An Attractor-Based Complexity Measurement for Boolean Recurrent Neural Networks

Jérémie Cabessa[1,2]*, **Alessandro E. P. Villa**[1,3]*

1 Neuroheuristic Research Group, Faculty of Business and Economics, University of Lausanne, Lausanne, Switzerland, 2 Laboratory of Mathematical Economics (LEMMA), University of Paris 2 – Panthéon-Assas, Paris, France, 3 Grenoble Institute of Neuroscience, Faculty of Medicine, University Joseph Fourier, Grenoble, France

Abstract

We provide a novel refined attractor-based complexity measurement for Boolean recurrent neural networks that represents an assessment of their computational power in terms of the significance of their attractor dynamics. This complexity measurement is achieved by first proving a computational equivalence between Boolean recurrent neural networks and some specific class of ω-automata, and then translating the most refined classification of ω-automata to the Boolean neural network context. As a result, a hierarchical classification of Boolean neural networks based on their attractive dynamics is obtained, thus providing a novel refined attractor-based complexity measurement for Boolean recurrent neural networks. These results provide new theoretical insights to the computational and dynamical capabilities of neural networks according to their attractive potentialities. An application of our findings is illustrated by the analysis of the dynamics of a simplified model of the basal ganglia-thalamocortical network simulated by a Boolean recurrent neural network. This example shows the significance of measuring network complexity, and how our results bear new founding elements for the understanding of the complexity of real brain circuits.

Citation: Cabessa J, Villa AEP (2014) An Attractor-Based Complexity Measurement for Boolean Recurrent Neural Networks. PLoS ONE 9(4): e94204. doi:10.1371/journal.pone.0094204

Editor: Sergio Gómez, Universitat Rovira i Virgili, Spain

Received May 3, 2013; **Accepted** March 14, 2014; **Published** April 11, 2014

Funding: This work is funded by the University of Lausanne, Switzerland. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: jcabessa@nhrg.org (JC); avilla@nhrg.org (AV)

Introduction

In neural computation, understanding the computational and dynamical properties of biological neural networks is an issue of central importance. In this context, much interest has been focused on comparing the computational power of diverse theoretical neural models with those of abstract computing devices. Nowadays, the computational capabilities of neural models is known to be tightly related to the nature of the activation function of the neurons, to the nature of their synaptic connections, to the eventual presence of noise in the model, to the possibility for the networks to evolve over time, and to the computational paradigm performed by the networks.

The first and seminal results in this direction were provided by McCulloch and Pitts, Kleene, and Minsky who proved that first-order Boolean recurrent neural networks were computationally equivalent to classical finite state automata [1–3]. Kremer extended these results to the class of Elman-style recurrent neural nets [4], and Sperduti discussed the computational power of different other architecturally constrained classes of networks [5].

Later, Siegelmann and Sontag proved that by considering rational synaptic weights and by extending the activation functions of the cells from Boolean to linear-sigmoid, the corresponding neural networks have their computational power drastically increased from finite state automata up to Turing machines [6–8]. Kilian and Siegelmann then generalised the Turing universality of neural networks to a broader class of sigmoidal activation functions [9]. The computational equivalence between so-called "rational recurrent neural networks" and Turing machines has now become standard result in the field.

Following von Neumann considerations [10], Siegelmann and Sontag further assumed that the variables appearing in the underlying chemical and physical phenomena could be modelled by continuous rather than discrete (rational) numbers, and therefore proposed a study of the computational capabilities of recurrent neural networks equipped with real instead of rational synaptic weights [11]. They proved that the so-called "analog recurrent neural networks" are computationally equivalent to Turing machines with advices, hence capable of super-Turing computational power from polynomial time of computation already [11]. In this context, a proper internal hierarchical classification of analog recurrent neural networks according to the Kolmogorov complexity of their underlying real synaptic weights was described [12].

It was also shown that the presence of arbitrarily small amount of analog noise seriously reduces the computational capability of both rational- and real-weighted recurrent neural networks to those of finite automata [13]. In the presence of Gaussian or other common analog noise distribution with sufficiently large support, the computational power of recurrent neural networks is reduced to even less than finite automata, namely to the recognition of definite languages [14].

Besides, the concept of evolvability has also turned out to be essential in the study of the computational power of circuits closer to the biological world. The research in this context has initially been focused almost exclusively on the application of genetic algorithms aimed at allowing networks with fully-connected topology and satisfying selected fitness functions (e.g., performed well on specific tasks) to reproduce and multiply [15–18]. This approach aimed to optimise the connection weights that determine the functionality of a network with fixed-topology. However, the topology of neural networks, i.e. their structure and connectivity patterns, greatly affects their functionality. The evolution of both topologies and connection weights following bioinspired rules that may also include features derived from the study of neural development, differentiation, genetically programmed cell-death and synaptic plasticity rules has become increasingly studied in recent years [19–26]. Along this line, Cabessa and Siegelmann provided a theoretical study proving that both models of rational-weighted and analog evolving recurrent neural networks are capable of super-Turing computational capabilities, equivalent to those of static analog neural networks [27].

Finally, from a general perspective, the classical computational approach from Turing [28] was argued to "no longer fully corresponds to the current notion of computing in modern systems" [29] – especially when it refers to bio-inspired complex information processing systems. In the brain (or in organic life in general), information is rather processed in an interactive way [30], where previous experience must affect the perception of future inputs, and where older memories may themselves change with response to new inputs. Following this perspective, Cabessa and Villa described the super-Turing computational power of analog recurrent neural networks involved in a reactive computational framework [31]. Cabessa and Siegelmann provided a characterisation of the Turing and super-Turing capabilities of rational and analog recurrent neural networks involved in a basic interactive computational paradigm, respectively [32]. Moreover, Cabessa and Villa proved that neural models combining the two crucial features of evolvability and interactivity were capable of super-Turing computational capabilities [33].

In this paper, we pursue the study of the computational power of neural models and provide two novel refined attractor-based complexity measurement for Boolean recurrent neural networks. More precisely, as a first step we provide a generalisation to the precise infinite input stream context of the classical equivalence result between Boolean neural networks and finite state automata [1–3]. Under some natural condition on the type specification of their attractors, we show that Boolean recurrent neural networks disclose the very same expressive power as deterministic Büchi automata [34]. This equivalence allows to establish a hierarchical classification of Boolean neural networks by translating the Wagner classification theory from the Büchi automaton to the neural network context [35]. The obtained classification consists of a pre-well ordering of width 2 and height $\omega+1$ (where ω denotes the first infinite ordinal). As a second step, we show that by totally relaxing the restrictions on the type specification of their attractors, the Boolean neural networks significantly increase their expressive power from deterministic Büchi automata up to Muller automata. Hence, another more refined hierarchical classification of Boolean neural networks is obtained by translating the Wagner classification theory from the Muller automaton to the neural network context. This classification consists of a pre-well ordering of width 2 and height ω^ω. The complexity measurements induced by these two hierarchical classifications refer to the possibility of networks' dynamics to maximally alternate between attractors of different types along their evolutions. They represent an assessment of the computational power of Boolean neural networks in terms of the significance of their attractor dynamics. Finally, an application of this approach to a Boolean model of the basal ganglia-thalamocortical network is provided. This practical example shows that our automata-theoretical approach might bear new founding elements for the understanding of the complexity of real brain circuits.

Materials and Methods

Network Model

In this work, we focus on synchronous discrete-time first-order recurrent neural networks made up of classical McCulloch and Pitts cells. Such a neural network is modelled by a general labelled directed graph. The nodes and labelled edges of the graph respectively represent the cells and synaptic connections of the network. At each time step, the status of each activation cell can be of only two kinds: firing or quiet. When firing, a cell instantaneously transmits an action potential throughout all its outgoing connections, the intensity of which being equal to the label of the underlying connection. Then, a given cell is firing at time $t+1$ whenever the summed intensity of all the incoming action potentials transmitted at time t by both its afferent cells and background activity exceeds its threshold (which we suppose without loss of generality to be equal to 1). The definition of such a network can be formalised as follows:

Definition 1. *A first-order Boolean recurrent neural network (RNN)* 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 input units, 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 weighted synaptic connections between cells, the weighted connections from the input units to the activation cells, and the background activity, respectively.

The activation value of cells x_j and input units 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 equal 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

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

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

According to Equation (1), the dynamics of the whole network $\mathcal{N}$ is described by the following governing equation

$$\vec{x}(t+1)=\sigma(a\cdot\vec{x}(t)+b\cdot\vec{u}(t)+c), \qquad (2)$$

where $\vec{x}(t)=(x_1(t),\ldots,x_N(t))$ and $\vec{u}(t)=(u_1(t),\ldots,u_M(t))$ are Boolean vectors describing the spiking configuration of the activation cells and input units, and σ denotes the Heaviside step function applied component by component.

Such Boolean neural networks have already been proven to reveal same computational capabilities as finite state automata [1–3]. Furthermore, it can be observed that rational- and real-weighted Boolean neural networks are actually computationally equivalent.

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

$$\begin{pmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \end{pmatrix} =$$

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

Attractors

Neurophysiological Meaningfulness. In bio-inspired complex systems, the concept of an *attractor* has been shown to carry strong biological and computational implications. According to Kauffman: "Because many complex systems harbour attractors to which the system settle down, the attractors literally are most of what the systems do" [36, p. 191]. The central hypothesis for brain attractors is that, once activated by appropriate activity, network behaviour is maintained by continuous reentry of activity [37,38]. This involves strong correlations between neuronal activities in the network and a high incidence of repeating firing patterns therein, being generated by the underlying attractors. Alternative attractors are commonly interpreted as alternative memories [39–46].

Certain pathways through the network may be favoured by preferred synaptic interactions between the neurons following developmental and learning processes [47–49]. The plasticity of these phenomena is likely to play a crucial role to shape the *meaningfulness* of an attractor and attractors must be stable at short time scales. Whenever the same information is presented in a network, the same pattern of activity is evoked in a circuit of functionally interconnected neurons, referred to as "cell assembly". In cell assemblies interconnected in this way, some ordered and precise neurophysiological activity referred to as preferred firing sequences, or spatio-temporal patterns of discharges, may recur above chance levels whenever the same information is presented [50–52]. Recurring firing patterns may be detected without a specific association to a stimulus in large networks of spiking neural networks or during spontaneous activity in electrophysiological recordings [53–55]. These patterns may be viewed as *spurious patterns* generated by *spurious attractors* that are associated with the underlying topology of the network rather than with a specific signal [56]. On the other hand, several examples exist of spatiotemporal firing patterns in behaving animals, from rats to primates [57–61], where preferred firing sequences can be associated to specific types of stimuli or behaviours. These can be viewed as *meaningful patterns* associated with *meaningful attractors*. However, meaningfulness cannot be reduced to the detection of a behavioural correlate [62–64]. The repeating activity in a network may also be considered meaningful if it allows the activation of neural elements that can be associated to other attractors, thus allowing the build-up of higher order dynamics by means of itinerancy between attractor basins and opening the way to chaotic neural dynamics [51,65–70].

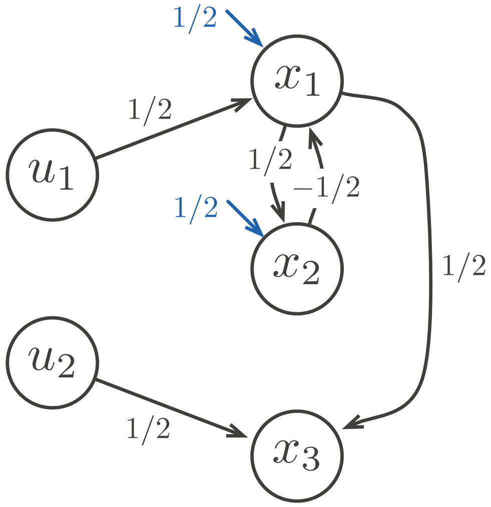

Figure 1. A simple neural network. The network is formed by two input units (u_1,u_2) and three activation cells (x_1,x_2,x_3). In this example the synaptic weights are all equal to 1/2, with positive sign corresponding to an excitatory input and a negative sign corresponding to a negative input. Notice that both cells x_1 and x_2 receive an excitatory background activity weighing 1/2.
doi:10.1371/journal.pone.0094204.g001

The dynamics of rather simple Boolean recurrent neural networks can implement an associative memory with bioinspired features [71,72]. In the Hopfield framework, stable equilibria of the network that do not represent any valid configuration of the optimisation problem are referred to as *spurious attractors*. Spurious modes can disappear by "unlearning" [71], but rational successive memory recall can actually be implemented by triggering spurious modes and achieving meaningful memory storage [66,73–77]. In this paper, the notions of attractors, meaningful attractors, and spurious attractors are reformulated in our precise Boolean network context. Networks will then be classified according to their ability to alternate between different types of attractive behaviours. For this purpose, the following definitions need to be introduced.

Formal Definitions. As preliminary notations, for any $k>0$, the space of k-dimensional Boolean vectors is denoted by $\mathbb{B}^k$. For any vector $\vec{x}\in\mathbb{B}^k$ and any $0<i\leq k$, the i-th component of $\vec{x}$ is denoted by $(\vec{x})_i$. Moreover, the spaces of finite and infinite sequences of k-dimensional Boolean vectors are denoted by $[\mathbb{B}^k]^*$ and $[\mathbb{B}^k]^\omega$, respectively. Any finite sequence of length n of $[\mathbb{B}^k]^*$ will be denoted by an expression of the form $\vec{b}_1\cdots\vec{b}_n$, and any infinite sequence of $[\mathbb{B}^k]^\omega$ by an expression of the form $\vec{b}_1\vec{b}_2\vec{b}_3\cdots$, where each $\vec{b}_i\in\mathbb{B}^k$. For any finite sequence of Boolean vectors v, we let the expression v^ω denote the infinite sequence obtained by infinitely many consecutive concatenations of v, i.e. $v^\omega = vvvv\cdots$.

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

Note that the set of all possible distinct states of a given Boolean network $\mathcal{N}$ is always finite; indeed, if $\mathcal{N}$ possesses N activation cells, then there are at most 2^N distinct possible states of $\mathcal{N}$. Hence, any infinite evolution e_s of $\mathcal{N}$ consists of an infinite sequence of only finitely many distinct states. Therefore, in any evolution e_s of $\mathcal{N}$, there necessarily exists at least one state that recurs infinitely many times in the infinite sequence e_s, irrespective of the fact that e_s is periodic or not. The non-empty set of all such states that recurs infinitely often in the evolution e_s will be denoted by $\inf(e_s)$.

By definition, every state $\vec{x}$ that is visited only finitely often in e_s will no longer occur in e_s after some time step $t_{\vec{x}}$. By taking the maximum of these time steps $t_{\vec{x}}$, we obtain a global time step t such that all states of e_s occurring after time t will necessarily repeat infinitely often in e_s. Formally, there necessarily exists an index t such that, for all $i \geq t$, one has $\vec{x}(i) \in \inf(e_s)$. It is important to note that the reoccurrence of the states belonging to $\inf(e_s)$ after time step t does not necessarily occur in a periodic manner during the evolution e_s. Therefore, any evolution e_s consists of a possibly empty prefix of successive states that repeat only finite many times, followed by an infinite suffix of successive states that repeat infinitely often, yet not necessarily in a periodic way. A set of states of the form $\inf(e_s)$ for some evolution e_s is commonly called an *attractor* of $\mathcal{N}$ [36]. A precise definition can be given as follows:

Definition 2. Let $\mathcal{N}$ be some Boolean neural network with N activation cells. A set $A = \{\vec{y}_0, \ldots, \vec{y}_k\} \subseteq \mathbb{B}^N$ is called an attractor for $\mathcal{N}$ if there exists an input stream s such that the corresponding evolution e_s satisfies $\inf(e_s) = A$.

In other words, an attractor of a Boolean neural network is a set of states such that the behaviour of the network could eventually become forever confined to that set of states. In this sense, the definition of an attractor requires the infinite input stream context to be properly formulated.

In this work, we suppose that attractors can only be of two distinct types, namely either *meaningful* or *spurious*. For instance, the type of each attractor could be determined by its topological features or by its neurophysiological significance with respect to measurable observations associated with certain behaviours or sensory discriminations (see Section "Neurophysiological Meaningfulness" above). From this point onwards, any given network is assumed to be provided with a corresponding classification of all of its attractors into meaningful and spurious types. Further discussions about the attribution of the attractors to either types will be addressed in the forthcoming sections.

An infinite 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. In other words, an input stream is called meaningful (respectively spurious) if the network dynamics induced by this input stream will eventually become confined into some meaningful (respectively spurious) attractor. Then, the set of all meaningful input streams of $\mathcal{N}$ is called the *neural language* of $\mathcal{N}$ and is denoted by $L(\mathcal{N})$. Finally, an arbitrary set of input streams $L \subseteq [\mathbb{B}^M]^\omega$ is said to be *recognisable* by some Boolean neural network if there exists a network $\mathcal{N}$ such that $L(\mathcal{N}) = L$.

Besides, if $\mathcal{N}$ denotes some Boolean neural network provided with an additional specification of the type of each of its attractors, then the *complementary* network $\mathcal{N}^\complement$ is defined to be the same network as $\mathcal{N}$ yet with a completely opposite type specification of its attractors. Then, an attractor A is meaningful for $\mathcal{N}$ iff A is a spurious attractor for $\mathcal{N}^\complement$ and one has $L(\mathcal{N}^\complement) = L(\mathcal{N})^\complement$. All preceding definitions are illustrated by the next Example 2.

Example 2. Let us consider the network $\mathcal{N}$ described in Example 1 and illustrated in Figure 1. Let us further assume that the network state where the three cells x_1, x_2, x_3 simultaneously fire determines the meaningfulness of the attractors of $\mathcal{N}$. In other words, the meaningful attractors of $\mathcal{N}$ are precisely those containing the state $(1,1,1)^T$; all other attractors are assumed to be spurious.

