INTERACTIVE EVOLVING RECURRENT NEURAL NETWORKS ARE SUPER-TURING

Jérémie Cabessa[1,2]

[1]*Department of Information Systems, University of Lausanne, CH-1015 Lausanne, Switzerland*

[2]*Department of Computer Science, University of Massachusetts Amherst, Amherst, MA 01003, U.S.A.*

Keywords: Recurrent neural networks, Turing machines, Reactive systems, Evolving systems, Interactive computation, Neural computation, Super-turing.

Abstract: We consider a model of evolving recurrent neural networks where the synaptic weights can change over time, and we study the computational power of such networks in a basic context of interactive computation. In this framework, we prove that both models of rational- and real-weighted interactive evolving neural networks are computationally equivalent to interactive Turing machines with advice, and hence capable of super-Turing capabilities. These results support the idea that some intrinsic feature of biological intelligence might be beyond the scope of the current state of artificial intelligence, and that the concept of evolution might be strongly involved in the computational capabilities of biological neural networks. It also shows that the computational power of interactive evolving neural networks is by no means influenced by nature of their synaptic weights.

1 INTRODUCTION

Understanding the intrinsic nature of biological intelligence is an issue of central importance. In this context, much interest has been focused on comparing the computational capabilities of diverse theoretical neural models and abstract computing devices (McCulloch and Pitts, 1943; Kleene, 1956; Minsky, 1967; Siegelmann and Sontag, 1994; Siegelmann and Sontag, 1995; Siegelmann, 1999). As a consequence, the computational power of neural networks has been shown to be intimately related to the nature of their synaptic weights and activation functions, hence capable to range from finite state automata up to super-Turing capabilities.

However, in this global line of thinking, the neural models which have been considered fail to capture some essential biological features that are significantly involved in the processing of information in the brain. In particular, the plasticity of biological neural networks as well as the interactive nature of information processing in bio-inspired complex systems have not been taken into consideration.

The present paper falls within this perspective and extends the works by Cabessa and Siegelmann concerning the computational power of evolving or interactive neural networks (Cabessa and Siegelmann, 2011b; Cabessa and Siegelmann, 2011a). More precisely, here, we consider a model of evolving recurrent neural networks where the synaptic strengths of the neurons can change over time rather than staying static, and we study the computational capabilities of such networks in a basic context of interactive computation in line with the framework proposed by van Leeuwen and Wiedermann (van Leeuwen and Wiedermann, 2001a; van Leeuwen and Wiedermann, 2008). In this context, we prove that rational- and real-weighted *interactive evolving recurrent neural networks* are both computationally equivalent to interactive Turing machines with advice, thus capable of super-Turing capabilities. These results support the idea that some intrinsic feature of biological intelligence might be beyond the scope of the current state of artificial intelligence, and that the concept of evolution might be strongly involved in the computational capabilities of biological neural networks. They also show that the nature of the synaptic weights has no influence on the computational power of interactive evolving neural networks.

2 PRELIMINARIES

Before entering into further considerations, the following definitions and notations need to be introduced. Given the binary bit alphabet $\{0,1\}$, we let $\{0,1\}^*$, $\{0,1\}^+$, $\{0,1\}^n$, and $\{0,1\}^\omega$ denote respec-

Cabessa J..
INTERACTIVE EVOLVING RECURRENT NEURAL NETWORKS ARE SUPER-TURING.
DOI: 10.5220/0003740603280333
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 328-333
ISBN: 978-989-8425-95-9

tively the sets of finite words, non-empty finite words, finite words of length n, and infinite words, all of them over alphabet $\{0,1\}$. We also let $\{0,1\}^{\leq\omega} = \{0,1\}^* \cup \{0,1\}^\omega$ be the set of all possible words (finite or infinite) over $\{0,1\}$.

Any function $\varphi : \{0,1\}^\omega \longrightarrow \{0,1\}^{\leq\omega}$ will be referred to as an ω-*translation*.

Besides, for any $x \in \{0,1\}^{\leq\omega}$, the *length* of x is denoted by $|x|$ and corresponds to the number of letters contained in x. If x is non-empty, we let $x(i)$ denote the $(i+1)$-th letter of x, for any $0 \leq i < |x|$. Hence, x can be written as $x = x(0)x(1)\cdots x(|x|-1)$ if it is finite, and as $x = x(0)x(1)x(2)\cdots$ otherwise. Moreover, the *concatenation* of x and y is written $x \cdot y$ or sometimes simply xy. The empty word is denoted λ.

