How to improve associative memories using neural coding

Vincent Gripon

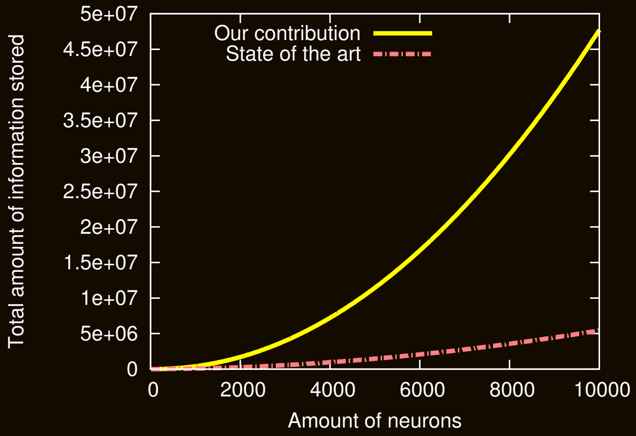
Joint work with:
Claude Berrou, Behrooz Kamary Aliabadi and Xiaoran Jiang

Télécom Bretagne, Lab-STICC

2012, Sept. 13th

In one figure...

Storing messages in recurrent neural networks



Outline

Plan

1. Associative memories and error correcting codes
   - Associative memory
   - Error correcting codes
2. Sparse networks, principles and performance
   - Storing
   - Retrieving
   - Performance
3. Developments
   - Blurred messages
   - Correlated sources
   - Sparse messages
4. Conclusion

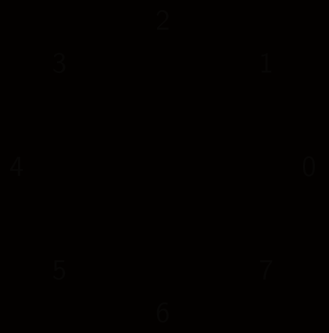
Associative memories and the Hopfield network

What is an associative memory?

Two operations:

- Store a message,
- Retrieve a previously stored message from part of its content.

Our reference: the Hopfield network



Example:

- Store binary message -11-111-1-11
- Retrieve it from -11-111-1?1

What is an associative memory?

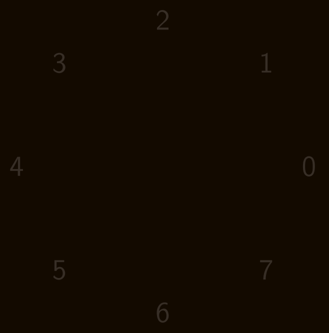
Two operations:

- Store a message,
- Retrieve a previously stored message from part of its content.

Our reference: the Hopfield network

Example:

- Store binary message -11-111-1-11
- Retrieve it from -11-111-1?1



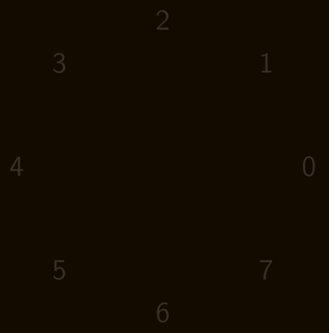
Associative memories and the Hopfield network

What is an associative memory?

Two operations:

- Store a message,
- Retrieve a previously stored message from part of its content.

Our reference: the Hopfield network



Example:

- Store binary message -11-111-1-11
- Retrieve it from -11-111-1?1

Associative memories and the Hopfield network

What is an associative memory?

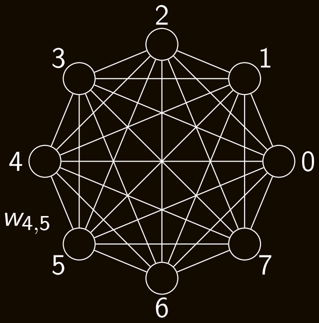
Two operations:

- Store a message,
- Retrieve a previously stored message from part of its content.

Our reference: the Hopfield network

Example:

- Store binary message -11-111-1-11
- Retrieve it from -11-111-1?1



Associative memories and the Hopfield network

What is an associative memory?

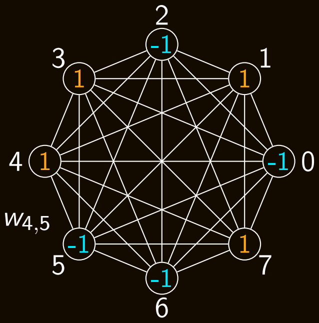
Two operations:

- Store a message,
- Retrieve a previously stored message from part of its content.

Our reference: the Hopfield network

Example:

- Store binary message -11-111-1-11
- Retrieve it from -11-111-1?1



What is an associative memory?

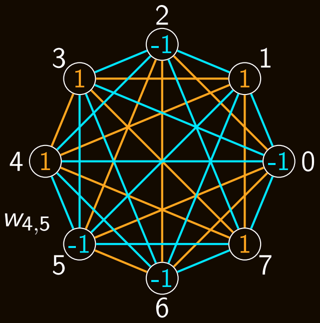
Two operations:

- Store a message,
- Retrieve a previously stored message from part of its content.

Our reference: the Hopfield network

Example:

- Store binary message -11-111-1-11
- Retrieve it from -11-111-1?1



Associative memories and the Hopfield network

What is an associative memory?

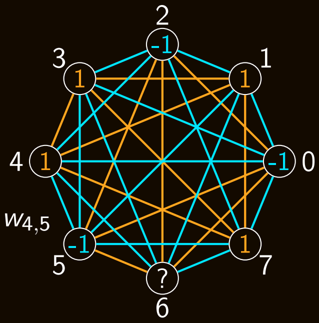
Two operations:

- Store a message,
- Retrieve a previously stored message from part of its content.

Our reference: the Hopfield network

Example:

- Store binary message -11-111-1-11
- Retrieve it from -11-111-1?1



What is an associative memory?

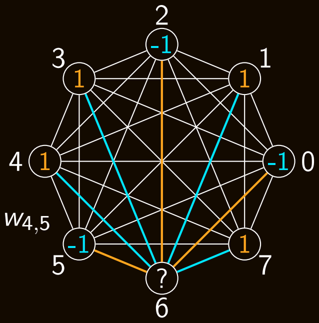
Two operations:

- Store a message,
- Retrieve a previously stored message from part of its content.

