Expressive power of Nondeterministic Evolving Recurrent Neural Networks in Terms of their Attractor Dynamics

Jérémie Cabessa and Jacques Duparc

Department of Mathematical Economics
University of Paris 2
France

31 August 2015

Introduction

- We assume that some aspect of information processing in the brain can be approached from the perspective of computability theory.
- The computational capabilities of recurrent neural networks have mainly been studied in the context of classical computation (McCulloch & Pitts, Turing, Kleene, von Neumann, Minsky, Papert,..., Siegelmann & Sontag,...).
- Here, we introduce a natural model of nondeterministic recurrent neural networks and study their expressive power in terms of their attractor dynamics, i.e., in the context of infinite (or non-terminating) computations.

Introduction

- We assume that some aspect of information processing in the brain can be approached from the perspective of computability theory.
- The computational capabilities of recurrent neural networks have mainly been studied in the context of classical computation (McCulloch & Pitts, Turing, Kleene, von Neumann, Minsky, Papert,..., Siegelmann & Sontag,...).
- Here, we introduce a natural model of nondeterministic recurrent neural networks and study their expressive power in terms of their attractor dynamics, i.e., in the context of infinite (or non-terminating) computations.

Introduction

- We assume that some aspect of information processing in the brain can be approached from the perspective of computability theory.
- The computational capabilities of recurrent neural networks have mainly been studied in the context of classical computation (McCulloch & Pitts, Turing, Kleene, von Neumann, Minsky, Papert,..., Siegelmann & Sontag,...).
- Here, we introduce a natural model of nondeterministic recurrent neural networks and study their expressive power in terms of their attractor dynamics, i.e., in the context of infinite (or non-terminating) computations.

Introduction

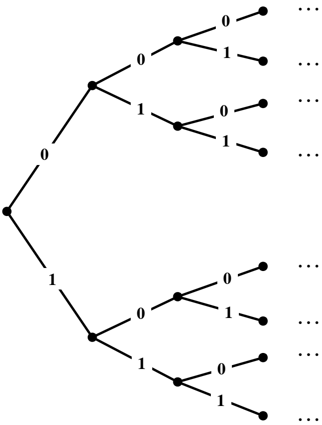
- We assume that some aspect of information processing in the brain can be approached from the perspective of computability theory.
- The computational capabilities of recurrent neural networks have mainly been studied in the context of classical computation (McCulloch & Pitts, Turing, Kleene, von Neumann, Minsky, Papert,..., Siegelmann & Sontag,...).
- Here, we introduce a natural model of nondeterministic recurrent neural networks and study their expressive power in terms of their attractor dynamics, i.e., in the context of infinite (or non-terminating) computations.

Introduction

- We assume that some aspect of information processing in the brain can be approached from the perspective of computability theory.
- The computational capabilities of recurrent neural networks have mainly been studied in the context of classical computation (McCulloch & Pitts, Turing, Kleene, von Neumann, Minsky, Papert,..., Siegelmann & Sontag,...).
- Here, we introduce a natural model of nondeterministic recurrent neural networks and study their expressive power in terms of their attractor dynamics, i.e., in the context of infinite (or non-terminating) computations.

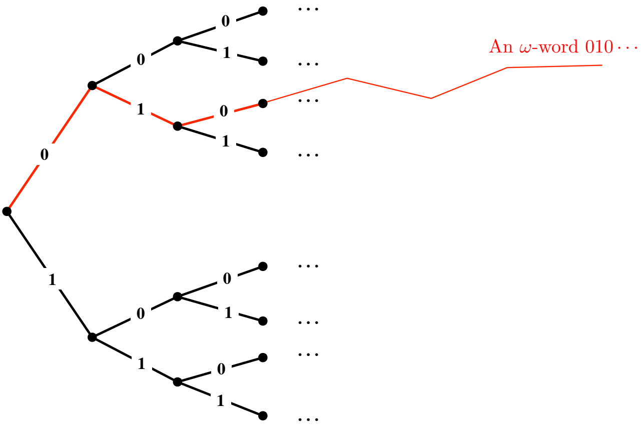
Topology

The Cantor space $\{0,1\}^\omega$
the set of infinite sequences of bits



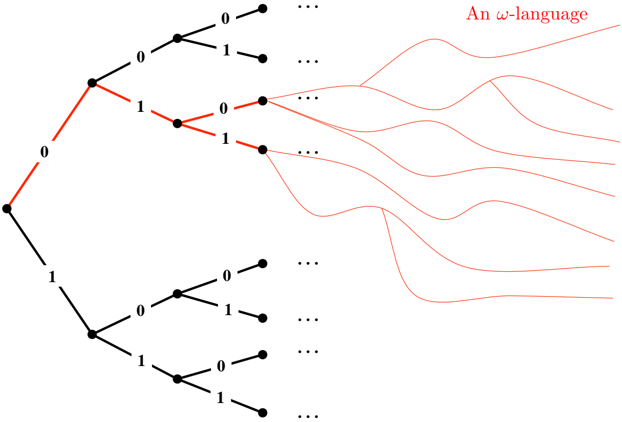
Topology

The Cantor space $\{0,1\}^\omega$
the set of infinite sequences of bits



Topology

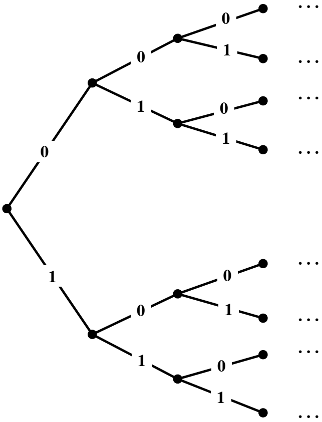
The Cantor space $\{0, 1\}^\omega$
the set of infinite sequences of bits



Topology

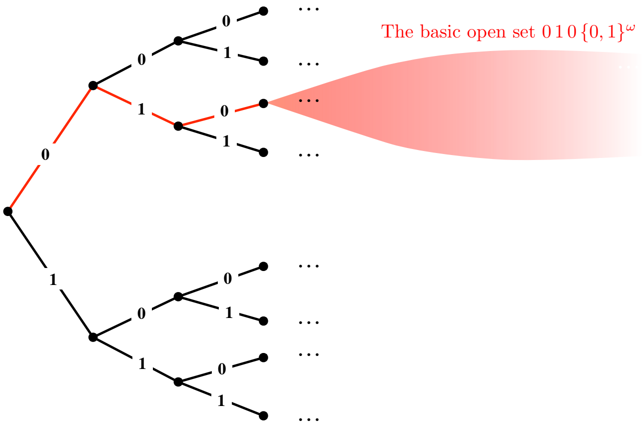
The Cantor space $\{0,1\}^\omega$
the set of infinite sequences of bits

