Interactive Evolving Recurrent Neural Networks Are Super-Turing Universal

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

[1] Laboratory of Mathematical Economics (LEMMA),
University of Paris 2 – Panthéon-Assas,
4 Rue Blaise Desgoffe,
75006 Paris, France
[2] Neuroheuristic Research Group,
Department of Information Systems,
University of Lausanne,
1015 Lausanne, Switzerland

Abstract. Understanding the dynamical and computational capabilities of neural models represents an issue of central importance. In this context, recent results show that interactive evolving recurrent neural networks are super-Turing, irrespective of whether their synaptic weights are rational or real. We extend these results by showing that interactive evolving recurrent neural networks are not only super-Turing, but also capable of simulating any other possible interactive deterministic system. In this sense, interactive evolving recurrent neural networks represents a super-Turing universal model of computation, irrespective of whether their synaptic weights are rational or real.

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

1 Introduction

Understanding the dynamical and computational capabilities of neural models represents an issue of central importance. In this context, much interest has been focused on comparing the computational capabilities of diverse theoretical neural models to those of abstract computing devices, see [9,17,8,11,10,13,14,12,2] as well as [15]. 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, and capable to range from finite state automata up to super-Turing capabilities.

In this global line of thinking, the computational capabilities of neural models have generally been characterised with respect to the classical computational framework introduced by Turing [16]. But this approach is inherently restrictive, and has nowadays been argued to "no longer fully corresponds to the current notion of computing in modern systems" [21], especially when it refers to bio-inspired complex information processing systems [18,21]. Indeed, in the brain (or

S. Wermter et al. (Eds.): ICANN 2014, LNCS 8681, pp. 57–64, 2014.

in organic life in general), information is rather processed in an interactive way, where previous experience must affect the perception of future inputs, and where older memories may themselves change with response to new inputs. Accordingly, the computational power of recurrent neural networks should rather be conceived from the perspective of *interactive computation* [22,7,20].

In this context, the capabilities of neural networks involved in an interactive computational paradigm have recently been studied [3,1,4,5]. It was proven that interactive static rational- and real-weighted recurrent neural networks are Turing equivalent and super-Turing, respectively [3,4], and that interactive evolving recurrent neural networks are super-Turing, irrespective of whether their synaptic weights are rational or real [1,5].

The present paper extends these results by showing that interactive evolving recurrent neural networks are not only super-Turing, but also capable of simulating any other possible interactive deterministic system. In this sense, interactive evolving recurrent neural networks represents a universal super-Turing model of computation, irrespective of whether their synaptic weights are modelled by rational or real numbers.

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. Let also $\Sigma^{\leq\omega} = \Sigma^* \cup \Sigma^\omega$ be the set of all possible words (finite or infinite) over Σ. The empty word is denoted 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 (if $x = y$, then $y - x = \lambda$). 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 $x^0 = \lambda$.

Given some sequence of finite words $\{x_i : i \in \mathbb{N}\}$ such that $x_i \subseteq x_{i+1}$ for all $i \geq 0$, one defines the *limit* of the x_i's, denoted by $\lim_{i\geq 0} x_i$, as the unique finite or infinite word which is ultimately approached by the sequence of growing prefixes $\{x_i : i \geq 0\}$. Formally, if the sequence $\{x_i : i \in \mathbb{N}\}$ is eventually constant, i.e. there exists an index $i_0 \in \mathbb{N}$ such that $x_j = x_{i_0}$ for all $j \geq i_0$, then $\lim_{i\geq 0} x_i = x_{i_0}$, meaning that $\lim_{i\geq 0} x_i$ corresponds to the smallest finite word containing each word of $\{x_i : i \in \mathbb{N}\}$ as a finite prefix; if the sequence $\{x_i : i \in \mathbb{N}\}$ is not eventually constant, then $\lim_{i\geq 0} x_i$ corresponds to the unique infinite word containing each word of $\{x_i : i \in \mathbb{N}\}$ as a finite prefix.

Besides, a function $f : \Sigma^* \to \Sigma^*$ is called *monotone* if the relation $x \subseteq y$ implies $f(x) \subseteq f(y)$, for all $x, y \in \Sigma^*$. Any function $\varphi : \Sigma^\omega \to \Sigma^{\leq\omega}$ mapping infinite words to finite or infinite words will be referred to as an *ω-translation*.

