sigma^0_\alpha}$, $\mathbf{\Pi^0_\alpha}$, and $\mathbf{\Delta^0_\alpha}$ can naturally be transposed in this case.

Furthermore, a function $f : \{0,1\}^\omega \longrightarrow \{0,1\}^\omega$ is said to be *continuous* if the preimage by f of any open set is open. In fact, if f is continuous and $X \in \mathbf{\Sigma^0_\alpha}$ (resp. $X \in \mathbf{\Pi^0_\alpha}$, $X \in \mathbf{\Delta^0_\alpha}$), then $f^{-1}(X) \in \mathbf{\Sigma^0_\alpha}$ (resp. $f^{-1}(X) \in \mathbf{\Pi^0_\alpha}$, $f^{-1}(X) \in \mathbf{\Delta^0_\alpha}$). Also, the function f is said to be *Lipschitz of modulus* k if for

any $x, y \in \{0,1\}^\omega$, one has $d(f(x), f(y)) \leq kd(x, y)$. The same definition and result hold for any function $g : \{0,1\}^\omega \times \{0,1\}^\omega \longrightarrow \{0,1\}^\omega$.

Now, a set $X \subseteq \{0,1\}^\omega$ is said to be *analytic* iff it is the projection of some $\mathbf{\Pi_2^0}$-set $Y \subseteq \{0,1\}^\omega \times \{0,1\}^\omega$, i.e. $X = \pi_1(Y) = \{x \in \{0,1\}^\omega : \exists y \ (x, y) \in Y\}$. Equivalently, $X \subseteq \{0,1\}^\omega$ is analytic iff it is the projection of some Borel set $Y \subseteq \{0,1\}^\omega \times \{0,1\}^\omega$. The class of analytic sets of $\{0,1\}^\omega$ is denoted by $\mathbf{\Sigma_1^1}$. It can be shown that the class of analytic sets strictly contains the class of all Borel sets, namely $\mathbf{\Delta_1^1} \subsetneq \mathbf{\Sigma_1^1}$. Finally, a set $X \subseteq \{0,1\}^\omega$ is *effectively analytic* iff it is recognized by some Turing machine with a Büchi or Muller acceptance condition [18]. The class of effective analytic sets of $\{0,1\}^\omega$ is denoted by Σ_1^1 (lightface). The relation $\Sigma_1^1 \subsetneq \mathbf{\Sigma_1^1}$ holds [20].

3. The Model

In this work, we assume that the dynamics of the neural network is synchronous. The rationale for this assumption is twofold. Firstly, the experimental observation of neural network activity is usually carried out by multiple extracellular recordings of the time series of the neuronal discharges. Such multivariate time series is discrete, usually with time steps in the range 0.1-1 ms, and provides many significant insights into neural network dynamics even when considering multiple time scales [22]. Moreover, the observation of recurrent firing patterns, with jitters in the order of few milliseconds at most, suggests that synchronous dynamics is likely to exist in brain circuits [23, 24, 25, 26]. Secondly, our aim is to extend the infinite input stream context of the synchronous analog recurrent neural network model introduced by Siegelmann and Sontag [8, 10]. We will then provide a complete characterization of the expressive power of deterministic and non-deterministic such networks. For this purpose, the concept of an analog recurrent neural network shall first be recalled.

An *analog recurrent neural network* (ARNN) consists of a synchronous network of neurons (or processors) whose architecture is specified by a general directed graph. The network contains a finite number of internal neurons $(x_j)_{j=1}^N$ as well as a finite number of external input cells $(u_j)_{j=1}^M$ transmitting to the net the input sent by the environment. The networks also admits a specification of p particular output cells $(x_{j_k})_{k=1}^p$ among the internal neurons $(x_j)_{j=1}^N$ that are used to communicate the output of the network to the environment. The cells of the network are related together by real-weighted synaptic connections, and might receive an external real-weighted bias from the external environment. At each time step, the activation value of every neuron is updated by applying a linear-sigmoid function to some real-weighted affine combination the cells' activation values at previous time step. More precisely, given the activation values of cells $(x_j)_{j=1}^N$ and $(u_j)_{j=1}^N$ at time t, the activation value of each cell x_i at time $t+1$ is then updated by the following equation

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

where all a_{ij}, b_{ij}, and c_i are real synaptic weights, and σ is the classical saturated-linear activation function defined by

$$\sigma(x) = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{if } 0 \leq x \leq 1, \\ 1 & \text{if } x > 1. \end{cases}$$

Equation (1) ensures that the whole dynamics of a ARNN can be described by some governing equation of the form

$$x(t\vec{+}1) = \sigma\left(A \cdot x(\vec{t}) + B \cdot u(\vec{t}) + \vec{c}\right), \tag{2}$$

where $x(\vec{t}) = (x_1(t), \ldots, x_N(t))$ and $u(\vec{t}) = (u_1(t), \ldots, u_M(t))$ are real-valued vectors describing the activation values of the internal neurons and external input cells at time t, σ denotes the saturated-linear function applied component by component, and A, B, and $\vec{c}$ are real-valued matrices.

Siegelmann and Sontag studied the computational complexity of ARNNs by considering them as language recognizers over the space of non-empty finite words of bits $\{0,1\}^+$. For this purpose, they restricted their attention to ARNNs where each input and output channels were forced to carry only binary signals, and they proceeded to a rigid encoding of the binary input signal and binary output answer to this signal via the existence of two input cells and two output cells (taking the role of a data line and a validation line for each case). Following these conventions, any ARNN $\mathcal{N}$ could be associated to some neural language $L(\mathcal{N}) \subseteq \{0,1\}^+$, called the language *recognized* by $\mathcal{N}$, and defined as the set of all finite words of bits that could be *positively classified* by $\mathcal{N}$ in some finite time of computation. The class of ARNNs was then shown disclose super-Turing computational capabilities. More precisely, ARNNs turn out to admit unbounded power (in the sense of being capable of recognizing all possible languages of $\{0,1\}^+$) in exponential time of computation. And when restricted to polynomial time of computation, ARNNs are computationally equivalent to polynomial time Turing machines with polynomially long advice, and thus recognize the complexity class of languages called P/poly [8].

Here, we provide a natural generalization of this situation to the context of ω-words. More precisely, we consider ARNNs as language recognizers over the space of infinite words of bits $\{0,1\}^\omega$. For this purpose, we suppose any ARNN $\mathcal{N}$ to be provided with only a single input cell u as well as a single output cell y, both being forced to carry only binary signals. More precisely, at each time step $t \geq 0$, the two cells u and y admit activation values $u(t)$ and $y(t)$ belonging to $\{0,1\}$. Then, assuming the initial activation vector of the network to be $x(\vec{0}) = \vec{0}$, any infinite input stream

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

processed bit by bit induces via Equation (2) a corresponding infinite output stream

$$o_s = (y(t))_{t \in \mathbb{N}} = y(0)y(1)y(2)\cdots \in \{0,1\}^\omega$$

also processed bit by bit. After ω time steps, an infinite input stream s will then said to be *accepted* by $\mathcal{N}$ iff the corresponding output stream o_s contains infinitely many $1's$, or in other words, iff $o_s \in F_\infty$. This Büchi-like accepting condition corresponds to the natural translation in the present context of the classical Büchi accepting condition for infinite word reading machines [12, 13]. It refers to the fact that an infinite input stream is considered to be meaningful for a given network iff the corresponding output remains forever active. The set of all infinite words accepted by $\mathcal{N}$ will then be called the *neural language* of $\mathcal{N}$, and will be denoted by $L(\mathcal{N})$. Moreover, a language $L \subseteq \{0,1\}^\omega$ will said to be *recognizable* if there exists some such ARNN $\mathcal{N}$ such that $L(\mathcal{N}) = L$. From this point onwards, any ARNN over infinite input streams satisfying the above conditions will be referred to as a *deterministic ω-ARNN*.