equipped with product topology
of the discrete topology on $\{0,1\}$



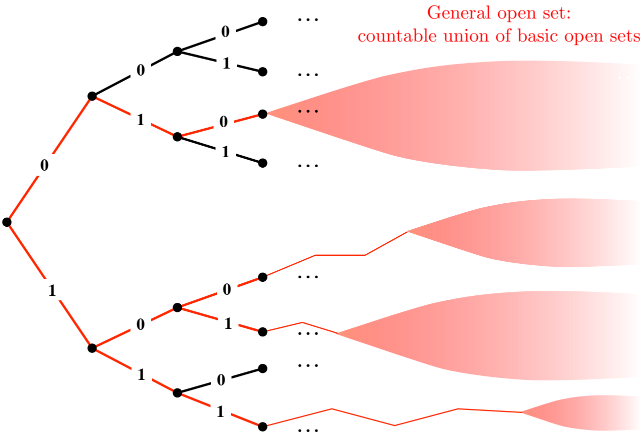
Topology

The Cantor space $\{0,1\}^\omega$
the set of infinite sequences of bits



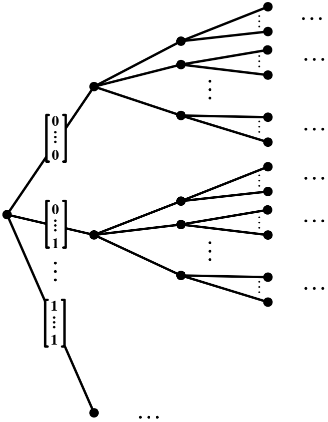
Topology

The Cantor space $\{0,1\}^\omega$
the set of infinite sequences of bits



Topology

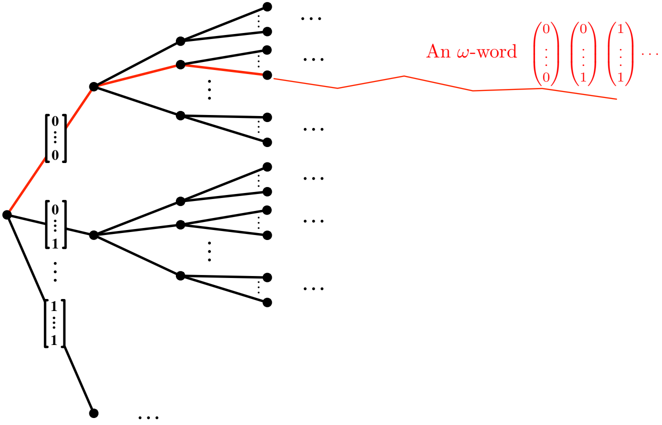
The space $(\mathbb{B}^N)^\omega$
the set of infinite sequences of N-dim. Boolean vectors



Topology

The space $(\mathbb{B}^N)^\omega$

the set of infinite sequences of N-dim. Boolean vectors

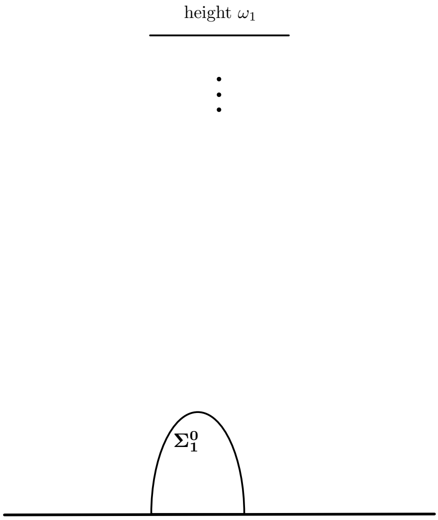


Topology

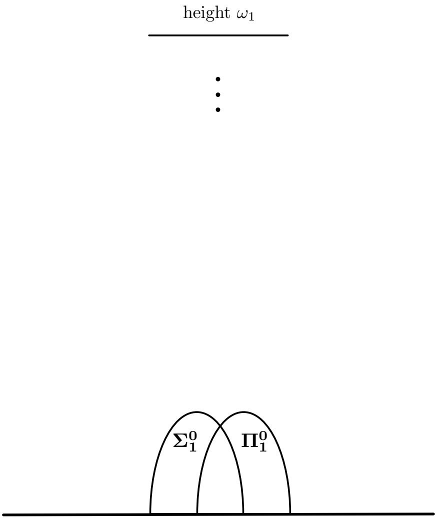
height ω_1

$\vdots$

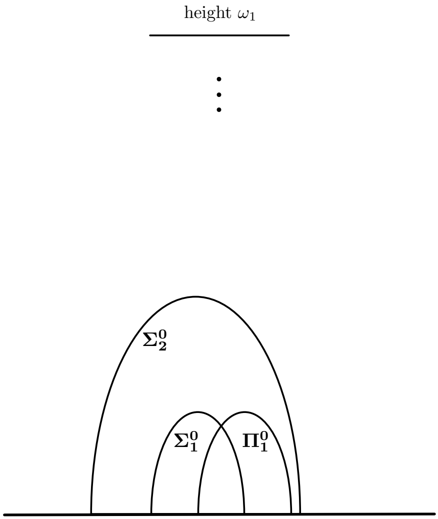
Topology



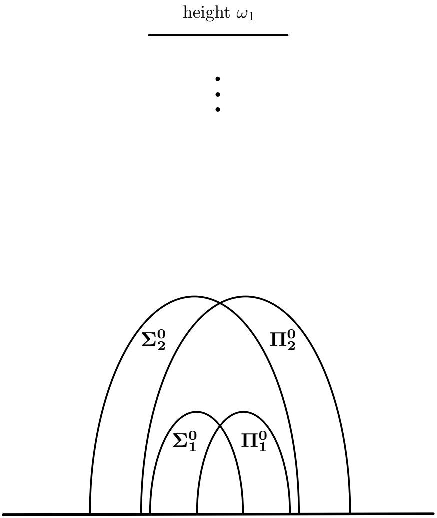
Topology



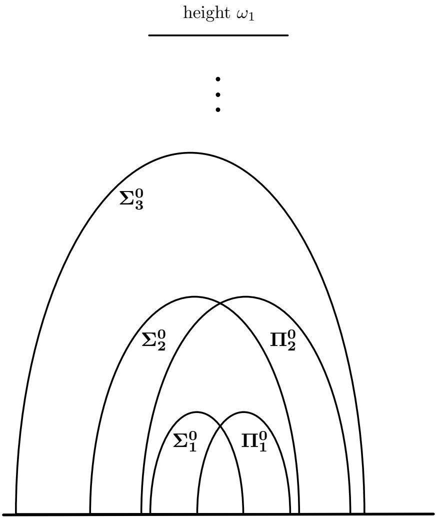
Topology


height ω_1
$\boldsymbol{\Sigma}^0_2$
$\boldsymbol{\Sigma}^0_1$
$\boldsymbol{\Pi}^0_1$

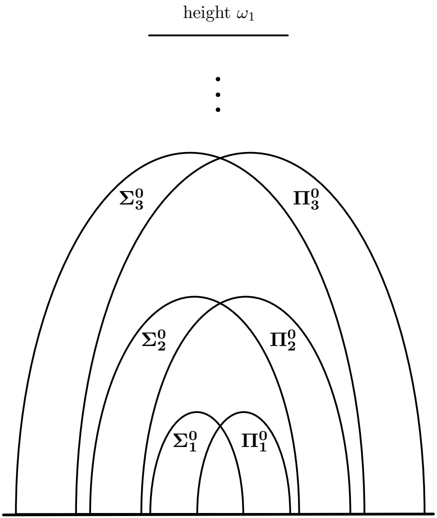
Topology


height ω_1
$\vdots$
$\mathbf{\Sigma^0_2}$
$\mathbf{\Pi^0_2}$
$\mathbf{\Sigma^0_1}$
$\mathbf{\Pi^0_1}$

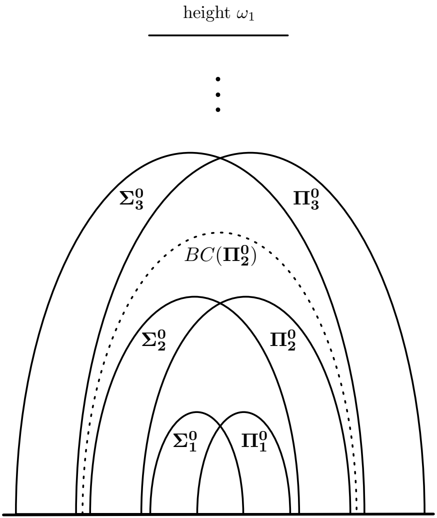
Topology


height ω_1
$\boldsymbol{\Sigma}^0_3$
$\boldsymbol{\Sigma}^0_2$
$\boldsymbol{\Pi}^0_2$
$\boldsymbol{\Sigma}^0_1$
$\boldsymbol{\Pi}^0_1$