Note that any monotone function $f : \{0,1\}^* \to \{0,1\}^*$ induces "in the limit" an ω-translation $f_\omega : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ defined by $f_\omega(x) = \lim_{i\geq 0} f(x[0{:}i])$ for all $x \in \{0,1\}^\omega$. The monotonicity of f ensures that the value $f_\omega(x)$ is well-defined for all $x \in \{0,1\}^\omega$. In words, the value $f_\omega(x)$ corresponds to the finite or infinite word ultimately approached by the sequence of growing prefixes $\{f(x[0{:}i]) : i \geq 0\}$. Finally, 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 $f_\omega = \psi$.

3 Interactive Computation

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 modelled by potentially infinite sequences of symbols (the case of finite sequences of symbols would necessarily reduce to the classical computational framework) [22,7,21].

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 [20]. Consequently, after infinitely many time steps, the system will have received an infinite sequence of consecutive input bits s and translated it into a corresponding finite or infinite sequence of not necessarily consecutive (non-λ) output bits o_s. Note that when the system is deterministic, the output stream o_s associated to the input stream s is necessarily unique.

Accordingly, any interactive deterministic system $\mathcal{S}$ realises an ω-translation $\varphi_\mathcal{S} : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ defined by $\varphi_\mathcal{S}(s) = o_s$, for every $s \in \{0,1\}^\omega$. An ω-translation ψ is then called *interactively computable* iff there exists an interactive deterministic system $\mathcal{S}$ such that $\varphi_\mathcal{S} = \psi$. Note that in this definition, we do absolutely not require for the system $\mathcal{S}$ to be driven by a Turing program nor to contain any computable component of whatever kind. We simply require that $\mathcal{S}$ is deterministic and performs ω-translations in conformity with our interactive paradigm described above.

Van Leeuwen and Widermann introduced the concepts of an *interactive Turing machine* (I-TM) and an *interactive Turing machine with advice* (I-TM/A) as relevant extensions of their classical counterparts to the context of interactive computation. Interactive Turing machines with advice were shown to be strictly more powerful than interactive Turing machines (without advice) [18,19], and computationally equivalent to several other non-uniform models of interactive computation, like sequences of interactive finite automata, site machines, web Turing machines [18], and more recently to interactive analog neural networks and interactive evolving neural networks [3,5].

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 N internal neurons $(x_i)_{i=1}^N$, M parallel input cells $(u_i)_{i=1}^M$, and P designated output neurons among the N. The dynamics of the network is computed as follows: 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 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 $x > 1$.

An *interactive evolving recurrent neural network* (I-Ev-RNN) $\mathcal{N}$ consists of an Ev-RNN provided with a single binary input cell u as well as two binary output cells: a data cell y_d and a validation cell y_v. Any infinite *input stream* $s = s(0)s(1)s(2)\cdots \in \{0,1\}^\omega$ transmitted to the input cell 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 $o_s = \langle y_d(i) : i \in \mathbb{N} \text{ and } y_v(i) = 1\rangle \in \{0,1\}^{\leq\omega}$. Hence, 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 $\varphi_{\mathcal{N}} = \psi$.

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

In a previous paper [5], we proved that interactive evolving recurrent neural networks were computationally equivalent to interactive Turing machines with advice, hence capable of a super-Turing computational power. Here, we extend this result by showing that interactive evolving recurrent neural networks are *super-Turing universal*, in the sense of being capable to realise any possible interactively computable ω-translation. Formally, for any possible interactively

computable ω-translation ψ, there exists some I-Ev-RNN $\mathcal{N}$ such that $\varphi_{\mathcal{N}} = \psi$. Equivalently, for any interactive deterministic system $\mathcal{S}$, there exists some I-Ev-RNN $\mathcal{N}$ such that $\varphi_{\mathcal{N}} = \varphi_{\mathcal{S}}$. In words, any interactive deterministic system can be simulated by some interactive evolving recurrent neural networks.

Theorem 1. *Interactive evolving recurrent neural networks are super-Turing universal, irrespective of whether their synaptic weights are rational or real.*

Proof. We prove the result in two steps. Firstly, we show that any interactively computable ω-translation is continuous. Secondly, we prove that any continuous ω-translation is realisable by some I-Ev-RNN[$\mathbb{Q}$]. Consequently, for any interactively computable ω-translation ψ, there exists some I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}$ such that $\varphi_{\mathcal{N}} = \psi$, meaning that I-Ev-RNN[$\mathbb{Q}$]s are super-Turing universal. Finally, if I-Ev-RNN[$\mathbb{Q}$]s are super-Turing universal, then so are I-Ev-RNN[$\mathbb{R}$]s, since I-Ev-RNN[$\mathbb{R}$]s are more powerful than I-Ev-RNN[$\mathbb{Q}$]s. We now give the proofs of steps 1 and 2.

