Recurrent Neural Networks
A Natural Model of Computation beyond the Turing Limits

Jérémie Cabessa and Alessandro E. P. Villa
Department of Information Systems, University of Lausanne, CH-1015 Lausanne, Switzerland

Keywords: Neural Computation, Analog Computation, Interactive Computation, Recurrent Neural Networks, Super-Turing.

Abstract: According to the Church-Turing Thesis, the classical Turing machine model is capable of capturing all possible aspects of algorithmic computation. However, in neural computation, several basic neural models were proven to be capable of computational capabilities located beyond the Turing limits. In this context, we present an overview of recent results concerning the super-Turing computational power of recurrent neural networks, and show that recurrent neural networks provide a suitable and natural model of computation beyond the Turing limits. We nevertheless don't draw any hasty conclusion about the controversial issue of a possible predominance of biological intelligence over the potentialities of artificial intelligence.

1 INTRODUCTION

Understanding the intrinsic nature of biological intelligence appears to be among the most challenging issues, with considerable repercussions ranging from theoretical and philosophical considerations to practical implications in the fields of artificial intelligence, machine learning, bio-inspired computing, etc. In this context, much interest has been focused on comparing the computational capabilities of brain-like models and abstract machines.

This comparative approach has been initiated by (McCulloch and Pitts, 1943) who proposed a modelization of the nervous system as a finite interconnection of logical devices. For the first time, neural networks were considered as discrete abstract machines, and the issue of their computational capabilities investigated from the automata-theoretic perspective. These considerations were further pursued by (Kleene, 1956), (Von Neumann, 1958), (Rosenblatt, 1957), (Minsky, 1967), and (Minsky and Papert, 1969) who opened up the way to the theoretical approach to neural computation.

Along these lines, it is nowadays well-known that, depending on several aspects under consideration, the computational power of diverse neural models may notably range from finite state automata up to super-Turing capabilities.

Here, we present an overview of recent results concerning the super-Turing computational power of recurrent neural networks, and show that recurrent neural networks provide a natural model of computation beyond the Turing limits. More precisely, the specific super-Turing model of a "Turing machine with advice" seems to have a natural fit to capture the computational capabilities of basic brain-like models, capable of capturing some biological mechanisms that cannot be apprehended by the classical Turing machine model but are most significantly involved in neural information processing.

We nevertheless don't draw any hasty conclusion about the controversial issue of a possible predominance of biological intelligence over the potentialities of artificial intelligence. Still, we believe that such an approach might in the long term improve the understanding of the intrinsic natures of both biological and artificial intelligences.

2 CLASSICAL AND INTERACTIVE COMPUTATION

The classical Turing paradigm of computation corresponds to the computational scenario where a system receives a finite input, processes this input, and either provides a corresponding output or never halts.

In this framework, the concept of a *Turing machine* (TM) provides a relevant model of computation to understand the limits of mechanical computa-

Cabessa J. and E. P. Villa A..
Recurrent Neural Networks - A Natural Model of Computation beyond the Turing Limits.
DOI: 10.5220/0004172905940599
In *Proceedings of the 4th International Joint Conference on Computational Intelligence* (NCTA-2012), pages 594-599
ISBN: 978-989-8565-33-4

tion (Turing, 1936)[1]. According to the Church-Turing Thesis, the Turing machine model actually captures all possible aspects of algorithmic computation.

The concept of a *Turing machine with advise* (TM/A) provides a model of computation beyond the Turing limits which plays an important role in the computational power of recurrent neural networks. It consists of a classical Turing machine provided with an additional advise function $\alpha : \mathbb{N} \longrightarrow \{0,1\}^+$ as well as an additional advise tape, and such that, on every input u of length n, the machine first copies the advise word $\alpha(n)$ on its advise tape and then continues its computation according to its finite Turing program. A *Turing machine with polynomial-bounded advise* (TM/poly(A)) consists of a TM/A whose advice length is bounded by some polynomial. Turing machines with (polynomial) advice are strictly more powerful than Turing machines.

But the classical Turing computational approach has nowadays been argued to "no longer fully corresponds to the current notion of computing in modern systems", especially when it refers to bio-inspired complex information processing systems (Van Leeuwen and Wiedermann, 2001; Van Leeuwen and Wiedermann, 2008). Indeed, in organic life, information is rather processed in an interactive way, where previous experience must affect the perception of future inputs, and where older memories themselves may change with response to new inputs. According to these considerations, the alternative framework of *interactive computation* appears to be particularly relevant for the consideration of natural or biological computational phenomena (Goldin et al., 2006).

