Interactive Evolving Neural Networks are Super-Turing Universal

Jérémie Cabessa
Joint work with Alessandro E.P. Villa

Department of Mathematical Economics
University of Paris 2
France

16 September 2014

Introduction

- We follow the so-called *mind-computer analogy* approach to cognitive science.
- We study the computational capabilities of basic models of recurrent neural networks.
- We show that recurrent neural networks provide a natural model of computation beyond the Turing limits.

Introduction

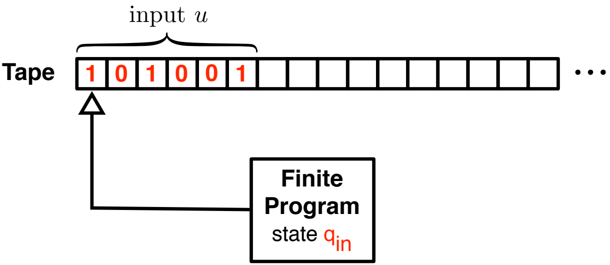
- We follow the so-called *mind-computer analogy* approach to cognitive science.
- We study the computational capabilities of basic models of recurrent neural networks.
- We show that recurrent neural networks provide a natural model of computation beyond the Turing limits.

Introduction

- We follow the so-called *mind-computer analogy* approach to cognitive science.
- We study the computational capabilities of basic models of recurrent neural networks.
- We show that recurrent neural networks provide a natural model of computation beyond the Turing limits.

Turing machine

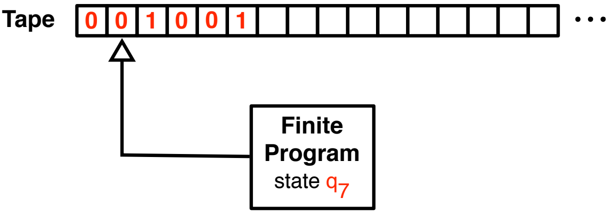
A Turing machine (TM) consists of an infinite tape, a read-write head, and a finite program.



- input u is *accepted* by $\mathcal{M}$ if $\mathcal{M}(u)$ reaches the state q_{acc}
- input u is *rejected* by $\mathcal{M}$ if $\mathcal{M}(u)$ reaches the state q_{rej}

Turing machine

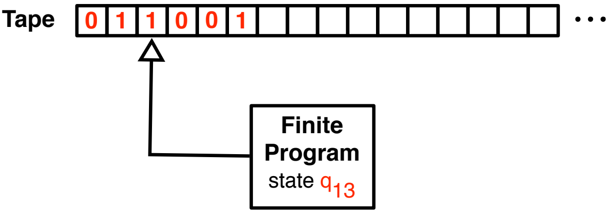
A Turing machine (TM) consists of an infinite tape, a read-write head, and a finite program.



- input u is *accepted* by $\mathcal{M}$ if $\mathcal{M}(u)$ reaches the state q_{acc}
- input u is *rejected* by $\mathcal{M}$ if $\mathcal{M}(u)$ reaches the state q_{rej}

Turing machine

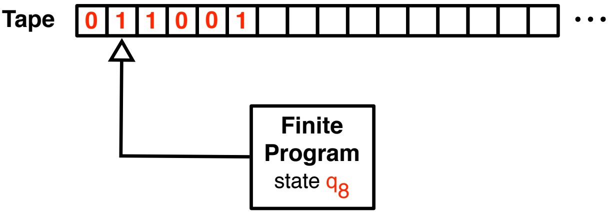
A Turing machine (TM) consists of an infinite tape, a read-write head, and a finite program.



- input u is *accepted* by $\mathcal{M}$ if $\mathcal{M}(u)$ reaches the state q_{acc}
- input u is *rejected* by $\mathcal{M}$ if $\mathcal{M}(u)$ reaches the state q_{rej}

Turing machine

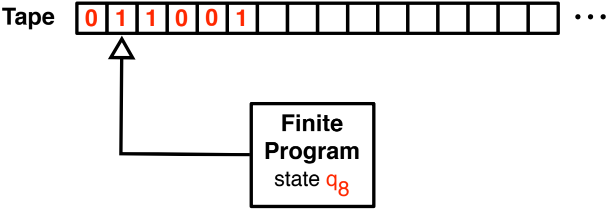
A Turing machine (TM) consists of an infinite tape, a read-write head, and a finite program.



- input u is *accepted* by $\mathcal{M}$ if $\mathcal{M}(u)$ reaches the state q_{acc}
- input u is *rejected* by $\mathcal{M}$ if $\mathcal{M}(u)$ reaches the state q_{rej}

Turing machine

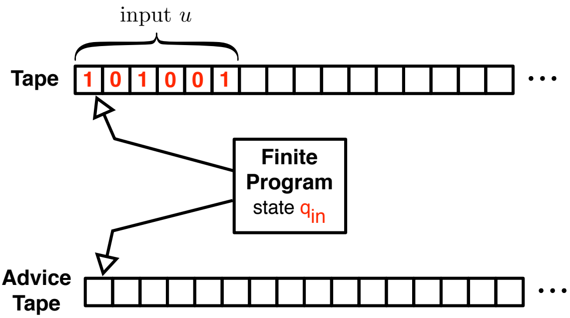
A Turing machine (TM) consists of an infinite tape, a read-write head, and a finite program.