Step 1: Let ψ be some interactively computable ω-translation. Then by definition, there exists an interactive deterministic system $\mathcal{S}$ such that $\varphi_{\mathcal{S}} = \psi$. Now, consider the function $f : \{0,1\}^* \to \{0,1\}^*$ which maps every finite word u to the unique corresponding finite word produced by $\mathcal{S}$ after exactly $|u|$ steps of computation over input stream u provided bit by bit. The deterministic nature of $\mathcal{S}$ ensures that the finite word $f(u)$ is unique, and thus that f is well-defined.

We show that f is monotone. Suppose that $u \subseteq v$. It follow that $v = u \cdot (v - u)$. Hence, according to our interactive paradigm, the output strings produced by $\mathcal{S}$ after $|v|$ time steps of computation over input stream v, namely $f(v)$, simply consists of the output string produced after $|u|$ time steps of computation over input u, namely $f(u)$, followed by the output string produced after $|v - u|$ time steps of computation over input $v - u$. Consequently, $f(u) \subseteq f(v)$.

We now prove that the ω-translation $\varphi_{\mathcal{S}}$ performed by $\mathcal{S}$ satisfies $\varphi_{\mathcal{S}} = f_\omega$. Towards this purpose, given some infinite input stream $s \in \{0,1\}^\omega$, we consider in turn the two possible cases where either $\varphi_{\mathcal{S}}(s) \in \{0,1\}^\omega$ or $\varphi_{\mathcal{S}}(s) \in \{0,1\}^*$.

Firstly, suppose that $\varphi_{\mathcal{S}}(s) \in \{0,1\}^\omega$. According to our interactive scenario, $f(s[0{:}i])$ is a prefix of $\varphi_{\mathcal{S}}(s)$, for all $i \geq 0$. Moreover, since $\varphi_{\mathcal{S}}(s) \in \{0,1\}^\omega$, the sequence of partial output strings produced by $\mathcal{S}$ on input s after i time steps of computation cannot be eventually constant, i.e. $\lim_{i\to\infty} |f(s[0{:}i])| = \infty$. Hence, the two properties $f(s[0{:}i]) \subseteq \varphi_{\mathcal{S}}(s) \in \{0,1\}^\omega$ for all $i \geq 0$ and $\lim_{i\to\infty} |f(s[0{:}i])| = \infty$ ensure that $\varphi_{\mathcal{S}}(s)$ is the unique infinite word containing each word of $\{f(s[0{:}i]) : i \geq 0\}$ as a finite prefix, which is to say by definition that $\varphi_{\mathcal{S}}(s) = \lim_{i\geq 0} f(s[0{:}i]) = f_\omega(s)$.

Secondly, suppose that $\varphi_{\mathcal{S}}(s) \in \{0,1\}^*$. Once again, one has that $f(s[0{:}i])$ is a prefix of $\varphi_{\mathcal{S}}(s)$, for all $i \geq 0$. Moreover, since $\varphi_{\mathcal{S}}(s) \in \{0,1\}^*$, the sequence of partial output strings produced by $\mathcal{S}$ on input s after i time steps of computation must become stationary from some time step j onwards, i.e. $\lim_{i\to\infty} |f(s[0{:}i])| < \infty$. Hence, the entire finite output stream $\varphi_{\mathcal{S}}(s)$ must necessarily have been produced after a finite amount of time, and thus $\varphi_{\mathcal{S}}(s) \in \{f(s[0{:}i]) : i \geq 0\}$. Consequently, the three properties $f(s[0{:}i]) \subseteq \varphi_{\mathcal{S}}(s) \in \{0,1\}^*$ for all $i \geq 0$, $\lim_{i\to\infty} |f(s[0{:}i])| < \infty$, and $\varphi_{\mathcal{S}}(s) \in \{f(s[0{:}i]) : i \geq 0\}$ ensure that $\varphi_{\mathcal{S}}(s)$ is

the smallest finite word that contains each word of $\{f(s[0{:}i]) : i \geq 0\}$ as a finite prefix, which is to say by definition that $\varphi_{\mathcal{S}}(s) = \lim_{i \geq 0} f(s[0{:}i]) = f_\omega(s)$. Consequently, $\varphi_{\mathcal{S}}(s) = f_\omega(s)$ for any $s \in \{0,1\}^\omega$, meaning that $\varphi_{\mathcal{S}} = f_\omega$.

We proved that f is a monotone function such that $\varphi_{\mathcal{S}} = f_\omega$. Hence, $\varphi_{\mathcal{S}}$ is continuous. Since $\varphi_{\mathcal{S}} = \psi$, it follows that ψ is also continuous.