Let us consider the periodic input stream $s = \left[\begin{pmatrix}0\\0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\right]^\omega$ and its corresponding evolution

$$\begin{array}{ccccc} e_s = \begin{pmatrix}0\\0\\0\end{pmatrix} & \Bigg[\begin{pmatrix}0\\0\\0\end{pmatrix} & \begin{pmatrix}1\\0\\0\end{pmatrix} & \begin{pmatrix}0\\1\\1\end{pmatrix}\Bigg]^\omega \\ t=0 & t=1 & t=2 & t=3 \\ & t=4 & t=5 & t=6 \\ & t=7 & \ldots & \end{array}$$

From time step $t=1$, the evolution e_s of $\mathcal{N}$ remains confined in a cyclic visit of the states $\inf(e_s) = \{(0,0,0)^T, (1,0,0)^T, (0,1,1)^T\}$. 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 the state $(1,1,1)^T$ does not belong to $\inf(e_s)$, the attractor $\inf(e_s)$ is spurious. Therefore, the input stream s is also spurious, and hence does not belong to the neural language of $\mathcal{N}$, i.e. $s \notin L(\mathcal{N})$.

Let us consider another periodic input stream $s' = \left[\begin{pmatrix}1\\1\end{pmatrix}\right]^\omega$ and its corresponding evolution

$$\begin{array}{cccc} e_{s'} = \Bigg[\begin{pmatrix}0\\0\\0\end{pmatrix} & \begin{pmatrix}1\\0\\0\end{pmatrix} & \begin{pmatrix}1\\1\\1\end{pmatrix} & \begin{pmatrix}0\\1\\1\end{pmatrix}\Bigg]^\omega \\ t=0 & t=1 & t=2 & t=3 \\ t=4 & \ldots & & \end{array}$$

The set of states $\inf(e_{s'}) = \{(0,0,0)^T, (1,0,0)^T, (1,1,1)^T, (0,1,1)^T\}$ is an attractor, and the evolution $e_{s'}$ of $\mathcal{N}$ is confined in $\inf(e_{s'})$ already from the very first time step $t=0$. Yet in this case, since the Boolean vector $(1,1,1)^T$ belongs to $\inf(e_{s'})$, the attractor $\inf(e_{s'})$ is meaningful. It follows that the input stream s' is also meaningful, and thus $s' \in L(\mathcal{N})$.

ω-Automata

Büchi Automata. A *finite deterministic Büchi automaton* [34] 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 as a directed labelled graph whose nodes and labelled edges correspond to the states and transitions of the automaton, respectively.

Given some 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 $\mathcal{A}$. Then, a *path* in $\mathcal{A}$ is 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, q_2, q_3 and the word $a_1 a_2 a_3 \cdots$ is the *label* of ρ. The state q_0 is called the *origin* of path ρ and ρ is said to be *initial* if its starting state is initial, i.e. if $q_0 = i$. If ρ is an infinite path, the set of states visited infinitely many times by ρ is denoted by $\inf(\rho)$.

An infinite initial path ρ of $\mathcal{A}$ is said to be *successful* if it visits at least one of the final states infinitely often, i.e. if $\inf(\rho) \bigcap \mathcal{F} \neq \emptyset$. An infinite word is then said to be *recognised* by $\mathcal{A}$ if it is the label of a successful infinite path in $\mathcal{A}$. The *language recognised by* $\mathcal{A}$, denoted by $L(\mathcal{A})$, is the set of all infinite words recognised by $\mathcal{A}$.

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 initial and final state and visiting precisely all states of c. A cycle c_j is said to be *accessible* from cycle c_i if there exists a path from some state of c_i to some state of c_j. Furthermore, a cycle is called *successful* if it contains a state belonging to $\mathcal{F}$, and *non-successful* otherwise.

An *alternating chain of length* $n \in \mathbb{N}$ (respectively *co-alternating chain of length* $n \in \mathbb{N}$) is a finite sequence of $n+1$ distinct cycles $(c_0, \ldots, c_n)$ 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, c_0 is accessible from c_1, and c_1 is also accessible from c_0 (we recall that ω denotes the least infinite ordinal). In this case, cycles c_0 and c_1 are said to communicate. For any $\alpha \leq \omega$, 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.

The above definitions are illustrated by the Example S1 and Figure S1 in File S1.

Muller Automata. A *finite deterministic Muller automaton* is a 5-tuple $\mathcal{A} = (Q, A, i, \delta, \mathcal{T})$, where Q, A, i, and δ are defined exactly like for deterministic Büchi automata, and $\mathcal{T} \subseteq \mathcal{P}(Q)$ is a set of states' sets called the *table of the automaton*. The notions of transition and path are defined as for deterministic Büchi automata. An infinite initial path ρ of $\mathcal{A}$ is now called *successful* if $\inf(\rho) \in \mathcal{T}$. Given a finite deterministic Muller automaton $\mathcal{A} = (Q, A, i, \delta, \mathcal{T})$, a cycle in $\mathcal{A}$ is called *successful* if it belongs to $\mathcal{T}$, and *non-succesful* otherwise. 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 *ω-language recognised by* $\mathcal{A}$, denoted by $L(\mathcal{A})$, is defined as the set of all infinite words recognised by $\mathcal{A}$. The class of all ω-languages recognisable by some deterministic Muller automata is precisely the class of *ω-rational languages* [79].

It can be shown that deterministic Muller automata are strictly more powerful than deterministic Büchi automata, but have an equivalent expressive power as non-deterministic Büchi automata, Rabin automata, Street automata, parity automata, and non-deterministic Muller automata [81].

For each ordinal α such that $0 < \alpha < \omega^\omega$, we introduce the concept of an *alternating tree* of length α in a deterministic 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, as illustrated in Figure 2. In order to describe this tree-like disposition, 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 deterministic Muller automata $\mathcal{A}$ and some strictly positive 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 *alternating tree* (respectively *co-alternating tree*) of length α 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:

(i) $C^{0,0}_{0,0}$ is successful (respectively non-successful);

(ii) $C^{i,j}_{k,l} \subsetneq C^{i,j}_{k,l+1}$, and $C^{i,j}_{k,l+1}$ is successful iff $C^{i,j}_{k,l}$ is non-successful;

(iii) $C^{i,j}_{k+1,0}$ is accessible from $C^{i,j}_{k,0}$, and $C^{i,j}_{k+1,0}$ is successful iff $C^{i,j}_{k,0}$ is non-successful;

(iv) $C^{i+1,2j}_{0,0}$ and $C^{i+1,2j+1}_{0,0}$ are both 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 non-successful.

An alternating tree of length α is said to be maximal in $\mathcal{A}$ if there is no alternating or co-altenrating tree in $\mathcal{A}$ of length $\beta > \alpha$. A co-alternating tree of length α is said to be maximal in $\mathcal{A}$ if exactly the same condition holds. An alternating tree of length α is illustrated in Figure 2.

The above definitions are illustrated by the Example S2 and Figure S2 in File S2.

Results

Hierarchical Classification of Neural Networks

Our notion of an attractor refers to a set of states such that the behaviour of the network could forever be confined into that set of states. In other words, an attractor corresponds to a cyclic behaviour of the network produced by an infinite input stream. According to these considerations, we provide a generalisation to this precise infinite input stream context of the classical equivalence result between Boolean neural networks and finite state automata [1–3]. More precisely, we show that, under some natural specific conditions on the specification of the type of their attractors, Boolean recurrent neural networks express the very same expressive power as deterministic Büchi automata. This equivalence result enables us to establish a hierarchical classification of neural networks by translating the Wagner classification theory from the Büchi automaton to the neural network context [35]. The obtained classification is intimately related to the attractive properties of the neural networks, and hence provides a new refined measurement of the computational power of Boolean neural networks in terms of their attractive behaviours.

Boolean Recurrent Neural Networks and Büchi Automata. We now prove that, under some natural conditions, Boolean recurrent neural networks are computationally equivalent to deterministic Büchi automata. Towards this purpose, we consider that the neural networks include selected elements belonging to an output layer. The activation of the output layer communicates the output of the system to the environment.

Formally, let us consider a recurrent neural network (X, U, a, b, c), as described in Definition 1, with N activation cells and M input units. In addition, let us assume that M' cells chosen among the N activation cells form the *output layer* of the neural network, denoted by $V = \{x_{i_j} : 1 \leq j \leq M'\} \subseteq X$. For graphical purpose, the activation cells of the output layer are represented as double-circled nodes in the next figures. Thus, a recurrent neural network is now defined by a tuple $\mathcal{N} = (X, U, V, a, b, c)$. Let us assume also that the specification type of the attractors of a network $\mathcal{N}$ is naturally related to its output layer as follows: an attractor $A = \{\vec{y}_0, \ldots, \vec{y}_k\}$ of $\mathcal{N}$ is considered *meaningful* if it contains at least one state where some output cell is spiking, i.e. if there exist $i \leq k$ and $j \leq N$ such that $x_j \in V$ and $(\vec{y}_i)_j = 1$; the attractor A is considered *spurious* otherwise. According to these assumptions, meaningful attractors refer to the cyclic behaviours of the network that induce some response activity of the system via its output layer, whereas spurious attractors refer to the cyclic behaviours of the system that do not evoke any response at all of the output layer.

It can be stated that the expressive powers of Boolean recurrent neural networks and deterministic Büchi automaton are equivalent. As a first step towards this result, the following proposition shows that any Boolean recurrent neural network can be simulated by some deterministic Büchi automaton.

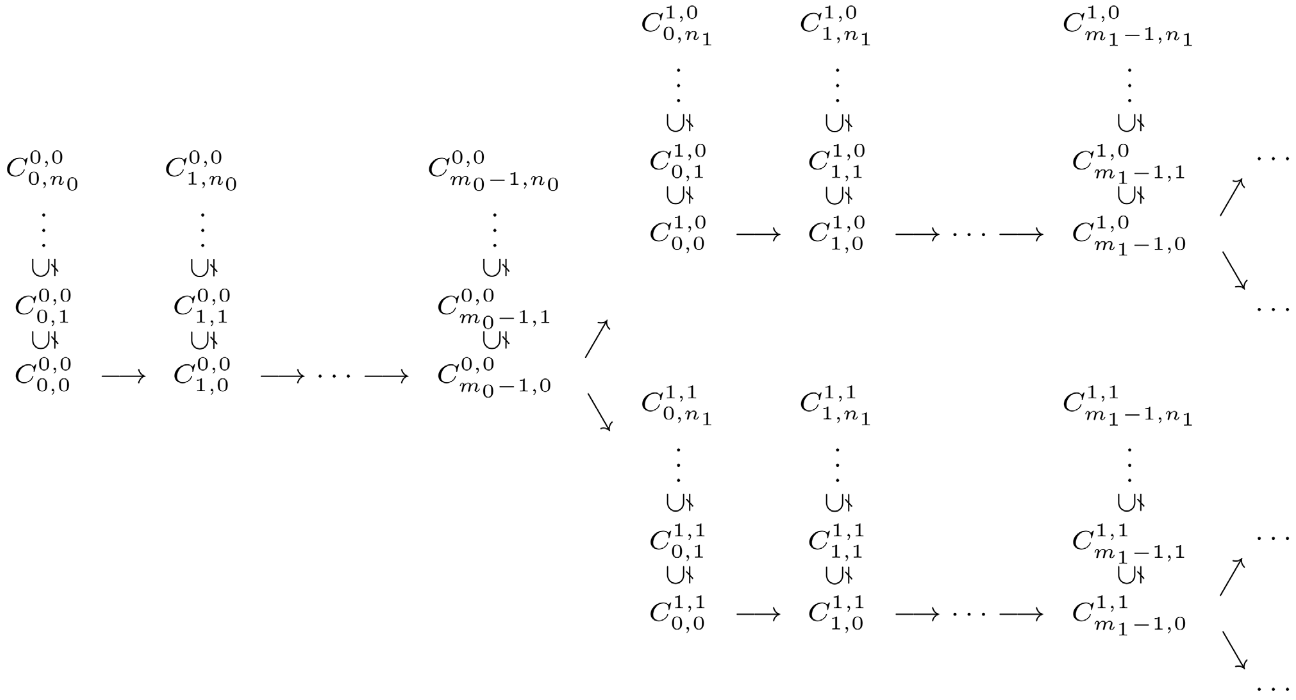

Figure 2. An alternating tree of length α, for some ordinal $0<\alpha<\omega^\omega$. Illustration of the inclusion and accessibility relations between cycles forming an alternating tree of length α.
doi:10.1371/journal.pone.0094204.g002

Proposition 1. *Let $\mathcal{N}$ be some Boolean recurrent neural network provided with an output layer. 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 some neural network given by the tuple (X,U,V,a,b,c), with $|X|=N$, $|U|=M$, and $V=\{x_{i_1},\ldots,x_{i_{M'}}\}\subseteq X$. Consider the deterministic Büchi automaton $\mathcal{A}_{\mathcal{N}}=(Q,A,i,\delta,\mathcal{F})$, where $Q=\mathbb{B}^N$, $A=\mathbb{B}^M$, i is the N-dimensional zero vector, $\mathcal{F}=\{\vec{x}\in Q : (\vec{x})_{i_k}=1 \text{ for some } 1\leq k\leq M'\}$, and $\delta : Q\times A\rightarrow Q$ is the function defined by $\delta(\vec{x},\vec{u})=\vec{x}'$ iff $\vec{x}'=\sigma(a\cdot\vec{x}+b\cdot\vec{u}+\vec{c})$. Note that the complexity of the transformation is exponential, since $|Q|=2^N$ and $|A|=2^M$.

According to this construction, any infinite evolution e_s of $\mathcal{N}$ naturally induces a corresponding infinite initial path $\rho(e_s)$ in $\mathcal{A}_{\mathcal{N}}$. Moreover, by the definitions of meaningful and spurious attractors of $\mathcal{N}$, an infinite input stream s is meaningful for $\mathcal{N}$ iff s is recognised by $\mathcal{A}_{\mathcal{N}}$. In other words, $s\in L(\mathcal{N})$ 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 1, any infinite evolution of the network $\mathcal{N}$ is naturally associated with a corresponding infinite initial path in the automaton $\mathcal{A}_{\mathcal{N}}$, and conversely, any infinite initial path in $\mathcal{A}_{\mathcal{N}}$ corresponds to some possible infinite evolution of $\mathcal{N}$. Consequently, there is a biunivocal correspondence between the *attractors* of the network $\mathcal{N}$ and the *cycles* in the graph of the corresponding Büchi automaton $\mathcal{A}_{\mathcal{N}}$. As a result, a procedure to compute all possible attractors of a given network $\mathcal{N}$ is obtained by firstly constructing the corresponding deterministic Büchi automaton $\mathcal{A}_{\mathcal{N}}$ and secondly listing all cycles in the graph of $\mathcal{A}_{\mathcal{N}}$.

As a second step towards the equivalence result, we prove now that any deterministic Büchi automaton can be simulated by some Boolean recurrent neural network.

Proposition 2. *Let $\mathcal{A}$ be some deterministic Büchi automaton over the alphabet $\mathbb{B}^M$, with $M\geq 1$. Then there exists a Boolean recurrent neural network $\mathcal{N}_{\mathcal{A}}$ provided with an output layer such that $L(\mathcal{A})=L(\mathcal{N}_{\mathcal{A}})$.*

Proof. Let $\mathcal{A}=(Q,\mathbb{B}^M,q_1,\delta,\mathcal{F})$ be some deterministic Büchi automaton over alphabet $\mathbb{B}^M$, with $Q=\{q_1,\ldots,q_N\}$, and $\mathcal{F}=\{q_{i_1},\ldots,q_{i_K}\}\subseteq Q$. Consider the network $\mathcal{N}_{\mathcal{A}}=(X,U,V,a,b,c)$ with $2^M+N+1+M$ cells given as follows: firstly, $X=\{x_i : 0\leq i\leq 2^M+N\}$, where X is decomposed into a set of 2^M "letter cells" $X_L=\{x_i : 0\leq i<2^M\}$, a "delay-cell" x_{2^M}, and a set of N "state cells" $X_S=\{x_i : 2^M<i\leq 2^M+N\}$; secondly, the set of $|M|$ "input units" $U=\{u_0,\ldots,u_{M-1}\}$, and thirdly, the outptut layer $V=\{x_{2^M+j} : q_j\in\mathcal{F}\}$. The idea of the simulation is that the "letter cells" and "state cells" of the network $\mathcal{N}_{\mathcal{A}}$ simulate the letters and states currently read and entered by the automaton $\mathcal{A}$, respectively.

Towards this purpose, the weight matrices a, b, and c are described as follows. Concerning the matrix b: for any $x_k\in X_L$, we consider the binary decomposition of k, namely $k=\sum_{j=0}^{M-1}\beta_{kj}\cdot 2^j$, with $\beta_{kj}\in\{0,1\}$, and for any $0\leq j<M$, we set the weight $b_{k,j}=\beta_{kj}\cdot 2^j+(\beta_{kj}-1)$; for all other k, we set $b_{k,j}=0$, for any $0\leq j<M$. Concerning the matrix c: for any $x_k\in X_L$, we set $c_k=1-k$; we also set $c_{2^M}=c_{2^M+1}=1$; for all other k, we set $c_k=0$. Concerning the matrix a: we set $a_{2^M+1,2^M}=-1$, and for any $x_k\in X_L$ and any $x_{2^M+i},x_{2^M+j}\in X_S$, we set $a_{2^M+j,k}=a_{2^M+j,2^M+i}=1/2$ iff $(q_i,\vec{\beta}_k,q_j)$ is a transition of $\mathcal{A}$; otherwise, for any pair of indices $i_1,i_2\in\{0,\ldots,2^M+N\}$ such that a_{i_1,i_2} has not been set to -1 or $1/2$, we set $a_{i_1,i_2}=0$. This construction is illustrated in Figure 3.

