A Hierarchical Classification of First-Order Recurrent Neural Networks

Jérémie Cabessa[1] and Alessandro E.P. Villa[1,2]

[1] GIN Inserm UMRS 836, University Joseph Fourier, FR-38041 Grenoble
[2] Faculty of Business and Economics, University of Lausanne, CH-1015 Lausanne
{jcabessa,avilla}@nhrg.org

Abstract. We provide a refined hierarchical classification of first-order recurrent neural networks made up of McCulloch and Pitts cells. The classification is achieved by first proving the equivalence between the expressive powers of such neural networks and Muller automata, and then translating the Wadge classification theory from the automata-theoretic to the neural network context. The obtained hierarchical classification of neural networks consists of a decidable pre-well ordering of width 2 and height ω^ω, and a decidability procedure of this hierarchy is provided. Notably, this classification is shown to be intimately related to the attractive properties of the networks, and hence provides a new refined measurement of the computational power of these networks in terms of their attractive behaviours.

1 Introduction

In neural computability, the issue of the computational power of neural networks has often been approached from the automata-theoretic perspective. In this context, McCulloch and Pitts, Kleene, and Minsky already early proved that the class of first-order recurrent neural networks discloses equivalent computational capabilities as classical finite state automata [5,7,8]. Later, Kremer extended this result to the class of Elman-style recurrent neural nets, and Sperduti discussed the computational power of different other architecturally constrained classes of networks [6,15].

Besides, the computational power of first-order recurrent neural networks was also proved to intimately depend on both the choice of the activation function of the neurons as well as the nature of the synaptic weights under consideration. Indeed, Siegelmann and Sontag showed that, assuming rational synaptic weights, but considering a saturated-linear sigmoidal instead of a hard-threshold activation function drastically increases the computational power of the networks from finite state automata up to Turing capabilities [12,14]. In addition, Siegelmann and Sontag also nicely proved that real-weighted networks provided with a saturated-linear sigmoidal activation function reveal computational capabilities beyond the Turing limits [10,11,13].

This paper concerns a more refined characterization of the computational power of neural nets. More precisely, we restrict our attention to the simple

A.-H. Dediu, H. Fernau, and C. Martín-Vide (Eds.): LATA 2010, LNCS 6031, pp. 142–153, 2010.

class of rational-weighted first-order recurrent neural networks made up of McCulloch and Pitts cells, and provide a refined classification of the networks of this class. The classification is achieved by first proving the equivalence between the expressive powers of such neural networks and Muller automata, and then translating the Wadge classification theory from the automata-theoretic to the neural network context [1,2,9,19]. The obtained hierarchical classification of neural networks consists of a decidable pre-well ordering of width 2 and height ω^ω, and a decidability procedure of this hierarchy is provided. Notably, this classification is shown to be intimately related to the attractive properties of the considered networks, and hence provides a new refined measurement of the computational capabilities of these networks in terms of their attractive behaviours.

2 The Model

In this work, we focus on synchronous discrete-time first-order recurrent neural networks made up of classical McCulloch and Pitts cells.

Definition 1. *A first-order recurrent neural network consists of a tuple* $\mathcal{N} = (X, U, a, b, c)$*, where* $X = \{x_i : 1 \leq i \leq N\}$ *is a finite set of* N *activation cells,* $U = \{u_i : 1 \leq i \leq M\}$ *is a finite set of* M *external input cells, and* $a \in \mathbb{Q}^{N \times N}$*,* $b \in \mathbb{Q}^{N \times M}$*, and* $c \in \mathbb{Q}^{N \times 1}$ *are rational matrices describing the weights of the synaptic connections between cells as well as the incoming background activity.*

The activation value of cells x_j and u_j at time t, respectively denoted by $x_j(t)$ and $u_j(t)$, is a boolean value equal to 1 if the corresponding cell is firing at time t and to 0 otherwise. Given the activation values $x_j(t)$ and $u_j(t)$, the value $x_i(t+1)$ is then updated by the following equation

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

where σ is the classical hard-threshold activation function defined by $\sigma(\alpha) = 1$ if $\alpha \geq 1$ and $\sigma(\alpha) = 0$ otherwise.

Note that Equation (1) ensures that the whole dynamics of network $\mathcal{N}$ is described by the following governing equation

$$\boldsymbol{x(t+1)} = \sigma\left(a \cdot \boldsymbol{x(t)} + b \cdot \boldsymbol{u(t)} + c\right), \tag{2}$$

where $\boldsymbol{x(t)} = (x_1(t), \ldots, x_N(t))$ and $\boldsymbol{u(t)} = (u_1(t), \ldots, u_M(t))$ are boolean vectors describing the spiking configuration of the activation and input cells, and σ denotes the classical hard threshold activation function applied component by component. An example of such a network is given below.

Example 1. Consider the network $\mathcal{N}$ depicted in Figure 1. The dynamics of this network is then governed by the following equation:

$$\begin{pmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \end{pmatrix} = \sigma\left[\begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{2} \\ 0 \end{pmatrix}\right]$$

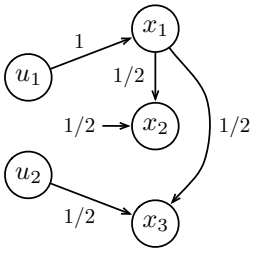


Fig. 1. A simple neural network

3 Attractors

The dynamics of recurrent neural networks made of neurons with two states of activity can implement an associative memory that is rather biological in its details [3]. In the Hopfield framework, stable equilibrium reached by the network that do not represent any valid configuration of the optimization problem are referred to as *spurious attractors*. According to Hopfield et al., spurious modes can disappear by "unlearning" [3], but Tsuda et al. have shown that rational successive memory recall can actually be implemented by triggering spurious modes [17]. Here, the notions of attractors, meaningful attractors, and spurious attractors are reformulated in our precise context. Networks will then be classified according to their ability to switch between different types of attractive behaviours. For this purpose, the following definitions need to be introduced.

