Chinese Journal of Physiology 53(6): 407-416, 2010
DOI: 10.4077/CJP.2010.AMM037

A Hierarchical Classification of First-Order Recurrent Neural Networks

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

[1]*Grenoble Institut des Neurosciences (GIN), INSERM, UMR_S 836, NeuroHeuristic Research Group*
Université Joseph Fourier, Grenoble, France
and
[2]*Neuroheuristic Research Group, Information Systems Department ISI,*
University of Lausanne, Switzerland

Abstract

We provide a decidable hierarchical classification of first-order recurrent neural networks made up of McCulloch and Pitts cells. This classification is achieved by proving an equivalence result between such neural networks and deterministic Büuchi automata, and then translating the Wadge classification theory from the abstract machine to the neural network context. The obtained hierarchy of neural networks is proved to have width 2 and height $\omega + 1$, 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.

Key Words: neural networks, attractors, Büchi automata, Wadge hierarchy

Introduction

The characteristic feature of a recurrent neural network (RNN) is that the connections between the cells form a directed cycle. In the automata-theoretic perspective McCulloch and Pitts (9), Kleene (7), and Minsky (10) proved that the class of first-order RNN discloses equivalent computational capabilities as classical finite state automata. Kremer extended this result to the class of Elman-style recurrent neural nets (8) and Sperduti discussed the computational power of other architecturally constrained classes of networks (18).

The computational power of first-order RNN depend on both the choice of the neuronal activation function and the nature of the synaptic weights. Assuming rational synaptic weights and a saturated-linear sigmoidal activation function, instead of a hard-threshold, Siegelmann and Sontag showed that the computational power of the networks drastically increases from finite state automata up to Turing capabilities (15, 17). Moreover, real-weighted networks provided with a saturated-linear sigmoidal activation function reveal computational capabilities beyond the Turing limits (13, 14, 16). Kilian and Siegelmann extended the Turing universality of neural networks to a more general class of sigmoidal activation functions (6). These results are of primary importance in order to understand the computational powers of different classes of neural networks.

In this paper we focus on a given class of neural networks and then we analyze the computational capabilities of each individual network of this class, instead of addressing the issue of the computational power of a whole given class of neural networks. More precisely, we restrict our attention on the class of first-order RNN made up of McCulloch and Pitts cells, and provide an internal transfinite hierarchical classification of the networks of this class according to their computational capabilities. This classification is achieved by proving an equivalence result between the considered neural networks and deterministic Büchi automata, and then translating theWadge classification theory (2-4, 12, 22) from the abstract machine to the neural network context. It is then shown that the degree of a network in the obtained hierarchy

Corresponding author: Dr. Jérémie Cabessa, Grenoble Institut des Neurosciences (GIN), INSERM, UMR_S 836, Equipe 7, Université Joseph Fourier, Grenoble, France, La Tronche BP 170, F-38042 Grenoble Cedex 9, France. Fax: +33-456-520369, E-mail: [jcabessa, avilla]@nhrg.org
Received: April 23, 2010; Revised: May 22, 2010; Accepted: May 23, 2010.
 ISSN : 0304-4920. http://www.cps.org.tw

corresponds precisely to the maximal capability of the network to punctually alternate between attractors of different types along its evolution.

The Model

In this paper, we consider discrete-time first-order RNN made up of classical McCulloch and Pitts cells (9). More precisely, our model consists of a synchronous network whose architecture is specified by a general directed graph with edges labelled by rational weights. The nodes of the graph are called cells (or processors) and the labelled edges are the synaptic connections between those. At every time step, the state of each cell can be of only two kinds, namely either firing or quiet. When firing, each cell instantaneously transmits a post-synaptic potential (p.s.p.) throughout each of its efferent projections with an amplitude determined by the weight of the synaptic connection (equal to the label of the edge). Then, any given cell will be firing at time $t + 1$ if and only if (denoted iff) the sum of all p.s.p. transmitted at time t plus the effect of background activity exceeds its threshold (which we suppose without loss of generality to be equal to 1). From now further the value of the p.s.p. is referred to as "intensity". As already mentioned, such networks have been proved to reveal same computational capabilities as finite state automata (7, 9, 10). The definition of such a network can be formalised as follows:

Definition 0.1. *A first-order recurrent neural network (RNN) consists of a tuple* $\mathcal{N} = (X, S, M, a, b, c)$*, where:* $X = \{x_i : 1 \le i \le N\}$ *is a finite set of N activation cells,* $S = \{s_i : 1 \le i \le K\}$ *is a finite set of K external sensory cells,* $M \subseteq X$ *is a distinguished subset of motor cells,* $a \in \mathbb{Q}^{X \times X}$ *and* $b \in \mathbb{Q}^{X \times U}$ *describe the weights of the synaptic connections between all cells, and* $c \in \mathbb{Q}^X$ *describes the afferent background activity, or bias.*[1] *The activation value of cells* x_j *and* s_j *at time t, denoted by* $x_j(t)$ *and* $s_j(t)$*, respectively, 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* $s_j(t)$*, the value* $x_i(t + 1)$ *is then updated by the following equation*

$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{i,j} x_j(t) + \sum_{j=1}^{K} b_{i,j} s_j(t) + c_i\right), \quad i = 1, \cdots, N \qquad [1]$$

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

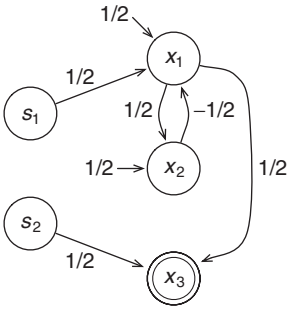

Fig. 1. A simple first-order recurrent neural network.

Note that Equation [1] ensures that the dynamics of any RNN $\mathcal{N}$ can be equivalently described by a discrete dynamical system of the form

$$\boldsymbol{x(t+1)} = \sigma(A \cdot \boldsymbol{x(t)} + B \cdot \boldsymbol{s(t)} + \boldsymbol{c}), \qquad [2]$$

where $\boldsymbol{x(t)} = (x_1(t), \cdots, x_N(t))$ and $\boldsymbol{s(t)} = (s_1(t), \cdots, s_K(t))$ are boolean vectors, A, B, and $\boldsymbol{c}$ are rational matrices of sizes $N \times N$, $N \times K$, and $N \times 1$, respectively, and σ denotes the classical hard threshold activation function applied component by component. An example of such a network is given below.

