Networks of neural cliques

Vincent Gripon

Télécom Bretagne, Lab-STICC

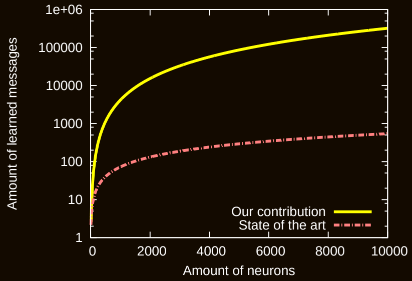
McGill University

2011, Nov. 11th

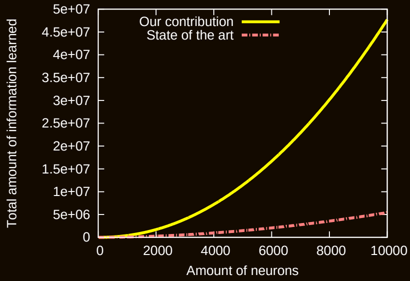
In a word...

Learning messages in recurrent neural networks

Learning diversity



Learning capacity



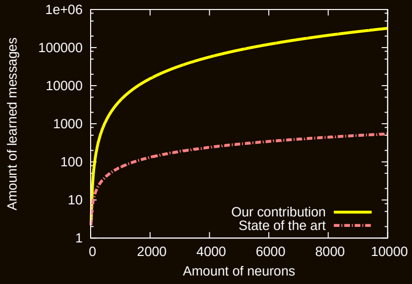
Our contribution

Hopfield neural networks

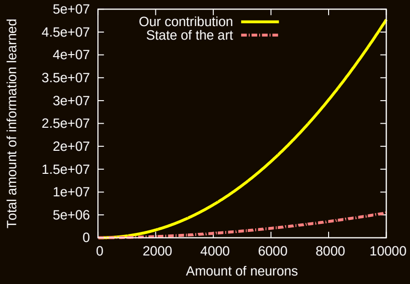
In a word. . .

Learning messages in recurrent neural networks

Learning diversity



Learning capacity



Our contribution

Sparsity

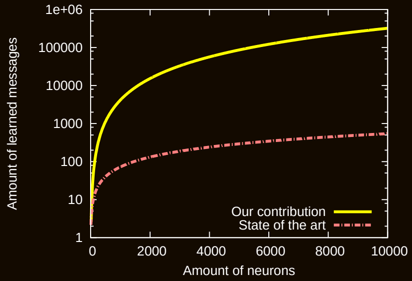
Error correcting code

Hopfield neural networks

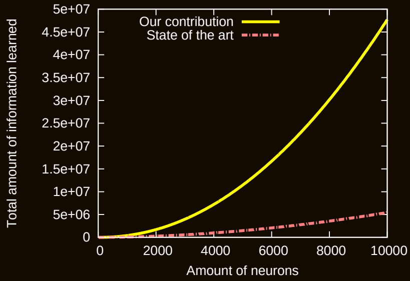
In a word...

Learning messages in recurrent neural networks

Learning diversity



Learning capacity



Our contribution

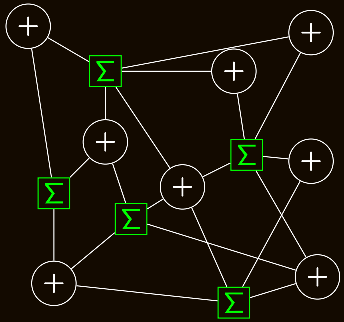
Sparsity
Error correcting code

Biologic plausibility

Hopfield neural networks

Starting idea

LDPC decoder



Neocortical “decoder”

decoding	=	remembering
nodes $\boxed{\Sigma}$	=	neurons
parity	=	?
?	=	learning

Outline

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

Plan

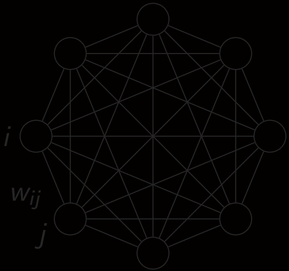
1. Associative memories and error correcting codes
   - Associative memory
   - Error correcting codes
   - Code of cliques
2. Sparse networks, principles and performance
   - Learning
   - Retrieving
   - Performance
3. Developments
   - Blurred messages
   - Correlated sources
   - Sparse messages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

Associative memories

Two operations:

- Learning messages,
- Retrieving previously learned messages from part of their content.

State of the art: the Hopfield network



- Learning: M binary messages $\mathbf{d}^m$:

$$w_{ij} = \sum_{m=1, i \neq j}^{M} d_i^m d_j^m,$$

- Retrieving: iterates

$$\forall i, v_i \leftarrow \mathrm{sgn}\left(\sum_{j \neq i} v_j w_{ij}\right).$$

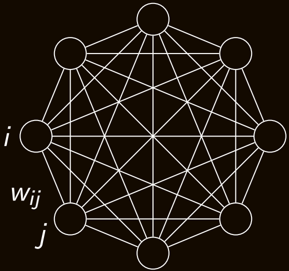
Associative memories and the Hopfield network

Associative memories

Two operations:

- Learning messages,
- Retrieving previously learned messages from part of their content.

State of the art: the Hopfield network



- Learning: M binary messages $\mathbf{d}^m$:

$$w_{ij} = \sum_{m=1, i \neq j}^{M} d_i^m d_j^m,$$

- Retrieving: iterates

$$\forall i, v_i \leftarrow \mathrm{sgn}(\sum_{j \neq i} v_j w_{ij}).$$

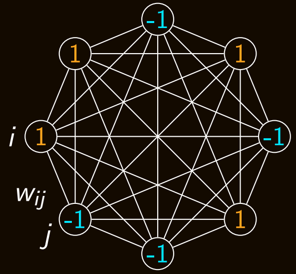
Associative memories and the Hopfield network

Associative memories

Two operations:

- Learning messages,
- Retrieving previously learned messages from part of their content.

State of the art: the Hopfield network



- Learning: M binary messages $\mathbf{d}^m$:

$$w_{ij} = \sum_{m=1, i \neq j}^{M} d_i^m d_j^m,$$

- Retrieving: iterates

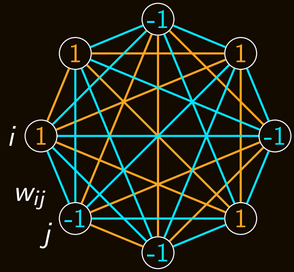
$$\forall i, v_i \leftarrow \mathrm{sgn}\left(\sum_{j \neq i} v_j w_{ij}\right).$$

Associative memories

Two operations:

- Learning messages,
- Retrieving previously learned messages from part of their content.