3 INTERACTIVE COMPUTATION

3.1 The Interactive Paradigm

Interactive computation refers to the computational framework where systems may react or interact with each other as well as with their environment during the computation (Goldin et al., 2006). This paradigm was theorized in contrast to classical computation which rather proceeds in a closed-box fashion and was argued to "no longer fully corresponds to the current notions of computing in modern systems" (van Leeuwen and Wiedermann, 2008). Interactive computation also provides a particularly appropriate framework for the consideration of natural and bio-inspired complex information processing systems (van Leeuwen and Wiedermann, 2001a; van Leeuwen and Wiedermann, 2008).

The general interactive computational paradigm consists of a step by step exchange of information between a system and its environment. In order to capture the unpredictability of next inputs at any time step, the dynamically generated input streams need to be modeled by potentially infinite sequences of symbols (the case of finite sequences of symbols would necessarily reduce to the classical computational framework) (Wegner, 1998; van Leeuwen and Wiedermann, 2008).

Throughout this paper, we consider a basic interactive computational scenario where at every time step, the environment sends a non-empty input bit to the system (full environment activity condition), the system next updates its current state accordingly, and then either produces a corresponding output bit, or remains silent for a while to express the need of some internal computational phase before outputting a new bit, or remains silent forever to express the fact that it has died. Consequently, after infinitely many time steps, the system will have received an infinite sequence of consecutive input bits and translated it into a corresponding finite or infinite sequence of not necessarily consecutive output bits. Accordingly, any interactive system $\mathcal{S}$ realizes an ω-translation $\varphi_{\mathcal{S}} : \{0,1\}^\omega \longrightarrow \{0,1\}^{\leq\omega}$.

3.2 Interactive Turing Machines

An *interactive Turing machine* (I-TM) $\mathcal{M}$ consists of a classical Turing machine yet provided with input and output ports rather than tapes in order to process the interactive sequential exchange of information between the device and its environment (van Leeuwen and Wiedermann, 2001a). According to our interactive scenario, it is assumed that at every time step, the environment sends a non-silent input bit to the machine and the machine answers by either producing a corresponding output bit or rather remaining silent (expressed by the fact of outputting the λ symbol).

According to this definition, for any infinite *input stream* $s \in \{0,1\}^\omega$, we define the corresponding *output stream* $o_s \in \{0,1\}^{\leq\omega}$ of $\mathcal{M}$ as the finite or infinite subsequence of (non-λ) output bits produced by $\mathcal{M}$ after having processed input s. In this manner, any machine $\mathcal{M}$ naturally induces an ω-translation $\varphi_{\mathcal{M}} : \{0,1\}^\omega \longrightarrow \{0,1\}^{\leq\omega}$ defined by $\varphi_{\mathcal{M}}(s) = o_s$, for each $s \in \{0,1\}^\omega$. Finally, an ω-translation $\psi : \{0,1\}^\omega \longrightarrow \{0,1\}^{\leq\omega}$ is said to be *realizable* by some interactive Turing machine iff there exists some I-TM $\mathcal{M}$ such that $\varphi_{\mathcal{M}} = \psi$.

Besides, an *interactive Turing machine with advice* (I-TM/A) $\mathcal{M}$ consists of an interactive Turing machine provided with an advice mechanism (van Leeuwen and Wiedermann, 2001a). The mechanism comes in the form of an *advice function* $\alpha : \mathbb{N} \longrightarrow \{0,1\}^*$. Moreover, the machine $\mathcal{M}$ uses two auxiliary special tapes, an *advice input tape* and an *advice output tape*, as well as a designated *advice state*. During its computation, $\mathcal{M}$ can write the binary representation of an integer m on its advice input tape, one bit at a time. Yet at time step n, the number m is not allowed to exceed n. Then, at any chosen time, the machine can enter its designated advice state and then have the finite string $\alpha(m)$ be written on the advice output tape in one time step, replacing the previous content of the tape. The machine can repeat this extra-recursive calling process as many times as it wants during its infinite computation.