Topology

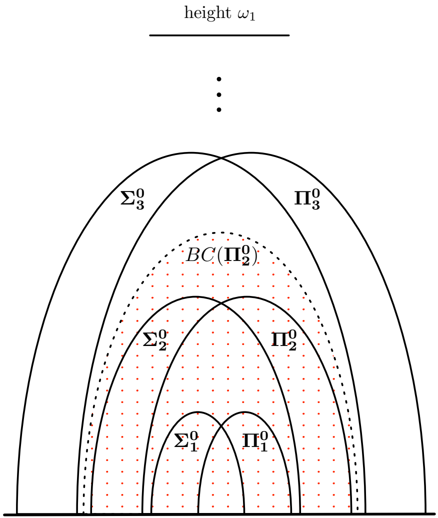


Topology



height ω_1
$\vdots$
$\mathbf{\Sigma}^0_3$
$\mathbf{\Pi}^0_3$
$BC(\mathbf{\Pi}^0_2)$
$\mathbf{\Sigma}^0_2$
$\mathbf{\Pi}^0_2$
$\mathbf{\Sigma}^0_1$
$\mathbf{\Pi}^0_1$

Topology

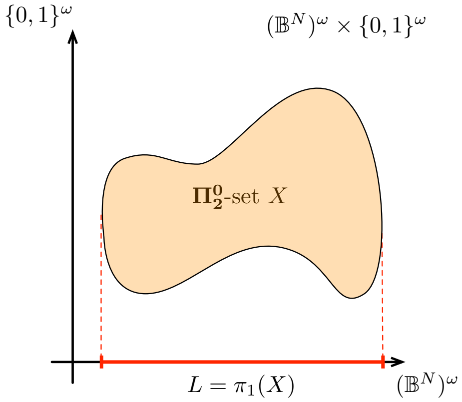

height ω_1
$\vdots$
$\mathbf{\Sigma}^0_3$
$\mathbf{\Pi}^0_3$
$BC(\mathbf{\Pi}^0_2)$
$\mathbf{\Sigma}^0_2$
$\mathbf{\Pi}^0_2$
$\mathbf{\Sigma}^0_1$
$\mathbf{\Pi}^0_1$

Topology

$$\begin{tikzcd}[column sep=small]
 & \vdots & \\
\boldsymbol{\Sigma}^0_3 & & \boldsymbol{\Pi}^0_3 \\
 & \boldsymbol{\Delta}^0_3 \arrow[ul, phantom, sloped, "\subsetneq"] \arrow[ur, phantom, sloped, "\supsetneq"] & \\
\boldsymbol{\Sigma}^0_2 \arrow[ur, phantom, sloped, "\subsetneq"] & & \boldsymbol{\Pi}^0_2 \arrow[ul, phantom, sloped, "\supsetneq"] \\
 & \boldsymbol{\Delta}^0_2 \arrow[ul, phantom, sloped, "\subsetneq"] \arrow[ur, phantom, sloped, "\supsetneq"] & \\
\boldsymbol{\Sigma}^0_1 \arrow[ur, phantom, sloped, "\subsetneq"] & & \boldsymbol{\Pi}^0_1 \arrow[ul, phantom, sloped, "\supsetneq"] \\
 & \boldsymbol{\Delta}^0_1 \arrow[ul, phantom, sloped, "\subsetneq"] \arrow[ur, phantom, sloped, "\supsetneq"] &
\end{tikzcd}$$

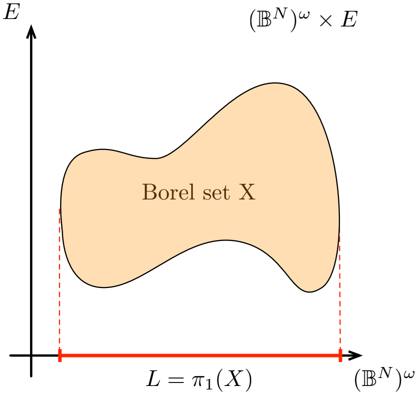
Topology

- An ω-language $L \subseteq (\mathbb{B}^N)^\omega$ is *analytic* ($\boldsymbol{\Sigma}_1^1$) iff it is the first projection of some $\boldsymbol{\Pi}_2^0$-set $X \subseteq (\mathbb{B}^N)^\omega \times \{0,1\}^\omega$.



Topology

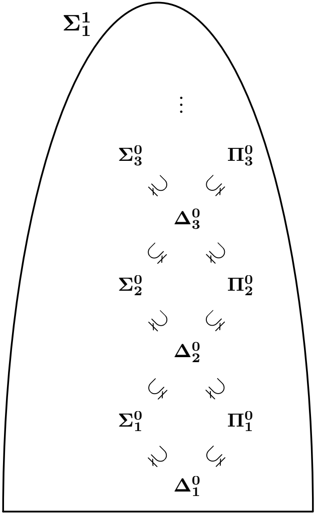
- An ω-language $L \subseteq (\mathbb{B}^N)^\omega$ is *analytic* ($\mathbf{\Sigma}^1_1$) iff it is the first projection of some Borel set $X \subseteq (\mathbb{B}^N)^\omega \times E$, where E is a Polish space.



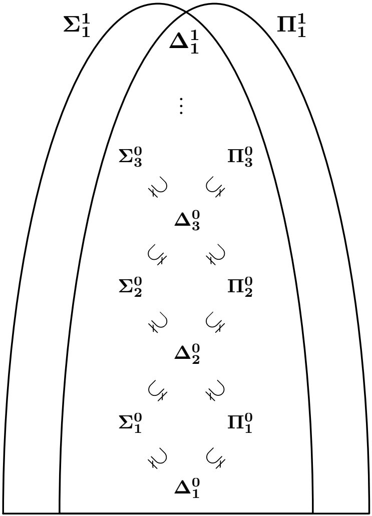
Topology

$$
\begin{tikzcd}[column sep=small]
 & \vdots & \\
\boldsymbol{\Sigma}^0_3 & & \boldsymbol{\Pi}^0_3 \\
 & \boldsymbol{\Delta}^0_3 \arrow[ul, hook] \arrow[ur, hook] & \\
\boldsymbol{\Sigma}^0_2 \arrow[ur, hook] & & \boldsymbol{\Pi}^0_2 \arrow[ul, hook] \\
 & \boldsymbol{\Delta}^0_2 \arrow[ul, hook] \arrow[ur, hook] & \\
\boldsymbol{\Sigma}^0_1 \arrow[ur, hook] & & \boldsymbol{\Pi}^0_1 \arrow[ul, hook] \\
 & \boldsymbol{\Delta}^0_1 \arrow[ul, hook] \arrow[ur, hook] &
\end{tikzcd}
$$

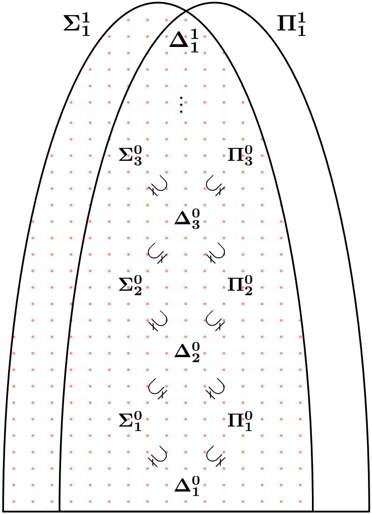
Topology


$\mathbf{\Sigma}^1_1$
$\vdots$
$\mathbf{\Sigma}^0_3$ $\mathbf{\Pi}^0_3$
$\mathbf{\Delta}^0_3$
$\mathbf{\Sigma}^0_2$ $\mathbf{\Pi}^0_2$
$\mathbf{\Delta}^0_2$
$\mathbf{\Sigma}^0_1$ $\mathbf{\Pi}^0_1$
$\mathbf{\Delta}^0_1$

