Attractor-Based Complexity of Boolean Recurrent Neural Networks

Application to a Simplified Model of the Basal Ganglia-Thalamocortical Network

Jérémie Cabessa
Joint work with Alessandro E.P. Villa

Department of Mathematical Economics
University Paris II
France

29 July 2016

Introduction

- We introduce an attractor-based complexity measure for Boolean recurrent neural networks.
- The measure reflects the ability of the networks to discriminate between their input streams via the manifestation of attractor dynamics.
- We provide an application of this complexity measure to a simplified Boolean model of the basal ganglia-thalamocortical network.

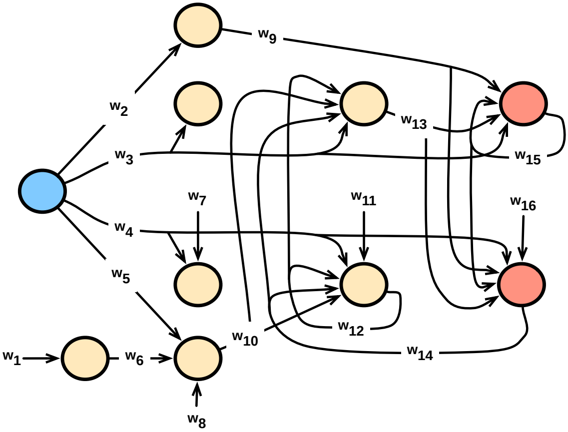
Introduction

- We introduce an attractor-based complexity measure for Boolean recurrent neural networks.
- The measure reflects the ability of the networks to discriminate between their input streams via the manifestation of attractor dynamics.
- We provide an application of this complexity measure to a simplified Boolean model of the basal ganglia-thalamocortical network.

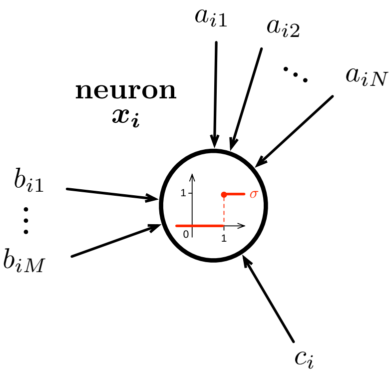
Introduction

- We introduce an attractor-based complexity measure for Boolean recurrent neural networks.
- The measure reflects the ability of the networks to discriminate between their input streams via the manifestation of attractor dynamics.
- We provide an application of this complexity measure to a simplified Boolean model of the basal ganglia-thalamocortical network.