According to the preceding definitions, any deterministic ω-ARNN $\mathcal{N}$ can naturally be identified with the function $f_\mathcal{N} : \{0,1\}^\omega \longrightarrow \{0,1\}^\omega$ defined by $f_\mathcal{N}(s) = o_s$, where o_s is the output generated by $\mathcal{N}$ when input s is received. One then has by definition that $s \in L(\mathcal{N})$ iff $o_s \in F_\infty$ iff $f_\mathcal{N}(s) \in F_\infty$, implying that the neural language of $\mathcal{N}$ can thus be written as

$$L(\mathcal{N}) = f_\mathcal{N}^{-1}(F_\infty).$$

Moreover, the following result shows that the dynamics of deterministic ω-ARNNs impose strong restrictions on the functional behaviors of such networks.

Lemma 1. *Let $\mathcal{N}$ be some deterministic ω-ARNN. Then the corresponding function $f_\mathcal{N}$ is Lipschitz.*

Proof. First of all, the dynamics of $\mathcal{N}$ ensures that for any input s and corresponding output o_s, and for any $k \geq 0$, the two bits $s(k)$ and $o_s(k)$ are always generated simultaneously at time step k. Now, let $s_1, s_2 \in \{0,1\}^\omega$, and let $o_{s_1} = f_\mathcal{N}(s_1)$ and $o_{s_2} = f_\mathcal{N}(s_2)$. If $s_1 \neq s_2$, the metric over $\{0,1\}^\omega$ ensures that $d(s_1, s_2) = 2^{-n}$ for some $n \geq 0$. Now, the above argument shows that the relation $d(s_1, s_2) = 2^{-n}$ implies $d(o_{s_1}, o_{s_2}) = d(f_\mathcal{N}(s_1), f_\mathcal{N}(s_2)) \leq 2^{-n}$. Therefore $d(f_\mathcal{N}(s_1), f_\mathcal{N}(s_2)) \leq d(s_1, s_2)$. If $s_1 = s_2$, then $f_\mathcal{N}(s_1) = f_\mathcal{N}(s_2)$ since $f_\mathcal{N}$ is a function. It follows that $d(f_\mathcal{N}(s_1), f_\mathcal{N}(s_2)) = 0 = d(s_1, s_2)$, and thus $d(f_\mathcal{N}(s_1), f_\mathcal{N}(s_2)) \leq d(s_1, s_2)$ in this case also. Therefore, $f_\mathcal{N}$ is Lipschitz of modulus 1. □

4. The Expressive Power of Deterministic ω-ARNN

In this section, we provide a complete characterization of the expressive power of the class of deterministic ω-ARNNs. More precisely, we prove that deterministic ω-ARNNs recognize precisely all $\mathbf{\Pi_2^0}$-sets of $\{0,1\}^\omega$, and no other ones. The result is not so surprising. The fact that any neural language recognized by some deterministic ω-ARNN is a $\mathbf{\Pi_2^0}$-set results from the chosen Büchi-like accepting condition of the networks. Conversely, the fact that any $\mathbf{\Pi_2^0}$-set can be recognized by some ω-ARNN is more technical and results from the possibility of encoding every such subset into the real synaptic weights of

the networks. Therefore, the expressive power of deterministic analog neural nets turn out to be closely related to the expressive power of any kind of deterministic Büchi **X**-automata, since the ω-languages recognized such machines also all belong to $\mathbf{\Pi_2^0}$ [16, 18]. In particular, deterministic ω-ARNNs admit a similar expressive power as deterministic Büchi automata, 1-counter automata, k-counter automata, pushdown automata, Petri-nets, and Turing machines.

To begin with, we show that the dynamics and accepting conditions of deterministic ω-ARNNs ensure that every neural language recognized by such a network is indeed a $\mathbf{\Pi_2^0}$-set.

Proposition 1. *Let $\mathcal{N}$ be some deterministic ω-ARNN. Then $L(\mathcal{N}) \in \mathbf{\Pi_2^0}$.*

Proof. First of all, recall that F_∞ is a $\mathbf{\Pi_2^0}$-set. Moreover, Lemma 1 shows that the function $f_\mathcal{N}$ is Lipschitz, thus continuous. Therefore $L(\mathcal{N}) = f_\mathcal{N}^{-1}(F_\infty) \in \mathbf{\Pi_2^0}$, [19, 20, 21]. □

Conversely, we now prove that any $\mathbf{\Pi_2^0}$-set of $\{0,1\}^\omega$ can be recognized by some deterministic ω-ARNN. For this purpose, we adopt an encoding approach as described in [8], but we stay close to the classical topological definition of $\mathbf{\Pi_2^0}$-sets instead of considering them from the point of view of circuit theory, as for instance in [27, 28]. More precisely, we show that the belonging problem to any $\mathbf{\Pi_2^0}$-set can be decided by some deterministic ω-ARNN containing a suitable real synaptic weight which encodes the given set.

We first need to provide a suitable encoding of $\mathbf{\Pi_2^0}$-sets of the Cantor space. Hence, consider a set $X \subseteq \{0,1\}^\omega$ such that $X \in \mathbf{\Pi_2^0}$. By definition, X can be written as a countable intersection of open sets, or equivalently, as a countable intersection of countable unions of basic open sets, i.e. $X = \bigcap_{i\geq 0} \bigcup_{j\geq 0} p_{(i,j)}\{0,1\}^\omega$, where each $p_{(i,j)} \in \{0,1\}^+$. The set X is thus completely determined by the countable sequence of finite prefixes $(p_{(i,j)})_{i,j\geq 0}$. Hence, in order to encode the subset X into some real number, it suffices to encode the corresponding sequence of prefixes $(p_{(i,j)})_{i,j\geq 0}$.

For this purpose, we adopt a similar approach as the one described in [8]. First of all, each finite prefix $p_{(i,j)} \in \{0,1\}^+$ is first encoded by some finite sequence of natural numbers $\ulcorner p_{(i,j)} \urcorner \in \{0,2,4\}^+$ obtained by first adding a 4 in front of the sequence $p_{(i,j)}$ and then doubling each of its bit, i.e.

$$\ulcorner p_{(i,j)} \urcorner(0) = 4 \text{ and } \ulcorner p_{(i,j)} \urcorner(k+1) = 2 \cdot p_{(i,j)}(k), \text{ for all } k < |p_{(i,j)}|.$$

For instance, $\ulcorner 010011 \urcorner = 4020022$. Now, let us consider some primitive recursive bijection from $\mathbb{N}^2$ onto $\mathbb{N}$, like for instance $b : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$ given by $b(i,j) = \frac{1}{2} \cdot (i+j) \cdot (i+j+1) + j$. Then, the sequence of prefixes $(p_{(i,j)})_{i,j\geq 0}$ can be encoded by the infinite sequence of integers $\ulcorner (p_{(i,j)})_{i,j\geq 0} \urcorner \in \{0,2,4\}^\omega$ defined by the successive concatenation of all finite sequences $\ulcorner p_{b^{-1}(k)} \urcorner$, for all $k \geq 0$, namely

$$\ulcorner (p_{(i,j)})_{i,j\geq 0} \urcorner = \ulcorner p_{b^{-1}(0)} \urcorner \ulcorner p_{b^{-1}(1)} \urcorner \ulcorner p_{b^{-1}(2)} \urcorner \cdots .$$