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). In most basic scenarios, the environment sends a non-empty input bit to the system at every time step (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.

In this context, an *interactive Turing machine* (I-TM) 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, 2001).

Moreover, an *interactive Turing machine with advice* (I-TM/A) $\mathcal{M}$ consists of an interactive Turing machine provided with an advice mechanism which takes the form of an advice function $\alpha : \mathbb{N} \longrightarrow \{0,1\}^*$ (Van Leeuwen and Wiedermann, 2001). 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. Interactive Turing machines with advice were proved to be strictly more powerful than interactive Turing machines (Van Leeuwen and Wiedermann, 2001).

3 RECURRENT NEURAL NETWORKS

Throughout this paper, a *recurrent neural network* (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, and P designated output neurons among the N whose role is to communicate the output of the network to the environment. 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

[1] We briefly recall that a Turing machine consists of a infinite tape, a head that can read and write on this tape, and a finite program which, according to the current computational state of the machine and the current symbol read by the head, determines the next symbol to be written by the head on the tape, the next move of the head (left or right), and the next computational state of the machine.

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

for $i = 1,\ldots,N$, where all a_{ij}, b_{ij}, and c_i are numbers describing the 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 allow mathematical comparison with the languages computed by abstract models of computation – e.g. Turing machines and Turing machines with advice in our case – the study of the computational power of RNNs involves the consideration of a specific model of RNNs that performs computation and decision of formal languages. For this purpose, (Siegelmann and Sontag, 1995) considered a notion of *formal RNN* which adheres to a rigid encoding of the way binary strings are processed as input and output between the network and the environment.

The nature of the synaptic weights under consideration has been proved to play a fundamental role in the computational power of neural networks. Hence, a recurrent neural network will be called *rational* (denoted by RNN[$\mathbb{Q}$]) if all its synaptic weights are rational numbers. It will be called *real* or *analog* (denoted by RNN[$\mathbb{R}$]) if all its all synaptic weights are real numbers. Since rational numbers are real, note that any rational network is a particular analog network by definition.

Besides this classical neural model, (Cabessa and Siegelmann, 2012) also introduced the concept of an *evolving recurrent neural network* (Ev-RNN) as a recurrent neural network equipped with time-dependent rather than static synaptic weights. Their dynamics are therefore governed by equations of the form

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

for $i = 1,\ldots,N$, where all $a_{ij}(t)$, $b_{ij}(t)$, and $c_i(t)$ are *bounded* and *time-dependent* synaptic weights, and σ is the classical saturated-linear activation function.

Recently, (Cabessa and Siegelmann, 2012) proposed the possibility to consider all these kinds neural networks in the more biologically oriented framework of interactive computation. They introduced the concepts of an *interactive recurrent neural network* (I-RNN) and an *interactive evolving recurrent neural network* (I-Ev-RNN) as a recurrent neural network equipped with only one binary input cell and one binary output cell in order to perform the interactive exchange of information between the network and its environment.

Therefore, all these definitions lead to the considerations of 8 basic models of recurrent neural networks: RNN[$\mathbb{Q}$]s, RNN[$\mathbb{R}$]s, Ev-RNN[$\mathbb{Q}$]s, Ev-RNN[$\mathbb{R}$]s, and their interactive counterparts, namely I-RNN[$\mathbb{Q}$]s, I-RNN[$\mathbb{R}$]s, I-Ev-RNN[$\mathbb{Q}$]s, I-Ev-RNN[$\mathbb{R}$]s. The following sections show that analog and evolving networks provide natural models of computation beyond the Turing limits, both in the classical as well as in the interactive computational frameworks.

4 RATIONAL RNNS

A first significant breakthrough concerning the computational power of recurrent neural networks has been made by (Siegelmann and Sontag, 1995) who chose to focus their attention to more realistic activation functions for the neurons. They showed that extending the activation functions of the cells from boolean to linear-sigmoid drastically increases the computational power of the networks from finite state automata up to Turing capabilities. In other words, they proved that *rational recurrent neural networks* (as presented in Section 3) are computationally equivalent to Turing machines.