As preliminary notations, for any $k > 0$, we let the space of k-dimensional boolean vectors be denoted by $\mathbb{B}^k$, and we let the space of all infinite sequences of k-dimensional boolean vectors be denoted by $[\mathbb{B}^k]^\omega$. Moreover, for any finite sequence of boolean vectors v, we let the expression $v^\omega = vvvv\cdots$ denote the infinite sequence obtained by infinitely many concatenations of v.

Now, let $\mathcal{N}$ be some network with N activation cells and M input cells. For each time step $t \geq 0$, the boolean vectors $\boldsymbol{x(t)} = (x_1(t), \ldots, x_N(t)) \in \mathbb{B}^N$ and $\boldsymbol{u(t)} = (u_1(t), \ldots, u_M(t)) \in \mathbb{B}^M$ describing the spiking configurations of both the activation and input cells of $\mathcal{N}$ at time t are respectively called the *state* of $\mathcal{N}$ at time t and the *input* submitted to $\mathcal{N}$ at time t. An *input stream* of $\mathcal{N}$ is then defined as an infinite sequence of consecutive inputs $s = (\boldsymbol{u(i)})_{i\in\mathbb{N}} = \boldsymbol{u(0)u(1)u(2)}\cdots \in [\mathbb{B}^M]^\omega$. Moreover, assuming the initial state of the network to be $\boldsymbol{x(0)} = \boldsymbol{0}$, any input stream $s = (\boldsymbol{u(i)})_{i\in\mathbb{N}} = \boldsymbol{u(0)u(1)u(2)}\cdots \in [\mathbb{B}^M]^\omega$ induces via Equation (2) an infinite sequence of consecutive states $e_s = (\boldsymbol{x(i)})_{i\in\mathbb{N}} = \boldsymbol{x(0)x(1)x(2)}\cdots \in [\mathbb{B}^N]^\omega$ that is called the *evolution* of $\mathcal{N}$ induced by the input stream s.

Along some evolution $e_s = \boldsymbol{x(0)x(1)x(2)}\cdots$, irrespective of the fact that this sequence is periodic or not, some state will repeat finitely often whereas other will repeat infinitely often. The (finite) set of states occurring infinitely often in the sequence e_s is denoted by $\inf(e_s)$. It can be observed that, for any evolution e_s, there exists a time step k after which the evolution e_s will necessarily remain confined in the set of states $\inf(e_s)$, or in other words, there exists an index k

such that $\boldsymbol{x(i)} \in \inf(e_s)$ for all $i \geq k$. However, along evolution e_s, the recurrent visiting of states in $\inf(e_s)$ after time step k does not necessarily occur in a periodic manner.

Now, given some network $\mathcal{N}$ with N activation cells, a set $A = \{\boldsymbol{y_0}, \ldots, \boldsymbol{y_k}\} \subseteq \mathbb{B}^N$ is called an *attractor* for $\mathcal{N}$ if there exists an input stream s such that the corresponding evolution e_s satisfies $\inf(e_s) = A$. Intuitively, an attractor can be seen a trap of states into which some network's evolution could become forever confined. We further assume that attractors can be of two distinct types, namely *meaningful* or *optimal* vs. *spurious* or *non-optimal*. In this study we do not extend the discussion about the attribution of the attractors to either type. From this point onwards, we assume any given network to be provided with the corresponding classification of its attractors into meaningful and spurious types.

Now, let $\mathcal{N}$ be some network provided with an additional type specification of each of its attractors. The complementary network $\mathcal{N}^{\complement}$ is then defined to be the same network as $\mathcal{N}$ but with an opposite type specification of its attractors.[1] In addition, an input stream s of $\mathcal{N}$ is called *meaningful* if $\inf(e_s)$ is a meaningful attractor, and it is called *spurious* if $\inf(e_s)$ is a spurious attractor. The set of all meaningful input streams of $\mathcal{N}$ is called the *neural language* of $\mathcal{N}$ and is denoted by $L(\mathcal{N})$. Note that the definition of the complementary network implies that $L(\mathcal{N}^{\complement}) = L(\mathcal{N})^{\complement}$. Finally, an arbitrary set of input streams $L \subseteq [\mathbb{B}^M]^\omega$ is defined as *recognizable* by some neural network if there exists a network $\mathcal{N}$ such that $L(\mathcal{N}) = L$. All preceding definitions are now illustrated in the next example.

Example 2. Consider again the network $\mathcal{N}$ described in Example 1, and suppose that an attractor is meaningful for $\mathcal{N}$ if and only if it contains the state $(1,1,1)^T$ (i.e. where the three activation cells simultaneously fire). The periodic input stream $s = [\binom{1}{1}\binom{1}{1}\binom{1}{1}\binom{0}{0}]^\omega$ induces the corresponding periodic evolution

$$e_s = \binom{0}{0}\binom{1}{0}\left[\binom{1}{1}\binom{1}{1}\binom{0}{1}\binom{1}{0}\right]^\omega .$$

Hence, $\inf(e_s) = \{(1,1,1)^T, (0,1,0)^T, (1,0,0)^T\}$, and the evolution e_s of $\mathcal{N}$ remains confined in a cyclic visiting of the states of $\inf(e_s)$ already from time step $t = 2$. Thence, the set $\{(1,1,1)^T, (0,1,0)^T, (1,0,0)^T\}$ is an attractor of $\mathcal{N}$. Moreover, this attractor is meaningful since it contains the state $(1,1,1)^T$.