- input u is *accepted* by $\mathcal{M}$ if $\mathcal{M}(u)$ reaches the state q_{acc}
- input u is *rejected* by $\mathcal{M}$ if $\mathcal{M}(u)$ reaches the state q_{rej}

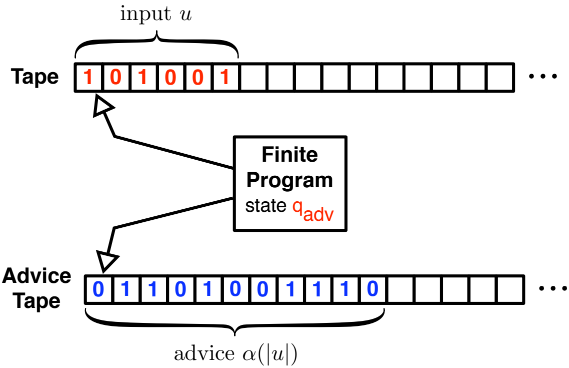
Turing machine with advice

A Turing machine with advice (TM/A) is a Turing machine provided with an additional advice tape and advice function $\alpha : \mathbb{N} \longrightarrow \{0, 1\}^*$.

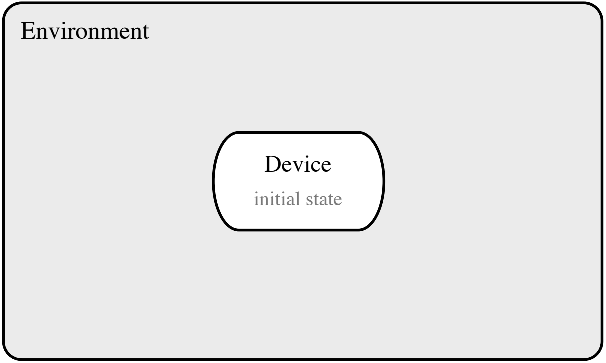


Turing machine with advice

A Turing machine with advice (TM/A) is a Turing machine provided with an additional advice tape and advice function $\alpha : \mathbb{N} \longrightarrow \{0,1\}^*$.

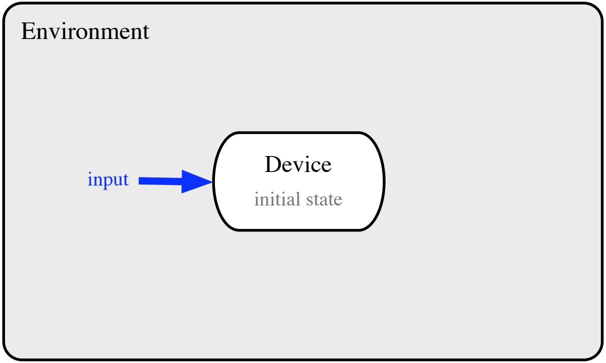


Classical Computation



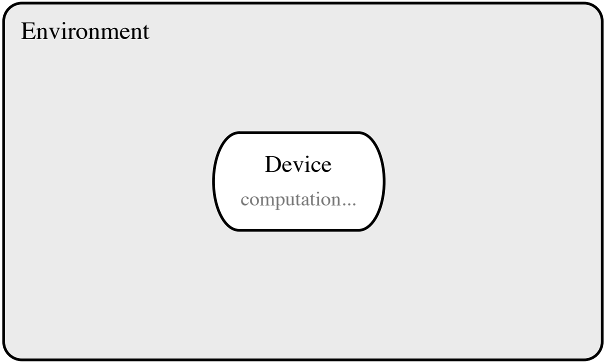
Closed-box and amnesic...

Classical Computation



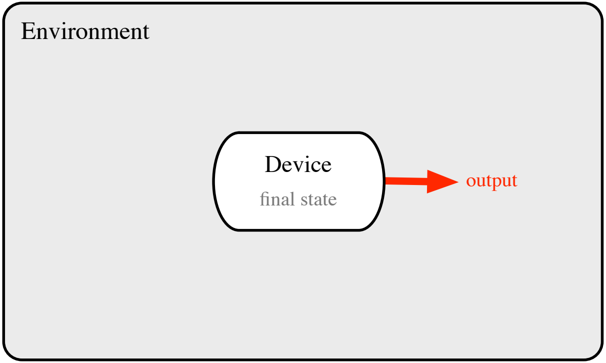
Closed-box and amnesic...