Using this first encoding, each finite block $\ulcorner p_{b^{-1}(k)} \urcorner$ can now be unambiguously re-encoded by the rational number $r(p_{b^{-1}(k)}) \in [0,1]$ given by the interpretation

of the sequence in base 5, namely

$$r(p_{b^{-1}(k)}) = \sum_{i=0}^{|\ulcorner p_{b^{-1}(k)} \urcorner|-1} \frac{\ulcorner p_{b^{-1}(k)} \urcorner(i)}{5^{i+1}}.$$

Finally, the set X itself can also be unambiguously re-encoded by the real number $r(X) \in [0,1]$ given by the interpretation of the infinite sequence $\ulcorner (p_{(i,j)})_{i,j\geq 0} \urcorner$ in base 5, namely

$$r(X) = \sum_{i=0}^{\infty} \frac{\ulcorner (p_{(i,j)})_{i,j\geq 0} \urcorner(i)}{5^{i+1}}.$$

Now, in order to prove our result, a preliminary lemma is first needed. More precisely, a direct generalization of [8, Lemma 3.2] shows that, for any $\mathbf{\Pi_2^0}$-set, there exists a corresponding ARNN which, given some suitable encoding of the integer k as input, is able to retrieve the rational encoding $r(p_{b^{-1}(k)})$ of the $(k+1)$-th block of the sequence $\ulcorner (p_{(i,j)})_{i,j\geq 0} \urcorner$ as output. Note that the following lemma states the existence of a general ARNN as described in [8], and not of some deterministic ω-ARNN.

Lemma 2. *Let $X \subseteq \{0,1\}^\omega$ be a $\mathbf{\Pi_2^0}$-set with corresponding sequence of prefixes $(p_{(i,j)})_{i,j\geq 0}$. Then there exists an ARNN $\mathcal{N}_{r(X)}$ containing one input cell, one output cell, and a synaptic real weight equal to $r(X)$, and such that, starting from the zero initial state, and given the input signal $(1-2^{-k})0^\omega$ produces an output of the form $0^* r(p_{b^{-1}(k)})0^\omega$.*

Proof. We only give a sketch of the proof, since it is a direct generalization of [8, Lemma 3.2]. The idea is that the network first stores the integer k in memory. Then, the network decodes step by step the infinite sequence $\ulcorner (p_{(i,j)})_{i,j\geq 0} \urcorner$ from its synaptic weight $r(X)$ until reaching the $(k+1)$-th letter 4 of that sequence. After that, the network knows that it has reached the suitable block $\ulcorner p_{b^{-1}(k)} \urcorner$ of the sequence $\ulcorner (p_{(i,j)})_{i,j\geq 0} \urcorner$, and proceeds to a re-encoding of that block into $r(p_{b^{-1}(k)})$. Once finished, the obtained rational number $r(p_{b^{-1}(k)})$ is finally provided as output. The technicalities of the proof reside in showing that the decoding and encoding procedures are indeed performable by such an ARNN. This property results from the fact that these procedures are recursive, and any recursive function can be simulated by some rational-weighted network, as shown in [6], and thus a fortiori by some ARNN also. □

It follows from the preceding result that any $\mathbf{\Pi_2^0}$-set can indeed be recognized by some deterministic ω-ARNN, as shown by the following result.

Proposition 2. *Let $X \subseteq \{0,1\}^\omega$ such that $X \in \mathbf{\Pi_2^0}$. Then there exists a deterministic ω-ARNN $\mathcal{N}_X$ such that $L(\mathcal{N}_X) = X$.*

Proof. The set $X \in \mathbf{\Pi_2^0}$ can be written as $X = \bigcap_{i\geq 0} \bigcup_{j\geq 0} p_{(i,j)}\{0,1\}^\omega$, where each $p_{(i,j)} \in \{0,1\}^+$. Hence, a given infinite input $s \in \{0,1\}^\omega$ belongs to X iff for all index $i \geq 0$ there exists an index $j \geq 0$ such that $s \in p_{(i,j)}\{0,1\}^\omega$, or

equivalently, iff for all $i \geq 0$ there exists $j \geq 0$ such that $p_{(i,j)} \subsetneq s$. Consequently, the problem of determining whether some input s provided bit by bit belongs to X or not can be decided in infinite time by the procedure described by Algorithm 1, which, after ω computation steps, would have returned infinitely many 1's iff $s \in \bigcap_{i\geq 0} \bigcup_{j\geq 0} p_{(i,j)}\{0,1\}^\omega = X$. In this procedure, observe that

Algorithm 1

1: Input s is provided bit by bit at successive time steps
2: $i \leftarrow 0$, $j \leftarrow 0$
3: **loop**
4: $k \leftarrow b(i,j)$
5: Submit input $(1-2^{-k})$ to $\mathcal{N}_{r(X)}$ `// where` $\mathcal{N}_{r(X)}$ `is given by Lemma 2`
6: Get output $r(p_{b^{-1}(k)})$ from $\mathcal{N}_{r(X)}$
7: $p_{b^{-1}(k)} \leftarrow decode(r(p_{b^{-1}(k)}))$
8: **if** $p_{b^{-1}(k)} \subsetneq s$ **then** `// in this case,` $s \in p_{(i,j)}\{0,1\}^\omega$
9: **return** 1 `// hence` $s \in \bigcup_{p\geq 0} p_{(i,p)}\{0,1\}^\omega$
10: $i \leftarrow i+1$, $j \leftarrow 0$ `// begin to test if` $s \in \bigcup_{p\geq 0} p_{(i+1,p)}\{0,1\}^\omega$
11: **else** `// in this case,` $s \notin p_{(i,j)}\{0,1\}^\omega$
12: **return** 0 `// hence` $s \notin \bigcup_{p\leq j} p_{(i,p)}\{0,1\}^\omega$
13: $i \leftarrow i$, $j \leftarrow j+1$ `// begin to test if` $s \in \bigcup_{p\leq j+1} p_{(i,p)}\{0,1\}^\omega$
14: **end if**
15: **end loop**

the test of line 7 can always be performed in finite time, since each sequence $p_{b^{-1}(k)}$ is finite. Now, note that Algorithm 1 actually consists of a succession of recursive computational steps as well as extrarecursive calls to the ARNN $\mathcal{N}_{r(X)}$ provided by Lemma 2. Hence, Algorithm 1 can be performed by a composition of an infinite Turing machine [29] and an ARNN $\mathcal{N}_{r(X)}$. Yet since the behaviors of any Turing machines can indeed be simulated by some ARNN [6], it follows that the procedure described by Algorithm 1 can indeed by simulated by some deterministic ω-ARNN $\mathcal{N}_X$ which, when receiving input s bit by bit, outputs infinitely many 1's iff the procedure returns infinitely many 1's, or equivalently, iff $s \in X$. Yet according to the accepting conditions of $\mathcal{N}_X$, this is equivalent to saying that $s \in L(\mathcal{N}_X)$ iff $s \in X$. Therefore $L(\mathcal{N}_X) = X$. □

Now, propositions 1 and 2 permit to deduce the following characterization of the expressive power of deterministic ω-ARNNs:

Theorem 1. *Let $X \subseteq \{0,1\}^\omega$. Then X is recognizable by some deterministic ω-ARNN iff $X \in \mathbf{\Pi_2^0}$.*

Theorem 1 together with the results in [16] show that the ω-languages recognized by deterministic ω-ARNNs and deterministic Büchi **X**-automata all belong to the same Borel class $\mathbf{\Pi_2^0}$. In this sense, the expressive power of deterministic ω-ARNNs is topologically comparable to the expressive powers of deterministic Büchi **X**-automata. However, note that even if their expressive

power turn out to be similar, the class deterministic ω-ARNNs recognizes strictly more ω-languages than any other class of deterministic Büchi **X**-automata. Indeed, on the one side, any **X**-automaton is a finite object, so that it can be encoded by some integer. Consequently, any class of deterministic Büchi **X**-automata contains only countably many representatives, and might therefore recognize only countably many ω-languages. Yet on the other side, deterministic ω-ARNNs are able to recognize the whole class of $\mathbf{\Pi_2^0}$-sets, namely $2^{\aleph_0}$ ω-languages, which is uncountably many more than any other class of deterministic Büchi **X**-automata. In this precise sense, the expressive power of deterministic ω-ARNNs is topologically comparable to but also strictly richer than the expressive powers of any other class of deterministic Büchi **X**-automata. In particular, if $\mathcal{L}_{TM}$ and $\mathcal{L}_{NN}$ respectively denote the classes of ω-languages recognized by Büchi deterministic Turing machines and deterministic ω-ARNNs, the following result holds.

