

Interactive Evolving Recurrent Neural Networks are Super-Turing

Jérémie Cabessa

Department of Information Systems
University of Lausanne
Switzerland

8 February 2012

Introduction

- ▶ *Interactive computation* captures the notion of bio-inspired computing better than classical Turing-like computation.
- ▶ *Architectural Evolution* is a crucial aspect of biological neural networks: synaptic plasticity, cell birth and death, . . .
- ▶ We studied the computational capabilities of a rate model of *interactive* and *evolving* recurrent neural networks.
- ▶ We proved that *Interactive* and *evolving* recurrent neural networks are super-Turing.

Introduction

- ▶ *Interactive computation* captures the notion of bio-inspired computing better than classical Turing-like computation.
- ▶ *Architectural Evolution* is a crucial aspect of biological neural networks: synaptic plasticity, cell birth and death, ...
- ▶ We studied the computational capabilities of a rate model of *interactive* and *evolving* recurrent neural networks.
- ▶ We proved that *Interactive* and *evolving* recurrent neural networks are super-Turing.

Introduction

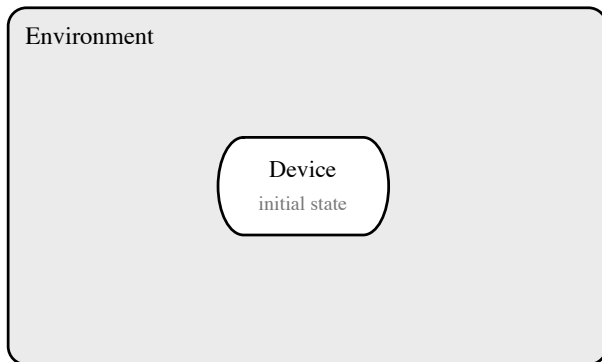
- ▶ *Interactive computation* captures the notion of bio-inspired computing better than classical Turing-like computation.
- ▶ *Architectural Evolution* is a crucial aspect of biological neural networks: synaptic plasticity, cell birth and death, ...
- ▶ We studied the computational capabilities of a rate model of *interactive* and *evolving* recurrent neural networks.
- ▶ We proved that *Interactive* and *evolving* recurrent neural networks are super-Turing.

Introduction

- ▶ *Interactive computation* captures the notion of bio-inspired computing better than classical Turing-like computation.
- ▶ *Architectural Evolution* is a crucial aspect of biological neural networks: synaptic plasticity, cell birth and death, ...
- ▶ We studied the computational capabilities of a rate model of *interactive* and *evolving* recurrent neural networks.
- ▶ We proved that *Interactive* and *evolving* recurrent neural networks are super-Turing.

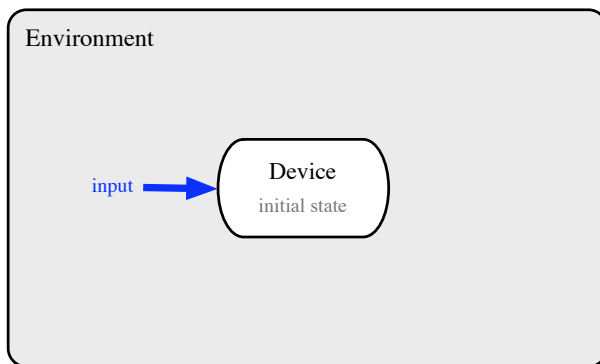


Interactive computation



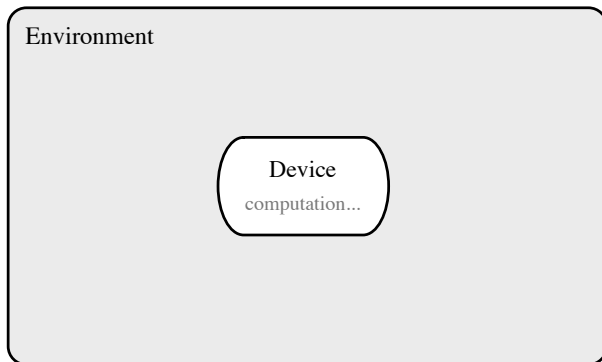
Sequentially interactive and memory active...