Once again, according to our interactive scenario, any I-TM/A $\mathcal{M}$ induces an ω-translation $\varphi_{\mathcal{M}} : \{0,1\}^\omega \longrightarrow \{0,1\}^{\leq\omega}$ which maps every infinite input stream s to the corresponding finite or infinite output stream o_s produced by $\mathcal{M}$. Finally, an ω-translation $\psi : \{0,1\}^\omega \longrightarrow \{0,1\}^{\leq\omega}$ is said to be *realizable* by some interactive Turing machine with advice iff there exists some I-TM/A $\mathcal{M}$ such that $\varphi_{\mathcal{M}} = \psi$.

4 INTERACTIVE EVOLVING RECURRENT NEURAL NETWORKS

We now consider a natural extension to the present interactive framework of the model of evolving recurrent neural network described by Cabessa and Siegelmann in (Cabessa and Siegelmann, 2011b).

An *evolving recurrent neural network* (Ev-RNN) consists of a synchronous network of neurons (or processors) related together in a general architecture – not necessarily loop free or symmetric. The network contains a finite number of neurons $(x_j)_{j=1}^N$, as well as M parallel input lines carrying the input stream transmitted by the environment into M of the N neurons, and P designated output neurons among the N whose role is to communicate the output of the network to the environment. Furthermore, the synaptic connections between the neurons are assumed to be time dependent rather than static. At each time step, the activation value of every neuron is updated by applying a linear-sigmoid function to some weighted affine combination of values of other neurons or inputs at previous time step.

Formally, given the activation values of the internal and input neurons $(x_j)_{j=1}^N$ and $(u_j)_{j=1}^M$ at time t, the activation value of each neuron x_i at time $t+1$ is then updated by the following equation

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

for $i = 1,\ldots,N$, where all $a_{ij}(t)$, $b_{ij}(t)$, and $c_i(t)$ are *time dependent* values describing the evolving weighted synaptic connections and weighted bias of the network, and σ is the classical saturated-linear activation function defined by $\sigma(x) = 0$ if $x < 0$, $\sigma(x) = x$ if $0 \leq x \leq 1$, and $\sigma(x) = 1$ if $x > 1$.

In order to stay consistent with our interactive scenario, we need to define the notion of an *interactive evolving recurrent neural network* (I-Ev-RNN) which adheres to a rigid encoding of the way input and output are interactively processed between the environment and the network.

First of all, we assume that any I-Ev-RNN is provided with a single binary input line u whose role is to transmit to the network the infinite input stream of bits sent by the environment. We also suppose that any I-Ev-RNN is equipped with two binary output lines, a data line y_d and a validation line y_v. The role of the data line is to carry the output stream of the network, while the role of the validation line is to describe when the data line is active and when it is silent. Accordingly, the output stream transmitted by the network to the environment will be defined as the (finite or infinite) subsequence of successive data bits that occur simultaneously with positive validation bits.

Hence, if $\mathcal{N}$ is an I-Ev-RNN with initial activation values $x_i(0) = 0$ for $i = 1,\ldots,N$, then any infinite *input stream*

$$s = s(0)s(1)s(2)\cdots$$

$\in \{0,1\}^\omega$ transmitted to input line u induces via Equation (1) a corresponding pair of infinite streams

$$(y_d(0)y_d(1)y_d(2)\cdots, y_v(0)y_v(1)y_v(2)\cdots)$$

$\in \{0,1\}^\omega \times \{0,1\}^\omega$. The *output stream* of $\mathcal{N}$ according to input s is then given by the finite or infinite subsequence o_s of successive data bits that occur simultaneously with positive validation bits, namely

$$o_s = \langle y_d(i) : i \in \mathbb{N} \text{ and } y_v(i) = 1\rangle \in \{0,1\}^{\leq\omega}.$$

It follows that any I-Ev-RNN $\mathcal{N}$ naturally induces an ω-translation $\varphi_{\mathcal{N}} : \{0,1\}^\omega \longrightarrow \{0,1\}^{\leq\omega}$ defined by $\varphi_{\mathcal{N}}(s) = o_s$, for each $s \in \{0,1\}^\omega$. An ω-translation $\psi : \{0,1\}^\omega \longrightarrow \{0,1\}^{\leq\omega}$ is said to be *realizable* by some I-Ev-RNN iff there exists some I-Ev-RNN $\mathcal{N}$ such that $\varphi_{\mathcal{N}} = \psi$.