Example 0.2. Consider the network $\mathcal{N}$ depicted in Fig. 1. This network consists of two sensory cells s_1 and s_2, three activation cells x_1, x_2, and x_3, among which only x_3 is a motor cell. The network contains five connections, as well as a constant background activity, or bias, of intensity 1/2 transmitted to x_1 and x_2. The dynamics of this network is then governed by the following system of equations:

$$\begin{pmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \end{pmatrix} = \sigma\left[\begin{pmatrix} 0 & -\frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{pmatrix}\right]$$

Meaningful and Spurious Attractors

Given some RNN $\mathcal{N}$ with N activation cells and K sensory cells, the boolean vector $\boldsymbol{x(t)} = (x_1(t), \cdots, x_N(t))$ describing the spiking configuration of the activation cells of $\mathcal{N}$ at time t is called the *state* of $\mathcal{N}$ at time t. The K-dimensional boolean vector $\boldsymbol{s(t)} =$

[1]From this point forward, for every indices i and j, the terms $a(x_i, x_j)$, $b(x_i, s_j)$ and $c(x_i)$ will be denoted by $a_{i,j}$, $b_{i,j}$, and c_i, respectively.

$(s_1(t), \cdots, s_K(t))$ describing the spiking configuration of the sensory cells of $\mathcal{N}$ at time t is called the *stimulus* submitted to $\mathcal{N}$ at time t. The set of all K-dimensional boolean vectors $\mathbb{B}^K$ then corresponds to the set of all possible stimuli of $\mathcal{N}$. A *stimulation* of $\mathcal{N}$ is then defined as an infinite sequence of consecutive stimuli $s = (\boldsymbol{s(i)})_{i\in\mathbb{N}} = \boldsymbol{s(0)s(1)s(2)}\cdots$. The set of all infinite sequences of K-dimensional boolean vectors, denoted by $[\mathbb{B}^K]^\omega$, thus corresponds to the set of all possible stimulations of $\mathcal{N}$. Let us assume the initial state to be $\boldsymbol{x(0)} = \boldsymbol{0}$, any stimulation $s = (\boldsymbol{s(i)})_{i\in\mathbb{N}} = \boldsymbol{s(0)s(1)s(2)}\cdots$ induces *via* Equation [2] an infinite sequence of consecutive states $e_s = (\boldsymbol{x(i)})_{i\in\mathbb{N}} = \boldsymbol{x(0)x(1)x(2)}\cdots$ that will be called the *evolution* of $\mathcal{N}$ under stimulation 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 will then be denoted by $inf(e_s)$. It is worth noting 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 visit of states in $inf(e_s)$ after time step k does not necessarily occur in a periodic manner.

In this work, the attractive behaviours of neural networks is an issue of key importance, and networks will further be classified according to their ability to switch between attractors of different types. Towards this purpose, the following definition needs to be introduced.

Definition 0.3. *Given a RNN $\mathcal{N}$ with N activation cells, a set of N-dimentional boolean vectors $A = \{\boldsymbol{y_0}, \cdots, \boldsymbol{y_k}\}$ is called an attractor for $\mathcal{N}$ if there exists a stimulation s such that the corresponding evolution e_s satisfies $inf(e_s) = A$.*

In other words, an attractor is a set of states into which some evolution of a network could eventually become confined for ever. It can be seen as a trap of states into which the network's behaviour could eventually get attracted in a never-ending cyclic but not necessarily periodic visit. Note that an attractor necessarily consists of a finite set of states (since the set of all possible states of $\mathcal{N}$ is finite).

We suppose further that attractors can be of two distinct types, namely either *meaningful* or *spurious*. More precisely, an attractor $A = \{\boldsymbol{y_0}, \cdots, \boldsymbol{y_k}\}$ of $\mathcal{N}$ is called *meaningful* if it contains at least one element $\boldsymbol{y_i}$ describing a spiking configuration of the system where some motor cell is spiking, *i.e.* if there exist $i \leq k$ and $j \leq N$ such that x_j is a motor cell and the j-th component of $\boldsymbol{y_i}$ is equal to 1. An attractor A is called *spurious* otherwise. Notice that by the term "motor" we refer more generally to a cell involved in producing a behaviour. Hence, meaningful attractors intuitively refer to the cyclic activity of the network that induce some motor/behavioural response of the system, whereas spurious attractors refer to the cyclic activity of the network that do not evoke any motor/behavioural response at all. More precisely, an evolution e_s such that $inf(e_s)$ is a meaningful attractor will necessarily induce infinitely many motor responses of the network during the recurrent visit of the attractive set of states $inf(e_s)$. Conversely, an evolution e_s such that $inf(e_s)$ is a spurious attractor will evoke only finitely many motor responses of the network that might necessarily occur before the evolution e_s gets forever trapped by the attractor $inf(e_s)$.

We extend the notions of meaningful and spurious to the stimulations such that a stimulation s is termed *meaningful* if $inf(e_s)$ is a meaningful attractor, and it is termed *spurious* if $inf(e_s)$ is a spurious attractor. In other words, meaningful stimulations are those whose corresponding evolutions get eventually confined into meaningful attractors, and spurious stimulations are those whose corresponding evolutions get eventually confined into spurious attractors.

The set of all meaningful stimulations of $\mathcal{N}$ is called the *neural language* of $\mathcal{N}$ and is denoted by $L(\mathcal{N})$. An arbitrary set of stimulations L is then said to be *recognisable* by some neural network if there exists a network $\mathcal{N}$ such that $L(\mathcal{N}) = L$. These definitions are illustrated in the following example.

Example 0.4. Consider again the network $\mathcal{N}$ described in Example 0.2 (illustrated in Fig. 1). For any finite sequence s, let $s^\omega = ssss\cdots$ denote the infinite sequence obtained by infinitely many concatenations of s. According to this notation, the periodic stimulation $s = \left[\binom{0}{0}\binom{1}{0}\binom{0}{1}\right]^\omega$ induces the corresponding evolution

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

Hence, $inf(e_s) = \{(0, 0, 0)^T, (1, 0, 0)^T, (0, 1, 1)^T\}$, and the evolution e_s of $\mathcal{N}$ remains confined into a cyclic visit of the states of $inf(e_s)$ from time step $t = 1$. Thence, the set $inf(e_s) = \{(0, 0, 0)^T, (1, 0, 0)^T, (0, 1, 1)^T\}$ is an attractor of $\mathcal{N}$. Moreover, since $(0, 1, 1)^T$ is a boolean vector of $inf(e_s)$ describing a spiking configuration of the system where the motor cell x_3 is spiking, the attractor $inf(e_s)$ is thus meaningful. Therefore, the stimulation s is also meaningful, and hence belongs to the neural language of $\mathcal{N}$, *i.e.* $s \in L(\mathcal{N})$. Besides, the periodic stimulation $s' = \left[\binom{1}{1}\binom{0}{0}\right]^\omega$ induces the

corresponding periodic evolution

$$e_{s'} = \left[\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right]^{\omega}.$$

Thence $inf(e_{s'}) = \{(0, 0, 0)^T, (1, 0, 0)^T, (0, 1, 0)^T\}$, and the evolution $e_{s'}$ of $\mathcal{N}$ begins its cyclic visit of the states of $inf(e_{s'})$ already from the first time step $t = 0$. Yet in this case, since the boolean vectors $(0, 0, 0)^T$, $(1, 0, 0)^T$, and $(0, 1, 0)^T$ of $inf(e_{s'})$ describe spiking configurations of the system where the motor cell x_3 remains quiet, the attractor $inf(e_{s'})$ is now spurious. It follows that the stimulation s' is also spurious, and thus $s' \notin L(\mathcal{N})$.

