Neural coding: a perspective for new associative memories

Vincent Gripon

Télécom Bretagne

February 26, 2012

Plan

Challenge

Context

Exponential growth of the amount of information

Challenge

User	Difficult to find the desired piece of information.
Engineer	Algorithms complexity is limiting, Ad-hoc solutions are often required.

Challenge

Context

Exponential growth of the amount of information

Challenge

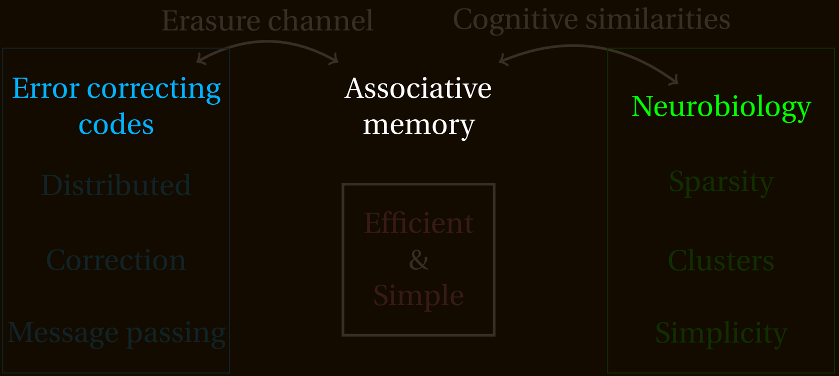
User	Difficult to find the desired piece of information.
Engineer	Algorithms complexity is limiting, Ad-hoc solutions are often required.

Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea



Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea

Erasure channel

Cognitive similarities

Error correcting codes

Associative memory

Neurobiology

Distributed

Correction

Message passing

Efficient & Simple

Sparsity

Clusters

Simplicity

Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea

Erasure channel

Cognitive similarities

Error correcting codes

Associative memory

Neurobiology

Distributed

Correction

Message passing

Efficient & Simple

Sparsity

Clusters

Simplicity

Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea

Erasure channel

Cognitive similarities

Error correcting codes

Distributed

Correction

Message passing

Associative memory

Efficient & Simple

Neurobiology

Sparsity

Clusters

Simplicity

Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea

Erasure channel

Cognitive similarities

Error correcting codes

Distributed

Correction

Message passing

Associative memory

Efficient & Simple

Neurobiology

Sparsity

Clusters

Simplicity

Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea

Erasure channel

Cognitive similarities

Error correcting codes

Distributed

Correction

Message passing

Associative memory

Efficient & Simple

Neurobiology

Sparsity

Clusters

Simplicity

Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea

Erasure channel

Cognitive similarities

Error correcting codes

Distributed

Correction

Message passing

Associative memory

Efficient & Simple

Neurobiology

Sparsity

Clusters

Simplicity

Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea

Erasure channel

Cognitive similarities

Error correcting codes

Distributed

Correction

Message passing

Associative memory

Efficient & Simple

Neurobiology

Sparsity

Clusters

Simplicity

Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea

Erasure channel

Cognitive similarities

Error correcting codes

Distributed

Correction

Message passing

Associative memory

Efficient & Simple

Neurobiology

Sparsity

Clusters

Simplicity

Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea

Erasure channel

Cognitive similarities

Error correcting codes

Distributed

Correction

Message passing

Associative memory

Efficient & Simple

Neurobiology

Sparsity

Clusters

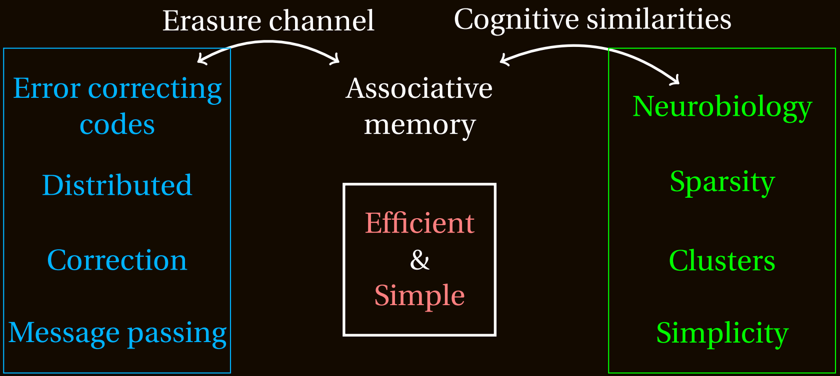
Simplicity

Approach

Definition

Associative memory = device that can retrieve previously learned messages from part of them.

