HAL
open science

Expressive Power of Evolving Neural Networks Working on Infinite Input Streams

Jérémie Cabessa, Olivier Finkel

To cite this version:

Jérémie Cabessa, Olivier Finkel. Expressive Power of Evolving Neural Networks Working on Infinite Input Streams. 21st International Symposium on Fundamentals of Computation Theory (FCT 2017), Sep 2017, Bordeaux, France. pp.150-163, 10.1007/978-3-662-55751-8_13 . hal-01590650

HAL Id: hal-01590650

https://hal.science/hal-01590650v1

Submitted on 19 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Expressive Power of Evolving Neural Networks Working on Infinite Input Streams

Jérémie Cabessa[1] and Olivier Finkel[2]

[1] Laboratoire d'économie mathématique – LEMMA
Université Paris 2, 4 Rue Blaise Desgoffe, 75006 Paris, France
jeremie.cabessa@u-paris2.fr
[2] Institut de Mathématiques de Jussieu - Paris Rive Gauche
CNRS et Université Paris Diderot, UFR de mathmatiques case 7012
75205 Paris Cedex 13, France
finkel@math.univ-paris-diderot.fr

Abstract. Evolving recurrent neural networks represent a natural model of computation beyond the Turing limits. Here, we consider evolving recurrent neural networks working on infinite input streams. The expressive power of these networks is related to their attractor dynamics and is measured by the topological complexity of their underlying neural ω-languages. In this context, the deterministic and non-deterministic evolving neural networks recognize the (boldface) topological classes of $BC(\mathbf{\Pi}^0_2)$ and $\mathbf{\Sigma}^1_1$ ω-languages, respectively. These results can actually be significantly refined: the deterministic and nondeterministic evolving networks which employ $\alpha \in 2^\omega$ as sole binary evolving weight recognize the (lightface) relativized topological classes of $BC(\Pi^0_2)(\alpha)$ and $\Sigma^1_1(\alpha)$ ω-languages, respectively. As a consequence, a proper hierarchy of classes of evolving neural nets, based on the complexity of their underlying evolving weights, can be obtained. The hierarchy contains chains of length ω_1 as well as uncountable antichains.

Keywords: neural networks; attractors; formal languages; ω-languages; Borel sets; analytic sets; effective Borel and analytic sets

1 Introduction

The theoretical approach to neural computation has mainly been focused on comparing the computational capabilities of diverse neural models with those of abstract computing machines. Nowadays, the computational capabilities of various models of neural networks have been shown to range from the finite automaton degree [14–16, 18], up to the Turing [20, 23] or even to the super-Turing level [4, 6, 19, 21].

In particular, the real-weighted (or analog) neural networks are strictly more powerful than the Turing machines. They decide the complexity class $P/poly$ in polynomial time of computation [19]. The precise computational capabilities of these networks can actually be characterized in terms of the Kolmogorov

complexity of their underlying synaptic real weights. As a consequence, a proper hierarchy of classes of analog neural nets employing real weights of increasing Kolmogorov complexity has been obtained [2]. On the other hand, the evolving neural networks (i.e., those employing time-dependent synaptic weights) turn out to be computationally equivalent to the analog ones, irrespectively of the nature (rational or real) of their underlying synaptic weights [4, 6].

More recently, based on biological as well as theoretical considerations, these studies have been extended to the paradigm of infinite computation [3, 5–10]. In this context, the expressive power of the networks is intrinsically related to their attractor dynamics, and is measured by the topological complexity of their underlying neural ω-languages. In this case, the Boolean recurrent neural networks provided with certain type specification of their attractors are computationally equivalent to Büchi or Muller automata [8]. The rational-weighted neural nets are equivalent to Muller Turing machines. The deterministic and nondeterministic analog and evolving neural networks recognize the (boldface) topological classes of $BC(\mathbf{\Pi}^0_2)$ and $\mathbf{\Sigma}^1_1$ ω-languages, respectively, and in this respect, are super-Turing [3, 9].

Here, we refine the above mentioned results for the case of evolving neural networks. More precisely, we focus without loss of generality on evolving neural nets employing only one time-dependent binary weight throughout their computational process. We show that the deterministic and nondeterministic evolving networks using the sole changing weight $\alpha \in 2^\omega$ recognize the (lightface) relativized topological classes of $BC(\Pi^0_2)(\alpha)$ and $\Sigma^1_1(\alpha)$ ω-languages, respectively. As a consequence, a proper hierarchy of classes of evolving neural nets, based on the complexity of their underlying evolving weights, can be obtained. The hierarchy contains chains of length ω_1 as well as uncountable antichains. These achievements generalize the proper hierarchy of classes of analog networks obtained in the context of classical computation [2].

2 Preliminaries

Given a finite set X, usually referred to as an *alphabet*, we let X^* and X^ω denote the sets of finite sequences (or *finite words*) and infinite sequences (or *infinite words*) of elements of X. A set $L \subseteq X^*$ or $L \subseteq X^\omega$ is called a *language* or an *ω-language*, respectively. In the sequel, any space of the form X^ω will be assumed to be equipped with the product topology of the discrete topology on X. Accordingly, the basic open sets of X^ω are of the form $p \cdot X^\omega$, for some $p \in X^*$. The general open sets are countable unions of basic open sets. In particular, the space of infinite words of bits (Cantor space) and that of infinite words of N-dimensional Boolean vectors will be denoted by $2^\omega = \{0,1\}^\omega$ and $(\mathbb{B}^N)^\omega$, respectively. They are assumed to be equipped with the above mentioned topology.