According to this construction, if we let $\vec{\beta}_k$ denote the boolean vector whose components are the β_{kj}'s (for $0\leq j<M$), one has that the "letter cell" x_k will spike at time $t+1$ iff the input vector

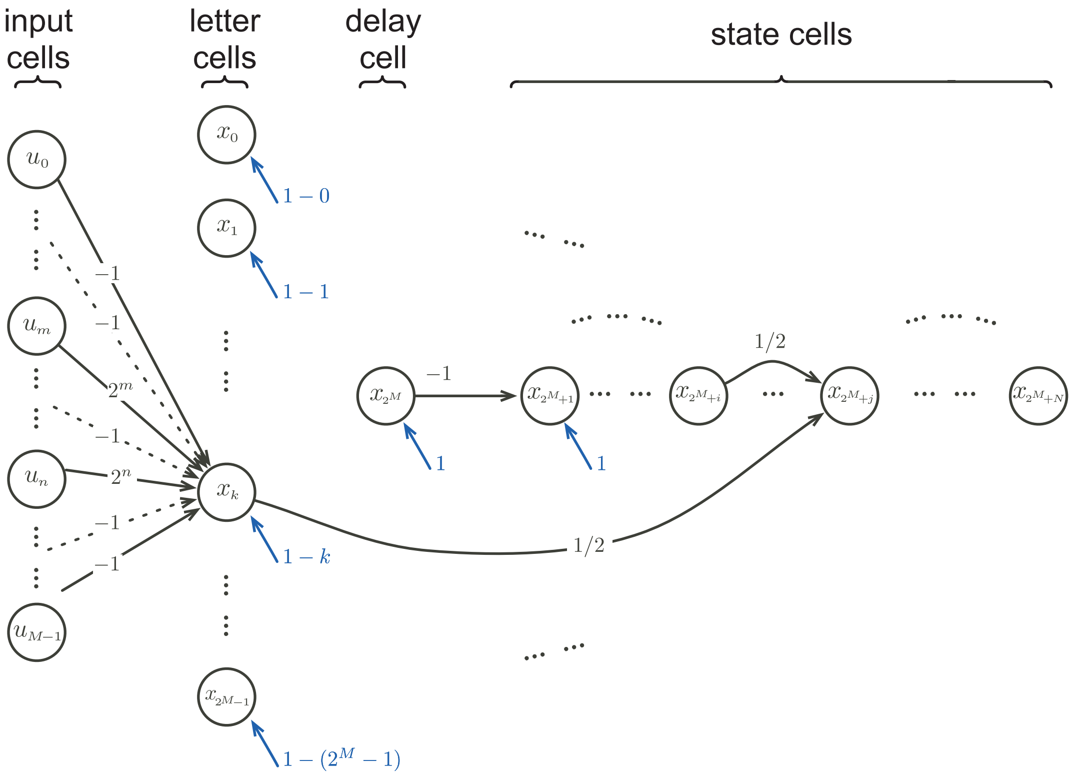

Figure 3. The network $\mathcal{N}_{\mathcal{A}}$ described in the proof of Proposition 2. The network is characterised by a set of M input cells $U=\{u_0,\ldots,u_{M-1}\}$ reading the alphabet $\mathbb{B}^M$, 2^M "letter cells" $X_L=\{x_i : 0\leq i<2^M\}$, a "delay-cell" x_{2^M}, and a set of N "state cells" $X_S=\{x_i : 2^M<i\leq 2^M+N\}$. The idea of the simulation is that the "letter cells" and "state cells" of the network $\mathcal{N}_{\mathcal{A}}$ simulate the letters and states currently read and entered by the automaton $\mathcal{A}$, respectively. In this illustration, we assume that the binary decomposition of k is given by $k=2^m+2^n$, so that the "letter cell" x_k receives synaptic connections of intensities 2^m and 2^n from input cells u_m and u_n, respectively, and it receives synaptic connections of intensities -1 from any other input cells. Consequently, the "letter cell" x_k becomes active at time $t+1$ iff the sole input cells u_m and u_n are active at time t. The synaptic connections to other "letter cells" are not illustrated. Moreover, the synaptic connections $a_{2^M+j,k}=a_{2^M+j,2^M+i}=1/2$ model the transition $(q_i,\vec{\beta}_k,q_j)$ of automaton $\mathcal{A}$. The synaptic connections modelling other transitions are not illustrated.
doi:10.1371/journal.pone.0094204.g003

$\vec{\beta}_k\in\mathbb{B}^M$ is received at time t. Moreover, at every time step $t>0$, a unique "letter cell" $x_k\in X_L$ and "state cell" $x_{2^M+i}\in X_S$ are spiking, and, if $\mathcal{A}$ performs the transition $(q_i,\vec{\beta}_k,q_j)$ at time t, then network $\mathcal{N}_{\mathcal{A}}$ evokes the spiking pattern $x_k(t)=x_{2^M+i}(t)=x_{2^M+j}(t+1)=1$. The relation between the final states $\mathcal{F}$ of $\mathcal{A}$ and the output layer V of $\mathcal{N}_{\mathcal{A}}$ ensures that any infinite input stream $s\in[\mathbb{B}^M]^\omega$ is recognised by $\mathcal{A}$ if and only if s is meaningful for $\mathcal{N}_{\mathcal{A}}$. Therefore, $L(\mathcal{A})=L(\mathcal{N}_{\mathcal{A}})$.

The proof of Proposition 2 can be generalised to any network dynamics driven by unate local transition functions $f_i:\mathbb{B}^{N+M}\rightarrow\mathbb{B}$, for $i=1,\ldots,N$, rather than by the N threshold local transition functions defined by Equation 1. Since unate functions are a generalisation of threshold functions, this proof can be interesting in the broader context of switching theory.

Propositions 1 and 2 yield to the following equivalence between recurrent neural networks and deterministic Büchi automata.

Theorem 1. *Let $L\subseteq[\mathbb{B}^k]^\omega$ for some $k\geq 1$. Then L is recognisable by some Boolean recurrent neural network provided with an output layer iff L is recognisable by some deterministic Büchi automaton.*

Proof. Proposition 1 shows that every language recognisable by some Boolean recurrent neural network is also recognisable by some deterministic Büchi automaton. Conversely, Proposition 2 shows that every language recognisable by some deterministic Büchi automaton is also recognisable by some Boolean recurrent neural network.

The two procedures given in the proofs of propositions 1 and 2 are illustrated by the Example S3 and Figure S3 in File S3.

RNN Hierarchy. In the theory of infinite word reading machines, abstract devices are commonly classified according to the topological complexity of their underlying ω-language (i.e., the languages of infinite words that they recognise). Such classifications provide an interesting measurement of the expressive power of various kinds of infinite word reading machines. In this context, the most refined hierarchical classification of ω-automata – or equivalently, of ω-rational languages – is the so-called *Wagner hierarchy* [35].

Here, this classification approach is translated from the ω-automaton to the neural network context. More precisely, according to the equivalence given by Theorem 1, the Wagner hierarchy can naturally be translated from Büchi automata to Boolean neural networks. As a result, a hierarchical classification of first-order Boolean recurrent neural networks is obtained. Interestingly, the obtained classification is tightly related to the attractive properties of the networks, and, more precisely, refers to the ability of the networks to switch between meaningful and spurious attractive behaviours along their evolutions. Hence, the obtained hierarchical classification provides a new measurement of complexity of neural networks associated with their abilities to switch between different types of attractors along their evolutions.

As a first step, 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 $\mathbb{B}^k$. Accordingly, one can show that the basic open sets of $[\mathbb{B}^k]^\omega$ are the sets of infinite sequences of k-dimensional Boolean vectors which all begin with a same prefix, or formally, the sets of the form $\vec{b}_1 \cdots \vec{b}_n [\mathbb{B}^k]^\omega$, where $\vec{b}_1, \ldots, \vec{b}_n \in \mathbb{B}^k$. An open set is then defined as a union of basic open sets. Moreover, as usual, 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 $[\mathbb{B}^k]^\omega$. Now, given two Boolean recurrent neural networks $\mathcal{N}_1$ and $\mathcal{N}_2$ with M_1 and M_2 input units respectively, we say that $\mathcal{N}_1$ *reduces* (or *Wadge reduces* or *continuously reduces*) to $\mathcal{N}_2$, denoted by $\mathcal{N}_1 \leq_W \mathcal{N}_2$, iff there exists a continuous function $f : [\mathbb{B}^{M_1}]^\omega \to [\mathbb{B}^{M_2}]^\omega$ such that, for any input stream $s \in [\mathbb{B}^{M_1}]^\omega$, one has $s \in L(\mathcal{N}_1) \Leftrightarrow f(s) \in L(\mathcal{N}_2)$, or equivalently, such that $L(\mathcal{N}_1) = f^{-1}(L(\mathcal{N}_2))$ [78]. Intuitively, $\mathcal{N}_1 \leq_W \mathcal{N}_2$ iff the problem of determining whether some input stream s belongs to the neural language of $\mathcal{N}_1$ (i.e. whether s is meaningful for $\mathcal{N}_1$) reduces via some simple function f to the problem of knowing whether $f(s)$ belongs to the neural language of $\mathcal{N}_2$ (i.e. whether s is meaningful for $\mathcal{N}_2$). 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 called *non-self-dual* if $\mathcal{N} \not\equiv_W \mathcal{N}^\complement$, which can be proved to be equivalent to saying that $\mathcal{N} \perp_W \mathcal{N}^\complement$ [78]. We recall that the network $\mathcal{N}^\complement$, as defined in Section "Formal Definitions", corresponds to the network $\mathcal{N}$ whose type specification of its attractors has been inverted. Consequently, $\mathcal{N}^\complement$ does not correspond *a priori* to some neural network provided with an output layer. 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.

The continuous reduction relation over the class of Boolean recurrent neural networks naturally induces a hierarchical classification of networks formally defined as follows:

Definition 3. The collection of all Boolean recurrent neural networks ordered by the reduction "$\leq_W$" is called the RNN *hierarchy*.

We now provide a precise description of the RNN hierarchy. The result is obtained by drawing a parallel between the RNN hierarchy and the restriction of the Wagner hierarchy to Büchi automata. For this purpose, let us define the *DBA hierarchy* to be the collection of all deterministic Büchi automata over multidimensional Boolean alphabets $\mathbb{B}^k$ ordered by the continuous reduction relation "$\leq_W$". More precisely, given two deterministic Büchi automata $\mathcal{A}_1$ and $\mathcal{A}_2$, we set $\mathcal{A}_1 \leq_W \mathcal{A}_2$ iff there exists a continuous function f such that, for any input stream s, one has $s \in L(\mathcal{A}_1) \Leftrightarrow f(s) \in L(\mathcal{A}_2)$. The following result shows that the RNN hierarchy and the DBA hierarchy are actually isomorphic. Moreover, a possible isomorphism is given by the mapping described in Proposition 1 which associates to every network $\mathcal{N}$ a corresponding deterministic Büchi automaton $\mathcal{A}_\mathcal{N}$.

Proposition 3. *The RNN hierarchy and the DBA hierarchy are isomorphic.*

Proof. Consider the mapping described in Proposition 1 which associates to every network $\mathcal{N}$ a corresponding deterministic automaton $\mathcal{A}_\mathcal{N}$. We prove that this mapping is an embedding from the RNN hierarchy into the DBA hierarchy. Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be any two networks, and let $\mathcal{A}_{\mathcal{N}_1}$ and $\mathcal{A}_{\mathcal{N}_2}$ be their corresponding deterministic Büchi automata. Proposition 1 ensures that $L(\mathcal{N}_1) = L(\mathcal{A}_{\mathcal{N}_1})$ and $L(\mathcal{N}_2) = L(\mathcal{A}_{\mathcal{N}_2})$. Hence, one has $\mathcal{N}_1 \leq_W \mathcal{N}_2$ iff by definition there exists a continuous function f such that $L(\mathcal{N}_1) = f^{-1}(L(\mathcal{N}_2))$ iff there exists a continuous function f such that $L(\mathcal{A}_{\mathcal{N}_1}) = f^{-1}(L(\mathcal{A}_{\mathcal{N}_2}))$, iff by definition $\mathcal{A}_{\mathcal{N}_1} \leq_W \mathcal{A}_{\mathcal{N}_2}$. Therefore $\mathcal{N}_1 \leq_W \mathcal{N}_2$ iff $\mathcal{A}_{\mathcal{N}_1} \leq_W \mathcal{A}_{\mathcal{N}_2}$. It follows that $\mathcal{N}_1 <_W \mathcal{N}_2$ iff $\mathcal{A}_{\mathcal{N}_1} <_W \mathcal{A}_{\mathcal{N}_2}$, proving that the considered mapping is an embedding. We now show that, up to the continuous equivalence relation "$\equiv_W$", this mapping is also onto. Let $\mathcal{A}$ be some deterministic Büchi automaton. By Proposition 2, there exists a network $\mathcal{M} = \mathcal{N}_\mathcal{A}$ such that $L(\mathcal{A}) = L(\mathcal{M})$. Moreover, by Proposition 1, the automaton $\mathcal{A}_\mathcal{M}$ satisfies $L(\mathcal{A}_\mathcal{M}) = L(\mathcal{M}) = L(\mathcal{A})$. It follows that for any infinite input stream s, one has $s \in L(\mathcal{A}_\mathcal{M})$ iff $s \in L(\mathcal{A})$, meaning that both $\mathcal{A}_\mathcal{M} \leq_W \mathcal{A}$ and $\mathcal{A} \leq_W \mathcal{A}_\mathcal{M}$ hold, and thus $\mathcal{A}_\mathcal{M} \equiv_W \mathcal{A}$. Therefore, for any deterministic Büchi automaton $\mathcal{A}$, there exists a neural network $\mathcal{M}$ such that $\mathcal{A}_\mathcal{M} \equiv_W \mathcal{A}$, meaning precisely that up to the continuous equivalence relation "$\equiv_W$", the mapping $\mathcal{N} \mapsto \mathcal{A}_\mathcal{N}$ described in Proposition 1 is onto. This concludes the proof.

By Proposition 3 and the usual results of the DBA hierarchy, a precise description of the RNN hierarchy can be given. 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 $\omega + 1$, meaning that the RNN hierarchy has a height of $\omega + 1$. (A strict chain of length α 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 strict chain is said to be maximal if its length is at least as large as the length of every other strict chain.) Furthermore, the maximal antichains of the RNN hierarchy have length 2, meaning that the RNN hierarchy has a width of 2. (An antichain of length α in the RNN hierarchy is a sequence of neural networks $(\mathcal{N}_k)_{k \in \alpha}$ such that $\mathcal{N}_i \perp_W \mathcal{N}_j$ for all $i, j \in \alpha$ with $i \neq j$; an antichain is said to be maximal if its length is at least as large as the length of every other antichain.) More precisely, it can be shown that incomparable networks are equivalent (for the relation $\equiv_W$) up to complementation, i.e., for any two networks $\mathcal{N}_1$ and $\mathcal{N}_2$, one has $\mathcal{N}_1 \perp_W \mathcal{N}_2$ iff $\mathcal{N}_1$ and $\mathcal{N}_2$ are non-self-dual and $\mathcal{N}_1 \equiv_W \mathcal{N}_2^\complement$. These properties imply that, up to equivalence and complementation, the RNN hierarchy is actually a well-ordering. In fact, the RNN hierarchy consists of an infinite alternating succession of pairs of non-self-dual and single self-dual classes, overhung by an additional single non-self-dual class at the first limit level ω, as illustrated in Figure 4.

For convenience reasons, the degree of a network $\mathcal{N}$ in the RNN hierarchy is defined such that the same degree is shared by both non-self-dual networks at some level and self-dual networks located just one level higher, as illustrated in Figure 4:

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

Moreover, the equivalence between the DBA and RNN hierarchies ensures that the RNN hierarchy is actually decidable, in the sense that there exists an algorithmic procedure which is able to compute the degree of any network in the RNN hierarchy. All the above properties of the RNN hierarchy are summarised in the following result.

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

Proof. The DBA hierarchy consists of a decidable pre-well-ordering of width 2 and height $\omega + 1$ [79]. Proposition 3 ensures that the RNN hierarchy and the DBA hierarchy are isomorphic.

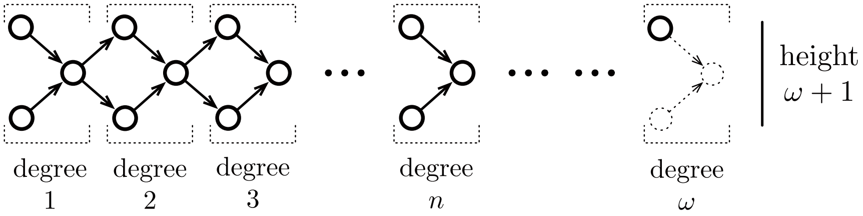

Figure 4. The RNN hierarchy. An infinite alternating succession of pairs of non-self-dual classes of networks followed by single self-dual classes of networks, all of them overhung by an additional single non-self-dual class at the first limit level. Circles represent the equivalence classes of networks (with respect to the relation "$\equiv_W$") and arrows between circles represent the strict reduction "$<_W$" between all elements of the corresponding classes.
doi:10.1371/journal.pone.0094204.g004

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 visits of meaningful and spurious attractors along some evolution.

Theorem 3. *Let $\mathcal{N}$ be some network provided with an additional specification of an output layer, $\mathcal{A}_{\mathcal{N}}$ be the corresponding deterministic Büchi automaton of $\mathcal{N}$, and $n>0$.*

- *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.*
- *Symmetrically, 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 $d(\mathcal{N})=\omega$ and $\mathcal{N}$ is non-self-dual.*

Proof. By Proposition 3, 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 contained in this automaton [79].

By Theorem 3, the decidability procedure of the degree of a neural network $\mathcal{N}$ in the RNN hierarchy consists firstly in translating the network $\mathcal{N}$ into its corresponding deterministic Büchi automaton $\mathcal{A}_{\mathcal{N}}$, as described in Proposition 1, and secondly in returning the ordinal $\alpha<\omega+1$ corresponding to the length of a maximal alternating chain or co-alternating chain contained in $\mathcal{A}_{\mathcal{N}}$. Note that this procedure can clearly be achieved 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, then 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 in fact directly related to the attractive properties of this network.

More precisely, it can be observed that the complexity measurement provided by the RNN hierarchy actually corresponds to the maximal number of times that a network might alternate between visits of meaningful and spurious attractors along some evolution. Indeed, the existence of a maximal alternating or co-alternating chain $(c_0,\ldots,c_n)$ 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 visits of successful and non-successful cycles. Yet this is precisely equivalent to saying that every evolution of $\mathcal{N}$ can only alternate at most n times between visits of meaningful and spurious attractors before eventually becoming trapped forever by a last attractor. In this case, Theorem 3 ensures that the degree of $\mathcal{N}$ is equal to n. Moreover, 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 visits of the cycles c_1 and c_2. This is equivalent to saying that there exists an evolution of $\mathcal{N}$ that might alternate infinitely many times between visits of a meaningful and a spurious attractor. By Theorem 3, the degree of $\mathcal{N}$ is equal to ω is this case. Therefore, the RNN hierarchy provides a new measurement of complexity of neural networks according to their ability to maximally alternate between different types of attractors along their evolutions.

Finally, the decidability procedure of the RNN hierarchy is illustrated by the Example S4 in File S4.