Classical Computation



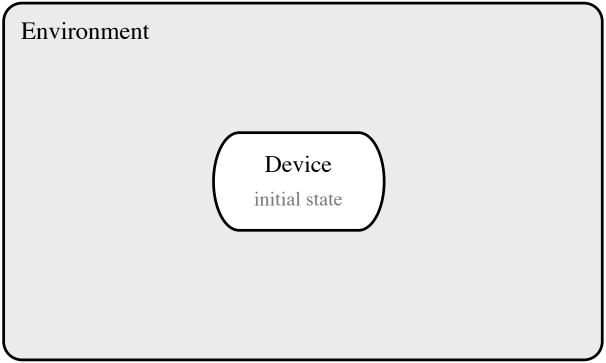
Closed-box and amnesic...

Classical Computation



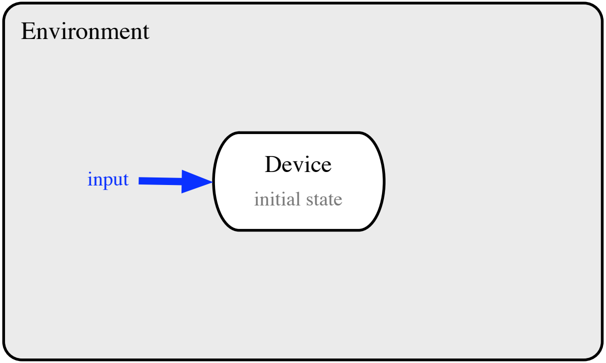
Closed-box and amnesic...

Classical Computation



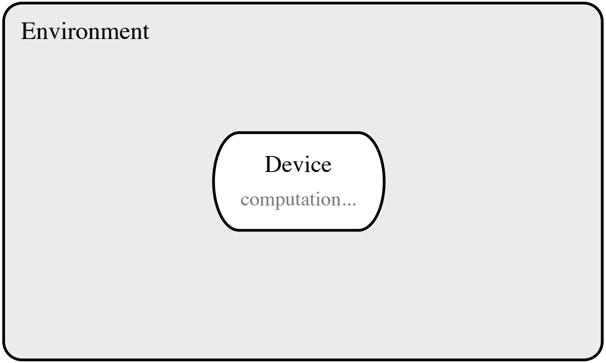
Closed-box and amnesic...

Classical Computation



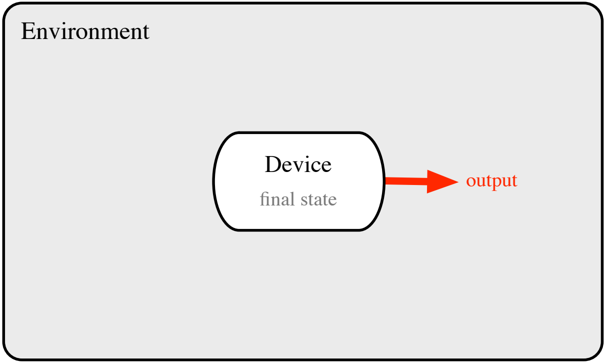
Closed-box and amnesic...

Classical Computation



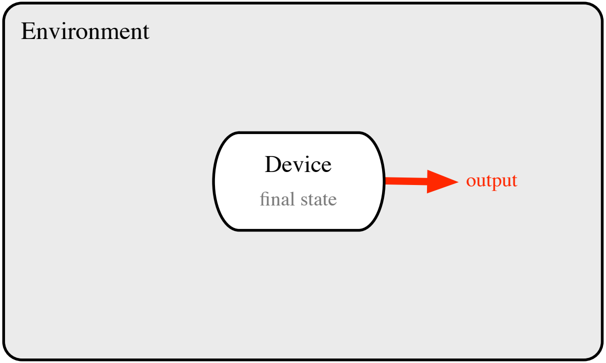
Closed-box and amnesic...

Classical Computation



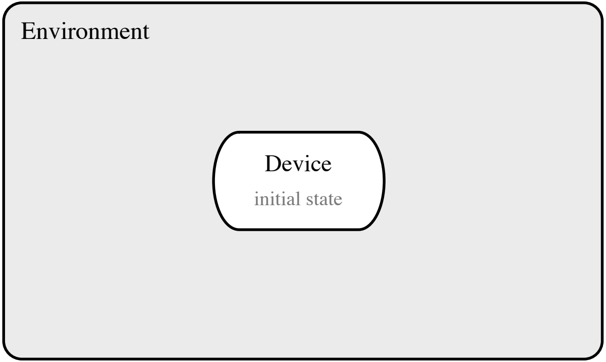
Closed-box and amnesic...

Classical Computation



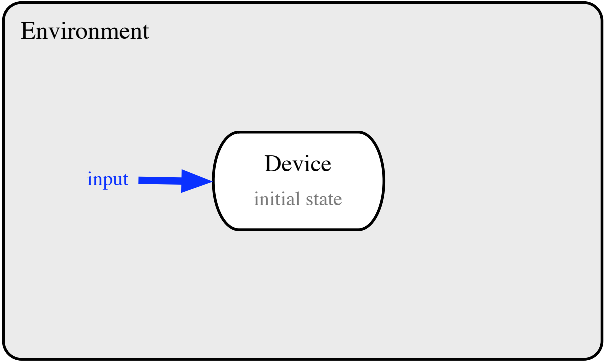
Closed-box and amnesic...