Interactive computation



Sequentially interactive and memory active...

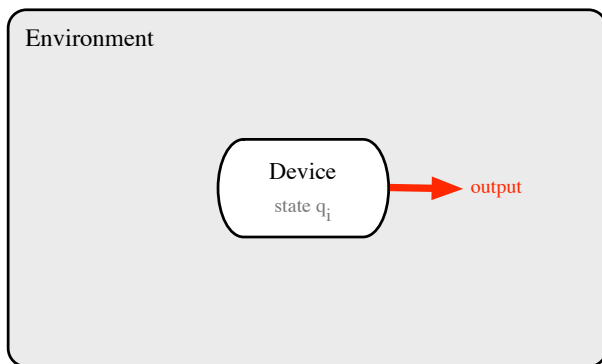
Interactive computation



Sequentially interactive and memory active...

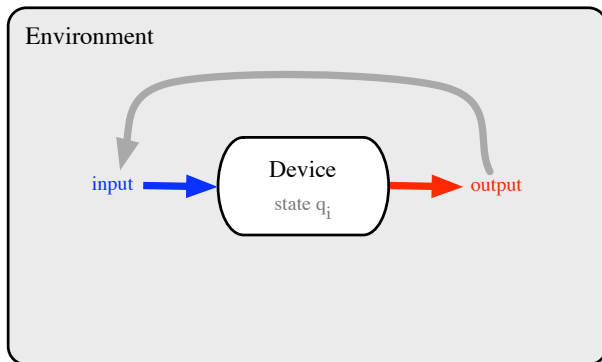


Interactive computation



Sequentially interactive and memory active...

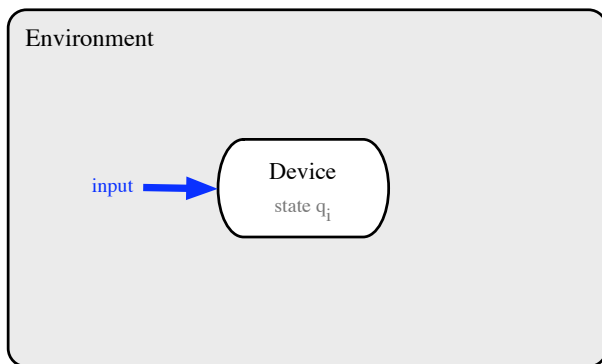
Interactive computation



Sequentially interactive and memory active...



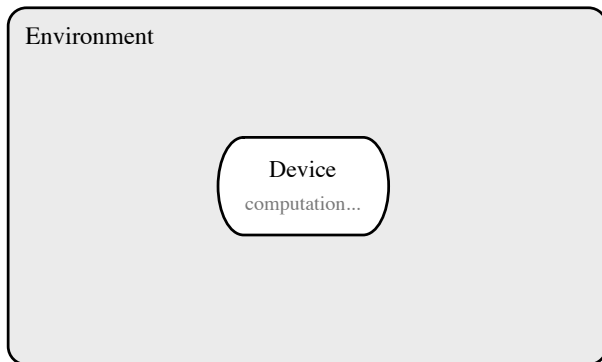
Interactive computation



Sequentially interactive and memory active...



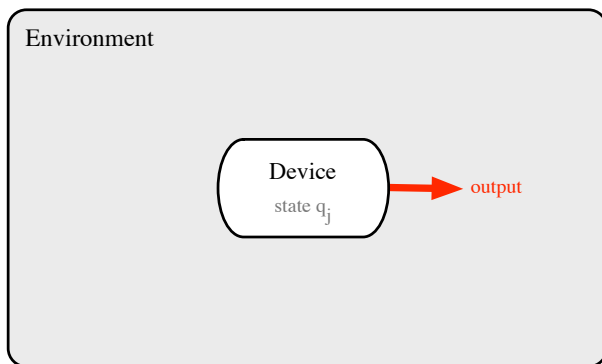
Interactive computation



Sequentially interactive and memory active...