Recurrent Neural Networks and Büchi Automata

In this section, we provide an extension of the classical result stating the equivalence of the computational capabilities of first-order RNN and finite state machines (10). In particular the issue of the expressive power of neural networks is approached here from the point of view of the theory of infinite word reading automata, and it is proved that first-order RNN as defined in Definition 0.1 actually show the very same expressive power as finite deterministic Büchi automata. Towards this purpose, the following definitions need to be recalled.

A *finite deterministic Büchi automaton* is a 5-tuple $\mathcal{A} = (Q, A, i, \delta, \mathcal{F})$, 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 $\mathcal{F}$ is a subset of Q called the set of final states. A finite deterministic Büchi automaton is generally represented by a directed labelled graph whose nodes and labelled edges respectively represent the states and transitions of the automaton, and double-circled nodes represent final states of the automaton.

Given a finite deterministic Büchi automaton $\mathcal{A} = (Q, A, i, \delta, \mathcal{F})$, 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 ρ usually 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 $q_0, q_1, \cdots$. The state q_0 is called the *origin* of ρ, the word $a_1a_2a_3 \cdots$ 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 $inf(\rho)$. In addition, an infinite initial path ρ of $\mathcal{A}$ is called *successful* if it visits infinitely often states that belong to $\mathcal{F}$, *i.e.* if $inf(\rho) \cap \mathcal{F} \neq \emptyset$. An infinite word is then said to be *recognised* by $\mathcal{A}$ if it is the label of a successful infinite path in $\mathcal{A}$, and the *language recognised* by $\mathcal{A}$, denoted by $L(\mathcal{A})$, is the set of all infinite words recognised by $\mathcal{A}$.

Furthermore, 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 which visits precisely all the sates of c. A cycle is called *successful* if it contains a state that belongs to $\mathcal{F}$, and *non-succesful* otherwise. For any $n \in \mathbb{N}$, an *alternating chain* (resp. *co-alternating chain*) of length n is a finite sequence of $n + 1$ distinct cycles $(c_0, \cdots, c_n)$ such that c_0 is successful (resp. c_0 is non-successful), c_i is successful iff c_{i+1} is non-successful, c_{i+1} is accessible from c_i, and c_i is not accessible from c_{i+1}, for all $i < n$. An alternating chain of length ω is a sequence of two cycles (c_0, c_1) such that c_0 is successful, c_1 is non-successful, and both c_0 and c_1 are accessible one from the other. An alternating chain of length α is said to be maximal in $\mathcal{A}$ if there is no alternating chain and no co-alternating chain in $\mathcal{A}$ with a length strictly larger than α. A co-alternating chain of length α is said to be maximal in $\mathcal{A}$ if exactly the same condition holds. These notions of alternating and co-alternaing chains will appear to be directly related to the complexity of the considered networks.

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

Proposition 0.5. *Let $\mathcal{N}$ be a RNN. Then there exists a deterministic Büchi automaton $\mathcal{A}_{\mathcal{N}}$ such that $L(\mathcal{N}) = L(\mathcal{A}_{\mathcal{N}})$.*

Proof. Let $\mathcal{N}$ be given by the tuple (X, S, M, a, b, c), with $card(X) = N$, $card(S) = K$, and $M = \{x_{i_1}, \cdots, x_{i_L}\} \subseteq X$. Now, consider the deterministic Büchi automaton $\mathcal{A}_{\mathcal{N}} = (Q, \Sigma, i, \delta, \mathcal{F})$, where $Q = \{\boldsymbol{x} \in \mathbb{B}^N : \boldsymbol{x}$ is a possible state of $\mathcal{N}\}$, $\Sigma = \mathbb{B}^K$, i is the N-dimensional zero vector, $\delta : Q \times \Sigma \to Q$ is defined by $\delta(\boldsymbol{x}, \boldsymbol{s}) = \boldsymbol{x}'$ iff $\boldsymbol{x}' = \sigma(A \cdot \boldsymbol{x} + B \cdot \boldsymbol{s} + \boldsymbol{c})$, where A, B, and $\boldsymbol{c}$ are the matrices and vectors corresponding to a, b, and c respectively, and where $\mathcal{F} = \{\boldsymbol{x} \in Q :$ the i_k-th component of $\boldsymbol{x}$ is equal to 1 for some $1 \leq k \leq L\}$. In other words, the states of $\mathcal{A}_{\mathcal{N}}$ correspond to all possible states of $\mathcal{N}$, the initial state of $\mathcal{A}_{\mathcal{N}}$ is the initial resting state of $\mathcal{N}$, the final states of $\mathcal{A}_{\mathcal{N}}$ are the states of $\mathcal{N}$ where at least one motor cell is spiking, the underlying alphabet of $\mathcal{A}_{\mathcal{N}}$ is the set of all possible stimuli of $\mathcal{N}$, and $\mathcal{A}_{\mathcal{N}}$ contains a transition from $\boldsymbol{x}$ to $\boldsymbol{x}'$ labelled by $\boldsymbol{s}$ iff the dynamical equations of $\mathcal{N}$ ensure that $\mathcal{N}$ transits from state $\boldsymbol{x}$ to state $\boldsymbol{x}'$ when it receives the stimulus $\boldsymbol{s}$. According to this construction, any evolution e_s of $\mathcal{N}$ naturally induces a corresponding infinite initial path $\rho(e_s)$ in $\mathcal{A}_{\mathcal{N}}$ that visits a final state infinitely often iff e_s evokes infinitely many motor responses. Consequently, any stimulation s of $\mathcal{N}$ is meaningful for $\mathcal{N}$ iff s is recognised by $\mathcal{A}_{\mathcal{N}}$. In other words, $s \in$