Our reference: the Hopfield network

Example:

- Store binary message -11-111-1-11
- Retrieve it from -11-111-1?1



Associative memories and the Hopfield network

What is an associative memory?

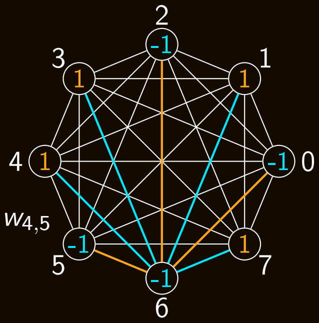
Two operations:

- Store a message,
- Retrieve a previously stored message from part of its content.

Our reference: the Hopfield network

Example:

- Store binary message -11-111-1-11
- Retrieve it from -11-111-1-11



Performance and bounds

Hopfield networks (n neurons $\longleftrightarrow$)

- Diversity : $M = \frac{n}{2log(n)}$, $\leftrightarrow$
- Capacity : $\frac{n^2}{2log(n)}$,
- Efficiency $\approx \frac{1}{log(n)log_2(M+1)}$.

Example with $n = 790$:

Hopfield networks (n neurons)

- Diversity : $M = \frac{n}{2log(n)}$,
- Capacity : $\frac{n^2}{2log(n)}$, =
- Efficiency $\approx \frac{1}{log(n)log_2(M+1)}$.

Example with $n = 790$:

Performance and bounds

Hopfield networks (n neurons)

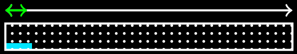
- **Diversity** : $M = \frac{n}{2log(n)}$,
- **Capacity** : $\frac{n^2}{2log(n)}$,
- **Efficiency** $\approx \frac{1}{log(n)log_2(M+1)}$.

Example with $n = 790$:

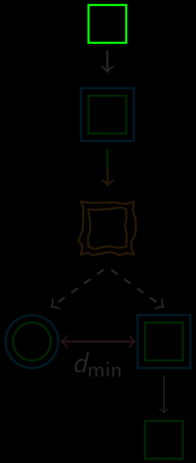
Hopfield networks (n neurons $\longleftrightarrow$)

- Diversity : $M = \frac{n}{2log(n)}$, $\leftrightarrow$
- Capacity : $\frac{n^2}{2log(n)}$, ▬ = ▮
- Efficiency $\approx \frac{1}{log(n)log_2(M+1)}$.

Example with $n = 790$:



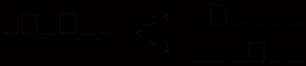
Error correcting codes



Example: the thrifty code

- Code containing only binary words with a single "1":

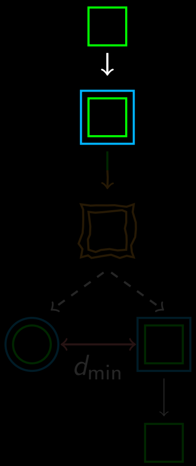
- Drawback: $d_{\min} = 2$:



- But easy to decode and minimise the energy:

- These codes can be associated like the distributed codes. . .

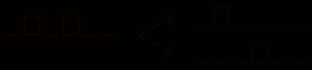
Error correcting codes



Example: the thrifty code

- Code containing only binary words with a single "1":

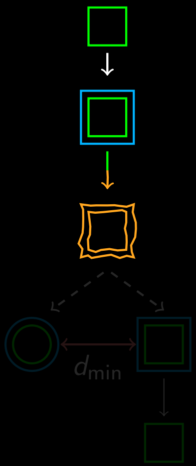
- Drawback: $d_{\min} = 2$:



- But easy to decode and minimise the energy:

- These codes can be associated like the distributed codes. . .

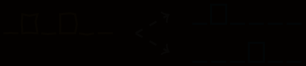
Error correcting codes



Example: the thrifty code

- Code containing only binary words with a single "1":

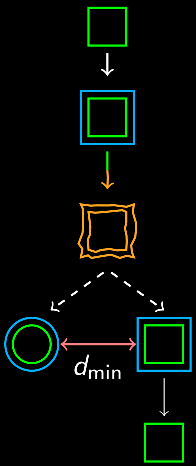
- Drawback: $d_{min} = 2$;



- But easy to decode and minimise the energy:

- These codes can be associated like the distributed codes. . .

Error correcting codes



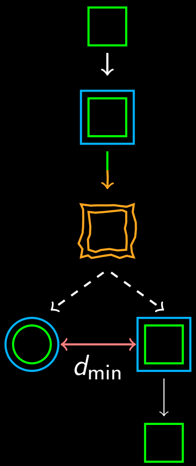
Example: the thrifty code

- Code containing only binary words with a single "1":
- Drawback: $d_{\min} = 2$;
- But easy to decode and minimise the energy:

winner-take-all

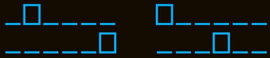
- These codes can be associated like the distributed codes. . .

Error correcting codes



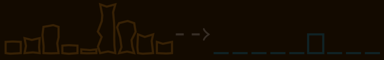
Example: the thrifty code

- Code containing only binary words with a single “1”:



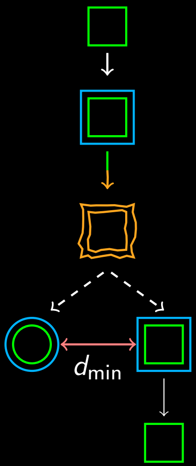
- Drawback: $d_{min} = 2$:

- But easy to decode and minimise the energy:



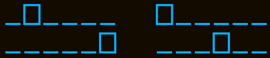
- These codes can be associated like the distributed codes. . .

Error correcting codes

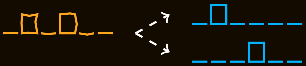


Example: the thrifty code

- Code containing only binary words with a single "1":



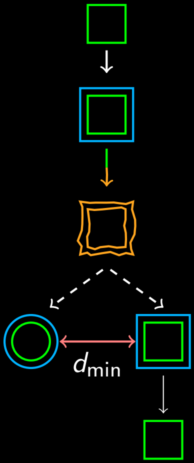
- Drawback: $d_{\min} = 2$:



- But easy to decode and minimise the energy:

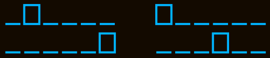
- These codes can be associated like the distributed codes. . .