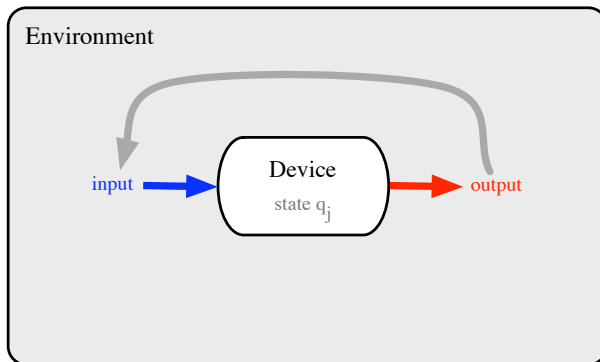


Interactive computation



Sequentially interactive and memory active...

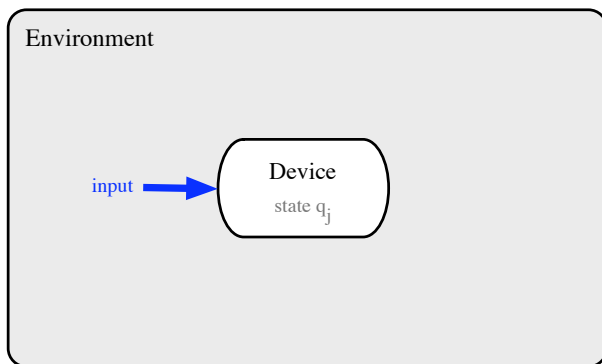
Interactive computation



Sequentially interactive and memory active...

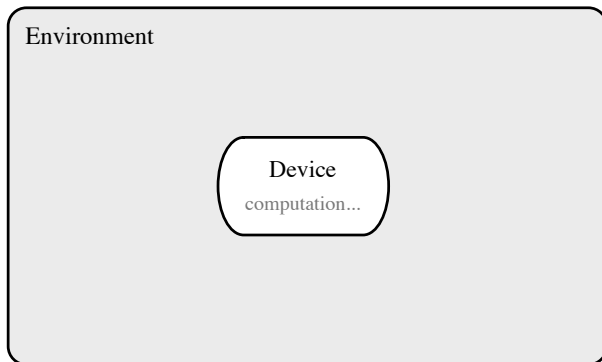


Interactive computation



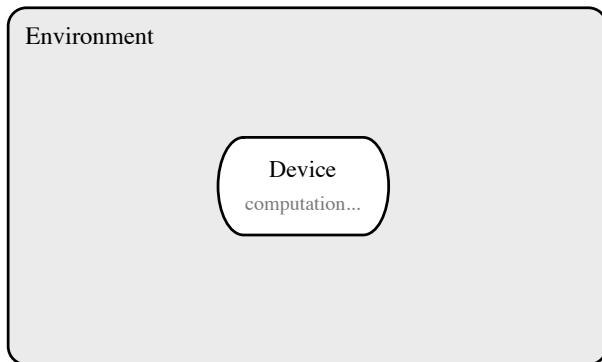
Sequentially interactive and memory active...

Interactive computation



Sequentially interactive and memory active...

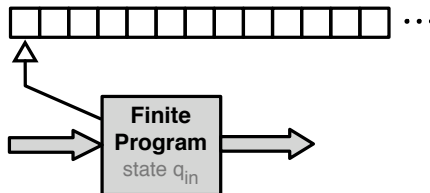
Interactive computation



Sequentially interactive and memory active...

