A Hierarchical Classification of First-Order Recurrent Neural Networks

Jérémie Cabessa
Joint work with Alessandro Villa

University Joseph Fourier – Grenoble 1

5 November 2009

Outline

Introduction

The fields of artificial neural networks and theoretical computer science have been linked since their inception (McCulloch and Pitts 1943, Kleene 1956, Minsky 1967).

Synaptic weights	Activation function	Computational power
rational	hard-threshold	finite state automaton
rational	(linear) sigmoid	Turing machine
real	(linear) sigmoid	super-Turing machine

We provide a refined classification of first-order recurrent neural networks with rational weights and hard-threshold activation function.

Introduction

The fields of artificial neural networks and theoretical computer science have been linked since their inception (McCulloch and Pitts 1943, Kleene 1956, Minsky 1967).

Synaptic weights	Activation function	Computational power
rational	hard-threshold	finite state automaton
rational	(linear) sigmoid	Turing machine
real	(linear) sigmoid	super-Turing machine

We provide a refined classification of first-order recurrent neural networks with rational weights and hard-threshold activation function.

Introduction

The fields of artificial neural networks and theoretical computer science have been linked since their inception (McCulloch and Pitts 1943, Kleene 1956, Minsky 1967).

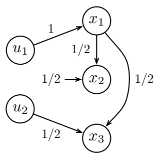
Synaptic weights	Activation function	Computational power
rational	hard-threshold	finite state automaton
rational	(linear) sigmoid	Turing machine
real	(linear) sigmoid	super-Turing machine

We provide a refined classification of first-order recurrent neural networks with rational weights and hard-threshold activation function.

First-order recurrent neural networks

Definition

A RNN is a tuple $\mathcal{N} = (X, U, a, b, c)$ where:



- X is a finite set of activation cells,
- U is a finite set of input cells,
- $a : X \times X \to \mathbb{Q}$ describes the synaptic weights,
- $b : X \times U \to \mathbb{Q}$ describes the input synaptic weights,
- $c : X \to \mathbb{Q}$ describes the incoming background activity.

The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{ij} \cdot x_j(t) + \sum_{j=1}^{M} b_{ij} \cdot u_j(t) + c_i\right), \quad \text{where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

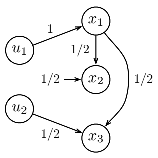
The dynamic of the whole $\mathcal{N}$ network is is given by

$$\mathbf{x(t+1)} = \sigma\left(A \cdot \mathbf{x(t)} + B \cdot \mathbf{u(t)} + \mathbf{c}\right)$$

First-order recurrent neural networks

Definition

A RNN is a tuple $\mathcal{N} = (X, U, a, b, c)$ where:



- X is a finite set of activation cells,
- U is a finite set of input cells,
- $a : X \times X \rightarrow \mathbb{Q}$ describes the synaptic weights,
- $b : X \times U \rightarrow \mathbb{Q}$ describes the input synaptic weights,
- $c : X \rightarrow \mathbb{Q}$ describes the incoming background activity.

The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{ij} \cdot x_j(t) + \sum_{j=1}^{M} b_{ij} \cdot u_j(t) + c_i\right), \quad \text{where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

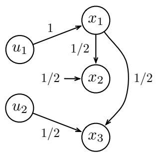
The dynamic of the whole $\mathcal{N}$ network is is given by

$$\mathbf{x(t+1)} = \sigma\left(A \cdot \mathbf{x(t)} + B \cdot \mathbf{u(t)} + \mathbf{c}\right)$$

First-order recurrent neural networks

Definition

A RNN is a tuple $\mathcal{N} = (X, U, a, b, c)$ where:



- X is a finite set of activation cells,
- U is a finite set of input cells,
- $a : X \times X \to \mathbb{Q}$ describes the synaptic weights,
- $b : X \times U \to \mathbb{Q}$ describes the input synaptic weights,
- $c : X \to \mathbb{Q}$ describes the incoming background activity.

The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{ij} \cdot x_j(t) + \sum_{j=1}^{M} b_{ij} \cdot u_j(t) + c_i\right), \quad \text{where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1\,, \\ 1 & \text{if } x \geq 1. \end{cases}$$

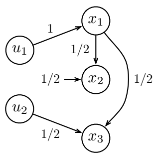
The dynamic of the whole $\mathcal{N}$ network is is given by

$$\mathbf{x(t+1)} = \sigma\left(A \cdot \mathbf{x(t)} + B \cdot \mathbf{u(t)} + \mathbf{c}\right)$$

First-order recurrent neural networks

Definition

A RNN is a tuple $\mathcal{N} = (X, U, a, b, c)$ where:



- X is a finite set of activation cells,
- U is a finite set of input cells,
- $a : X \times X \to \mathbb{Q}$ describes the synaptic weights,
- $b : X \times U \to \mathbb{Q}$ describes the input synaptic weights,
- $c : X \to \mathbb{Q}$ describes the incoming background activity.

The dynamic of cell i is given by

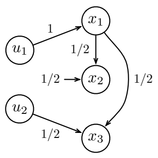
$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{ij} \cdot x_j(t) + \sum_{j=1}^{M} b_{ij} \cdot u_j(t) + c_i\right), \quad \text{where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

The dynamic of the whole $\mathcal{N}$ network is is given by

$$\mathbf{x(t+1)} = \sigma\left(A \cdot \mathbf{x(t)} + B \cdot \mathbf{u(t)} + \mathbf{c}\right)$$

First-order recurrent neural networks

Definition



A RNN is a tuple $\mathcal{N} = (X, U, a, b, c)$ where:

- X is a finite set of activation cells,
- U is a finite set of input cells,
- $a : X \times X \to \mathbb{Q}$ describes the synaptic weights,
- $b : X \times U \to \mathbb{Q}$ describes the input synaptic weights,
- $c : X \to \mathbb{Q}$ describes the incoming background activity.