Interactive Computation



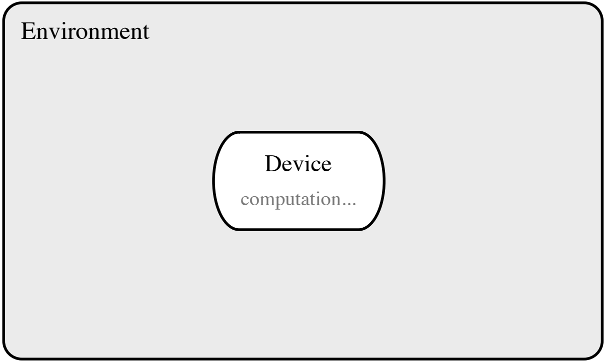
Sequentially interactive and memory active...

Interactive Computation



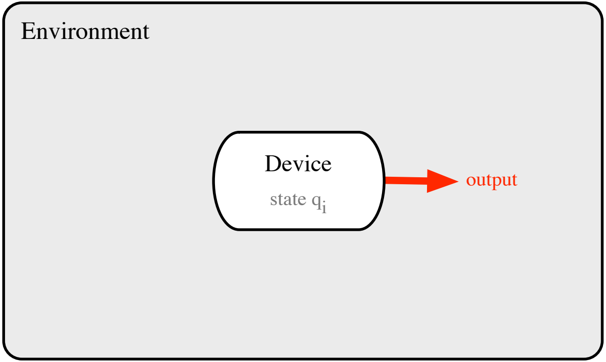
Sequentially interactive and memory active...

Interactive Computation



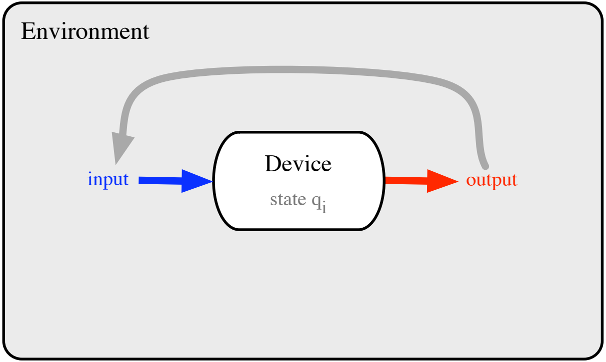
Sequentially interactive and memory active...

Interactive Computation



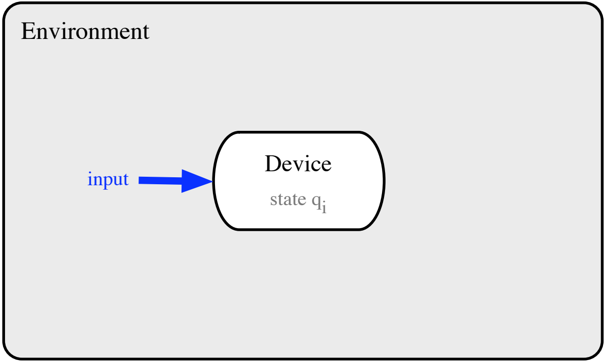
Sequentially interactive and memory active...

Interactive Computation



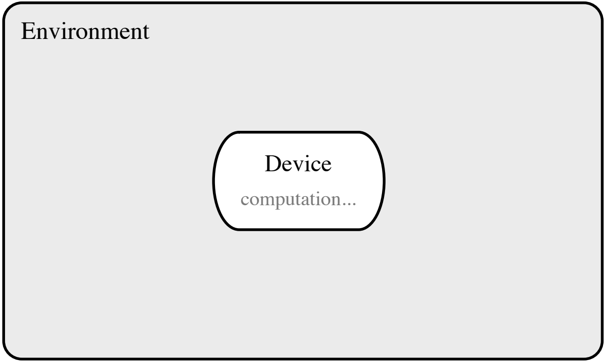
Sequentially interactive and memory active...

Interactive Computation



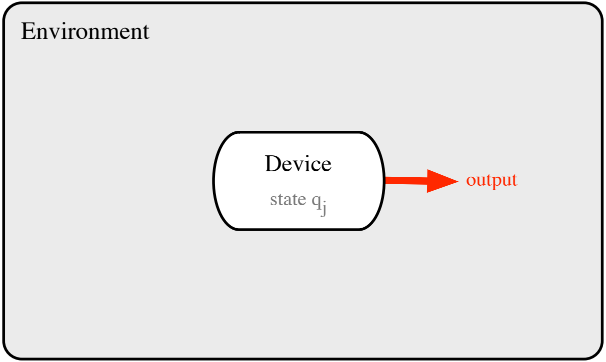
Sequentially interactive and memory active...