State of the art: the Hopfield network



- Learning: M binary messages $\mathbf{d}^m$:

$$w_{ij} = \sum_{m=1, i \neq j}^{M} d_i^m d_j^m,$$

- Retrieving: iterates

$$\forall i, v_i \leftarrow \mathrm{sgn}\left(\sum_{j \neq i} v_j w_{ij}\right).$$

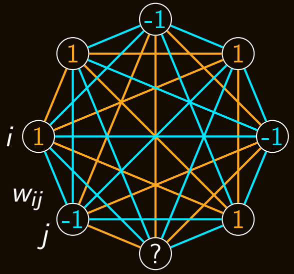
Associative memories and the Hopfield network

Associative memories

Two operations:

- Learning messages,
- Retrieving previously learned messages from part of their content.

State of the art: the Hopfield network



- Learning: M binary messages $\mathbf{d}^m$:

$$w_{ij} = \sum_{m=1, i \neq j}^{M} d_i^m d_j^m,$$

- Retrieving: iterates

$$\forall i, v_i \leftarrow \operatorname{sgn}\left(\sum_{j \neq i} v_j w_{ij}\right).$$

Performance and bounds

Hopfield networks (n neurons $\longleftrightarrow$)

- Diversity : $M = \frac{n}{2log(n)}$, $\leftrightarrow$
- Capacity : $\frac{n^2}{2log(n)}$, $=$
- Total amount of required memory: $\binom{n}{2} log_2(M+1)$,
- $\Rightarrow$ Efficiency $\approx \frac{1}{log(n)log_2(M+1)}$.
- Sensitive connections, length of messages = size of the network, messages and their inverse are learned at the same time. . .

Example with $n = 790$: 

Performance and bounds

Hopfield networks (n neurons $\longleftrightarrow$)

- Diversity : $M = \frac{n}{2log(n)}$, $\leftrightarrow$
- Capacity : $\frac{n^2}{2log(n)}$, $=$
- Total amount of required memory: $\binom{n}{2} log_2(M+1)$,
- $\Rightarrow$ Efficiency $\approx \frac{1}{log(n)log_2(M+1)}$.
- Sensitive connections, length of messages = size of the network, messages and their inverse are learned at the same time...

Example with $n = 790$: 

Performance and bounds

Hopfield networks (n neurons $\longleftrightarrow$)

- Diversity : $M = \frac{n}{2log(n)}$, $\leftrightarrow$
- Capacity : $\frac{n^2}{2log(n)}$, $=$
- Total amount of required memory: $\binom{n}{2} log_2(M+1)$,
- $\Rightarrow$ Efficiency $\approx \frac{1}{log(n)log_2(M+1)}$.
- Sensitive connections, length of messages = size of the network, messages and their inverse are learned at the same time...

Example with $n = 790$:

Performance and bounds

Hopfield networks (n neurons $\longleftrightarrow$)

- Diversity : $M = \frac{n}{2log(n)}$, $\leftrightarrow$
- Capacity : $\frac{n^2}{2log(n)}$, $=$
- Total amount of required memory: $\binom{n}{2} log_2(M+1)$,
- $\Rightarrow$ Efficiency $\approx \frac{1}{log(n)log_2(M+1)}$.
- Sensitive connections, length of messages = size of the network, messages and their inverse are learned at the same time...

Example with $n = 790$:

Performance and bounds

Hopfield networks (n neurons $\longleftrightarrow$)

- Diversity : $M = \frac{n}{2log(n)}$, $\leftrightarrow$
- Capacity : $\frac{n^2}{2log(n)}$, $=$
- Total amount of required memory: $\binom{n}{2} log_2(M+1)$,
- $\Rightarrow$ Efficiency $\approx \frac{1}{log(n)log_2(M+1)}$.
- Sensitive connections, length of messages = size of the network, messages and their inverse are learned at the same time...

Example with $n = 790$:

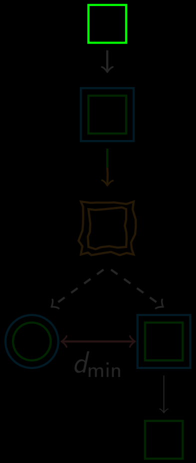
Performance and bounds

Hopfield networks (n neurons $\longleftrightarrow$)

- Diversity : $M = \frac{n}{2log(n)}$, $\leftrightarrow$
- Capacity : $\frac{n^2}{2log(n)}$, $=$
- Total amount of required memory: $\binom{n}{2} log_2(M+1)$,
- $\Rightarrow$ Efficiency $\approx \frac{1}{log(n) log_2(M+1)}$.
- Sensitive connections, length of messages = size of the network, messages and their inverse are learned at the same time...

Example with $n = 790$: 

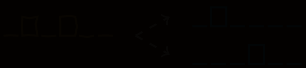
Error correcting codes



Example: the thrifty code

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

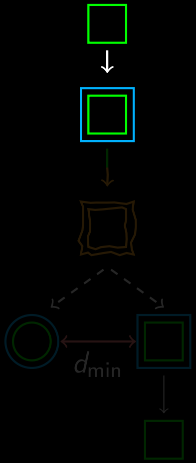
- Drawback: $d_{\min} = 2$:



- But easy to decode and minimise the energy:

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

Error correcting codes



Example: the thrifty code

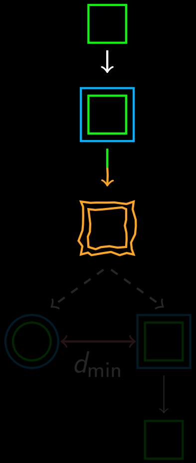
- Code containing only binary words with a single "1":

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

- But easy to decode and minimise the energy:

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

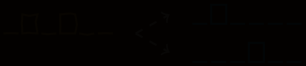
Error correcting codes



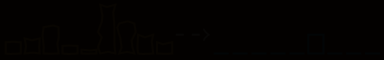
Example: the thrifty code

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

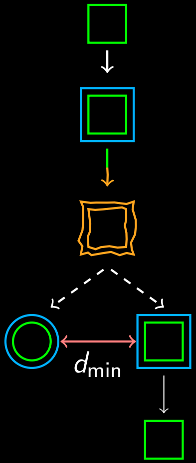
- Drawback: $d_{min} = 2$:



- But easy to decode and minimise the energy:



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



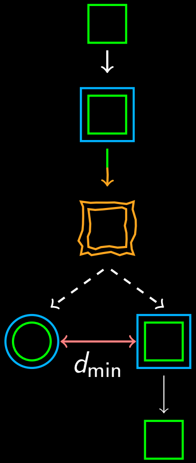
Example: the thrifty code

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

winner-take-all

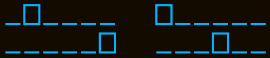
- These codes can be associated like the distributed codes. . .