The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{ij} \cdot x_j(t) + \sum_{j=1}^{M} b_{ij} \cdot u_j(t) + c_i\right), \quad \text{where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

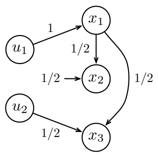
The dynamic of the whole $\mathcal{N}$ network is is given by

$$\mathbf{x(t+1)} = \sigma\left(A \cdot \mathbf{x(t)} + B \cdot \mathbf{u(t)} + \mathbf{c}\right)$$

First-order recurrent neural networks

Definition

A RNN is a tuple $\mathcal{N} = (X, U, a, b, c)$ where:



- X is a finite set of activation cells,
- U is a finite set of input cells,
- $a : X \times X \to \mathbb{Q}$ describes the synaptic weights,
- $b : X \times U \to \mathbb{Q}$ describes the input synaptic weights,
- $c : X \to \mathbb{Q}$ describes the incoming background activity.

The dynamic of cell i is given by

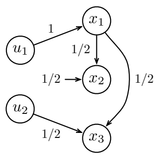
$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{ij} \cdot x_j(t) + \sum_{j=1}^{M} b_{ij} \cdot u_j(t) + c_i\right), \quad \text{where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1\,, \\ 1 & \text{if } x \geq 1. \end{cases}$$

The dynamic of the whole $\mathcal{N}$ network is is given by

$$\mathbf{x(t+1)} = \sigma\left(A \cdot \mathbf{x(t)} + B \cdot \mathbf{u(t)} + \mathbf{c}\right)$$

First-order recurrent neural networks

Definition



A RNN is a tuple $\mathcal{N} = (X, U, a, b, c)$ where:

- X is a finite set of activation cells,
- U is a finite set of input cells,
- $a : X \times X \to \mathbb{Q}$ describes the synaptic weights,
- $b : X \times U \to \mathbb{Q}$ describes the input synaptic weights,
- $c : X \to \mathbb{Q}$ describes the incoming background activity.

The dynamic of cell i is given by

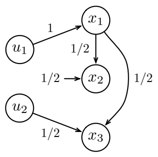
$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{ij} \cdot x_j(t) + \sum_{j=1}^{M} b_{ij} \cdot u_j(t) + c_i\right), \quad \text{where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1\,, \\ 1 & \text{if } x \geq 1. \end{cases}$$

The dynamic of the whole $\mathcal{N}$ network is is given by

$$\mathbf{x(t+1)} = \sigma\left(A \cdot \mathbf{x(t)} + B \cdot \mathbf{u(t)} + \mathbf{c}\right)$$

First-order recurrent neural networks

Definition



A RNN is a tuple $\mathcal{N} = (X, U, a, b, c)$ where:

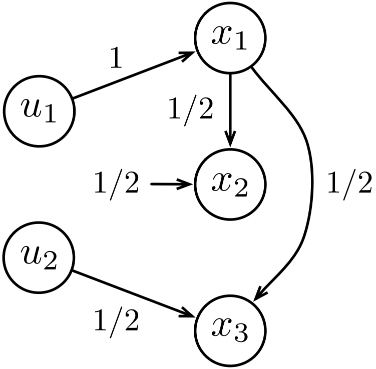
- X is a finite set of activation cells,
- U is a finite set of input cells,
- $a : X \times X \longrightarrow \mathbb{Q}$ describes the synaptic weights,
- $b : X \times U \longrightarrow \mathbb{Q}$ describes the input synaptic weights,
- $c : X \longrightarrow \mathbb{Q}$ describes the incoming background activity.

The dynamic of cell i is given by

$$x_i(t+1) = \sigma\left(\sum_{j=1}^{N} a_{ij} \cdot x_j(t) + \sum_{j=1}^{M} b_{ij} \cdot u_j(t) + c_i\right), \quad \text{where } \sigma(x) = \begin{cases} 0 & \text{if } x < 1, \\ 1 & \text{if } x \geq 1. \end{cases}$$

The dynamic of the whole $\mathcal{N}$ network is is given by

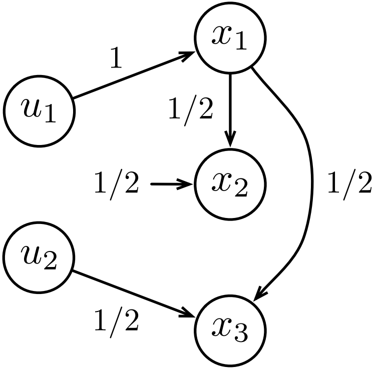
$$\mathbf{x(t+1)} = \sigma\left(A \cdot \mathbf{x(t)} + B \cdot \mathbf{u(t)} + \mathbf{c}\right)$$



Every stimulation s induced an evolution e_s. For instance:

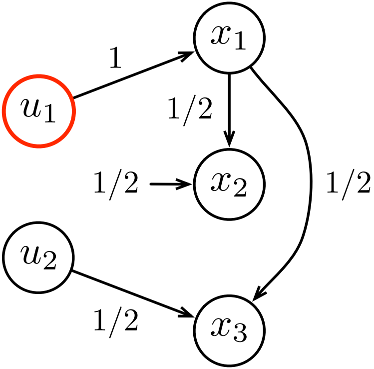
$$\text{stimulation} \quad s = \begin{pmatrix}1\\0\end{pmatrix} \begin{pmatrix}0\\1\end{pmatrix} \begin{pmatrix}1\\1\end{pmatrix} \begin{pmatrix}0\\1\end{pmatrix} \begin{pmatrix}1\\0\end{pmatrix} \begin{pmatrix}0\\1\end{pmatrix} \cdots\cdots$$

$$\text{evolution} \quad e_s = \begin{pmatrix}0\\0\\0\end{pmatrix} \begin{pmatrix}1\\0\\0\end{pmatrix} \begin{pmatrix}0\\1\\1\end{pmatrix} \begin{pmatrix}1\\0\\0\end{pmatrix} \begin{pmatrix}0\\1\\1\end{pmatrix} \begin{pmatrix}1\\0\\0\end{pmatrix} \cdots\cdots$$

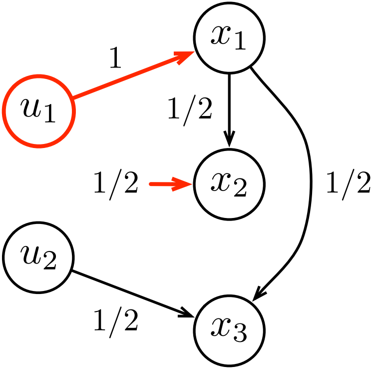


Every stimulation s induced an evolution e_s. For instance:

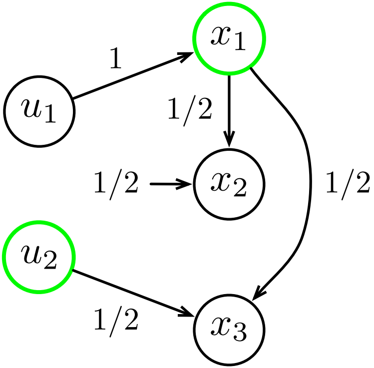
stimulation	$s =$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



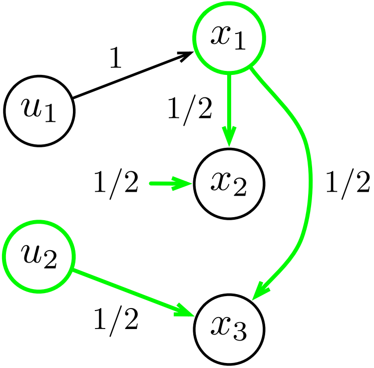
time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\binom{1}{0}$	$\binom{0}{1}$	$\binom{1}{1}$	$\binom{0}{1}$	$\binom{1}{0}$	$\binom{0}{1}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



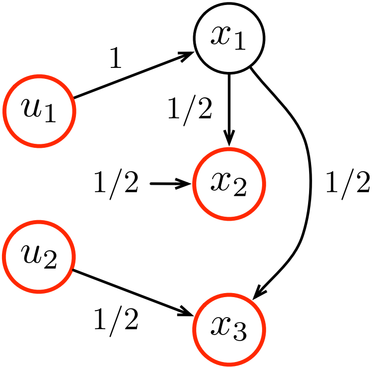
time steps	$t =$	0	1	2	3	4	5	······
stimulation	$s =$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	······
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	······



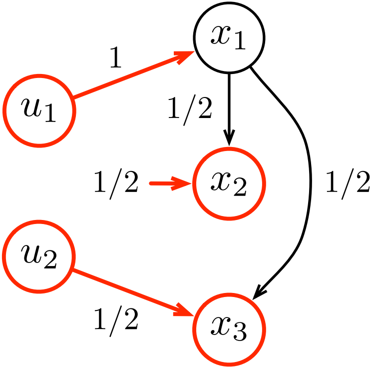
time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\binom{1}{0}$	$\binom{0}{1}$	$\binom{1}{1}$	$\binom{0}{1}$	$\binom{1}{0}$	$\binom{0}{1}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



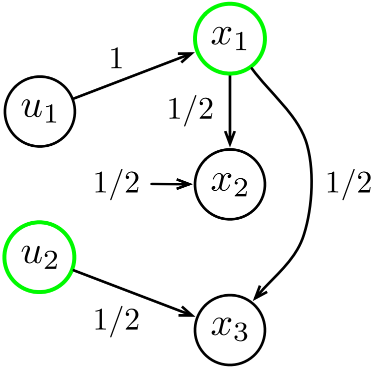
time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



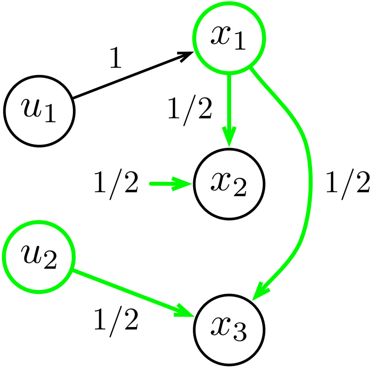
time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\binom{1}{0}$	$\binom{0}{1}$	$\binom{1}{1}$	$\binom{0}{1}$	$\binom{1}{0}$	$\binom{0}{1}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



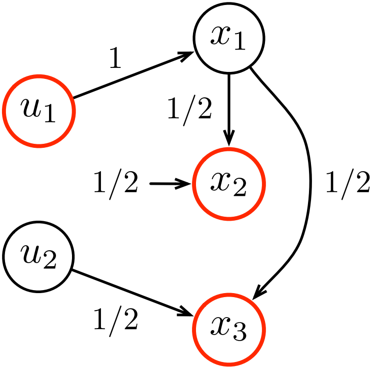
time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