Idea



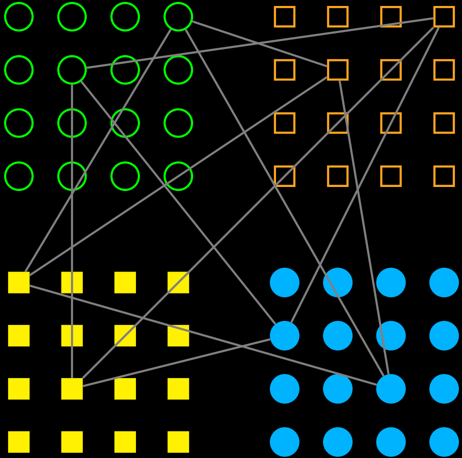
Outline

Plan

1. Introduction
   - Challenge
   - Approach
2. Our model
   - Principle
   - Performance
   - Application example
3. Direction: learning sequences
4. Openings

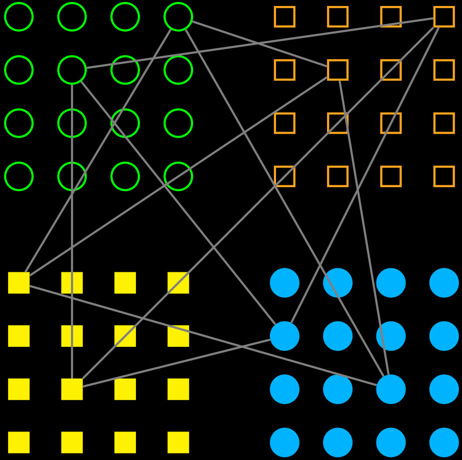
Our model: learning

- Example: $c = 4$ clusters made of $l = 16$ neurons each,
- Learn 16-ary message: $\underbrace{8}_{j_1 \text{ in } c_1}\underbrace{3}_{j_2 \text{ in } c_2}\underbrace{2}_{j_3 \text{ in } c_3}\underbrace{9}_{j_4 \text{ in } c_4}$,

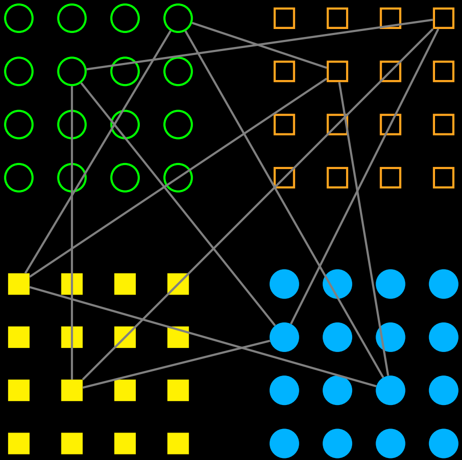


Our model: learning

- Example: $c = 4$ clusters made of $l = 16$ neurons each,
- Learn 16-ary message: $\underbrace{8}_{j_1 \text{ in } c_1} \underbrace{3}_{j_2 \text{ in } c_2} \underbrace{2}_{j_3 \text{ in } c_3} \underbrace{9}_{j_4 \text{ in } c_4}$,