4 Recurrent Neural Networks and Muller Automata

In this section, we provide an extension of the classical result stating the equivalence of the computational capabilities of first-order recurrent neural networks and finite state machines [5,7,8]. More precisely, here, the issue of the expressive power of neural networks is approached from the point of view of the theory of automata on infinite words, and it is proved that first-order recurrent neural

[1] More precisely, A is a meaningful attractor for $\mathcal{N}^{\complement}$ if and only if A is a spurious attractor for $\mathcal{N}$.

networks actually disclose the very same expressive power as finite Muller automata. Towards this purpose, the following definitions first need to be recalled.

A *finite Muller automaton* is a 5-tuple $\mathcal{A} = (Q, A, i, \delta, \mathcal{T})$, where Q is a finite set called the set of states, A is a finite alphabet, i is an element of Q called the initial state, δ is a partial function from $Q \times A$ into Q called the transition function, and $\mathcal{T} \subseteq \mathcal{P}(Q)$ is a set of set of states called the table of the automaton. A finite Muller automaton is generally represented by a directed labelled graph whose nodes and labelled edges respectively represent the states and transitions of the automaton.

Given a finite Muller automaton $\mathcal{A} = (Q, A, i, \delta, \mathcal{T})$, every triple (q, a, q') such that $\delta(q, a) = q'$ is called a *transition* of $\mathcal{A}$. A *path* in $\mathcal{A}$ is then a sequence of consecutive transitions $\rho = ((q_0, a_1, q_1), (q_1, a_2, q_2), (q_2, a_3, q_3), \ldots)$, also denoted by $\rho : q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} q_3 \cdots$. The path ρ is said to successively *visit* the states $q_0, q_1, \ldots$. The state q_0 is called the *origin* of ρ, the word $a_1 a_2 a_3 \cdots$ is the *label* of ρ, and the path ρ is said to be *initial* if $q_0 = i$. If ρ is an infinite path, the set of states visited infinitely often by ρ is denoted by $\inf(\rho)$. Besides, a *cycle* in $\mathcal{A}$ consists of a finite set of states c such that there exists a finite path in $\mathcal{A}$ with same origin and ending state that visits precisely all the sates of c. A cycle is called *successful* if it belongs to $\mathcal{T}$, and *non-succesful* otherwise. Moreover, an infinite initial path ρ of $\mathcal{A}$ is called *successful* if $\inf(\rho) \in \mathcal{T}$. An infinite word is then said to be *recognized* by $\mathcal{A}$ if it is the label of a successful infinite path in $\mathcal{A}$, and the *ω-language recognized by $\mathcal{A}$*, denoted by $L(\mathcal{A})$, is defined as the set of all infinite words recognized by $\mathcal{A}$. The class of all ω-languages recognizable by some Muller automata is precisely the class of *ω-rational languages*.

Now, for each ordinal $\alpha < \omega^\omega$, we introduce the concept of an α-alternating tree in a Muller automaton $\mathcal{A}$, which consists of a tree-like disposition of the successful and non-successful cycles of $\mathcal{A}$ induced by the ordinal α (see Figure 2). We first recall that any ordinal $0 < \alpha < \omega^\omega$ can uniquely be written of the form $\alpha = \omega^{n_p} \cdot m_p + \omega^{n_{p-1}} \cdot m_{p-1} + \ldots + \omega^{n_0} \cdot m_0$, for some $p \geq 0$, $n_p > n_{p-1} > \ldots > n_0 \geq 0$, and $m_i > 0$. Then, given some Muller automata $\mathcal{A}$ and some ordinal $\alpha = \omega^{n_p} \cdot m_p + \omega^{n_{p-1}} \cdot m_{p-1} + \ldots + \omega^{n_0} \cdot m_0 < \omega^\omega$, an α-alternating tree (resp. α-co-alternating tree) is a sequence of cycles of $\mathcal{A}$ $(C^{i,j}_{k,l})_{i \leq p, j < 2^i, k < m_i, l \leq n_i}$ such that: firstly, $C^{0,0}_{0,0}$ is successful (resp. not successful); secondly, $C^{i,j}_{k,l} \subsetneq C^{i,j}_{k,l+1}$, and $C^{i,j}_{k,l+1}$ is successful iff $C^{i,j}_{k,l}$ is not successful; thirdly, $C^{i,j}_{k+1,0}$ is strictly accessible from $C^{i,j}_{k,0}$, and $C^{i,j}_{k+1,0}$ is successful iff $C^{i,j}_{k,0}$ is not successful; fourthly, $C^{i+1,2j}_{0,0}$ and $C^{i+1,2j+1}_{0,0}$ are both strictly accessible from $C^{i,j}_{m_i-1,0}$, and each $C^{i+1,2j}_{0,0}$ is successful whereas each $C^{i+1,2j+1}_{0,0}$ is not successful. An α-alternating tree is said to be maximal in $\mathcal{A}$ if there is no β-alternating tree in $\mathcal{A}$ such that $\beta > \alpha$.

We now come up to the equivalence of the expressive power of recurrent neural networks and Muller automaton. First of all, we prove that any first-order recurrent neural network can be simulated by some Muller automaton.