Finally, throughout this paper, two models of interactive evolving recurrent neural networks are considered according to whether their underlying synaptic weights are confined to the class of rational or real numbers. A *rational* interactive evolving recurrent neural network (I-Ev-RNN[$\mathbb{Q}$]) denotes an I-Ev-RNN whose all synaptic weights are rational numbers, and a *real* interactive evolving recurrent neural network (I-Ev-RNN[$\mathbb{R}$]) stands for an I-Ev-RNN whose all synaptic weights are real numbers. Note that since rational numbers are included in real numbers, every I-Ev-RNN[$\mathbb{Q}$] is also a particular I-Ev-RNN[$\mathbb{R}$] by definition.

5 THE COMPUTATIONAL POWER OF INTERACTIVE EVOLVING RECURRENT NEURAL NETWORKS

In this section, we prove that interactive evolving recurrent neural networks are computationally equivalent to interactive Turing machine with advice, irrespective of whether their synaptic weights are rational or real. It directly follows that interactive evolving neural networks are indeed capable super-Turing computational capabilities.

Towards this purpose, we first show that the two models of rational- and real-weighted neural networks under considerations are indeed computationally equivalent.

Proposition 1. *I-Ev-RNN[$\mathbb{Q}$]s and I-Ev-RNN[$\mathbb{R}$]s are computationally equivalent.*

Proof. First of all, recall that every I-Ev-RNN[$\mathbb{Q}$] is also a I-Ev-RNN[$\mathbb{R}$] by definition. Hence, any ω-translation $\varphi : \{0,1\}^\omega \longrightarrow \{0,1\}^{\leq\omega}$ realizable by some I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}$ is also realizable by some I-Ev-RNN[$\mathbb{R}$], namely $\mathcal{N}$ itself.

Conversely, let $\mathcal{N}$ be some I-Ev-RNN[$\mathbb{R}$]. We prove the existence of an I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}'$ which realizes the same ω-translation as $\mathcal{N}$. The idea is to encode all possible intermediate output values of $\mathcal{N}$ into some evolving synaptic weight of $\mathcal{N}'$, and to make $\mathcal{N}'$ decode and output these successive values in order to answer precisely like $\mathcal{N}$ on every possible input stream.

More precisely, for every finite word $x \in \{0,1\}^+$, let $\mathcal{N}(x) \in \{0,1,2\}$ denote the encoding of the output answer of $\mathcal{N}$ on input x at precise time step $t = |x|$, where $\mathcal{N}(x) = 0$, $\mathcal{N}(x) = 1$, and $\mathcal{N}(x) = 2$ respectively mean that $\mathcal{N}$ has answered λ, 0, and 1 on input x at time step $t = |x|$. Moreover, for any $n > 0$, let $x_{n,1}, \ldots, x_{n,2^n}$ be the lexicographical enumeration of the words of $\{0,1\}^n$, and let $w_n \in \{0,1,2,3\}^*$ be the finite word given by $w_n = 3 \cdot \mathcal{N}(x_{n,1}) \cdot 3 \cdot \mathcal{N}(x_{n,2}) \cdot 3 \cdots 3 \cdot \mathcal{N}(x_{n,2^n}) \cdot 3$. Then, consider the rational encoding q_n of the word w_n given by

$$q_n = \sum_{i=1}^{|w_n|} \frac{2 \cdot w_n(i) + 1}{8^i}.$$

It follows that $q_n \in]0,1[$ for all $n > 0$, and that $q_n \neq q_{n+1}$, since $w_n \neq w_{n+1}$ for all $n > 0$. This encoding provides a corresponding decoding procedure which is recursive (Siegelmann and Sontag, 1994; Siegelmann and Sontag, 1995). Hence, every finite word w_n can be decoded from the value q_n by some Turing machine, or equivalently, by some rational recurrent neural network. This feature is important for our purpose.