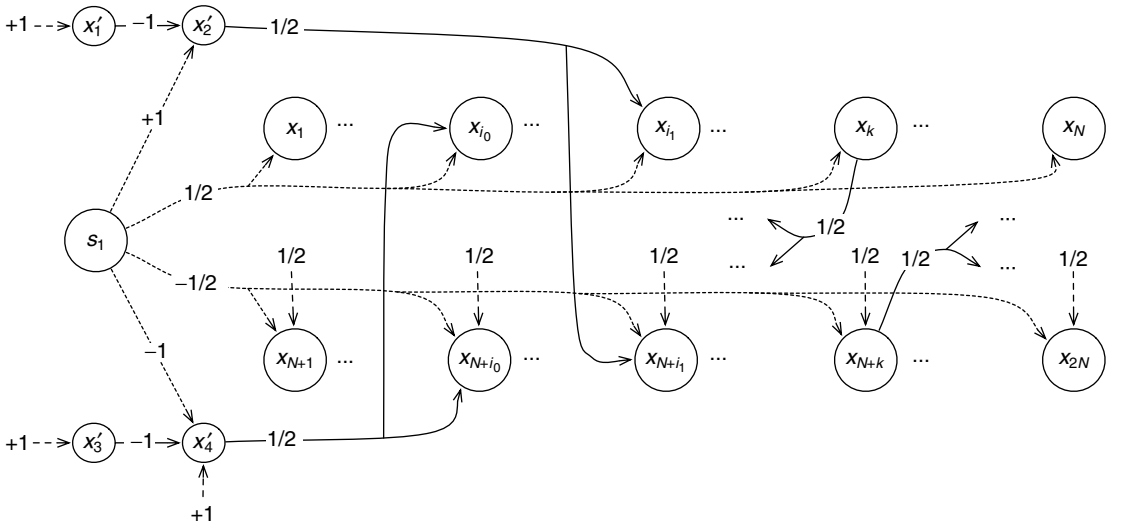

Fig. 2. Construction of the network $\mathcal{N}_{\mathcal{A}}$ recognising the same language as a deterministic Büchi automaton $\mathcal{A}$.

$L(\mathcal{N})$ iff $s \in L(\mathcal{A}_{\mathcal{N}})$, and therefore $L(\mathcal{N}) = L(\mathcal{A}_{\mathcal{N}})$. □

According to the construction given in the proof of Proposition 0.5, any evolution e_s of network $\mathcal{N}$ naturally induces a corresponding infinite initial path $\rho(e_s)$ in the deterministic Büchi automaton $\mathcal{A}_{\mathcal{N}}$. Conversely, any infinite initial path ρ in $\mathcal{A}_{\mathcal{N}}$ can be associated to some evolution $e_s(\rho)$ of $\mathcal{N}$. Hence, given some set of states A of $\mathcal{N}$, there exists a stimulation s of $\mathcal{N}$ such that $inf(e_s) = A$ iff there exists an infinite initial path ρ in $\mathcal{A}_{\mathcal{N}}$ such that $inf(\rho) = A$, or equivalently, iff A is a cycle in $\mathcal{A}_{\mathcal{N}}$. Notably, 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 Büchi automaton $\mathcal{A}_{\mathcal{N}}$. Consequently, a procedure to compute all possible attractors of a given network $\mathcal{N}$ is simply obtained by constructing at first the corresponding deterministic Büchi automaton $\mathcal{A}_{\mathcal{N}}$ and then listing all cycles in the graph of $\mathcal{A}_{\mathcal{N}}$.

We can prove now that any deterministic Büchi automaton can be simulated by some first-order RNN. For the sake of convenience, we choose to restrict our attention to deterministic Büchi automata over the binary alphabet $\mathbb{B}^1 = \{(0), (1)\}$. Such a restriction does not weaken the forthcoming results, for the expressive power of deterministic Büchi automata is already completely achieved by deterministic Büchi automata over binary alphabets.

Proposition 0.6. *Let $\mathcal{A}$ be some deterministic Büchi automaton over the alphabet $\mathbb{B}^1$. Then there exists a RNN $\mathcal{N}_{\mathcal{A}}$ such that $L(\mathcal{A}) = L(\mathcal{N}_{\mathcal{A}})$.*

Proof. Let $\mathcal{A}$ be given by the tuple $(Q, A, q_1, \delta, \mathcal{F})$, with $Q = \{q_1, \cdots, q_N\}$ and $\mathcal{F} = \{q_{i_1}, \cdots, q_{i_k}\} \subseteq Q$. Now, consider the network $\mathcal{N}_{\mathcal{A}} = (X, S, M, a, b, c)$ defined by $X = X_{main} \cup X_{aux}$, where $X_{main} = \{x_i : 1 \le i \le 2N\}$ and $X_{aux} = \{x'_1, x'_2, x'_3, x'_4\}$, $S = \{s_1\}$, $M = \{x_{i_j} : 1 \le j \le K\} \cup \{x_{N+i_j} : 1 \le j \le K\}$, and the functions a, b, and c are defined as follows. First of all, both cells x'_1 and x'_3 receive a background activity of intensity 1, and receive no other afferent connections. The cell x'_2 receives two afferent connections of intensities –1 and 1 from cells x'_1 and s_1, and the cell x'_4 receives two afferent connections of same intensity –1 from cells x'_3 and s_1 as well as a background activity of intensity 1. Moreover, each state q_i in the automaton $\mathcal{A}$ gives rise to a corresponding cell layer in the network $\mathcal{N}_{\mathcal{A}}$ consisting of the two cells x_i and x_{N+i}. For each $1 \le i \le N$, the cell x_i receives a weighted connection of intensity $\frac{1}{2}$ from the input s_1, and the cell x_{N+1} receives a weighted connection of intensity $-\frac{1}{2}$ from the input s_1, as well as a background activity of intensity $\frac{1}{2}$. Furthermore, let i_0 and i_1 denote the indices such that $\delta(q_1, (0)) = q_{i_0}$ and $\delta(q_1, (1)) = q_{i_1}$, respectively, then both cells x_{i_0} and x_{N+i_0} receive a connection of intensity $\frac{1}{2}$ from cell x'_4, and both cells x_{i_1} and x_{N+i_1} receive a connection of intensity $\frac{1}{2}$ from cell x'_2, as illustrated in Fig. 2. Moreover, for each $1 \le i, j \le N$, there exist two weighted connections of intensity $\frac{1}{2}$ from cell x_i to both cells x_j and x_{N+j} if $\delta(q_1, (1)) = q_j$, and there exist two weighted connections of intensity $\frac{1}{2}$ from cell x_{N+i} to both cells x_j and x_{N+j} iff $\delta(q_1, (0)) = q_j$, as partially illustrated in Fig. 2 only for the k-th layer. Finally, the definition of the set of motor cells M ensures that, for each $1 \le i \le N$, the two cells of the layer $\{x_i, x_{N+i}\}$ are motor cells of $\mathcal{N}_{\mathcal{A}}$ iff q_i is a final state of $\mathcal{A}$. The network $\mathcal{N}_{\mathcal{A}}$ obtained from $\mathcal{A}$ by means of the aforementioned construction is illustrated in Fig. 2, where connections between activation cells are partially represented by full lines, efferent con-