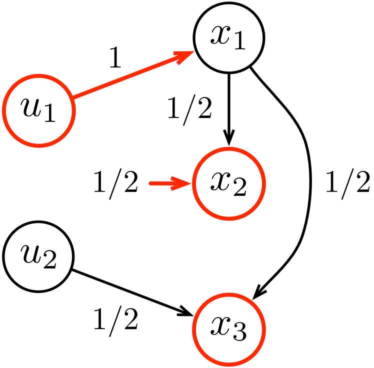
time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



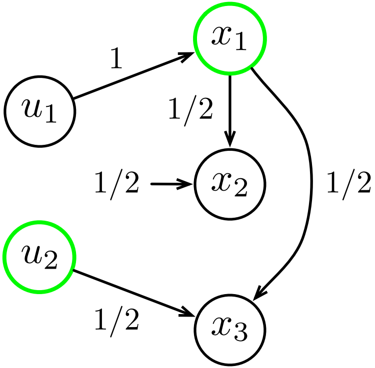
time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\binom{1}{0}$	$\binom{0}{1}$	$\binom{1}{1}$	$\binom{0}{1}$	$\binom{1}{0}$	$\binom{0}{1}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



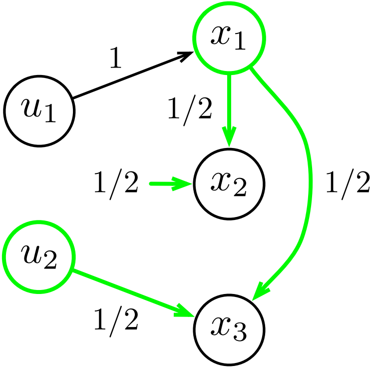
time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\binom{1}{0}$	$\binom{0}{1}$	$\binom{1}{1}$	$\binom{0}{1}$	$\binom{1}{0}$	$\binom{0}{1}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\binom{1}{0}$	$\binom{0}{1}$	$\binom{1}{1}$	$\binom{0}{1}$	$\binom{1}{0}$	$\binom{0}{1}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$



time steps	$t =$	0	1	2	3	4	5	$\cdots\cdots$
stimulation	$s =$	$\binom{1}{0}$	$\binom{0}{1}$	$\binom{1}{1}$	$\binom{0}{1}$	$\binom{1}{0}$	$\binom{0}{1}$	$\cdots\cdots$
evolution	$e_s =$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\0\\0\end{pmatrix}$	$\cdots\cdots$

- For every infinite stimulation s, let $Attr(e_s)$ denote the set of states of $\mathcal{N}$ that appear infinitely often in the evolution e_s.

$Attr(e_s)$ is called the *attractor* of the evolution e_s

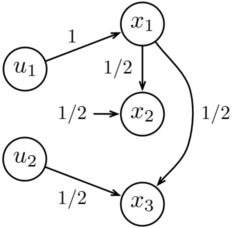
$$s = \left[\begin{pmatrix}1\\1\end{pmatrix}\right]^\omega \qquad e_s = \begin{pmatrix}0\\0\\0\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}\left[\begin{pmatrix}1\\1\\1\end{pmatrix}\right]^\omega \qquad Attr(e_s) = \left\{\begin{pmatrix}1\\1\\1\end{pmatrix}\right\}$$

$$s = \left[\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\right]^\omega \qquad e_s = \begin{pmatrix}0\\0\\0\end{pmatrix}\left[\begin{pmatrix}1\\0\\0\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix}\right]^\omega \qquad Attr(e_s) = \left\{\begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}\right\}$$

$$s = \left[\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\right]^\omega \qquad e_s = \begin{pmatrix}0\\0\\0\end{pmatrix}\left[\begin{pmatrix}1\\0\\0\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix}\right]^\omega \qquad Attr(e_s) = \left\{\begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}1\\1\\1\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}\right\}$$

- For every infinite stimulation s, let $Attr(e_s)$ denote the set of states of $\mathcal{N}$ that appear infinitely often in the evolution e_s.

$Attr(e_s)$ is called the *attractor* of the evolution e_s



$$s = \left[\begin{pmatrix}1\\1\end{pmatrix}\right]^\omega \qquad e_s = \begin{pmatrix}0\\0\\0\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}\left[\begin{pmatrix}1\\1\\1\end{pmatrix}\right]^\omega \qquad Attr(e_s) = \left\{\begin{pmatrix}1\\1\\1\end{pmatrix}\right\}$$

$$s = \left[\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\right]^\omega \qquad e_s = \begin{pmatrix}0\\0\\0\end{pmatrix}\left[\begin{pmatrix}1\\0\\0\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix}\right]^\omega \qquad Attr(e_s) = \left\{\begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}\right\}$$

$$s = \left[\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\right]^\omega \qquad e_s = \begin{pmatrix}0\\0\\0\end{pmatrix}\left[\begin{pmatrix}1\\0\\0\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix}\right]^\omega \qquad Attr(e_s) = \left\{\begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}1\\1\\1\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}\right\}$$

- We assume that all attractors can be classified as either *meaningful* or *spurious*.
- An infinite evolution e_s is called *meaningful* if $Attr(e_s)$ is meaningful and e_s is *spurious* if $Attr(e_s)$ is spurious.
- An infinite stimulation s is called *meaningful* if it induces a meaningful evolution e_s and s is *spurious* if it induces a meaningful evolution e_s.
- Given a network $\mathcal{N}$, we let $MeanStim(\mathcal{N})$ denote the set of all meaningful infinite stimulations of $\mathcal{N}$.