Step 2: Let ψ be a continuous ω-translation. Then there exists some monotone function $f : \{0,1\}^* \to \{0,1\}^*$ such that $f_\omega = \psi$. We begin by encoding all possible values of f into successive distinct rational numbers. For any $n > 0$, let $w_{n,1}, \ldots, w_{n,2^n}$ be the lexicographical enumeration of $\{0,1\}^n$, and let $w_n \in \{0,1,2\}^*$ be the finite word given by $w_n = 2 \cdot f(w_{n,1}) \cdot 2 \cdot f(w_{n,2}) \cdot 2 \cdots 2 \cdot f(w_{n,2^n}) \cdot 2$. Then, consider the rational encoding q_n of the word w_n given by $q_n = \sum_{i=1}^{|w_n|} \frac{2 \cdot w_n(i)+1}{6^i}$. Note that for all $n > 0$, one has $q_n \in]0,1[$ and $q_n \neq q_{n+1}$, since $w_n \neq w_{n+1}$. Moreover, it can be shown that w_n can be decoded from q_n by some Turing machine, or equivalently, by some rational recurrent neural network [13,14].

Now, consider Procedure 1 below. Note that its only non-recursive instruction is "wait for next value q_{i+1} to come". We show that there exists some I-Ev-RNN[$\mathbb{Q}$] $\mathcal{N}$ that performs Procedure 1. The network $\mathcal{N}$ consists of one evolving and one non-evolving rational sub-network connected together. The evolving rational-weighted part of $\mathcal{N}$ is made up of a single processor x_e receiving a background activity of evolving intensity $c_e(t)$. The synaptic weight $c_e(t)$ takes the successive values $q_1, q_2, q_3, \ldots$, by switching from value q_k to q_{k+1} after every N_k time steps, for some large enough $N_k > 0$ to be described. The non-evolving rational-weighted part of $\mathcal{N}$ is designed in order to perform the successive recursive steps of Procedure 1 every time neuron x_e receives some new activation value q_k [14]. For each $k > 0$, the time interval N_k is chosen large enough in order for $\mathcal{N}$ to be able to perform all such successive steps before the apparition of the next value q_{k+1}. Moreover, the network $\mathcal{N}$ outputs the current pair $(v - u, 1^{|v-u|})$ bit by bit every time it reaches up the instructions "$p_s \leftarrow p_s \cdot (v - u)$" and "$q_s \leftarrow q_s \cdot 1^{|v-u|}$", and it keeps outputting pairs of bits $(0,0)$s meanwhile.

Procedure 1.

Infinite input stream $s = s(0)s(1)s(2)\cdots \in \{0,1\}^\omega$ provided bit by bit
$i \leftarrow 0$; $u \leftarrow \lambda$; $v \leftarrow \lambda$; $p_s \leftarrow \lambda$; $q_s \leftarrow \lambda$;
loop
  Wait for next value q_{i+1} to come; Decode $f(s[0{:}i])$ from q_{i+1}; $v \leftarrow f(s[0{:}i])$;
  if $u \subsetneq v$ **then**
    $p_s \leftarrow p_s \cdot (v - u)$; $q_s \leftarrow q_s \cdot 1^{|v-u|}$;
  else
    $p_s \leftarrow p_s \cdot 0$; $q_s \leftarrow q_s \cdot 0$;
  end if
  $i \leftarrow i + 1$; $u \leftarrow v$;
end loop

It remains to prove that $\varphi_{\mathcal{N}} = \psi$. Note that, for any input stream $s \in \{0,1\}^\omega$, the finite word that has been output by $\mathcal{N}$ at the end of each instruction "output $v-u$ bit by bit" corresponds precisely to the finite word $f(s[0{:}i])$ currently stored in the variable v. Hence, after infinitely many time steps, the finite or infinite word $\varphi_{\mathcal{N}}(s)$ output by $\mathcal{N}$ contains all words of $\{f(x[0{:}i]) : i \geq 0\}$ as a finite prefix. Moreover, if $\varphi_{\mathcal{N}}(s)$ is finite, its value necessarily corresponds to some current content of the variable v, i.e to some finite word $f(s[0{:}j])$, for some $j \geq 0$. Hence, irrespective of whether $\varphi_{\mathcal{N}}(s)$ is finite or infinite, one always has $\varphi_{\mathcal{N}}(s) = \lim_{i \geq 0} f(x[0{:}i]) = f_\omega(s)$, for any $s \in \{0,1\}^\omega$. Therefore, $\varphi_{\mathcal{N}} = f_\omega = \psi$, meaning that ψ is realised by $\mathcal{N}$. □

By putting together previous Theorem 1 and Theorem 1 of [5], one obtains the following complete characterisation of the computational power of interactive evolving recurrent neural networks.