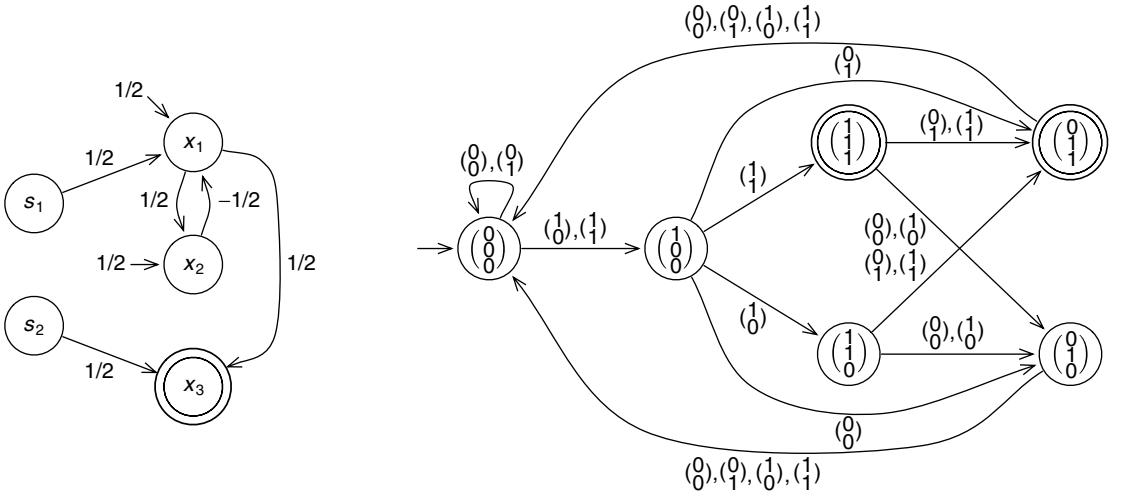

Fig. 3. The translation from some given network $\mathcal{N}$ to its corresponding deterministic Büchi automaton $\mathcal{A}_{\mathcal{N}}$.

nections from the sensory cell s_1 are represented by dotted lines, and background activity connections are represented by dashed lines. According to the this construction of the network $\mathcal{N}_{\mathcal{A}}$, one and only one cell of X_{main} will fire at every time step $t \geq 2$, and a cell in X_{main} will fire at time $t+1$ iff it receives simultaneously at time t an activity of intensity $\frac{1}{2}$ from the sensory cell s_1 as well as an activity of intensity $\frac{1}{2}$ from a cell in X_{main}. More precisely, any infinite sequence $s = \boldsymbol{s(0)s(1)s(2)} \cdots \in [\mathbb{B}^1]^\omega$ induces both a corresponding infinite path $\rho_s : q_1 \xrightarrow{\boldsymbol{s(0)}} q_{j_1} \xrightarrow{\boldsymbol{s(1)}} q_{j_2} \xrightarrow{\boldsymbol{s(2)}} q_{j_3} \cdots$ in $\mathcal{A}$ as well as a stimulation $e_s = \boldsymbol{x(0)x(1)x(2)} \cdots$ in $\mathcal{N}_{\mathcal{A}}$. The network $\mathcal{N}_{\mathcal{A}}$ then satisfies precisely the following property: for every time step $t \geq 2$, if $\boldsymbol{s(t-1)} = (1)$, then the state $\boldsymbol{x(t)}$ corresponds to a spiking configuration where only the cells x'_1, x'_3, and $x_{j_{t1}}$ are spiking, and if $\boldsymbol{s(t-1)} = (0)$, then the state $\boldsymbol{x(t)}$ corresponds to a spiking configuration where only the cells x'_1, x'_3, and $x_{N+j_{t-1}}$ are spiking. In other words, the infinite path ρ_s and the stimulation e_s evolve in parallel and satisfy the property that the cell x_j is spiking in $\mathcal{N}_{\mathcal{A}}$ iff the automaton $\mathcal{A}$ is in state q_j and reads letter (1), and the cell x_{N+j} is spiking in $\mathcal{N}_{\mathcal{A}}$ iff the automaton $\mathcal{A}$ is in state q_j and reads letter (0). Hence, for any infinite infinite sequence $s \in [\mathbb{B}^1]^\omega$, the infinite path ρ_s in $\mathcal{A}$ visits infinitely many final states iff the evolution e_s in $\mathcal{N}_{\mathcal{A}}$ evoked infinitely many motor responses. This means that s is recognised by $\mathcal{A}$ iff s is meaningful for $\mathcal{N}_{\mathcal{A}}$. Therefore, $L(\mathcal{A}) = L(\mathcal{N}_{\mathcal{A}})$.

Actually, it can be proved that the translation between deterministic Büchi automata and RNN described in Proposition 0.6 can be generalised to any alphabet $\mathbb{B}^K$ with $K > 0$. Hence, Proposition 0.5 together with a suitable generalisation of Proposition 0.6 to all alphabets of multidimensional boolean vectors permit to deduce the following equivalence between first-order RNN and deterministic Büchi automata.

Theorem 0.7. *Let $K > 0$ and let $L \subseteq [\mathbb{B}^K]^\omega$. Then L is recognisable by some first-order RNN iff L is recognisable by some deterministic Büchi automaton.*

Finally, the following example provides an illustration of the two procedures given in the proofs of Propositions 0.5 and 0.6 describing the translations, on the one hand, from a given RNN to a corresponding deterministic Büchi automaton, and on the other hand, from a given deterministic Büchi automaton to a corresponding RNN.

Example 0.8. The translation from the network $\mathcal{N}$ described in Example 0.2 to its corresponding deterministic Büchi automaton $\mathcal{A}_{\mathcal{N}}$ is illustrated in Fig. 3. Proposition 0.5 ensures that $L(\mathcal{N}) = L(\mathcal{N}_{\mathcal{A}})$. Conversely, the translation from some given deterministic Büchi automaton $\mathcal{A}$ over the alphabet $\mathbb{B}^1$ to its corresponding network $\mathcal{N}_{\mathcal{A}}$ is illustrated in Fig. 4. Proposition 0.6 ensures that $L(\mathcal{A}) = L(\mathcal{N}_{\mathcal{A}})$. In both cases, motor cells of networks as well as final states of Büchi automata are double-circled.