Recurrent Neural Network



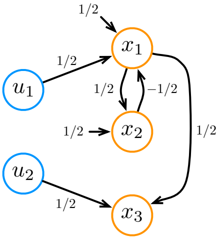
Boolean Recurrent Neural Network



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

From Boolean Neural Networks to Automata

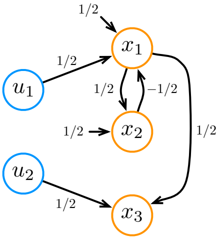
Boolean Neural Network



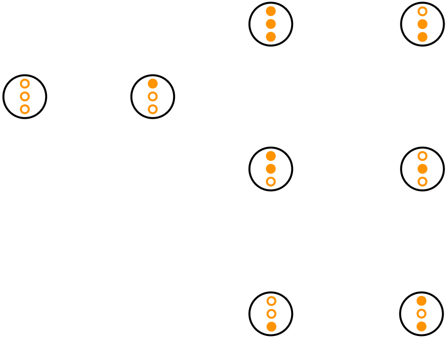
Automaton

From Boolean Neural Networks to Automata

Boolean Neural Network



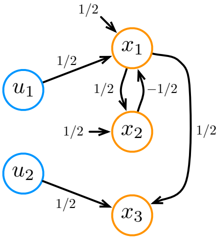
Automaton

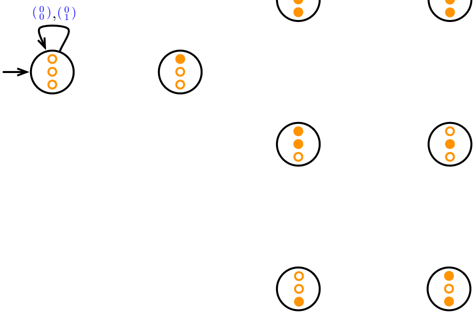


From Boolean Neural Networks to Automata

Boolean Neural Network

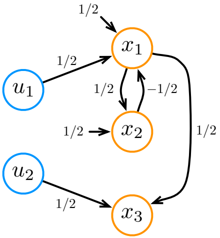
Automaton



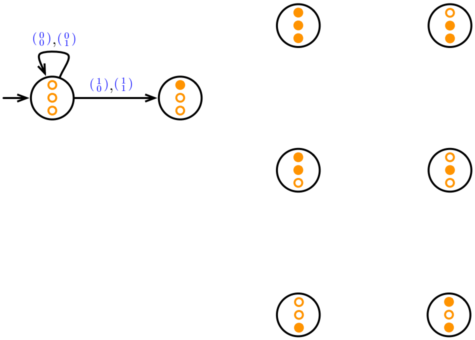


From Boolean Neural Networks to Automata

Boolean Neural Network

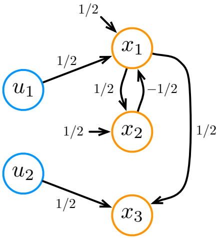


Automaton

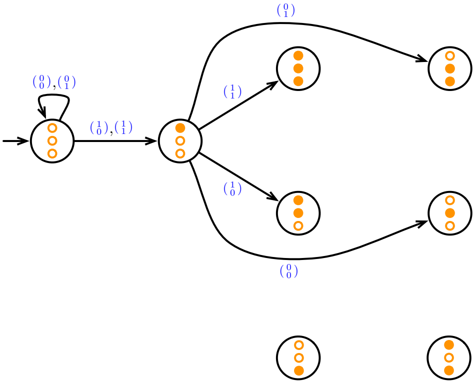


From Boolean Neural Networks to Automata

Boolean Neural Network

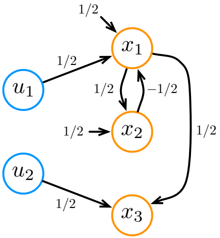


Automaton

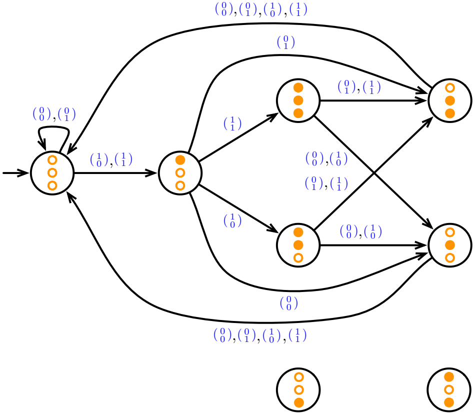


From Boolean Neural Networks to Automata

Boolean Neural Network



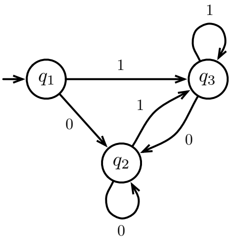
Automaton



From Automata to Boolean Neural Networks

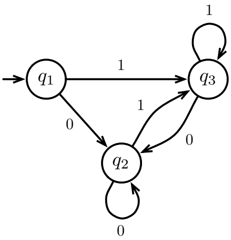
Automaton

Boolean Neural Network

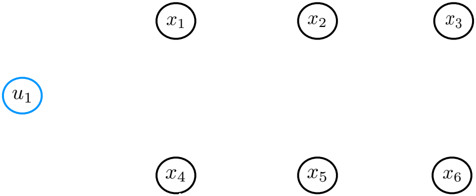


From Automata to Boolean Neural Networks

Automaton

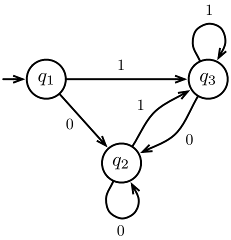


Boolean Neural Network

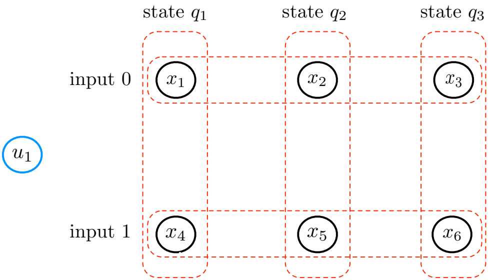


From Automata to Boolean Neural Networks

Automaton

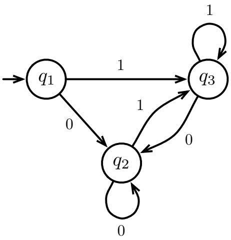


Boolean Neural Network

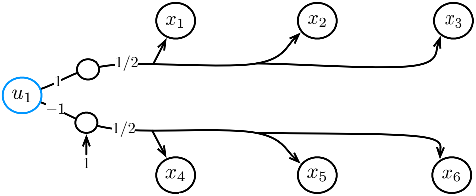


From Automata to Boolean Neural Networks

Automaton

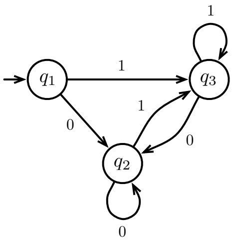


Boolean Neural Network

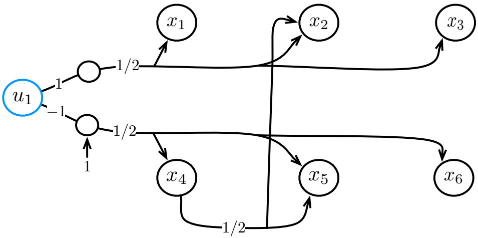


From Automata to Boolean Neural Networks

Automaton

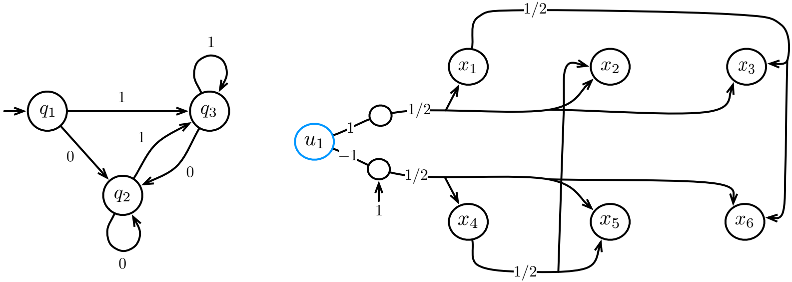


Boolean Neural Network



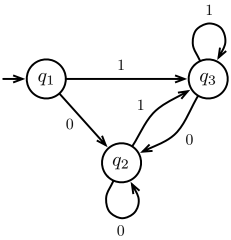
From Automata to Boolean Neural Networks