- We assume that all attractors can be classified as either *meaningful* or *spurious*.
- An infinite evolution e_s is called *meaningful* if $Attr(e_s)$ is meaningful and e_s is *spurious* if $Attr(e_s)$ is spurious.
- An infinite stimulation s is called *meaningful* if it induces a meaningful evolution e_s and s is *spurious* if it induces a meaningful evolution e_s.
- Given a network $\mathcal{N}$, we let $MeanStim(\mathcal{N})$ denote the set of all meaningful infinite stimulations of $\mathcal{N}$.

- We assume that all attractors can be classified as either *meaningful* or *spurious*.
- An infinite evolution e_s is called *meaningful* if $Attr(e_s)$ is meaningful and e_s is *spurious* if $Attr(e_s)$ is spurious.
- An infinite stimulation s is called *meaningful* if it induces a meaningful evolution e_s and s is *spurious* if it induces a meaningful evolution e_s.
- Given a network $\mathcal{N}$, we let $MeanStim(\mathcal{N})$ denote the set of all meaningful infinite stimulations of $\mathcal{N}$.

- We assume that all attractors can be classified as either *meaningful* or *spurious*.
- An infinite evolution e_s is called *meaningful* if $Attr(e_s)$ is meaningful and e_s is *spurious* if $Attr(e_s)$ is spurious.
- An infinite stimulation s is called *meaningful* if it induces a meaningful evolution e_s and s is *spurious* if it induces a meaningful evolution e_s.
- Given a network $\mathcal{N}$, we let $MeanStim(\mathcal{N})$ denote the set of all meaningful infinite stimulations of $\mathcal{N}$.

A classification of recurrent neural networks

Let $\mathcal{N}$ and $\mathcal{N}'$ be two RNN, we set:

$$\mathcal{N} \leq_W \mathcal{N}' \quad \text{iff} \quad \text{there exists } f : Stim(\mathcal{N}) \to Stim(\mathcal{N}') \text{ continuous s.t.}$$
$$s \in MeanStim(\mathcal{N}) \Leftrightarrow f(s) \in MeanStim(\mathcal{N}')$$

Then as usual

$$\mathcal{N} <_W \mathcal{N}' \quad \text{iff} \quad \mathcal{N} \leq_W \mathcal{N}' \text{ and } \mathcal{N}' \not\leq_W \mathcal{N}$$
$$\mathcal{N} \equiv_W \mathcal{N}' \quad \text{iff} \quad \mathcal{N} \leq_W \mathcal{N}' \text{ and } \mathcal{N}' \leq_W \mathcal{N}$$

Definition

The collection of all RNN ordered by $\leq_W$ is called *the RNN hierarchy*.

A classification of recurrent neural networks

Let $\mathcal{N}$ and $\mathcal{N}'$ be two RNN, we set:

$$\begin{aligned} \mathcal{N} \leq_W \mathcal{N}' \quad \text{iff} \quad & \text{there exists } f : Stim(\mathcal{N}) \to Stim(\mathcal{N}') \text{ continuous s.t.} \\ & s \in MeanStim(\mathcal{N}) \Leftrightarrow f(s) \in MeanStim(\mathcal{N}') \end{aligned}$$

Then as usual

$$\begin{aligned} \mathcal{N} <_W \mathcal{N}' \quad & \text{iff} \quad \mathcal{N} \leq_W \mathcal{N}' \text{ and } \mathcal{N}' \not\leq_W \mathcal{N} \\ \mathcal{N} \equiv_W \mathcal{N}' \quad & \text{iff} \quad \mathcal{N} \leq_W \mathcal{N}' \text{ and } \mathcal{N}' \leq_W \mathcal{N} \end{aligned}$$

Definition

The collection of all RNN ordered by $\leq_W$ is called *the RNN hierarchy*.

A classification of recurrent neural networks

Let $\mathcal{N}$ and $\mathcal{N}'$ be two RNN, we set:

$$\mathcal{N} \leq_W \mathcal{N}' \quad \text{iff} \quad \begin{array}{l} \text{there exists } f : \mathit{Stim}(\mathcal{N}) \to \mathit{Stim}(\mathcal{N}') \text{ continuous s.t.} \\ s \in \mathit{MeanStim}(\mathcal{N}) \Leftrightarrow f(s) \in \mathit{MeanStim}(\mathcal{N}') \end{array}$$

Then as usual

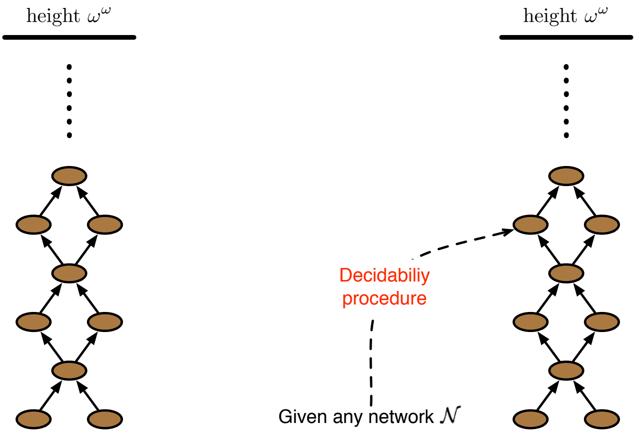
$$\begin{array}{lll} \mathcal{N} <_W \mathcal{N}' & \text{iff} & \mathcal{N} \leq_W \mathcal{N}' \text{ and } \mathcal{N}' \not\leq_W \mathcal{N} \\ \mathcal{N} \equiv_W \mathcal{N}' & \text{iff} & \mathcal{N} \leq_W \mathcal{N}' \text{ and } \mathcal{N}' \leq_W \mathcal{N} \end{array}$$

Definition

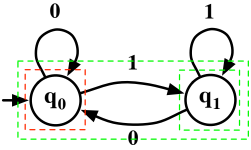
The collection of all RNN ordered by $\leq_W$ is called *the RNN hierarchy*.