Theorem 1. *RNN[$\mathbb{Q}$]s are computationally equivalent to TMs. More precisely, a language L is decidable by some RNN[$\mathbb{Q}$] if and only if L is decidable by some TM (i.e., iff L is recursive).*

(Siegelmann and Sontag, 1995) pointed out several interesting consequences of their result. For instance, the problem of determining if a given neuron ever assumes the value “0” is effectively undecidable, since the halting problem can be reduced to it. The problem of determining whether a dynamical system of the particular form $x(t+1) = \sigma(A\cdot x(t) + c)$ ever reaches an equilibrium point from a given initial state is also effectively undecidable, for it reduces to the halting problem as well. Besides, this result provides a direct proof that higher-order neural networks are computationally equivalent, up to polynomial time, to first-order neural networks (higher-order neural networks can be simulated by Turing machines which in turn can be simulated by first-order neural nets).

(Kilian and Siegelmann, 1996) further generalized the Turing universality of rational neural networks to a broader class of sigmoidal activation functions. The computational equivalence between so-called rational recurrent neural networks and Turing machines has now become standard result in the field.

Recently, (Cabessa and Siegelmann, 2012) provided a direct generalization of this result to the more biologically oriented framework of interactive computation. They introduced the concept of an *interactive recurrent neural network* (I-RNN) (as presented in Section 3), and showed that *interactive rational recurrent neural networks* are computationally equivalent to interactive Turing machines. They also provided a precise mathematical characterization of the translations of bit streams performed by these interactive models of computation.

Theorem 2. *I-RNN[$\mathbb{Q}$]s are computationally equivalent to I-TMs. More precisely, an* ω*-translation* φ *is realizable by some I-RNN[$\mathbb{Q}$] if and only if* φ *is realizable by some I-TM (i.e., iff* φ *is recursive continuous).*

5 ANALOG RNNS

(Siegelmann and Sontag, 1994) achieved another important breakthrough by proposing an approach to the computational power of recurrent neural networks from the perspective of *analog computation*. Following von Neumann considerations, they assumed that the variables appearing in the underlying chemical and physical phenomena could be modeled by continuous rather than discrete numbers. They introduced the concept of an *analog recurrent neural network* as a classical linear-sigmoid neural net equipped with real- instead of rational-weighted synaptic connections (as presented in Section 3). They further showed that analog recurrent neural networks are strictly more powerful than their rational counterparts, hence capable of super-Turing computational capabilities. In fact, the analog networks can achieve unbounded power in exponential time of computation (i.e., are capable of deciding all binary languages), and when restricted to polynomial time of computation, the networks turn out to be computationally equivalent to Turing machines with polynomial-bounded advice, thus deciding the complexity class **P/poly**.[2] Since **P/poly** strictly includes the class **P** and even contains non-recursive languages, the analog networks are capable of super-Turing computational power from polynomial time of computation already.

Theorem 3. *RNN[$\mathbb{R}$]s are super-Turing. More precisely, a language L is decidable in polynomial time by some RNN[$\mathbb{R}$] if and only if L is decidable in polynomial time by some TM/poly(A) (i.e., iff* $L \in$ **P/poly***); furthermore, any language L can be decided in exponential time by some RNN[$\mathbb{R}$].*

This analog information processing model turns out to be capable of capturing the non-linear dynamical properties that are most relevant to brain dynamics, such as rich chaotic behaviors (Kaneko and Tsuda, 2003; Tsuda, 1991; Tsuda, 2001). Moreover, many dynamical and idealized chaotic systems that cannot be described by the universal Turing machine model are also well captured within this analog framework (Siegelmann, 1995). These considerations led Siegelmann and Sontag to propose the concept of analog recurrent neural network as standard in the field of analog computation, similar to that of an universal Turing machine in digital computation. They formulated the so-called Thesis of Analog Computation – an analogous to the Church-Turing thesis, but in the realm of analog computation – stating that no reasonable abstract analog device can be more powerful than first-order analog recurrent neural networks (Siegelmann and Sontag, 1994; Siegelmann, 1995). These results might support the opinion that some intrinsic dynamical and computational capabilities of neurobiological systems lie be beyond the scope of standard artificial models of computation.

(Cabessa and Siegelmann, 2012) provided a generalization of this result to the context of interactive computation. They proved that *interactive analog recurrent neural networks* (as presented in Section 3) are computationally equivalent to interactive Turing machines with advice, and also provided a precise mathematical characterization of the translations of bit streams performed by these interactive models of computation. Hence, in the interactive just as in the classical framework, analog neural networks turn out to reveal super-Turing computational capabilities.