Automaton **Boolean Neural Network**

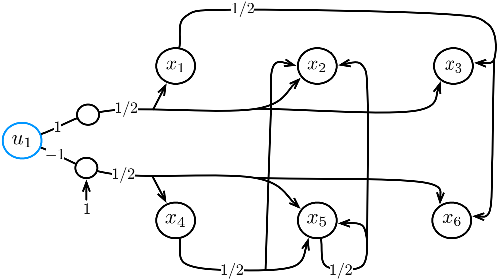


From Automata to Boolean Neural Networks

Automaton

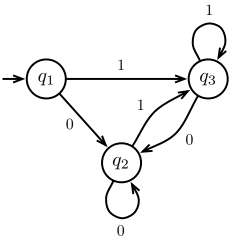


Boolean Neural Network

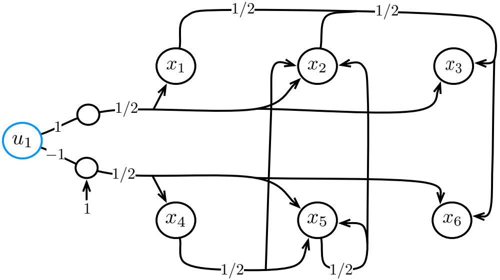


From Automata to Boolean Neural Networks

Automaton

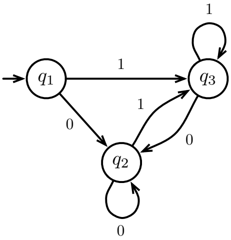


Boolean Neural Network

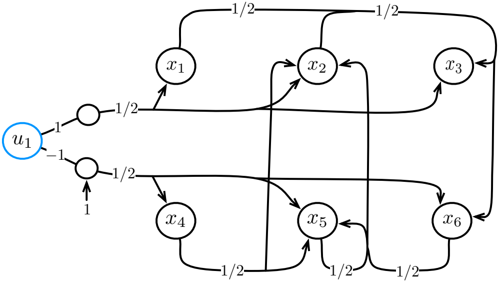


From Automata to Boolean Neural Networks

Automaton

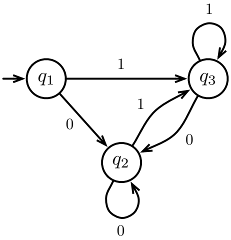


Boolean Neural Network

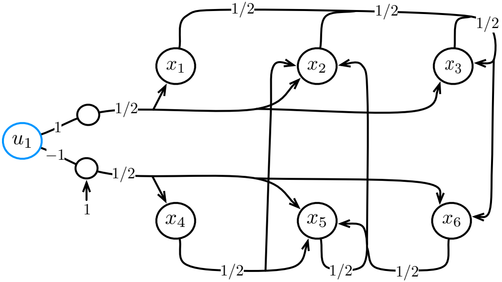


From Automata to Boolean Neural Networks

Automaton

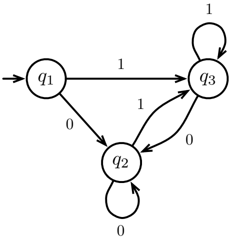


Boolean Neural Network

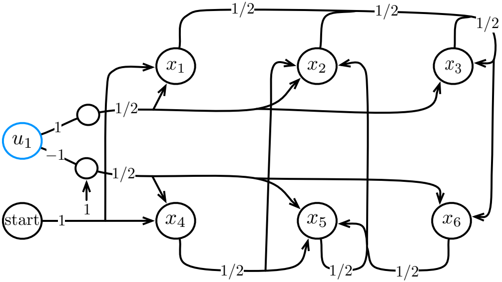


From Automata to Boolean Neural Networks

Automaton



Boolean Neural Network



Equivalence between Boolean Neural Networks and Automata

Theorem (Minsky 67)

"It is evident that each neural network of the kind we have been considering is a finite-state machine."

"[...] It is interesting and even surprising that there is a converse to this. Every finite-state machine is equivalent to, and can be "simulated" by, some neural net."

Equivalence between Boolean Neural Networks and Automata

Theorem (Minsky 67)

"It is evident that each neural network of the kind we have been considering is a finite-state machine."

"[...] It is interesting and even surprising that there is a converse to this. Every finite-state machine is equivalent to, and can be "simulated" by, some neural net."

Equivalence between Boolean Neural Networks and Automata

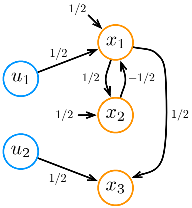
Theorem (Minsky 67)

"It is evident that each neural network of the kind we have been considering is a finite-state machine."

"[...] It is interesting and even surprising that there is a converse to this. Every finite-state machine is equivalent to, and can be "simulated" by, some neural net."