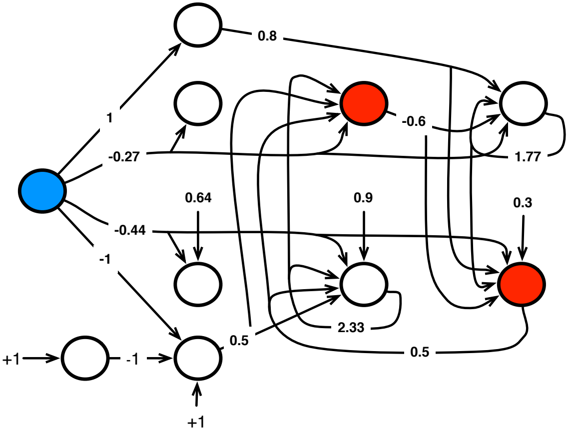
Topology


$\mathbf{\Sigma}_1^1$
$\mathbf{\Pi}_1^1$
$\mathbf{\Delta}_1^1$
$\vdots$
$\mathbf{\Sigma}_3^0$
$\mathbf{\Pi}_3^0$
$\mathbf{\Delta}_3^0$
$\mathbf{\Sigma}_2^0$
$\mathbf{\Pi}_2^0$
$\mathbf{\Delta}_2^0$
$\mathbf{\Sigma}_1^0$
$\mathbf{\Pi}_1^0$
$\mathbf{\Delta}_1^0$

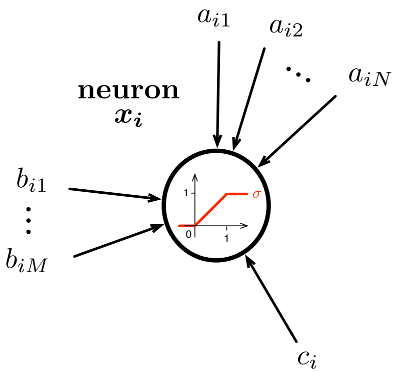
Topology


$\mathbf{\Sigma}^1_1$
$\mathbf{\Pi}^1_1$
$\mathbf{\Delta}^1_1$
$\vdots$
$\mathbf{\Sigma}^0_3$
$\mathbf{\Pi}^0_3$
$\mathbf{\Delta}^0_3$
$\mathbf{\Sigma}^0_2$
$\mathbf{\Pi}^0_2$
$\mathbf{\Delta}^0_2$
$\mathbf{\Sigma}^0_1$
$\mathbf{\Pi}^0_1$
$\mathbf{\Delta}^0_1$

Recurrent neural network

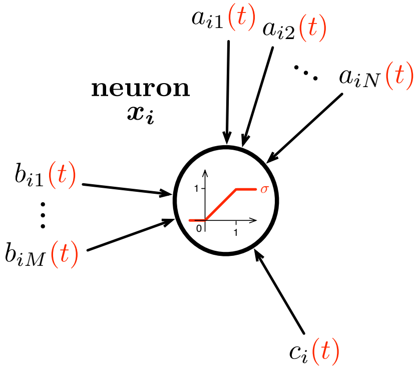


Static recurrent neural networks



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

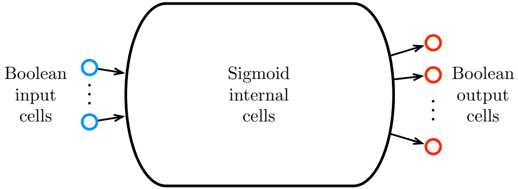
Evolving recurrent neural networks



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

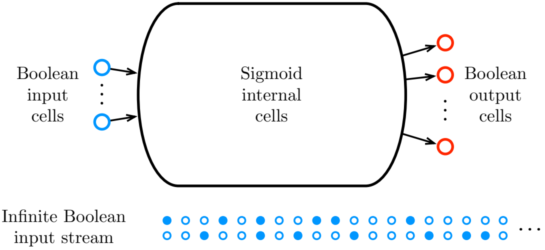
Hybrid recurrent neural networks

We consider RNNs with Boolean input cells, sigmoid internal cells, Boolean output cells, and working on infinite input streams.



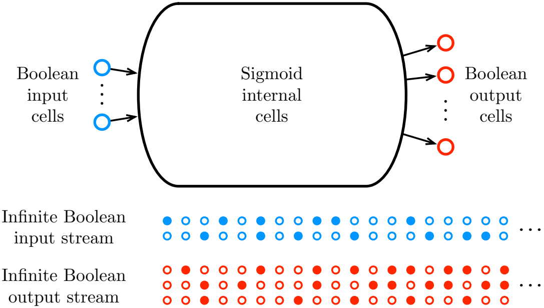
Hybrid recurrent neural networks

We consider RNNs with Boolean input cells, sigmoid internal cells, Boolean output cells, and working on infinite input streams.



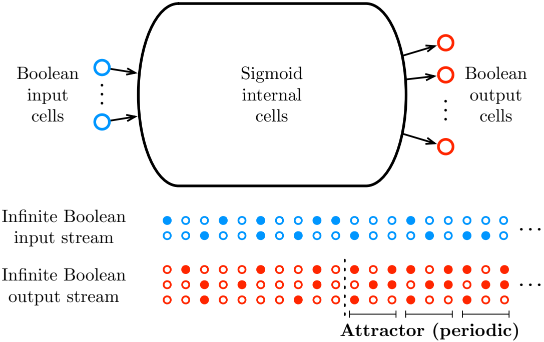
Hybrid recurrent neural networks

We consider RNNs with Boolean input cells, sigmoid internal cells, Boolean output cells, and working on infinite input streams.



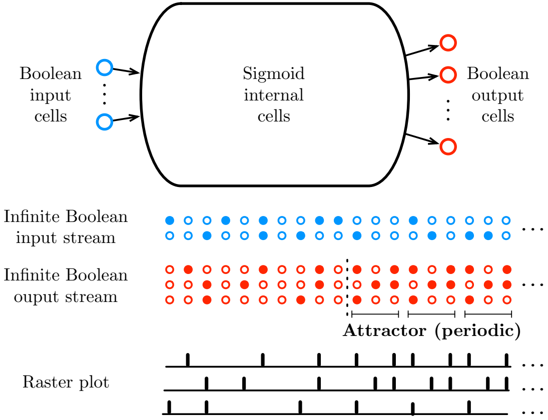
Hybrid recurrent neural networks

We consider RNNs with Boolean input cells, sigmoid internal cells, Boolean output cells, and working on infinite input streams.



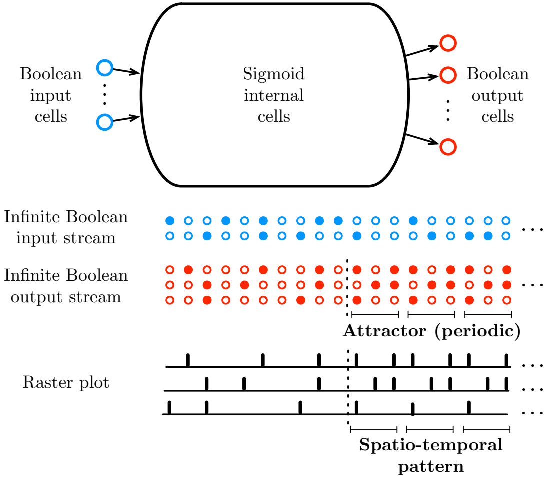
Hybrid recurrent neural networks

We consider RNNs with Boolean input cells, sigmoid internal cells, Boolean output cells, and working on infinite input streams.

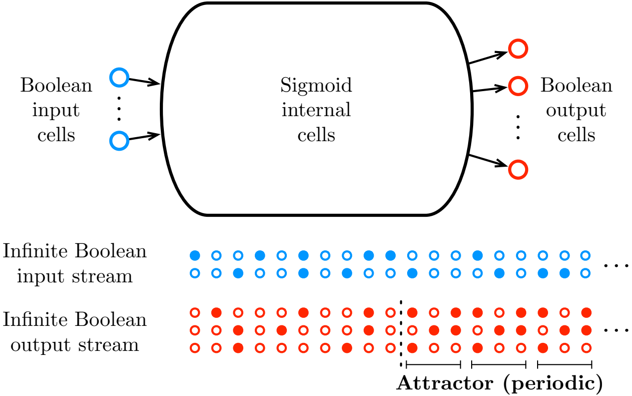


Hybrid recurrent neural networks

We consider RNNs with Boolean input cells, sigmoid internal cells, Boolean output cells, and working on infinite input streams.