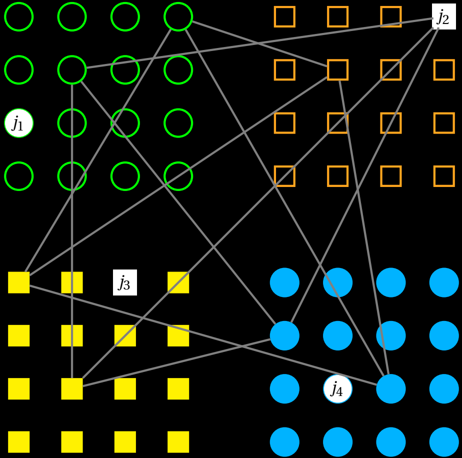


- Example: $c = 4$ clusters made of $l = 16$ neurons each,
- Learn 16-ary message: $\underbrace{8}_{j_1 \text{ in } c_1} \underbrace{3}_{j_2 \text{ in } c_2} \underbrace{2}_{j_3 \text{ in } c_3} \underbrace{9}_{j_4 \text{ in } c_4}$,



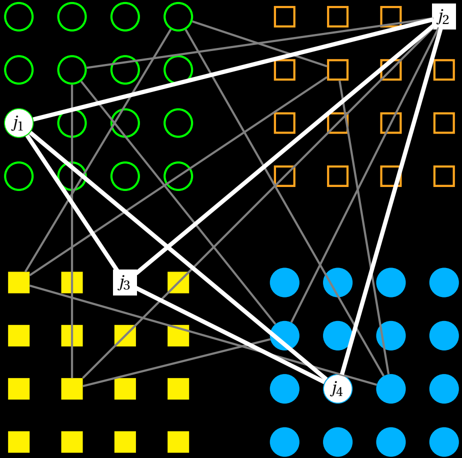
Our model: learning

- Example: $c = 4$ clusters made of $l = 16$ neurons each,
- Learn 16-ary message: $\underbrace{8}_{j_1 \text{ in } c_1}\ \underbrace{3}_{j_2 \text{ in } c_2}\ \underbrace{2}_{j_3 \text{ in } c_3}\ \underbrace{9}_{j_4 \text{ in } c_4}$,



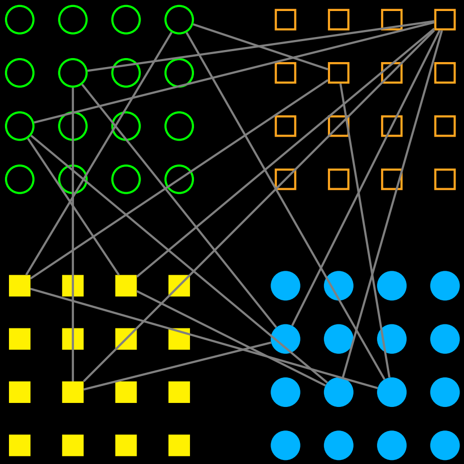
Our model: learning

- Example: $c = 4$ clusters made of $l = 16$ neurons each,
- Learn 16-ary message: $\underbrace{8}_{j_1 \text{ in } c_1} \underbrace{3}_{j_2 \text{ in } c_2} \underbrace{2}_{j_3 \text{ in } c_3} \underbrace{9}_{j_4 \text{ in } c_4}$,



Our model: retrieving

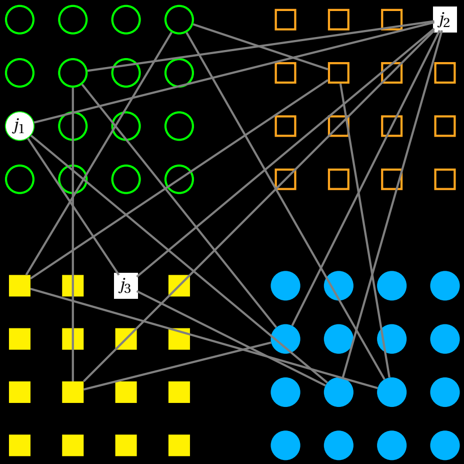
$\underbrace{8}_{j_1 \text{ in } c_1}$ $\underbrace{3}_{j_2 \text{ in } c_2}$ $\underbrace{2}_{j_3 \text{ in } c_3}$?