Error correcting codes

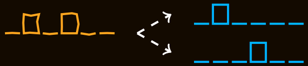


Example: the thrifty code

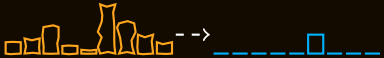
- Code containing only binary words with a single "1":



- Drawback: $d_{\text{min}} = 2$:



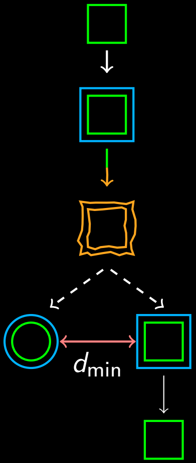
- But easy to decode and minimise the energy:



winner-take-all

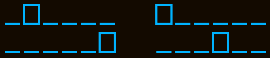
- These codes can be associated like the distributed codes...

Error correcting codes

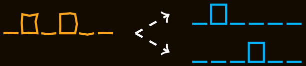


Example: the thrifty code

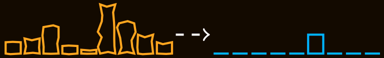
- Code containing only binary words with a single "1":



- Drawback: $d_{min} = 2$:



- But easy to decode and minimise the energy:



winner-take-all

- These codes can be associated like the distributed codes. . .

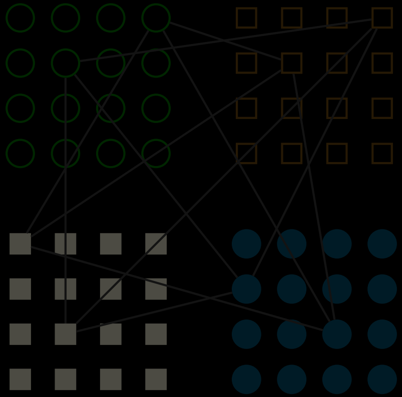
Plan

Our model: Storing

Example:

Message to store: 1000001100101001

- For example: a network of $c = 4$ clusters made of $l = 16$ neurons each,
- $\underbrace{1000}_{j_1 \text{ in } c_1}\ \underbrace{0011}_{j_2 \text{ in } c_2}\ \underbrace{0010}_{j_3 \text{ in } c_3}\ \underbrace{1001}_{j_4 \text{ in } c_4}$,

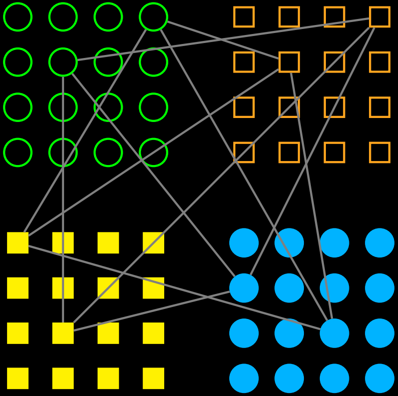


Our model: Storing

Example:

Message to store: 1000001100101001

- For example: a network of $c = 4$ clusters made of $l = 16$ neurons each,
- $\underbrace{1000}_{j_1 \text{ in } c_1}\ \underbrace{0011}_{j_2 \text{ in } c_2}\ \underbrace{0010}_{j_3 \text{ in } c_3}\ \underbrace{1001}_{j_4 \text{ in } c_4}$,

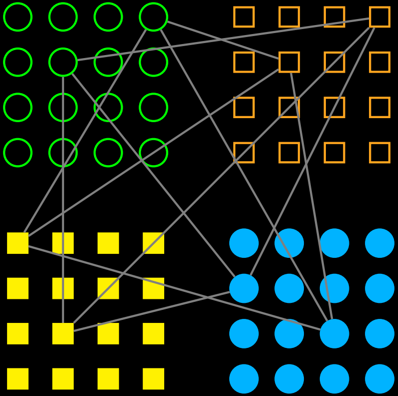


Our model: Storing

Example:

Message to store: 1000001100101001

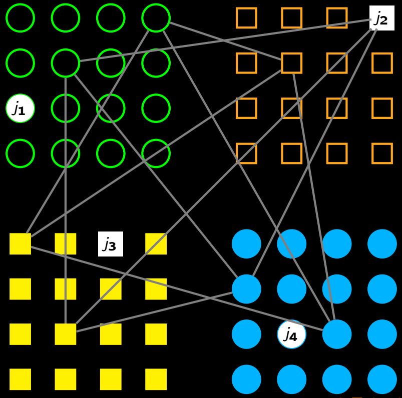
- For example: a network of $c = 4$ clusters made of $l = 16$ neurons each,
- $\underbrace{1000}_{j_1 \text{ in } c_1}\ \underbrace{0011}_{j_2 \text{ in } c_2}\ \underbrace{0010}_{j_3 \text{ in } c_3}\ \underbrace{1001}_{j_4 \text{ in } c_4}$,



Example:

Message to store: 1000001100101001

- For example: a network of $c = 4$ clusters made of $l = 16$ neurons each,
- $\underbrace{1000}_{j_1 \text{ in } c_1}\ \underbrace{0011}_{j_2 \text{ in } c_2}\ \underbrace{0010}_{j_3 \text{ in } c_3}\ \underbrace{1001}_{j_4 \text{ in } c_4}$,

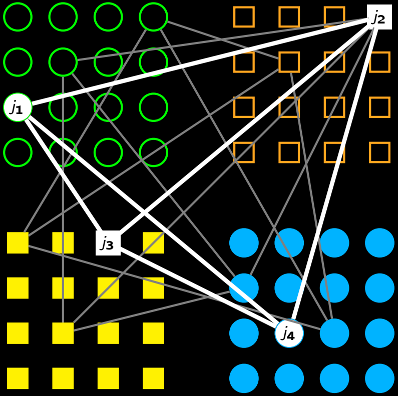


Our model: Storing

Example:

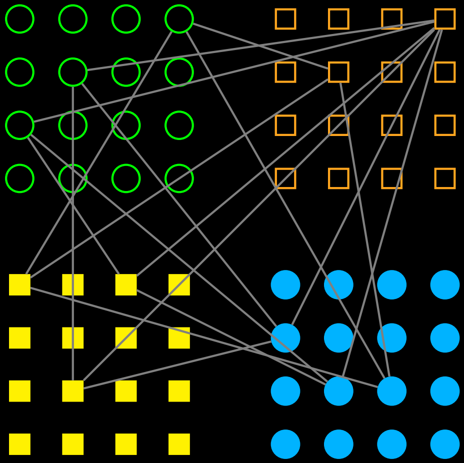
Message to store: 1000001100101001

- For example: a network of $c = 4$ clusters made of $l = 16$ neurons each,
- $\underbrace{1000}_{j_1 \text{ in } c_1} \underbrace{0011}_{j_2 \text{ in } c_2} \underbrace{0010}_{j_3 \text{ in } c_3} \underbrace{1001}_{j_4 \text{ in } c_4}$,



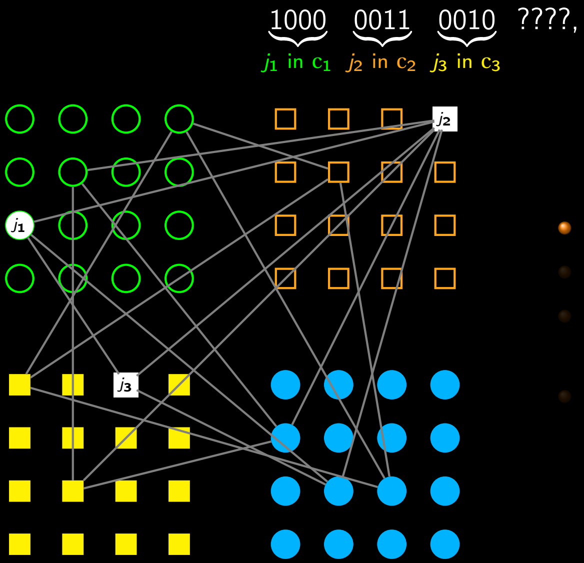
Our model: retrieving

$\underbrace{1000}_{j_1 \text{ in } c_1}\ \underbrace{0011}_{j_2 \text{ in } c_2}\ \underbrace{0010}_{j_3 \text{ in } c_3}\ ????,$



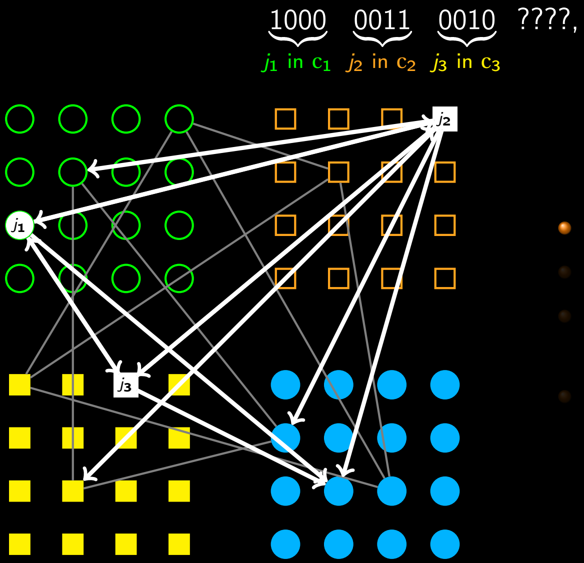
- Local connection,
- Global decoding: sum,
- Local decoding: winner-take-all,
- Possibly iterate the two decodings.

Our model: retrieving



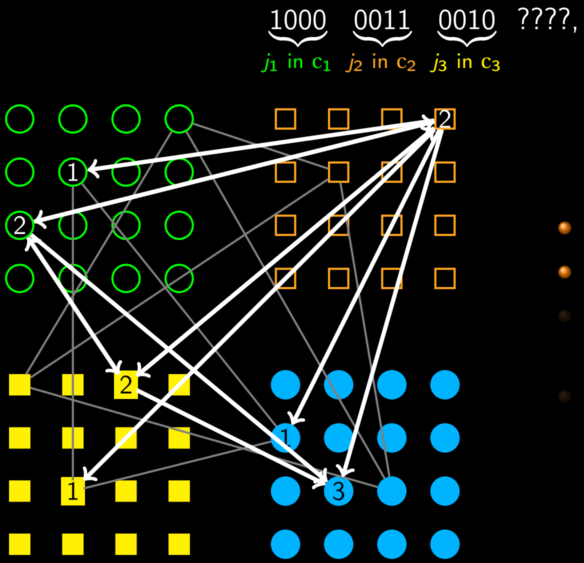
- Local connection,
- Global decoding: sum,
- Local decoding: winner-take-all,
- Possibly iterate the two decodings.

Our model: retrieving



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

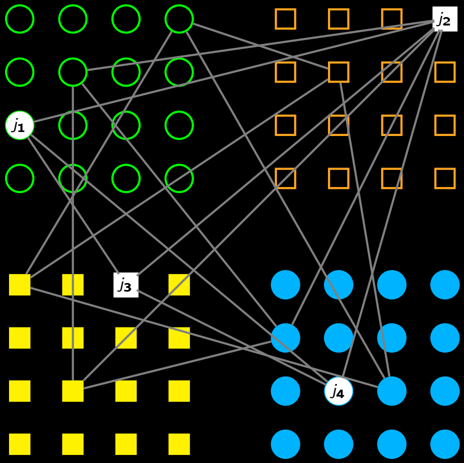
Our model: retrieving



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

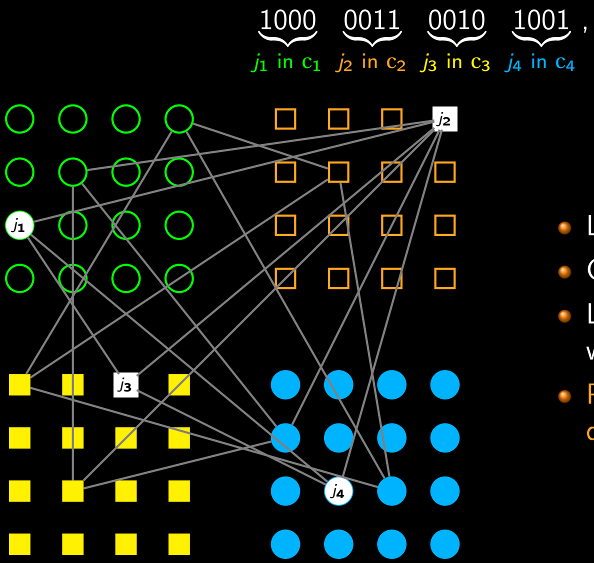
Our model: retrieving

$\underbrace{1000}_{j_1 \text{ in } c_1} \; \underbrace{0011}_{j_2 \text{ in } c_2} \; \underbrace{0010}_{j_3 \text{ in } c_3} \; \underbrace{1001}_{j_4 \text{ in } c_4},$