Theorem 4. *I-RNN[$\mathbb{R}$]s are super-Turing. More precisely, I-RNN[$\mathbb{R}$]s are computationally equivalent to I-TM/As, and hence realize uncountably many more* ω*-translations than I-TMs.*

(Cabessa and Villa, 2012) proposed another generalization of this result in a different intereactive-like computational framework. They introduced the concept of an ω-*analog recurrent neural network* (ω-RNN[$\mathbb{R}$]) as an interactive recurrent neural network with real synaptic weights (as presented in Section 3) yet preforming language recognition over the space of infinite streams of bits rather than ω-translations of infinite streams of bits. More precisely, the network receives an infinite input stream of bits from its environment and produces a corresponding output stream of bits. The input stream is then said to be accepted by the network if the corresponding output remains forever active, i.e. never shuts down to 0 from some time step onwards. The language recognized by the network is then defined as the set of input streams that are accepted by the network. In this

[2] The complexity classes **P** and **P/poly** correspond to the sets of languages decided in polynomial time by TMs and TM/poly(A), respectively.

context, Cabessa and Villa provided a precise characterization of the expressive power of analog neural networks, and showed that analog recurrent neural networks turn out to be strictly more expressive that deterministic and non-deterministic Turing machines equipped with Büchi or Muller accepting conditions.

Theorem 5. *Determnistic* ω*-RNN[*$\mathbb{R}$*]s are strictly more expressive than deterministic Büchi TMs. Non-determnistic* ω*-RNN[*$\mathbb{R}$*]s are strictly more expressive than non-deterministic Büchi or Muller TMs.*

6 EVOLVING RNNS

The brain computes, but it does so differently than today's computers. Neural memories are updated when being retrieved in a process called reconsolidation which causes adaptation to changing conditions; the geometric architecture itself changes continuously as well, with synapses updating their connectivity patterns all the time; current levels of hormonal and chemical concentrations change constantly and affect the computation performed by the neural architecture. But until recently, these crucial biological considerations have generally been neglected in the classical literature concerning the computational capabilities of brain-like models. Hence, the following questions naturally arise: Can we approach the issue of the brain's capabilities from a non-static perspective? Can we understand and characterize the computational capabilities of an ever-changing neural model?

According to these considerations, (Cabessa and Siegelmann, 2011) considered a new approach to the computational power of neural networks from the perspective of *evolving systems*. They introduced a more biologically-oriented model of *evolving recurrent neural networks* (as presented in Section 3) where the synaptic weights can evolve rather than stay static. They further proved that both models of *evolving rational neural networks* and *evolving real (or analog) neural networks* are actually computationally equivalent to static analog networks, thus capable of super-Turing computational capabilities from polynomial time of computation already.

Theorem 6. *Ev-RNN[*$\mathbb{Q}$*]s and Ev-RNN[*$\mathbb{R}$*]s are super-Turing. Both models are computationally equivalent to RNN[[*$\mathbb{R}$*]s.*

Theorems 1, 3, and 6 show that when stepping from the static to the evolving context, the computational power of rational neural networks turn out to be drastically increased from the Turing to the super-Turing level, whereas the computational capabilities of analog neural networks actually remain at the same super-Turing level, equivalent to that of static analog neural networks. These results support once again the Thesis of Analog Computation stating that no reasonable abstract analog device can be more powerful than first-order analog recurrent neural networks (Siegelmann and Sontag, 1994; Siegelmann, 1995).

Moreover, Theorem 6 shows that the consideration of *architectural evolving capabilities* in a basic first-order rate neural model provides an alternative and equivalent way to the consideration of the *power of the continuum* towards the achievement of super-Turing computational capabilities. This feature is particularly interesting since it allows to replace the controversial "analog assumption" by natural "evolving considerations" towards the achievement of super-Turing computational capabilities of neural networks. These results also emphasizes the role that the mechanisms of evolution and plasticity might indeed play in the computational capabilities of neural networks.

It is worth noting that the super-Turing capabilities of evolving neural networks 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.

Besides, (Cabessa, 2012) generalized once again these results to the context of interactive computation. He proved that both models of *rational- and real-weighted interactive evolving neural networks* (as presented in Section 3) are computationally equivalent to interactive Turing machines with advice, and hence capable of super-Turing capabilities.