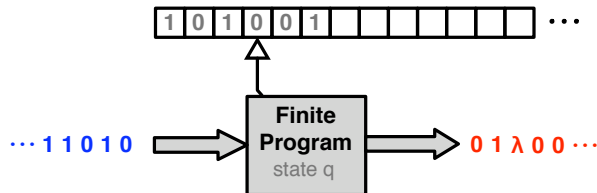


Interactive Turing machine (Int-TM)



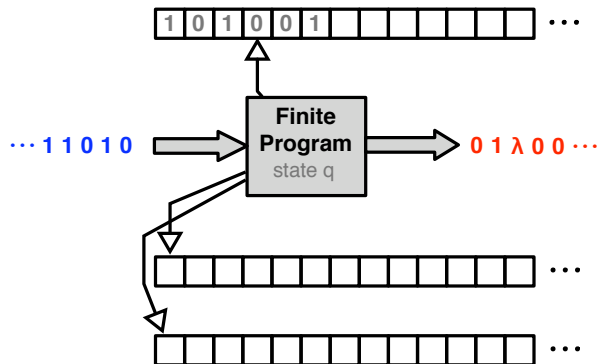


Interactive Turing machine (Int-TM)



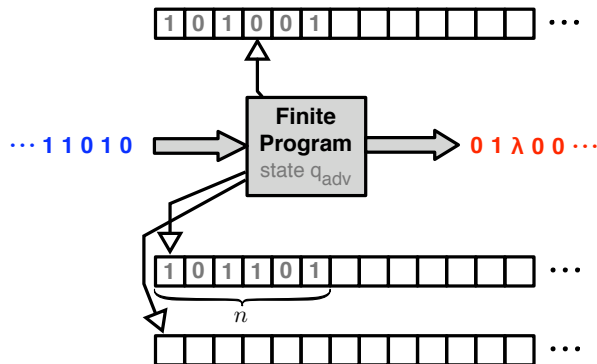
Interactive Turing machine with advice (Int-TM/A)

An Int-TM provided with additional advice input and output tapes and advice function $\alpha : \mathbb{N} \longrightarrow \{0, 1\}^*$



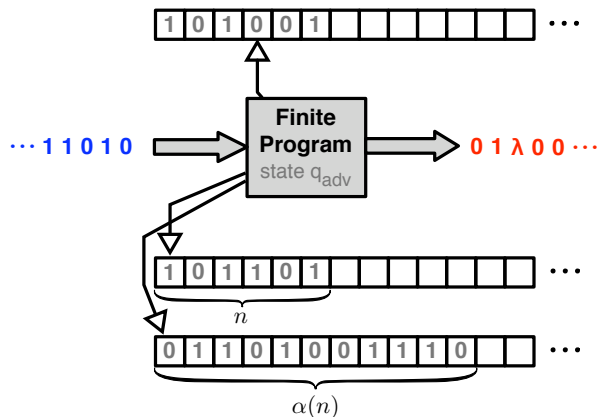
Interactive Turing machine with advice (Int-TM/A)

An Int-TM provided with additional advice input and output tapes and advice function $\alpha : \mathbb{N} \longrightarrow \{0, 1\}^*$



Interactive Turing machine with advice (Int-TM/A)

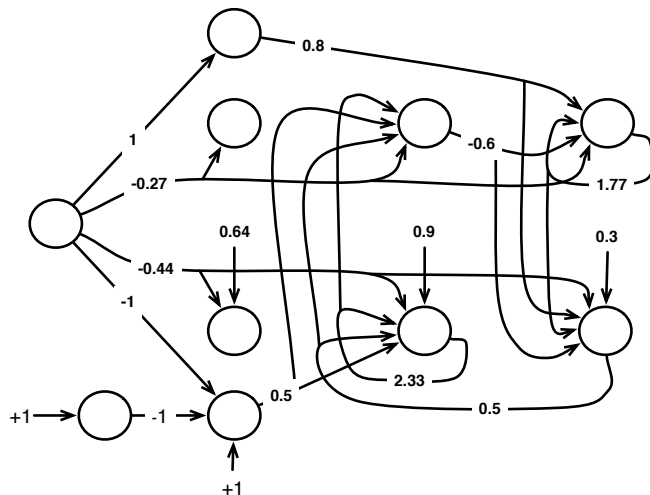
An Int-TM provided with additional advice input and output tapes and advice function $\alpha : \mathbb{N} \longrightarrow \{0, 1\}^*$