Theorem

- *The RNN hierarchy is well-founded, has width* 2 *and height* ω^ω.
- *The RNN hierarchy is decidable.*



Proof: We consider Muller automata. . .

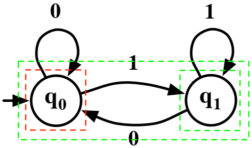


We show that, for any set of k-dimensional infinite stimulations $L \subseteq (\mathbb{B}^k)^\omega$, we have that:

L is to the set of meaningful stimulation of some RNN *if and only if* L is accepted by some deterministic Muller automaton.

Therefore the classification of RNN by $\leq_W$ coincide with the classification of Muller automata by $\leq_W$, namely a decidable pre-well ordering of width 2 and height ω^ω.

Proof: We consider Muller automata...

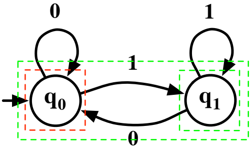


We show that, for any set of k-dimensional infinite stimulations $L \subseteq (\mathbb{B}^k)^\omega$, we have that:

L is to the set of meaningful stimulation of some RNN *if and only if* L is accepted by some deterministic Muller automaton.

Therefore the classification of RNN by $\leq_W$ coincide with the classification of Muller automata by $\leq_W$, namely a decidable pre-well ordering of width 2 and height ω^ω.

Proof: We consider Muller automata...

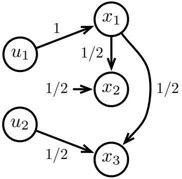


We show that, for any set of k-dimensional infinite stimulations $L \subseteq (\mathbb{B}^k)^\omega$, we have that:

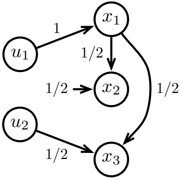
L is to the set of meaningful stimulation of some RNN *if and only if* L is accepted by some deterministic Muller automaton.

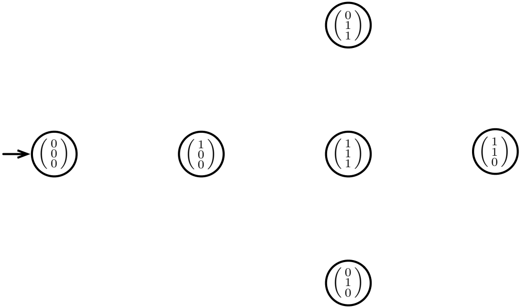
Therefore the classification of RNN by $\leq_W$ coincide with the classification of Muller automata by $\leq_W$, namely a decidable pre-well ordering of width 2 and height ω^ω.

From RNN to deterministic Muller automata. . .

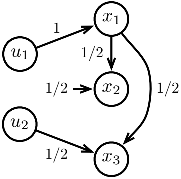


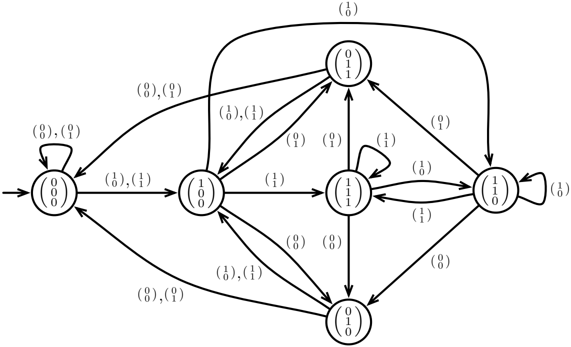
From RNN to deterministic Muller automata…



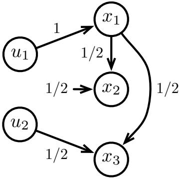


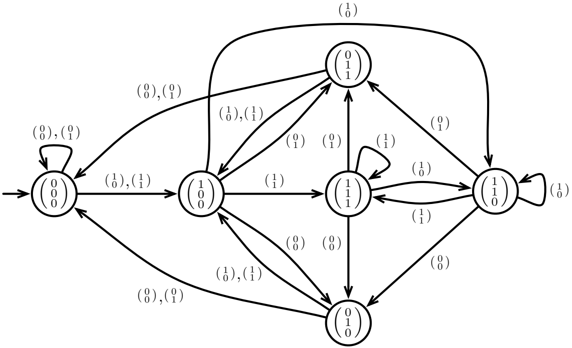
From RNN to deterministic Muller automata…





From RNN to deterministic Muller automata...





Attractors of the network

$A_1 = \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$ meaningful attractor

$A_2 = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$ spurious attractor

...