Error correcting codes



Example: the thrifty code

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

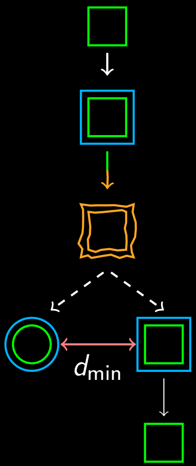


- Drawback: $d_{\min} = 2$:

- But easy to decode and minimise the energy:

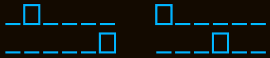
- These codes can be associated like the distributed codes. . .

Error correcting codes

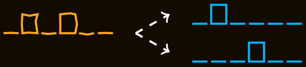


Example: the thrifty code

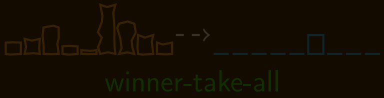
- Code containing only binary words with a single “1”:



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

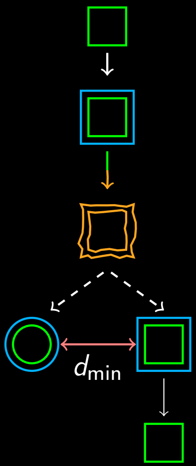


- But easy to decode and minimise the energy:



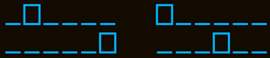
- These codes can be associated like the distributed codes. . .

Error correcting codes

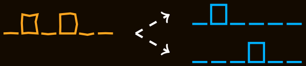


Example: the thrifty code

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



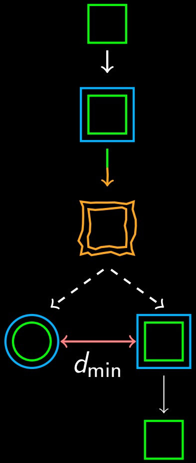
- Drawback: $d_{\text{min}} = 2$:



- But easy to decode and minimise the energy:

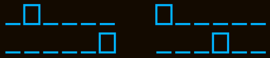
- These codes can be associated like the distributed codes. . .

Error correcting codes

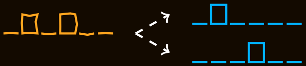


Example: the thrifty code

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



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

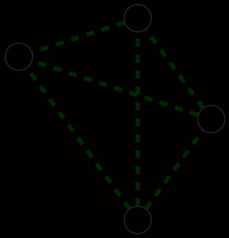


- But easy to decode and minimise the energy:

winner-take-all

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

Example: codewords = 4 nodes cliques



2 distinct nodes
$\Rightarrow d_{\min} = 6$ edges

Clique

Set of nodes that are all connected one to another.

Codes of cliques of size $c \ll n$

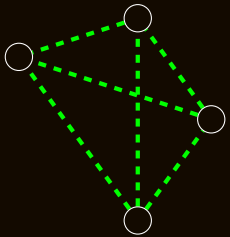
- $d_{\min} = 2(c-1) \approx 2c$,
- $\Rightarrow F = r d_{\min} \approx 2$,
- Cliques are codewords of a very interesting error correcting code.

Codes made of cliques of constant size

Clique

Set of nodes that are all connected one to another.

Example: codewords = 4 nodes cliques



Symbols = edges

2 distinct nodes
$\Rightarrow d_{\min} = 6$ edges

Codes of cliques of size $c \ll n$

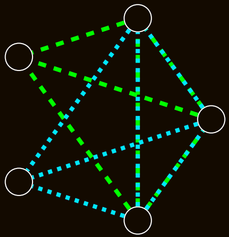
- $d_{\min} = 2(c-1) \approx 2c$,
- $\Rightarrow F = rd_{\min} \approx 2$,
- Cliques are codewords of a very interesting error correcting code.

Codes made of cliques of constant size

Clique

Set of nodes that are all connected one to another.

Example: codewords = 4 nodes cliques



Symbols = edges

2 distinct nodes
$\Rightarrow d_{\min} = 6$ edges

Codes of cliques of size $c \ll n$

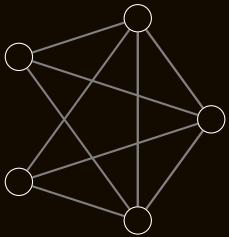
- $d_{\min} = 2(c-1) \approx 2c,$
- $\Rightarrow F = r d_{\min} \approx 2,$
- Cliques are codewords of a very interesting error correcting code.

Codes made of cliques of constant size

Clique

Set of nodes that are all connected one to another.

Example: codewords = 4 nodes cliques



Symbols = edges

2 distinct nodes
$\Rightarrow d_{\min} = 6$ edges

Codes of cliques of size $c \ll n$

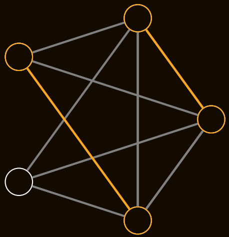
- $d_{\min} = 2(c-1) \approx 2c$, rate $r \approx \frac{c}{2}\binom{c}{2}^{-1}$
- $\Rightarrow F = r d_{\min} \approx 2$,
- Cliques are codewords of a very interesting error correcting code. . . and they are free!

Codes made of cliques of constant size

Clique

Set of nodes that are all connected one to another.

Example: codewords = 4 nodes cliques



Symbols = edges

2 distinct nodes
$\Rightarrow d_{\min} = 6$ edges

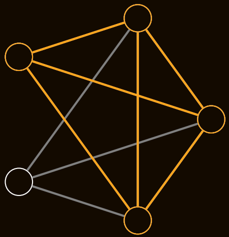
Codes of cliques of size $c \ll n$

- $d_{\min} = 2(c-1) \approx 2c$, rate $r \approx \frac{c}{2}\binom{c}{2}^{-1}$
- $\Rightarrow F = rd_{\min} \approx 2$,
- Cliques are codewords of a very interesting error correcting code. . . and they are free!

Clique

Set of nodes that are all connected one to another.

Example: codewords = 4 nodes cliques



Symbols = edges

2 distinct nodes
$\Rightarrow d_{\min} = 6$ edges

Codes of cliques of size $c \ll n$

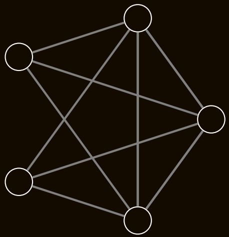
- $d_{\min} = 2(c-1) \approx 2c$, rate $r \approx \frac{c}{2}\binom{c}{2}^{-1}$
- $\Rightarrow F = rd_{\min} \approx 2$,
- Cliques are codewords of a very interesting error correcting code... and they are free!

Codes made of cliques of constant size

Clique

Set of nodes that are all connected one to another.