Proposition 1. *Let $\mathcal{N}$ be a network provided with a type specification of its attractors. Then there exists a Muller automaton $\mathcal{A}_\mathcal{N}$ such that $L(\mathcal{N}) = L(\mathcal{A}_\mathcal{N})$.*

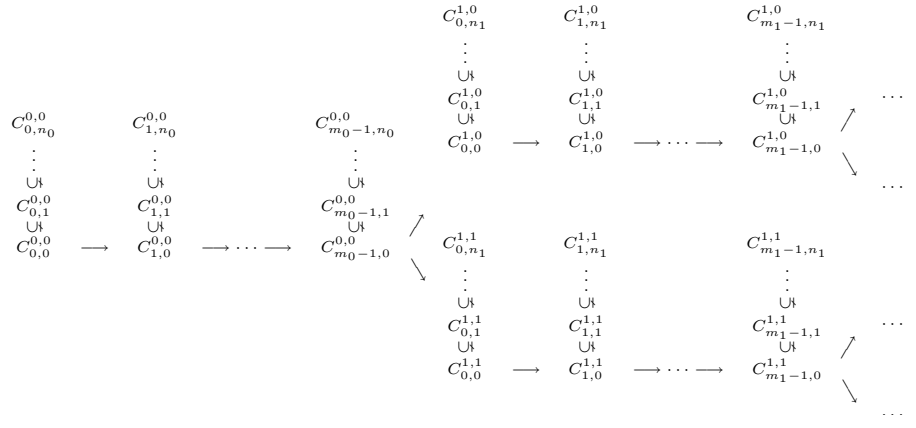


Fig. 2. The inclusion and accessibility relations between cycles in an α-alternating tree

Proof. Let $\mathcal{N}$ be given by the tuple (X, U, a, b, c), with $card(X) = N$, $card(U) = M$, and let the meaningful attractors of $\mathcal{N}$ be given by $A_1, \ldots, A_K$. Now, consider the Muller automaton $\mathcal{A}_{\mathcal{N}} = (Q, A, i, \delta, \mathcal{T})$, where $Q = \mathbb{B}^N$, $A = \mathbb{B}^M$, i is the N-dimensional zero vector, $\delta : Q \times A \to Q$ is defined by $\delta(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{x'}$ if and only if $\boldsymbol{x'} = \sigma\left(a \cdot \boldsymbol{x} + b \cdot \boldsymbol{u} + c\right)$, and $\mathcal{T} = \{A_1, \ldots, A_K\}$. According to this construction, any input stream s of $\mathcal{N}$ is meaningful for $\mathcal{N}$ if and only if s is recognized by $\mathcal{A}_{\mathcal{N}}$. In other words, $s \in L(\mathcal{N})$ if and only if $s \in L(\mathcal{A}_{\mathcal{N}})$, and therefore $L(\mathcal{N}) = L(\mathcal{A}_{\mathcal{N}})$. □

According to the construction given in the proof of Proposition 1, any evolution of the network $\mathcal{N}$ naturally induces a corresponding infinite initial path in the Muller automaton $\mathcal{A}_{\mathcal{N}}$, and conversely, any infinite initial path in $\mathcal{A}_{\mathcal{N}}$ corresponds to some possible evolution of $\mathcal{N}$. This observation ensures the existence of a biunivocal correspondence between *the attractors* of the network $\mathcal{N}$ and *the cycles* in the graph of the corresponding Muller automaton $\mathcal{A}_{\mathcal{N}}$. Consequently, a procedure to compute all possible attractors of a given network $\mathcal{N}$ is simply obtained by first constructing the corresponding Muller automaton $\mathcal{A}_{\mathcal{N}}$ and then listing all cycles in the graph of $\mathcal{A}_{\mathcal{N}}$.

Conversely, we now prove that any Muller automaton can be simulated by some first-order recurrent neural network. For the sake of convenience, we choose to restrict our attention to Muller automata over the binary alphabet $\mathbb{B}^1$.

Proposition 2. *Let $\mathcal{A}$ be some Muller automaton over the alphabet $\mathbb{B}^1$. Then there exists a network $\mathcal{N}_{\mathcal{A}}$ such that $L(\mathcal{A}) = L(\mathcal{N}_{\mathcal{A}})$.*

Proof. Let $\mathcal{A}$ be given by the tuple $(Q, A, q_1, \delta, \mathcal{T})$, with $Q = \{q_1, \ldots, q_N\}$ and $\mathcal{T} \subseteq \mathcal{P}(Q)$. Now, consider the network $\mathcal{N}_{\mathcal{A}} = (X, U, a, b, c)$ defined as follows: First of all, $X = \{x_i : 1 \leq i \leq 2N\} \cup \{x'_1, x'_2, x'_3, x'_4\}$, $U = \{u_1\}$, and each state q_i in the automaton $\mathcal{A}$ gives rise to a two cell layer $\{x_i, x_{N+i}\}$ in the network $\mathcal{N}_{\mathcal{A}}$ as illustrated in Figure 3. Moreover, the synaptic weights between