Refined Hierarchical Classification of Neural Networks

In this section, we show that by relaxing the restrictions on the specification of the type of their attractors, the networks significantly increase their expressive power from deterministic Büchi automata up to Muller automata [80]. Hence, by translating once again the Wagner classification theory from the Muller automaton to the neural network context, another more refined hierarchical classification of Boolean neural networks can be obtained. The obtained classification is also tightly related to the attractive properties of the networks, and hence provides once again a new refined measurement of complexity of Boolean recurrent neural networks in terms of their attractive behaviours.

Boolean Recurrent Neural Networks and Muller Automata. The assumption that the networks are provided with an additional description of an output layer, which would subsequently influence the type specifications (meaningful/spurious) of their attractors, is not necessary anymore from this point onwards. Instead, let us assume that, for any network, the precise classification of its attractors into meaningful and spurious types is known. How the meaningfulness of the attractors is determined is an issue that is not considered here. For instance, the specification of the type of each attractor might have been determined by its neurophysiological significance with respect to measurable observations associated to certain behaviours or sensory discriminations. Formally, in this section, a recurrent neural network consists of a tuple $\mathcal{N}=(X,U,a,b,c)$, as described in Definition 1, but also provided with an additional specification into meaningful and spurious type for each one of its attractors.

We now prove that, by totally relaxing the restrictions on the specification of the type of their attractors, the Boolean neural networks significantly increase their expressive powers from deterministic Büchi automata up to Muller automata. The following straightforward generalisation of Proposition 1 states that any such Boolean recurrent neural network can be simulated by some deterministic Muller automaton.

Proposition 4. *Let $\mathcal{N}$ be some Boolean recurrent neural network provided with a type specification of each of its attractors. Then there exists a deterministic Muller 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,U,a,b,c), with $|X| = N$, $|U| = M$, and let the meaningful attractors of $\mathcal{N}$ be given by $A_1,\ldots,A_K$, all others being spurious. Now, consider the deterministic 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 $\delta(\vec{x},\vec{u}) = \vec{x}'$ iff $\vec{x}' = \sigma(a\cdot\vec{x} + b\cdot\vec{u} + c)$, and $\mathcal{T} = \{A_1,\ldots,A_K\}$. According to this construction, any input stream s is meaningful for $\mathcal{N}$ iff s is recognised by $\mathcal{A}_{\mathcal{N}}$. In other words, $s \in L(\mathcal{N})$ iff $s \in L(\mathcal{A}_{\mathcal{N}})$, and therefore $L(\mathcal{N}) = L(\mathcal{A}_{\mathcal{N}})$.

Conversely, as a generalisation of Proposition 2, we can prove that any deterministic Muller automaton can be simulated by some Boolean recurrent neural network provided with a suitable type specification of its attractors.

Proposition 5. *Let $M > 0$ and let $\mathcal{A}$ be some deterministic Muller automaton over the alphabet $\mathbb{B}^M$. Then there exists a Boolean recurrent neural network $\mathcal{N}_{\mathcal{A}}$ provided with a type specification of each of its attractors 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 $A = \mathbb{B}^M$, $Q = \{q_1,\ldots,q_N\}$ and $\mathcal{T} \subseteq \mathcal{P}(Q)$. Now, consider the network $\mathcal{N}_{\mathcal{A}} = (X,U,a,b,c)$ described in the proof of Proposition 2. It remains to define which are the meaningful and spurious attractors of $\mathcal{N}_{\mathcal{A}}$. As mentioned in the proof of Proposition 2, at every time step $t > 0$, only one among the "state cells" $\{x_{2^M+1},\ldots,x_{2^M+N}\}$ is spiking. Hence, for any state $\vec{y}$ of $\mathcal{N}_{\mathcal{A}}$ that might occur at some time step $t > 0$, let $i(\vec{y}) \in \{1,\ldots,N\}$ be the index such that $x_{2^M+i(\vec{y})}$ is the unique "state cell" which is spiking during state $\vec{y}$. An attractor $\{\vec{y}_0,\ldots,\vec{y}_k\}$ of $\mathcal{N}_{\mathcal{A}}$ is then said to be meaningful iff $\{q_{i(\vec{y}_0)},\ldots,q_{i(\vec{y}_k)}\} \in \mathcal{T}$.

Consequently, for any infinite infinite sequence $s \in [\mathbb{B}^M]^\omega$, the infinite path ρ_s in $\mathcal{A}$ satisfies $\inf(\rho_s) \in \mathcal{T}$ iff the evolution e_s in $\mathcal{N}_{\mathcal{A}}$ is such that $\inf(e_s)$ is a meaningful attractor. Therefore, s is recognised by $\mathcal{A}$ iff s is meaningful for $\mathcal{N}_{\mathcal{A}}$, showing that $L(\mathcal{A}) = L(\mathcal{N}_{\mathcal{A}})$.

Propositions 4 and 5 yield the following equivalence between Boolean recurrent neural networks and deterministic Muller automata.

Theorem 4. *Let $L \subseteq [\mathbb{B}^k]^\omega$ for some $k > 0$. Then the following conditions are equivalent:*

(a) *L is recognisable by some Boolean recurrent neural network provided with a type specification of its attractors;*

(b) *L is recognisable by some deterministic Muller automaton;*

(c) *L is ω-rational.*

Proof. The equivalence between conditions a and b is given by propositions 4 and 5. The equivalence between conditions b and c is a well-known result of automata theory [79].

The two procedures described in the proofs of propositions 4 and 5 are illustrated by the Example S5 and Figure S4 in File S5.

Complete RNN Hierarchy. In this section, we prove that the collection of Boolean recurrent neural networks ordered by the continuous reduction corresponds to a refined hierarchical classification of height ω^ω. This classification is directly related to the attractive properties of the networks, and therefore provides a new refined measurement of complexity of neural networks according to their attractive behaviours. This hierarchical classification is formally defined as follows.

Definition 4. *The collection of all Boolean recurrent neural networks provided with a type specification of their attractors ordered by the continuous reduction "$\leq_W$" is called the complete RNN hierarchy.*

Like in Section "RNN Hierarchy", a precise characterisation of the complete RNN hierarchy can be obtained by translating the Wagner classification theory from the Muller automaton to the neural network context. For this purpose, we shall consider the collection of all deterministic Muller automata over multidimensional Boolean alphabets $\mathbb{B}^k$ ordered by the continuous reduction "$\leq_W$". This hierarchy is commonly referred to as the *Wagner hierarchy* [35]. A generalisation of Proposition 3 shows that the complete RNN hierarchy and the Wagner hierarchy are isomorphic, and a possible isomorphism is also given by the mapping described in Proposition 4 which associates to every network $\mathcal{N}$ a corresponding deterministic Muller automaton $\mathcal{A}_{\mathcal{N}}$.

Proposition 6. *The complete RNN hierarchy and the Wagner hierarchy are isomorphic.*

Proof. Consider the mapping described in Proposition 4 which associates to every network $\mathcal{N}$ a corresponding deterministic Muller automaton $\mathcal{A}_{\mathcal{N}}$. A similar reasoning as the one presented in the proof of Proposition 3 shows that this mapping is an isomorphism between the complete RNN hierarchy and the Wagner hierarchy.

By Proposition 6 and the usual results on the Wagner hierarchy [35], the following precise description of the complete RNN hierarchy can be given. First of all, like the RNN hierarchy, the complete RNN hierarchy also consists of a pre-well ordering of width 2, and any two networks $\mathcal{N}_1$ and $\mathcal{N}_2$ satisfy the incomparability relation $\mathcal{N}_1 \perp_W \mathcal{N}_2$ iff $\mathcal{N}_1$ and $\mathcal{N}_2$ are non-self-dual networks such that $\mathcal{N}_1 \equiv_W \mathcal{N}^{\complement}_2$. However, while the RNN hierarchy has only height $\omega + 1$, the complete RNN hierarchy shows a height of ω^ω levels. In fact, the complete RNN hierarchy consists of an infinite alternating succession of pairs of non-self-dual and single self-dual classes, with non-self-dual classes at each limit level, as illustrated in Figure 5. For convenience reasons, the degree $d^*(\mathcal{N})$ of a network $\mathcal{N}$ in the complete RNN hierarchy is also defined such that the same degree is shared by both non-self-dual networks at some level and self-dual networks located just one level higher, namely:

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

Besides, the isomorphism between the Wagner hierarchy and the complete RNN hierarchy ensures that the complete RNN hierarchy is actually decidable, in the sense that there exists an algorithmic procedure allowing to compute the degree of any network in the complete RNN hierarchy. The following theorem summarises the properties of the complete RNN hierarchy.

Theorem 5. *The complete 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 [35]. Proposition 6 ensures that the complete RNN hierarchy and the Wagner hierarchy are isomorphic.

The following result provides a detailed description of the decidability procedure of the complete 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

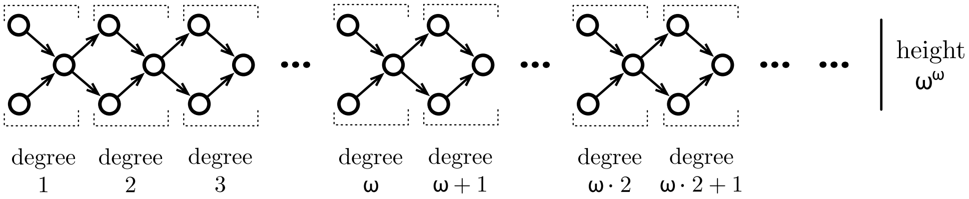

Figure 5. The complete RNN hierarchy. A transfinite alternating succession of pairs of non-self-dual classes of networks followed by single self-dual classes of networks, with non-self-dual classes at each limit level.
doi:10.1371/journal.pone.0094204.g005

there exists an alternating tree or a co-alternating tree of length α in the deterministic Muller automaton $\mathcal{A}_{\mathcal{N}}$.

Theorem 6. *Let $\mathcal{N}$ be some Boolean recurrent network provided with a type specification of its attractors, $\mathcal{A}_{\mathcal{N}}$ be the corresponding deterministic 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 alternating tree of length α and no maximal co-alternating tree of length α, then $d^*(\mathcal{N})=\alpha$ and $\mathcal{N}$ is non-self-dual.*
- *If there exists in $\mathcal{A}_{\mathcal{N}}$ a maximal co-alternating tree of length α and no maximal alternating tree of length α, then $d^*(\mathcal{N})=\alpha$ and $\mathcal{N}$ is non-self-dual.*
- *If there exist in $\mathcal{A}_{\mathcal{N}}$ both a maximal alternating tree as well as a maximal co-alternating tree of length α, then $d^*(\mathcal{N})=\alpha$ and $\mathcal{N}$ is self-dual.*

Proof. By Proposition 6, the degree of a network $\mathcal{N}$ in the complete RNN hierarchy is equal to the degree of its corresponding deterministic Muller automaton $\mathcal{A}_{\mathcal{N}}$ in the Wagner hierarchy. Moreover, the degree of a deterministic Muller automaton in the Wagner hierarchy corresponds precisely to the length of a maximal alternating or co-alternating tree contained in this automaton [35,82].

The decidability procedure of the degree of a neural network $\mathcal{N}$ in the complete RNN hierarchy thus consists in first translating the network $\mathcal{N}$ into its corresponding deterministic Muller automaton $\mathcal{A}_{\mathcal{N}}$, as described in Proposition 4, and then returning the ordinal $\alpha<\omega^\omega$ corresponding to the length of a maximal alternating tree, or co-alternating tree, contained in $\mathcal{A}_{\mathcal{N}}$. Note that this procedure can be achieved by some graph analysis of the automaton $\mathcal{A}_{\mathcal{N}}$, since the graph of $\mathcal{A}_{\mathcal{N}}$ is always finite.

By Theorem 6, the degree of a neural network $\mathcal{N}$ in the complete RNN hierarchy corresponds precisely to the length of a maximal alternating, or co-alternating, tree contained in $\mathcal{A}_{\mathcal{N}}$. Since alternating and co-alternating trees are defined in terms of cycles in the graph of the Muller automaton, and according to the biunivocal correspondence between cycles in $\mathcal{A}_{\mathcal{N}}$ and attractors of $\mathcal{N}$, it can be deduced that, like for the RNN hierarchy, the complexity of a network in the complete RNN hierarchy is also directly related to the attractive properties of this network. In fact, the complexity measurement provided by the complete RNN hierarchy refers to the maximal number of times that a network might alternate between visits of meaningful and spurious attractors along some evolution.

More precisely, the ω first levels of the complete RNN hierarchy provide a classification of the collection of networks that might switch at most *finitely* many times between different types of attractors along their evolutions. Indeed, by Theorem 6, for any $n\in\mathbb{N}^*$, a network $\mathcal{N}$ satisfies $d^*(\mathcal{N})=n$ iff $\mathcal{A}_{\mathcal{N}}$ contains a maximal alternating, or co-alternating, tree of length n. In other words, for any $n\in\mathbb{N}^*$, a network $\mathcal{N}$ satisfies $d^*(\mathcal{N})=n$ iff $\mathcal{N}$ is able to switch at most n times between visits of different types of attractors during all its possible evolutions.

The levels of transfinite degrees provide a refined classification of the collection of networks that are able to alternate *infinitely* many times between different types of attractors. Indeed, according to Theorem 6, for any ordinal α such that $\omega\leq\alpha<\omega^\omega$, a network $\mathcal{N}$ satisfies $d^*(\mathcal{N})=\alpha$ iff $\mathcal{A}_{\mathcal{N}}$ contains a maximal alternating or co-alternating tree of length α. Since $\alpha\geq\omega$, this implies that the graph of $\mathcal{A}_{\mathcal{N}}$ necessarily contains at least two cycles c_1 and c_2 such that $c_1\subsetneq c_2$ and c_1 is successful iff c_2 is non-successful. But since $c_1\subsetneq c_2$, it follows that c_1 and c_2 are both accessible one from the other in the graph of $\mathcal{A}_{\mathcal{N}}$. By the biunivocal correspondence between cycles and attractors, the network $\mathcal{N}$ contains at least the two attractors c_1 and c_2, and the accessibility between those ensures that the network is capable of alternating infinitely often between visits of c_1 and c_2 along some evolution. In fact, the collection of levels of transfinite degrees of the complete RNN hierarchy provides a refined classification of these potentially infinitely switching networks based on the intricacy of their underlying attractive structures (tree-like representation induced by the inclusion and accessibility relations between the attractors, as illustrated in Figure 2).

It can be noticed, according to the definition of alternating and co-alternating trees, that if some given Muller automaton contains either an alternating or a co-alternating tree of length α in its underlying graph, then this automaton also necessarily contains in its graph both an alternating and a co-alternating tree of length β, for all $\beta<\alpha$. Therefore, any network of the complete RNN hierarchy is capable of disclosing an attractive behaviour analogous to any other network of strictly smaller degree. In this precise sense, a network of the complete RNN hierarchy potentially contains in its structure all the possible attractive behaviours of every other networks of strictly smaller degrees. In this framework, the concept of alternation between different types of attractors corresponds to the transient trajectories between attractor basins, a concept referred to as "itinerancy" elsewhere in the literature [51,65–67,83,84].

The decidability procedure of the complete RNN hierarchy is illustrated by the Example S6 in File S6.

It is worth noting that the complete RNN hierarchy can actually be seen as a proper extension of the RNN hierarchy. Indeed, the next result shows that the networks of RNN hierarchy and the networks of the specific initial segment of length $\omega+1$ of the complete RNN hierarchy recognise precisely the same languages. In this precise sense, the RNN hierarchy consists of an initial segment of length $\omega+1$ of the complete RNN hierarchy.

Proposition 7. *Let $L\subseteq[\mathbb{B}^k]^\omega$. Then L is recognisable by some network $\mathcal{N}$ of the RNN hierarchy iff L is also recognisable by some network $\mathcal{N}'$ of the complete RNN hierarchy such that either $d^*(\mathcal{N}')<\omega$ or $d^*(\mathcal{N}')=\omega$ and $\mathcal{N}'$ contains a maximal co-alternating tree of length ω but no maximal alternating tree of length ω.*

Proof. Given any deterministic Muller automaton $\mathcal{A}$, let the degree of $\mathcal{A}$ in the Wagner hierarchy be denoted by $d_W(\mathcal{A})$. Then, the relationship between the DBA and the Wagner hierarchies ensures that L is recognisable by some deterministic Büchi automaton iff L is also recognisable by some deterministic Muller automaton $\mathcal{A}$ such that either $d_W(\mathcal{A}) < \omega$ or $d_W(\mathcal{A}) = \omega$ and $\mathcal{A}$ contains a maximal co-alternating tree of length ω but no maximal alternating tree of length ω [79]. Theorems 1 and 4 together with Proposition 6 allow to translate these results to the neural network context, and therefore lead to the conclusion.

We recall that the RNN hierarchy consists of the classification of networks whose attractors' type specifications are induced by the existence of an output layer, whereas the complete RNN hierarchy consists of the classification of networks whose attractors' type specifications are *a priori* given without any restriction at all. For any ordinal $\alpha \leq \omega$, the two notions of alternating chain and alternating tree of length α coincide. Hence, by Theorem 3 and Theorem 6, the two decidability procedures of the RNN hierarchy and the complete RNN hierarchy reduce to the very same, and the decidability procedures simply consist in computing the length of a maximal alternating or co-alternating tree contained in the underlying automata.

However, it is important to clarify the difference between the RNN hierarchy and the complete RNN hierarchy, illustrated in Figure 6. The restriction on the type specification of the attractors imposed by the existence of an output layer ensures that the networks of the RNN hierarchy will never be able to contain a maximal alternating or co-alternating tree of length strictly larger than ω in their underlying Büchi automata. Indeed, if c_1 and c_2 are two cycles in a deterministic Büchi automaton such that c_1 is successful and $c_1 \subseteq c_2$, then c_2 is necessarily also successful (since it visits the same final states as c_1 plus potentially some other ones), meaning that no meaningful cycle could ever be included in some spurious cycle in a deterministic Büchi automaton; consequently, the maximal number alternations between different type of cycles that can be found in a deterministic Büchi automaton is bounded by one – a spurious cycle included in a meaningful cycle – and therefore no alternating or co-alternating trees of length strictly larger than ω^1 will never exist in a deterministic Büchi automaton. From this observation, it follows that the degree of any network of the RNN hierarchy is bounded by ω^1, meaning that the length of the RNN hierarchy cannot exceed $\omega + 1$, whereas the length of the complete RNN hierarchy climbs up to ω^ω, as illustrated by Figure 6.