Attractors and Cycles

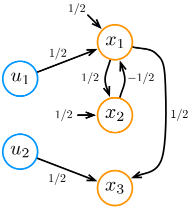
Boolean Neural Network



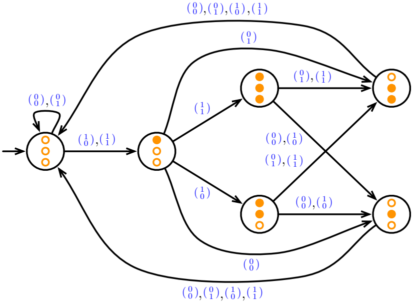
Automaton

Attractors and Cycles

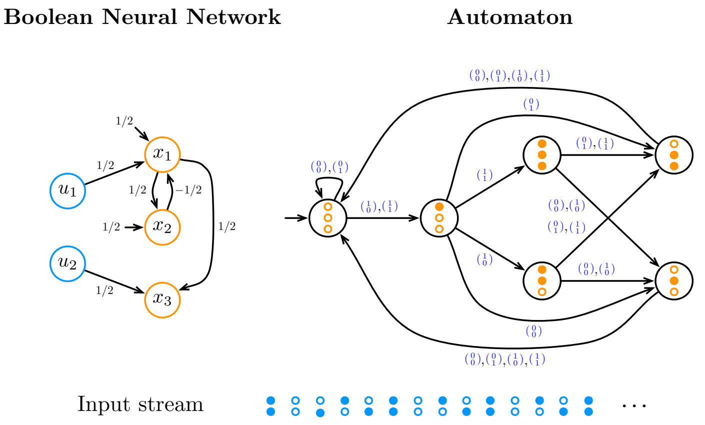
Boolean Neural Network



Automaton

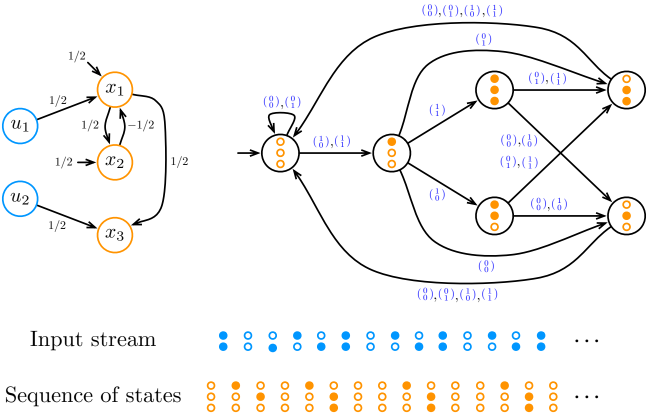


Attractors and Cycles



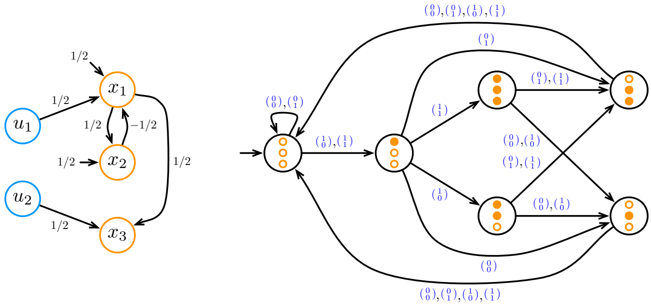
Attractors and Cycles

Boolean Neural Network **Automaton**



Attractors and Cycles

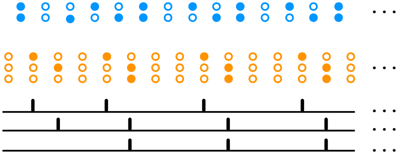
Boolean Neural Network **Automaton**



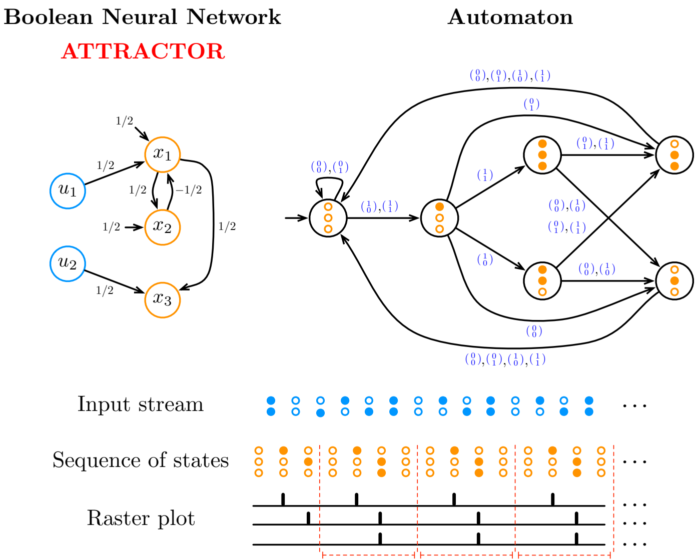
Input stream

Sequence of states

Raster plot

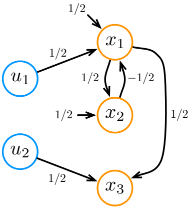


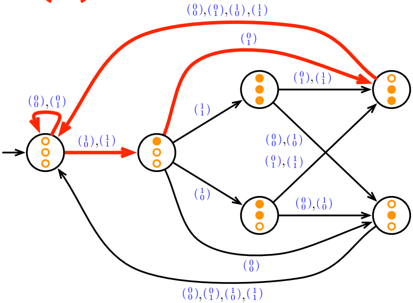
Attractors and Cycles



Attractors and Cycles

Boolean Neural Network — **ATTRACTOR** ⟺ **Automaton** — **CYCLE**

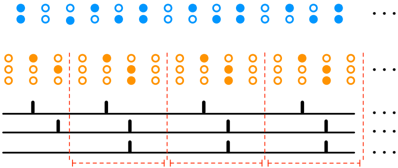




Input stream

Sequence of states

Raster plot



Attractor-Based Complexity of RNNs

- We assume that some aspect of the computational capabilities of recurrent neural networks are related to their attractor dynamics.
- We introduce attractor-based complexity measure inspired from automata theory.
- We assume that the attractors are classified into two categories: meaningful or spurious.
- The attractor-based complexity refers to the maximal number of alternations between meaningful and spurious attractors that are included one into the other.

Attractor-Based Complexity of RNNs

- We assume that some aspect of the computational capabilities of recurrent neural networks are related to their attractor dynamics.
- We introduce attractor-based complexity measure inspired from automata theory.
- We assume that the attractors are classified into two categories: meaningful or spurious.
- The attractor-based complexity refers to the maximal number of alternations between meaningful and spurious attractors that are included one into the other.