Let $(\mathcal{X}, \mathcal{T})$ be one of the above topological spaces, or a product of such spaces. The class of *Borel subsets* of $\mathcal{X}$, denoted by $\mathbf{\Delta}^1_1$ (boldface), is the σ-algebra generated by $\mathcal{T}$, i.e., the smallest collection of subsets of $\mathcal{X}$ containing

all open sets and closed under countable union and complementation. For every non-null countable ordinal $\alpha < \omega_1$, where ω_1 is the first uncountable ordinal, the Borel classes $\mathbf{\Sigma}^0_\alpha$, $\mathbf{\Pi}^0_\alpha$ and $\mathbf{\Delta}^0_\alpha$ of $\mathcal{X}$ are defined as follows:

- $\mathbf{\Sigma}^0_1$ is the class of open subsets of $\mathcal{X}$ (namely $\mathcal{T}$)
- $\mathbf{\Pi}^0_1$ is the class of closed subsets of $\mathcal{X}$, i.e., that of complements of open sets
- $\mathbf{\Sigma}^0_\alpha$ is the class of countable unions of subsets of $\mathcal{X}$ in $\bigcup_{\gamma<\alpha} \mathbf{\Pi}^0_\gamma$
- $\mathbf{\Pi}^0_\alpha$ is the class of countable intersections of subsets of $\mathcal{X}$ in $\bigcup_{\gamma<\alpha} \mathbf{\Sigma}^0_\gamma$.
- $\mathbf{\Delta}^0_\alpha = \mathbf{\Sigma}^0_\alpha \cap \mathbf{\Pi}^0_\alpha$

The Borel classes $\mathbf{\Sigma^0_\alpha}$, $\mathbf{\Pi^0_\alpha}$ and $\mathbf{\Delta^0_\alpha}$ provide a stratification of the class of Borel sets known as the *Borel hierarchy*. One has $\mathbf{\Delta}^1_1 = \bigcup_{\alpha<\omega_1} \mathbf{\Sigma^0_\alpha} = \bigcup_{\alpha<\omega_1} \mathbf{\Pi^0_\alpha}$ [12]. The *rank* of a Borel set $A \subseteq \mathcal{X}$ is the smallest ordinal α such that $A \in \mathbf{\Sigma^0_\alpha} \cup \mathbf{\Pi^0_\alpha}$. It is commonly considered as a relevant measure of the topological complexity of Borel sets. The class of sets obtained as finite Boolean combinations (unions, intersections and complementations) of $\mathbf{\Pi}^0_2$-sets is denoted by $BC(\mathbf{\Pi}^0_2)$.

Analytic sets are more complicated than Borel sets. They are obtained as projections of either $\mathbf{\Pi}^0_2$-sets or general Borel sets [12]. More precisely, a set $A \subseteq \mathcal{X}$ is *analytic* if there exists some $\mathbf{\Pi}^0_2$-set $B \subseteq \mathcal{X} \times 2^\omega$ such that $A = \{x \in \mathcal{X} : (x, \beta) \in B, \text{for some } \beta \in 2^\omega\} = \pi_1(B)$ [12]. The class of analytic sets is denoted by $\mathbf{\Sigma}^1_1$. It strictly contains that of Borel sets, i.e., $\mathbf{\Delta}^1_1 \subsetneq \mathbf{\Sigma}^1_1$ [12].

The *effective* (lightface) counterpart of the Borel and analytic classes, denoted by $\Sigma^0_{\mathrm{n}}, \Pi^0_{\mathrm{n}}, \Delta^0_{\mathrm{n}}$ as well as Δ^1_1 and Σ^1_1, are obtained by a similar effective construction, yet starting from the class Σ^0_1 of effective open sets [17]. The class of finite Boolean combinations of Π^0_2-sets, denoted by $BC(\Pi^0_2)$ (lightface), and that of effective analytic sets, denoted by Σ^1_1 (lightface), correspond to the collections of ω-languages recognizable by deterministic and nondeterministic Muller Turing machines, respectively [22]. One has $BC(\Pi^0_2) \subsetneq BC(\mathbf{\Pi}^0_2)$ and $\Sigma^1_1 \subsetneq \mathbf{\Sigma}^1_1$.

Any topological class Γ of the underlying topological space $\mathcal{X}$ will also be written as $\Gamma \restriction \mathcal{X}$, whenever we want $\mathcal{X}$ to be specified. In addition, for any point $x \in \mathcal{X}$, we will use the notation $x \in \mathit{\Gamma}$ to mean that $\{x\} \in \mathit{\Gamma}$. Besides, any product space $\mathcal{X} \times \mathcal{Y}$ is assumed to be equipped with the product topology. If $A \subseteq \mathcal{X} \times \mathcal{Y}$ and $y \in \mathcal{Y}$, the *y-section* of A is defined by $A_y = \{x \in \mathcal{X} : (x, y) \in A\}$. We shall consider in the sequel, for any class Γ being equal to Σ^0_1, $BC(\Pi^0_2)$, Σ^1_1, or Π^1_1, with underlying product space $\mathcal{X} \times \mathcal{Y}$ and for any $y \in \mathcal{Y}$, the *relativization of* Γ *to* y, denoted by $\Gamma(y)$, which is the class of all y-sections of sets in Γ. In other words: $A \in \Gamma(y) \restriction \mathcal{X}$ if and only if there exists $B \in \Gamma \restriction \mathcal{X} \times \mathcal{Y}$ such that $A = B_y$. Moreover, we denote as usual $\Delta^1_1(y) = \Sigma^1_1(y) \cap \Pi^1_1(y)$ [17, p. 118].

For any $\alpha \in 2^\omega$, one can show that the relativized classes $BC(\Pi^0_2)(\alpha)$ and $\Sigma^1_1(\alpha)$ correspond to the collections of ω-languages recognizable by deterministic and nondeterministic Muller Turing machine with oracle α, respectively. In addition, it can be shown that $x \in \Sigma^0_1(\alpha)$ if and only if the successive letters of x can be produced step by step by some TM with oracle α. Besides, one has $x \in \Sigma^1_1(\alpha)$ iff $x \in \Delta^1_1(\alpha)$, for any $\alpha \in 2^\omega$ [17].

Finally, the spaces $(\mathbb{B}^M)^\omega \times 2^\omega$ and $(\mathbb{B}^{M+1})^\omega$ are isomorphic via the natural identification. Accordingly, subsets of these spaces will be identified without it being explicitly mentioned.

3 Recurrent Neural Networks on Infinite Input Streams

We consider first-order recurrent neural networks composed of Boolean input cells, Boolean output cells and sigmoidal internal cells. The sigmoidal internal neurons introduce the biological source of nonlinearity which is crucial to neural computation. They provide the possibility to surpass the capabilities of finite state automata, or even of Turing machines. The Boolean input and output cells carry out the exchange of discrete information between the network and the environment. When some infinite input stream is supplied, the output cells eventually enter into some attractor dynamics. The expressive power of the networks is related to the attractor dynamics of their Boolean output cells.