Example: codewords = 4 nodes cliques



Symbols = edges

2 distinct nodes
$\Rightarrow d_{\min} =$ 6 edges

Codes of cliques of size $c \ll n$

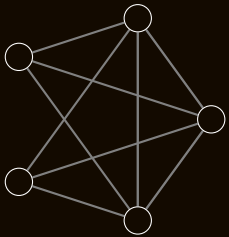
- $d_{\min} = 2(c-1) \approx 2c$, rate $r \approx \frac{c}{2}\binom{c}{2}^{-1}$
- $\Rightarrow F = r d_{\min} \approx 2$,
- Cliques are codewords of a very interesting error correcting code. . . and they are free!

Codes made of cliques of constant size

Clique

Set of nodes that are all connected one to another.

Example: codewords = 4 nodes cliques



Symbols = edges

2 distinct nodes $\Rightarrow d_{\min} = 6$ edges

Codes of cliques of size $c \ll n$

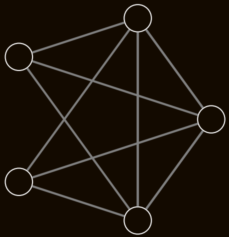
- $d_{\min} = 2(c-1) \approx 2c$, rate $r \approx \frac{c}{2}\binom{c}{2}^{-1}$
- $\Rightarrow F = rd_{\min} \approx 2$,
- Cliques are codewords of a very interesting error correcting code... and they are free!

Codes made of cliques of constant size

Clique

Set of nodes that are all connected one to another.

Example: codewords = 4 nodes cliques



Symbols = edges

2 distinct nodes
$\Rightarrow d_{\min} = 6$ edges

Codes of cliques of size $c \ll n$

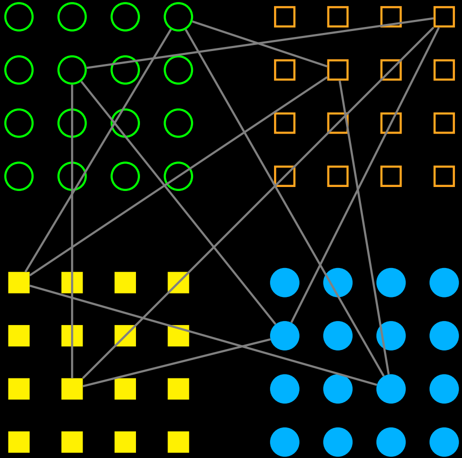
- $d_{\min} = 2(c-1) \approx 2c$, rate $r \approx \frac{c}{2}\binom{c}{2}^{-1}$
- $\Rightarrow F = rd_{\min} \approx 2$,
- Cliques are codewords of a very interesting error correcting code... and they are free!

Plan

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

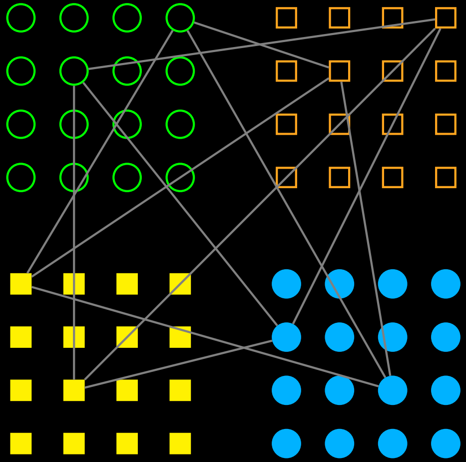
Our model: learning

- Example: $c = 4$ clusters made of $l = 16$ neurons each,
- $\underbrace{1000 = 8}_{j_1 \text{ in } c_1} \underbrace{0011 = 3}_{j_2 \text{ in } c_2} \underbrace{0010 = 2}_{j_3 \text{ in } c_3} \underbrace{1001 = 9}_{j_4 \text{ in } c_4}$,



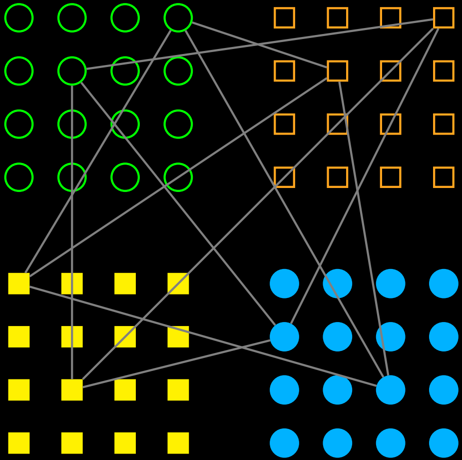
Our model: learning

- Example: $c = 4$ clusters made of $l = 16$ neurons each,
- $\underbrace{1000 = 8}_{j_1 \text{ in } c_1} \underbrace{0011 = 3}_{j_2 \text{ in } c_2} \underbrace{0010 = 2}_{j_3 \text{ in } c_3} \underbrace{1001 = 9}_{j_4 \text{ in } c_4}$,



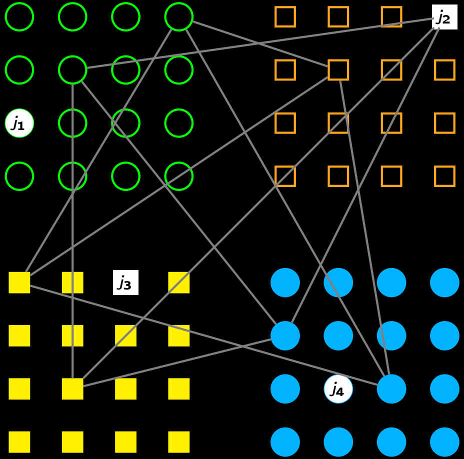
Our model: learning

- Example: $c = 4$ clusters made of $l = 16$ neurons each,
- $\underbrace{1000 = 8}_{j_1 \text{ in } c_1}\ \underbrace{0011 = 3}_{j_2 \text{ in } c_2}\ \underbrace{0010 = 2}_{j_3 \text{ in } c_3}\ \underbrace{1001 = 9}_{j_4 \text{ in } c_4}$,



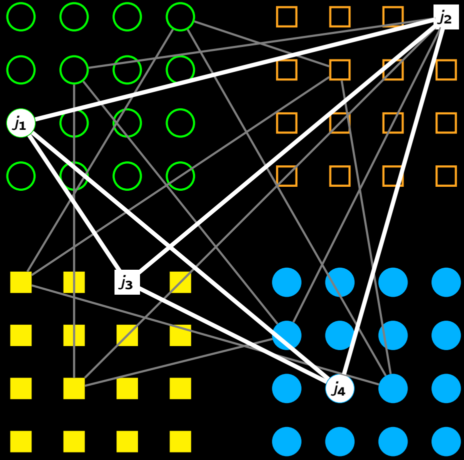
Our model: learning

- Example: $c = 4$ clusters made of $l = 16$ neurons each,
- $\underbrace{1000 = 8}_{j_1 \text{ in } c_1} \underbrace{0011 = 3}_{j_2 \text{ in } c_2} \underbrace{0010 = 2}_{j_3 \text{ in } c_3} \underbrace{1001 = 9}_{j_4 \text{ in } c_4}$,