Theorem 2. $\mathcal{L}_{TM} \subsetneq \mathcal{L}_{NN} = \mathbf{\Pi_2^0}$.

5. The Expressive Power of Non-Deterministic ω-ARNN

Siegelmann and Sontag introduced the concept of a *non-deterministic* processor net as a modification of a deterministic one, obtained by incorporating a *guess* input channel in addition to the classical input channel [6]. The concept was introduced in the context of rational-weighted networks, but can naturally be extended to real-weighted networks (i.e. ARNNs). And in both rational- and real-weighted cases, it can be observed that such particular concept of non-determinism does actually not increase the computational power of the corresponding networks. Indeed, in the rational-weighted case, deterministic and non-deterministic neural networks were shown to be computationally equivalent to deterministic and non-deterministic Turing machines, respectively [6]. The classical equivalence between deterministic and non-deterministic Turing machines [30] then implies the existence of a similar equivalence between deterministic and non-deterministic networks, showing that non-determinism doesn't bring any extra computational power. In the real-weighted case, since deterministic neural networks already disclose unbounded power, it directly follows that their non-deterministic counterparts cannot be more powerful [8].

Here, we consider a natural extension of this concept of non-determinism to our particular case of analog networks over infinite input streams, and subsequently analyze the expressive power of such networks. More precisely, we provide a definition of a non-deterministic ω-ARNN and show that in our context, as opposed to the case of finite input streams, the translation from the determinism to non-determinism induces an extensive complexification of the expressive power of the corresponding networks from $\mathbf{\Pi_2^0}$ to $\mathbf{\Sigma_1^1}$-sets. This topological gap shows the significance of the concept of non-determinism for ω-ARNNs. It follows that non-deterministic ω-ARNNs turn out to be extensively more expressive than any other kind of non-deterministic **X**-automata equipped with Büchi or Muller accepting conditions – since all ω-languages recognized such

machines belong to the class Σ_1^1 (see [16, 17, 18]) and it holds that $\Sigma_1^1 \subsetneq \mathbf{\Sigma_1^1}$ [20]. In particular, non-deterministic ω-ARNNs happen to be strictly more expressive than non-deterministic Büchi or Muller classical automata, 1-counter automata, k-counter automata, Pushdown automata, and Turing machines.

Now, in order to sate the expected result, the following definition first need to be introduced. A *non-deterministic ω-ARNN* $\tilde{\mathcal{N}}$ consists of an ARNN provided with an input cell u_1, a second input cell u_2 playing the role of a *guess cell*, and an output cell y, all of them being forced to carry only binary signals. Hence, at each time step $t \geq 0$, the three cells u_1, u_2, and y admit activation values $u_1(t)$, $u_2(t)$, and $y(t)$ belonging to $\{0,1\}$. Then, assuming the initial activation vector of the network to be $x(\vec{0}) = \vec{0}$, any input stream

$$s = (u_1(t))_{t\in\mathbb{N}} = u_1(0)u_1(1)u_1(2)\cdots \in \{0,1\}^\omega$$

and guess stream

$$g = (u_2(t))_{t\in\mathbb{N}} = u_2(0)u_2(1)u_2(2)\cdots \in \{0,1\}^\omega$$

processed bit by bit induce via Equation (2) a corresponding infinite output stream

$$o_{s,g} = (y(t))_{t\in\mathbb{N}} = y(0)y(1)y(2)\cdots \in \{0,1\}^\omega$$

also processed bit by bit. Now, an input stream s will said to be *accepted* by $\tilde{\mathcal{N}}$ iff there exists a guess stream g such that the corresponding output stream $o_{s,g}$ contains infinitely many $1's$. This accepting condition corresponds to the natural Büchi-like translation in the present infinite input context of the accepting condition for non-deterministic processor nets on finite inputs stated by Siegelmann and Sontag [6]. Finally, as usual, the set of all infinite words accepted by $\tilde{\mathcal{N}}$ will be called the *neural ω-language* of $\tilde{\mathcal{N}}$, and will be denoted by $L(\tilde{\mathcal{N}})$.

According to the preceding definitions, any non-deterministic ω-ARNN $\tilde{\mathcal{N}}$ can naturally be identified with the function $f_{\tilde{\mathcal{N}}} : \{0,1\}^\omega \times \{0,1\}^\omega \longrightarrow \{0,1\}^\omega$ defined by $f_{\tilde{\mathcal{N}}}(s,g) = o_{s,g}$, where $o_{s,g}$ is the output generated by $\tilde{\mathcal{N}}$ when input s and guess g are received. By definition of the accepting condition of $\tilde{\mathcal{N}}$, the neural ω-language of $\tilde{\mathcal{N}}$ can thus be written as

$$L(\tilde{\mathcal{N}}) \quad = \quad \left\{ s : \exists g \ (s,g) \in f_{\tilde{\mathcal{N}}}^{-1}(F_\infty) \right\} = \pi_1 \left(f_{\tilde{\mathcal{N}}}^{-1}(F_\infty) \right).$$

Once again, the dynamics of non-deterministic ω-ARNNs impose strong restrictions on the functional behaviors of such networks.

Lemma 3. *Let $\tilde{\mathcal{N}}$ be some non-deterministic ω-ARNN. Then the corresponding function $f_{\tilde{\mathcal{N}}}$ is Lipschitz.*

Proof. First of all, the dynamics of $\tilde{\mathcal{N}}$ ensures that for any input s, any guess g, any corresponding output $o_{s,g}$, and any $k \geq 0$, the three bits $s(k)$, $g(k)$, and $o_s(k)$ are always generated simultaneously at time step k. Now, let (s_1, g_1), $(s_2, g_2) \in \{0,1\}^\omega \times \{0,1\}^\omega$, and let $o_{s_1,g_1} = f_{\tilde{\mathcal{N}}}(s_1, g_1)$ and $o_{s_2,g_2} = f_{\tilde{\mathcal{N}}}(s_2, g_2)$. If