3.1 Deterministic case

A *deterministic (first-order) recurrent neural network*, denoted by D-RNN, consists of a synchronous network of neurons related together in a general architecture. It is composed of M Boolean input cells $(u_i)_{i=1}^M$, N sigmoidal internal 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 internal neurons $(u_j)_{j=1}^M$ and $(x_j)_{j=1}^N$ at time t, the activation values of each internal and output neuron x_i and y_i at time $t+1$ are updated by the following equations, respectively:

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

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

where $a_{ij}(t)$, $b_{ij}(t)$, and $c_i(t)$ are the time dependent *synaptic weights* and *bias* of the network at time t, and σ and θ are the linear-sigmoid[3] and Heaviside step activation functions defined by

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

A synaptic weight or a bias w will be called *static* if it remains constant over time, i.e., if $w(t) = c$ for all $t \ge 0$. It will be called *bi-valued evolving* if it varies among two possible values over time, i.e., if $w(t) \in \{0,1\}$ for all $t \ge 0$. A D-RNN is illustrated in Figure 1.

[3] The results of the paper remain valid for any other kind of sigmoidal activation function satisfying the properties mentioned in [13, Section 4].

The dynamics of a D-RNN $\mathcal{N}$ 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

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

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

Consider some D-RNN $\mathcal{N}$ provided with M Boolean input cells, N sigmoidal internal cells, and P Boolean output cells. For each time step $t \geq 0$, the *state* of $\mathcal{N}$ at time t consists of a pair of the form

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

The second element of this pair, namely $\boldsymbol{y}(t)$, is the *output state* of $\mathcal{N}$ at time t.

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

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

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

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

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

$$bc_s = (\boldsymbol{y}(t))_{t \in \mathbb{N}} = \boldsymbol{y}(0)\boldsymbol{y}(1)\boldsymbol{y}(2) \cdots \in (\mathbb{B}^P)^\omega$$

is the *Boolean computation* of $\mathcal{N}$ induced by s. The computation of such a D-RNN is illustrated in Figure 1.

Note that any D-RNN $\mathcal{N}$ with P Boolean output cells can only have 2^P – i.e., finitely many – possible distinct output states. Consequently, any Boolean computation bc_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 – denoted by $\inf(bc_s)$. A set of states of the form $\inf(bc_s) \subseteq \mathbb{B}^P$ will be called an *attractor*[4] of $\mathcal{N}$ [8], as illustrated in Figure 1. A precise definition can be given as follows:

Definition 1. *Let $\mathcal{N}$ be some D-RNN. A set $A = \{\boldsymbol{y_0}, \ldots, \boldsymbol{y_k}\} \subseteq \mathbb{B}^P$ is an attractor for $\mathcal{N}$ if there exists some infinite input stream s such that the corresponding Boolean computation bc_s satisfies* $\inf(bc_s) = A$.

We suppose that the attractors are of two distinct types, either *accepting* or *rejecting*. The type specification of these attractors is not the subject of this work (cf. [8]), and from this point onwards, we assume that any D-RNN is equipped

[4] In words, an attractor of $\mathcal{N}$ is a set of output states into which the Boolean computation of the network could become forever trapped – yet not necessarily in a periodic manner.

with a corresponding classification of all of its attractors into accepting and rejecting types.

This classification of attractors yields the following Muller acceptance condition: given some D-RNN $\mathcal{N}$, an infinite input stream $s \in (\mathbb{B}^M)^\omega$ is *accepted* $\mathcal{N}$ if $\inf(bc_s)$ is an accepting attractor; it is *rejected* by $\mathcal{N}$ if $\inf(bc_s)$ is a rejecting attractor. The set of all accepted input streams of $\mathcal{N}$ is called the *neural ω-language recognized by* $\mathcal{N}$, denoted by $L(\mathcal{N})$. A set $L \subseteq (\mathbb{B}^M)^\omega$ is said to be *recognizable* by some D-RNN if there exists a network $\mathcal{N}$ such that $L(\mathcal{N}) = L$.

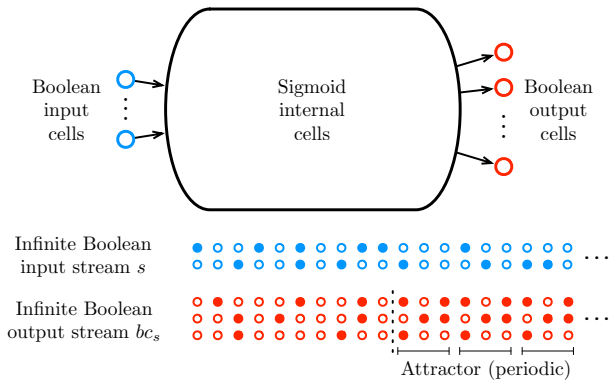

Fig. 1. Illustration of the computational process performed by some D-RNN. The infinite Boolean input stream $s = \boldsymbol{u}(0)\boldsymbol{u}(1)\boldsymbol{u}(2)\cdots \in (\mathbb{B}^M)^\omega$, represented by the blue pattern, induces a corresponding Boolean output stream – or Boolean computation – $bc_s = \boldsymbol{y}(0)\boldsymbol{y}(1)\boldsymbol{y}(2)\cdots \in (\mathbb{B}^P)^\omega$, represented by the red pattern. The filled and empty circles represent active and quiet Boolean cells, respectively. From some time step onwards, a certain set of output states begins to repeat infinitely often, which corresponds to the attractor dynamics associated with input stream s.

Two different models of D-RNNs can be considered according to the nature of their synaptic weights:

1. The class of *deterministic static rational neural nets*, denoted by D-St-RNN[$\mathbb{Q}$], which refers to the D-RNNs whose every weights are static and modelled by rational values.
2. The class of *deterministic bi-valued evolving rational neural nets*, denoted by D-Ev$_2$-RNN[$\mathbb{Q}$], which refers to the D-RNNs whose every evolving weights are bi-valued and every static weights are rational. In this case, the subclass of networks containing $\alpha_1, \ldots, \alpha_k \in 2^\omega$ as sole bi-valued evolving weights, all other ones being static rational, is denoted by D-Ev$_2$-RNN[$\mathbb{Q}, \alpha_1, \ldots, \alpha_k$].