The "basal ganglia-thalamocortical network"

Neurobiological description. In order to illustrate the application of our method to a case study, we have considered one of the main systems of the brain involved in information processing, the basal ganglia-thalamocortical network. This network is formed by several parallel and segregated circuits involving different areas of the cerebral cortex, striatum, pallidum, thalamus, subthalamic nucleus and midbrain [85–94]. This network has been investigated for many years in particular in relation to disorders of the motor system and of the sleep-waking cycle. The simulations were generally performed by considering the basal ganglia-thalamocortical network as a circuit composed of several interconnected areas, each area being modeled by a network of spiking neurons, and were analysed using statistical approaches based on mean-field theory [95–106].

In the basal ganglia-thalamocortical network are several types of connections and transmitters. Based on the observation that almost all neurons of the central nervous system can be subdivided into projection neurons and interneurons, we consider the connections mediated by projection neurons, both glutamatergic excitatory projections and GABAergic inhibitory projections, as part of an information transmitting system. The local connections established by the interneurons, i.e. the connections remaining confined within a small distance from the projection neurons, are considered forming part of a regulatory system. The other connections, mainly produced by different types of projection neurons, i.e. the dopaminergic (including those from the substantia nigra pars compacta like the nigrostriatal and those from the ventromedial tegmental area), cholinergic (including those from the basal forebrain), the noradrenergic (including those from locus coeruleus), serotoninergic (including those from the dorsal raphe), histaminergic (from the tuberomamillary nucleus) and orexinergic projections (from the lateral and posterior hypothalamus) are considered forming part of a modulatory system. The three systems, information transmitting, regulatory and modulatory have an extensive pattern of reciprocal interconnectivity at various levels that is not addressed in this paper.

A characteristic of all the circuits of the basal ganglia-thalamocortical network is a combination of "open" and "closed" loops with ascending sensory afferences reaching the thalamus and the midbrain, and with descending motor efferences from the midbrain (the tectospinal tract) and the cortex (the corticospinal tract). We assume that the encoding of a large amount of the information treated by the brain is performed by recurrent patterns of activity circulating in the information transmitting system. For this reason, we focus our attention on the complexity of the dynamics that may emerge from that system. To this purpose, we present a Boolean recurrent neural network model of the information transmitting system of the basal ganglia-thalamocortical network, illustrated by Figure 7.

The pattern of connectivity corresponds to the wealth of data reported in the literature [85–94]. We assume that each brain area is formed by a neural network and that the network of brain areas corresponding to the basal ganglia-thalamocortical network can be modeled by a Boolean neural network formed by 9 nodes: superior colliculus (SC), Thalamus, thalamic reticular nucleus (NRT), Cerebral Cortex, the two functional parts (striatopallidal and the striatonigral components) of the striatum (Str), the subthalamic nucleus (STN), the external part of the pallidum (GPe), and the output nuclei of the basal ganglia formed by the GABAergic

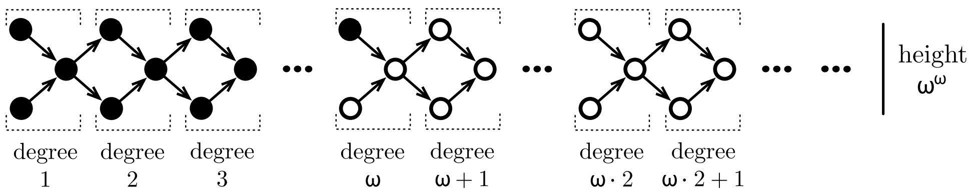

Figure 6. Comparison between the RNN and the complete RNN hierarchies. The RNN hierarchy, depicted by the sequence of blacks classes, consists of an initial segment of length $\omega + 1$ of the complete RNN hierarchy, which has height ω^ω.
doi:10.1371/journal.pone.0094204.g006

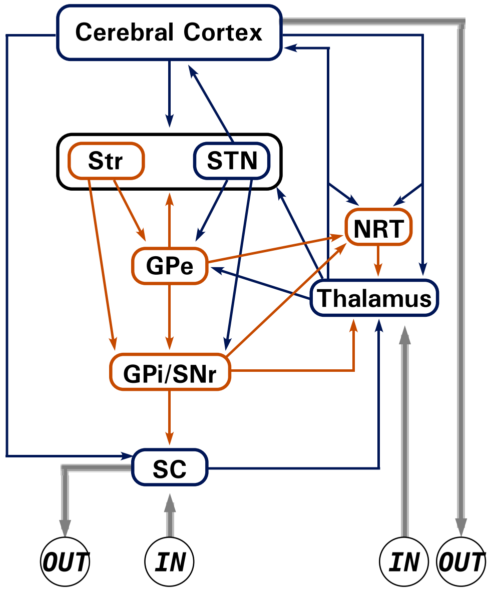

Figure 7. Model of the basal ganglia-thalamocortical network. The network is constituted of 9 different interconnected brain areas, each one represented by a single node in the Boolean neural network model: superior colliculus (SC), Thalamus, thalamic reticular nucleus (NRT), Cerebral Cortex, the striatopallidal and the striatonigral components of the striatum (Str), the subthalamic nucleus (STN), the external part of the pallidum (GPe), and the output nuclei of the basal ganglia formed by the GABAergic projection neurons of the intermediate part of the pallidum and of the substantia nigra pars reticulata (GPi/SNR). We consider also the inputs (IN) from the ascending sensory pathway and the motor outputs (OUT). The excitatory pathways are labeled in blue and the inhibitory ones in orange.
doi:10.1371/journal.pone.0094204.g007

projection neurons of the intermediate part of the pallidum and of the substantia nigra pars reticulata (GPi/SNR).

We consider the ascending sensory pathway (IN), that reaches SC and the Thalamus. SC does not send other projections to the system and sends a projection outside of the system (OUT), to the motor system. The thalamus sends excitatory connections within the system via the thalamo-pallidal, thalamo-striatal and thalamo-cortical projections. Notice that STN receives also an excitatory projection from the Thalamus. NRT receives excitatory collateral projections from both the thalamo-cortical and cortico-thalamic projections. In turn, NRT sends an inhibitory projection to the Thalamus. The Cerebral Cortex receives also an excitatory input from STN and sends corticofugal projections to the basal ganglia (striatum and STN), to the thalamus, to the midbrain and to the motor system (OUT). The only excitatory nucleus of the basal ganglia is STN, that sends projections to the Cerebral Cortex, to GPe and to GPi/SNR. In the striatum (Str) the striatopallidal neurons send inhibitory projections to GPe and the striatonigral neurons send inhibitory projections to GPi/SNR, via the so-called "direct" pathway. The pallidum (GPe) plays a paramount role because it is an inhibitory nucleus, with reciprocal connections back to the striatum and to STN and a downstream inhibitory projection to GPi/SNR via the so-called "indirect" pathway. It is interesting to notice the presence of inhibitory projections that inhibit the inhibitory nuclei within the basal ganglia, thus leading to a kind of "facilitation", but also inhibitory projections that inhibit RTN, that is a major nucleus in regulating the activity of the thalamus. The connectivity of the Boolean model of the basal ganglia-thalamocortical network is described by the adjacency matrix of the network in Table 1.

Computation of attractor-based complexity. For sake of simplicity, we consider that the two inputs to the basal ganglia-thalamocortical network (Figure 7) are reduced to 1 input node sending projections to Thalamus and SC with synaptic weight equal to 1. We reduce our neurobiological model to a Boolean recurrent neural network $\mathcal{N}$ that contains 9 activation nodes and 1 input node. Every activation node can be either active or quiet, which means $2^9 = 512$ possible states for the network $\mathcal{N}$. Every state of $\mathcal{N}$ is represented by a 9-dimensional Boolean vector describing the sequence of active and quiet activation nodes. For example, the network state (0,1,0,0,1,1,1,1,1) means that the nodes #1 (SC), #3 (RTN) and #4 (GPi/SNR) are quiet, whereas every other activation node is active.

In this section, we provide a practical illustration of our new attractor-based complexity measurement applied to the simplified model of the basal ganglia-thalamocortical network. Since the behaviour of network $\mathcal{N}$ is not determined by any designated output layer, the attractor-based complexity of $\mathcal{N}$ will be measured with respect to the complete RNN hierarchy rather than with respect to the RNN hierarchy, as described in Section "Complete RNN Hierarchy". According to these considerations, as mentioned in Theorem 6, the attractor-based complexity of network $\mathcal{N}$ relies on the graphical structure of its corresponding deterministic Muller automaton $\mathcal{A}_{\mathcal{N}}$. Hence, we shall now describe the structure of the deterministic Muller automaton $\mathcal{A}_{\mathcal{N}}$ associated to network $\mathcal{N}$.

Firstly, as mentioned in the proof of Proposition 4, the states of the Muller automaton $\mathcal{A}_{\mathcal{N}}$ correspond precisely to the states of network $\mathcal{N}$. Hence, the deterministic Muller automaton associated to the basal ganglia-thalamocortical network contains 512 states, numbered from 0 to 511. The numbering of the states is chosen such that state $(b_1,b_2,\ldots,b_9)$ is numbered by n, where n is the decimal representation of the 9-digit binary number $b_1b_2\cdots b_9$. For instance, state (1,1,0,1,0,0,0,0,1) is referred to as number 417, since 417 is the decimal representation of the binary number 110100001. Secondly, also as mentioned in the proof of Proposition 4, the transitions of the Muller automaton $\mathcal{A}_{\mathcal{N}}$ are constructed as follows: there is a transition labelled by 0 (resp. by 1) from state m to state n if and only if network $\mathcal{N}$ transits from state m to state n when it receives input 0 (resp. 1). Hence, the deterministic Muller automaton $\mathcal{A}_{\mathcal{N}}$ contains 1024 transitions (one 0-labelled and one 1-labelled outgoing transition from each of the 512 state), among which 512 are labelled by 0 and 512 are labelled by 1. For instance, in the Muller automaton $\mathcal{A}_{\mathcal{N}}$ there is a transition labelled by 1 (drawn in red in Figure 8) from state 31 to state 417 because network $\mathcal{N}$ transits from state (0,0,0,0,1,1,1,1,1) to state (1,1,0,1,0,0,0,0,1) when it receives input 1. Figure 8a illustrates the graph of the deterministic Muller automaton associated to the basal ganglia-thalamocortical network.

An analysis of the graph of the automaton $\mathcal{A}_{\mathcal{N}}$ reveals that it contains only one strongly connected component $\mathcal{C}$ given by the states 0, 31, 33, 63, 95, 127, 128, 159, 161, 191, 223, 255, 384, 417, 479, 511 and the transitions between those states, as illustrated in Figure 8b (we recall that a directed graph is called strongly connected if there is a path from every vertex of the graph to every other vertex). This strongly connected component $\mathcal{C}$ corresponds to the subgraph of $\mathcal{A}_{\mathcal{N}}$ constituted by all states reachable from the initial state 0. In other words, any state of $\mathcal{A}_{\mathcal{N}}$ outside the strongly connected component $\mathcal{C}$ cannot be reached from the initial state 0, meaning that it can never occur in the dynamics of network $\mathcal{N}$ starting from initial state 0, and hence plays no role in the attractor-based complexity of network $\mathcal{N}$. In

Table 1. The adjancency matrix of the Boolean model of the basal ganglia-thalamocortical network.

Source		**Target**								
Node	**Name**	**SC**	**Thalamus**	**RTN**	**GPi/SNr**	**STN**	**GPe**	**Str-D2**	**Str-D1**	**CCortex**
1	SC	·	1	·	·	·	·	·	·	·
2	Thalamus	·	·	1	·	1	1	1	1	1
3	RTN	·	−1	·	·	·	·	·	·	·
4	GPi/SNr	−1	−1	−1	·	·	·	·	·	·
5	STN	·	·	·	2	·	2	·	·	2
6	GPe	·	·	−1/2	−1/2	−1/2	·	−1/2	−1/2	·
7	Str-D2	·	·	·	·	·	−1	·	·	·
8	Str-D1	·	·	·	−1/2	·	−1/2	·	·	·
9	Cer. Cortex	1/2	1/2	1/2	·	1/2	·	1/2	1/2	·

doi:10.1371/journal.pone.0094204.t001

fact, the attractor-based complexity of network $\mathcal{N}$ will be precisely determined by the cyclic structure of the strongly connected component $\mathcal{C}$ of automaton $\mathcal{A}_{\mathcal{N}}$.

In order to complete the description of the Muller automaton $\mathcal{A}_{\mathcal{N}}$, it is necessary to specify its table, or in other words, to determine among all possible cycles of $\mathcal{A}_{\mathcal{N}}$ which ones are successful and which ones are non-successful. Since every cycle of $\mathcal{A}_{\mathcal{N}}$ is by definition contained in a strongly connected component of $\mathcal{C}$ and since $\mathcal{C}$ is the only strongly connected component of $\mathcal{A}_{\mathcal{N}}$, it follows that all cycles of $\mathcal{A}_{\mathcal{N}}$ are necessarily contained in $\mathcal{C}$. Therefore, the specification of the table of $\mathcal{A}_{\mathcal{N}}$ amounts to the assignment of a type specification to every cycle of the strongly connected component $\mathcal{C}$. According to the biunivocal correspondence between cycles of $\mathcal{A}_{\mathcal{N}}$ and attractors of $\mathcal{N}$, this assignment procedure consists in determining the type specification (meaningful or spurious) of all possible attractors of the network $\mathcal{N}$.

In order to assign a type specification to every cycle of the strongly connected component $\mathcal{C}$, we have computed the list of all cycles starting from every state of $\mathcal{C}$, and for each cycle, we have further computed its decomposition into constitutive cycles (cycles which do not visit the same vertex two times). The results are summarised in Table 2.

Then, we have assigned a type specification to each cycle of $\mathcal{C}$ according to the following neurobiological criteria. First, a constitutive cycle is considered to be spurious if it is characterised either by active SC and quiet Thalamus at the same time step or by a quiet GPi/SNR during the majority of the duration of the constitutive cycle. A constitutive cycle is meaningful otherwise. Second, a non-constitutive cycle is considered to be meaningful if it contains a majority of meaningful constitutive cycles, and spurious if it contains a majority of spurious constitutive cycles – and in case of it containing as much meaningful as spurious constitutive cycles, its type specification was chosen to be

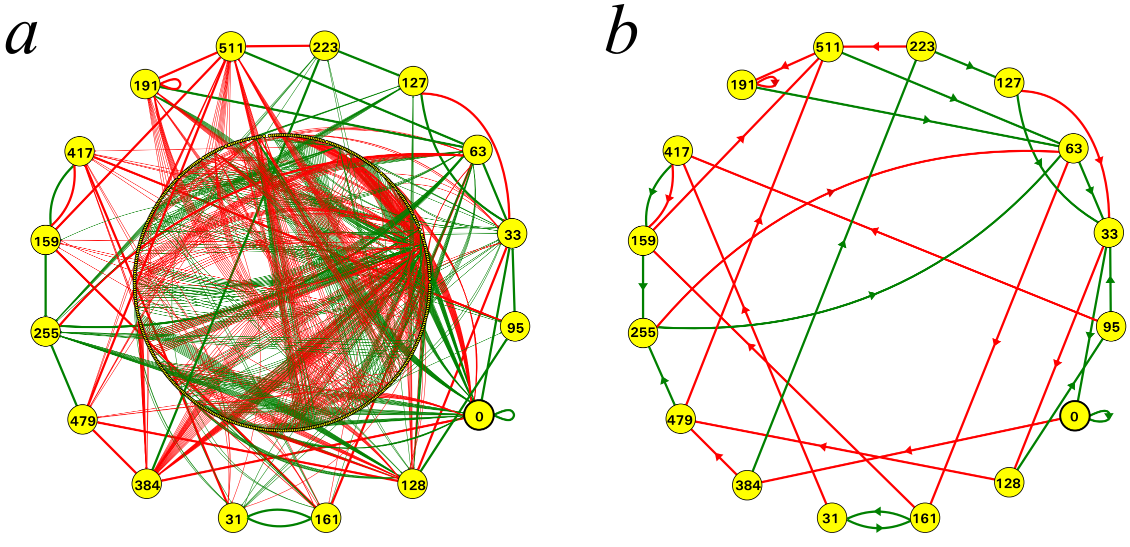

Figure 8. Deterministic Muller automaton based on the "basal ganglia-thalamocortical" network of Figure 7. a. The graph of the automaton $\mathcal{A}_{\mathcal{N}}$ associated to network $\mathcal{N}$ contains 512 states and 1024 directed transitions. The colours of the transitions represent their labels: green for label 0 and red for label 1. For sake of readability, the directions of the transitions have been removed. The states and transitions of the strongly connected component $\mathcal{C}$ of $\mathcal{A}_{\mathcal{N}}$ have been pulled out of the central graph and drawn in larger font. **b.** The graph of the strongly connected component $\mathcal{C}$ of $\mathcal{A}_{\mathcal{N}}$. Every state and transition of $\mathcal{A}_{\mathcal{N}}$ that does not belong to $\mathcal{C}$ has been erased. The directions of the transitions are indicated by the arrowheads.
doi:10.1371/journal.pone.0094204.g008

Table 2. Number of cycles and constitutive cycles found for each starting state of the strongly connected component $\mathcal{C}$.

State	# cycles	# constitutive cycles
0	68	24
31	47	20
33	87	24
63	93	21
95	39	21
127	21	17
128	63	24
159	77	22
161	72	20
191	52	19
223	43	21
255	53	17
384	67	24
417	35	20
479	48	16
511	84	21

doi:10.1371/journal.pone.0094204.t002

meaningful. In order to illustrate this procedure, let us consider for example the cycles starting from state 0. Table 2 shows that there are overall 68 cycles and 24 constitutive cycles starting from state 0. We consider here the example of one out of the 68 cycles, e.g. cycle $c=(0,0,384,223,511,191,63,33,128,95,33,0)$ (Figure 9a). This cycle can be decomposed into three constitutive cycles (Figure 9b), namely $c_{c1}=(0,0)$, $c_{c2}=(0,384,223,511,191,63,33,0)$, and $c_{c3}=(33,128,95,33)$. When state 0 receives input 0 the network dynamics evolves into the constitutive cycle c_{c1} (Figure 9c), whereas if state 0 receives input 1 the dynamics evolves into the constitutive cycle c_{c2} (Figure 9d). According to the aforementioned neurobiological criteria, the constitutive cycles c_{c1} and c_{c3} are spurious, whereas c_{c2} is meaningful, and therefore cycle c is spurious.