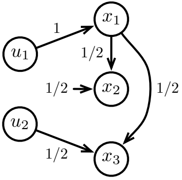
Cycles in the automaton

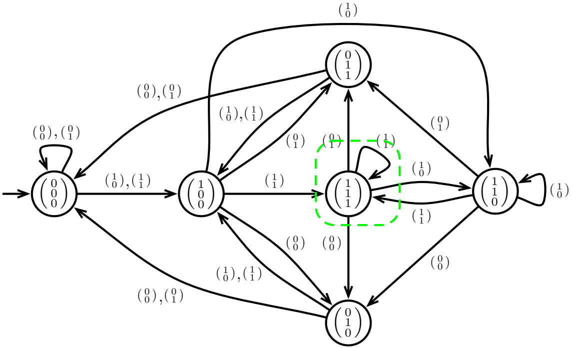
$C_1 = \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$ accepting cycle

$C_2 = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$ rejecting cycle

...

From RNN to deterministic Muller automata. . .





Attractors of the network

$A_1 = \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$ meaningful attractor

$A_2 = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$ spurious attractor

. . .

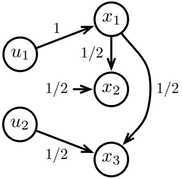
Cycles in the automaton

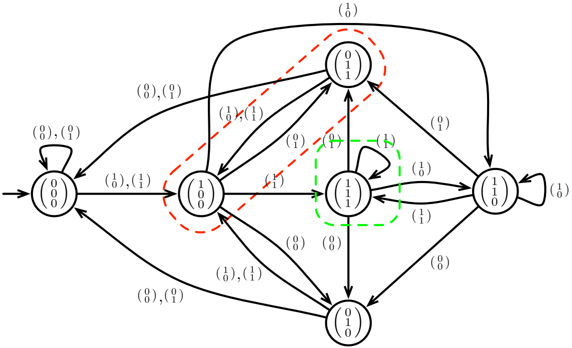
$C_1 = \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$ accepting cycle

$C_2 = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$ rejecting cycle

. . .

From RNN to deterministic Muller automata. . .





Attractors of the network

$A_1 = \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$ meaningful attractor

$A_2 = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$ spurious attractor

. . .

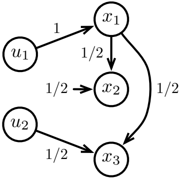
Cycles in the automaton

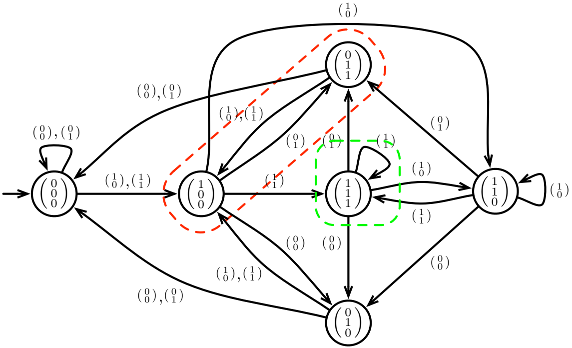
$C_1 = \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}$ accepting cycle

$C_2 = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$ rejecting cycle

. . .

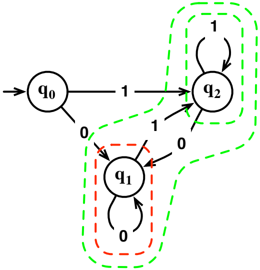
From RNN to deterministic Muller automata…



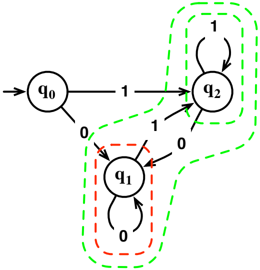


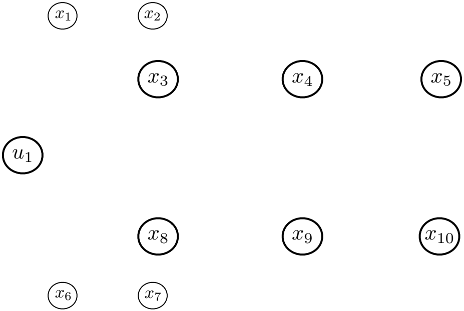
By construction, $\mathit{MeanStim}(\mathcal{N}) = L(\mathcal{A})$, thus $\mathit{MeanStim}(\mathcal{N})$ is ω-rational.

From deterministic Muller automata to RNN. . .

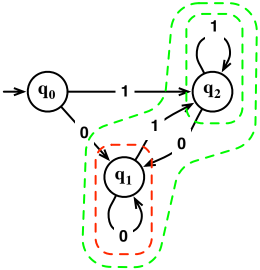


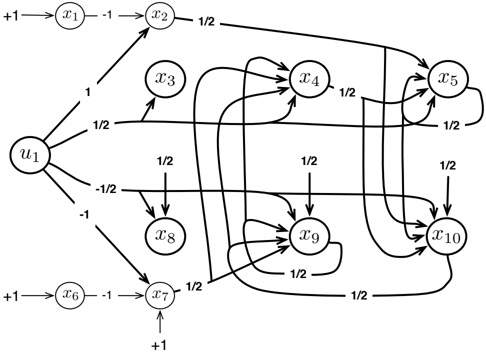
From deterministic Muller automata to RNN. . .



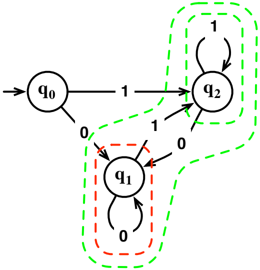


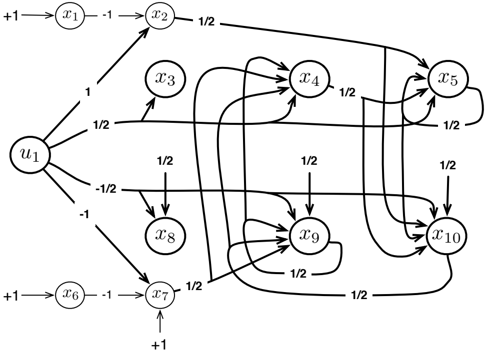
From deterministic Muller automata to RNN. . .