3.2 Nondeterministic case

We also consider nondeterministic recurrent neural networks, as introduced in [19, 20]. The nondeterminism is expressed by means of an external binary guess stream processed via some additional Boolean guess cell.

Formally, a *nondeterministic (first-order) recurrent neural network*, denoted by N-RNN, consists of a recurrent neural network $\mathcal{N}$ as described in previous Section 3.1, except that it contains $M+1$ Boolean input cells $(u_i)_{i=1}^{M+1}$, rather than M. The cell u_{M+1}, called the *guess cell*, carries the Boolean source of nondeterminism to be considered [3, 7, 9, 19, 20].

Given some N-RNN $\mathcal{N}$, any sequence $g = g(0)g(1)g(2)\cdots \in 2^\omega$ submitted to guess cell u_{M+1} is a *guess stream* for $\mathcal{N}$. Assuming the initial state of the network to be $\langle \boldsymbol{x}(0), \boldsymbol{y}(0)\rangle = \langle \boldsymbol{0}, \boldsymbol{0}\rangle$, any infinite input and guess streams

$$s = (\boldsymbol{u}(t))_{t\in\mathbb{N}} \in (\mathbb{B}^M)^\omega \text{ and } g = (g(t))_{t\in\mathbb{N}} \in 2^\omega$$

induce via Equations (1) and (2) two infinite sequences of states and output states

$$c_{(s,g)} = (\langle \boldsymbol{x}(t), \boldsymbol{y}(t)\rangle)_{t\in\mathbb{N}} \in ([0,1]^N \times \mathbb{B}^P)^\omega$$
$$bc_{(s,g)} = (\boldsymbol{y}(t))_{t\in\mathbb{N}} \in (\mathbb{B}^P)^\omega$$

called the *computation* and *Boolean computation* of $\mathcal{N}$ induced by (s, g), respectively. Furthermore, Definition 1 of an *attractor* remains unchanged in this case.

We also assume that any N-RNN $\mathcal{N}$ is equipped with a corresponding classification of all of its attractors into accepting and rejecting types. An infinite input stream $s \in (\mathbb{B}^M)^\omega$ is *accepted* by $\mathcal{N}$ if there exists some guess stream $g \in 2^\omega$ such that $\inf(bc_{(s,g)})$ is an accepting attractor. It is *rejected* by $\mathcal{N}$ otherwise, i.e., if for all guess streams $g \in 2^\omega$, the set $\inf(bc_{(s,g)})$ is a rejecting attractor. The set of all accepted input streams is the *neural ω-language recognized by $\mathcal{N}$*, denoted by $L(\mathcal{N})$. A set $L \subseteq (\mathbb{B}^M)^\omega$ is said to be *recognizable* by some nondeterministic recurrent neural network if there exists a N-RNN $\mathcal{N}$ such that $L(\mathcal{N}) = L$.

As for the deterministic case, the following classes and subclasses of N-RNNs will be considered according to the nature of their synaptic weights:

1. The class of *nondeterministic static rational neural nets*, denoted by N-St-RNN[$\mathbb{Q}$].
2. The class of *nondeterministic bi-valued evolving rational neural nets*, denoted by N-Ev$_2$-RNN[$\mathbb{Q}$], which is stratified into the subclasses of N-Ev$_2$-RNN[$\mathbb{Q}, \alpha_1, \ldots, \alpha_k$], where $\alpha_1, \ldots, \alpha_k \in 2^\omega$.

4 Expressive Power of Neural Networks

We provide a precise characterization of the expressive power of the evolving neural networks, according to the specific evolving weights that they employ. As a consequence, a proper hierarchy of classes of evolving networks, related to the complexity of their underlying evolving weights, will be obtained in Section 5.

4.1 Deterministic case

The expressive powers of the classes D-St-RNN[$\mathbb{Q}$] and D-Ev$_2$-RNN[$\mathbb{Q}$] have been established in [9, Theorems 1, 2]. We recall these results:

Theorem 1 ([9, Thms 1 & 2]). *Let $L \subseteq (\mathbb{B}^M)^\omega$ be some ω-language.*

(a) L is recognizable by some D-St-RNN$[\mathbb{Q}]$ iff L is recognizable by some deterministic Muller Turing machine iff $L \in BC(\Pi^2_0)$.
(b) L is recognizable by some D-Ev$_2$-RNN$[\mathbb{Q}]$ iff $L \in BC(\mathbf{\Pi^2_0})$.

Theorem 1 states that D-St-RNN$[\mathbb{Q}]$s are Turing equivalent and D-Ev$_2$-RNN$[\mathbb{Q}]$s are strictly more powerful than deterministic Muller Turing machines, since $BC(\Pi^2_0) \subsetneq BC(\mathbf{\Pi^2_0})$. In this sense, the deterministic evolving neural networks are *super-Turing*.

Remark 1. The proof of Theorem 1 (b) [9, Theorem 2] shows that any ω-language $L \in BC(\mathbf{\Pi^2_0})$ can be recognized by some D-Ev$_2$-RNN$[\mathbb{Q}]$ employing only one bi-valued evolving weight given in the form of a bias. In other words, any D-Ev$_2$-RNN$[\mathbb{Q}]$ is expressively equivalent to some D-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$, for some $\alpha \in 2^\omega$. Hence, from this point onwards, we will without loss of generality focus on the latter subclass of networks.

A precise characterization of the expressive power of the subclass of D-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$ can be obtained, for any $\alpha \in 2^\omega$. The result is achieved by forthcoming Propositions 1 and 2.

Proposition 1. *Let $L \subseteq (\mathbb{B}^M)^\omega$ be some ω-language and $\alpha \in 2^\omega$. If $L \in BC(\Pi^2_0)(\alpha)$, then L is recognizable by some D-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$.*

Proof. If $L \in BC(\Pi^2_0)(\alpha) \restriction (\mathbb{B}^M)^\omega$, then by definition, there exists $L' \in BC(\Pi^2_0) \restriction (\mathbb{B}^{M+1})^\omega$ such that $L = L'_\alpha = \{s \in (\mathbb{B}^M)^\omega : (s, \alpha) \in L'\}$. Theorem 1 ensures that there exists a D-St-RNN$[\mathbb{Q}]$ $\mathcal{N}'$ with $M+1$ input cells $u_1, \ldots, u_{M+1}$ such that $L(\mathcal{N}') = L'$.