After the assignation of the type specification to every cycle, the attractor-based complexity of the network $\mathcal{N}$ can be explicitly computed. More precisely, according to Theorem 6, the attractor-based degree of $\mathcal{N}$ is given by the length of a maximal (co-)alternating tree contained in $\mathcal{A}_{\mathcal{N}}$. Since $\mathcal{A}_{\mathcal{N}}$ contains only one strongly connected component $\mathcal{C}$, the maximal (co-)alternating tree of $\mathcal{A}_{\mathcal{N}}$ is necessarily contained in $\mathcal{C}$. Indeed, every cycle of $\mathcal{A}_{\mathcal{N}}$ is, being a cycle, necessarily contained in a strongly connected component of $\mathcal{A}_{\mathcal{N}}$; hence in particular, every cycle of the maximal (co-)alternating tree is also contained in a strongly connected component of $\mathcal{A}_{\mathcal{N}}$; yet since $\mathcal{C}$ is the only strongly connected component of $\mathcal{A}_{\mathcal{N}}$, every cycle of the maximal (co-)alternating tree is contained in $\mathcal{C}$, meaning that the maximal (co-)alternating tree itself is contained in $\mathcal{C}$.

After an exhaustive analysis of the strongly connected component $\mathcal{C}$ and of all its cycles (Table 2) we observed no maximal alternating trees with length above ω^5. Conversely, we found 3 maximal co-alternating trees of $\mathcal{A}_{\mathcal{N}}$ with length ω^6. For sake of clarity, we describe one such maximal co-alternating tree: it consists of an alternating sequence of seven cycles included one into the other, summarised in Table 3 below and illustrated in Figure 10. Notice that there is no alternation between C_0 and C_1 because both cycles $C_0=(0,\ 0)$ and $C_1=(0,\ 384,\ 223,\ 511,\ 63,\ 33,\ 0)$ are spurious. According to these results, it follows from Theorem 6 that the attractor-based complexity of network $\mathcal{N}$ is ω^6 and that $\mathcal{N}$ is non-self-dual.

Discussion

The present work revisits and extends in light of modern automata theory the seminal studies by McCulloch and Pitts, Minsky and Kleene concerning the computational power of recurrent neural networks [1–3]. We present two novel attractor-based complexity measures for Boolean neural networks, and finally illustrate the application of our results to a model of the basal ganglia-thalamocortical network.

More precisely, we prove two computational equivalence between Boolean neural networks and Büchi and Muller automata, and deduce from these results two hierarchical classifications of Boolean recurrent neural networks based on their attractive behaviours. The hierarchical classifications are obtained by translating the Wagner classification theory from the ω-automaton to the neural network context. The first classification concerns the neural networks characterised by the specification of an output layer and the properties of the attractor dynamics associated with the activation of that output layer. In this case, the obtained hierarchical classification corresponds to a decidable pre-well ordering of width 2 and height of $\omega+1$. The second classification concerns the neural networks whose conditions on the type specifications of their attractors have been totally relaxed. In this case, the resulting hierarchy is significantly richer and consists of a decidable pre-well ordering of width 2 and height of ω^{ω}. We prove that both hierarchical classifications are decidable and provide the decidability procedures aimed at computing the degrees of the networks in the respective hierarchies. We also show that the shorter hierarchy corresponds to an initial segment of the longer one in a precise sense. The notable result is the proof that the two hierarchical classifications are directly related to the attractive properties of the neural networks. More precisely, the degrees of the Boolean neural networks in the hierarchies correspond to the ability of the networks to maximally alternate between visits of meaningful and spurious attractors along their evolutions. The two hierarchies therefore provide two novel complexity measurments of Boolean recurrent neural networks according to their attractive potentialities. These complexity measurements represents an assessment of the computational power of Boolean neural networks in terms of the significance of their attractor dynamics.

Attractor-Based Complexity Measurement

The degree of a neural network $\mathcal{N}$ in the RNN hierarchy or in the complete RNN hierarchy corresponds precisely to the length of a maximal alternating chain or alternating tree contained in the graph of its corresponding automaton $\mathcal{A}_{\mathcal{N}}$, respectively. Since alternating chains and trees are described in terms of accessibility and inclusion relations between cycles of $\mathcal{A}_{\mathcal{N}}$, and according to the biunivocal correspondence between cycles of $\mathcal{A}_{\mathcal{N}}$ and attractors of $\mathcal{N}$, it follows that the degree of a neural network $\mathcal{N}$ corresponds precisely to some intricacy relation – accessibility and inclusion – between the set of its meaningful and spurious attractors.

In order to better explain the attractor-based complexity measurement, suppose that some network $\mathcal{N}$ follows the periodic infinite evolution $e_s=[\vec{x}_0\vec{x}_1\vec{x}_0\vec{x}_2]^{\omega}$, where $\vec{x}_0,\vec{x}_1,\vec{x}_2$ are states of $\mathcal{N}$. It follows that $\mathcal{N}$ alternates infinitely often between the two cycles of states $\vec{x}_0\vec{x}_1\vec{x}_0$ and $\vec{x}_0\vec{x}_2\vec{x}_0$, or equivalently, between the two attractors $A_1=\{\vec{x}_0,\vec{x}_1\}$ and $A_2=\{\vec{x}_0,\vec{x}_2\}$. If we suppose that A_1 is

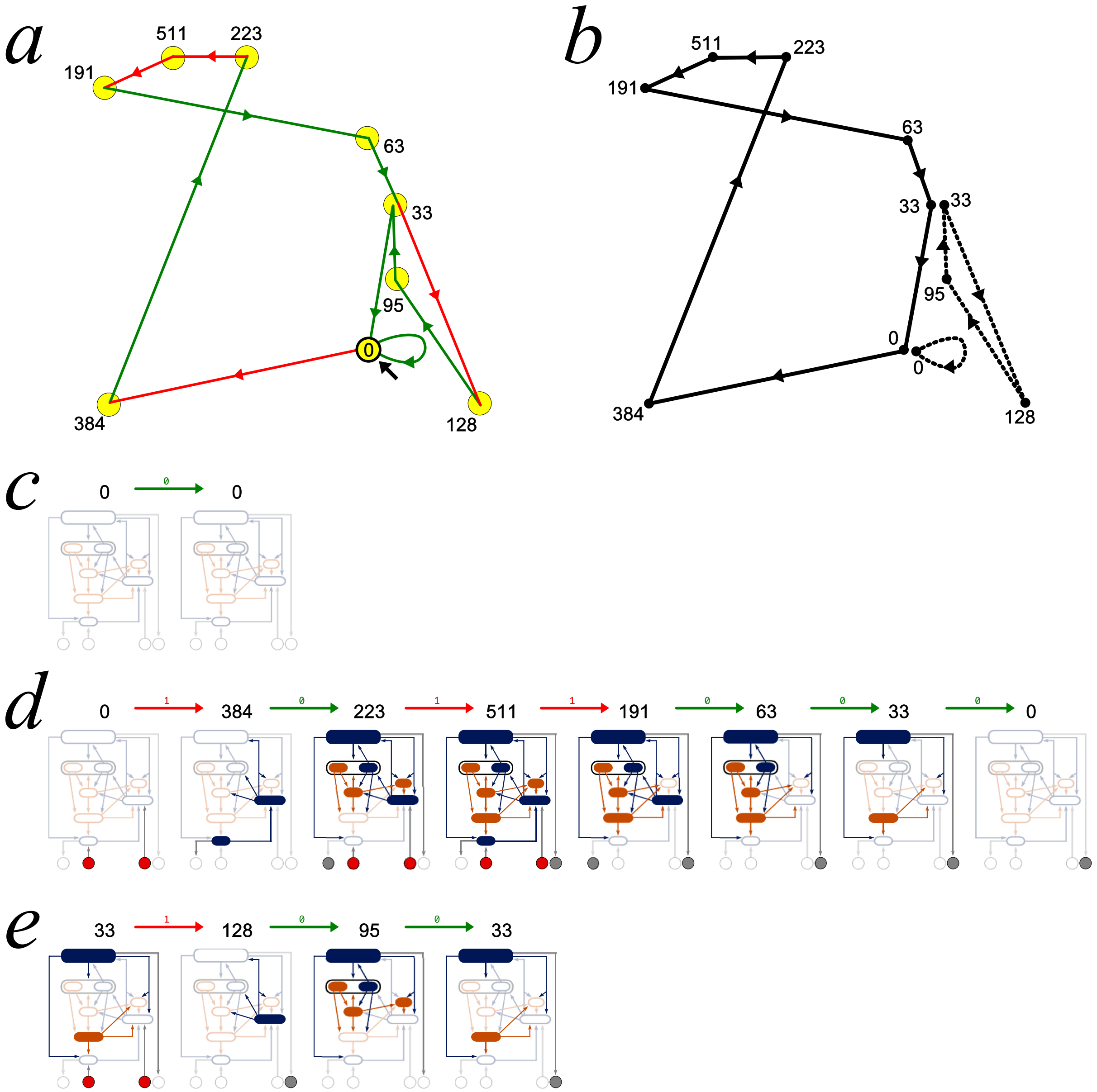

Figure 9. A cycle and its constitutive cycles. a. Among all cycles that can be observed starting from state 0 (indicated by the short arrow showing the entry point), we consider here an example, i.e. the cycle (0, 0, 384, 223, 511, 191, 63, 33, 128, 95, 33, 0). **b.** This cycle contains three constitutive cycles (0,0), (0, 384, 223, 511, 191, 63, 33, 0) and (33, 128, 95, 33) that were assigned with type specification spurious (dotted line), meaningful (solid line), and spurious (dotted line), respectively. **c.** Sequence of states with graphical representation of the corresponding activated nodes of the basal ganglia-thalamocortical network for the spurious constitutive cycle (0,0). **d.** Sequence of states and activated network areas for the meaningful constitutive cycle (0, 384, 223, 511, 191, 63, 33, 0). **e.** Sequence of states and activated network areas for the spurious constitutive cycle (33, 128, 95, 33).
doi:10.1371/journal.pone.0094204.g009

meaningful and A_2 is spurious, then $\mathcal{N}$ alternates infinitely often between a meaningful and a spurious attractor along the evolution e_s. However, note that $\mathcal{N}$ also visits infinitely often the composed attractor $A_{12} = \{\vec{x}_0, \vec{x}_1\} \cup \{\vec{x}_0, \vec{x}_2\} = \{\vec{x}_0, \vec{x}_1, \vec{x}_2\}$. Hence, if A_{12} is meaningful (resp. spurious), then $\mathcal{N}$ not only alternates infinitely often between a meaningful and a spurious attractor A_1 and A_2 respectively, but also visits infinitely often the third composed meaningful (resp. spurious) attractor A_{12}.

In fact, for any infinite evolution e_s, there always exists a unique such maximal attractor (maximal for the inclusion relation) that is visited infinitely often. Let us call this attractor the *global attractor* associated to e_s. The attractor-based complexity measurement can now be understood as follows. A network $\mathcal{N}$ is more complex than a network $\mathcal{N}'$ iff for any infinite evolution $e_{s'}$ of $\mathcal{N}'$, there exists a corresponding infinite evolution e_s of $\mathcal{N}$ that can be build "simultaneously" to $e_{s'}$ (in a precise sense described below) and such that, after infinitely many time steps, the types of global attractors visited by e_s and $e_{s'}$ are the very same. In other words, a network $\mathcal{N}$ is more complex than a network $\mathcal{N}'$ iff $\mathcal{N}$ is able to mimic step by step every possible infinite evolution of $\mathcal{N}'$ in order to finally obtain a global attractor of the same type.

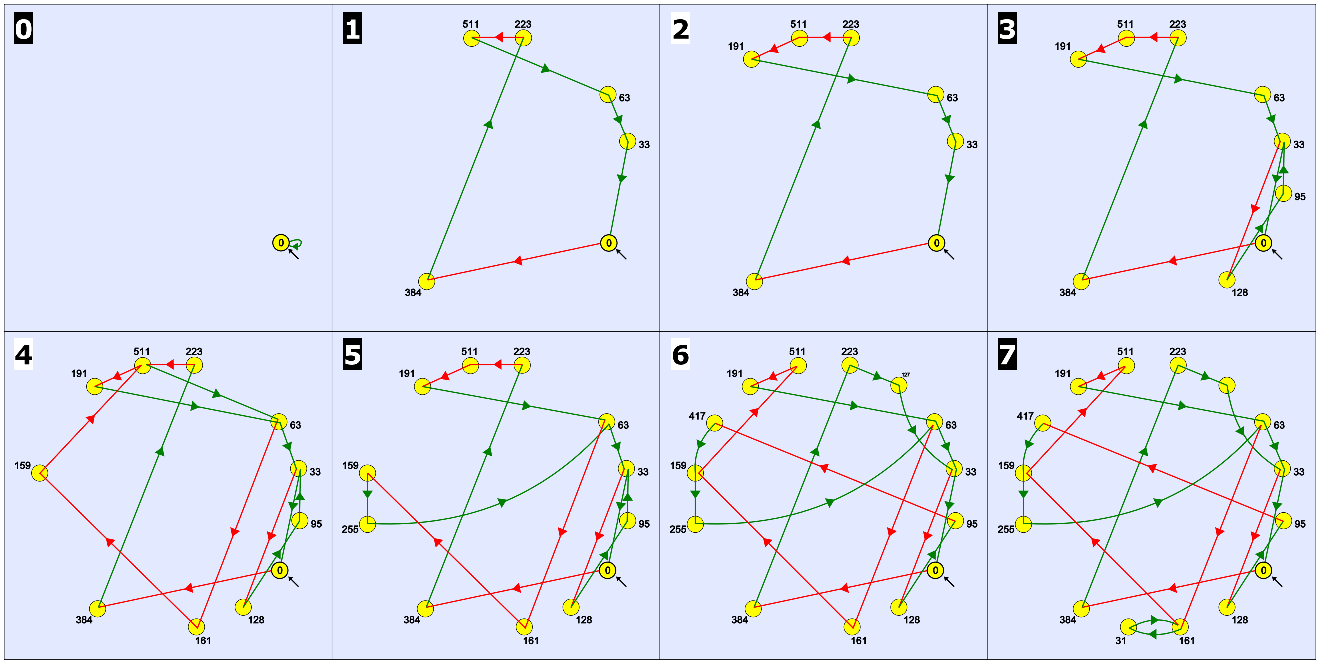

Figure 10. A maximal co-alternating tree of the deterministic Muller automaton $\mathcal{A}_{\mathcal{N}}$. Panels 0 to 7 illustrate the sequence of eight cycles $(C_0,C_1,C_2,C_3,C_4,C_5,C_6,C_7)$ one included into the next. Cycles C_0, C_1, C_3, C_5, and C_7 are spurious whereas cycles C_2, C_4, and C_6 are meaningful. The sequence of cycles $(C_1,C_2,C_3,C_4,C_5,C_6,C_7)$ compose a maximal co-alternating tree of $\mathcal{A}_{\mathcal{N}}$. This maximal co-alternating tree contains 6 alternations between spurious and meaningful cycles, and thus has a length of ω^6. Therefore, the attractor-based degree of $\mathcal{N}$ equals ω^6.
doi:10.1371/journal.pone.0094204.g010

This property can actually be precisely expressed in terms of game-theoretic considerations. Consider the game $G(\mathcal{N}_1,\mathcal{N}_2)$ between networks $\mathcal{N}_1$ and $\mathcal{N}_2$ wholes rules are the following. Both networks begin in the rest state. Network $\mathcal{N}_1$ begins the game and $\mathcal{N}_1$ and $\mathcal{N}_2$ play in turn during infinitely many rounds. At every step, $\mathcal{N}_1$ chooses a possible next state (accessible from its previous one), and $\mathcal{N}_2$ answers by either also choosing a possible next state (accessible from its previous one), or by skipping its turn. However, $\mathcal{N}_2$ is obliged to chose infinitely many next states during the game. After infinitely many time steps, $\mathcal{N}_1$ and $\mathcal{N}_2$ will have produced two infinite evolutions e_{s_1} and e_{s_2}, respectively. If the types of the global attractors of $\mathcal{N}_1$ and $\mathcal{N}_2$ are the same, $\mathcal{N}_2$ wins the game. Otherwise, $\mathcal{N}_1$ wins the game. One can prove that the attractor based complexity measures of $\mathcal{N}_1$ and $\mathcal{N}_2$ can then be expressed as follows: the degree of $\mathcal{N}_2$ is higher than that of $\mathcal{N}_1$ iff $\mathcal{N}_2$ has a winning strategy in the game $G(\mathcal{N}_1,\mathcal{N}_2)$.

In other words, a network $\mathcal{N}$ is more complex than $\mathcal{N}'$ according to our attractor-based complexity iff $\mathcal{N}$ is capable of mimicking $\mathcal{N}'$ in every of its possible attractive behaviours. Two networks $\mathcal{N}$ and $\mathcal{N}'$ are equivalent if both are capable of mimicking each other in every one of its possible attractive behaviours. Assuming that the set of all possible attractive behaviours of a network is related to its computational power, our attractor-based complexity degree therefore represents a measurement of the computational power of Boolean neural networks in terms of the significance of their attractor dynamics.

Finally, note that the degree of a neural network in the RNN hierarchy or in the complete RNN hierarchy is intimately related to the structure of this network, more precisely to its connectivity. Indeed, for any neural network $\mathcal{N}$ that would be given without any output layer or type specification of its attractors, it is possible to compute, by some graph analysis, the maximal alternating chains or alternating trees that could be contained in the graph of

Table 3. A maximal co-alternating tree of $\mathcal{N}$ of length ω^6 referred to Figure 10.

Name	State sequence	Specification type
C_1	(0,384,223,511,63,33,0)	spurious
C_2	(0,384,223,511,191,63,33,0)	meaningful
C_3	(0,384,223,511,191,63,33,128,95,33,0)	spurious
C_4	(0,384,223,511,63,161,159,511,191,63,33,128,95,33,0)	meaningful
C_5	(0,384,223,511,191,63,161,159,255,63,33,128,95,33,0)	spurious
C_6	(0,384,223,127,33,128,95,417,159,255,63,161,159,511,191,63,33,0)	meaningful
C_7	(0,384,223,127,33,128,95,417,159,255,63,61,31,161,159,511,191,63,33,0)	spurious

doi:10.1371/journal.pone.0094204.t003

its corresponding automaton $\mathcal{A}_{\mathcal{N}}$, and therefore, by theorems 3 and 6, to know the maximal degree that this network could be able to achieve in the RNN or in the complete RNN hierarchy, if the type specification of its attractors were optimally distributed. In other words, any neural network, according to its connectivity structure, contains a potential maximal degree, which is achieved only if the set of its attractors are optimally discriminated into meaningful and spurious types. Hence, based on its connectivity, a certain network could be characterised by a high potential maximal degree, but in practice, due to a very limited discrimination – i.e. non-alternation – between its spurious and meaningful attractors, it will exhibit a low degree of network complexity.