Theorem 2. *Let $\psi : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ be an ω-translation. The following conditions are equivalent:*

1. *ψ is interactively computable;*
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 realisable by some I-TM/A;*
5. *ψ is continuous.*

Proof. The equivalences between 2, 3, 4, and 5 are proven in [5]. The implication $1 \Rightarrow 2$ is stated in Theorem 1, and the implication $2 \Rightarrow 1$ holds by definition. □

6 Discussion

Interactive evolving neural networks (I-Ev-RNNs) are computationally equivalent to interactive machines with advice (I-TM/As), hence capable of super-Turing potentialities, irrespective of whether their synaptic weights are rational or real [5]. They are also capable of simulating any other possible interactive deterministic system. In this sense, I-Ev-RNNs and I-TM/As represent two equivalent super-Turing universal models of computation.

These results can be understood as follows: similarly to the classical context, where every possible partial function from integers to integers can be computed by some Turing machine with oracle [16], in the interactive context, every possible ω-translation performed in an interactive way can be computed by some interactive Turing machine with advice, or equivalently, by some interactive evolving recurrent neural network. These results support the extension of the Church-Turing Thesis to the context of interactive computation stated by van Leeuwen and Wiedermann [19]: "Any (non-uniform interactive) computation can be described in terms of interactive Turing machines with advice."

The question of the possible achievement of such super-Turing capabilities by real biological neural networks remains beyond the scope of this paper. We refer to Copeland's extensive work for deeper philosophical considerations about hypercomputation in general [6].

References

1. Cabessa, J.: Interactive evolving recurrent neural networks are super-Turing. In: ICAART 2012, pp. 328–333. SciTePress (2012)
2. Cabessa, J., Siegelmann, H.T.: Evolving recurrent neural networks are super-turing. In: IJCNN 2011, 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. Cabessa, J., Villa, A.E.P.: The expressive power of analog recurrent neural networks on infinite input streams. Theor. Comput. Sci. 436, 23–34 (2012)
5. Cabessa, J., Villa, A.E.P.: The super-turing computational power of interactive evolving recurrent neural networks. In: Mladenov, V., Koprinkova-Hristova, P., Palm, G., Villa, A.E.P., Appollini, B., Kasabov, N. (eds.) ICANN 2013. LNCS, vol. 8131, pp. 58–65. Springer, Heidelberg (2013)
6. Copeland, B.J.: Hypercomputation. Minds Mach. 12(4), 461–502 (2002)
7. Goldin, D., Wegner, P.: Principles of interactive computation. In: Goldin, D., Smolka, S.A., Wegner, P. (eds.) Interactive Computation, pp. 25–37. Springer (2006)
8. 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 (1956)
9. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysic 5, 115–133 (1943)
10. Minsky, M.L.: Computation: finite and infinite machines. Prentice-Hall, Inc. (1967)
11. von Neumann, J.: The computer and the brain. Yale University Press (1958)
12. Siegelmann, H.T.: Neural networks and analog computation: beyond the Turing limit. Birkhauser Boston Inc. (1999)
13. Siegelmann, H.T., Sontag, E.D.: Analog computation via neural networks. Theor. Comput. Sci. 131(2), 331–360 (1994)
14. Siegelmann, H.T., Sontag, E.D.: On the computational power of neural nets. J. Comput. Syst. Sci. 50(1), 132–150 (1995)
15. Síma, J., Orponen, P.: General-purpose computation with neural networks: A survey of complexity theoretic results. Neural Comput. 15(12), 2727–2778 (2003)
16. Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proc. London Math. Soc. 2(42), 230–265 (1936)
17. Turing, A.M.: Intelligent machinery. Technical report, National Physical Laboratory, Teddington, UK (1948)
18. 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)
19. van Leeuwen, J., Wiedermann, J.: The Turing machine paradigm in contemporary computing. In: Engquist, B., Schmid, W. (eds.) Mathematics Unlimited - 2001 and Beyond, pp. 1139–1155. Springer (2001)
20. van Leeuwen, J., Wiedermann, J.: A theory of interactive computation. In: Goldin, D., Smolka, S.A., Wegner, P. (eds.) Interactive Computation, pp. 119–142. Springer (2006)
21. 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)
22. Wegner, P.: Interactive foundations of computing. Theor. Comput. Sci. 192, 315–351 (1998)