Theorem 7. *I-Ev-RNN[*$\mathbb{Q}$*]s and I-Ev-RNN[*$\mathbb{R}$*]s are super-Turing. Both models are computationally equivalent to I-RNN[[*$\mathbb{R}$*]s, hence to I-TM/As, and thus realize uncountably many more* ω*-translations than I-TMs.*

7 CONCLUSIONS

Theorems 3, 4, 5, 6, and 7 show that recurrent neural networks provide a natural model of computation beyond the Turing limits. The specific super-Turing model of a Turing machine with advice seems to have a natural fit to capture the computational capabilities of basic brain-like models. Such model provides the possibility to capture analog and/or evolving considerations that cannot be apprehended by the classical Turing machine model yet play a crucial role in many aspects of neural computation

According to (Siegelmann, 2003), such results “embeds a possible answer to the superiority of biological intelligence within the framework of classical computer science”. We prefer to remain cautious on this issue, and do not intend to argue in favor of an ontological existence of super-Turing capabilities of biological neural networks in nature, but rather in favor of the relevance of considering super-Turing neural models in order to describe neurobiological features that fail be captured via Turing-equivalent classical models of computation. In this sense, we believe that the consideration of super-Turing brain-like computational models present some interest beyond the philosophical controversial considerations about hypercomputation (Copeland, 2004).

Finally, we expect that such theoretical studies about the computational power of neural models might lead to a better understanding of the basic principles that underlie the processing of information in the brain. In this context, we believe that the comparative approach between the computational powers of bio-inspired and artificial abstract models of computation shall ultimately provide a better understanding of the intrinsic natures of both biological and artificial intelligences. We further believe that the foundational approach to alternative models of computation might in the long term not only lead to relevant theoretical considerations, but also to important practical implications. Similarly to the theoretical work from Turing which played a crucial role in the practical realization 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

Cabessa, J. (2012). Interactive evolving recurrent neural networks are super-Turing. In Filipe, J. and Fred, A., editors, *ICAART 2012: Proceedings of the 4th International Conference on Agents and Artificial Intelligence 2012*, volume 1, pages 328–333. SciTePress.

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

Cabessa, J. and Siegelmann, H. T. (2012). The computational power of interactive recurrent neural networks. *Neural Comput.*, 24(4):996–1019.

Cabessa, J. and Villa, A. E. (2012). The expressive power of analog recurrent neural networks on infinite input streams. *Theor. Comput. Sci.*, 436:23–34.

Copeland, B. J. (2004). Hypercomputation: philosophical issues. *Theor. Comput. Sci.*, 317(1–3):251–267.

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

Kaneko, K. and Tsuda, I. (2003). Chaotic itinerancy. *Chaos*, 13(3):926–936.

Kilian, J. and Siegelmann, H. T. (1996). The dynamic universality of sigmoidal neural networks. *Inf. Comput.*, 128(1):48–56.

Kleene, S. C. (1956). Representation of events in nerve nets and finite automata. In *Automata Studies*, volume 34 of *Annals of Mathematics Studies*, pages 3–42. Princeton University Press, Princeton, N. J.

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. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press.

Minsky, M. L. (1967). *Computation: finite and infinite machines*. Prentice-Hall, Inc.

Rosenblatt, F. (1957). The perceptron: A perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, Ithaca, New York.

Siegelmann, H. T. (1995). Computation beyond the Turing limit. *Science*, 268(5210):545–548.

Siegelmann, H. T. (2003). Neural and super-Turing computing. *Minds Mach.*, 13(1):103–114.

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.

Tsuda, I. (1991). Chaotic itinerancy as a dynamical basis of hermeneutics of brain and mind. *World Futures*, 32:167–185.

Tsuda, I. (2001). Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems. *Behav. Brain Sci.*, 24(5):793–847.

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 2(42):230–265.

Van Leeuwen, J. and Wiedermann, J. (2001). Beyond the Turing limit: Evolving interactive systems. In Pacholski, L. and Ružicka, P., editors, *SOFSEM 2001: Theory and Practice of Informatics*, volume 2234 of *LNCS*, pages 90–109. Springer-Verlag.

Van Leeuwen, J. and Wiedermann, J. (2008). How we think of computing today. In Beckmann, A., Dimitracopoulos, C., and Löwe, B., editors, *Logic and Theory of Algorithms*, volume 5028 of *LNCS*, pages 579–593. Springer-Verlag.

Von Neumann, J. (1958). *The computer and the brain*. Yale University Press, New Haven, CT, USA.

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