Significance of measuring network complexity

In an application of our network complexity measurement to a model of a real brain circuit, we demonstrated that, under specific assumptions of connectivity and dynamics, the basal ganglia-thalamocortical network can be modeled by a network of degree ω^6. Why is it so interesting to know this degree? What kind of increased understanding of that network do we gain from that? The degree of network complexity for a given network is important to be determined if we want to assess the computational power that can be achieved by that network. In other words, the degree of network complexity is a functional characteristic of a given network.

For example, a model of the basal ganglia-thalamocortical network with a complexity of degree ω^6 is able to perform all possible computations made by a model of the same network with a complexity of degree ω^4 and many more computations in addition. If we were able to associate certain functional states of cognitive relevance (or certain pathological conditions of clinical relevance, respectively) to an increase (or to a decrease, respectively) in network complexity, we would certainly gain a better insight into the role and the factors that modulate the operations executed by certain brain circuits.

Then, how and why the network complexity of a model of the basal ganglia-thalamocortical network could vary? The degree of complexity of a network is upper bounded by its potential maximal degree. In the next section, we discuss how control parameters can affect network dynamics and eventually its complexity degree.

Control parameters of network dynamics

The central hypothesis for brain attractors is that, once activated by appropriate activity, the network behaviour is maintained by continuous reentry of activity, thus generating a high incidence of repeating firing patterns associated with underlying attractors [37,38]. The question whether the attractors revealed by certain patterns of activity are spurious or meaningful cannot be answered easily. Certain patterns may repeat above chance and occur transiently during the evolution of a network [23,55] and during the transient inactivation of part of the newtwork, as shown experimentally with thalamic firing patterns during reversible inactivation of the cerebral cortex [60,107]. On the other hand, patterns and attractors *per se* may reveal an epiphenomenon or a byproduct of the network dynamics, thus being classified as spurious. However, changing conditions and association of attractors into higher-order attractors may turn a spurious into a meaningful type, and vice versa. For this reason, in the present paper, we have emphasised the importance of the specification types of the constitutive cycles and how these affect the specification type of a cycle.

The measurements of networks complexities refer to the possibility of networks' dynamics to maximally alternate between attractors of different types along their evolutions. This is interesting for an overall assessment of the properties of a network because it associates the computational power of that network with the significance of their attractor dynamics.

The excitatory/inhibitory balance in a neural network is the major factor affecting the dynamics of its activity [38,109–111]. The activity of the basal ganglia-thalamocortical network is modulated by a complex set of brain structures, including the dopaminergic (including those from the substantia nigra pars compacta like the nigrostriatal and those from the ventromedial tegmental area), cholinergic (including those from the basal forebrain), the noradrenergic (including those from locus coeruleus), serotoninergic (including those from the dorsal raphe), histaminergic (from the tuberomamillary nucleus) and orexinergic nuclei (from the lateral and posterior hypothalamus) [103,112–115]. These neuromodulators affect, among other parameters, the synaptic kinetics (i.e., the decay time of the synaptic interaction) and the cellular excitability, thus producing stable or unstable spatiotemporally organised modes of activity and rapid state switches [69,111,116–119]. The effect of cholinergic modulation exerted by the basal forebrain is particularly noticeable to this aspect [120–122].

The possible different dynamics of a given network can be represented by an equilibrium surface where each point is determined by a network complexity associated with two (in the simplest abstraction) independent variables. Such a situation is illustrated in Figure 11 by the cusp catastrophe of the Catastrophe theory [108]. In our example, the two control parameters are the excitability and the synaptic kinetics. Depending on the ranges of the parameters that control the network dynamics, the network complexity may remain identical or only slightly modified, in which case we refer to a "smooth" path on the network dynamics surface. In other cases, small changes in the parameter values may provoke rapid state switches corresponding to "sudden" changes in network complexity (e.g., see [111]).

The network dynamics surface has a singularity represented by a fold (or Riemann-Hugoniot cusp) in it. A bifurcation set is defined by the thresholds where sudden changes can occur, depending on the initial conditions, by projecting the cusp onto the control surface. The network complexity, as defined in this study, depends on the maximal (co-)alternation between spurious and meaningful attractors. In the network dynamics surface, the edge toward the fold (point A, in Figure 11) is the starting point of separation between two surfaces. One surface is the top sheet representing network dynamics with a high degree of complexity because of the presence of deterministic chaos enabling the possibility to increase the (co-)alternation by mean of chaotic itinerancy (point B, in Figure 11) [66,67,69]. The other surface is the bottom sheet reflecting the dominance of stochastic dynamics, hence absence of alternation (point C, in Figure 11). Hence, as the network dynamics moves out from the edge near the fold the dynamics is diverging and forced to move toward one of the two opposing behaviours. The path that will be followed by the dynamics depends on the values of the control parameters defining the state of the neural network just prior to reaching the fold. Sudden transitions are accounted for at the edges of the fold, for example as the stochastic dynamics moves along the surface toward the pleat, at some point a small change in control parameters may cause a sudden shift such that, after a long interval without cyclic activity, quasi-random activity develops into quasi-attractors and long cycles may suddenly appear containing many constitutive cycles and many alternations between spurious

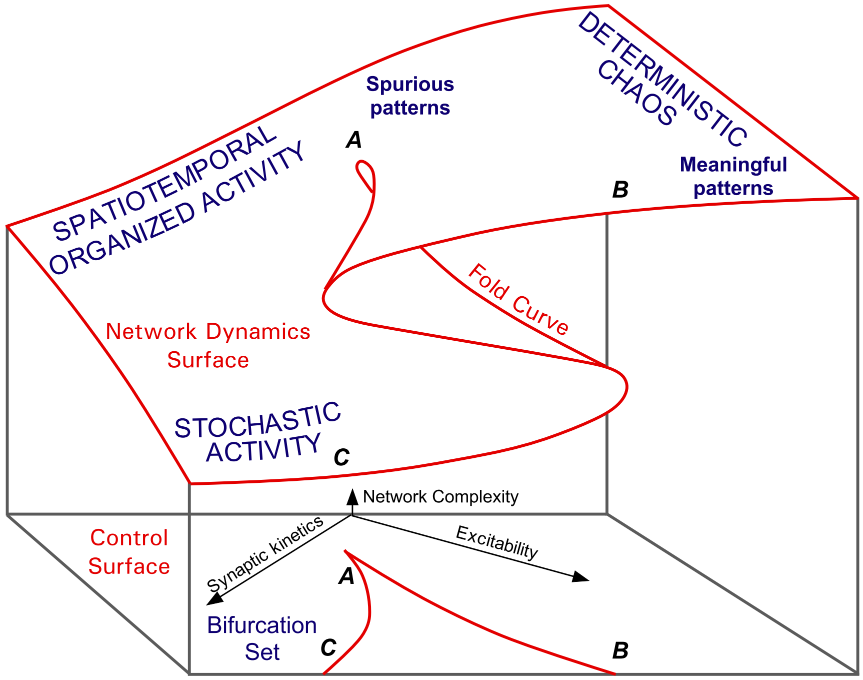

Figure 11. Cusp catastrophe model. We consider an example of network dynamics controlled by two independent parameters, the synaptic kinetics and the cell excitability. Divergent behaviour is accounted for since as the dynamics moves out from the edge (point A) toward the fold, which is the starting point of separation between an upper and lower limbs, the network dynamics is forced to move towards one of the two opposing behaviours: either point B for network dynamics dominated by deterministic chaos and chaotic itinerancy, or point C for network dynamics dominated by stochastic activity.
doi:10.1371/journal.pone.0094204.g011

and meaningful attractors, e.g. tuning thalamic activity by corticofugal activity [123,124].

Conclusion

The present work can be extended in at least three directions. First, it is expected to study the computational and dynamical complexity of neural networks induced by other mathematical bio-inspired criteria. Indeed, the approach followed in this paper provides a hierarchical classification of neural networks according to the topological complexity of their underlying neural languages, and subsequently, according to the complexity of their attractive behaviours. However, it remains to be clarified how this natural mathematical criterion could be translated into the real biological complexity of the networks. Other complexity measures might bring further insights to the global understanding of brain information processing.

Secondly, it is envisioned to describe the computational power of more biologically oriented neuronal models. For instance, first-order recurrent neural networks provided with some simple spike-timing dependent plasticity (STDP) rule could be of interest [48,125–128]. Also, neural networks equipped with more complex activation function or dynamical equations governing the membrane dynamics could be relevant [129]. Important preliminary steps in this direction were made by providing a description of the computational capabilities of static/evolving rational-weighted/analog recurrent neural networks involved in a classical as well as in a memory active and interactive paradigm of computation [6,11,27,31–33].

The third and maybe most important extension of our study is oriented towards the application of our new attractor-based complexity measurement to other examples of real neural networks, and to studying the effect of modulatory projections in controlling the network complexity. Indeed, the parameters that control neural dynamics (e.g., excitability and synaptic kinetics) are driven by so-called modulatory projections, such as the cholinergic and serotoninergic projections.

Finally, we believe that the theoretical approach to the computational power of neural models might ultimately bring further insight to the understanding of the intrinsic natures of both biological as well as artificial intelligences. On the one hand, the study of the computational and dynamical capabilities of brain-like models might improve the understanding of the biological features that are most relevant to brain information processing. On the other hand, foundational approaches to alternative models of computation might lead in the long term not only to relevant theoretical considerations [130,131], but also to practical applications.

Supporting Information

File S1 Example S1, Description of a deterministic Büchi automaton, and illustration of the concept of an alternating chain. **Figure S1, A deterministic Büchi automaton $\mathcal{A}$.** The nodes and edges correspond to the states and transitions of $\mathcal{A}$, respectively. The node i corresponds to the initial state, as indicated by the short input arrow. The double-circled red nodes correspond to the final states of $\mathcal{A}$. The Büchi automaton $\mathcal{A}$ contains a maximal alternating chain of length 2, and a maximal co-alternating chain of length 2 also.
(ZIP)

File S2 Example S2, Description of a deterministic Muller automaton, and illustration of the concept of an alternating tree. **Figure S2, A Muller automaton $\mathcal{A}$.** The underlying alphabet of $\mathcal{A}$ is $\{a,b,c,d,e\}$. The table $\mathcal{T} \subseteq \mathcal{P}(Q)$ represents the set of cycles

of $\mathcal{A}$ that are successful. All other cycles of $\mathcal{A}$ are by definition non-successful. The successful and non-successful cycles are denoted in green and red, respectively. This Muller automaton $\mathcal{A}$ contains a maximal alternating tree of length $\omega^1 \cdot 3 + \omega^0 \cdot 2$.
(ZIP)

File S3 Example S3, Illustration of the translation procedures described in Propositions 1 and 2. **Figure S3, Panels a, b.** Translation from a neural network to its corresponding deterministic Büchi automaton. **a.** The neural network $\mathcal{N}$ of Figure 1 provided with an additional specification of an output layer $V = \{x_3\}$ denoted in red and double-circled. **b.** The deterministic Büchi automaton $\mathcal{A}_{\mathcal{N}}$ corresponding to the neural network $\mathcal{N}$ of panel *a*. The final states are denoted in red and double-circled, and the active status of the output layer, namely cell x_3, is emphasised by a bold red 1. **Panels c, d.** Translation from a deterministic Büchi automaton to its corresponding neural network. **c.** A deterministic Büchi automaton $\mathcal{A}$ with three states. The initial state q_1 is denoted with an incoming edge. The final state q_3 is emphasised in red and double-circled. **d.** The network $\mathcal{N}_{\mathcal{A}}$ corresponding to the Büchi automaton $\mathcal{A}$. The output layer is represented by the cell x_5, denoted in red and double-circled. The background activities are labeled in blue.
(ZIP)

File S4 Example S4, Illustration of the decidability procedure of the RNN hierarchy.
(ZIP)

File S5 Example S5, Illustration of the translation procedures described in Propositions 4 and 5. **Figure S4, Panels a, b.** Translation from a Boolean neural network provided with a type specification of its attractors to its corresponding deterministic Muller automaton. **a.** A neural network $\mathcal{N}$ provided with an additional type specification of each of its attractors. In this case, $\mathcal{N}$ contains only one meaningful attractor determined by the following set of states $\{(0,0,0)^T,(1,0,0)^T,(0,1,1)^T\}$; all other ones are considered as spurious. **b.** The deterministic Muller automaton $\mathcal{A}_{\mathcal{N}}$ corresponding to the neural network $\mathcal{N}$ of panel *a*. Automaton $\mathcal{A}_{\mathcal{N}}$ works over alphabet $\mathbb{B}^2$, contains six states, and possesses in its table $\mathcal{T}$ the sole cycle $\{(0,0,0)^T,(1,0,0)^T,(0,1,1)^T\}$, which corresponds to the sole meaningful attractor of $\mathcal{N}$. **Panels c, d.** Translation from a deterministic Muller automaton to its corresponding Boolean neural network provided with a type specification of its attractors. **c.** A deterministic Muller automaton $\mathcal{A}$. The automaton works over alphabet $\mathbb{B}^1$, has three states, and possesses the two successful cycles $\{q_2\}$ and $\{q_3\}$, as mentioned by its table $\mathcal{T} = \{\{q_2\},\{q_3\}\}$. **d.** The neural network $\mathcal{N}_{\mathcal{A}}$ corresponding to the Muller automaton $\mathcal{A}$ of panel *c*. The network $\mathcal{N}_{\mathcal{A}}$ contains two letter cells, one delay cell, and three state cells to simulate the two possible inputs and three states of automaton $\mathcal{A}$. It has only two meaningful attractors corresponding to the two successful cycles of automaton $\mathcal{A}$.
(ZIP)

File S6 Example S6, Illustration of the decidability procedure of the complete RNN hierarchy.
(ZIP)

Acknowledgments

The authors thank all the reviewers for their comments, and in particular the last reviewer for his suggestions concerning the proof of Proposition 2.

Author Contributions

Conceived and designed the experiments: JC AV. Performed the experiments: JC AV. Analyzed the data: JC AV. Contributed reagents/materials/analysis tools: JC AV. Wrote the paper: JC AV.

References

1. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysic 5: 115–133.
2. Kleene SC (1956) Representation of events in nerve nets and finite automata. In: Automata Studies, Princeton, N. J.: Princeton University Press, volume 34 of *Annals of Mathematics Studies*. pp. 3–42.
3. Minsky ML (1967) Computation: finite and infinite machines. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.
4. Kremer SC (1995) On the computational power of elman-style recurrent networks. Neural Networks, IEEE Transactions on 6: 1000–1004.
5. Sperduti A (1997) On the computational power of recurrent neural networks for structures. Neural Netw 10: 395–400.
6. Siegelmann HT, Sontag ED (1995) On the computational power of neural nets. J Comput Syst Sci 50: 132–150.
7. Hyötyniemi H (1996) Turing machines are recurrent neural networks. In: Proceedings of STeP'96. Finnish Artificial Intelligence Society, pp. 13–24.
8. Neto JaPG, Siegelmann HT, Costa JF, Araujo CPS (1997) Turing universality of neural nets (revisited). In: EUROCAST '97: Proceedings of the A Selection of Papers from the 6th International Workshop on Computer Aided Systems Theory. London, UK: Springer-Verlag, pp. 361–366.
9. Kilian J, Siegelmann HT (1996) The dynamic universality of sigmoidal neural networks. Inf Comput 128: 48–56.
10. Neumann Jv (1958) The computer and the brain. New Haven, CT, USA: Yale University Press.
11. Siegelmann HT, Sontag ED (1994) Analog computation via neural networks. Theor Comput Sci 131: 331–360.
12. Balcázar JL, Gavaldà R, Siegelmann HT (1997) Computational power of neural networks: a characterization in terms of kolmogorov complexity. IEEE Transactions on Information Theory 43: 1175–1183.
13. Maass W, Orponen P (1998) On the effect of analog noise in discrete-time analog computations. Neural Comput 10: 1071–1095.
14. Maass W, Sontag ED (1999) Analog neural nets with gaussian or other common noise distributions cannot recognize arbitary regular languages. Neural Comput 11: 771–782.
15. Fogel DB, Fogel LJ, Porto V (1990) Evolving neural networks. Biological Cybernetics 63: 487–493.
16. Whitley D, Dominic S, Das R, Anderson CW (1993) Genetic reinforcement learning for neurocontrol problems. Machine Learning 13: 259–284.
17. Moriarty DE, Miikkulainen R (1997) Forming neural networks through efficient and adaptive coevolution. Evolutionary Computation 5: 373–399.
18. Yao X, Liu Y (1997) A new evolutionary system for evolving artificial neural networks. Trans Neur Netw 8: 694–713.
19. Angeline PJ, Saunders GM, Pollack JB (1994) An evolutionary algorithm that constructs recurrent neural networks. Neural Networks, IEEE Transactions on 5: 54–65. Angeline1994pp54
20. Chechik G, Meilijson I, Ruppin E (1999) Neuronal regulation: A mechanism for synaptic pruning during brain maturation. Neural Comput 11: 2061–2080.
21. Iglesias J, Eriksson J, Pardo B, Tomassini T, Villa A (2005) Emergence of oriented cell assemblies associated with spike-timing-dependent plasticity. Lecture Notes in Computer Science 3696: 127–132.
22. Chao TC, Chen CM (2005) Learning-induced synchronization and plasticity of a developing neural network. Journal of Computational Neuroscience 19: 311–324.
23. Iglesias J, Villa AEP (2010) Recurrent spatiotemporal firing patterns in large spiking neural networks with ontogenetic and epigenetic processes. J Physiol Paris 104: 137–146.
24. Perrig S, Iglesias J, Shaposhnyk V, Chibirova O, Dutoit P, et al. (2010) Functional interactions in hierarchically organized neural networks studied with spatiotemporal firing patterns and phase-coupling frequencies. Chin J Physiol 53: 382–395.
25. Tetzlaff C, Okujeni S, Egert U, Wörgötter F, Butz M (2010) Self-organized criticality in developing neuronal networks. PLoS computational biology 6: e1001013.
26. Shaposhnyk V, Villa AEP (2012) Reciprocal projections in hierarchically organized evolvable neural circuits affect EEG-like signals. Brain Res 1434: 266–276.
27. Cabessa J, Siegelmann HT (2011) Evolving recurrent neural networks are super-turing. In: IJCNN. IEEE, pp. 3200–3206.
28. Turing AM (1936) On computable numbers, with an application to the Entscheidungsproblem. Proc London Math Soc 2: 230–265.
29. van Leeuwen J, Wiedermann J (2008) How we think of computing today. In: Beckmann A, Dimitracopoulos C, Löwe B, editors, Logic and Theory of Algorithms, Springer Berlin/Heidelberg, volume 5028 of *LNCS*. pp. 579–593.
30. Goldin D, Smolka SA, Wegner P (2006) Interactive Computation: The New Paradigm. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