Interactive Computation



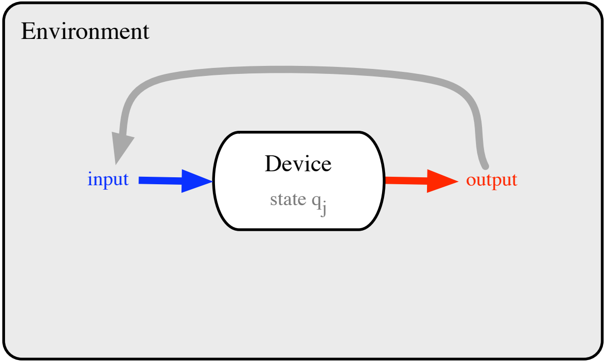
Sequentially interactive and memory active...

Interactive Computation



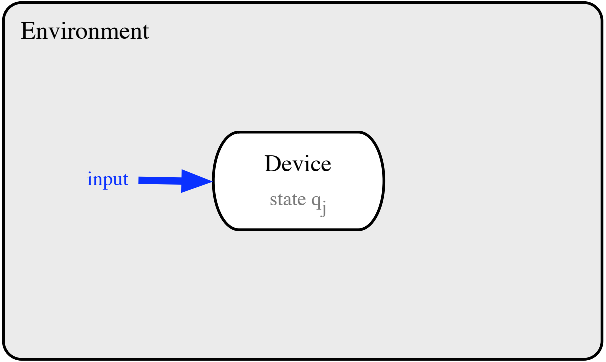
Sequentially interactive and memory active...

Interactive Computation



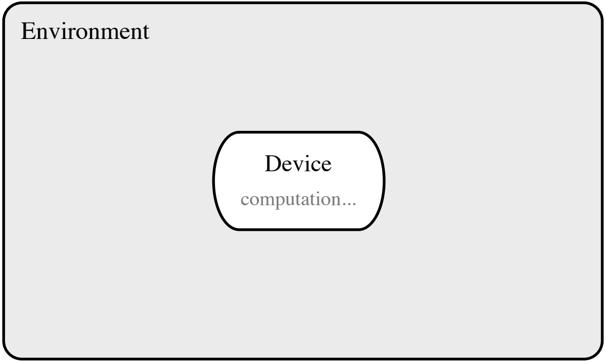
Sequentially interactive and memory active...

Interactive Computation



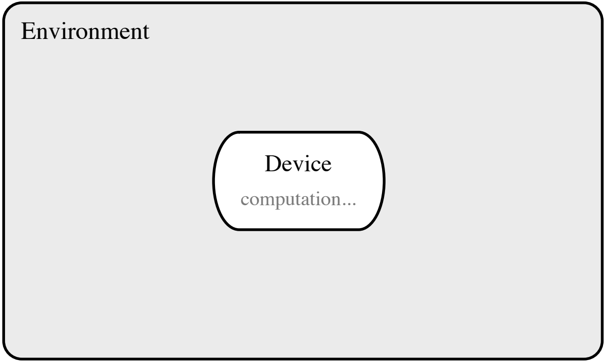
Sequentially interactive and memory active...

Interactive Computation



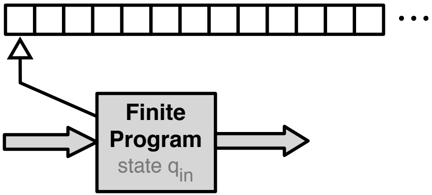
Sequentially interactive and memory active...

Interactive Computation

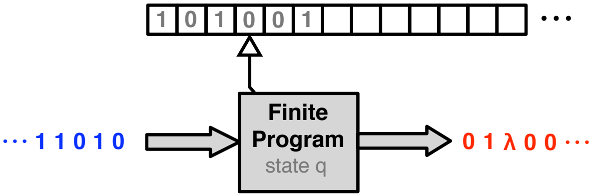


Sequentially interactive and memory active...