u_1 and all activation cells, between all cells in $\{x'_1, x'_2, x'_3, x'_4\}$, as well as the background activity are precisely as depicted in Figure 3. Furthermore, for each $1 \leq i \leq N$, both cells x_i and x_{N+i} receive a weighted connection of intensity $\frac{1}{2}$ from cell x'_4 (resp. x'_2) if and only if $\delta(q_1, (0)) = q_i$ (resp. $\delta(q_1, (1)) = q_i$), as also shown in Figure 3. Farther, for each $1 \leq i, j \leq N$, there exist two weighted connection of intensity $\frac{1}{2}$ from cell x_i (resp. from cell x_{N+i}) to both cell x_j and x_{N+j} if and only if $\delta(q_i, (1)) = q_j$ (resp. $\delta(q_i, (0)) = q_j$), as partially illustrated in Figure 3 only for the k-th layer. This description of the network $\mathcal{N}_\mathcal{A}$ ensures that, for any possible evolution of $\mathcal{N}_\mathcal{A}$, the two cells x'_1 and x'_3 are firing at each time step $t \geq 1$, and furthermore, one and only one cell of $\{x_i : 1 \leq i \leq 2N\}$ are firing at each time step $t \geq 2$. According to this observation, for any $1 \leq j \leq N$, let $\mathbf{1}_j \in \mathbb{B}^{2N+4}$ (resp. $\mathbf{1}_{N+j} \in \mathbb{B}^{2N+4}$) denote the boolean vector describing the spiking configuration where only the cells x'_1, x'_3, and x_j (resp. x'_1, x'_3, and x_{N+j}) are firing. Hence, any evolution $\boldsymbol{x(0)x(1)x(2)}\cdots$ of $\mathcal{N}_\mathcal{A}$ satisfies $\boldsymbol{x(t)} \in \{\mathbf{1}_k : 1 \leq k \leq N\} \cup \{\mathbf{1}_{N+l} : 1 \leq l \leq N\}$ for all $t \geq 2$, and thus any attractor A of $\mathcal{N}$ can necessarily be written of the form $A = \{\mathbf{1}_k : k \in K\} \cup \{\mathbf{1}_{N+l} : l \in L\}$, for some $K, L \subseteq \{1, 2, \ldots, N\}$. Now, any infinite sequence $s = \boldsymbol{u(0)u(1)u(2)}\cdots \in [\mathbb{B}^1]^\omega$ induces both a corresponding infinite path $\rho_s : q_1 \xrightarrow{\boldsymbol{u(0)}} q_{j_1} \xrightarrow{\boldsymbol{u(1)}} q_{j_2} \xrightarrow{\boldsymbol{u(2)}} q_{j_3} \cdots$ in $\mathcal{A}$ as well as a corresponding evolution $e_s = \boldsymbol{x(0)x(1)x(2)}\cdots$ in $\mathcal{N}_\mathcal{A}$. The network $\mathcal{N}_\mathcal{A}$ is then related to the automaton $\mathcal{A}$ via the following important property: for each time step $t \geq 1$, if $\boldsymbol{u(t)} = (1)$, then $\boldsymbol{x(t+1)} = \mathbf{1}_{j_t}$, and if $\boldsymbol{u(t)} = (0)$, then $\boldsymbol{x(t+1)} = \mathbf{1}_{N+j_t}$. In other words, the infinite path ρ_s and the evolution e_s evolve in parallel and satisfy the property that the cell x_j is spiking in $\mathcal{N}_\mathcal{A}$ if and only if the automaton $\mathcal{A}$ is in state q_j and reads letter (1), and the cell x_{N+j} is spiking in $\mathcal{N}_\mathcal{A}$ if and only if the automaton $\mathcal{A}$ is in state q_j and reads letter (0). Finally, an attractor $A = \{\mathbf{1}_k : k \in K\} \cup \{\mathbf{1}_{N+l} : l \in L\}$ with $K, L \subseteq \{1, 2, \ldots, N\}$ is set to be meaningful if and only if $\{q_k : k \in K\} \cup \{q_l : l \in L\} \in \mathcal{T}$. Consequently, for any infinite infinite sequence $s \in [\mathbb{B}^1]^\omega$, the infinite path ρ_s in $\mathcal{A}$ satisfies $\inf(\rho_s) \in \mathcal{T}$

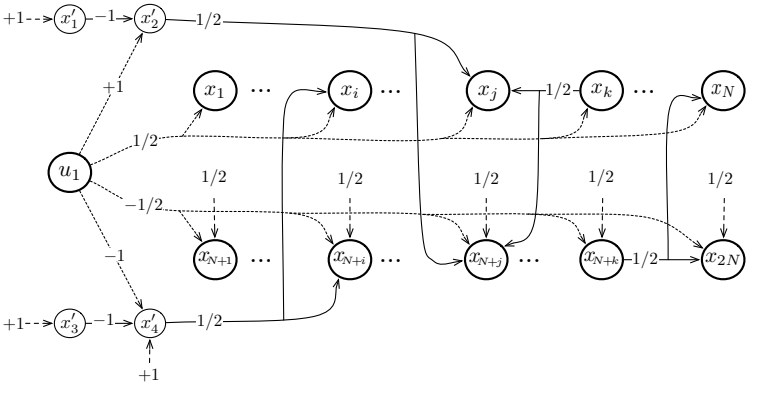
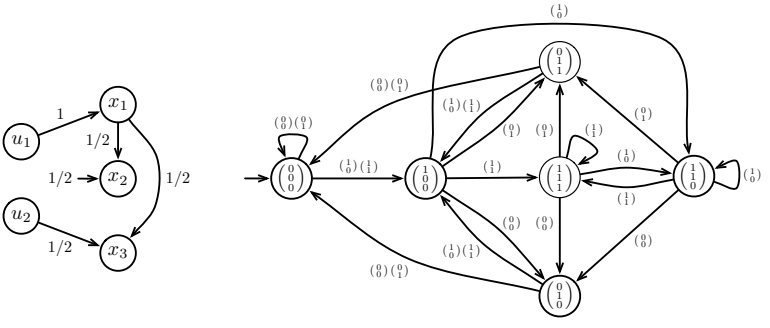


Fig. 3. The network $\mathcal{N}_\mathcal{A}$

if and only if the evolution e_s in $\mathcal{N}_\mathcal{A}$ is such that $\inf(e_s)$ is a meaningful attractor. Therefore, $L(\mathcal{A}) = L(\mathcal{N}_\mathcal{A})$. □

Finally, the following example provides an illustration of the two translating procedures described in the proofs of propositions 1 and 2.