Now, we consider the D-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$ $\mathcal{N}$ which consists in a slight modification of the D-St-RNN$[\mathbb{Q}]$ $\mathcal{N}'$. More precisely, $\mathcal{N}$ contains the same cells and synaptic connections as $\mathcal{N}'$, it admits only $u_1, \ldots, u_M$ as its input cells, but u_{M+1} is transformed into an internal cell receiving the bi-valued evolving weight $\alpha \in 2^\omega$ in the form of a bias. Moreover, the attractors of $\mathcal{N}$ are the same as those of $\mathcal{N}'$. By construction, on every input $s \in (\mathbb{B}^M)^\omega$, $\mathcal{N}$ receives the bi-valued evolving weight α as bias and it works precisely like $\mathcal{N}'$ on input $(s, \alpha) \in (\mathbb{B}^{M+1})^\omega$. Consequently, $s \in L(\mathcal{N})$ if and only if $(s, \alpha) \in L(\mathcal{N}') = L'$. Therefore, $L(\mathcal{N}) = L'_\alpha = L$, meaning that L is recognized by the D-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$ $\mathcal{N}$. □

Proposition 2. *Let $L \subseteq (\mathbb{B}^M)^\omega$ be some ω-language and $\alpha \in 2^\omega$. If L is recognizable by some D-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$, then $L \in BC(\Pi^2_0)(\alpha)$.*

Proof. Let $\mathcal{N}$ be a D-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$ such that $L(\mathcal{N}) = L$. By Remark 1, we may assume without loss generality that the bi-valued evolving weight α of $\mathcal{N}$ is a bias related to some cell x. Let $\mathcal{N}'$ be the D-St-RNN$[\mathbb{Q}]$ obtained by replacing in $\mathcal{N}$ the evolving bias $\alpha \in 2^\omega$ by a new input cell u_{M+1} related to x with a weight of 1. Hence, $\mathcal{N}'$ is a D-St-RNN$[\mathbb{Q}]$ with $M+1$ input cells, and Theorem 1 ensures that $L(\mathcal{N}') \in BC(\Pi^2_0)$. By construction, if $\mathcal{N}'$ receives input $(s, \alpha) \in (\mathbb{B}^{M+1})^\omega$, then

it works precisely like $\mathcal{N}$ on input $s \in (\mathbb{B}^M)^\omega$, which means that $(s, \alpha) \in L(\mathcal{N}')$ if and only if $s \in L(\mathcal{N})$. Thus $L(\mathcal{N}) = L(\mathcal{N}')_\alpha$. Since $L(\mathcal{N}') \in BC(\Pi^2_0)$, it follows that $L(\mathcal{N}) \in BC(\Pi^2_0)(\alpha)$. □

By combining Propositions 1 and 2, one obtains the following theorem:

Theorem 2. *Let $L \subseteq (\mathbb{B}^M)^\omega$ be some ω-language and $\alpha \in 2^\omega$. The following conditions are equivalent:*

(a) $L \in BC(\Pi^2_0)(\alpha)$;
(b) L is recognizable by some D-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$.

From Theorem 2 and Remark 1, the following set-theoretical result can be retrieved: $BC(\mathbf{\Pi^2_0}) = \bigcup_{\alpha \in 2^\omega} BC(\Pi^2_0)(\alpha)$. Indeed, $L \in BC(\mathbf{\Pi^2_0})$ if and only if, by Remark 1, L is recognizable by some D-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$, for some $\alpha \in 2^\omega$, if and only if, by Theorem 2, $L \in BC(\Pi^2_0)(\alpha)$, for some $\alpha \in 2^\omega$. In words, the relativized classes $BC(\Pi^2_0)(\alpha)$ span the class $BC(\mathbf{\Pi^2_0})$, when α varies over 2^ω.

4.2 Nondeterministic case

The expressive power of the classes N-St-RNN$[\mathbb{Q}]$ and N-Ev$_2$-RNN$[\mathbb{Q}]$ has also been established in [3, Theorems 1, 2]. We have the following results:

Theorem 3 ([3, Thms 1 & 2]). *Let $L \subseteq (\mathbb{B}^M)^\omega$ be some ω-language.*

(a) L is recognizable by some N-St-RNN$[\mathbb{Q}]$ iff $L \in \Sigma^1_1$.
(b) L is recognizable by some N-Ev$_2$-RNN$[\mathbb{Q}]$ iff $L \in \mathbf{\Sigma^1_1}$.

Theorem 3 states that N-St-RNN$[\mathbb{Q}]$s are Turing equivalent and that N-Ev$_2$-RNN$[\mathbb{Q}]$ are strictly more powerful than nondeterministic Muller Turing machines, since $\Sigma^1_1 \subsetneq \mathbf{\Sigma^1_1}$. In this sense, the nondeterministic evolving neural networks are also *super-Turing*.

Remark 2. The nondeterministic counterpart of Remark 1 holds. More precisely, the proof of Theorem 3 (b) [3, Theorem 2] shows that any ω-language $L \in \mathbf{\Sigma^1_1}$ can be recognized by some N-Ev$_2$-RNN$[\mathbb{Q}]$ employing only one bi-valued evolving weight given in the form of a bias. Consequently, from this point onwards, we will without loss of generality focus on the subclass of N-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$, for $\alpha \in 2^\omega$.

We now provide a precise characterization of the expressive power of the subclass of N-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$, for some given $\alpha \in 2^\omega$. The result is achieved via forthcoming Propositions 3 and 4, which are simple generalizations of Propositions 1 and 2, respectively.