Interactive Turing machine

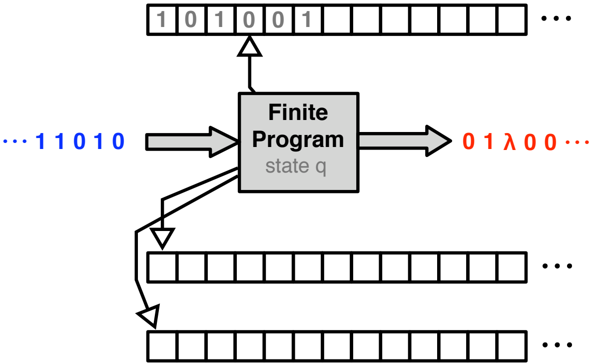


Interactive Turing machine



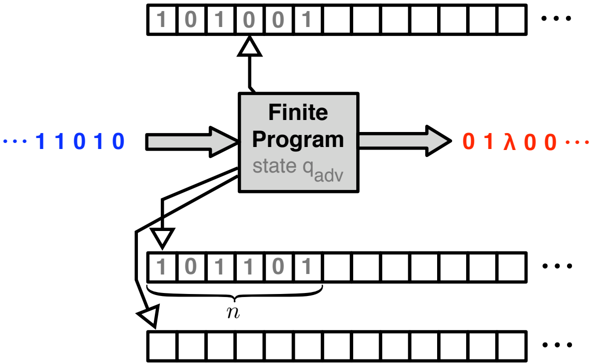
Interactive Turing machine with advice

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



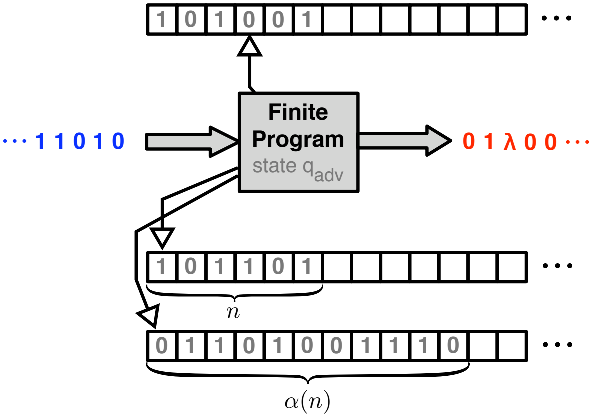
Interactive Turing machine with advice

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

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



Lemma

Turing machines with advice are strictly more powerful than Turing machines.

Lemma

Interactive Turing machines with advice are strictly more powerful than interactive Turing machines.

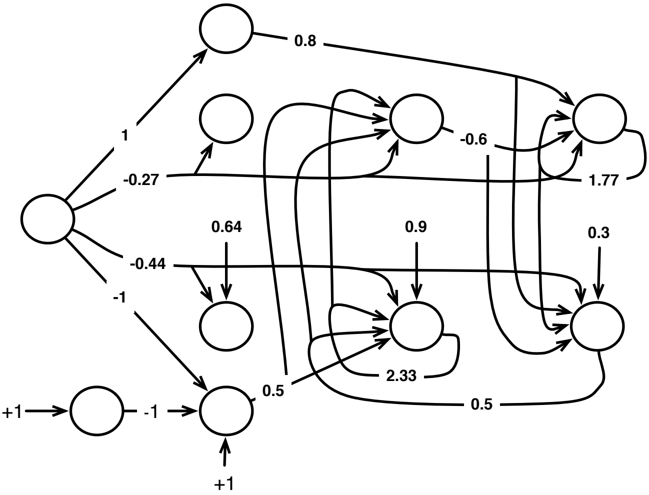
Lemma

Turing machines with advice are strictly more powerful than Turing machines.

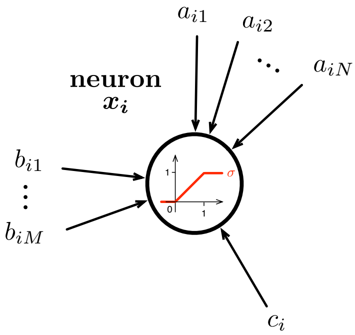
Lemma

Interactive Turing machines with advice are strictly more powerful than interactive Turing machines.

Recurrent Neural Networks



Dynamics: static synaptic weights

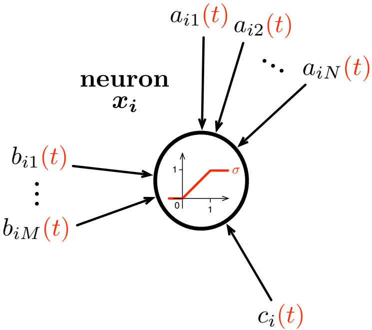


$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{ij} \cdot x_j(t) + \sum_{j=1}^{M} b_{ij} \cdot u_j(t) + c_i\right)$$

Results

	Static Architecture
$\mathbb{Q}$	Turing Siegelmann & Sontag 95 (classical comp.) Cabessa & Siegelmann 12 (interactive comp.)
$\mathbb{R}$	Super-Turing Siegelmann & Sontag 94 (classical comp.) Cabessa & Siegelmann 12 (interactive comp.) Cabessa & Villa 12 (interactive comp.)

Dynamics: evolving synaptic weights

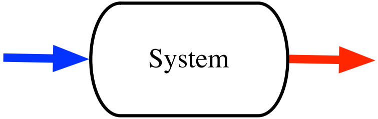


$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{ij}(t) \cdot x_j(t) + \sum_{j=1}^{M} b_{ij}(t) \cdot u_j(t) + c_i(t)\right)$$

Results

	Evolving Architecture
$\mathbb{Q}$	Super-Turing Cabessa & Siegelmann 11 & 14 (classical comp.) Cabessa & Villa 13 (interactive comp.)
$\mathbb{R}$	Super-Turing Cabessa & Siegelmann 11 & 14 (classical comp.) Cabessa & Villa 13 (interactive comp.)