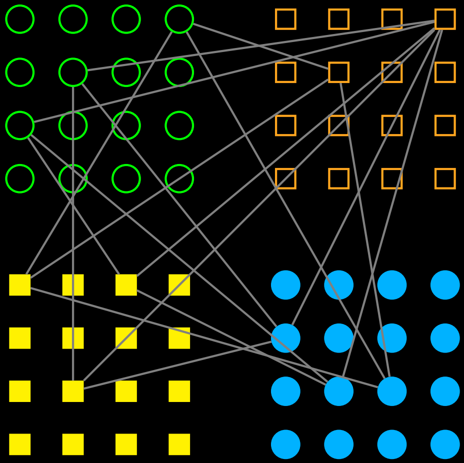
Our model: learning

- Example: $c = 4$ clusters made of $l = 16$ neurons each,
- $\underbrace{1000 = 8}_{j_1 \text{ in } c_1}$ $\underbrace{0011 = 3}_{j_2 \text{ in } c_2}$ $\underbrace{0010 = 2}_{j_3 \text{ in } c_3}$ $\underbrace{1001 = 9}_{j_4 \text{ in } c_4}$,



Our model: retrieving

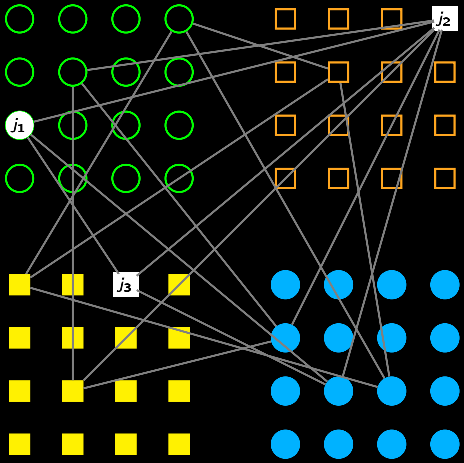
$\underbrace{1000}_{j_1 \text{ in } c_1}\ \underbrace{0011}_{j_2 \text{ in } c_2}\ \underbrace{0010}_{j_3 \text{ in } c_3}\ ????,$



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

Our model: retrieving

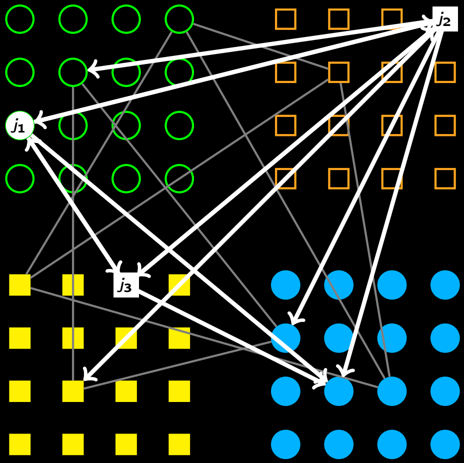
$$\underbrace{1000}_{j_1 \text{ in } c_1} \quad \underbrace{0011}_{j_2 \text{ in } c_2} \quad \underbrace{0010}_{j_3 \text{ in } c_3} \quad ????,$$



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

Our model: retrieving

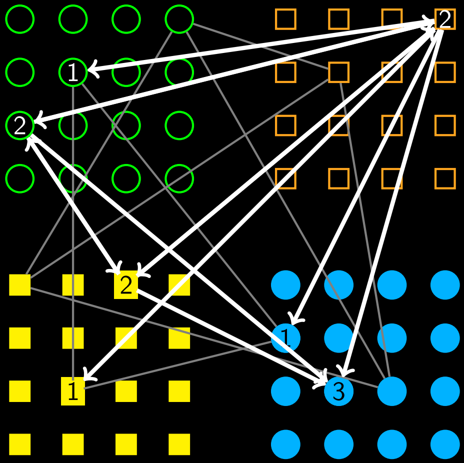
$\underbrace{1000}_{j_1 \text{ in } c_1}\ \underbrace{0011}_{j_2 \text{ in } c_2}\ \underbrace{0010}_{j_3 \text{ in } c_3}$????,



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

Our model: retrieving

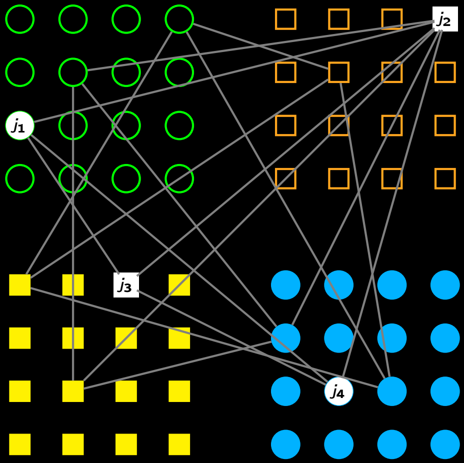
$$\underbrace{1000}_{j_1 \text{ in } c_1}\ \underbrace{0011}_{j_2 \text{ in } c_2}\ \underbrace{0010}_{j_3 \text{ in } c_3}\ ????,$$



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

Our model: retrieving

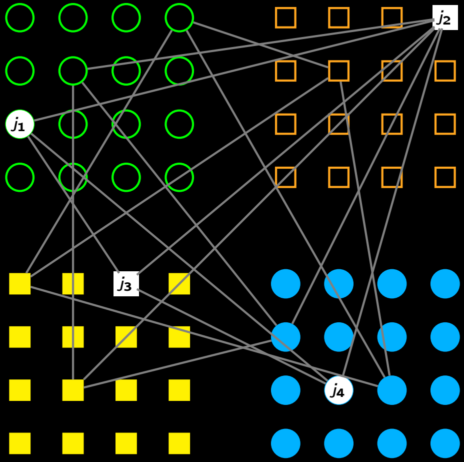
$$\underbrace{1000}_{j_1 \text{ in } c_1}\ \underbrace{0011}_{j_2 \text{ in } c_2}\ \underbrace{0010}_{j_3 \text{ in } c_3}\ \underbrace{1001}_{j_4 \text{ in } c_4},$$



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

Our model: retrieving

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



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

Density

Introducing a new parameter

- Density d is the ratio of the number of used connections to the total number of possible ones,
- If messages are i.i.d.: $d \approx 1 - \left(1 - \frac{1}{l^2}\right)^M$.

Curves

Remarks

- $d = 1$: no more distinction between learned and not learned messages,
- $d = f(l, M)$, not depending on c,
- $d \approx \frac{M}{l^2}$, for $M \ll l^2$.

Density

Introducing a new parameter

- Density d is the ratio of the number of used connections to the total number of possible ones,
- If messages are i.i.d.: $d \approx 1 - \left(1 - \frac{1}{l^2}\right)^M$.