Proposition 3. *Let $L \subseteq (\mathbb{B}^M)^\omega$ be some ω-language. If $L \in \Sigma^1_1(\alpha)$, with $\alpha \in 2^\omega$, then L is recognizable by some N-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$.*

Proof. If $L \in \Sigma_1^1(\alpha) \restriction (\mathbb{B}^M)^\omega$, then by definition, there exists $L' \in \Sigma_1^1 \restriction (\mathbb{B}^{M+1})^\omega$ such that $L = L'_\alpha = \{s \in (\mathbb{B}^M)^\omega : (s, \alpha) \in L'\}$. Theorem 3 ensures that there exists a N-St-RNN[$\mathbb{Q}$] $\mathcal{N}'$ with $M+1$ input cells such that $L(\mathcal{N}') = L'$. As in the proof of Proposition 1, one can modify network $\mathcal{N}'$ to obtain a N-Ev$_2$-RNN[$\mathbb{Q}, \alpha$] $\mathcal{N}_1$ such that $L(\mathcal{N}_1) = L'_\alpha = L$. □

Proposition 4. *Let $L \subseteq (\mathbb{B}^M)^\omega$ be some ω-language. If, for some $\alpha \in 2^\omega$, L is recognizable by some N-Ev$_2$-RNN[$\mathbb{Q}, \alpha$], then $L \in \Sigma_1^1(\alpha)$.*

Proof. Let $\mathcal{N}$ be a N-Ev$_2$-RNN[$\mathbb{Q}, \alpha$] such that $L(\mathcal{N}) = L$. By Remark 2, we may assume without loss generality that the bi-valued evolving weight α of $\mathcal{N}$ is a bias. As in the proof of Proposition 2, there exists a N-St-RNN[$\mathbb{Q}$] $\mathcal{N}'$ with $P+1$ input cells such that $(s, \alpha) \in L(\mathcal{N}')$ if and only if $s \in L(\mathcal{N})$. This means that $L(\mathcal{N}) = L(\mathcal{N}')_\alpha$. In addition, Theorem 3 ensures that $L(\mathcal{N}') \in \Sigma_1^1$. Therefore, $L(\mathcal{N}) \in \Sigma_1^1(\alpha)$. □

By combining Propositions 3 and 4, the following theorem is obtained:

Theorem 4. *Let $L \subseteq (\mathbb{B}^M)^\omega$ be some ω-language and $\alpha \in 2^\omega$. The following conditions are equivalent:*

(a) $L \in \Sigma_1^1(\alpha)$;
(b) L is recognizable by some N-Ev$_2$-RNN[$\mathbb{Q}, \alpha$].

From Theorem 4 and Remark 2, the following set-theoretical result can be retrieved: $\mathbf{\Sigma_1^1} = \bigcup_{\alpha \in 2^\omega} \Sigma_1^1(\alpha)$. In other words, the relativized classes $\Sigma_1^1(\alpha)$ span the class $\mathbf{\Sigma_1^1}$, when α varies over 2^ω.

5 The hierarchy theorem

Theorems 2 and 4 provide a precise characterization of the expressive power of the classes of D-Ev$_2$-RNN[$\mathbb{Q}, \alpha$] and N-Ev$_2$-RNN[$\mathbb{Q}, \alpha$], for $\alpha \in 2^\omega$. We first present some conditions that the evolving weights satisfy whenever their corresponding relativized classes are included one into the other.

Proposition 5. *Let $\alpha, \beta \in 2^\omega$. The following relations hold:*

$$BC(\Pi_2^0)(\alpha) \subseteq BC(\Pi_2^0)(\beta) \;\longrightarrow\; \alpha \in \Delta_1^1(\beta) \tag{3}$$

$$\Sigma_1^1(\alpha) \subseteq \Sigma_1^1(\beta) \;\longleftrightarrow\; \alpha \in \Delta_1^1(\beta) \tag{4}$$

Proof. We prove both left-to-right implications. Recall that $\alpha \in \Sigma_1^0(\alpha)$. In the first case, one has $\alpha \in \Sigma_1^0(\alpha) \subseteq BC(\Pi_2^0)(\alpha) \subseteq BC(\Pi_2^0)(\beta) \subseteq \Delta_1^1(\beta)$. In the second case, $\alpha \in \Sigma_1^0(\alpha) \subseteq \Sigma_1^1(\alpha) \subseteq \Sigma_1^1(\beta)$, and thus $\alpha \in \Delta_1^1(\beta)$, by [17].

For the converse implication of relation (4), suppose that $\alpha \in \Delta_1^1(\beta)$. Then $\alpha \in \Sigma_1^1(\beta)$, which means that the ω-language $\{\alpha\}$ is recognized by some nondeterministic Muller TM $\mathcal{M}_1$ with oracle β. Now, let $L \in \Sigma_1^1(\alpha)$. Then L is recognized by a nondeterministic Muller TM $\mathcal{M}_2$ with oracle α. Consider the

nondeterministic Muller TM $\mathcal{M}$ with oracle β which works as follows: if x is written on its input tape, then $\mathcal{M}$ nondeterministically writes some $y \in 2^\omega$ bit by bit on one of its work tape, and concomitantly, it simulates in parallel the behaviors of $\mathcal{M}_1$ on y as well as that of $\mathcal{M}_2$ with oracle y on x. The TM $\mathcal{M}$ is suitably programmed in order to always have enough bits of y being written on its work tape so that the next simulations steps of $\mathcal{M}_1$ with oracle y can be performed without fail. In addition, the machine $\mathcal{M}$ accepts input x iff both simulation processes of $\mathcal{M}_1$ and $\mathcal{M}_2$ are accepting, i.e., iff $y = \alpha$ and the simulation of $\mathcal{M}_2$ with oracle $y = \alpha$ accepts x, which is to say that $x \in L(\mathcal{M}_2) = L$. Hence, $\mathcal{M}$ recognizes L also, and thus $L \in \Sigma^1_1(\beta)$. This shows that $\Sigma^1_1(\alpha) \subseteq \Sigma^1_1(\beta)$. $\square$

We now show the existence of an infinite sequence of weights whose corresponding succession of relativized classes properly stratify the "super-Turing" classes of $BC(\mathbf{\Pi}^0_2)$ and $\mathbf{\Sigma}^1_1$ neural ω-languages. In fact, the hierarchy induced by the inclusion relation between the relativized classes possesses chains of length ω_1 as well as uncountable antichains.

Proposition 6. *There exists a sequence $(\alpha_i)_{i<\omega_1}$, where $\alpha_i \in 2^\omega$ for all $i < \omega_1$, such that*

(a) $BC(\Pi^0_2)(\alpha_0) = BC(\Pi^0_2)$ and $BC(\Pi^0_2)(\alpha_i) \subsetneq BC(\Pi^0_2)(\alpha_j)$, for all $i < j < \omega_1$
(b) $\Sigma^1_1(\alpha_0) = \Sigma^1_1$ and $\Sigma^1_1(\alpha_i) \subsetneq \Sigma^1_1(\alpha_j)$, for all $i < j < \omega_1$

Moreover, there exists some uncountable set $A \subseteq 2^\omega$ such that $BC(\Pi^0_2)(\alpha_i) \not\subseteq BC(\Pi^0_2)(\alpha_j)$ and $\Sigma^1_1(\alpha_i) \not\subseteq \Sigma^1_1(\alpha_j)$, for every distinct $\alpha_i, \alpha_j \in A$.