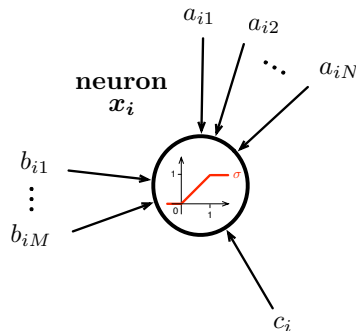
Lemma

Interactive Turing machines with advice (Int-TM/As) are strictly more powerful than interactive Turing machines (Int-TMs), i.e., they can compute strictly more ω -translations.

Recurrent Neural Networks



Dynamics: static synaptic weights



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

Previous Results

Int-St-RNN[\mathbb{Q}]s: interactive recurrent neural networks with static *rational* weights

Int-St-RNN[\mathbb{R}]s: interactive recurrent neural networks with static *real* weights (analog)

Theorem (Cabessa & Siegelmann)

- ▶ *Int-St-RNN[\mathbb{Q}]s are computationally equivalent to Int-TMs.*
- ▶ *Int-St-RNN[\mathbb{R}]s are computationally equivalent to Int-TM/As, hence super-Turing.*

Previous Results

Int-St-RNN[\mathbb{Q}]s: interactive recurrent neural networks with static *rational* weights

Int-St-RNN[\mathbb{R}]s: interactive recurrent neural networks with static *real* weights (analog)

Theorem (Cabessa & Siegelmann)

- ▶ *Int-St-RNN[\mathbb{Q}]s are computationally equivalent to Int-TMs.*
- ▶ *Int-St-RNN[\mathbb{R}]s are computationally equivalent to Int-TM/As, hence super-Turing.*

Previous Results

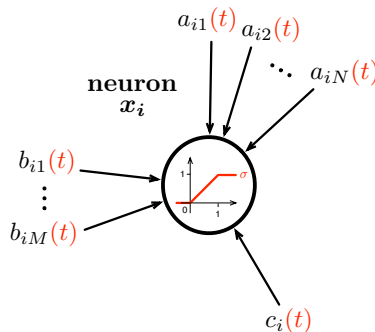
Int-St-RNN[\mathbb{Q}]s: interactive recurrent neural networks with static *rational* weights

Int-St-RNN[\mathbb{R}]s: interactive recurrent neural networks with static *real* weights (analog)

Theorem (Cabessa & Siegelmann)

- ▶ *Int-St-RNN[\mathbb{Q}]s are computationally equivalent to Int-TMs.*
- ▶ *Int-St-RNN[\mathbb{R}]s are computationally equivalent to Int-TM/As, hence super-Turing.*

Dynamics: evolving synaptic weights



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

New Results

Int-Ev-RNN[\mathbb{Q}]s**: interactive evolving recurrent neural networks with static *rational* weights**

Int-Ev-RNN[\mathbb{R}]s**: interactive evolving recurrent neural networks with static *real* weights (analog)**

Theorem (Cabessa & Siegelmann)

- ▶ *Int-Ev-RNN[\mathbb{Q}]**s** are computationally equivalent to Int-TM/As, hence super-Turing.*
- ▶ *Int-Ev-RNN[\mathbb{R}]**s** are computationally equivalent to Int-TM/As, hence super-Turing.*

New Results

Int-Ev-RNN[\mathbb{Q}]s**: interactive evolving recurrent neural networks with static *rational* weights**

Int-Ev-RNN[\mathbb{R}]s**: interactive evolving recurrent neural networks with static *real* weights (analog)**

Theorem (Cabessa & Siegelmann)

- ▶ *Int-Ev-RNN[\mathbb{Q}]**s** are computationally equivalent to Int-TM/As, hence super-Turing.*
- ▶ *Int-Ev-RNN[\mathbb{R}]**s** are computationally equivalent to Int-TM/As, hence super-Turing.*