From deterministic Muller automata to RNN…





Cycles in the automaton

$C_1 = \{q_2\}$ accepting cycle

$C_2 = \{q_1\}$ rejecting cycle

$C_3 = \{q_1, q_2\}$ accepting cycle

Attractors of the network

$A_1 = \{\mathbf{1}_{1,5,6}\}$ meaningful attractor

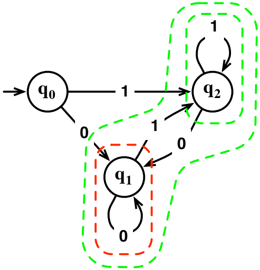
$A_2 = \{\mathbf{1}_{1,6,9}\}$ spurious attractor

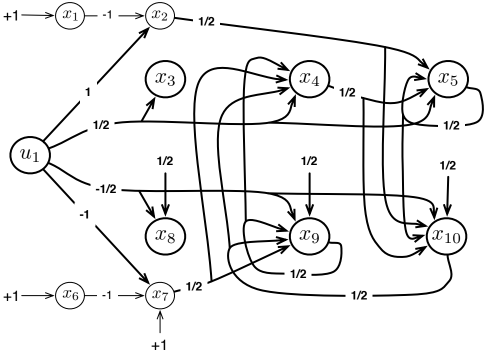
$A_3 = \{\mathbf{1}_{1,6,10}, \mathbf{1}_{1,4,6}\}$, $A_4 = \{\mathbf{1}_{1,5,6}, \mathbf{1}_{1,6,10}, \mathbf{1}_{1,4,6}\}$,

$A_5 = \{\mathbf{1}_{1,6,10}, \mathbf{1}_{1,4,6}, \mathbf{1}_{1,6,9}\}$, $A_6 = \{\mathbf{1}_{1,5,6}, \mathbf{1}_{1,6,10}, \mathbf{1}_{1,4,6}, \mathbf{1}_{1,6,9}\}$

meaningful attractors

From deterministic Muller automata to RNN...

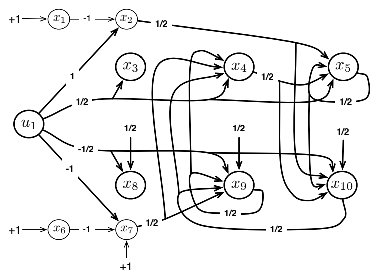




By construction, $L(\mathcal{A}) = MeanStim(\mathcal{N})$.

Example

Consider the following RNN $\mathcal{N}$:



$A_1 = \{\mathbf{1}_{1,5,6}\}$ meaningful attractor

$A_2 = \{\mathbf{1}_{1,6,9}\}$ spurious attractor

$A_3 = \{\mathbf{1}_{1,6,10}, \mathbf{1}_{1,4,6}\}$ meaningful attractor

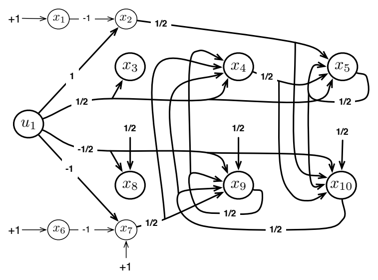
$A_4 = \{\mathbf{1}_{1,5,6}, \mathbf{1}_{1,6,10}, \mathbf{1}_{1,4,6}\}$ meaningful attractor

$A_5 = \{\mathbf{1}_{1,6,10}, \mathbf{1}_{1,4,6}, \mathbf{1}_{1,6,9}\}$ meaningful attractor

$A_6 = \{\mathbf{1}_{1,5,6}, \mathbf{1}_{1,6,10}, \mathbf{1}_{1,4,6}, \mathbf{1}_{1,6,9}\}$ meaningful attractors

Example

Consider the following RNN $\mathcal{N}$:



Then the degree of $\mathcal{N}$ in the RNN hierarchy is ω.

Conclusion

- We presented a decidable transfinite classification of simple neural nets based on their computational capability.
- The height of a network in the RNN hierarchy is the new index of complexity that we propose.
- This classification is more refined than classifications based on the number of layers or cells of the network
- Future work: search for more biologically oriented classification; investigate the computational capabilities of more biologically plausible neural nets.

Conclusion

- We presented a decidable transfinite classification of simple neural nets based on their computational capability.
- The height of a network in the RNN hierarchy is the new index of complexity that we propose.
- This classification is more refined than classifications based on the number of layers or cells of the network
- Future work: search for more biologically oriented classification; investigate the computational capabilities of more biologically plausible neural nets.

Conclusion

- We presented a decidable transfinite classification of simple neural nets based on their computational capability.
- The height of a network in the RNN hierarchy is the new index of complexity that we propose.
- This classification is more refined than classifications based on the number of layers or cells of the network
- Future work: search for more biologically oriented classification; investigate the computational capabilities of more biologically plausible neural nets.

Conclusion

- We presented a decidable transfinite classification of simple neural nets based on their computational capability.
- The height of a network in the RNN hierarchy is the new index of complexity that we propose.
- This classification is more refined than classifications based on the number of layers or cells of the network
- Future work: search for more biologically oriented classification; investigate the computational capabilities of more biologically plausible neural nets.