Proof. Take $\alpha_0 \in \Sigma^0_1$. Suppose that for $\gamma < \omega_1$, the sequence $(\alpha_i)_{i<\gamma}$ has been constructed and satisfies the required property. We build the next element α_γ of that sequence, i.e., the element such that $\Sigma^1_1(\alpha_i) \subsetneq \Sigma^1_1(\alpha_\gamma)$, for all $i < \gamma$. Note that, for each $i < \gamma$, the set $\Delta^1_1(\alpha_i)$ is countable. Since $\gamma < \omega_1$, the union $\bigcup_{i<\gamma} \Delta^1_1(\alpha_i)$ is countable too. Hence, there exists $\alpha \in 2^\omega \setminus \bigcup_{i<\gamma} \Delta^1_1(\alpha_i)$. Now, let $\{\beta_i : i < \omega\}$ be an enumeration of the countable set $\{\alpha\} \cup \{\alpha_i : i < \gamma\}$, and let $\alpha_\gamma \in 2^\omega$ be the encoding of $\{\beta_i : i < \omega\}$ given by $\alpha_\gamma(\langle i, n\rangle) = \beta_i(n)$, where $\langle ., . \rangle : \omega^2 \to \omega$ is a classical recursive pairing function. Each function $f_i : \alpha_\gamma \mapsto (\alpha_\gamma)_i = \beta_i$ is recursive, and therefore, $\beta_i \in \Sigma^0_1(\alpha_\gamma)$, for each $i < \omega$.

We show that $BC(\Pi^0_2)(\alpha_j) \subseteq BC(\Pi^0_2)(\alpha_\gamma)$, for all $j < \gamma$. Let $L \in BC(\Pi^0_2)(\alpha_j) = BC(\Pi^0_2)(\beta_i)$, for some $i < \omega$. This means that L is recognizable by some deterministic Muller TM $\mathcal{M}$ with oracle β_i. Since $\beta_i \in \Sigma^0_1(\alpha_\gamma)$, L is also recognized by the deterministic Muller TM $\mathcal{M}'$ with oracle α_γ which, in a suitable alternating manner, produces β_i bit by bit from α_γ, and works precisely like $\mathcal{M}$ with oracle β_i. Therefore, $L \in BC(\Pi^0_2)(\alpha_\gamma)$. By replacing in this argument every occurrences of "$BC(\Pi^0_2)$" by "Σ^1_1" and of "deterministic" by "nondeterministic", one ontains that $\Sigma^1_1(\alpha_j) \subseteq \Sigma^1_1(\alpha_\gamma)$, for all $j < \gamma$.

We now show that $BC(\Pi^0_2)(\alpha_j) \subsetneq BC(\Pi^0_2)(\alpha_\gamma)$ and $\Sigma^1_1(\alpha_j) \subsetneq \Sigma^1_1(\alpha_\gamma)$, for all $j < \gamma$. Towards a contradiction, suppose that $BC(\Pi^0_2)(\alpha_\gamma) \subseteq BC(\Pi^0_2)(\alpha_j)$ or $\Sigma^1_1(\alpha_\gamma) \subseteq \Sigma^1_1(\alpha_j)$, for some $j < \gamma$. Then Equations (3) and (4) ensure that $\alpha_\gamma \in \Delta^1_1(\alpha_j)$. But $\alpha = \beta_k$ for some $k < \omega$, and by the above stated fact,

$\alpha = \beta_k \in \Sigma^0_1(\alpha_\gamma)$. The two relations $\alpha \in \Sigma^0_1(\alpha_\gamma)$ and $\alpha_\gamma \in \Delta^1_1(\alpha_j)$ imply that $\alpha \in \Delta^1_1(\alpha_j)$. This contradicts the fact that $\alpha \in 2^\omega \setminus \bigcup_{i<\gamma} \Delta^1_1(\alpha_i)$.

We finally prove the existence of an uncountable antichain. There exists an uncountable set $A \subseteq 2^\omega$ such that $\alpha_i \notin \Delta^1_1(\alpha_j)$, for all distinct $\alpha_i, \alpha_j \in A$ [1]. By Equations (3) and (4), $BC(\Pi^0_2)(\alpha_i) \not\subseteq BC(\Pi^0_2)(\alpha_j)$ and $\Sigma^1_1(\alpha_i) \not\subseteq \Sigma^1_1(\alpha_j)$, for all distinct $\alpha_i, \alpha_j \in A$. □

Let $\mathcal{L}(\text{D-Ev}_2\text{-RNN}[\mathbb{Q}, \alpha])$ and $\mathcal{L}(\text{N-Ev}_2\text{-RNN}[\mathbb{Q}, \alpha])$ denote the classes of neural ω-languages recognized by D-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$ and N-Ev$_2$-RNN$[\mathbb{Q}, \alpha]$, respectively. Theorems 2 and 4 together with Proposition 6 imply the existence of two proper hierarchies of classes of deterministic and nondeterministic evolving neural networks of increasing expressive power.

Theorem 5. *There exists a sequence of binary evolving weights $(\alpha_i)_{i<\omega_1}$ such that*

(a) $\mathcal{L}(\text{D-Ev}_2\text{-RNN}[\mathbb{Q}, \alpha_i]) \subsetneq \mathcal{L}(\text{D-Ev}_2\text{-RNN}[\mathbb{Q}, \alpha_j])$, for all $i < j < \omega_1$
(b) $\mathcal{L}(\text{N-Ev}_2\text{-RNN}[\mathbb{Q}, \alpha_i]) \subsetneq \mathcal{L}(\text{N-Ev}_2\text{-RNN}[\mathbb{Q}, \alpha_j])$, for all $i < j < \omega_1$

Finally, let R be the equivalence relation defined by $R(\alpha, \beta)$ iff $\mathcal{L}(\text{N-Ev}_2\text{-RNN}[\mathbb{Q}, \alpha]) = \mathcal{L}(\text{N-Ev}_2\text{-RNN}[\mathbb{Q}, \beta])$. This relation represents the decision problem of whether two classes nondeterministic evolving networks (determined by the evolving weights α and β) have the same expressive power. We show that this relation is undecidable and of complexity of $\Pi^1_1 \setminus \Sigma^1_1$.