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

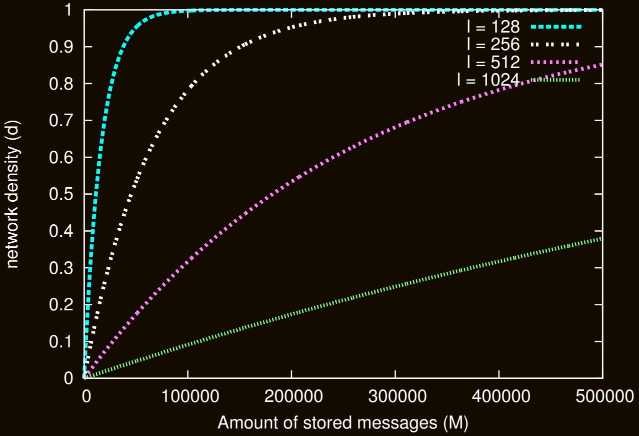
Our model: retrieving



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

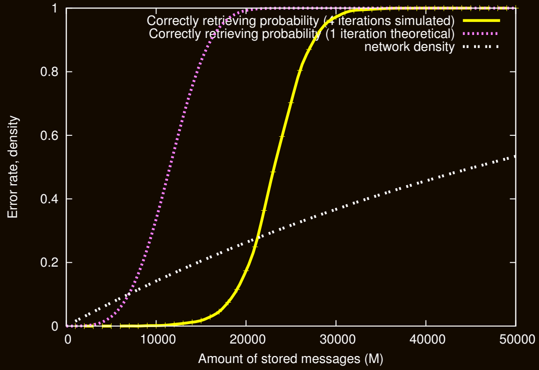
Density

A parameter to assess performance



Performance (1/3)

As an associative memory

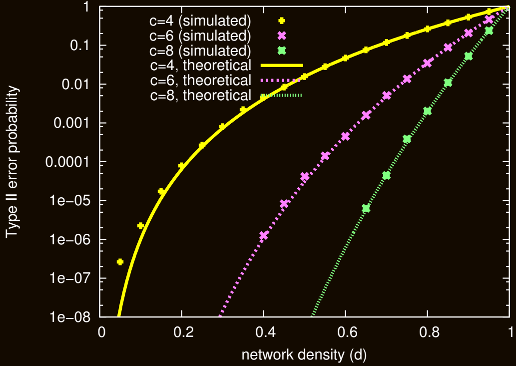


$c = 8$ clusters of $l = 256$ neurons each ($\sim$ messages of 64 bits),
Error probability when retrieving messages half erased.

Hopfield network ($n = 790$)

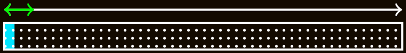
Our network

Classification

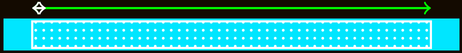


Type II error rate for various sizes of clusters c and for $l = 512$ neurons per cluster.

Hopfield network ($n = 740$)

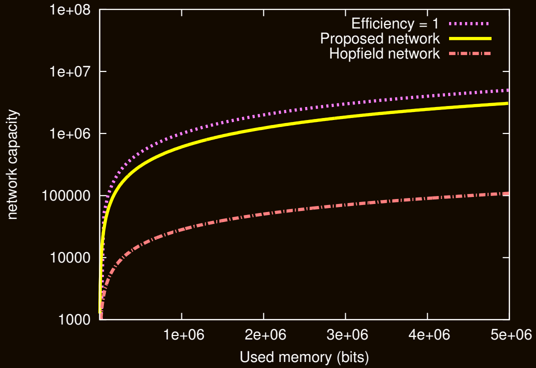


Our network



Comparison of capacities of our network and of the Hopfield one

Performance (3/3)

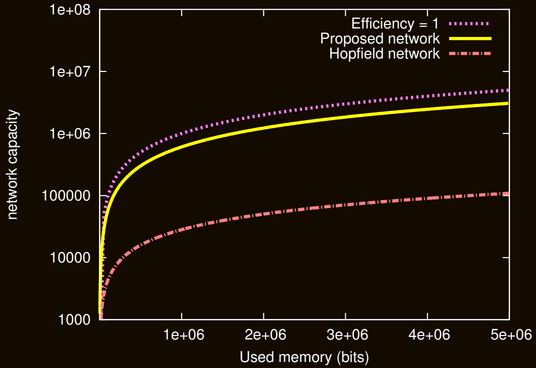


Comparison of the capacities of the Hopfield network with ours (as associative memories) and for the same amount of memory used.

But with some limitations...

Comparison of capacities of our network and of the Hopfield one

Performance (3/3)



Comparison of the capacities of the Hopfield network with ours (as associative memories) and for the same amount of memory used.

But with some limitations...

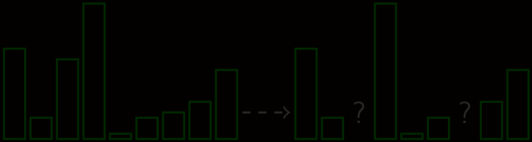
Plan

Blurred messages

Limitation

Partial messages must contain perfect information.

Noise model



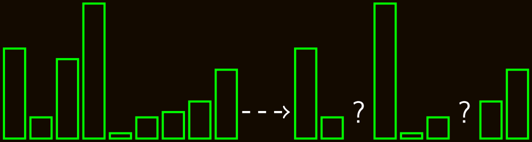
Soft decoding

Blurred messages

Limitation

Partial messages must contain perfect information.