- Projection to the network,
- Global decoding: sum,
- Local decoding: winner-take-all,
- Possibly iterate the two decodings.

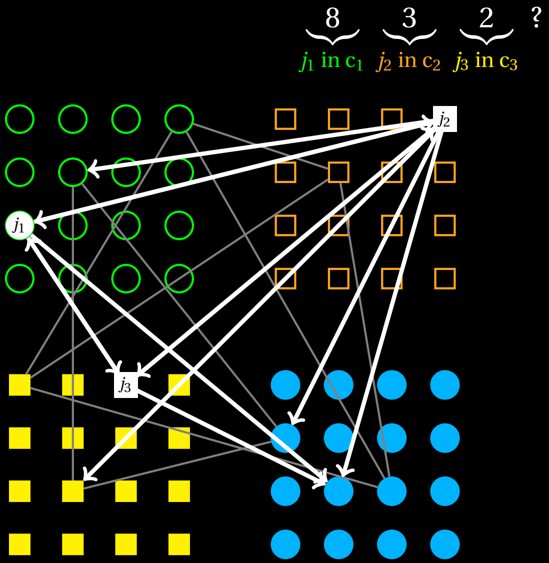
Our model: retrieving

$$\underbrace{8}_{j_1 \text{ in } c_1}\ \underbrace{3}_{j_2 \text{ in } c_2}\ \underbrace{2}_{j_3 \text{ in } c_3}\ ?$$



- Projection to the network,
- Global decoding: sum,
- Local decoding: winner-take-all,
- Possibly iterate the two decodings.

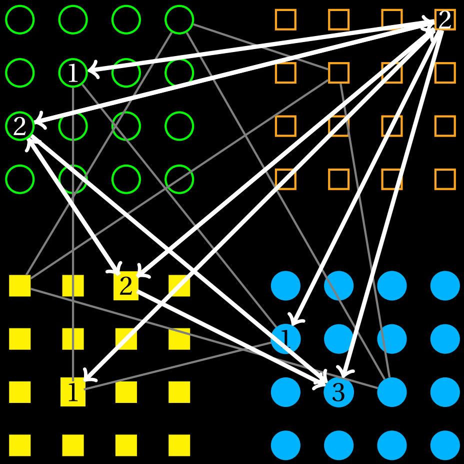
Our model: retrieving



- Projection to the network,
- Global decoding: sum,
- Local decoding: winner-take-all,
- Possibly iterate the two decodings.

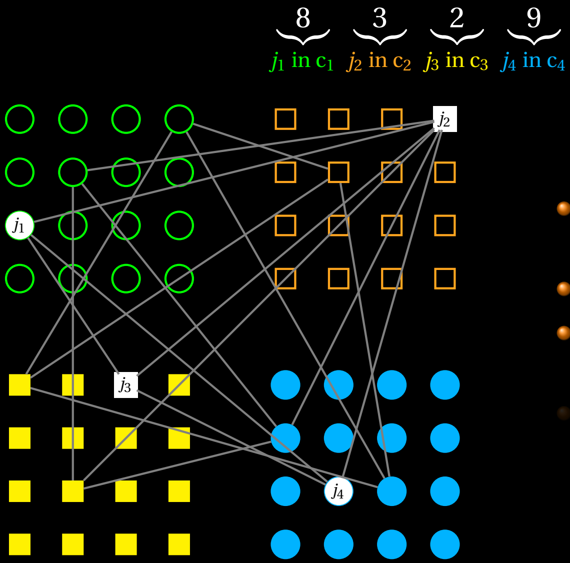
Our model: retrieving

$$\underbrace{8}_{j_1 \text{ in } c_1} \; \underbrace{3}_{j_2 \text{ in } c_2} \; \underbrace{2}_{j_3 \text{ in } c_3} \; ?$$