$s_1 \neq s_2$ or $g_1 \neq g_2$, the metric d' over $\{0,1\}^\omega \times \{0,1\}^\omega$ ensures that $d'((s_1,g_1),(s_2,g_2)) = \frac{1}{2}\cdot d(s_1,s_2) + \frac{1}{2}\cdot d(g_1,g_2) = \frac{1}{2}\cdot 2^{-m} + \frac{1}{2}\cdot 2^{-n}$ for some $m,n \in \mathbb{N}\cup\{\infty\}$ such that either m or n or both are distinct from ∞. Now, suppose without loss of generality that $m \leq n$ and $m < \infty$. It follows that $s_1(0)\cdots s_1(m-1) = s_2(0)\cdots s_2(m-1)$, and $g_1(0)\cdots g_1(m-1) = g_2(0)\cdots g_2(m-1)$. Yet by the above argument, it follows that $o_{s_1,g_1}(0)\cdots o_{s_1,g_1}(m-1) = o_{s_2,g_2}(0)\cdots o_{s_2,g_2}(m-1)$, and thus $d(o_{s_1,g_1}, o_{s_2,g_2}) \leq 2^{-m}$. Therefore,

$$\begin{aligned} d\left(f_{\tilde{\mathcal{N}}}(s_1,g_1), f_{\tilde{\mathcal{N}}}(s_2,g_2)\right) &= d(o_{s_1,g_1}, o_{s_2,g_2}) \leq 2^{-m} \\ &\leq 2\cdot\left(\frac{1}{2}\cdot 2^{-m} + \frac{1}{2}\cdot 2^{-n}\right) = 2\cdot d'((s_1,g_1),(s_2,g_2)), \end{aligned}$$

showing that $f_{\tilde{\mathcal{N}}}$ is Lipschitz of modulus 2. If $s_1 = s_2$ and $g_1 = g_2$, then $f_{\tilde{\mathcal{N}}}(s_1,g_1) = f_{\tilde{\mathcal{N}}}(s_2,g_2)$ since $f_{\tilde{\mathcal{N}}}$ is a function. It follows that $d(f_{\tilde{\mathcal{N}}}(s_1,g_1), f_{\tilde{\mathcal{N}}}(s_2,g_2)) = 0 = d'((s_1,g_1),(s_2,g_2))$. Therefore, in all cases, $f_{\mathcal{N}}$ is Lipschitz of modulus 2. □

We now provide a complete characterization of the expressive power of the class of non-deterministic ω-ARNNs. More precisely, we prove that the class of ω-languages recognized by non-deterministic ω-ARNNs correspond precisely to the class of analytic subsets of $\{0,1\}^\omega$. First of all, we show that any neural ω-language recognized by some non-deterministic ω-ARNN is an analytic set.

Proposition 3. *Let $\tilde{\mathcal{N}}$ be a non-determinisitc ω-ARNN. Then $L(\tilde{\mathcal{N}}) \in \mathbf{\Sigma_1^1}$.*

Proof. As already mentioned, one has $F_\infty \in \mathbf{\Pi_2^0}$. Moreover, Lemma 3 shows that the function $f_{\tilde{\mathcal{N}}} : \{0,1\}\times\{0,1\} \longrightarrow \{0,1\}$ is Lipschitz, thus continuous. Hence, $f_{\tilde{\mathcal{N}}}^{-1}(F_\infty) \in \mathbf{\Pi_2^0}$. Therefore, $L(\tilde{\mathcal{N}}) = \pi_1(f_{\tilde{\mathcal{N}}}^{-1}(F_\infty)) \in \mathbf{\Sigma_1^1}$, see [19, 20, 21]. □

Conversely, we now prove that any analytic subset of $\{0,1\}^\omega$ can be recognized by some non-deterministic ω-ARNN. We proceed as in Section 4. For this purpose, let $X \subseteq \{0,1\}^\omega$ such that $X \in \mathbf{\Sigma_1^1}$. Then there exists a set $Y \subseteq \{0,1\}^\omega \times \{0,1\}^\omega$ such that $Y \in \mathbf{\Pi_2^0}$ and $X = \pi_1(Y)$. Yet according to the product topology on $\{0,1\}^\omega \times \{0,1\}^\omega$, the set Y can be written as $Y = \bigcap_{i\geq 0}\bigcup_{j\geq 0}(p_{(i,j)}\{0,1\}^\omega \times q_{(i,j)}\{0,1\}^\omega)$, where each $p_{(i,j)}, q_{(i,j)} \in \{0,1\}^+$. Consequently, the set Y, and hence also the set X, are thus completely determined by the countable sequence of pairs of finite prefixes $((p_{(i,j)}, q_{(i,j)}))_{i,j\geq 0}$. Hence, in order to encode the subset X into some real number, it suffices to encode the corresponding sequence of prefixes $((p_{(i,j)}, q_{(i,j)}))_{i,j\geq 0}$.

To begin with, each pair of finite prefixes $(p_{(i,j)}, q_{(i,j)}) \in \{0,1\}^+ \times \{0,1\}^+$ is first encoded by the finite sequence of natural numbers $\ulcorner (p_{(i,j)}, q_{(i,j)})\urcorner \in \{0,2,4,6\}^+$ defined by

$$\ulcorner (p_{(i,j)}, q_{(i,j)})\urcorner = 6\ulcorner p_{(i,j)}\urcorner 4 \ulcorner q_{(i,j)}\urcorner,$$

where $\ulcorner p_{(i,j)}\urcorner$ and $\ulcorner q_{(i,j)}\urcorner$ are the encodings of finite prefixes defined in Section 4. For instance, $\ulcorner (01, 110)\urcorner = 6024220$. Now, the sequence of pairs of

prefixes $((p_{(i,j)}, q_{(i,j)}))_{i,j\geq 0}$ can be encoded by the infinite sequence of integers $\ulcorner((p_{(i,j)}, q_{(i,j)}))_{i,j\geq 0}\urcorner \in \{0,2,4,6\}^\omega$ defined by the successive concatenation of all finite sequences $\ulcorner(p_{b^{-1}(k)}, q_{b^{-1}(k)})\urcorner$, for all $k \geq 0$, namely

$$\ulcorner((p_{(i,j)}, q_{(i,j)}))_{i,j\geq 0}\urcorner = \ulcorner(p_{b^{-1}(0)}, q_{b^{-1}(0)})\urcorner\ulcorner(p_{b^{-1}(1)}, q_{b^{-1}(1)})\urcorner\ulcorner(p_{b^{-1}(2)}, q_{b^{-1}(2)})\urcorner\cdots .$$

Using this first encoding, each finite block $\ulcorner(p_{b^{-1}(k)}, q_{b^{-1}(k)})\urcorner$ can now be unambiguously re-encoded by the rational number $r((p_{b^{-1}(k)}, q_{b^{-1}(k)})) \in [0,1]$ given by the interpretation of the sequence in base 7, namely

$$r((p_{b^{-1}(k)}, q_{b^{-1}(k)})) = \sum_{i=0}^{|\ulcorner(p_{b^{-1}(k)}, q_{b^{-1}(k)})\urcorner|-1} \frac{\ulcorner(p_{b^{-1}(k)}, q_{b^{-1}(k)})\urcorner(i)}{7^{i+1}}.$$

Finally, the set X itself can also be unambiguously re-encoded by the real number $r(X) \in [0,1]$ given by the interpretation of the infinite sequence $\ulcorner(p_{(i,j)}, q_{(i,j)}))_{i,j\geq 0}\urcorner$ in base 7, namely

$$r(X) = \sum_{i=0}^{\infty} \frac{\ulcorner((p_{(i,j)}, q_{(i,j)}))_{i,j\geq 0}\urcorner(i)}{7^{i+1}}.$$

Now, a generalization of Lemma 2 in this precise context can indeed be obtained.

Lemma 4. *Let $X \subseteq \{0,1\}^\omega$ be a $\mathbf{\Sigma_1^1}$-set with corresponding sequence of pair of prefixes $((p_{(i,j)}, q_{(i,j)}))_{i,j\geq 0}$. Then there exists an ARNN $\tilde{\mathcal{N}}_{r(X)}$ containing one input cell, one output cell, and a synaptic real weight equal to $r(X)$, and such that, starting from the zero initial state, and given the input signal $(1-2^{-k})0^\omega$ produces an output of the form $0^* r((p_{b^{-1}(k)}, q_{b^{-1}(k)}))0^\omega$.*

Proof. A straightforward generalization of the proof of Lemma 2. □

The next result now shows that any $\mathbf{\Sigma_1^1}$-set of $\{0,1\}^\omega$ can indeed be recognized by some non-deterministic ω-ARNN.