Interactive Scenario

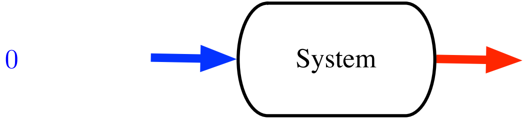


In this way, any deterministic interactive system $\mathcal{S}$ realises an ω-*translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Interactive Scenario

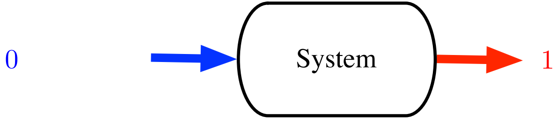


In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Interactive Scenario

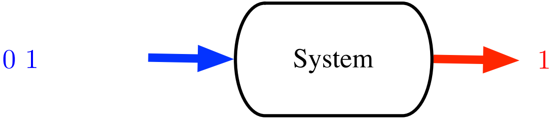


In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Interactive Scenario

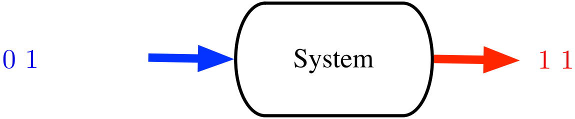


In this way, any deterministic interactive system $\mathcal{S}$ realises an ω-*translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Interactive Scenario

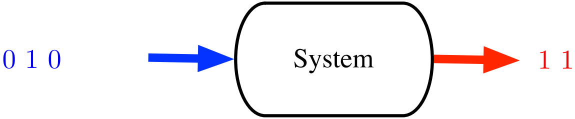


In this way, any deterministic interactive system $\mathcal{S}$ realises an ω-*translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Interactive Scenario

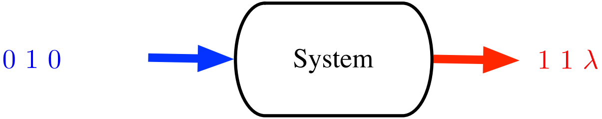


In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Interactive Scenario

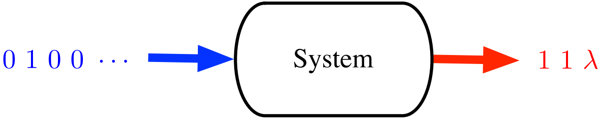


In this way, any deterministic interactive system $\mathcal{S}$ realises an ω-*translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Interactive Scenario

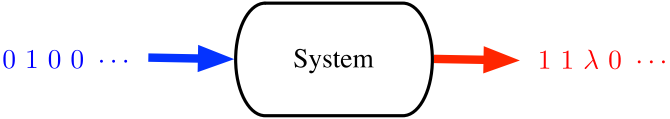


In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Interactive Scenario

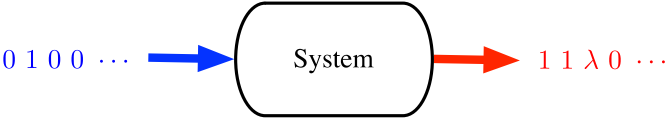


In this way, any deterministic interactive system $\mathcal{S}$ realises an *ω-translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Interactive Scenario

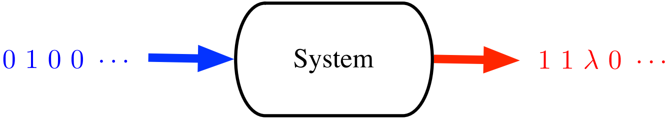


In this way, any deterministic interactive system $\mathcal{S}$ realises an ω-*translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Interactive Scenario



In this way, any deterministic interactive system $\mathcal{S}$ realises an ω-*translation* $\varphi_{\mathcal{S}} : \{0,1\}^{\omega} \to \{0,1\}^{\leq\omega}$.

Definition

An ω-translation ψ is said to be *interactively computable* if there exists a deterministic interactive system $\mathcal{S}$ (of any kind) such that $\varphi_{\mathcal{S}} = \psi$.

Results

Theorem

Interactive Evolving RNNs are Super-Turing universal.

More precisely, let ψ be some ω-translation. The following conditions are equivalent:

1. *ψ is interactively computable (by* any *det. int. system)*
2. *ψ is realisable by some Int-Ev-RNN[$\mathbb{B}$]*
3. *ψ is realisable by some Int-Ev-RNN[$\mathbb{Q}$]*
4. *ψ is realisable by some Int-St-RNN[$\mathbb{R}$]*
5. *ψ is realisable by some Int-TM/A*
6. *ψ is continuous (i.e. "limit" of monotone function)*

Results

Theorem

Interactive Evolving RNNs are Super-Turing universal.

More precisely, let ψ be some ω-translation. The following conditions are equivalent:

1. *ψ is interactively computable (by any det. int. system)*
2. *ψ is realisable by some Int-Ev-RNN[$\mathbb{R}$]*
3. *ψ is realisable by some Int-Ev-RNN[$\mathbb{Q}$]*
4. *ψ is realisable by some Int-St-RNN[$\mathbb{R}$]*
5. *ψ is realisable by some Int-TM/A*
6. *ψ is continuous (i.e. "limit" of monotone function)*

Results

Theorem

Interactive Evolving RNNs are Super-Turing universal.