Attractor-Based Complexity of RNNs

- We assume that some aspect of the computational capabilities of recurrent neural networks are related to their attractor dynamics.
- We introduce attractor-based complexity measure inspired from automata theory.
- We assume that the attractors are classified into two categories: meaningful or spurious.
- The attractor-based complexity refers to the maximal number of alternations between meaningful and spurious attractors that are included one into the other.

Attractor-Based Complexity of RNNs

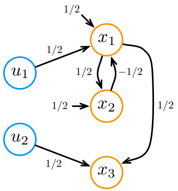
- We assume that some aspect of the computational capabilities of recurrent neural networks are related to their attractor dynamics.
- We introduce attractor-based complexity measure inspired from automata theory.
- We assume that the attractors are classified into two categories: meaningful or spurious.
- The attractor-based complexity refers to the maximal number of alternations between meaningful and spurious attractors that are included one into the other.

Attractor-Based Complexity of BRNNs

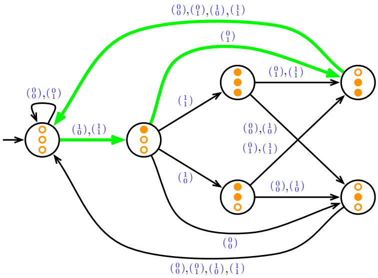
Example:

- Assume that $\mathcal{N}$ contains only one meaningful attractor; all others being spurious.
- Then, the attractor-based complexity of $\mathcal{N}$ is 2. Maximal "growing" sequence of 2 alternations between spurious and meaningful attractors.

Boolean Neural Network



Automaton

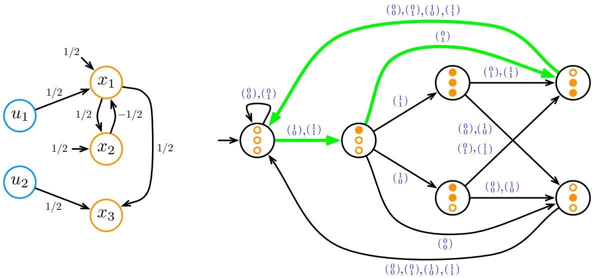


Attractor-Based Complexity of BRNNs

Example:

- Assume that $\mathcal{N}$ contains only one meaningful attractor; all others being spurious.
- Then, the attractor-based complexity of $\mathcal{N}$ is 2. Maximal "growing" sequence of 2 alternations between spurious and meaningful attractors.

Boolean Neural Network **Automaton**

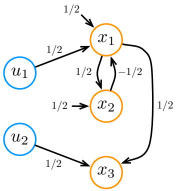


Attractor-Based Complexity of BRNNs

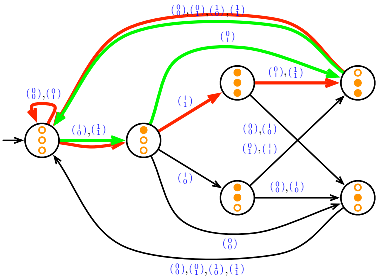
Example:

- Assume that $\mathcal{N}$ contains only one meaningful attractor; all others being spurious.
- Then, the attractor-based complexity of $\mathcal{N}$ is 2. Maximal "growing" sequence of 2 alternations between spurious and meaningful attractors.

Boolean Neural Network



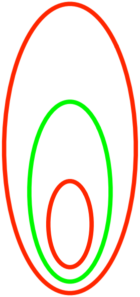
Automaton



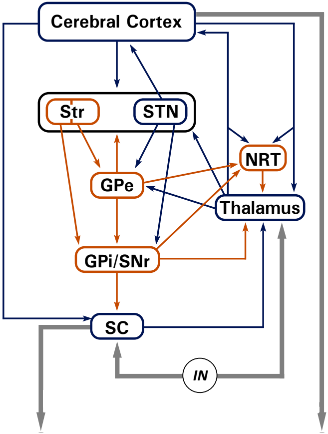
Attractor-Based Complexity of BRNNs

Example:

- Assume that $\mathcal{N}$ contains only one meaningful attractor; all others being spurious.
- Then, the attractor-based complexity of $\mathcal{N}$ is 2. Maximal "growing" sequence of 2 alternations between spurious and meaningful attractors.

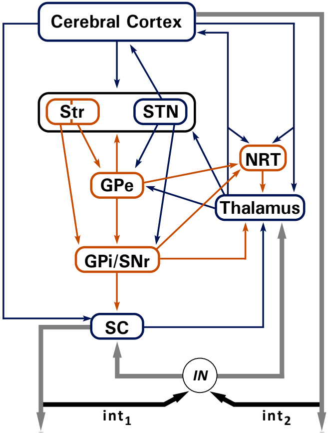


Boolean Model of the Basal Ganglia-Thalamocortical Network



IN	input node
SC	superior colliculus
GPi/SNr	output nuclei of the basal ganglia formed by the GABAergic projection neurons of the intermediate part of the pallidum and of the substantia nigra pars reticulata
Thalamus	thalamus
GPe	external part of the pallidum
NRT	thalamic reticular nucleus
Str-D1	striatopallidal component of the striatum
Str-D2	striatonigral component of the striatum
STN	subthalamic nucleus
Cerebral Cortex	cerebral cortex

Boolean Model of the Basal Ganglia-Thalamocortical Network



IN	input node
SC	superior colliculus
GPi/SNr	output nuclei of the basal ganglia formed by the GABAergic projection neurons of the intermediate part of the pallidum and of the substantia nigra pars reticulata
Thalamus	thalamus
GPe	external part of the pallidum
NRT	thalamic reticular nucleus
Str-D1	striatopallidal component of the striatum
Str-D2	striatonigral component of the striatum
STN	subthalamic nucleus
Cerebral Cortex	cerebral cortex