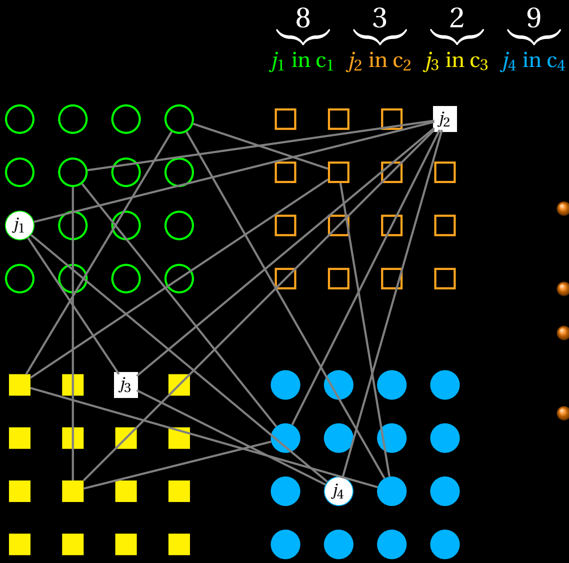
- Projection to the network,
- Global decoding: sum,
- Local decoding: winner-take-all,
- Possibly iterate the two decodings.

Our model: retrieving



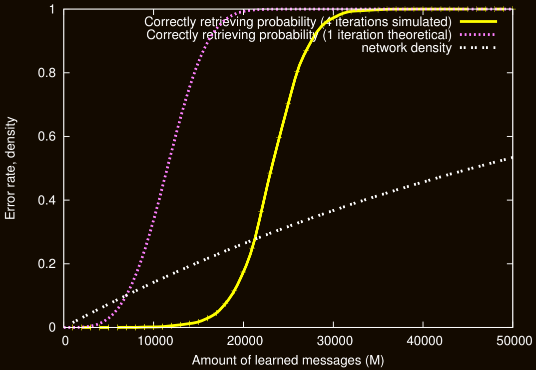
- Projection to the network,
- Global decoding: sum,
- Local decoding: winner-take-all,
- Possibly iterate the two decodings.

Our model: retrieving



- Projection to the network,
- Global decoding: sum,
- Local decoding: winner-take-all,
- Possibly iterate the two decodings.

Associative memory



$c = 8$, $l = 256$ neurons each. Input messages have just 4 known symbols.

State-of-the-art Hopfield network

Our network

An example: cache design

Role

Speed-up the lookup of frequently accessed data.

Context

- With associativity decreases the miss rate,
- Current implementations are limited to 8-ways associativity,
- Speed and area limitations.

Our proposal

- Adaptation of the network:
  - Given the address, find the associated data,
- Additionnal error probability: 10^{-30},
- Area consumption reduced by 65%,
- Pipelined architecture using a few clock cycles.

An example: cache design

Role

Speed-up the lookup of frequently accessed data.

Context

- With associativity decreases the miss rate,
- Current implementations are limited to 8-ways associativity,
- Speed and area limitations.

Our proposal

- Adaptation of the network:
  - Given the address, find the associated data,
- Additionnal error probability: 10^{-30},
- Area consumption reduced by 65%,
- Pipelined architecture using a few clock cycles.

An example: cache design

Role

Speed-up the lookup of frequently accessed data.

Context

- With associativity decreases the miss rate,
- Current implementations are limited to 8-ways associativity,
- Speed and area limitations.

Our proposal

- Adaptation of the network:
  - Given the address, find the associated data,
- Additionnal error probability: 10^{-30},
- Area consumption reduced by 65%,
- Pipelined architecture using a few clock cycles.

An example: cache design

Role

Speed-up the lookup of frequently accessed data.

Context

- With associativity decreases the miss rate,
- Current implementations are limited to 8-ways associativity,
- Speed and area limitations.

Our proposal

- Adaptation of the network:
  - Given the address, find the associated data,
- Additionnal error probability: 10^{-30},
- Area consumption reduced by 65%,
- Pipelined architecture using a few clock cycles.

An example: cache design

Role

Speed-up the lookup of frequently accessed data.

Context

- With associativity decreases the miss rate,
- Current implementations are limited to 8-ways associativity,
- Speed and area limitations.