New Results

Int-Ev-RNN[\mathbb{Q}]s: interactive evolving recurrent neural networks with static *rational* weights

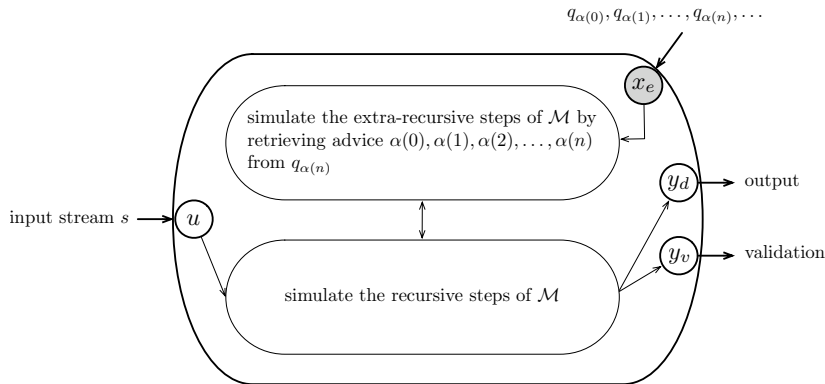
Int-Ev-RNN[\mathbb{R}]s: interactive evolving recurrent neural networks with static *real* weights (analog)

Theorem (Cabessa & Siegelmann)

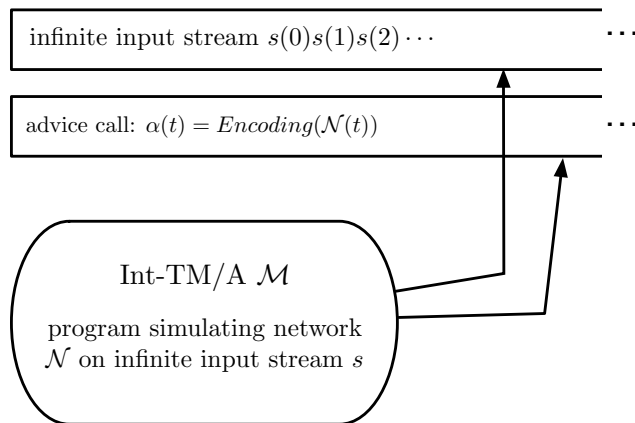
- ▶ *Int-Ev-RNN[\mathbb{Q}]s are computationally equivalent to Int-TM/As, hence super-Turing.*
- ▶ *Int-Ev-RNN[\mathbb{R}]s are computationally equivalent to Int-TM/As, hence super-Turing.*



Proof: from Int-TM/A to Int-Ev-RNN[Q]



Proof: from Int-Ev-RNN[\mathbb{Q}] to Int-TM/A



Summary

	Static	Evolving
\mathbb{Q}	Turing	Super-Turing
\mathbb{R}	Super-Turing	Super-Turing

Conclusions

- ▶ *Architectural Evolution* is an alternative way to the *power of the continuum* to achieve super-Turing capabilities.
- ▶ The results support the idea that *architectural evolution* might play a crucial role in the computational capabilities of biological neural networks.
- ▶ Future work: study the computational power of more biologically oriented neural models involved in more bio-inspired computational frameworks.



Conclusions

- ▶ *Architectural Evolution* is an alternative way to the *power of the continuum* to achieve super-Turing capabilities.
- ▶ The results support the idea that *architectural evolution* might play a crucial role in the computational capabilities of biological neural networks.
- ▶ Future work: study the computational power of more biologically oriented neural models involved in more bio-inspired computational frameworks.

Conclusions

- ▶ *Architectural Evolution* is an alternative way to the *power of the continuum* to achieve super-Turing capabilities.
- ▶ The results support the idea that *architectural evolution* might play a crucial role in the computational capabilities of biological neural networks.
- ▶ Future work: study the computational power of more biologically oriented neural models involved in more bio-inspired computational frameworks.