Proposition 4. *Let $X \subseteq \{0,1\}^\omega$ such that $X \in \mathbf{\Sigma_1^1}$. Then there exists a non-deterministic ω-ARNN $\tilde{\mathcal{N}}_X$ such that $L(\tilde{\mathcal{N}}_X) = X$.*

Proof. The set $X \in \mathbf{\Sigma_1^1}$ can be written as $X = \pi_1(Y)$, for some $Y \in \mathbf{\Pi_2^0}$, and the set Y can itself be written as $Y = \bigcap_{i\geq 0}\bigcup_{j\geq 0}(p_{(i,j)}\{0,1\}^\omega \times q_{(i,j)}\{0,1\}^\omega)$ for some $p_{(i,j)}, q_{(i,j)} \in \{0,1\}^+$. Hence, a given infinite input $s \in \{0,1\}^\omega$ belongs to X iff there exists a infinite guess $g \in \{0,1\}^\omega$ such that, for all index $i \geq 0$, there exists an index $j \geq 0$ satisfying $(s,g) \in p_{(i,j)}\{0,1\}^\omega \times q_{(i,j)}\{0,1\}^\omega$. Equivalently, $s \in X$ iff there exists $g \in \{0,1\}^\omega$ such that, for all $i \geq 0$, there exists $j \geq 0$ satisfying both $p_{(i,j)} \subsetneq s$ and $q_{(i,j)} \subsetneq g$. Thence, as for the deterministic case, the problem of determining whether some input-guess (s,g) provided bit by bit belongs to Y or not can be decided in infinite time by the procedure described by Algorithm 2, which, after ω computation steps, would have returned infinitely many 1's iff $(s,g) \in \bigcap_{i\geq 0}\bigcup_{j\geq 0}(p_{(i,j)}\{0,1\}^\omega \times q_{(i,j)}\{0,1\}^\omega) = Y$. Moreover, as

Algorithm 2

Input s and guess g are provided bit by bit at successive time steps
$i \leftarrow 0$, $j \leftarrow 0$
loop
  $k \leftarrow b(i,j)$
  Submit input $(1-2^{-k})$ to $\tilde{\mathcal{N}}_{r(X)}$ `// where` $\tilde{\mathcal{N}}_{r(X)}$ `is given by Lemma 4`
  Get output $r((p_{b^{-1}(k)}, q_{b^{-1}(k)}))$ from $\tilde{\mathcal{N}}_{r(X)}$
  $(p_{b^{-1}(k)}, q_{b^{-1}(k)}) \leftarrow decode(r((p_{b^{-1}(k)}, q_{b^{-1}(k)})))$
  if $p_{b^{-1}(k)} \subsetneq s$ and $q_{b^{-1}(k)} \subsetneq g$ **then**
    return 1
    $i \leftarrow i+1$, $j \leftarrow 0$
  else
    return 0
    $i \leftarrow i$, $j \leftarrow j+1$
  end if
end loop

for the deterministic case again, Algorithm 2 actually consists of a succession of recursive computational steps as well as extrarecursive calls to the ARNN $\tilde{\mathcal{N}}_{r(X)}$ provided by Lemma 4. Hence, the procedure can indeed by simulated by some ARNN $\tilde{\mathcal{N}}_X$ provided with two input cells as well as one output cell, and such that, when receiving the infinite input stream (s, g) bit by bit, the network $\tilde{\mathcal{N}}_X$ outputs infinitely many 1's iff the procedure returns infinitely many 1's, or equivalently, iff $(s, g) \in Y$. Hence, the function $f_{\tilde{\mathcal{N}}_X}$ naturally associated with $\tilde{\mathcal{N}}_X$ satisfies $f^{-1}_{\tilde{\mathcal{N}}_X}(F_\infty) = Y$. Finally, if we further consider that $\tilde{\mathcal{N}}_X$ is equipped with the accepting conditions of non-deterministic ω-ARNN, then the ω-language of $\tilde{\mathcal{N}}_X$ is precisely given by $L(\tilde{\mathcal{N}}_X) = \pi_1(f^{-1}_{\tilde{\mathcal{N}}_X}(F_\infty)) = \pi_1(Y) = X$. □

Now, propositions 1 and 2 induce the following characterization of the expressive power of non-deterministic ω-ARNNs:

Theorem 3. *Let $X \subseteq \{0,1\}^\omega$. Then X is recognizable by some non-deterministic ω-ARNN iff $X \in \mathbf{\Sigma^1_1}$.*

Consequently, Theorem 3 ensures that non-determnisitc ω-ARNNs turn out to be strictly more expressive than any other kind of **X**-automata equipped with a Büchi or Muller acceptance condition (since the ω-languages recognized by such machines strictly belong to the class Σ^1_1, and $\Sigma^1_1 \subsetneq \mathbf{\Sigma^1_1}$ [18]). In particular, we state the result for the case of non-determinisitc Turing machines, for they correspond to the most powerful abstract devices on infinite words.

Theorem 4. *Non-determnistic ω-ARNNs are strictly more expressive than non-deterministic Büchi or Muller Turing machines.*

Proof. By Theorem 3 and [18], the classes of ω-languages recognized by non-determninstic ω-ARNNs and non-deterministic Büchi or Muller Turing machines correspond respectively to the classes of $\mathbf{\Sigma_1^1}$-sets and Σ_1^1-sets. But it holds that $\mathbf{\Sigma_1^1} \supsetneq \Sigma_1^1$, which concludes the proof [20]. □

Finally, theorems 1 and 3 show a significant topological complexification between the expressive powers of deterministic and non-deterministic ω-ARNNs from $\mathbf{\Pi_2^0}$-sets to $\mathbf{\Sigma_1^1}$-sets. It is worth noting that a similar topological gap also holds for several Büch or Muller **X**-automata. For instance, the translation from determinism to non-determinism increases the expressive power of classical Büchi automata from $\mathbf{\Pi_2^0}$-sets to $\mathbf{\Delta_3^0}$-sets [12, 13]. Moreover, the expressive power of deterministic and non-deterministic Büchi 1-counter automata, k-counter automata, pushdown automata, and Turing machines turns out to be increased from $\mathbf{\Pi_2^0}$-sets to Σ_1^1-sets. The expressive power of all of their Muller counterparts also turns out to be increased from $\mathbf{\Delta_3^0}$-sets to Σ_1^1-sets [16, 17, 18]. However, such a complexification does not hold for all kind of usual abstract machines. For instance, Muller automata, Rabin automata, and Streett automata are shown to have a same expressive power of $\mathbf{\Delta_3^0}$ in their deterministic and non-determninistic versions [12, 13].

6. Extension to ω-ARNNs equipped with any kind of Borel Accepting Conditions

In the preceding sections, we have provided a complete characterization of the expressive power of ω-ARNNs equipped with some simple yet natural Büchi-like accepting condition. More precisely, the accepting condition was represented by the $\mathbf{\Pi_2^0}$ accepting set F_∞, and the neural ω-languages of any deterministic and non-deterministic ω-ARNNs $\mathcal{N}$ and $\tilde{\mathcal{N}}$ were respectively given by $L(\mathcal{N}) = f_{\mathcal{N}}^{-1}(F_\infty)$ and $L(\tilde{\mathcal{N}}) = \pi_1(f_{\tilde{\mathcal{N}}}^{-1}(F_\infty))$. Now, a natural extension of this work would be to investigate the expressive power of ω-ARNNs equipped with more topologically complex accepting conditions. In this context, we prove that a topological upper bound on the expressive power of ω-ARNNs equipped with any kind of Borel accepting condition can easily be obtained.