Our proposal

- Adaptation of the network:
  - Given the address, find the associated data,
- Additionnal error probability: 10^{-30},
- Area consumption reduced by 65%,
- Pipelined architecture using a few clock cycles.

An example: cache design

Role

Speed-up the lookup of frequently accessed data.

Context

- With associativity decreases the miss rate,
- Current implementations are limited to 8-ways associativity,
- Speed and area limitations.

Our proposal

- Adaptation of the network:
  - Given the address, find the associated data,
- Additionnal error probability: 10^{-30},
- Area consumption reduced by 65%,
- Pipelined architecture using a few clock cycles.

An example: cache design

Role

Speed-up the lookup of frequently accessed data.

Context

- With associativity decreases the miss rate,
- Current implementations are limited to 8-ways associativity,
- Speed and area limitations.

Our proposal

- Adaptation of the network:
  - Given the address, find the associated data,
- Additionnal error probability: 10^{-30},
- Area consumption reduced by 65%,
- Pipelined architecture using a few clock cycles.

An example: cache design

Role

Speed-up the lookup of frequently accessed data.

Context

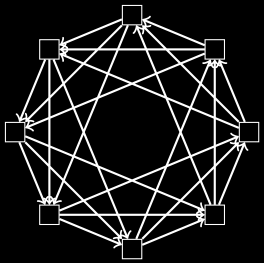
- With associativity decreases the miss rate,
- Current implementations are limited to 8-ways associativity,
- Speed and area limitations.

Our proposal

- Adaptation of the network:
  - Given the address, find the associated data,
- Additionnal error probability: 10^{-30},
- Area consumption reduced by 65%,
- Pipelined architecture using a few clock cycles.

Plan

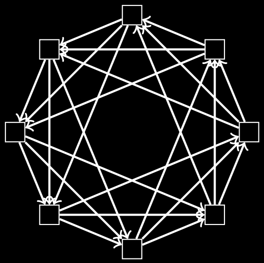
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

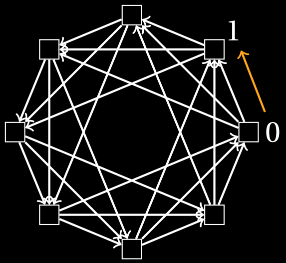
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

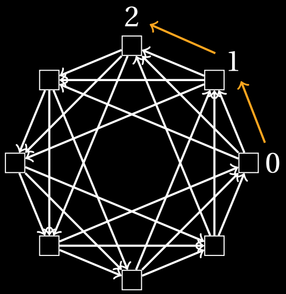
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

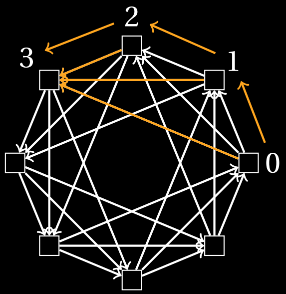
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

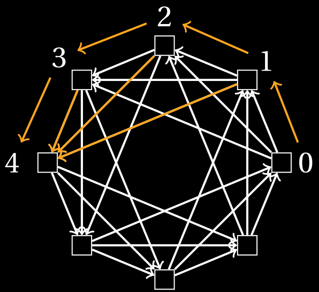
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

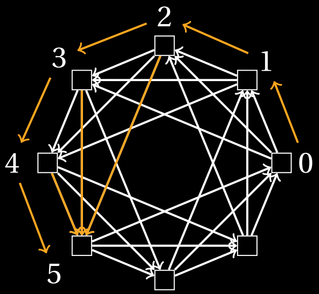
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

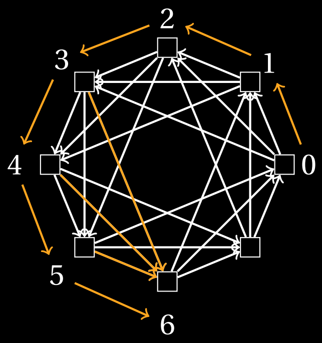
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