Now, the I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}'$ consists of one evolving and one non-evolving rational-weighted subnetwork connected together in a specific manner. More precisely, the evolving rational-weighted part of $\mathcal{N}'$ is made up of a single designated processor x_e receiving a background activity of evolving intensity $c_e(t)$. The synaptic weight $c_e(t)$ successively takes the rational bounded values $q_1, q_2, q_3, \ldots$, by switching from value q_k to q_{k+1} after t_k time steps, for some t_k large enough to satisfy the conditions of the procedure described below. The non-evolving rational-weighted part of $\mathcal{N}'$ is designed and connected to the neuron x_e in such a way as to perform the following recursive procedure: for any infinite input stream $s \in \{0,1\}^\omega$ provided bit by bit, the sub-network stores in its memory the successive incoming bits $s(0), s(1), \ldots$ of s, and simultaneously, for each successive $t > 0$, the sub-network first waits for the synaptic weight q_t to occur as a background activity of neuron x_e, decodes the output value $\mathcal{N}(s(0)s(1)\cdots s(t-1))$ from q_t, outputs it, and then continues the same routine with respect to the next step $t+1$. Note that the equivalence between Turing machines and rational-weighted recurrent neural networks ensures that the above recursive procedure can indeed be performed by some non-evolving rational-weighted recurrent neural subnetwork (Siegelmann and Sontag, 1995).

In this way, the infinite sequence of successive non-empty output bits provided by networks $\mathcal{N}$ and $\mathcal{N}'$ are the very same, so that $\mathcal{N}$ and $\mathcal{N}'$ indeed realize the same ω-translation. □

We now prove that rational-weighted interactive evolving neural networks are computationally equivalent to interactive Turing machines with advice.

Proposition 2. *I-Ev-RNN[$\mathbb{Q}$]s and I-TM/As are computationally equivalent.*

Proof. First of all, let $\mathcal{N}$ be some I-Ev-RNN[$\mathbb{Q}$]. We give the description of an I-TM/A $\mathcal{M}$ which realizes the same ω-translation as $\mathcal{N}$. Towards this purpose, for each $t > 0$, let $\mathcal{N}(t)$ be the description of the synaptic weights of network $\mathcal{N}$ at time t. Since all synaptic weights of $\mathcal{N}$ are rational, the whole synaptic description $\mathcal{N}(t)$ can be encoded by some finite word $\alpha(t) \in \{0,1\}^+$ (every rational number can be encoded by some finite word of bits, hence so does every finite sequence of rational numbers).

Now, consider the I-TM/A $\mathcal{M}$ whose advice function is precisely α, and which, thanks to the advice α, provides a step by step simulation of the behavior of $\mathcal{N}$ in order to eventually produce the very same output stream as $\mathcal{N}$. More precisely, on every infinite input stream $s \in \{0,1\}^\omega$, the machine $\mathcal{M}$ stores in its memory the successive incoming bits $s(0), s(1), \ldots$ of s, and simultaneously, for each successive $t \geq 0$, it retrieves the activation values $\overrightarrow{x}(t)$ of $\mathcal{N}$ at time t from its memory, calls its advice $\alpha(t)$ in order to retrieve the synaptic description $\mathcal{N}(t)$, uses this information in order to compute via Equation (1) the activation and output values $\overrightarrow{x}(t+1)$, $y_d(t+1)$, and $y_v(t+1)$ of $\mathcal{N}$ at next time step $t+1$, provides the corresponding output encoded by $y_d(t+1)$ and $y_v(t+1)$, and finally stores the activation values $\vec{x}(t+1)$ of $\mathcal{N}$ in order to be able to repeat the same routine with respect to the next step $t+1$.

In this way, the infinite sequence of successive non-empty output bits provided by the network $\mathcal{N}$ and the machine $\mathcal{M}$ are the very same, so that $\mathcal{N}$ and $\mathcal{M}$ indeed realize the same ω-translation.

Conversely, let $\mathcal{M}$ be some I-TM/A with advice function α. We build an I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}$ which realizes the same ω-translation as $\mathcal{M}$. The idea is to encode the successive advice values $\alpha(0), \alpha(1), \alpha(2), \ldots$ of $\mathcal{M}$ into some evolving rational synaptic weight of $\mathcal{N}$, and to store them in the memory of $\mathcal{N}$ in order to be capable of simulating with $\mathcal{N}$ every recursive and extra-recursive computational step of $\mathcal{M}$.