Curves

Remarks

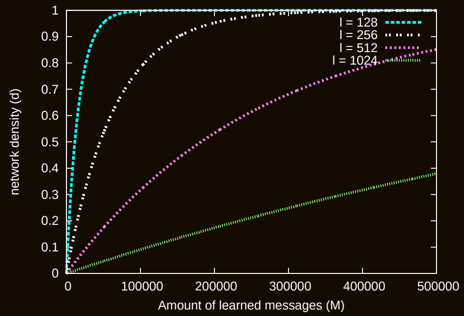
- $d = 1$: no more distinction between learned and not learned messages,
- $d = f(l, M)$, not depending on c,
- $d \approx \frac{M}{l^2}$, for $M \ll l^2$.

Density

Introducing a new parameter

- Density d is the ratio of the number of used connections to the total number of possible ones,
- If messages are i.i.d.: $d \approx 1 - \left(1 - \frac{1}{l^2}\right)^M$.

Curves



Remarks

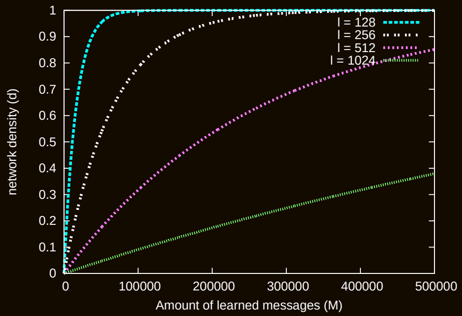
- $d = 1$: no more distinction between learned and not learned messages,
- $d = f(l, M)$, not depending on c,
- $d \approx \frac{M}{l^2}$, for $M \ll l^2$.

Density

Introducing a new parameter

- Density d is the ratio of the number of used connections to the total number of possible ones,
- If messages are i.i.d.: $d \approx 1 - \left(1 - \frac{1}{l^2}\right)^M$.

Curves



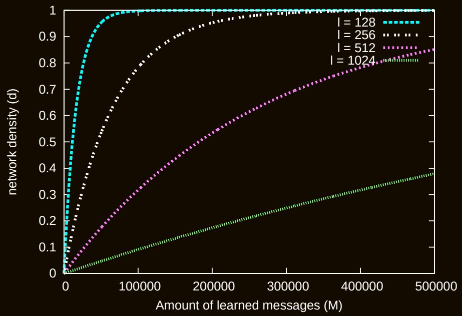
Remarks

- $d = 1$: no more distinction between learned and not learned messages,
- $d = f(l, M)$, not depending on c,
- $d \approx \frac{M}{l^2}$, for $M \ll l^2$.

Introducing a new parameter

- Density d is the ratio of the number of used connections to the total number of possible ones,
- If messages are i.i.d.: $d \approx 1 - \left(1 - \frac{1}{l^2}\right)^M$.

Curves



Remarks

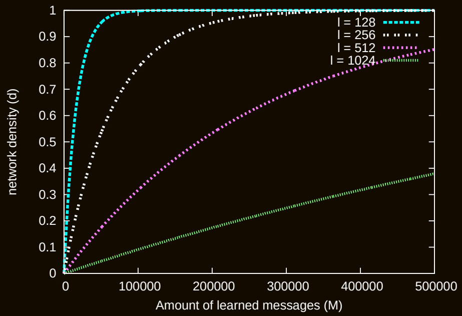
- $d = 1$: no more distinction between learned and not learned messages,
- $d = f(l, M)$, not depending on c,
- $d \approx \frac{M}{l^2}$, for $M \ll l^2$.

Density

Introducing a new parameter

- Density d is the ratio of the number of used connections to the total number of possible ones,
- If messages are i.i.d.: $d \approx 1 - \left(1 - \frac{1}{l^2}\right)^M$.

Curves

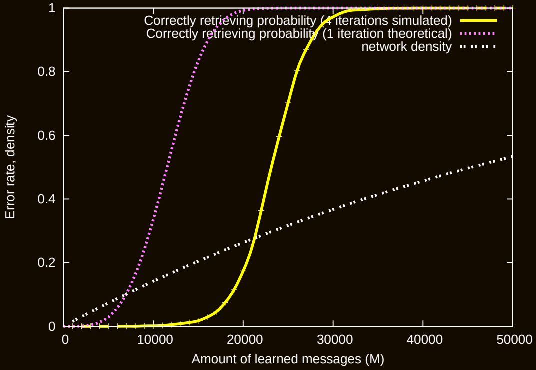


Remarks

- $d = 1$: no more distinction between learned and not learned messages,
- $d = f(l, M)$, not depending on c,
- $d \approx \frac{M}{l^2}$, for $M \ll l^2$.

Performance (1/3)

As an associative memory

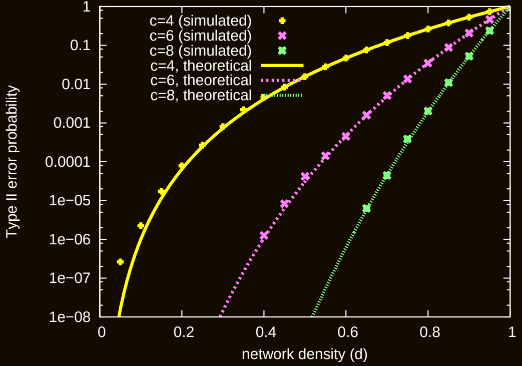


$c = 8$ clusters of $l = 256$ neurons each ($\sim$ messages of 64 bits),
Error probability when retrieving messages half erased.

Hopfield network ($n = 790$)

Our network

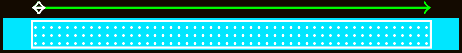
Set implementation



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

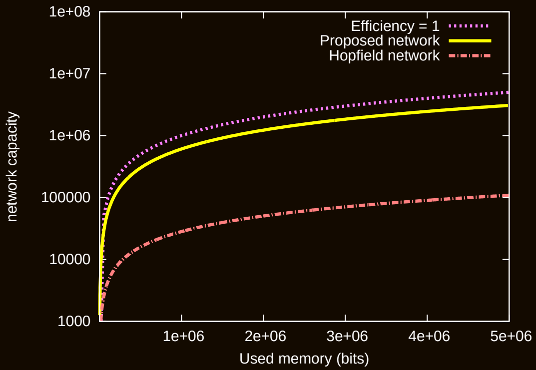
Hopfield network ($n = 740$)

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.

A word about plausibility

Analogies

Our network		Neuroscience litterature
Cliques of neurons	$\leftrightarrow$	Neural cliques
Local decoding	$\leftrightarrow$	Winner-take-all
Clusters	$\leftrightarrow$	Neocortical columns
Thrifty code	$\leftrightarrow$	Specific neurons

Limitations

- Necessity to provide a perfect - yet incomplete - content,
- Messages must not be correlated,
- Clusters must be large and few,
- Constant messages length,
- Systematic use of all clusters,
- Bidirectional connections and full inter-connectivity.