Proposition 7. *The equivalence relation R is in the class $\Pi^1_1 \setminus \Sigma^1_1$.*

Proof. According to Theorem 4 and Relation (4), the relation $R \subseteq 2^\omega \times 2^\omega$ satisfies $R(\alpha_1, \alpha_2)$ iff $\alpha_1 \in \Delta^1_1(\alpha_2)$ and $\alpha_2 \in \Delta^1_1(\alpha_1)$. It is known that the relation "$\alpha \in \Delta^1_1(\beta)$" is a Π^1_1 relation which can be expressed by a Π^1_1-formula $\phi(\alpha, \beta)$, see [17, 4D.14 p. 171] and [11]. Thus R is a Π^1_1-relation. Towards a contradiction, assume now that R is Σ^1_1, and take $\beta \in \Sigma^0_1$. Then $R(., \beta) = \{\alpha : R(\alpha, \beta)\} = \{\alpha : \alpha \in \Delta^1_1(\beta) \ \& \ \beta \in \Delta^1_1(\alpha)\} = \{\alpha : \alpha \in \Delta^1_1(\beta)\} = \{\alpha : \alpha \in \Delta^1_1\}$ should also be in Σ^1_1. But it is known that the set $\{\alpha : \alpha \in \Delta^1_1\}$ is not Σ^1_1, see [17, 4D.16 p. 171]. This concludes the proof. □

6 Conclusion

The expressive power of evolving neural networks working on infinite input streams has been finely characterized in terms of relativized topological classes. As a consequence, a proper hierarchy of classes of evolving neural nets, based on the complexity of their underlying evolving weights, has been obtained. The hierarchy contains chains of length ω_1 as well as uncountable antichains.

These results (together with [3, 9]) show that evolving and analog neural networks represent a natural model for oracle-based ω-computation. For future work, a similar refined characterization of the expressive power of analog neural networks is expected to be studied. In fact, we prove in an extended version of this paper that if $r_\alpha \in \mathbb{R}$ is some recursive encoding of $\alpha \in 2^\omega$, then the analog networks employing $r_\alpha \in \mathbb{R}$ as sole real weight are computationally equivalent to the evolving networks employing α as sole evolving weight.

References

1. Apt, K.R.: ω-models in analytical hierarchy. Bulletin de l'académie polonaise des sciences XX(11), 901–904 (1972)
2. Balcázar, J.L., Gavaldà, R., Siegelmann, H.T.: Computational power of neural networks: a characterization in terms of kolmogorov complexity. IEEE Transactions on Information Theory 43(4), 1175–1183 (1997)
3. Cabessa, J., Duparc, J.: Expressive power of nondeterministic recurrent neural networks in terms of their attractor dynamics. IJUC 12(1), 25–50 (2016)
4. Cabessa, J., Siegelmann, H.T.: Evolving recurrent neural networks are super-Turing. In: Proceedings of IJCNN 2011. pp. 3200–3206. IEEE (2011)
5. Cabessa, J., Siegelmann, H.T.: The computational power of interactive recurrent neural networks. Neural Computation 24(4), 996–1019 (2012)
6. Cabessa, J., Siegelmann, H.T.: The super-Turing computational power of plastic recurrent neural networks. Int. J. Neural Syst. 24(8) (2014)
7. Cabessa, J., Villa, A.E.P.: The expressive power of analog recurrent neural networks on infinite input streams. Theor. Comput. Sci. 436, 23–34 (2012)
8. Cabessa, J., Villa, A.E.P.: An attractor-based complexity measurement for boolean recurrent neural networks. PLoS ONE 9(4), e94204+ (2014)
9. Cabessa, J., Villa, A.E.P.: Expressive power of first-order recurrent neural networks determined by their attractor dynamics. J. Comput. Syst. Sci. 82(8), 1232–1250 (2016)
10. Cabessa, J., Villa, A.E.: Artificial Neural Networks: Methods and Applications in Bio-/Neuroinformatics, chap. Recurrent neural networks and super-Turing interactive computation, pp. 1–29. Springer (2015)
11. Finkel, O.: Ambiguity of omega-languages of Turing machines. Logical Methods in Computer Science 10(3) (2014)
12. Kechris, A.S.: Classical descriptive set theory, Graduate Texts in Mathematics, vol. 156. Springer, New York, NY, USA (1995)
13. Kilian, J., Siegelmann, H.T.: The dynamic universality of sigmoidal neural networks. Inf. Comput. 128(1), 48–56 (1996)
14. Kleene, S.C.: Representation of events in nerve nets and finite automata. In: Shannon, C., McCarthy, J. (eds.) Automata Studies, pp. 3–41. Princeton University Press, Princeton, NJ (1956)
15. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysic 5, 115–133 (1943)
16. Minsky, M.L.: Computation: finite and infinite machines. Prentice-Hall, Inc., Englewood Cliffs, N. J. (1967)
17. Moschovakis, Y.N.: Descriptive Set Theory. Mathematical surveys and monographs, American Mathematical Society, second edn. (2009)
18. Siegelmann, H.T.: Recurrent neural networks and finite automata. Computational Intelligence 12, 567–574 (1996)
19. Siegelmann, H.T., Sontag, E.D.: Analog computation via neural networks. Theor. Comput. Sci. 131(2), 331–360 (1994)
20. Siegelmann, H.T., Sontag, E.D.: On the computational power of neural nets. J. Comput. Syst. Sci. 50(1), 132–150 (1995)
21. Síma, J., Orponen, P.: General-purpose computation with neural networks: A survey of complexity theoretic results. Neural Computation 15(12), 2727–2778 (2003)
22. Staiger, L.: ω-languages. In: Handbook of formal languages, vol. 3: beyond words, pp. 339–387. Springer, New York, NY, USA (1997)

23. Turing, A.M.: Intelligent machinery. Technical report, National Physical Laboratory, Teddington, UK (1948)