Towards this purpose, for any Borel set $F \subseteq \{0,1\}^\omega$, we say that $\mathcal{N}$ is a deterministic ω-ARNN with accepting condition F if the neural ω-language of $\mathcal{N}$ is given by $L(\mathcal{N}) = f_{\mathcal{N}}^{-1}(F)$. We say that $\tilde{\mathcal{N}}$ is a non-deterministic ω-ARNN with accepting condition F if the neural ω-language of $\tilde{\mathcal{N}}$ is given by $L(\tilde{\mathcal{N}}) = \pi_1(f_{\tilde{\mathcal{N}}}^{-1}(F))$.

Now, we first prove that the expressive power of deterministic ω-ARNN is potentially increased by the consideration of more topologically complex Borel accepting conditions.

Proposition 5. *Let $\mathcal{N}$ be some deterministic ω-ARNN with accepting condition $F \in \mathbf{\Sigma_\alpha^0}$, for some $0 < \alpha < \omega_1$. Then $L(\mathcal{N}) \in \mathbf{\Sigma_\alpha^0}$. The same result applies for $F \in \mathbf{\Pi_\alpha^0}$, and $F \in \mathbf{\Delta_\alpha^0}$.*

Proof. One has $L(\mathcal{N}) = f_{\mathcal{N}}^{-1}(F)$, and Lemma 1 shows that $f_{\mathcal{N}}$ is Lipschitz, thus continuous. Yet since $F \in \mathbf{\Sigma_\alpha^0}$, it follows that $L(\mathcal{N}) = f_{\mathcal{N}}^{-1}(F) \in \mathbf{\Sigma_\alpha^0}$, see [19, 20, 21]. □

However, in the non-deterministic case, the consideration of more topologically complex Borel accepting conditions would absolutely not increase the expressive power of the corresponding non-determistic ω-ARNNs above the analytic level.

Proposition 6. *Let $\tilde{\mathcal{N}}$ be some non-deterministic ω-ARNN with accepting condition $F \in \mathbf{\Sigma_\alpha^0} \cup \mathbf{\Pi_\alpha^0}$, for some $0 < \alpha < \omega_1$. Then $L(\tilde{\mathcal{N}}) \in \mathbf{\Sigma_1^1}$.*

Proof. One has $L(\tilde{\mathcal{N}}) = \pi_1(f_{\tilde{\mathcal{N}}}^{-1}(F))$, and Lemma 3 shows that $f_{\tilde{\mathcal{N}}}$ is Lipschitz, thus continuous. Yet since $F \in \mathbf{\Sigma_\alpha^0} \cup \mathbf{\Pi_\alpha^0}$, it follows that $f_{\tilde{\mathcal{N}}}^{-1}(F) \in \mathbf{\Sigma_\alpha^0} \cup \mathbf{\Pi_\alpha^0}$. Hence, $L(\tilde{\mathcal{N}}) = \pi_1(f_{\tilde{\mathcal{N}}}^{-1}(F))$ consists of a projection of a Borel set of the space $\{0,1\}^\omega \times \{0,1\}^\omega$, and therefore $L(\tilde{\mathcal{N}}) \in \mathbf{\Sigma_1^1}$, see [19, 20, 21]. □

Proposition 5 shows that the topological complexity of the expressive power of deterministic ω-ARNNs is bounded by the topological complexity of their accepting conditions. Proposition 6 shows that, for any Borel accepting condition, the expressive power of the corresponding class of non-deterministic ω-ARNNs stays confined into the analytic level. Therefore, increasing the topological complexity of the accepting condition potentially reduces the topological gap between the corresponding classes of deterministic and non-deterministic ω-netowrks. Moreover, note that in both deterministic and non-deterministic cases, the question of whether a given accepting condition $F \in \mathbf{\Sigma_\alpha^0}$ (resp. $F \in \mathbf{\Pi_\alpha^0}$) would suffice to exhaust the whole classes of $\mathbf{\Sigma_\alpha^0}$ (resp. $\mathbf{\Pi_\alpha^0}$) and $\mathbf{\Sigma_1^1}$-sets (as for the condition F_∞) cannot be solved by simply considering the Borel rank of the condition.

Finally, as already mentioned, Siegelmann and Sontag proved that the class of ARNNs over finite words admits unbounded computational power, in the sense of being capable of recognizing all possible languages of $\{0,1\}^+$ [8]. In our case, propositions 5 and 6 directly imply that this result does actually not extend in the present infinite word context, since the topological complexity of any class of deterministic or non-deterministic ω-ARNNs over some given Borel accepting condition is always bounded.

7. Conclusion

We introduced a concept of deterministic and non-deterministic analog recurrent neural networks on infinite words, and proved that the ω-languages recognized by such networks exhaust precisely to the whole classes of $\mathbf{\Pi_2^0}$-sets and $\mathbf{\Sigma_1^1}$-sets. Consequently, the expressive power of deterministic ω-ARNNs turn out to be topologically comparable to the expressive power of deterministic Büchi abstract machines, whereas the expressive power of non-deterministic ω-ARNNs turn out to significantly more important than the expressive power of

any non-deterministic Büchi or Muller abstract machines. Yet it is worth noting once again that, in the deterministic case, even if their underlying ω-languages are bounded by the same Borel rank, the class of ω-ARNNs still recognizes uncountably many more ω-languages than any other class of deterministic Büchi abstract machines. Furthermore, we proved that a Borel upper bound on the expressive powers of deterministic and non-deterministic ω-ARNNs equipped with any kind of Borel accepting conditions could also be obtained. Consequently, as frequently observed for several Büchi or Muller abstract machines, we noticed the existence of a topological gap between the expressive powers of the deterministic and non-deterministic versions of our computational model.

These results significantly differ from those of occurring in classical finite word context. Indeed, in the case of finite words, deterministic and non-deterministic analog networks were shown to admit the same computational power, independently of the nature of their synaptic weights (may they be rational or reals) [8, 6].

Furthermore, apart from the Borel hierarchy, the Wadge hierarchy[1] also provides a relevant tool for the study of the topological complexity of ω-languages [31]. Indeed, the Wadge hierarchy provides an extensive refinement of the Borel hierarchy, and hence permits a closer analysis of the topological complexity of classes of ω-languages. In this context, Finkel surprisingly proved that the Wadge hierarchy – and hence also the Borel hierarchy – of ω-languages accepted by non-deterministic real time 1-counter Büchi automata turns out to be the same as the Wadge hierarchy of ω-languages accepted by non-deterministic Büchi or Muller Turing machines (i.e. the class Σ^1_1) [17]. Consequently, the Wadge hierarchies – hence also the Borel hierarchies – of ω-languages accepted by all kind $\mathbf{X}$-automata whose expressive powers are situated between real time 1-counter Büchi automata and Muller Turing machines are the same.

In our case, theorems 1 and 3 provide a direct characterization of the Borel and Wadge hierarchies of ω-languages recognized by deterministic and non-deterministic ω-ARNNs, namely:

Theorem 5.
- *The Borel and Wadge hierarchies of ω-languages recognized by deterministic ω-ARNNs correspond respectively to the Borel and Wadge hierarchies of the class of all $\mathbf{\Pi^0_2}$-sets.*
- *The Borel and Wadge hierarchies of ω-languages recognized by non-deterministic ω-ARNNs correspond respectively to the Borel and Wadge hierarchies of the class of all $\mathbf{\Sigma^1_1}$-sets.*

The preceding theorem together with the results presented by Finkel [17] permit to compare the Borel and Wadge hierarchies of ω-ARNNs and $\mathbf{X}$-automata. On the one hand, the Borel hierarchies of deterministic ω-ARNNs and Büchi $\mathbf{X}$-automata coincide, and the Borel hierarchy of non-deterministic ω-ARNNs

[1] The Wadge hierarchy of a given class of ω-languages corresponds to the collection of all ω-languages of this class ordered by the Wadge reduction $\leq_W$. The Wadge reduction $\leq_W$ is defined by $X \leq_W Y$ iff there exists f continuous such that $X = f^{-1}(Y)$.

strictly contains the Borel hierarchy of any kind of non-deterministic **X**-automata. On the other hand, the Wadge hierarchies of both deterministic and non-deterministic ω-ARNNs strictly contain the Wadge hierarchies of any kind of deterministic and non-deterministic **X**-automata. Hence, in the deterministic context, the Wadge complexity reveals a refined distinction between the expressive powers of ω-ARNNs and Büchi **X**-automata that cannot be captured from the point of view of the Borel complexity. In the non-deterministic context, both Borel and Wadge complexities show that ω-ARNNs are strictly more expressive that any **X**-automata. Therefore, the refined Wadge analysis actually shows that the ω-ARNNs are strictly more expressive than any other kind of **X**-automaton, both in their deterministic and non-deterministic versions. This difference of expressivity is however clearly more significant in the non-deterministic than in the deterministic context.