More precisely, let ψ be some ω-translation. The following conditions are equivalent:

1. ψ *is interactively computable (by* **any** *det. int. system)*
2. ψ *is realisable by some Int-Ev-RNN[$\mathbb{R}$]*
3. ψ *is realisable by some Int-Ev-RNN[$\mathbb{Q}$]*
4. ψ *is realisable by some Int-St-RNN[$\mathbb{R}$]*
5. ψ *is realisable by some Int-TM/A*
6. ψ *is continuous (i.e. "limit" of monotone function)*

Results

Theorem

Interactive Evolving RNNs are Super-Turing universal.

More precisely, let ψ be some ω-translation. The following conditions are equivalent:

1. ψ *is interactively computable (by* **any** *det. int. system)*
2. ψ *is realisable by some Int-Ev-RNN[$\mathbb{R}$]*
3. ψ *is realisable by some Int-Ev-RNN[$\mathbb{Q}$]*
4. ψ *is realisable by some Int-St-RNN[$\mathbb{R}$]*
5. ψ *is realisable by some Int-TM/A*
6. ψ *is continuous (i.e. "limit" of monotone function)*

Results

Theorem

Interactive Evolving RNNs are Super-Turing universal.

More precisely, let ψ be some ω-translation. The following conditions are equivalent:

1. ψ *is interactively computable (by* **any** *det. int. system)*
2. ψ *is realisable by some Int-Ev-RNN[$\mathbb{R}$]*
3. ψ *is realisable by some Int-Ev-RNN[$\mathbb{Q}$]*
4. ψ *is realisable by some Int-St-RNN[$\mathbb{R}$]*
5. ψ *is realisable by some Int-TM/A*
6. ψ *is continuous (i.e. "limit" of monotone function)*

Results

Theorem

Interactive Evolving RNNs are Super-Turing universal.

More precisely, let ψ be some ω-translation. The following conditions are equivalent:

1. ψ *is interactively computable (by* **any** *det. int. system)*
2. ψ *is realisable by some Int-Ev-RNN[$\mathbb{R}$]*
3. ψ *is realisable by some Int-Ev-RNN[$\mathbb{Q}$]*
4. ψ *is realisable by some Int-St-RNN[$\mathbb{R}$]*
5. ψ *is realisable by some Int-TM/A*
6. ψ *is continuous (i.e. "limit" of monotone function)*

Results

Theorem

Interactive Evolving RNNs are Super-Turing universal.

More precisely, let ψ be some ω-translation. The following conditions are equivalent:

1. ψ *is interactively computable (by* **any** *det. int. system)*
2. ψ *is realisable by some Int-Ev-RNN[$\mathbb{R}$]*
3. ψ *is realisable by some Int-Ev-RNN[$\mathbb{Q}$]*
4. ψ *is realisable by some Int-St-RNN[$\mathbb{R}$]*
5. ψ *is realisable by some Int-TM/A*
6. ψ *is continuous (i.e. "limit" of monotone function)*

Results

Theorem

Interactive Evolving RNNs are Super-Turing universal.

More precisely, let ψ be some ω-translation. The following conditions are equivalent:

1. *ψ is interactively computable (by* **any** *det. int. system)*
2. *ψ is realisable by some Int-Ev-RNN[$\mathbb{R}$]*
3. *ψ is realisable by some Int-Ev-RNN[$\mathbb{Q}$]*
4. *ψ is realisable by some Int-St-RNN[$\mathbb{R}$]*
5. *ψ is realisable by some Int-TM/A*
6. *ψ is continuous (i.e. "limit" of monotone function)*

Summary

	Static	Evolving
$\mathbb{Q}$	Turing	Super-Turing universal (interactive case)
$\mathbb{R}$	Super-Turing universal (interactive case)	Super-Turing universal (interactive case)

Conclusions

- The results are theoretical: any implementation of an evolving RNN will never be super-Turing.
- Evolving-RNNs provide a natural abstract computational model beyond the Turing limits.
- *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.
- The results do not prove that the brain is super-Turing...

Conclusions

- The results are theoretical: any implementation of an evolving RNN will never be super-Turing.
- Evolving-RNNs provide a natural abstract computational model beyond the Turing limits.
- *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.
- The results do not prove that the brain is super-Turing...

Conclusions

- The results are theoretical: any implementation of an evolving RNN will never be super-Turing.
- Evolving-RNNs provide a natural abstract computational model beyond the Turing limits.
- *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.
- The results do not prove that the brain is super-Turing...

Conclusions

- The results are theoretical: any implementation of an evolving RNN will never be super-Turing.
- Evolving-RNNs provide a natural abstract computational model beyond the Turing limits.
- *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.
- The results do not prove that the brain is super-Turing...

Conclusions

- The results are theoretical: any implementation of an evolving RNN will never be super-Turing.
- Evolving-RNNs provide a natural abstract computational model beyond the Turing limits.
- *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.
- The results do not prove that the brain is super-Turing...

Conclusions

- The results are theoretical: any implementation of an evolving RNN will never be super-Turing.
- Evolving-RNNs provide a natural abstract computational model beyond the Turing limits.
- *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.
- The results do not prove that the brain is super-Turing...