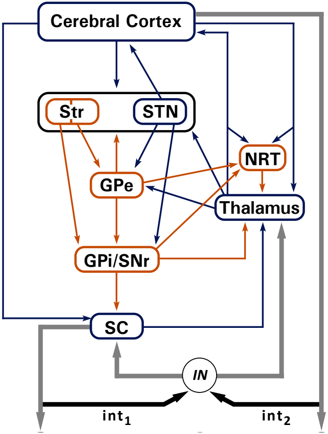
Boolean Model of the Basal Ganglia-Thalamocortical Network

Source	Target (Node #)									
Node # (Name)	0	1	2	3	4	5	6	7	8	9
0 (IN)	·	1	1	·	·	·	·	·	·	·
1 (SC)	int_1	·	1	·	·	·	·	·	·	·
2 (Thalamus)	·	·	·	1	·	1	1	1	1	1
3 (RTN)	·	·	-1	·	·	·	·	·	·	·
4 (GPi/SNr)	·	-1	-1	-1	·	·	·	·	·	·
5 (STN)	·	·	·	·	2	·	2	·	·	2
6 (GPe)	·	·	·	-1/2	-1/2	-1/2	·	-1/2	-1/2	·
7 (Str-D2)	·	·	·	·	·	·	-1	·	·	·
8 (Str-D1)	·	·	·	·	-1/2	·	-1/2	·	·	·
9 (CCortex)	int_2	1/2	1/2	1/2	·	1/2	·	1/2	1/2	·

Table: Adjancency matrix

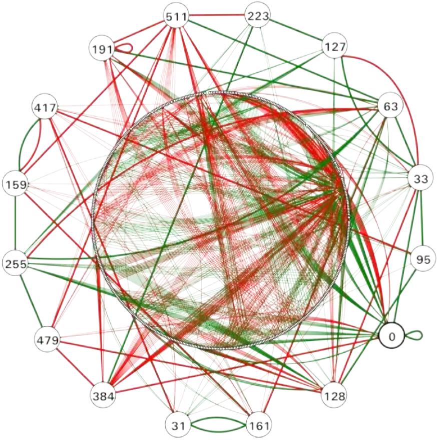
Corresponding Automaton

- 9 activation nodes and 1 input node in the network $\Rightarrow$ 512 states and a binary alphabet for the automaton



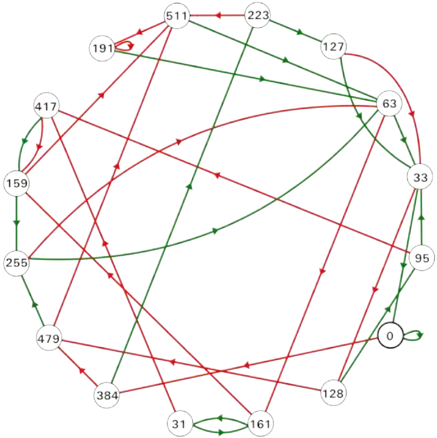
Corresponding Automaton

- 9 activation nodes and 1 input node in the network $\Rightarrow$ 512 states and a binary alphabet for the automaton



Corresponding Automaton

- 9 activation nodes and 1 input node in the network $\Rightarrow$ 512 states and a binary alphabet for the automaton



Corresponding Automaton

Meaningfulness criterion

- A constitutive cycle – i.e., a basic attractor – is *spurious* if it is characterised either by active SC and quiet Thalamus at the same time step, or by a quiet GPi/SNr during the majority of the duration of the cycle.
- A constitutive cycle is *meaningful* otherwise.
- A non-constitutive cycle – i.e., a composed attractor – is considered *meaningful* if it contains a majority of meaningful constitutive cycles.
- A non-constitutive cycle is *spurious* otherwise.

Corresponding Automaton

Meaningfulness criterion

- A constitutive cycle – i.e., a basic attractor – is *spurious* if it is characterised either by active SC and quiet Thalamus at the same time step, or by a quiet GPi/SNr during the majority of the duration of the cycle.
- A constitutive cycle is *meaningful* otherwise.
- A non-constitutive cycle – i.e., a composed attractor – is considered *meaningful* if it contains a majority of meaningful constitutive cycles.
- A non-constitutive cycle is *spurious* otherwise.

Corresponding Automaton

Meaningfulness criterion

- A constitutive cycle – i.e., a basic attractor – is *spurious* if it is characterised either by active SC and quiet Thalamus at the same time step, or by a quiet GPi/SNr during the majority of the duration of the cycle.
- A constitutive cycle is *meaningful* otherwise.
- A non-constitutive cycle – i.e., a composed attractor – is considered *meaningful* if it contains a majority of meaningful constitutive cycles.
- A non-constitutive cycle is *spurious* otherwise.

Corresponding Automaton

Meaningfulness criterion

- A constitutive cycle – i.e., a basic attractor – is *spurious* if it is characterised either by active SC and quiet Thalamus at the same time step, or by a quiet GPi/SNr during the majority of the duration of the cycle.
- A constitutive cycle is *meaningful* otherwise.
- A non-constitutive cycle – i.e., a composed attractor – is considered *meaningful* if it contains a majority of meaningful constitutive cycles.
- A non-constitutive cycle is *spurious* otherwise.