More precisely, for each $n \geq 0$, let $w_{\alpha(n)} \in \{0,1,2\}^*$ be the finite word given by $w_{\alpha(n)} = 2 \cdot \alpha(0) \cdot 2 \cdot \alpha(1) \cdot 2 \cdots 2 \cdot \alpha(n) \cdot 2$, and let $q_{\alpha(n)}$ be the rational encoding of the word $w_{\alpha(n)}$ given by

$$q_{\alpha(n)} = \sum_{i=1}^{|w_n|} \frac{2 \cdot w_n(i) + 1}{6^i}.$$

Note that $q_{\alpha(n)} \in]0,1[$ for all $n > 0$, and that $q_{\alpha(n)} \neq q_{\alpha(n+1)}$, since $w_{\alpha(n)} \neq w_{\alpha(n+1)}$ for all $n > 0$. Moreover, it can be shown that the finite word $w_{\alpha(n)}$ can be decoded from the value $q_{\alpha(n)}$ by some Turing machine, or equivalently, by some rational recurrent neural network (Siegelmann and Sontag, 1994; Siegelmann and Sontag, 1995).

Now, the I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}$ consists of one evolving and one non-evolving rational-weighted subnetwork connected together. More precisely, the evolving rational-weighted part of $\mathcal{N}$ is made up of a single designated processor x_e receiving a background activity of evolving intensity $c_e(0) = q_{\alpha(0)}$, $c_e(1) = q_{\alpha(1)}$, $c_e(2) = q_{\alpha(2)}$, ... The non-evolving rational-weighted part of $\mathcal{N}$ is designed and connected to x_e in order to simulate the behavior of $\mathcal{M}$ as follows: every recursive computational step of $\mathcal{M}$ is simulated by $\mathcal{N}$ in the classical way (Siegelmann and Sontag, 1995); moreover, every time $\mathcal{M}$ proceeds to some extra-recursive call to some value $\alpha(m)$, the network stores the current synaptic weight $q_{\alpha(t)}$ in its memory, retrieves the string $\alpha(m)$ from the rational value $q_{\alpha(t)}$ – which is possible as one necessarily has $m \leq t$, since $\mathcal{N}$ cannot proceed faster than $\mathcal{M}$ by construction –, and then pursues the simulation of the next recursive step of $\mathcal{M}$ in the classical way.

In this manner, the infinite sequence of successive non-empty output bits provided by the machine $\mathcal{M}$ and the network $\mathcal{N}$ are the very same on every possible infinite input stream, so that $\mathcal{M}$ and $\mathcal{N}$ indeed realize the same ω-translation. □

Propositions 1 and 2 directly imply the equivalence between interactive evolving recurrent neural networks and interactive Turing machines with advice. Since interactive Turing machines with advice are strictly more powerful than their classical counterparts (van Leeuwen and Wiedermann, 2001a; van Leeuwen and Wiedermann, 2001b), it follows that interactive evolving networks are capable of a super-Turing computational power, irrespective of whether their underlying synaptic weights are rational or real.

Theorem 1. *I-Ev-RNN[$\mathbb{Q}$]s, I-Ev-RNN[$\mathbb{R}$]s, and I-TM/As are equivalent super-Turing models of computation.*

6 DISCUSSION

The present paper provides a characterization of the computational power of evolving recurrent neural networks in a basic context of interactive and active memory computation. It is shown that interactive evolving neural networks are computationally equivalent to interactive machines with advice, irrespective of whether their underlying synaptic weights are rational or real. Consequently, the model of interactive evolving neural networks under consideration is potentially capable of super-Turing computational capabilities.

These results provide a proper generalization to the interactive context of the super-Turing and equivalent capabilities of rational- and real-weighted evolving neural networks established in the case of classical computation (Cabessa and Siegelmann, 2011b).

In order to provide a deeper understanding of the present contribution, the results concerning the computational power of interactive *static* recurrent neural networks need to be recalled. In the static case, rational- and real-weighted interactive neural networks (resp. denoted by I-St-RNN[$\mathbb{Q}$]s and I-St-RNN[$\mathbb{R}$]s) were proven to be computationally equivalent to interactive Turing machines and interactive Turing machines with advice, respectively (Cabessa and Siegelmann, 2011a). Consequently, I-Ev-RNN[$\mathbb{Q}$]s, I-Ev-RNN[$\mathbb{R}$]s, and I-St-RNN[$\mathbb{R}$]s are all computationally equivalent to I-TM/As, whereas I-St-RNN[$\mathbb{Q}$]s are equivalent to I-TMs.