A word about plausibility

Analogies

Our network		Neuroscience litterature
Cliques of neurons	$\leftrightarrow$	Neural cliques
Local decoding	$\leftrightarrow$	Winner-take-all
Clusters	$\leftrightarrow$	Neocortical columns
Thrifty code	$\leftrightarrow$	Specific neurons

Limitations

- Necessity to provide a perfect - yet incomplete - content,
- Messages must not be correlated,
- Clusters must be large and few,
- Constant messages length,
- Systematic use of all clusters.
- Bidirectional connections and full inter-connectivity.

A word about plausibility

Analogies

Our network		Neuroscience litterature
Cliques of neurons	$\leftrightarrow$	Neural cliques
Local decoding	$\leftrightarrow$	Winner-take-all
Clusters	$\leftrightarrow$	Neocortical columns
Thrifty code	$\leftrightarrow$	Specific neurons

Limitations

- Necessity to provide a perfect - yet incomplete - content,
- Messages must not be correlated,
- Clusters must be large and few,
- Constant messages length,
- Systematic use of all clusters.
- Bidirectional connections and full inter-connectivity.

A word about plausibility

Analogies

Our network		Neuroscience litterature
Cliques of neurons	$\leftrightarrow$	Neural cliques
Local decoding	$\leftrightarrow$	Winner-take-all
Clusters	$\leftrightarrow$	Neocortical columns
Thrifty code	$\leftrightarrow$	Specific neurons

Limitations

- Necessity to provide a perfect - yet incomplete - content,
- Messages must not be correlated,
- Clusters must be large and few,
- Constant messages length,
- Systematic use of all clusters.
- Bidirectional connections and full inter-connectivity.

A word about plausibility

Analogies

Our network		Neuroscience litterature
Cliques of neurons	$\leftrightarrow$	Neural cliques
Local decoding	$\leftrightarrow$	Winner-take-all
Clusters	$\leftrightarrow$	Neocortical columns
Thrifty code	$\leftrightarrow$	Specific neurons

Limitations

- Necessity to provide a perfect - yet incomplete - content,
- Messages must not be correlated,
- Clusters must be large and few,
- Constant messages length,
- Systematic use of all clusters.
- Bidirectional connections and full inter-connectivity.

A word about plausibility

Analogies

Our network		Neuroscience litterature
Cliques of neurons	$\leftrightarrow$	Neural cliques
Local decoding	$\leftrightarrow$	Winner-take-all
Clusters	$\leftrightarrow$	Neocortical columns
Thrifty code	$\leftrightarrow$	Specific neurons

Limitations

- Necessity to provide a perfect - yet incomplete - content,
- Messages must not be correlated,
- Clusters must be large and few,
- Constant messages length,
- Systematic use of all clusters.
- Bidirectional connections and full inter-connectivity.

A word about plausibility

Analogies

Our network		Neuroscience litterature
Cliques of neurons	$\leftrightarrow$	Neural cliques
Local decoding	$\leftrightarrow$	Winner-take-all
Clusters	$\leftrightarrow$	Neocortical columns
Thrifty code	$\leftrightarrow$	Specific neurons

Limitations

- Necessity to provide a perfect - yet incomplete - content,
- Messages must not be correlated,
- Clusters must be large and few,
- Constant messages length,
- Systematic use of all clusters.
- Bidirectional connections and full inter-connectivity.

Plan

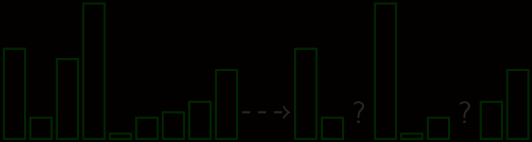
1. Associative memories and error correcting codes
   - Associative memory
   - Error correcting codes
   - Code of cliques
2. Sparse networks, principles and performance
   - Learning
   - Retrieving
   - Performance
3. Developments
   - Blurred messages
   - Correlated sources
   - Sparse messages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

Blurred messages

Limitation

Partial messages must contain perfect information.

Noise model



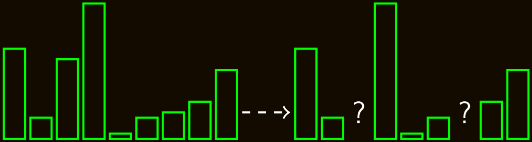
Soft decoding

Blurred messages

Limitation

Partial messages must contain perfect information.

Noise model



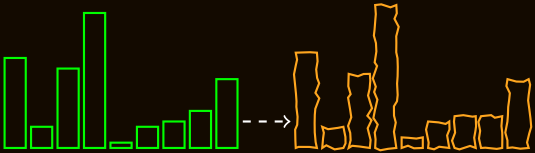
Soft decoding

Blurred messages

Limitation

Partial messages must contain perfect information.

Noise model



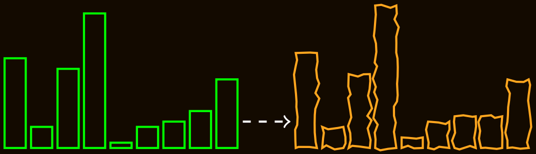
Soft decoding

Blurred messages

Limitation

Partial messages must contain perfect information.

Noise model



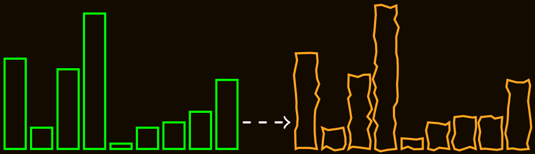
Soft decoding

Blurred messages

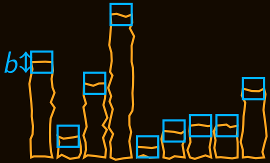
Limitation

Partial messages must contain perfect information.

Noise model



Soft decoding

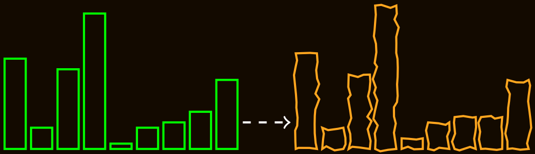


Blurred messages

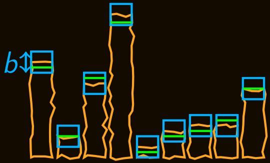
Limitation

Partial messages must contain perfect information.

Noise model

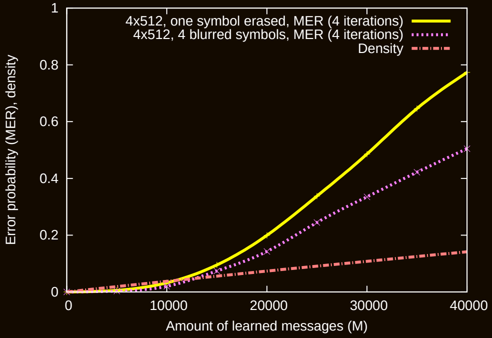


Soft decoding



Performance

Simulations



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

Why performance are better?