Noise model



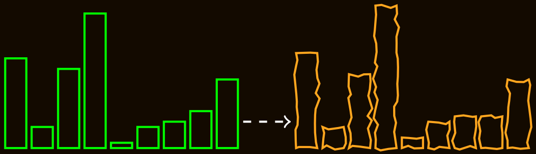
Soft decoding

Blurred messages

Limitation

Partial messages must contain perfect information.

Noise model



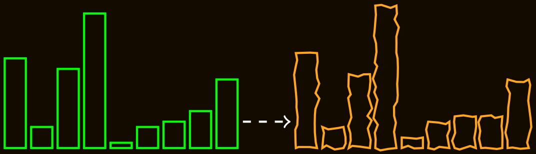
Soft decoding

Blurred messages

Limitation

Partial messages must contain perfect information.

Noise model



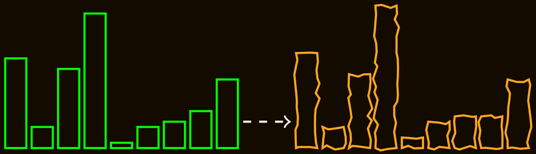
Soft decoding

Blurred messages

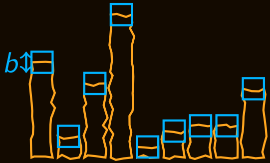
Limitation

Partial messages must contain perfect information.

Noise model



Soft decoding

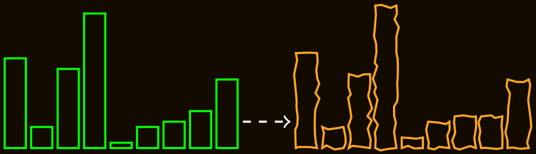


Blurred messages

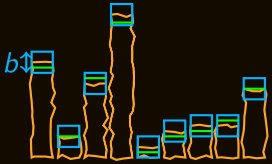
Limitation

Partial messages must contain perfect information.

Noise model

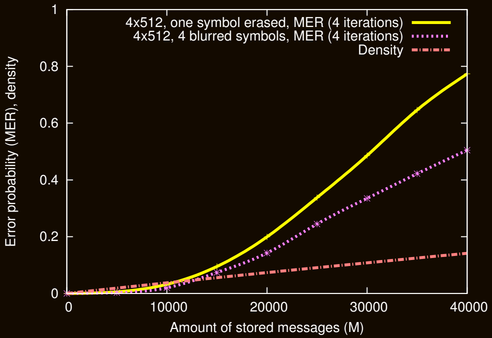


Soft decoding



Performance

Simulations



Comparison of performance when messages are partially erased and when they are blurred ($b = 5$).

Correlated messages

Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

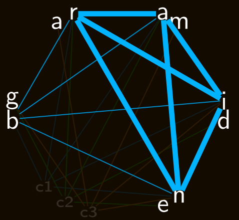
Correlated messages

Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

brain



Correlated messages

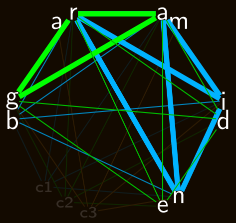
Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

brain

grade



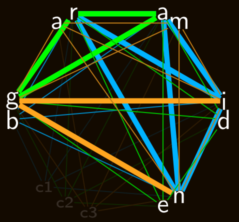
Correlated messages

Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

brain
grade
gamin



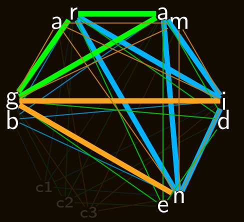
Correlated messages

Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

brain
grade
gamin
grain



Correlated messages

Limitation

With correlations grows the number of Type II errors.

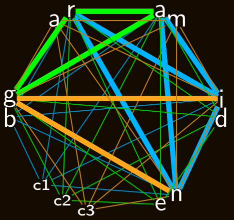
Fighting correlation by adding random redundancy

brain +c1

grade +c2

gamin +c3

grain +c?



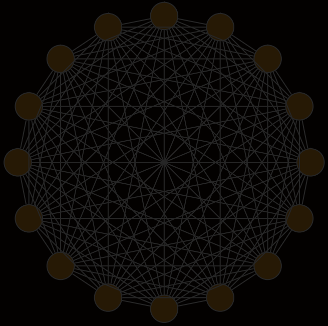
Towards a fourth level of sparsity

Limitations

- Clusters must be large and few,
- Stored messages are all of the same length.

Illustration

0000101000001011



Idea

1. Shorter messages,
2. Clusters and thrifty codes,
3. Sparse network,
4. Sparse messages.

Solution

- Global winner-take-all.

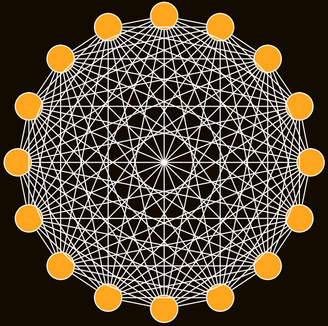
Towards a fourth level of sparsity

Limitations

- Clusters must be large and few,
- Stored messages are all of the same length.

Illustration

0000101000001011



Idea

1. Shorter messages,
2. Clusters and thrifty codes,
3. Sparse network,
4. Sparse messages.

Solution

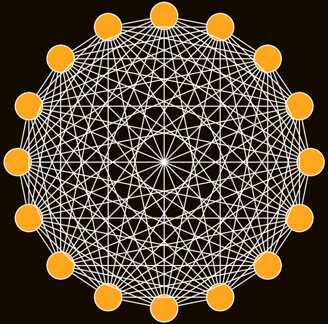
- Global winner-take-all.

Limitations

- Clusters must be large and few,
- Stored messages are all of the same length.

Illustration

→00100101←



Idea

1. Shorter messages,
2. Clusters and thrifty codes,
3. Sparse network,
4. Sparse messages.

Solution

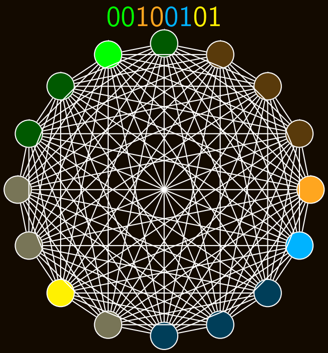
- Global winner-take-all.