Example 3. The translation from the network $\mathcal{N}$ described in Example 2 to its corresponding Muller automaton $\mathcal{A}_\mathcal{N}$ is illustrated in Figure 4. Proposition 1 ensures that $L(\mathcal{N}) = L(\mathcal{A}_\mathcal{N})$. Conversely, the translation from some given Muller automaton $\mathcal{A}$ over the alphabet $\mathbb{B}^1$ to its corresponding network $\mathcal{N}_\mathcal{A}$ is illustrated in Figure 5. Proposition 2 ensures that $L(\mathcal{A}) = L(\mathcal{N}_\mathcal{A})$.



$A \subseteq \mathbb{B}^3$ is meaningful for $\mathcal{N}$ if and only if $(1,1,1)^T \in A$

Table $\mathcal{T} = \{A \in \mathbb{B}^3 : A \text{ is meaningful for } \mathcal{N}\}$

Fig. 4. Translation from a given network $\mathcal{N}$ provided with a type specification of its attractors to a corresponding Muller automaton $\mathcal{A}_\mathcal{N}$

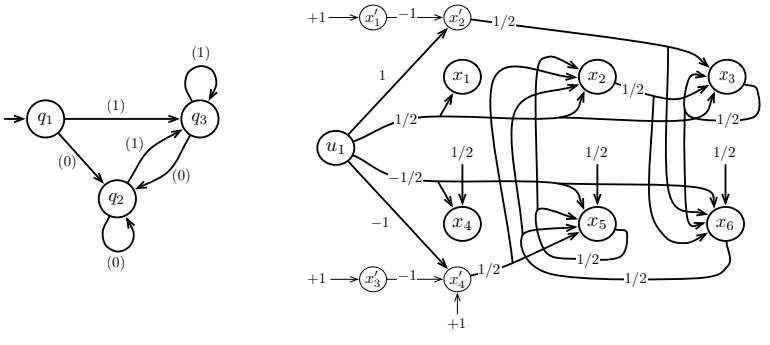


Table $\mathcal{T} = \{\{q_2\}, \{q_3\}\}$

Meaningful attractors: $A_1 = \{\mathbf{1}_5\}$ and $A_2 = \{\mathbf{1}_3\}$.

Fig. 5. Translation from a given Muller automaton $\mathcal{A}$ to a corresponding network $\mathcal{N}_\mathcal{A}$ provided with a type specification of its attractors

5 The RNN Hierarchy

In the theory of automata on infinite words, abstract machines are commonly classified according the topological complexity of their underlying ω-language, as for instance in [1,2,9,19]. Here, this approach is translated from the automata to the neural networks context, in order to obtain a refined classification of first-order recurrent neural networks. Notably, the obtained classification actually refers to the ability of the networks to switch between meaningful and spurious attractive behaviours.

For this purpose, the following facts and definitions need to be introduced. To begin with, for any $k > 0$, the space $[\mathbb{B}^k]^\omega$ can naturally be equipped with the product topology of the discrete topology over $\mathbb{B}^k$. Thence, a function $f : [\mathbb{B}^k]^\omega \to [\mathbb{B}^l]^\omega$ is said to be continuous if and only if the inverse image by f of every open set of $[\mathbb{B}^l]^\omega$ is an open set of $[\mathbb{B}^k]^\omega$. Now, given two first-order recurrent neural networks $\mathcal{N}_1$ and $\mathcal{N}_2$ with M_1 and M_2 input cells respectively, we say that $\mathcal{N}_1$ *Wadge reduces* [18] (or *continuously reduces* or simply *reduces*) to $\mathcal{N}_2$, denoted by $\mathcal{N}_1 \leq_W \mathcal{N}_2$, if any only if there exists a continuous function $f : [\mathbb{B}^{M_1}]^\omega \to [\mathbb{B}^{M_2}]^\omega$ such that any input stream s of $\mathcal{N}_1$ satisfies $s \in L(\mathcal{N}_1) \Leftrightarrow f(s) \in L(\mathcal{N}_2)$. The corresponding strict reduction, equivalence relation, and incomparability relation are then naturally defined by $\mathcal{N}_1 <_W \mathcal{N}_2$ iff $\mathcal{N}_1 \leq_W \mathcal{N}_2 \not\leq_W \mathcal{N}_1$, as well as $\mathcal{N}_1 \equiv_W \mathcal{N}_2$ iff $\mathcal{N}_1 \leq_W \mathcal{N}_2 \leq_W \mathcal{N}_1$, and $\mathcal{N}_1 \perp_W \mathcal{N}_2$ iff $\mathcal{N}_1 \not\leq_W \mathcal{N}_2 \not\leq_W \mathcal{N}_1$. Moreover, a network $\mathcal{N}$ is called *self-dual* if $\mathcal{N} \equiv_W \mathcal{N}^{\complement}$; it is *non-self-dual* if $\mathcal{N} \not\equiv_W \mathcal{N}^{\complement}$, which can be proved to be equivalent to saying that $\mathcal{N} \perp_W \mathcal{N}^{\complement}$. By extension, an $\equiv_W$-equivalence class of networks is called *self-dual* if all its elements are self-dual, and *non-self-dual* if all its elements are non-self-dual.

Now, the Wadge reduction over the class of neural networks naturally induces a hierarchical classification of networks. Formally, the collection of all first-order recurrent neural networks ordered by the Wadge reduction "$\leq_W$" is called the *RNN hierarchy*.