The RNN Hierarchy

In theoretical computer science, infinite word reading machines are often classified according the topological complexity of the languages that they recognise, as for instance in (2-4, 12, 22). Such classifications provide an interesting complexity measure of the expressive power of different kinds of infinite word reading machines. Here, this approach is translated from the ω-automata to the neural network context, and a hierarchical classification of first-order

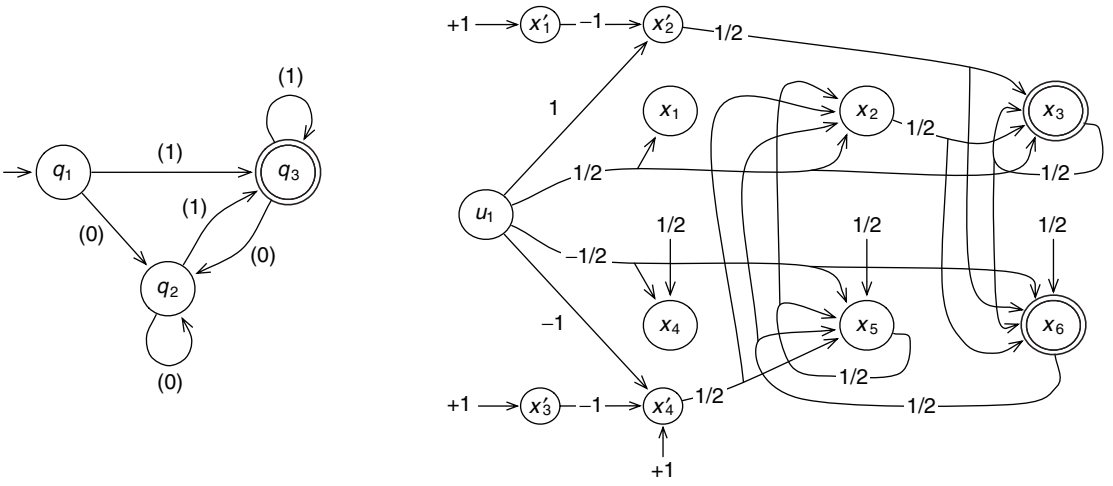

Fig. 4. Translation from some given deterministic Büchi automaton $\mathcal{A}$ to its corresponding network $\mathcal{N}_{\mathcal{A}}$.

RNN is obtained. Notably, this classification will be tightly related to the attractive properties of the networks.

More precisely, along the sequential presentation of a stimulation s, the induced evolution e_s of a network might seem to successively fall into several distinct attractors before getting eventually trapped by the attractor $inf(e_s)$. In other words, the sequence of successive states e_s might visit the same set of states for a while, but then escapes from this pattern and visits another set of states for some while again, and so forth until it finally gets attracted for ever by the set of states $inf(e_s)$. We specially focus on this feature and provide a refined hierarchical classification of first-order RNN according to their capacity to punctually switch between attractors of different types along their evolutions.

For this purpose, the following facts and definitions need to be introduced. To begin with, for any $k > 0$, the space of all infinite sequences of k-dimensional boolean vectors $[\mathbb{B}^k]^\omega$ 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 iff the inverse image by f of every open set of $[\mathbb{B}^l]^\omega$ is an open set of $[\mathbb{B}^k]^\omega$ according to the aforementioned topologies over $[\mathbb{B}^l]^\omega$ and $[\mathbb{B}^l]^\omega$.

Now, given two RNN $\mathcal{N}_1$ and $\mathcal{N}_2$ with K_1 and K_2 sensory cells respectively, we say that $\mathcal{N}_1$ c*ontinuously reduces* (or *Wadge reduces*, or simply *reduces*) to $\mathcal{N}_2$, denoted by $\mathcal{N}_1 \leq_W \mathcal{N}_2$, iff there exists a continuous function $f : [\mathbb{B}^{K_1}]^\omega \to [\mathbb{B}^{K_2}]^\omega$ such that any stimulation s of $\mathcal{N}_1$ satisfies $s \in L(\mathcal{N}_1) \Leftrightarrow f(s) \in L(\mathcal{N}_2)$ (21).

Intuitively, $\mathcal{N}_1 \leq_W \mathcal{N}_2$ iff the problem of determining whether some stimulation s is meaningful for $\mathcal{N}_1$ reduces *via* some simple function f to the problem of knowing whether $f(s)$ is meaningful for $\mathcal{N}_2$. Then, the corresponding strict reduction is defined by $\mathcal{N}_1 <_W \mathcal{N}_2$ iff $\mathcal{N}_1 \leq_W \mathcal{N}_2 \not\leq_W \mathcal{N}_1$, the equivalence relation is defined by $\mathcal{N}_1 \equiv_W \mathcal{N}_2$ iff $\mathcal{N}_1 \leq_W \mathcal{N}_2 \leq_W \mathcal{N}_1$, and the incomparability relation is defined by $\mathcal{N}_1 \perp_W \mathcal{N}_2$ iff $\mathcal{N}_1 \not\leq_W \mathcal{N}_1 \not\leq_W \mathcal{N}_1$. Equivalence classes of networks according to Wadge reduction are denoted $\equiv_W$-equivalence classes. The continuous reduction over neural networks then naturally induces a hierarchical classification of neural networks formally defined as follows:

Definition 0.9. *The collection of all first-order RNN as defined in Definition 0.1 ordered by the reduction relation "$\leq_W$" will be called the RNN hierarchy.*

We can now provide a complete description of the RNN hierarchy. Firstly, it can be proved that the RNN hierarchy is well founded.[2] Moreover, it can also be shown that the maximal chains[3] in the RNN hierarchy have length ω+1, which is to say that the RNN hierarchy has a height of ω+1. Furthermore, the maximal antichains[4] of the RNN hierarchy have length 2, meaning that the RNN hierarchy has a width of 2. More precisely, the RNN hierarchy actually consists of ω alternating successions of pairs of incomparable $\equiv_W$-equivalence classes and single $\equiv_W$-equivalence classes, overhung by a ultimate single $\equiv_W$-equivalence class, as illustrated in Fig. 5, where circle represent $\equiv_W$-

[2]The fact that the RNN hierarchy is well founded means that every non-empty set of neural networks has a $\leq_W$-minimal element.
[3]A chain 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 $i < j$. A maximal chain is a chain whose length is at least as large as every other chain.
[4]An antichain of the RNN hierarchy is a sequence of pairwise incomparable neural networks. A maximal antichain is an antichain whose length is at least as large as every other antichain.