Besides, this work can be extended in several directions. For instance, we think that the study of analog neural ω-networks equipped with more complex or more biologically oriented accepting condition would be of specific interest. Moreover, Balcázar et al. described a hierarchical classification of analog networks according to the Kolmogorov complexity of their real-weights [32]. This classification can directly be translated in the present infinite word context, for it is completely independent of the accepting condition of the networks. Hence, a natural question would be to investigate the possible links between the Kolmogorov and the topological complexity of analog ω-networks.

Moreover, a natural extension of this work would be to pursue the study the computational power of analog recurrent neural networks in the context of interactive computation [33]. Indeed, van Leeuwen and Wiedermann argued that classical computation "no longer fully corresponds to the current notion of computing in modern systems", and proposed an interactive infinite computational framework that turn out to be relevant for the modeling of the behavior of bio-inspired complex information processing systems [34, 35].

Finally, the comparison between diverse bio-inspired and artificial models of computation intends to capture the fundamental distinctions and similarities that exist between the processes of transfer of information in biological systems on the one side and in artificial devices on the other. We believe that this theoretical comparative approach to neural computability might bring further insight into the key issue of information processing in the brain, and might contribute ultimately to provide a better understanding of the intrinsic nature of biological intelligence. The present paper hopes to make a step forward in this direction.

References

[1] J. v. Neumann, The computer and the brain, Yale University Press, New Haven, CT, USA, 1958.

[2] W. S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysic 5 (1943) 115–133.

[3] S. C. Kleene, Representation of events in nerve nets and finite automata, in: Automata Studies, Vol. 34 of Annals of Mathematics Studies, Princeton University Press, Princeton, N. J., 1956, pp. 3–42.

[4] M. L. Minsky, Computation: finite and infinite machines, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.

[5] H. T. Siegelmann, E. D. Sontag, Turing computability with neural nets, Applied Mathematics Letters 4 (6) (1991) 77–80.

[6] H. T. Siegelmann, E. D. Sontag, On the computational power of neural nets, J. Comput. Syst. Sci. 50 (1) (1995) 132–150.

[7] J. Kilian, H. T. Siegelmann, The dynamic universality of sigmoidal neural networks, Inf. Comput. 128 (1) (1996) 48–56.

[8] H. T. Siegelmann, E. D. Sontag, Analog computation via neural networks, Theor. Comput. Sci. 131 (2) (1994) 331–360.

[9] H. T. Siegelmann, Computation beyond the Turing limit, Science 268 (5210) (1995) 545–548.

[10] H. T. Siegelmann, Neural and super-Turing computing, Minds Mach. 13 (1) (2003) 103–114.

[11] A. M. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proc. London Math. Soc. 2 (42) (1936) 230–265.

[12] D. Perrin, J.-E. Pin, Infinite Words, Vol. 141 of Pure and Applied Mathematics, Elsevier, 2004, iSBN 0-12-532111-2.

[13] W. Thomas, Automata on infinite objects, in: Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B), Elsevier and MIT Press, 1990, pp. 133–192.

[14] W. Thomas, Automata, logics, and infinite games: a guide to current research, Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[15] J. Cabessa, A. E. P. Villa, A hierarchical classification of first-order recurrent neural networks, in: A. H. Dediu, H. Fernau, C. Martín-Vide (Eds.), LATA, Vol. 6031 of Lecture Notes in Computer Science, Springer, 2010, pp. 142–153.

[16] J. Engelfriet, H. J. Hoogeboom, X-automata on ω-words, Theor. Comput. Sci. 110 (1) (1993) 1–51.

[17] O. Finkel, Borel ranks and wadge degrees of context free ω-languages, Mathematical. Structures in Comp. Sci. 16 (5) (2006) 813–840.

[18] 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.

[19] T. Jech, Set theory. The third millennium edition, revised and expanded., Springer Monographs in Mathematics. Berlin: Springer. xiii, 769 p., 2003.

[20] A. S. Kechris, Classical descriptive set theory, Vol. 156 of Graduate Texts in Mathematics, Springer-Verlag, New York, 1995.

[21] S. M. Srivastava, A Course on Borel Sets (Graduate Texts in Mathematics), Springer, 1998.

[22] V. Del Prete, L. Martignon, A. E. P. Villa, Detection of syntonies between multiple spike trains using a coarse-grain binarization of spike count distributions, Network: Comput. Neural Syst. 15 (2004) 13–28.

[23] M. Abeles, Local Cortical Circuits, Studies of brain function ; vol. 6 ; 102 p., Springer Verlag, Berlin ; New York, 1982.

[24] M. Abeles, Corticonics, Cambridge University Press, 1991.

[25] A. E. P. Villa, I. V. Tetko, B. Hyland, A. Najem, Spatiotemporal activity patterns of rat cortical neurons predict responses in a conditioned task, Proceedings of the National Academy of Sciences of the USA 96 (1999) 1006–1011.

[26] I. V. Tetko, A. E. P. Villa, A pattern grouping algorithm for analysis of spatiotemporal patterns in neuronal spike trains. 2. application to simultaneous single unit recordings, J. Neurosci. Meth. 105 (2001) 15–24.

[27] M. Sipser, Borel sets and circuit complexity, in: STOC '83: Proceedings of the fifteenth annual ACM symposium on Theory of computing, ACM, New York, NY, USA, 1983, pp. 61–69.

[28] M. Sipser, A topological view of some problems in complexity theory, in: M. Chytil, V. Koubek (Eds.), MFCS, Vol. 176 of Lecture Notes in Computer Science, Springer, 1984, pp. 567–572.

[29] J. D. Hamkins, Infinite time turing machines, Minds Mach. 12 (4) (2002) 521–539.

[30] J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 1979.

[31] W. W. Wadge, Reducibility and determinateness on the Baire space, Ph.D. thesis, University of California, Berkeley (1983).

[32] J. L. Balcázar, R. Gavaldà, H. T. Siegelmann, Computational power of neural networks: a characterization in terms of kolmogorov complexity, IEEE Transactions on Information Theory 43 (4) (1997) 1175–1183.

[33] D. Goldin, S. A. Smolka, P. Wegner, Interactive Computation: The New Paradigm, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[34] J. van Leeuwen, J. Wiedermann, Beyond the turing limit: Evolving interactive systems, in: L. Pacholski, P. Ružicka (Eds.), SOFSEM 2001: Theory and Practice of Informatics, Vol. 2234 of LNCS, Springer Berlin / Heidelberg, 2001, pp. 90–109.

[35] J. van Leeuwen, J. Wiedermann, How we think of computing today, in: Logic and Theory of Algorithms, Vol. 5028 of LNCS, Springer Berlin / Heidelberg, 2008, pp. 579–593.