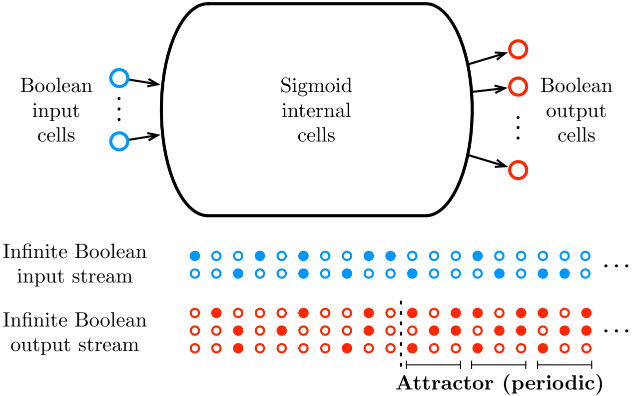


Hybrid recurrent neural networks



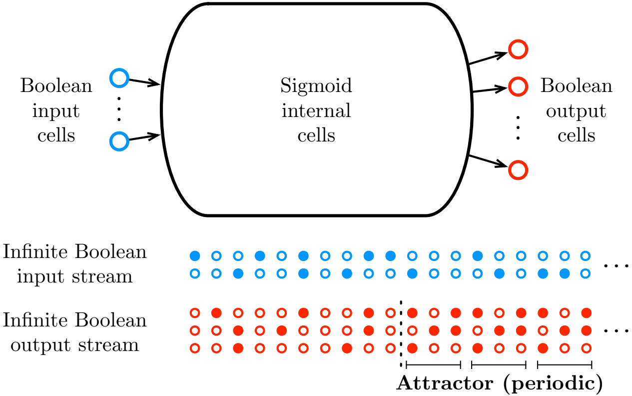
- The attractors are assumed to be classified into two possible kinds: *meaningful* or *spurious*.

Hybrid recurrent neural networks



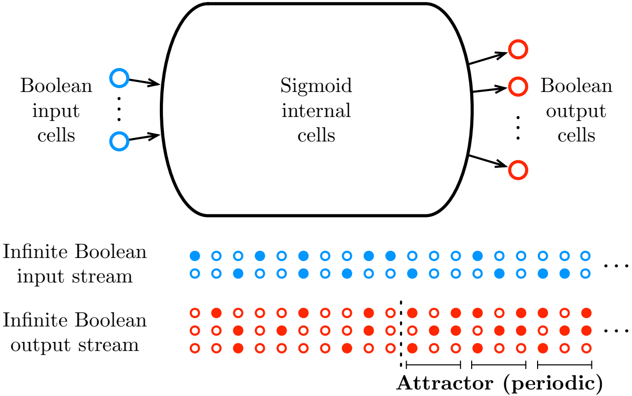
- An infinite Boolean input stream is *accepted* by $\mathcal{N}$ if the corresponding Boolean output stream visits a *meaningful* attractor.

Hybrid recurrent neural networks



- An infinite Boolean input stream is *rejected* by $\mathcal{N}$ if the corresponding Boolean output stream visits a *spurious* attractor.

Hybrid recurrent neural networks

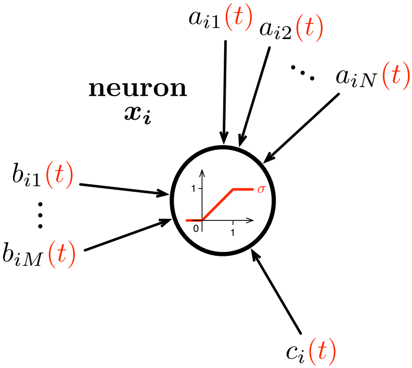


- The set of all input streams that are accepted by $\mathcal{N}$ is the *ω-language* recognized by $\mathcal{N}$.

Results

	Static	Bi-valued Evolving	Evolving
$\mathbb{Q}$	Turing (Muller) $\in BC(\mathbf{\Pi^0_2})$	super-Turing $= BC(\mathbf{\Pi^0_2})$	super-Turing $= BC(\mathbf{\Pi^0_2})$
$\mathbb{R}$	super-Turing $= BC(\mathbf{\Pi^0_2})$	super-Turing $= BC(\mathbf{\Pi^0_2})$	super-Turing $= BC(\mathbf{\Pi^0_2})$

Evolving recurrent neural networks



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

Nondeterminism

Suppose that the neural net contains K evolving synaptic weights:

- A K-dimensional vector $\vec{w}(t) = (w_1(t), \dots, w_K(t))^T \in \mathbb{Q}^K$ or $\mathbb{R}^K$ describes the value of the K synaptic weights at time t.
- A *possible evolution* is an infinite sequence
$$e = \vec{w}(0)\,\vec{w}(1)\,\vec{w}(2) \cdots \in (\mathbb{Q}^K)^\omega \text{ or } (\mathbb{R}^K)^\omega$$
which describes the synaptic weights at successive time steps.
- We suppose that each network is provided with a corresponding *evolution set*: a closed set $E \subseteq (\mathbb{Q}^K)^\omega$ or $(\mathbb{R}^K)^\omega$ describing all possible evolutions of the net.
- **Nondeterminism:** *At the beginning of the computation, the network selects one possible evolution $e \in E$, and sticks to it throughout its whole computational process.*

Nondeterminism

Suppose that the neural net contains K evolving synaptic weights:

- A K-dimensional vector $\vec{w}(t) = (w_1(t), \dots, w_K(t))^T \in \mathbb{Q}^K$ or $\mathbb{R}^K$ describes the value of the K synaptic weights at time t.
- A *possible evolution* is an infinite sequence

$$e = \vec{w}(0)\,\vec{w}(1)\,\vec{w}(2) \cdots \in (\mathbb{Q}^K)^\omega \text{ or } (\mathbb{R}^K)^\omega$$

  which describes the synaptic weights at successive time steps.
- We suppose that each network is provided with a corresponding *evolution set*: a closed set $E \subseteq (\mathbb{Q}^K)^\omega$ or $(\mathbb{R}^K)^\omega$ describing all possible evolutions of the net.
- **Nondeterminism:** *At the beginning of the computation, the network selects one possible evolution $e \in E$, and sticks to it throughout its whole computational process.*

Nondeterminism

Suppose that the neural net contains K evolving synaptic weights:

- A K-dimensional vector $\vec{w}(t) = (w_1(t), \ldots, w_K(t))^T \in \mathbb{Q}^K$ or $\mathbb{R}^K$ describes the value of the K synaptic weights at time t.
- A *possible evolution* is an infinite sequence

$$e = \vec{w}(0)\,\vec{w}(1)\,\vec{w}(2) \cdots \in (\mathbb{Q}^K)^\omega \text{ or } (\mathbb{R}^K)^\omega$$

  which describes the synaptic weights at successive time steps.
- We suppose that each network is provided with a corresponding *evolution set*: a closed set $E \subseteq (\mathbb{Q}^K)^\omega$ or $(\mathbb{R}^K)^\omega$ describing all possible evolutions of the net.
- **Nondeterminism:** *At the beginning of the computation, the network selects one possible evolution $e \in E$, and sticks to it throughout its whole computational process.*

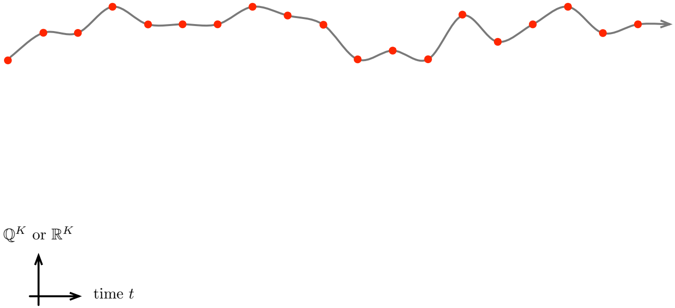
Nondeterminism

Suppose that the neural net contains K evolving synaptic weights:

- A K-dimensional vector $\vec{w}(t) = (w_1(t), \ldots, w_K(t))^T \in \mathbb{Q}^K$ or $\mathbb{R}^K$ describes the value of the K synaptic weights at time t.
- A *possible evolution* is an infinite sequence
$$e = \vec{w}(0)\,\vec{w}(1)\,\vec{w}(2) \cdots \in (\mathbb{Q}^K)^\omega \text{ or } (\mathbb{R}^K)^\omega$$
which describes the synaptic weights at successive time steps.
- We suppose that each network is provided with a corresponding *evolution set*: a closed set $E \subseteq (\mathbb{Q}^K)^\omega$ or $(\mathbb{R}^K)^\omega$ describing all possible evolutions of the net.
- **Nondeterminism:** *At the beginning of the computation, the network selects one possible evolution $e \in E$, and sticks to it throughout its whole computational process.*

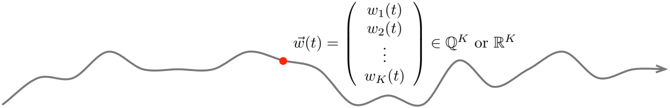
Nondeterminism

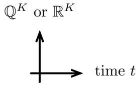
A possible **evolution** $e = \vec{w}(0)\,\vec{w}(1)\,\vec{w}(2)\cdots \in \mathbb{Q}^K$ or $\mathbb{R}^K$



Nondeterminism

A possible **evolution** $e = \vec{w}(0)\,\vec{w}(1)\,\vec{w}(2)\cdots \in \mathbb{Q}^K$ or $\mathbb{R}^K$

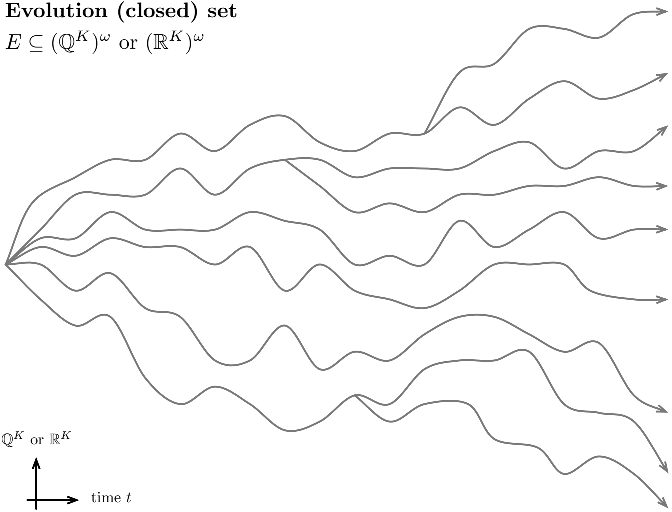




Nondeterminism

Evolution (closed) set

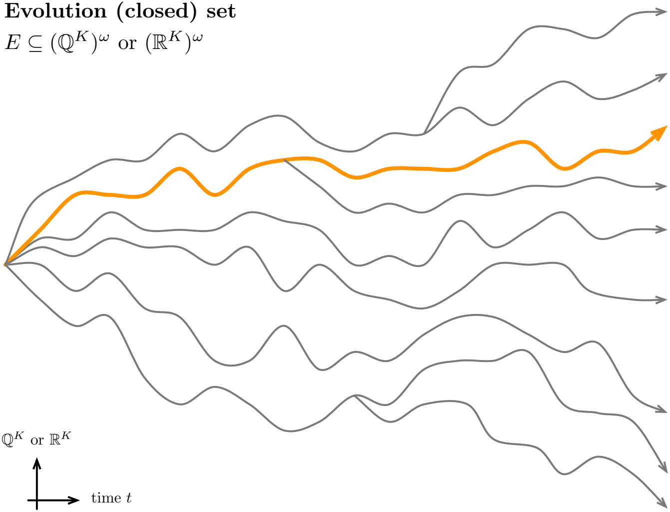
$E \subseteq (\mathbb{Q}^K)^\omega$ or $(\mathbb{R}^K)^\omega$



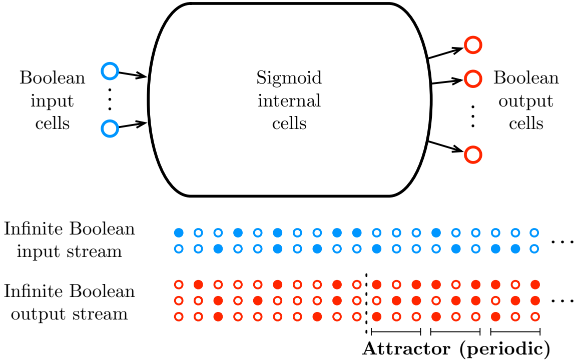
Nondeterminism

Evolution (closed) set

$E \subseteq (\mathbb{Q}^K)^\omega$ or $(\mathbb{R}^K)^\omega$

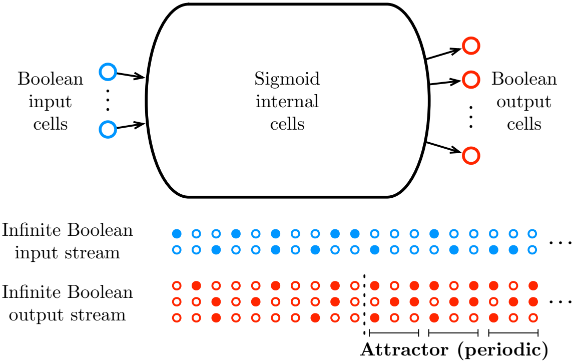


Nondet. hybrid recurrent neural networks



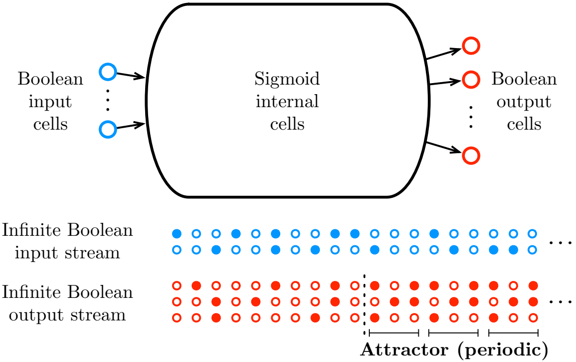
- The attractors are assumed to be classified into two possible kinds: *meaningful* or *spurious*.

Nondet. hybrid recurrent neural networks



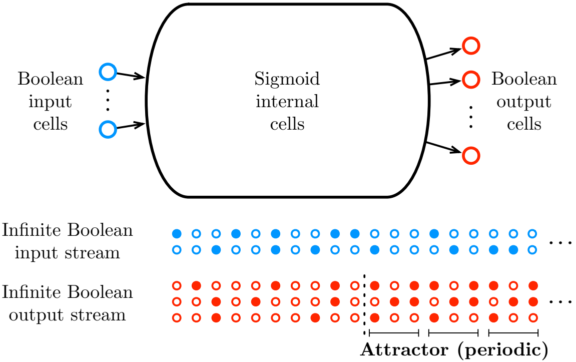
- An infinite Boolean input stream is *accepted* by $\mathcal{N}$ if *there exists a possible evolution* $e \in E$ such that the corresponding Boolean output stream visits a *meaningful* attractor.

Nondet. hybrid recurrent neural networks



- An infinite Boolean input stream is *rejected* by $\mathcal{N}$ if *for all possible evolution* $e \in E$, the corresponding Boolean output stream visits a *spurious* attractor.

Nondet. hybrid recurrent neural networks



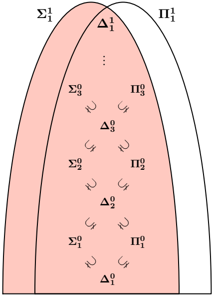
- The set of all input streams that are accepted by $\mathcal{N}$ is the ω-*language* recognized by $\mathcal{N}$.

Results

Theorem

Let $L \subseteq (\mathbb{B}^M)^\omega$. The following conditions are equivalent.