Attractor-Based Complexity

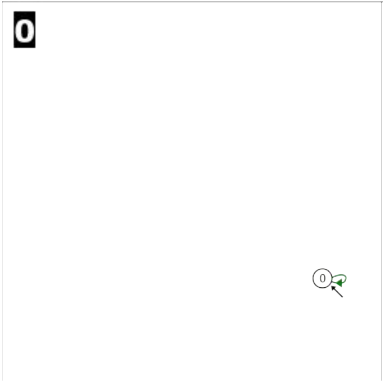
In the corresponding Muller automaton, we found a maximal "alternating tree of cycles" of length ω^6

- $\Rightarrow$ **Complexity of the Boolean model is 6.**

Attractor-Based Complexity

In the corresponding Muller automaton, we found a maximal "alternating tree of cycles" of length ω^6

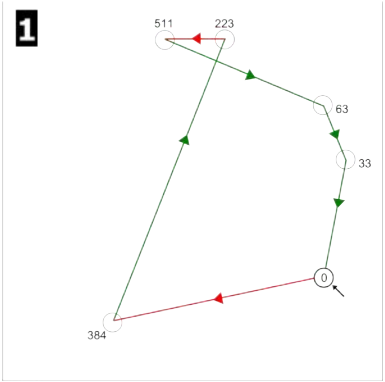
- $\Rightarrow$ **Complexity of the Boolean model is 6.**



Attractor-Based Complexity

In the corresponding Muller automaton, we found a maximal "alternating tree of cycles" of length ω^6

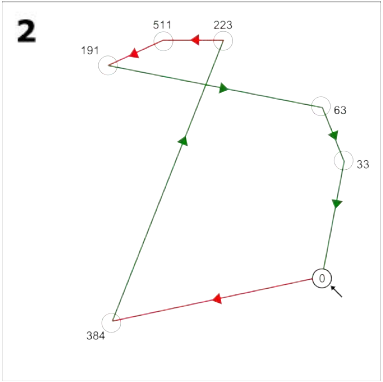
- $\Rightarrow$ **Complexity of the Boolean model is 6.**



Attractor-Based Complexity

In the corresponding Muller automaton, we found a maximal "alternating tree of cycles" of length ω^6

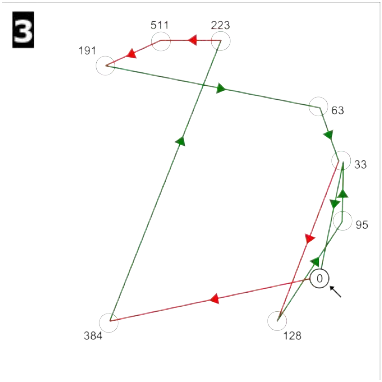
- $\Rightarrow$ **Complexity of the Boolean model is 6.**



Attractor-Based Complexity

In the corresponding Muller automaton, we found a maximal "alternating tree of cycles" of length ω^6

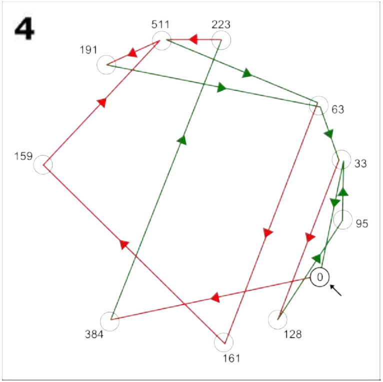
- $\Rightarrow$ **Complexity of the Boolean model is 6.**



Attractor-Based Complexity

In the corresponding Muller automaton, we found a maximal "alternating tree of cycles" of length ω^6

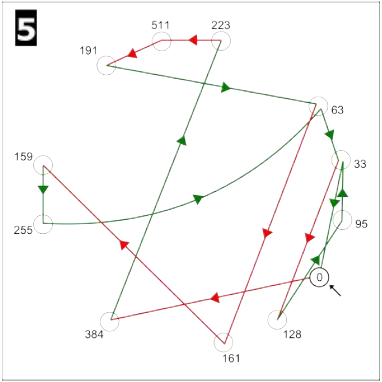
- $\Rightarrow$ **Complexity of the Boolean model is 6.**



Attractor-Based Complexity

In the corresponding Muller automaton, we found a maximal "alternating tree of cycles" of length ω^6

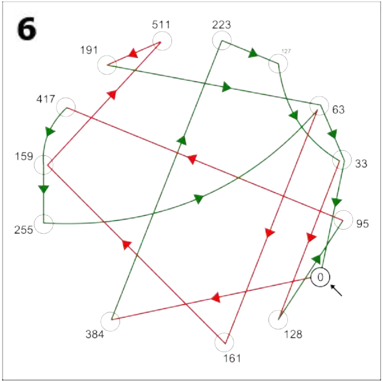
- $\Rightarrow$ **Complexity of the Boolean model is 6.**



Attractor-Based Complexity

In the corresponding Muller automaton, we found a maximal "alternating tree of cycles" of length ω^6

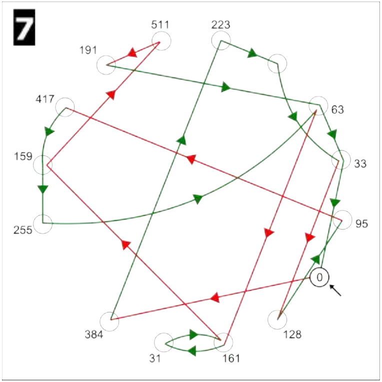
- $\Rightarrow$ **Complexity of the Boolean model is 6.**



Attractor-Based Complexity

In the corresponding Muller automaton, we found a maximal "alternating tree of cycles" of length ω^6

