The Super-Turing Computational Power of Interactive Evolving Recurrent Neural Networks

Jérémie Cabessa and Alessandro E.P. Villa

Department of Information Systems
University of Lausanne
CH-1015 Lausanne, Switzerland

Abstract. Understanding the dynamical and computational capabilities of neural models represents an issue of central importance. Here, we consider a model of first-order recurrent neural networks provided with the possibility to evolve over time and involved in a basic interactive and memory active computational paradigm. In this context, we prove that the so-called *interactive evolving recurrent neural networks* are computationally equivalent to interactive Turing machines with advice, hence capable of super-Turing potentialities. We further provide a precise characterisation of the ω-translations realised by these networks. Therefore, the consideration of evolving capabilities in a first-order neural model provides the potentiality to break the Turing barrier.

Keywords: recurrent neural networks, neural computation, interactive computation, analog computation, Turing machines with advice, super-Turing.

1 Introduction

Unerstanding the dynamical and computational capabilities of neural models represents an issue of central importance to assess the performances at reach by neural networks. In this context, much interest has been focused on comparing the computational capabilities of diverse theoretical neural models to those of abstract computing devices [7,14,6,9,8,11,12,10]. 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 only recently started to be investigated [2,3].

The present paper falls within this perspective and concerns the computational capabilities of a model of interactive evolving recurrent neural networks. This work is a direct extension of previous results by Cabessa [1]. More precisely,

V. Mladenov et al. (Eds.): ICANN 2013, LNCS 8131, pp. 58–65, 2013.

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 [15,17]. 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, hence capable of super-Turing capabilities. Moreover, we provide a precise mathematical characterisation of the ω-translations realised by these neural models. 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

Given some finite alphabet Σ, we let Σ^*, Σ^+, Σ^n, and Σ^ω denote respectively the sets of finite words, non-empty finite words, finite words of length n, and infinite words, all of them over alphabet Σ. We also let $\Sigma^{\leq\omega} = \Sigma^* \cup \Sigma^\omega$ be the set of all possible words (finite or infinite) over Σ. The empty word is denoted by λ.

For any $x \in \Sigma^{\leq\omega}$, 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 $0 \leq i < |x|$. The *prefix* $x(0)\cdots x(i)$ of x is denoted by $x[0{:}i]$, for any $0 \leq i < |x|$. For any $x \in \Sigma^*$ and $y \in \Sigma^{\leq\omega}$, the fact that x is a *prefix* (resp. *strict prefix*) of y is denoted by $x \subseteq y$ (resp. $x \subsetneq y$). If $x \subseteq y$, we let $y - x = y(|x|)\cdots y(|y|-1)$ be the *suffix* of y that is not common to x (we have $y - x = \lambda$ if $x = y$). Moreover, the *concatenation* of x and y is denoted by $x \cdot y$ or sometimes simply by xy. The word x^n consists of n copies of x concatenated together, with the convention that $x^0 = \lambda$.

A function $f : \Sigma^* \to \Sigma^*$ is called *monotone* if the relation $x \subseteq y$ implies $f(x) \subseteq f(y)$, for all $x, y \in \Sigma^*$. It is called *recursive* if it can be computed by some Turing machine over Σ. Furthermore, throughout this paper, any function $\varphi : \Sigma^\omega \to \Sigma^{\leq\omega}$ will be referred to as an *ω-translation*.

Note that any monotone function $f : \Sigma^* \to \Sigma^*$ *induces in the limit* an ω-translation $f_\omega : \Sigma^\omega \to \Sigma^{\leq\omega}$ defined by

$$f_\omega(x) = \lim_{i\geq 0} f(x[0{:}i])$$

where $\lim_{i\geq 0} f(x[0{:}i])$ denotes the smallest finite word that contains each word of $\{f(x[0{:}i]) : i \geq 0\}$ as a finite prefix if $\lim_{i\to\infty} |f(x[0{:}i])| < \infty$, and $\lim_{i\geq 0} f(x[0{:}i])$ denotes the unique infinite word that contains each word of $\{f(x[0{:}i]) : i \geq 0\}$ as a finite prefix if $\lim_{i\to\infty} |f(x[0{:}i])| = \infty$. Note that the monotonicity of f ensures that the value $f_\omega(x)$ is well-defined for all $x \in \Sigma^\omega$. Intuitively, the value $f_\omega(x)$ corresponds to the finite or infinite word that is ultimately approached by the sequence of growing prefixes $\langle f(x[0{:}i]) : i \geq 0 \rangle$.