31. Cabessa J, Villa AEP (2012) The expressive power of analog recurrent neural networks on infinite input streams. Theor Comput Sci 436: 23–34.
32. Cabessa J, Siegelmann HT (2012) The computational power of interactive recurrent neural networks. Neural Computation 24: 996–1019.
33. Cabessa J, Villa AEP (2013) The super-turing computational power of interactive evolving recurrent neural networks. In: Mladenov V, Koprinkova-Hristova PD, Palm G, Villa AEP, Appollini B, et al., editors, ICANN. Springer, volume 8131 of *Lecture Notes in Computer Science*, pp. 58–65.
34. Büchi JR (1966) Symposium on decision problems: On a decision method in restricted second order arithmetic. Studies in Logic and the Foundations of Mathematics 44: 1–11.
35. Wagner K (1979) On ω-regular sets. Inform and Control 43: 123–177.
36. Kauffman SA (1993) The origins of order: Self-organization and selection in evolution. New York: Oxford University Press.
37. Abeles M (1991) Corticonics: Neural Circuits of the Cerebral Cortex. Cambridge University Press, first edition.
38. Amit DJ (1992) Modeling brain function: The world of attractor neural networks. Cambridge University Press.
39. Little WA (1974) The existence of persistent states in the brain. Mathematical biosciences 19: 101–120.
40. Little WA, Shaw GL (1978) Analytical study of the memory storage capacity of a neural network. Mathematical biosciences 39: 281–290.
41. Seung HS (1998) Learning continuous attractors in recurrent networks. In: Advances in Neural Information Processing Systems. MIT Press, pp. 654–660.
42. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences 79: 2554–2558.
43. Amit DJ, Tsodyks MV (1991) Quantitative study of attractor neural network retrieving at low spike rates: I. substrate–spikes, rates and neuronal gain. Network: Computation in Neural Systems 2: 259–273.
44. Coolen T, Sherrington D (1993) Dynamics of Attractor Neural Networks. In: Taylor J, editor, Mathematical Approaches to Neural Networks, Elsevier, volume 51 of *North-Holland Mathematical Library*. pp. 293–306. doi:http://dx.doi.org/10.1016/S0924-6509(08)70041-2. Available: http://www.sciencedirect.com/science/article/pii/S0924650908700412.
45. Eliasmith C (2005) A unified approach to building and controlling spiking attractor networks. Neural Comput 17: 1276–1314.
46. Knierim JJ, Zhang K (2012) Attractor dynamics of spatially correlated neural activity in the limbic system. Annu Rev Neurosci 35: 267–285.
47. Braitenberg V, Schüz A (1998) Cortex: Statistics and Geometry of Neuronal Connectivity. Berlin, Germany: Springer, 249 pp. ISBN: 3-540-63816-4.
48. Iglesias J, Eriksson J, Grize F, Tomassini M, Villa AE (2005) Dynamics of pruning in simulated large-scale spiking neural networks. BioSystems 79: 11–20.
49. Iglesias J, Villa AEP (2008) Emergence of preferred firing sequences in large spiking neural networks during simulated neuronal development. Int J Neural Syst 18: 267–277.
50. Abeles M, Gerstein GL (1988) Detecting spatiotemporal firing patterns among simultaneously recorded single neurons. J Neurophysiol 60: 909–924.
51. Villa AEP (2000) Empirical Evidence about Temporal Structure in Multi-unit Recordings. In: Miller R, editor, Time and the brain, Amsterdam, The Netherlands: Harwood Academic, volume 3 of *Conceptual Advances in Brain Research*, chapter 1. pp. 1–51.
52. Tetko IV, Villa AEP (2001) A pattern grouping algorithm for analysis of spatiotemporal patterns in neuronal spike trains. 1. Detection of repeated patterns. J Neurosci Meth 105: 1–14.
53. Villa AEP, Abeles M (1990) Evidence for spatiotemporal firing patterns within the auditory thalamus of the cat. Brain Res 509: 325–327.
54. Tetko IV, Villa AEP (1997) Fast combinatorial methods to estimate the probability of complex temporal patterns of spikes. Biol Cybern 76: 397–408.
55. Iglesias J, Villa AEP (2007) Effect of stimulus-driven pruning on the detection of spatiotemporal patterns of activity in large neural networks. BioSystems 89: 287–293.
56. Iglesias J, Chibirova O, Villa A (2007) Nonlinear dynamics emerging in large scale neural networks with ontogenetic and epigenetic processes. Lecture Notes in Computer Science 4668: 579–588.
57. Abeles M, Bergman H, Margalit E, Vaadia E (1993) Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. J Neurophysiol 70: 1629–1638.
58. Prut Y, Vaadia E, Bergman H, Slovin H, Abeles M (1998) Spatiotemporal structure of cortical activity: Properties and behavioral relevance. J Neurophysiol 79: 2857–2874.
59. Villa AEP, Tetko IV, Hyland B, Najem A (1999) Spatiotemporal activity patterns of rat cortical neurons predict responses in a conditioned task. Proc Natl Acad Sci U S A 96: 1106–1111.
60. Tetko IV, Villa AE (2001) A pattern grouping algorithm for analysis of spatiotemporal patterns in neuronal spike trains. 2. application to simultaneous single unit recordings. J Neurosci Meth 105: 15–24.
61. Shmiel T, Drori R, Shmiel O, Ben-Shaul Y, Nadasdy Z, et al. (2005) Neurons of the cerebral cortex exhibit precise interspike timing in correspondence to behavior. Proc Natl Acad Sci U S A 102: 18655–18657.
62. Amari SI (1975) Homogeneous nets of neuron-like elements. Biol Cybern 17: 211–220.
63. Skarda CA, Freeman WJ (1987) How brains make chaos in order to make sense of the world. Behavioral and brain sciences 10: 161–195.
64. Freeman WJ (2003) A neurobiological theory of meaning in perception Part I: Information and meaning in nonconvergent and nonlocal brain dynamics. International Journal of Bifurcation and Chaos 13: 2493–2511.
65. Freeman W (1975) Mass action in the nervous system. Academic Press.
66. Tsuda I, Koerner E, Shimizu H (1987) Memory dynamics in asynchronous neural networks. Prog Th Phys 78: 51–71.
67. Freeman W (1990) On the problem of anomalous dispersion in chaoto-chaotic phase transitions of neural masses, and its significance for the management of perceptual information in brains. In: Haken H, Stadler M, editors, Synergetics of Cognition, Springer Berlin Heidelberg, volume 45 of *Springer Series in Synergetics*. pp. 126–143.
68. Tsuda I (2001) Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems. Behav Brain Sci 24: 793–810.
69. Segundo JP (2003) Nonlinear dynamics of point process systems and data. International Journal of Bifurcation and Chaos 13: 2035–2116.
70. Fujii H, Tsuda I (2004) Neocortical gap junction-coupled interneuron systems may induce chaotic behavior itinerant among quasi-attractors exhibiting transient synchrony. Neurocomputing 58: 151–157.
71. Hopfield JJ, Feinstein DI, Palmer RG (1983) 'Unlearning' has a stabilizing effect in collective memories. Nature 304: 158–159.
72. Watta PB, Wang K, Hassoun MH (1997) Recurrent neural nets as dynamical boolean systems with application to associative memory. IEEE Trans Neural Netw 8: 1268–1280.
73. Amit DJ, Treves A (1989) Associative memory neural network with low temporal spiking rates. Proc Natl Acad Sci U S A 86: 7871–7875.
74. Griniasty M, Tsodyks MV, Amit DJ (1993) Conversion of temporal correlations between stimuli to spatial correlations between attractors. Neural Computation 5: 1–17.
75. Nara S, Davis P, Totsuji H (1993) Memory search using complex dynamics in a recurrent neural network model. Neural Networks 6: 963–973.
76. Sandberg A, Lansner A, Petersson KM, Ekeberg O (2002) A bayesian attractor network with incremental learning. Network 13: 179–194.
77. Knoblauch A (2011) Neural associative memory with optimal bayesian learning. Neural Computation 23: 1393–1451.
78. Wadge WW (1983) Reducibility and determinateness on the Baire space. Ph.D. thesis, University of California, Berkeley.
79. Perrin D, Pin JE (2004) Infinite Words, volume 141 of *Pure and Applied Mathematics*. Elsevier. ISBN 0-12-532111-2.
80. McNaughton R (1966) Testing and generating infinite sequences by a finite automaton. Information and control 9: 521–530.
81. Piterman N (2007) From nondeterministic büchi and streett automata to deterministic parity automata. Logical Methods in Computer Science 3: 1–21.
82. Selivanov VL (1998) Fine hierarchy of regular omega-languages. Theor Comput Sci 191: 37–59.
83. Tsuda I (1991) Chaotic itinerancy as a dynamical basis of hermeneutics of brain and mind. World Futures 32: 167–185.
84. Kaneko K, Tsuda I (2003) Chaotic itinerancy. Chaos 13: 926–936.
85. Alexander GE, Crutcher MD (1990) Functional architecture of basal ganglia circuits: neural substrates of parallel processing. Trends Neurosci 13: 266–271.
86. Hoover JE, Strick PL (1993) Multiple output channels in the basal ganglia. Science 259: 819–821.
87. Asanuma C (1994) GABAergic and pallidal terminals in the thalamic reticular nucleus of squirrel monkeys. Exp Brain Res 101: 439–451.
88. Groenewegen HJ, Galis-de Graaf Y, Smeets WJ (1999) Integration and segregation of limbic cortico-striatal loops at the thalamic level: an experimental tracing study in rats. J Chem Neuroanat 16: 167–185.
89. Yasukawa T, Kita T, Xue Y, Kita H (2004) Rat intralaminar thalamic nuclei projections to the globus pallidus: a biotinylated dextran amine anterograde tracing study. J Comp Neurol 471: 153–167.
90. Cebrián C, Parent A, Prensa L (2005) Patterns of axonal branching of neurons of the substantia nigra pars reticulata and pars lateralis in the rat. J Comp Neurol 492: 349–369.
91. Degos B, Deniau JM, Le Cam J, Mailly P, Maurice N (2008) Evidence for a direct subthalamo-cortical loop circuit in the rat. Eur J Neurosci 27: 2599–2610.
92. Smith Y, Raju D, Nanda B, Pare JF, Galvan A, et al. (2009) The thalamostriatal systems: anatomical and functional organization in normal and parkinsonian states. Brain Res Bull 78: 60–68.
93. Gandhi NJ, Katnani HA (2011) Motor functions of the superior colliculus. Annu Rev Neurosci 34: 205–231. Gandhi2011pp205
94. Krauzlis RJ, Lovejoy LP, Zénon A (2013) Superior colliculus and visual spatial attention. Annu Rev Neurosci 36: 165–182.
95. Terman D, Rubin JE, Yew AC, Wilson CJ (2002) Activity patterns in a model for the subthalamopallidal network of the basal ganglia. J Neurosci 22: 2963–2976.
96. Nakahara H, Amari Si S, Hikosaka O (2002) Self-organization in the basal ganglia with modulation of reinforcement signals. Neural Comput 14: 819–844.
97. Rubin JE, Terman D (2004) High frequency stimulation of the subthalamic nucleus eliminates pathological thalamic rhythmicity in a computational model. J Comput Neurosci 16: 211–235.

98. Jones BE (2005) From waking to sleeping: neuronal and chemical substrates. Trends Pharmacol Sci 26: 578–586.
99. Leblois A, Boraud T, Meissner W, Bergman H, Hansel D (2006) Competition between feedback loops underlies normal and pathological dynamics in the basal ganglia. J Neurosci 26: 3567–3583.
100. Silkis I (2007) A hypothetical role of cortico-basal ganglia-thalamocortical loops in visual processing. Biosystems 89: 227–235.
101. Tsujino N, Sakurai T (2009) Orexin/hypocretin: a neuropeptide at the interface of sleep, energy homeostasis, and reward system. Pharmacol Rev 61: 162–176.
102. van Albada SJ, Robinson PA (2009) Mean-field modeling of the basal ganglia-thalamocortical system. I Firing rates in healthy and parkinsonian states. J Theor Biol 257: 642–663.
103. Reinoso-Suárez F, De Andrés I, Garzón M (2011) The Sleep–Wakefulness Cycle, volume 208 of *Advances in Anatomy, Embryology and Cell Biology*. Berlin Heidelberg: Springer, 1–128 pp.
104. Meijer HG, Krupa M, Cagnan H, Lourens MA, Heida T, et al. (2011) From Parkinsonian thalamic activity to restoring thalamic relay using deep brain stimulation: new insights from computational modeling. J Neural Eng 8: 066005–066005.
105. Kerr CC, Neymotin SA, Chadderdon GL, Fietkiewicz CT, Francis JT, et al. (2012) Electrostimulation as a prosthesis for repair of information flow in a computer model of neocortex. Neural Systems and Rehabilitation Engineering, IEEE Transactions on 20: 153–160.
106. Guthrie M, Leblois A, Garenne A, Boraud T (2013) Interaction between cognitive and motor cortico-basal ganglia loops during decision making: a computational study. J Neurophysiol 109: 3025–3040.
107. Villa AEP, Tetko IV, Dutoit P, De Ribaupierre Y, De Ribaupierre F (1999) Corticofugal modulation of functional connectivity within the auditory thalamus of rat, guinea pig and cat revealed by cooling deactivation. J Neurosci Methods 86: 161–178.
108. Thom R (1972) Stabilité structurelle et morphogenèse. Essai d'une théorie générale des modèles. oaris: InterÉditions.
109. Douglas RJ, Martin KA, Whitteridge D (1989) A canonical microcircuit for neocortex. Neural computation 1: 480–488.
110. van Vreeswijk C, Sompolinsky H (1996) Chaos in neuronal networks with balanced excitatory and inhibitory activity. Science 274: 1724–1726.
111. Hill S, Villa AEP (1997) Dynamic transitions in global network activity influenced by the balance of excitation and inhibtion. Network: computational neural networks 8: 165–184.
112. Phillis JW, Kirkpatrick JR (1980) The actions of motilin, luteinizing hormone releasing hormone, cholecystokinin, somatostatin, vasoactive intestinal peptide, and other peptides on rat cerebral cortical neurons. Can J Physiol Pharmacol 58: 612–623.
113. Steriade M, Jones EG, Llinás R (1990) Thalamic oscillations and signalling. New York: Wiley.
114. Wright JJ (1990) Reticular activation and the dynamics of neuronal networks. Biol Cybern 62: 289–298.
115. Parent A, Hazrati LN (1995) Functional anatomy of the basal ganglia. i. the cortico-basal ganglia-thalamo-cortical loop. Brain Res Brain Res Rev 20: 91–127.
116. Fukai T, Shiino M (1990) Asymmetric neural networks incorporating the Dale hypothesis and noise-driven chaos. Phys Rev Lett 64: 1465–1468.
117. Tsodyks MV, Sejnowski T (1995) Rapid state switching in balanced cortical network models. Network: Computation in Neural Systems 6: 111–124.
118. Taylor JG, Villa AEP (2001) The "Conscious I": A Neuroheuristic Approach to the Mind. In: Baltimore D, Dulbecco R, Jacob F, Levi Montalcini R, editors, Frontiers of Life, Academic Press, volume III. pp. 349–270. ISBN: 0-12-077340-6.
119. Kanamaru T, Sekine M (2005) Synchronized firings in the networks of class 1 excitable neurons with excitatory and inhibitory connections and their dependences on the forms of interactions. Neural Computation 17: 1315–1338.
120. Villa AEP, Bajo Lorenzana VM, Vantini G (1996) Nerve growth factor modulates information processing in the auditory thalamus. Brain Res Bull 39: 139–147.
121. Villa AEP, Tetko IV, Dutoit P, Vantini G (2000) Non-linear cortico-cortical interactions modulated by cholinergic afferences from the rat basal forebrain. Biosystems 58: 219–228.
122. Kanamaru T, Fujii H, Aihara K (2013) Deformation of attractor landscape via cholinergic presynaptic modulations: a computational study using a phase neuron model. PLoS One 8.
123. Lien AD, Scanziani M (2013) Tuned thalamic excitation is amplified by visual cortical circuits. Nat Neurosci 16: 1315–1323.
124. Lintas A, Schwaller B, Villa AE (2013) Visual thalamocortical circuits in parvalbumin-deficient mice. Brain Res.
125. Turova TS, Villa AEP (2007) On a phase diagram for random neural networks with embedded spike timing dependent plasticity. Biosystems 89: 280–286.
126. Kozloski J, Cecchi GA (2010) A theory of loop formation and elimination by spike timing-dependent plasticity. Front Neural Circuits 4: e7.
127. Waddington A, Appleby PA, De Kamps M, Cohen N (2012) Triphasic spike-timing-dependent plasticity organizes networks to produce robust sequences of neural activity. Front Comput Neurosci 6: e88. Waddington2012e88
128. Kerr RR, Burkitt AN, Thomas DA, Gilson M, Grayden DB (2013) Delay selection by spike-timing-dependent plasticity in recurrent networks of spiking neurons receiving oscillatory inputs. PLoS Comput Biol 9.
129. Asai Y, Villa AEP (2012) Integration and transmission of distributed deterministic neural activity in feed-forward networks. Brain Res 1434: 17–33.
130. Copeland BJ (2002) Hypercomputation. Minds Mach 12: 461–502.
131. Copeland BJ (2004) Hypercomputation: philosophical issues. Theor Comput Sci 317: 251–267.