- $L \in \mathbf{\Sigma}^1_1$
- *L is recognizable by some nondeterministic evolving RNN[$\mathbb{Q}$]*
- *L is recognizable by some nondeterministic evolving RNN[$\mathbb{R}$]*

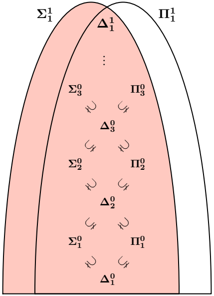


Results

Theorem

Let $L \subseteq (\mathbb{B}^M)^\omega$. The following conditions are equivalent.

- $L \in \mathbf{\Sigma}^1_1$
- L *is recognizable by some nondeterministic evolving RNN[$\mathbb{Q}$]*
- L *is recognizable by some nondeterministic evolving RNN[$\mathbb{R}$]*

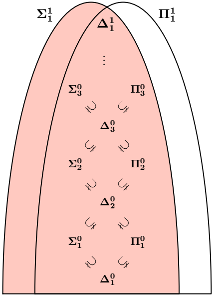


Results

Theorem

Let $L \subseteq (\mathbb{B}^M)^\omega$. The following conditions are equivalent.

- $L \in \boldsymbol{\Sigma}_1^1$
- L *is recognizable by some nondeterministic evolving RNN[$\mathbb{Q}$]*
- L *is recognizable by some nondeterministic evolving RNN[$\mathbb{R}$]*



Results

Theorem

Let $L \subseteq (\mathbb{B}^M)^\omega$. The following conditions are equivalent.

- $L \in \mathbf{\Sigma}^1_1$
- *L is recognizable by some nondeterministic evolving RNN[$\mathbb{Q}$]*
- *L is recognizable by some nondeterministic evolving RNN[$\mathbb{R}$]*

$$\begin{array}{c}
\mathbf{\Sigma}^1_1 \\
\rotatebox{90}{$\subseteq$} \\
\mathcal{L}(N\!-\!Ev\!-\!RNN[\mathbb{R}]s) \\
\rotatebox{90}{$\subseteq$} \\
\mathcal{L}(N\!-\!Ev\!-\!RNN[\mathbb{Q}]s) \\
\rotatebox{90}{$\subseteq$} \\
\mathbf{\Sigma}^1_1
\end{array}$$

Results

Theorem

Let $L \subseteq (\mathbb{B}^M)^\omega$. The following conditions are equivalent.

- $L \in \mathbf{\Sigma}^1_1$
- *L is recognizable by some nondeterministic evolving RNN[$\mathbb{Q}$]*
- *L is recognizable by some nondeterministic evolving RNN[$\mathbb{R}$]*

$$
\begin{array}{c}
\mathbf{\Sigma}^1_1 \\
\uparrow \\
\subseteq \\
\mathcal{L}(N{-}Ev{-}RNN[\mathbb{R}]s) \\
\uparrow \\
\subseteq \\
\mathcal{L}(N{-}Ev{-}RNN[\mathbb{Q}]s) \\
\uparrow \\
\subseteq \\
\mathbf{\Sigma}^1_1
\end{array}
$$

Results

Theorem

Let $L \subseteq (\mathbb{B}^M)^\omega$. The following conditions are equivalent.

- $L \in \mathbf{\Sigma}^1_1$
- *L is recognizable by some nondeterministic evolving RNN[$\mathbb{Q}$]*
- *L is recognizable by some nondeterministic evolving RNN[$\mathbb{R}$]*

$$\begin{array}{c}
\mathbf{\Sigma}^1_1 \\
\rotatebox{90}{$\subseteq$} \\
\mathcal{L}(N{-}Ev{-}RNN[\mathbb{R}]s) \\
\rotatebox{90}{$\subseteq$} \\
\mathcal{L}(N{-}Ev{-}RNN[\mathbb{Q}]s) \\
\rotatebox{90}{$\subseteq$} \\
\mathbf{\Sigma}^1_1
\end{array}$$

Results

Proposition

Let $L \in \mathbf{\Sigma}^1_1$. Then L is recognizable by some nondet. evolving RNN[$\mathbb{Q}$].

Proof (sketch):

- Since $L \in \Sigma^1_1$, there exists some Π^0_2 set $X \subseteq (\mathbb{B}^M)^\omega \times \{0,1\}^\omega$ such that $L = \pi_1(X)$.
- X can thus be encoded into some infinite word $w_X \in \{0,1\}^\omega$.
- We can then build an nondet. evolving RNN[$\mathbb{Q}$] – *whose one of its evolving synaptic weights follows w_X* – which, on input $s \in (\mathbb{B}^M)^\omega$ and evolution $e \in \{0,1\}^\omega$, enters a meaningful attractor iff $(s, e) \in X$.
- It follows that $L(\mathcal{N}) = \pi_1(X) = L$.

Results

Proposition

Let $L \in \mathbf{\Sigma}^1_1$. Then L is recognizable by some nondet. evolving RNN[$\mathbb{Q}$].

Proof (sketch):

- Since $L \in \mathbf{\Sigma}^1_1$, there exists some $\mathbf{\Pi}^0_2$ set $X \subseteq (\mathbb{B}^M)^\omega \times \{0,1\}^\omega$ such that $L = \pi_1(X)$.
- X can thus be encoded into some infinite word $w_X \in \{0,1\}^\omega$.
- We can then build an nondet. evolving RNN[$\mathbb{Q}$] – *whose one of its evolving synaptic weights follows w_X* – which, on input $s \in (\mathbb{B}^M)^\omega$ and evolution $e \in \{0,1\}^\omega$, enters a meaningful attractor iff $(s,e) \in X$.
- It follows that $L(\mathcal{N}) = \pi_1(X) = L$. $\square$

Results

Proposition

Let $L \in \mathbf{\Sigma}^1_1$. Then L is recognizable by some nondet. evolving RNN[$\mathbb{Q}$].

Proof (sketch):

- Since $L \in \mathbf{\Sigma}^1_1$, there exists some $\mathbf{\Pi}^0_2$ set $X \subseteq (\mathbb{B}^M)^\omega \times \{0,1\}^\omega$ such that $L = \pi_1(X)$.
- X can thus be encoded into some infinite word $w_X \in \{0,1\}^\omega$.
- We can then build an nondet. evolving RNN[$\mathbb{Q}$] – *whose one of its evolving synaptic weights follows w_X* – which, on input $s \in (\mathbb{B}^M)^\omega$ and evolution $e \in \{0,1\}^\omega$, enters a meaningful attractor iff $(s, e) \in X$.
- It follows that $L(\mathcal{N}) = \pi_1(X) = L$.

Results

Proposition

Let $L \in \mathbf{\Sigma}^1_1$. Then L is recognizable by some nondet. evolving RNN[$\mathbb{Q}$].

Proof (sketch):

- Since $L \in \mathbf{\Sigma}^1_1$, there exists some $\mathbf{\Pi}^0_2$ set $X \subseteq (\mathbb{B}^M)^\omega \times \{0,1\}^\omega$ such that $L = \pi_1(X)$.
- X can thus be encoded into some infinite word $w_X \in \{0,1\}^\omega$.
- We can then build an nondet. evolving RNN[$\mathbb{Q}$] – *whose one of its evolving synaptic weights follows* w_X – which, on input $s \in (\mathbb{B}^M)^\omega$ and evolution $e \in \{0,1\}^\omega$, enters a meaningful attractor iff $(s, e) \in X$.
- It follows that $L(\mathcal{N}) = \pi_1(X) = L$.

Results

Proposition

Let $L \in \mathbf{\Sigma}^1_1$. Then L is recognizable by some nondet. evolving RNN[$\mathbb{Q}$].