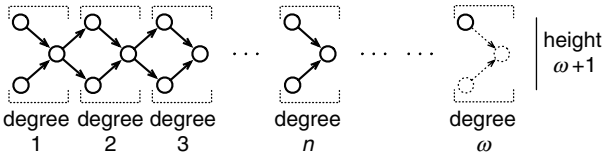

Fig. 5. The RNN hierarchy: an alternating succession of pairs of incomparable classes and single classes of networks overhung by a ultimate single class.

equivalence classes of networks and arrows between circles represent the strict reduction "$<_W$" between all elements of the corresponding classes. The pairs of incomparable $\equiv_W$-equivalence classes are called the non-self-dual levels of the RNN hierarchy and the single $\equiv_W$-equivalence classes are called the self-dual levels of the RNN hierarchy. Then, the *degree* of a RNN $\mathcal{N}$, denoted by $d(\mathcal{N})$, is defined as being equal to n if $\mathcal{N}$ belongs either to the n-th non-self-dual level or to the n-th self-dual level of the RNN hierarchy, for all $n > 0$, and the degree of $\mathcal{N}$ is equal to ω if it belongs to the ultimate overhanging $\equiv_W$-equivalence class. Besides, it can also be proved that the RNN hierarchy is actually decidable, in the sense that there exists an algorithmic procedure computing the degree of any network in the RNN hierarchy. All the aforementioned properties of the RNN hierarchy are now summarised in the following result.

Theorem 0.10. *The RNN hierarchy is a decidable pre-well ordering of width 2 and height $\omega + 1$.*

Proof. The collection of all deterministic Büchi automata ordered by the reduction relation "$\leq_W$", called the DBA hierarchy, can be proved to be decidable pre-well ordering of width 2 and height ω+1 (1, 11). Propositions 0.5 and 0.6 as well as Theorem 0.7 ensure that the RNN hierarchy and DBA hierarchy are isomorphic, which concludes the proof. □

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 maximal number of times that this network might switch between punctual evocations of meaningful and spurious attractors along some evolution.

Theorem 0.11. *Let n be some strictly positive integer, $\mathcal{N}$ be a network, and $\mathcal{A}_\mathcal{N}$ be the corresponding deterministic Büchi automaton of $\mathcal{N}$.*

If there exists in $\mathcal{A}_\mathcal{N}$ a maximal alternating chain of length n and no maximal co-alternating chain of length n, then $d(\mathcal{N}) = n$ and $\mathcal{N}$ is non-self-dual.

If there exists in $\mathcal{A}_\mathcal{N}$ a maximal co-alternating chain of length n but no maximal alternating chain of length n, then also $d(\mathcal{N}) = n$ and $\mathcal{N}$ is non-self-dual.

If there exist in $\mathcal{A}_\mathcal{N}$ a maximal alternating chain of length n as well as a maximal co-alternating chain of length n, then $d(\mathcal{N}) = n$ and $\mathcal{N}$ is self-dual.

If there exist in $\mathcal{A}_\mathcal{N}$ a maximal alternating chain of length ω, then $d(\mathcal{N}) = \omega$.

Proof. It can be shown that the translation procedure described in Proposition 0.5 is actually an isomorphism from the RNN hierarchy to the DBA hierarchy. Therefore, the degree of a network $\mathcal{N}$ in the RNN hierarchy is equal to the degree of its corresponding deterministic Büchi automaton $\mathcal{A}_\mathcal{N}$ in the DBA hierarchy. Moreover, the degree of a deterministic Büchi automaton in the DBA hierarchy corresponds precisely to the length of a maximal alternating or co-alternating chain of contained this automaton (22, 11). □

By Theorem 0.11, 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 deterministic Büchi automaton $\mathcal{A}_\mathcal{N}$, as described in Proposition 0.5, and then returning the ordinal $\alpha < \omega + 1$ corresponding to the length of the maximal alternating chains or co-alternating chains contained in $\mathcal{A}_\mathcal{N}$. Note that this procedure can clearly beachieved by some graph analysis of the automaton $\mathcal{A}_\mathcal{N}$, since the graph of $\mathcal{A}_\mathcal{N}$ is always finite. Furthermore, since alternating and co-alternating chains are defined in terms of cycles in the graph of the automaton, and according to the biunivocal correspondence between cycles in $\mathcal{A}_\mathcal{N}$ and attractors of $\mathcal{N}$, it can be deduced that the complexity of a network in the RNN hierarchy is indeed tightly related to the attractive properties of this network.

More precisely, it can be observed that the measure of complexity provided by the RNN hierarchy actually corresponds precisely to the maximal number of times that a network might alternate between punctual evocations of meaningful and spurious attractors along some evolution. Indeed, the existence of a maximal alternating or co-alternating chain $(c_0, \cdots, c_n)$ of length n in $\mathcal{A}_\mathcal{N}$ means that every infinite initial path in $\mathcal{A}_\mathcal{N}$ might alternate at most n times between punctual visits of successful and non-successful cycles. Yet, according to the biunivocal correspondence between cycles in $\mathcal{A}_\mathcal{N}$ and attractors of $\mathcal{N}$, this is precisely equivalent to saying that every evolution of $\mathcal{N}$ can only alternate at most n times between punctual evocations of meaningful and spurious attractors before getting eventually forever trapped by a last attractor. In this case, Theorem 0.11 ensures that the degree of $\mathcal{N}$ is equal to n. 2Moreover,

the existence of an alternating chain (c_1, c_2) of length ω in $\mathcal{A}_\mathcal{N}$ is equivalent to the existence of an infinite initial path in $\mathcal{A}_\mathcal{N}$ that might alternate infinitely many times between punctual visits of the cycles c_1 and c_2. Yet, this is equivalent to saying that there exists an evolution of $\mathcal{N}$ that might alternate ω times between punctual visits of a meaningful and a spurious attractor. By Theorem 0.11, the degree of $\mathcal{N}$ is equal to ω is this case. Therefore, RNN hierarchy provides a new measure complexity of neural networks according to their maximal capability to alternate between punctual evocations of different types of attractors along their evolutions. Moreover, it is worth noting that the concept of alternation between different types of attractors mentioned in our context tightly resembles the relevant notion of chaotic itinerancy widely studied by Tsuda *et al*. (5, 19, 20). Finally, the following example illustrates the decidability procedure of the RNN hierarchy.