An ω-translation $\psi : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ will be called *continuous* if there exists a monotone function $f : \{0,1\}^* \to \{0,1\}^*$ such that $f_\omega = \psi$; it will be called *recursive continuous* if there exists a monotone and recursive function $f : \{0,1\}^* \to \{0,1\}^*$ such that $f_\omega = \psi$.

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 [5]. This paradigm was theorised in contrast to classical computation [13] 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" [17]. Interactive computation also provides a particularly appropriate framework for the consideration of natural and bio-inspired complex information processing systems [15,17].

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) [18,17].

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 $\mathcal{S}$ realises an ω-translation $\varphi_{\mathcal{S}} : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$.

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 [15]. 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).

An *interactive Turing machine with advice* (I-TM/A) $\mathcal{M}$ consists of an interactive Turing machine provided with an advice mechanism which comes in the form of an *advice function* $\alpha : \mathbb{N} \to \{0,1\}^*$ [15]. 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.

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 $\varphi_\mathcal{M} : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ defined by $\varphi_\mathcal{M}(s) = o_s$, for each $s \in \{0,1\}^\omega$. Finally, an ω-translation $\psi : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ is said to be *realisable* by some interactive Turing machine with advice iff there exists some I-TM/A $\mathcal{M}$ such that $\varphi_\mathcal{M} = \psi$.

Interactive Turing machines with advice are strictly more powerful than interactive Turing machines (without advice) [15], and were shown to be computationally equivalent to several others other non-uniform models of interactive computation, like sequences of interactive finite automata, site machines, and web Turing machines [15].

4 Interactive Evolving Recurrent Neural Networks

An *evolving recurrent neural network* (Ev-RNN) consists of a synchronous network of neurons (or processors) related together in a general architecture. The network contains a finite number of neurons $(x_i)_{i=1}^N$, M parallel input lines $(u_i)_{i=1}^M$, and P designated output neurons among the N. 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

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

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 $\varphi_\mathcal{N} : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ defined by $\varphi_\mathcal{N}(s) = o_s$, for each $s \in \{0,1\}^\omega$. An ω-translation $\psi : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ is said to be *realisable* 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. Rational- and real-weighted interactive evolving recurrent neural network will be dented by I-Ev-RNN[$\mathbb{Q}$] and I-Ev-RNN[$\mathbb{R}$], respectively. 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

The following result states 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. A precise mathematical characterisation of the ω-translations realised by these networks is also provided. It directly follows that interactive evolving neural networks are capable super-Turing computational potentialities.

Theorem 1. *Let $\psi : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ be an ω-translation. The following conditions are equivalent:*

1. *ψ is realisable by some I-TM/A;*
2. *ψ is realisable by some I-Ev-RNN[$\mathbb{Q}$];*
3. *ψ is realisable by some I-Ev-RNN[$\mathbb{R}$];*
4. *ψ is continuous.*

Proof (sketch). (1) $\Rightarrow$ (2): We will use the fact that every Turing machine can be simulated by some rational-weighted recurrent neural network [12]. Let $\mathcal{M}$ be some I-TM/A with advice function $\alpha : \mathbb{N} \to \{0,1\}^*$. We show that there exists an I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}$ which realises the same ω-translation as $\mathcal{M}$. First, for each $i > 0$, let q_i be a rational number encoding in a recursive manner the sequence of successive advice values $\langle \alpha(0), \ldots, \alpha(i) \rangle$. Note that such an encoding is indeed possible since by definition of α every $\alpha(i)$ is a finite word. Now, consider the following description of an I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}$. The network $\mathcal{N}$ contains a specific evolving synaptic connection which takes as evolving weights the successive values q_i's defined above, for all $i > 0$. The network $\mathcal{N}$ also contains a non-evolving rational-weighted part which is designed is order to simulate $\mathcal{M}$ as follows: every recursive computational step of $\mathcal{M}$ is simulated by $\mathcal{N}$ in the classical way, as described in [12]; moreover, for every extra-recursive call to some advice value $\alpha(m)$ performed by $\mathcal{M}$ at some time $t \geq m$, $\mathcal{N}$ first waits for the synaptic weight q_t to occur, then stores the synaptic weight q_t in its memory, then decodes the specific string $\alpha(m)$ from the rational value q_t (which is possible since $t \geq m$), and then pursues the simulation of the next recursive steps of $\mathcal{M}$ in the classical way [12]. In this manner, $\mathcal{N}$ realises the same ω-translation as $\mathcal{M}$.