Propositions 1 and 2 ensure that the RNN hierarchy and the Wagner hierarchy – the collection of all ω-rational languages ordered by the Wadge reduction [19] – coincide up to Wadge equivalence. Accordingly, a precise description of the RNN hierarchy can therefore be given as follows. First of all, the RNN hierarchy is well founded, i.e. there is no infinite strictly descending sequence of networks $\mathcal{N}_0 >_W \mathcal{N}_1 >_W \mathcal{N}_2 >_W \ldots$. Moreover, the maximal strict chains in the RNN hierarchy have length ω^ω, meaning that the RNN hierarchy has a height of ω^ω. Furthermore, the maximal antichains of the RNN hierarchy have length 2, meaning that the RNN hierarchy has a width of 2.[2] More precisely, any two networks $\mathcal{N}_1$ and $\mathcal{N}_2$ satisfy the incomparability relation $\mathcal{N}_1 \perp_W \mathcal{N}_2$ if and only if $\mathcal{N}_1$ and $\mathcal{N}_2$ are non-self-dual networks such that $\mathcal{N}_1 \equiv_W \mathcal{N}_2^{\complement}$. These properties imply that, up to Wadge equivalence and complementation, the RNN

[2] A strict chain (resp. an antichain) in the RNN hierarchy is a sequence of neural networks $(\mathcal{N}_k)_{k\in\alpha}$ such that $\mathcal{N}_i <_W \mathcal{N}_j$ iff $i < j$ (resp. such that $\mathcal{N}_i \perp_W \mathcal{N}_j$ for all $i, j \in \alpha$ with $i \neq j$). A strict chain (resp. an antichain) is said to be maximal if its length is at least as large as the length of every other strict chain (resp. antichain).

hierarchy is actually a well-ordering. In fact, the RNN hierarchy consists of an alternating succession of non-self-dual and self-dual classes with pairs of non-self-dual classes at each limit level, as illustrated in Figure 6, where circle represent the Wadge equivalence classes of networks and arrows between circles represent the strict Wadge reduction between all elements of the corresponding classes. For convenience reasons, the degree of a network $\mathcal{N}$ in the RNN hierarchy is now defined in order to make the non-self-dual (n.s.d.) networks and the self-dual ones located just one level above share the same degree, as illustrated in Figure 6:

$$d(\mathcal{N}) = \begin{cases} 1 & \text{if } L(\mathcal{N}) = \emptyset \text{ or } \emptyset^{\complement}, \\ \sup\{d(\mathcal{M}) + 1 : \mathcal{M} \text{ n.s.d. and } \mathcal{M} <_W \mathcal{N}\} & \text{if } \mathcal{N} \text{ is non-self-dual}, \\ \sup\{d(\mathcal{M}) : \mathcal{M} \text{ n.s.d. and } \mathcal{M} <_W \mathcal{N}\} & \text{if } \mathcal{N} \text{ is self-dual}. \end{cases}$$

Also, the equivalence between the Wagner and RNN hierarchies ensure that the RNN hierarchy is actually decidable, in the sense that there exists a algorithmic procedure computing the degree of any network in the RNN hierarchy. All the above properties of the RNN hierarchy are summarized in the following result.

Theorem 1. *The RNN hierarchy is a decidable pre-well-ordering of width* 2 *and height* ω^ω*.*

Proof. The Wagner hierarchy consists of a decidable pre-well-ordering of width 2 and height ω^ω [19]. Propositions 1 and 2 ensure that the RNN hierarchy and Wagner hierarchy coincide up to Wadge equivalence. □

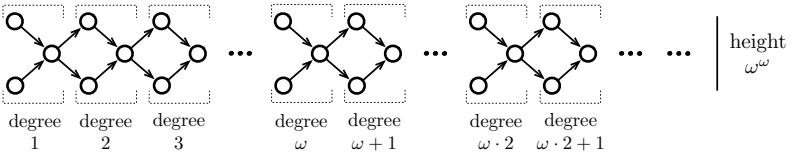


Fig. 6. The RNN hierarchy

The following result provides a detailed description of the decidability procedure of the RNN hierarchy. More precisely, it is shown that the degree of a network $\mathcal{N}$ in the RNN hierarchy corresponds precisely to the largest ordinal α such that there exists an α-alternating tree or an α-co-alternating tree in the Muller automaton $\mathcal{A}_\mathcal{N}$.

Theorem 2. *Let* $\mathcal{N}$ *be a network provided with a type specification of its attractors,* $\mathcal{A}_\mathcal{N}$ *be the corresponding Muller automaton of* $\mathcal{N}$*, and* α *be an ordinal such that* $0 < \alpha < \omega^\omega$*.*

- *If there exists in* $\mathcal{A}_\mathcal{N}$ *a maximal* α*-alternating tree and no maximal* α*-co-alternating tree, then* $d(\mathcal{N}) = \alpha$ *and* $\mathcal{N}$ *is non-self-dual.*

- *If there exists in $\mathcal{A}_\mathcal{N}$ a maximal α-co-alternating tree and no maximal α-alternating tree, then $d(\mathcal{N}) = \alpha$ and $\mathcal{N}$ is non-self-dual.*
- *If there exist in $\mathcal{A}_\mathcal{N}$ both a maximal α-alternating tree as well as a maximal α-co-alternating tree, then $d(\mathcal{N}) = \alpha$ and $\mathcal{N}$ is self-dual.*

Proof. For any ω-rational language L, let $d_W(L)$ denote the degree of L in the Wagner hierarchy. On the one hand, propositions 1 and 2 ensure that $d(\mathcal{N}) = d_W(L(\mathcal{A}_\mathcal{N}))$. On the other hand, the decidability procedure of the Wagner hierarchy states that $d_W(L(\mathcal{A}_\mathcal{N}))$ corresponds precisely to the largest ordinal α such that there exists a maximal α-(co)-alternating tree in $\mathcal{A}_\mathcal{N}$ [19]. □