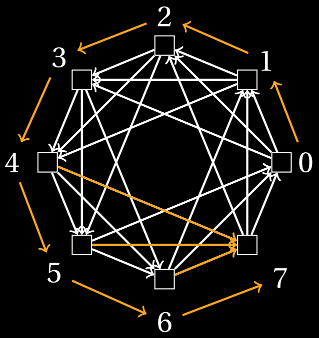
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

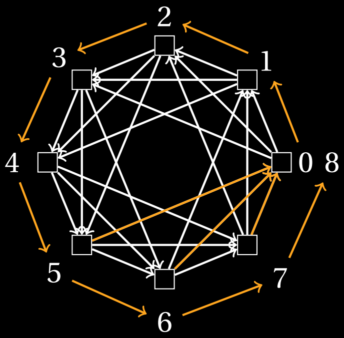
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

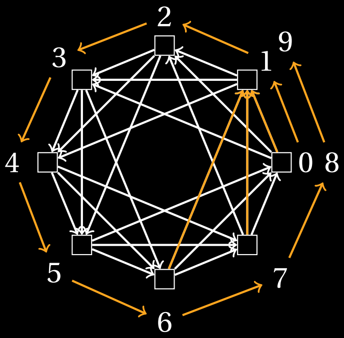
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

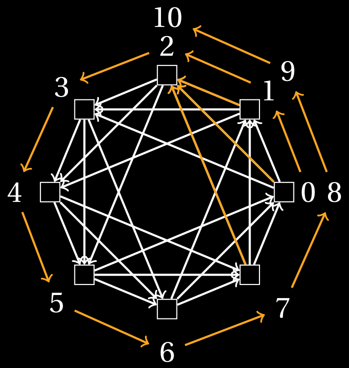
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

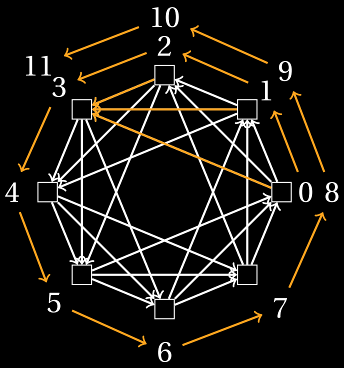
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

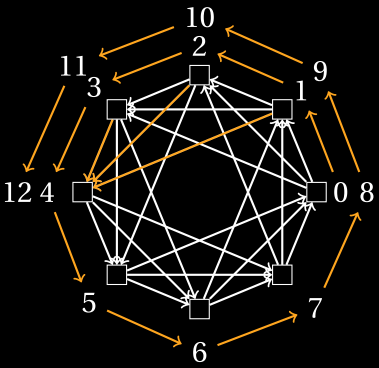
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

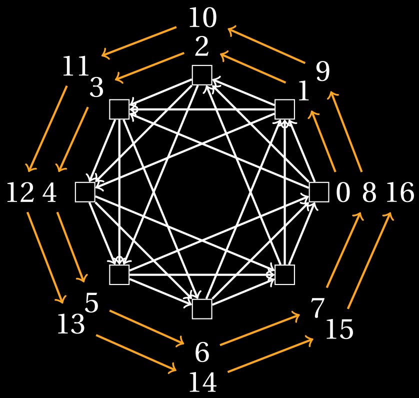
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

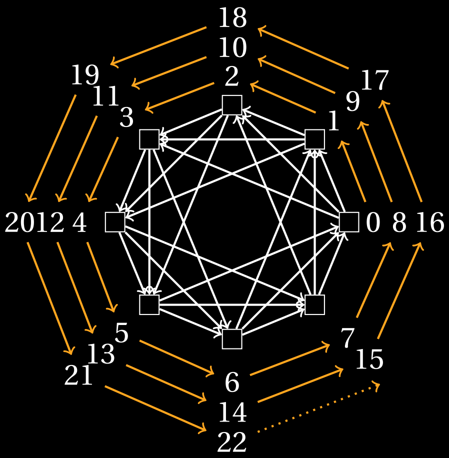
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