- $\Rightarrow$ **Complexity of the Boolean model is 6.**

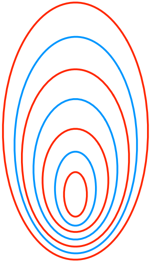


Attractor-Based Complexity

In the corresponding Muller automaton, we found a maximal "alternating tree of cycles" of length ω^6

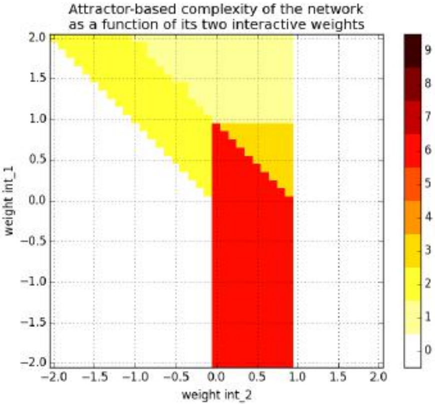
- $\Rightarrow$ **Complexity of the Boolean model is 6.**

Wagner degree



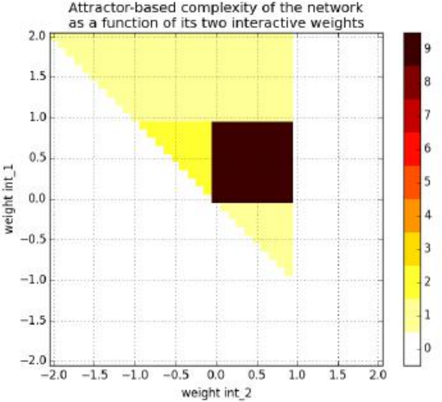
$[-]\ \omega^6$

Attractor-Based Complexity



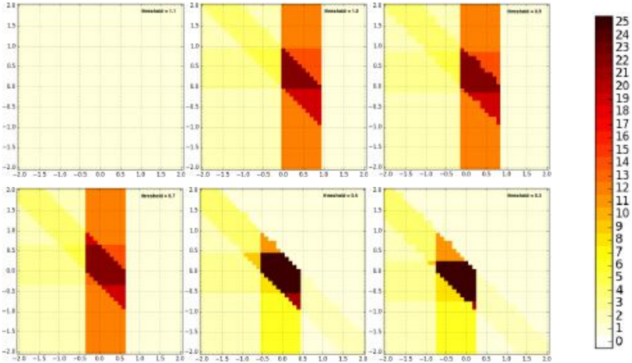
The *interactive (or feedback) connections* play a significant role in the maintenance and robustness of an optimal level of complexity.

Attractor-Based Complexity



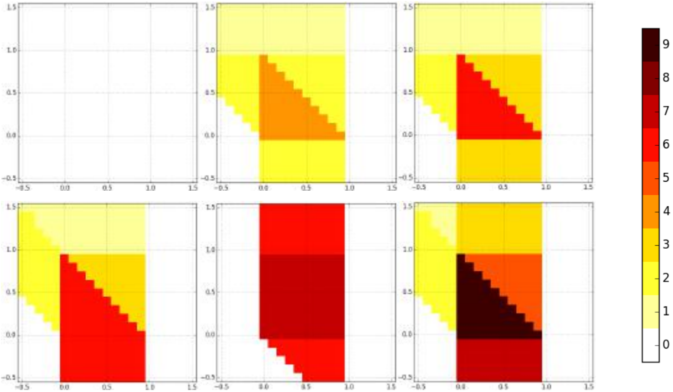
By slightly varying the weights of the networks by ± 0.2, one could increase the optimal complexity from 6 to 9.

Number of Basic Attractors: global threshold (or weights) modifications



Lowering the global threshold, i.e., potentiating the global synaptic level, increases the maximal numbers of attractors and improves the robustness of the "no interactivity" configuration.

Attractor-Based Complexity: local weights modifications



Even single weight variations of ± 0.1 can significantly affect the complexity pattern.

Conclusions

- The *number of attractors* and the *attractor-based complexity* might be relevant measures of the computational capabilities for Boolean recurrent neural networks.
- Global and local modifications of the it synaptic weights significantly affect the attractor complexity of the networks.
- The values of the *interactive connections* also play a significant role in the maintenance and robustness of an optimal level of attractor-based complexity.
- These considerations support the rationale that *synaptic plasticity* might be crucially involved in the computational capabilities of neural networks.

Conclusions

- The *number of attractors* and the *attractor-based complexity* might be relevant measures of the computational capabilities for Boolean recurrent neural networks.
- Global and local modifications of the it synaptic weights significantly affect the attractor complexity of the networks.
- The values of the *interactive connections* also play a significant role in the maintenance and robustness of an optimal level of attractor-based complexity.
- These considerations support the rationale that *synaptic plasticity* might be crucially involved in the computational capabilities of neural networks.

Conclusions

- The *number of attractors* and the *attractor-based complexity* might be relevant measures of the computational capabilities for Boolean recurrent neural networks.
- Global and local modifications of the it synaptic weights significantly affect the attractor complexity of the networks.
- The values of the *interactive connections* also play a significant role in the maintenance and robustness of an optimal level of attractor-based complexity.
- These considerations support the rationale that *synaptic plasticity* might be crucially involved in the computational capabilities of neural networks.

Conclusions

- The *number of attractors* and the *attractor-based complexity* might be relevant measures of the computational capabilities for Boolean recurrent neural networks.
- Global and local modifications of the it synaptic weights significantly affect the attractor complexity of the networks.
- The values of the *interactive connections* also play a significant role in the maintenance and robustness of an optimal level of attractor-based complexity.
- These considerations support the rationale that *synaptic plasticity* might be crucially involved in the computational capabilities of neural networks.