The decidability procedure of the degree of a network $\mathcal{N}$ in the the RNN hierarchy thus consists in first translating the network $\mathcal{N}$ into its corresponding Muller automaton $\mathcal{A}_\mathcal{N}$ (as described in Proposition 1), and then returning the ordinal α associated to the maximal α-(co)-alternating tree(s) in contained in $\mathcal{A}_\mathcal{N}$ (which can be achieved by some graph analysis of the automaton $\mathcal{A}_\mathcal{N}$). In other words, the complexity of a network $\mathcal{N}$ is directly related to the relative disposition of the successful and non-successful cycles in the Muller automaton $\mathcal{A}_\mathcal{N}$, or in other words, to how some infinite path in $\mathcal{A}_\mathcal{N}$ could maximally alternate between successful and non-successful cycles along its evolution. Therefore, according to the biunivocal correspondence between cycles in $\mathcal{A}_\mathcal{N}$ and attractors of $\mathcal{N}$, as well as between infinite paths in $\mathcal{A}_\mathcal{N}$ and evolutions of the network $\mathcal{N}$, it follows that the complexity of a network $\mathcal{N}$ in the RNN hierarchy actually refers to the capacity of this network to maximally alternate between punctual visitings of meaningful and spurious attractors along some possible evolution – a concept close to chaotic itinerancy [16,4].

Example 4. Let $\mathcal{N}$ be the network of Example 2. Then $d(\mathcal{N}) = \omega$ and $\mathcal{N}$ is non-self-dual. Indeed, $\{(0,0,0)^T\} \subsetneq \{(0,0,0)^T, (1,0,0)^T, (1,1,1)^T, (0,1,1)^T\}$ is a maximal ω^1-co-alternating tree in the Muller automaton $\mathcal{A}_\mathcal{N}$ of Figure 4.

6 Conclusion

The present work proposes a new approach of neural computability from the point of view infinite word reading automata theory. More precisely, the Wadge classification of infinite word languages is translated from the automata-theoretic to the neural network context, and a transfinite decidable hierarchical classification of first-order recurrent neural network is obtained. This classification provides a better understanding of this simple class of neural networks that could be relevant for implementation issues. Moreover, the Wadge hierarchies of deterministic pushdown automata or deterministic Turing Machines both with Muller conditions [1,9] ensure that such Wadge-like classifications of strictly more powerful models of neural networks could also be described; however, in these cases, the decidability procedures of the obtained hierarchies remain hard open problems.

Besides, this work is envisioned to be extended in several directions. First of all, it could be of interest to study the same kind of hierarchical classification

applied to more biologically oriented models, like neural networks provided with some additional simple STDP rule. In addition, neural networks' computational capabilities should also rather be approached from the point of view of finite word instead of infinite word reading automata, as for instance in [6,10,11,12,13,14,15]. Unfortunately, as opposed to the case of infinite words, the classification theory of finite words reading machines is still a widely undeveloped, yet promising issue. Finally, the study of hierarchical classifications of neural networks induced by more biologically oriented reduction relations than the Wadge reduction would be of specific interest.

References

1. Duparc, J.: A hierarchy of deterministic context-free ω-languages. Theor. Comput. Sci. 290(3), 1253–1300 (2003)
2. Finkel, O.: An effective extension of the Wagner hierarchy to blind counter automata. In: Fribourg, L. (ed.) CSL 2001 and EACSL 2001. LNCS, vol. 2142, pp. 369–383. Springer, Heidelberg (2001)
3. Hopfield, J.J., Feinstein, D.I., Palmer, R.G.: 'unlearning' has a stabilizing effect in collective memories. Nature 304, 158–159 (1983)
4. Kaneko, K., Tsuda, I.: Chaotic itinerancy. Chaos 13(3), 926–936 (2003)
5. Kleene, S.C.: Representation of events in nerve nets and finite automata. In: Automata Studies. Annals of Mathematics Studies, vol. 34, pp. 3–42. Princeton University Press, Princeton (1956)
6. Kremer, S.C.: On the computational power of elman-style recurrent networks. IEEE Transactions on Neural Networks 6(4), 1000–1004 (1995)
7. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysic 5, 115–133 (1943)
8. Minsky, M.L.: Computation: finite and infinite machines. Prentice-Hall, Inc., Upper Saddle River (1967)
9. Selivanov, V.: Wadge degrees of ω-languages of deterministic Turing machines. Theor. Inform. Appl. 37(1), 67–83 (2003)
10. Siegelmann, H.T.: Computation beyond the Turing limit. Science 268(5210), 545–548 (1995)
11. Siegelmann, H.T.: Neural and super-Turing computing. Minds Mach. 13(1), 103–114 (2003)
12. Siegelmann, H.T., Sontag, E.D.: Turing computability with neural nets. Applied Mathematics Letters 4(6), 77–80 (1991)
13. Siegelmann, H.T., Sontag, E.D.: Analog computation via neural networks. Theor. Comput. Sci. 131(2), 331–360 (1994)
14. Siegelmann, H.T., Sontag, E.D.: On the computational power of neural nets. J. Comput. Syst. Sci. 50(1), 132–150 (1995)
15. Sperduti, A.: On the computational power of recurrent neural networks for structures. Neural Netw. 10(3), 395–400 (1997)
16. Tsuda, I.: Chaotic itinerancy as a dynamical basis of hermeneutics of brain and mind. World Futures 32, 167–184 (1991)
17. Tsuda, I., Koerner, E., Shimizu, H.: Memory dynamics in asynchronous neural networks. Prog. Th. Phys. 78(1), 51–71 (1987)
18. Wadge, W.W.: Reducibility and determinateness on the Baire space. PhD thesis, University of California, Berkeley (1983)
19. Wagner, K.: On ω-regular sets. Inform. and Control 43(2), 123–177 (1979)