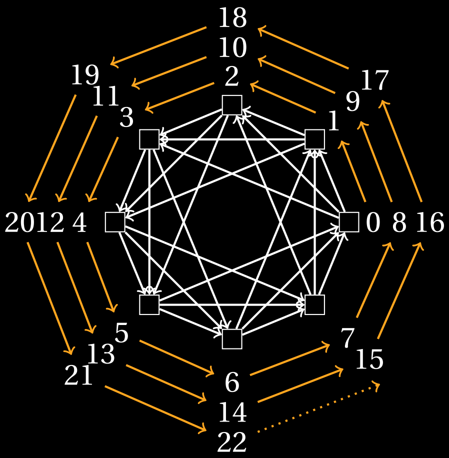
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

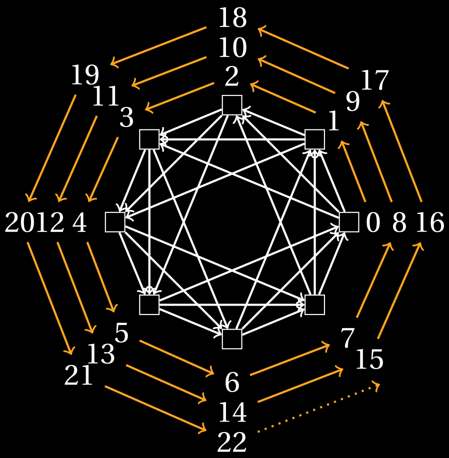
Direction II: learning arbitrarily long sequences



Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

Direction II: learning arbitrarily long sequences

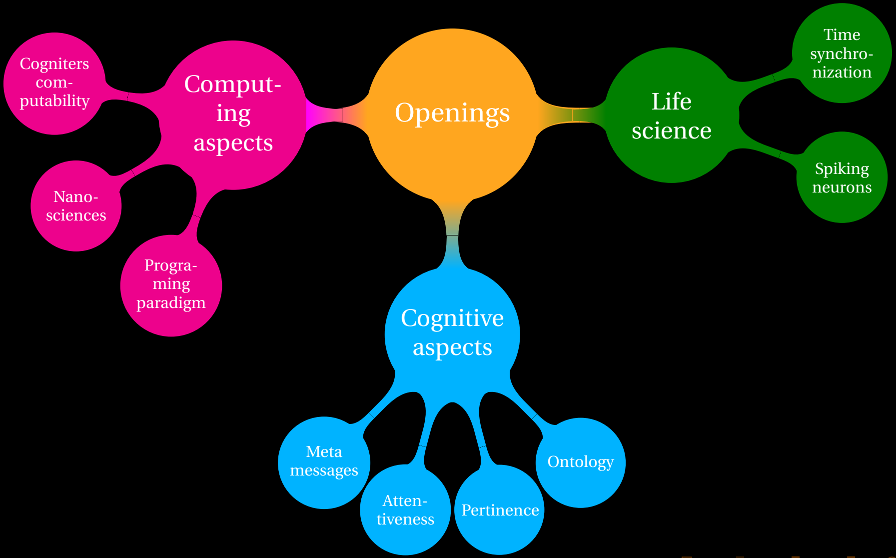


Performance

- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in sequences,
- $m = 1823$ learned sequences,
- $P_e \leq 0.01$.

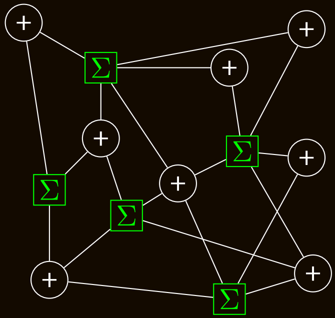
Plan

Openings


Openings
Comput-ing aspects
Cogniters com-putability
Nano-sciences
Progra-ming paradigm
Life science
Time synchro-nization
Spiking neurons
Cognitive aspects
Meta messages
Atten-tiveness
Pertinence
Ontology

Thank you for your attention. I am at your disposal if you have any question.

Error correcting code



Neocortical network