Towards a fourth level of sparsity

Limitations

- Clusters must be large and few,
- Stored messages are all of the same length.

Illustration



Idea

1. Shorter messages,
2. Clusters and thrifty codes,
3. Sparse network,
4. Sparse messages.

Solution

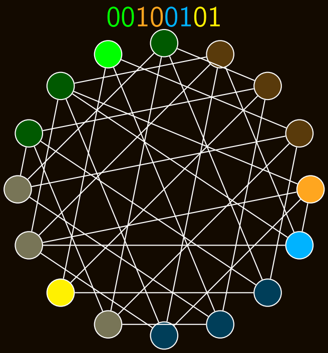
- Global winner-take-all.

Towards a fourth level of sparsity

Limitations

- Clusters must be large and few,
- Stored messages are all of the same length.

Illustration



Idea

1. Shorter messages,
2. Clusters and thrifty codes,
3. Sparse network,
4. Sparse messages.

Solution

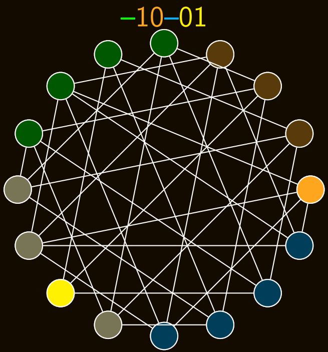
- Global winner-take-all.

Towards a fourth level of sparsity

Limitations

- Clusters must be large and few,
- Stored messages are all of the same length.

Illustration



Idea

1. Shorter messages,
2. Clusters and thrifty codes,
3. Sparse network,
4. Sparse messages.

Solution

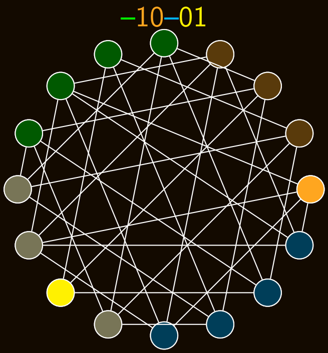
- Global winner-take-all.

Towards a fourth level of sparsity

Limitations

- Clusters must be large and few,
- Stored messages are all of the same length.

Illustration



Idea

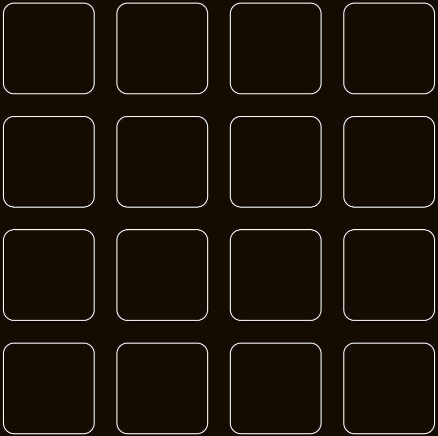
1. Shorter messages,
2. Clusters and thrifty codes,
3. Sparse network,
4. Sparse messages.

Solution

- Global winner-take-all.

Global winner-take-all

Illustration



Idea

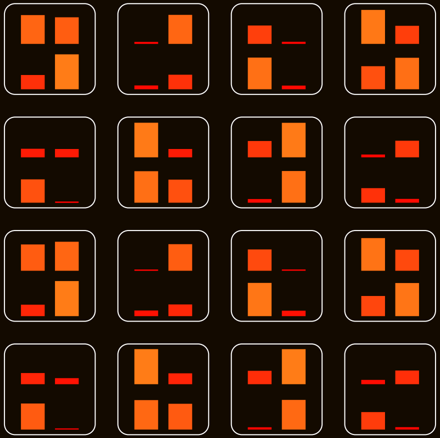
- After global message passing...
- After local maximum selections...
- Global maximum selection.

Interests

- Diversity $\propto c^2$,
- Stored messages length may vary.

Global winner-take-all

Illustration



Idea

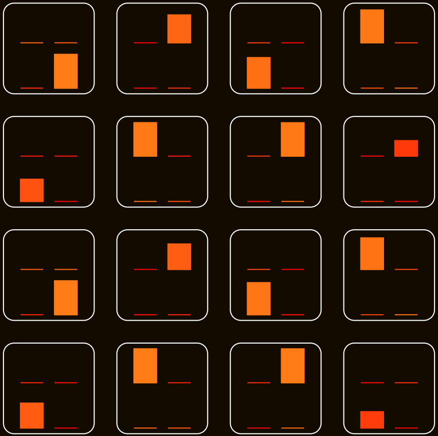
- After global message passing. . .
- After local maximum selections. . .
- Global maximum selection.

Interests

- Diversity $\propto c^2$,
- Stored messages length may vary.

Global winner-take-all

Illustration



Idea

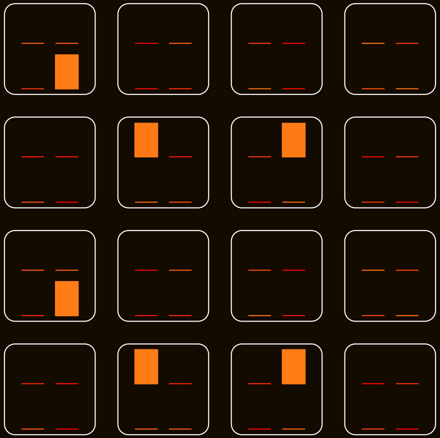
- After global message passing. . .
- After local maximum selections. . .
- Global maximum selection.

Interests

- Diversity $\propto c^2$,
- Stored messages length may vary.

Global winner-take-all

Illustration



Idea

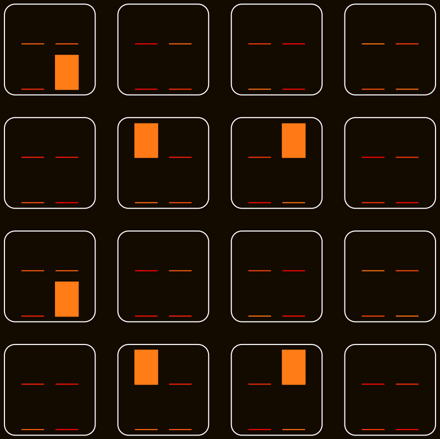
- After global message passing...
- After local maximum selections...
- Global maximum selection.