(2) $\Rightarrow$ (3): Note that every I-Ev-RNN[$\mathbb{Q}$] is also an I-Ev-RNN[$\mathbb{R}$] by definition. Hence, if ψ is an ω-translation realised by some I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}$, then ψ is also realised by some I-Ev-RNN[$\mathbb{R}$], namely by $\mathcal{N}$ itself.

(3) $\Rightarrow$ (4): Let $\varphi_{\mathcal{N}}$ be an ω-translation realised by some I-Ev-RNN[$\mathbb{R}$] $\mathcal{N}$. We show that $\varphi_{\mathcal{N}}$ is continuous. For this purpose, consider the function $f : \{0,1\}^* \to \{0,1\}^*$ which maps every finite word u to the unique corresponding finite word output by $\mathcal{N}$ after precisely $|u|$ steps of computation, when $u \cdot x$ is provided as input bit by bit, for any possible suffix $x \in \{0,1\}^\omega$. The definition of our interactive scenario ensures that f is well-defined (i.e., that $f(u)$ is independent of the suffix x), and that f is monotone. We can prove that the function $\varphi_{\mathcal{N}}$ realised by $\mathcal{N}$ corresponds precisely to the limit of the monotone function f as defined in Section 2, or in other words, that $\varphi_{\mathcal{N}} = f_\omega$. Therefore, $\varphi_{\mathcal{N}}$ is continuous.

(4) $\Rightarrow$ (1): Let $\psi : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ be some continuous ω-translation. Then, by definition, there exists some monotone function $f : \{0,1\}^* \to \{0,1\}^*$ such that $f_\omega = \psi$. For each $i \geq 0$, let $(z_{i,j})_{j=1}^{2^i}$ be the lexicographic enumeration of all binary words of length i. Let $\alpha : \mathbb{N} \to \{0,1\}^*$ be the advice function such that $\alpha(i)$ represents some recursive encoding of the successive values $f(z_{i,j})$ separated by $\sharp$'s, for $j = 1, \ldots, 2^i$ (for instance, $\alpha(2)$ is a binary encoding of $\sharp f(00) \sharp f(01) \sharp f(10) \sharp f(11) \sharp$). Now, consider the I-TM/A $\mathcal{M}$ with advice α working on every infinite input $s = s(0)s(1)s(2) \cdots \in \{0,1\}^\omega$ as follows: for each new input bit $s(i+1)$, $\mathcal{M}$ calls its advice value $\alpha(i+1)$, decodes the specific value $f(s(0) \cdots s(i+1))$ from it, checks if $f(s(0) \cdots s(i+1))$ strictly extends the

previous decoded value $f(s(0)\cdots s(i))$, and if this is the case, outputs this extension bit by bit. We can show that the function $\varphi_{\mathcal{M}}$ realised by $\mathcal{M}$ in this manner corresponds precisely to the limit of the monotone function f as defined in Section 2, or in other words, that $\varphi_{\mathcal{M}} = f_\omega$. Yet since $f_\omega = \psi$, one has $\varphi_{\mathcal{M}} = \psi$, meaning that ψ is realised by the I-TM/A $\mathcal{M}$. □

6 Discussion

The present paper provides a complete mathematical characterisation of the computational power of evolving recurrent neural networks involved in a basic context of interactive and memory active computation. It is shown that interactive evolving neural networks are computationally equivalent to interactive machines with advice, hence capable of super-Turing potentialities, irrespective of whether their underlying synaptic weights are rational or real.