Example 0.12. Let $\mathcal{N}$ be the network described in Example 0.2. The corresponding deterministic Büchi automaton $\mathcal{A}_\mathcal{N}$ of $\mathcal{N}$ represented in Fig. 3 contains the successful cycle $c_1 = \{(0, 0, 0)^T, (1, 0, 0)^T, (0, 1, 1)^T\}$, the non-successful cycle $c_2 = \{(0, 0, 0)^T, (1, 0, 0)^T, (0, 1, 0)^T\}$, and both c_1 and c_2 are accessible one from the other. Hence, (c_1, c_2) is an alternating chain of length ω in $\mathcal{A}_\mathcal{N}$, and Theorem 0.11 ensures that the degree of $\mathcal{N}$ in the RNN hierarchy is equal to ω.

Discussion

We provided a hierarchical classification of first-order RNN based on the capability of the networks to punctually switch between attractors of different types along their evolutions. This hierarchy is proved to be a decidable pre-well ordering of width 2 and height of $\omega + 1$. A decidability procedure computing the degree of a network in this hierarchy is finally described. Therefore, the hierarchical classification that we obtained provides a new measure of complexity of first-order RNN according to their attractive properties.

Note that a comparable classification of sigmoidal-threshhold activation function instead of hard-threshhold neuronal model could also be obtained. Indeed, as already mentioned in the introduction of this work, the consideration of saturated-linear sigmoidal instead of hard-threshold activation functions drastically increases the computational capabilities of the respective networks from finite state automata up to Turing capabilities (15, 17). Therefore, a similar hierarchical classification of RNN provided with linear sigmoidal activation functions might be achieved by translating the Wadge classification theory from the Turing machine to the neural network context (12). In this case, the obtained hierarchical classification would consist of a very refined transfinite pre-well ordering of width 2 and height $(\omega_1^{CK})^\omega$, where ω_1^{CK} is the first non-recursice ordinal known as the Church-Kleene ordinal. Unfortunately, the decidability procedure of this hierarchy is still missing and remains a hard open problem in theoretical computer science. As long as such a decidability procedure will not be understood, the precise relationship between the obtained hierarchical classification and the internal and attractive properties of the networks will also necessarily remain unclear, thus reducing the sphere of significance of the corresponding classification of neural networks.

The present work can be extended in at least three directions. Firstly, it is envisioned to study similar Wadge-like hierarchical classifications applied to more biologically oriented neuronal models. For instance, Wadge-like classifications of RNN provided with some simple spike-timing dependent plasticity rule could be of interest. Also, Wadge-like classifications of neural networks characterized by complex activation function or dynamical governing equations could be relevant. However, it is worth mentioning once again that, as soon as the computational capabilities of the considered neuronal model shall reach the expressive power of infinite words deterministic Turing machines, the complexity measure induced by a corresponding Wadge-like classification of these networks becomes significantly misunderstood.

Secondly, it is expected to describe hierarchical classifications of neural networks induced by more biologically plausible reduction relations than the continuous (or Wadge) reduction. Indeed, the hierarchical classification described in this paper provides a classification of networks according to the topological complexity of the underlying neural language, but it still remains unclear how this natural mathematical criteria is related to the real biological complexity of the networks.

Thirdly, from a biological perspective, the understanding of the complexity of neural networks should rather be approached from the point of view of finite words reading machines instead of infinite words reading machines, as for instance in (8, 13-18). Unfortunately, as opposed to the case of infinite words reading machines, the classification theory of finite words reading machines is still a widely undeveloped, yet promising, issue.

Acknowledgments

The authors ackowledge the support by the European Union FP6 grant #043309 (GABA). J. Cabessa would like to thank Cinthia Camposo for her valuable support during this work.

References

1. Duparc, J. Wadge hierarchy and Veblen hierarchy part i: Borel sets of finite rank. *J. Symb. Log*. 66: 56-86, 2001.
2. Duparc, J. A hierarchy of deterministic context-free ω-languages. *Theor. Comput. Sci*. 290: 1253-1300, 2003.
3. Duparc, J., Finkel, O. and Ressayre, J.-P. Computer science and the fine structure of Borel sets. *Theor. Comput. Sci*. 257: 85-105, 2001.
4. Finkel, O. An effective extension of the wagner hierarchy to blind counter automata. *Lect. Notes Comput. Sci.* 2142: 369-383, 2001.
5. Kaneko, K. and Tsuda, I. Chaotic itinerancy. *Chaos*, 13: 926-936, 2003.
6. Kilian, J. and Siegelmann, H.T. The dynamic universality of sigmoidal neural networks. *Inf. Comput*. 128: 48-56, 1996.
7. Kleene, S.C. Representation of events in nerve nets and finite automata. In: *Automata Studies*, volume 34 of *Annals of Mathematics Studies*, pages 3-42. Princeton University Press, Princeton, N. J., 1956.
8. Kremer, S.C. On the computational power of elman-style recurrent networks. *Neural Networks, IEEE Transactions on*, 6: 1000-1004, 1995.
9. McCulloch, W.S. and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* 5: 115-133, 1943.
10. Minsky, M.L. *Computation: finite and infinite machines.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.
11. Perrin, D. and Pin, J.-E. Infinite Words, volume 141 of *Pure and Applied Mathematics*. Elsevier, 2004. ISBN 0-12-532111-2.
12. Selivanov, V. Wadge degrees of ω-languages of deterministic Turing machines. *Theor. Inform. Appl*. 37: 67-83, 2003.
13. Siegelmann, H.T. Computation beyond the Turing limit. *Science*, 268: 545-548, 1995.
14. Siegelmann, H.T. Neural and super-Turing computing. *Minds Mach*. 13: 103-114, 2003.
15. Siegelmann, H.T. and Sontag, E.D. Turing computability with neural nets. *Appl. Math. Lett.* 4: 77-80, 1991.
16. Siegelmann, H.T. and Sontag, E.D. Analog computation *via* neural networks. *Theor. Comput. Sci*. 131: 331-360, 1994.
17. Siegelmann, H.T. and Sontag, E.D. On the computational power of neural nets. *J. Comput. Syst. Sci*. 50: 132-150, 1995.
18. Sperduti, A. On the computational power of recurrent neural networks for structures. *Neural Netw*. 10: 395-400, 1997.
19. Tsuda, I. Chaotic itinerancy as a dynamical basis of hermeneutics of brain and mind. *World Futures*, 32: 167-185, 1991.
20. Tsuda, I., Koerner, E. and Shimizu, H. Memory dynamics in asynchronous neural networks. *Prog. Th. Phys*. 78: 51-71, 1987.
21. Wadge, W.W. *Reducibility and determinateness on the Baire space*. PhD thesis, University of California, Berkeley, 1983.
22. Wagner, K. On ω-regular sets. *Inform. Control* 43: 123-177, 1979.