Interests

- Diversity $\propto c^2$,
- Stored messages length may vary.

Global winner-take-all

Illustration



Idea

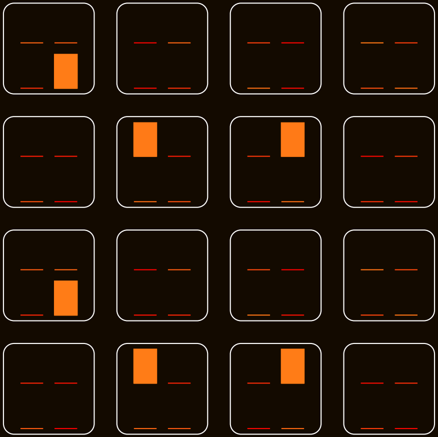
- After global message passing. . .
- After local maximum selections. . .
- Global maximum selection.

Interests

- Diversity $\propto c^2$,
- Stored messages length may vary.

Global winner-take-all

Illustration



Idea

- After global message passing. . .
- After local maximum selections. . .
- Global maximum selection.

Interests

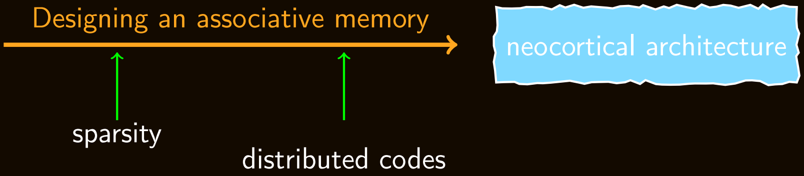
- Diversity $\propto c^2$,
- Stored messages length may vary.

Plan

1. Associative memories and error correcting codes
   - Associative memory
   - Error correcting codes
2. Sparse networks, principles and performance
   - Storing
   - Retrieving
   - Performance
3. Developments
   - Blurred messages
   - Correlated sources
   - Sparse messages
4. Conclusion

Conclusion

Approach

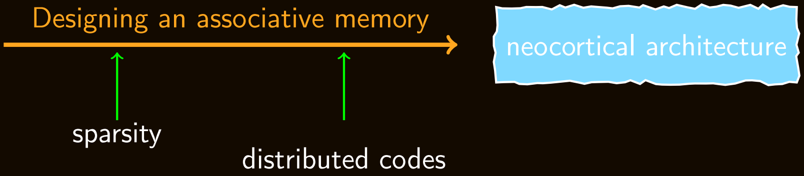


Results

- Nearly optimal capacities, substantial diversities,
- Massively parallel architecture,
- Analogies with neurobiological architecture and functioning,
- Robustness, resiliency. . . ,
- Degrees of freedom: inhibitions, time, weights,
- No trade off required between performance and plausibility.

Conclusion

Approach



Results

- Nearly optimal capacities, substantial diversities,
- Massively parallel architecture,
- Analogies with neurobiological architecture and functioning,
- Robustness, resiliency. . . ,
- Degrees of freedom: inhibitions, time, weights,
- No trade off required between performance and plausibility.

Conclusion

Approach

Designing an associative memory → neocortical architecture

↑ sparsity

↑ distributed codes

Results

- Nearly optimal capacities, substantial diversities,
- Massively parallel architecture,
- Analogies with neurobiological architecture and functioning,
- Robustness, resiliency. . . ,
- Degrees of freedom: inhibitions, time, weights,
- No trade off required between performance and plausibility.

Conclusion

Approach

Designing an associative memory → neocortical architecture

↑ sparsity

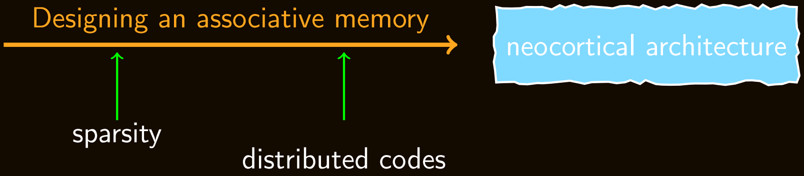
↑ distributed codes

Results

- Nearly optimal capacities, substantial diversities,
- Massively parallel architecture,
- Analogies with neurobiological architecture and functioning,
- Robustness, resiliency. . . ,
- Degrees of freedom: inhibitions, time, weights,
- No trade off required between performance and plausibility.

Conclusion

Approach

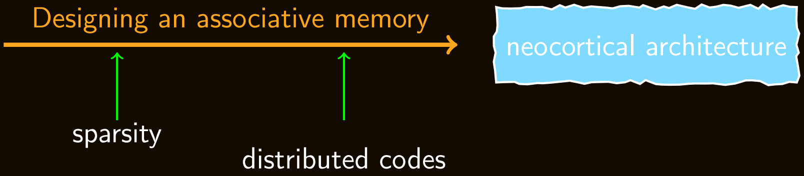


Results

- Nearly optimal capacities, substantial diversities,
- Massively parallel architecture,
- Analogies with neurobiological architecture and functioning,
- Robustness, resiliency. . . ,
- Degrees of freedom: inhibitions, time, weights,
- No trade off required between performance and plausibility.

Conclusion

Approach

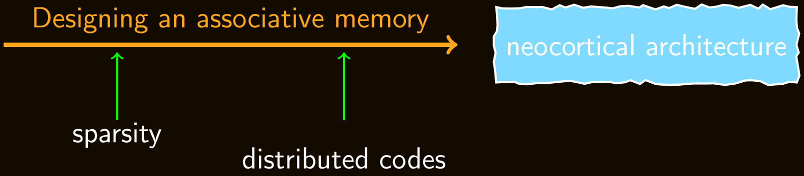


Results

- Nearly optimal capacities, substantial diversities,
- Massively parallel architecture,
- Analogies with neurobiological architecture and functioning,
- Robustness, resiliency. . . ,
- Degrees of freedom: inhibitions, time, weights,
- No trade off required between performance and plausibility.

Conclusion

Approach



Results

- Nearly optimal capacities, substantial diversities,
- Massively parallel architecture,
- Analogies with neurobiological architecture and functioning,
- Robustness, resiliency. . . ,
- Degrees of freedom: inhibitions, time, weights,
- No trade off required between performance and plausibility.