These results show that the consideration of evolving capabilities in a first-order neural model provides the potentiality to break the Turing barrier. The super-Turing computational equivalence between I-Ev-RNN[$\mathbb{Q}$]s and I-Ev-RNN[$\mathbb{R}$]s reveals two important considerations. First, the incorporation of the power of the continuum in the model does not increase further the computational capabilities of the networks. This feature supports the extension of the Church-Turing Thesis to the context of interactive computation stated by van Leeuwen and Wiedermann [16]:

> "Any (non-uniform interactive) computation can be described in terms of interactive Turing machines with advice."

Second and most importantly, the super-Turing computational capabilities can be achieved without the need of a framework based on the power of the continuum – in the case of the Ev-RNN[$\mathbb{Q}$] model. This feature is particularly meaningful, since while the power of the continuum is a pure conceptualisation of the mind, the evolving capabilities of the networks are, by contrast, really observable in nature. However, note 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). The question of the existence in nature of such non-recursive patterns of evolution remains beyond the scope of this paper. We refer to Copeland's extensive work for deeper philosophical considerations about hypercomputation in general [4].

From a general perspective, we believe that such theoretical studies about the computational power of bio-inspired neural models might ultimately bring further insight to the understanding of the intrinsic natures of both biological as well as artificial intelligences. We also think that foundational approaches to alternative models of computation might in the long term not only lead to relevant theoretical considerations, but also to important practical applications. Similarly to the theoretical work from Turing which played a crucial role in the practical realisation of modern computers, further foundational considerations of alternative models of computation will certainly contribute to the emergence of novel computational technologies and computers, and step by step, open the way to the next computational era.

References

1. Cabessa, J.: Interactive evolving recurrent neural networks are super-turing. In: Filipe, J., Fred, A.L.N. (eds.) ICAART (1), pp. 328–333. SciTePress (2012)
2. Cabessa, J., Siegelmann, H.T.: Evolving recurrent neural networks are super-turing. In: IJCNN, pp. 3200–3206. IEEE (2011)
3. Cabessa, J., Siegelmann, H.T.: The computational power of interactive recurrent neural networks. Neural Computation 24(4), 996–1019 (2012)
4. Jack Copeland, B.: Hypercomputation. Minds Mach. 12(4), 461–502 (2002)
5. Goldin, D., Smolka, S.A., Wegner, P.: Interactive Computation: The New Paradigm. Springer-Verlag New York, Inc., Secaucus (2006)
6. Kleene, S.C.: Representation of events in nerve nets and finite automata. In: Shannon, C., McCarthy, J. (eds.) Automata Studies, pp. 3–41. Princeton University Press, Princeton (1956)
7. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysic 5, 115–133 (1943)
8. Minsky, M.L.: Computation: finite and infinite machines. Prentice-Hall, Inc., Englewood Cliffs (1967)
9. von Neumann, J.: The computer and the brain. Yale University Press, New Haven (1958)
10. Siegelmann, H.T.: Neural networks and analog computation: beyond the Turing limit. Birkhauser Boston Inc., Cambridge (1999)
11. Siegelmann, H.T., Sontag, E.D.: Analog computation via neural networks. Theor. Comput. Sci. 131(2), 331–360 (1994)
12. Siegelmann, H.T., Sontag, E.D.: On the computational power of neural nets. J. Comput. Syst. Sci. 50(1), 132–150 (1995)
13. Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proc. London Math. Soc. 2(42), 230–265 (1936)
14. Turing, A.M.: Intelligent machinery. Technical report, National Physical Laboratory, Teddington, UK (1948)
15. van Leeuwen, J., Wiedermann, J.: Beyond the Turing Limit: Evolving Interactive Systems. In: Pacholski, L., Ružička, P. (eds.) SOFSEM 2001. LNCS, vol. 2234, pp. 90–109. Springer, Heidelberg (2001)
16. van Leeuwen, J., Wiedermann, J.: The Turing machine paradigm in contemporary computing. In: Engquist, B., Schmid, W. (eds.) Mathematics Unlimited - 2001 and Beyond. LNCS, pp. 1139–1155. Springer, Heidelberg (2001)
17. Wiedermann, J., van Leeuwen, J.: How we think of computing today. In: Beckmann, A., Dimitracopoulos, C., Löwe, B. (eds.) CiE 2008. LNCS, vol. 5028, pp. 579–593. Springer, Heidelberg (2008)
18. Wegner, P.: Interactive foundations of computing. Theor. Comput. Sci. 192, 315–351 (1998)