Given such considerations, the case of rational-weighted interactive neural networks appears to be of specific interest. In this context, the translation from the static to the evolving framework really brings up an additional super-Turing computational power to the networks. However, it is worth noting that such super-Turing capabilities can only be achieved in cases where the evolving synaptic patters are themselves non-recursive (i.e., non Turing-computable), since the consideration of any kind of recursive evolution would necessarily restrain the corresponding networks to no more than Turing capabilities. Hence, according to this model, the existence of super-Turing potentialities of evolving neural networks depends on the possibility for "nature" to realize non-recursive patterns of synaptic evolution.

By contrast, in the case of real-weighted interactive neural networks, the translation from the static to the evolving framework doesn't bring any additional computational power to the networks. In other words, the computational capabilities brought up by the power of the continuum cannot be overcome by incorporating some further possibilities of synaptic evolution in the model.

To summarize, the possibility of *synaptic evolution* in a basic first-order interactive rate neural model provides an alternative and equivalent way to the consideration of *analog* synaptic weights towards the achievement super-Turing computational capabilities of neural networks. Yet even if the concepts of evolution on the one hand and analog continuum on the other hand turn out to be mathematically equivalent in this sense, they are nevertheless conceptually well distinct. Indeed, while the power of the continuum is a pure conceptualization of the mind, the synaptic plasticity of the networks is itself something really observable in nature.

The present work is envisioned to be extended in three main directions. Firstly, a deeper study of the issue from the perspective of computational complexity could be of interest. Indeed, the simulation of an I-Ev-RNN[$\mathbb{R}$] $\mathcal{N}$ by some I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}'$ described in the proof of Proposition 1 is clearly not effective in the sense that for any output move of $\mathcal{N}$, the network $\mathcal{N}'$ needs first to decode the word w_n of size exponential in n before being capable of providing the same output as $\mathcal{N}$. In the proof of Proposition 2, the effectivity of the two simulations that are described depend on the complexity of the synaptic configurations $\mathcal{N}(t)$ of $\mathcal{N}$ as well as on the complexity of the advice function $\alpha(n)$ of $\mathcal{M}$.

Secondly, it is expected to consider more realistic neural models capable of capturing biological mechanisms that are significantly involved in the computational and dynamical capabilities of neural networks as well as in the processing of information in the brain in general. For instance, the consideration of biological features such as spike timing dependent plasticity, neural birth and death, apoptosis, chaotic behaviors of neural networks could be of specific interest.

Thirdly, it is envision to consider more realistic paradigms of interactive computation, where the processes of interaction would be more elaborated and biologically oriented, involving not only the network and its environment, but also several distinct components of the network as well as different aspects of the environment.

Finally, we believe that the study of the computational power of neural networks from the perspective of theoretical computer science shall ultimately bring further insight towards a better understanding of the intrinsic nature of biological intelligence.

REFERENCES

Cabessa, J. and Siegelmann, H. T. (2011a). The computational power of interactive recurrent neural networks. Submitted to *Neural Comput.*

Cabessa, J. and Siegelmann, H. T. (2011b). Evolving recurrent neural networks are super-Turing. In *International Joint Conference on Neural Networks, IJCNN 2011*, pages 3200–3206. IEEE.

Goldin, D., Smolka, S. A., and Wegner, P. (2006). *Interactive Computation: The New Paradigm*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Kleene, S. C. (1956). Representation of events in nerve nets and finite automata. In *Automata Studies*, volume 34 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, USA.

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

Minsky, M. L. (1967). *Computation: finite and infinite machines*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Siegelmann, H. T. (1999). *Neural networks and analog computation: beyond the Turing limit*. Birkhauser Boston Inc., Cambridge, MA, USA.

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

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

van Leeuwen, J. and Wiedermann, J. (2001a). Beyond the Turing limit: Evolving interactive systems. In *SOFSEM 2001: Theory and Practice of Informatics*, volume 2234 of *LNCS*, pages 90–109. Springer Berlin / Heidelberg.

van Leeuwen, J. and Wiedermann, J. (2001b). The Turing machine paradigm in contemporary computing. In *Mathematics Unlimited - 2001 and Beyond*, LNCS, pages 1139–1155. Springer-Verlag.

van Leeuwen, J. and Wiedermann, J. (2008). How we think of computing today. In *Logic and Theory of Algorithms*, volume 5028 of *LNCS*, pages 579–593. Springer Berlin / Heidelberg.

Wegner, P. (1998). Interactive foundations of computing. *Theor. Comput. Sci.*, 192:315–351.