Proof (sketch):

- Since $L \in \mathbf{\Sigma}^1_1$, there exists some $\mathbf{\Pi}^0_2$ set $X \subseteq (\mathbb{B}^M)^\omega \times \{0,1\}^\omega$ such that $L = \pi_1(X)$.
- X can thus be encoded into some infinite word $w_X \in \{0,1\}^\omega$.
- We can then build an nondet. evolving RNN[$\mathbb{Q}$] – *whose one of its evolving synaptic weights follows* w_X – which, on input $s \in (\mathbb{B}^M)^\omega$ and evolution $e \in \{0,1\}^\omega$, enters a meaningful attractor iff $(s,e) \in X$.
- It follows that $L(\mathcal{N}) = \pi_1(X) = L$.

$\square$

Results

Proposition

Let $\mathcal{N}$ be some nondet. evolving RNN[$\mathbb{R}$]. Then $L(\mathcal{N}) \in \mathbf{\Sigma^1_1}$.

Proof (sketch):

- The function $f_{\mathcal{N}} : (\mathbb{B}^M)^\omega \times E \to (\mathbb{B}^P)^\omega$ associated with the dynamics of $\mathcal{N}$ is of Baire class 1 (preimage of a Σ^0_1 is a Σ^0_2).
- Accordingly, the ω-language $L(\mathcal{N})$ can be expressed as the first projection of a finite Boolean combination of $\mathbf{\Sigma^0_3}$ and $\mathbf{\Pi^0_3}$ sets (i.e., of a Borel set) of the Polish space $(\mathbb{B}^M)^\omega \times E$.
- Therefore, $L(\mathcal{N}) \in \mathbf{\Sigma^1_1}$.

Results

Proposition

Let $\mathcal{N}$ be some nondet. evolving RNN[$\mathbb{R}$]. Then $L(\mathcal{N}) \in \mathbf{\Sigma}^1_1$.

Proof (sketch):

- The function $f_{\mathcal{N}} : (\mathbb{B}^M)^\omega \times E \to (\mathbb{B}^P)^\omega$ associated with the dynamics of $\mathcal{N}$ is of Baire class 1 (preimage of a $\mathbf{\Sigma}^0_1$ is a $\mathbf{\Sigma}^0_2$).
- Accordingly, the ω-language $L(\mathcal{N})$ can be expressed as the first projection of a finite Boolean combination of $\mathbf{\Sigma}^0_3$ and $\mathbf{\Pi}^0_3$ sets (i.e., of a Borel set) of the Polish space $(\mathbb{B}^M)^\omega \times E$.
- Therefore, $L(\mathcal{N}) \in \mathbf{\Sigma}^1_1$. $\square$

Results

Proposition

Let $\mathcal{N}$ be some nondet. evolving RNN[$\mathbb{R}$]. Then $L(\mathcal{N}) \in \mathbf{\Sigma}_\mathbf{1}^\mathbf{1}$.

Proof (sketch):

- The function $f_\mathcal{N} : (\mathbb{B}^M)^\omega \times E \to (\mathbb{B}^P)^\omega$ associated with the dynamics of $\mathcal{N}$ is of Baire class 1 (preimage of a $\mathbf{\Sigma}_1^0$ is a $\mathbf{\Sigma}_2^0$).
- Accordingly, the ω-language $L(\mathcal{N})$ can be expressed as the first projection of a finite Boolean combination of $\mathbf{\Sigma_3^0}$ and $\mathbf{\Pi_3^0}$ sets (i.e., of a Borel set) of the Polish space $(\mathbb{B}^M)^\omega \times E$.
- Therefore, $L(\mathcal{N}) \in \mathbf{\Sigma}_1^1$. $\square$

Results

Proposition

Let $\mathcal{N}$ be some nondet. evolving RNN[$\mathbb{R}$]. Then $L(\mathcal{N}) \in \mathbf{\Sigma_1^1}$.

Proof (sketch):

- The function $f_{\mathcal{N}} : (\mathbb{B}^M)^\omega \times E \to (\mathbb{B}^P)^\omega$ associated with the dynamics of $\mathcal{N}$ is of Baire class 1 (preimage of a $\mathbf{\Sigma}_1^0$ is a $\mathbf{\Sigma}_2^0$).
- Accordingly, the ω-language $L(\mathcal{N})$ can be expressed as the first projection of a finite Boolean combination of $\mathbf{\Sigma_3^0}$ and $\mathbf{\Pi_3^0}$ sets (i.e., of a Borel set) of the Polish space $(\mathbb{B}^M)^\omega \times E$.
- Therefore, $L(\mathcal{N}) \in \mathbf{\Sigma_1^1}$. $\square$

Results

	Bi-valued Evolving	Evolving
$\mathbb{Q}$	super-Turing $= \mathbf{\Sigma^1_1}$	super-Turing $= \mathbf{\Sigma^1_1}$
$\mathbb{R}$	super-Turing $= \mathbf{\Sigma^1_1}$	super-Turing $= \mathbf{\Sigma^1_1}$

Summary

DET.	Static	Bi-valued Evolving	Evolving
$\mathbb{Q}$	Turing (Muller) $\in BC(\mathbf{\Pi_2^0})$	super-Turing $= BC(\mathbf{\Pi_2^0})$	super-Turing $= BC(\mathbf{\Pi_2^0})$
$\mathbb{R}$	super-Turing $= BC(\mathbf{\Pi_2^0})$	super-Turing $= BC(\mathbf{\Pi_2^0})$	super-Turing $= BC(\mathbf{\Pi_2^0})$

NONDET.	Bi-valued Evolving	Evolving
$\mathbb{Q}$	super-Turing $= \mathbf{\Sigma_1^1}$	super-Turing $= \mathbf{\Sigma_1^1}$
$\mathbb{R}$	super-Turing $= \mathbf{\Sigma_1^1}$	super-Turing $= \mathbf{\Sigma_1^1}$

Conclusion

- We provided a characterization of the expressive power of recurrent neural networks in terms of their attractor dynamics.
- In general, the super-Turing computational capabilities of neural models raises the question of *hypercomputation*.
- Current physical theories are consistent with the possibility of hypercomputational systems (quantum, relativistic, etc.). No such systems are currently feasible or harnessable.
- Philosophical and scientific literature about hypercomputation is however flourishing.

Conclusion

- We provided a characterization of the expressive power of recurrent neural networks in terms of their attractor dynamics.
- In general, the super-Turing computational capabilities of neural models raises the question of *hypercomputation*.
- Current physical theories are consistent with the possibility of hypercomputational systems (quantum, relativistic, etc.). No such systems are currently feasible or harnessable.
- Philosophical and scientific literature about hypercomputation is however flourishing.

Conclusion

- We provided a characterization of the expressive power of recurrent neural networks in terms of their attractor dynamics.
- In general, the super-Turing computational capabilities of neural models raises the question of *hypercomputation*.
- Current physical theories are consistent with the possibility of hypercomputational systems (quantum, relativistic, etc.). No such systems are currently feasible or harnessable.
- Philosophical and scientific literature about hypercomputation is however flourishing.

Conclusion

- We provided a characterization of the expressive power of recurrent neural networks in terms of their attractor dynamics.
- In general, the super-Turing computational capabilities of neural models raises the question of *hypercomputation*.
- Current physical theories are consistent with the possibility of hypercomputational systems (quantum, relativistic, etc.). No such systems are currently feasible or harnessable.
- Philosophical and scientific literature about hypercomputation is however flourishing.

Summary

DET.	Static	Bi-valued Evolving	Evolving
$\mathbb{Q}$	Turing (Muller) $\in BC(\mathbf{\Pi_2^0})$	super-Turing $= BC(\mathbf{\Pi_2^0})$	super-Turing $= BC(\mathbf{\Pi_2^0})$
$\mathbb{R}$	super-Turing $= BC(\mathbf{\Pi_2^0})$	super-Turing $= BC(\mathbf{\Pi_2^0})$	super-Turing $= BC(\mathbf{\Pi_2^0})$

NONDET.	Bi-valued Evolving	Evolving
$\mathbb{Q}$	super-Turing $= \mathbf{\Sigma_1^1}$	super-Turing $= \mathbf{\Sigma_1^1}$
$\mathbb{R}$	super-Turing $= \mathbf{\Sigma_1^1}$	super-Turing $= \mathbf{\Sigma_1^1}$