- Erasing: $\searrow$ competitive cliques ($\approx l$) $\nearrow$ probability ($\approx d^{c-1}$),
- Bruit : $\nearrow$ competitive cliques ($\approx b^c$) $\searrow$ probability ($\approx d^{\frac{c(c-1)}{2}}$).

Correlated messages

Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

- There are two effects of correlation:
  - An inescapable effect: *brain* and *train* are learned $\rightarrow$ $*$*rain* ?
  - Another effect coming from our network:

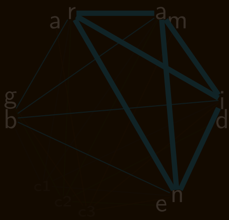
Correlated messages

Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

- There are two effects of correlation:
  - An inescapable effect: *brain* and *train* are learned $\rightarrow$ $*$*rain* ?
  - Another effect coming from our network:



Correlated messages

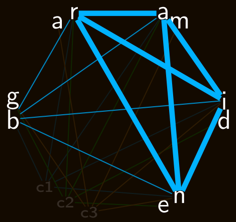
Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

- There are two effects of correlation:
  - An inescapable effect: *brain* and *train* are learned $\rightarrow$ $*$*rain* ?
  - Another effect coming from our network:

brain



Correlated messages

Limitation

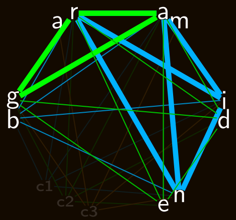
With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

- There are two effects of correlation:
  - An inescapable effect: *brain* and *train* are learned $\rightarrow$ $*$*rain* ?
  - Another effect coming from our network:

brain

grade

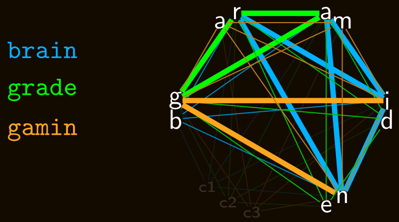


Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

- There are two effects of correlation:
  - An inescapable effect: *brain* and *train* are learned $\rightarrow$ $*$*rain* ?
  - Another effect coming from our network:



Correlated messages

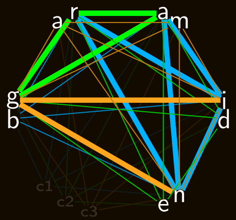
Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

- There are two effects of correlation:
  - An inescapable effect: *brain* and *train* are learned $\rightarrow$ $*$*rain* ?
  - Another effect coming from our network:

brain
grade
gamin
grain



Correlated messages

Limitation

With correlations grows the number of Type II errors.

Fighting correlation by adding random redundancy

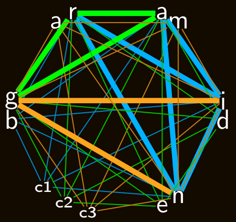
- There are two effects of correlation:
  - An inescapable effect: *brain* and *train* are learned $\rightarrow$ $*$*rain* ?
  - Another effect coming from our network:

brain +c1

grade +c2

gamin +c3

grain +c?



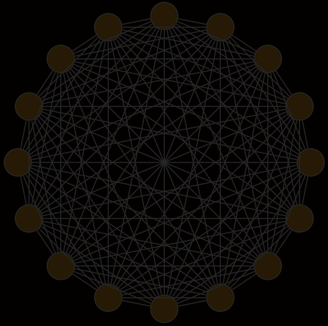
Towards a fourth level of sparsity

Limitations

- Clusters must be large and few,
- Learned 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.

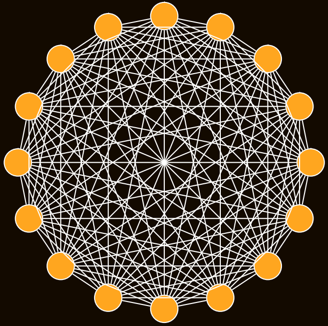
Towards a fourth level of sparsity

Limitations

- Clusters must be large and few,
- Learned 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

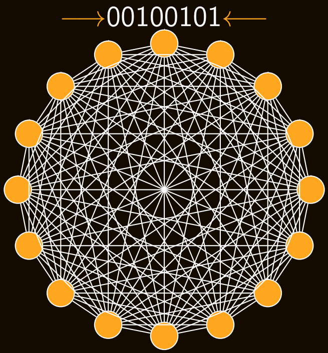
- Global winner-take-all.

Towards a fourth level of sparsity

Limitations

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

Illustration



Idea

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

Solution

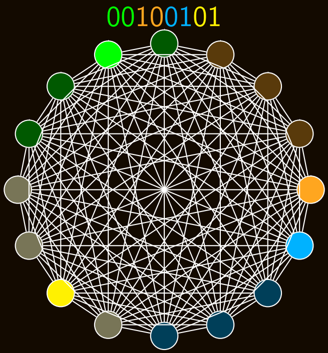
- Global winner-take-all.

Towards a fourth level of sparsity

Limitations

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

Illustration



Idea

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

Solution

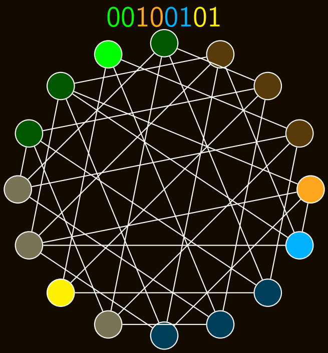
- Global winner-take-all.

Towards a fourth level of sparsity

Limitations

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

Illustration



Idea

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

Solution

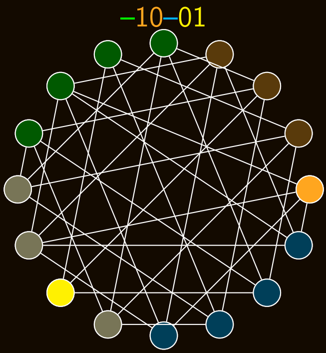
- Global winner-take-all.

Towards a fourth level of sparsity

Limitations

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

Illustration



Idea

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

Solution

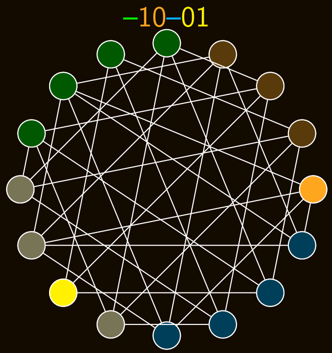
- Global winner-take-all.

Towards a fourth level of sparsity

Limitations

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

Illustration



Idea

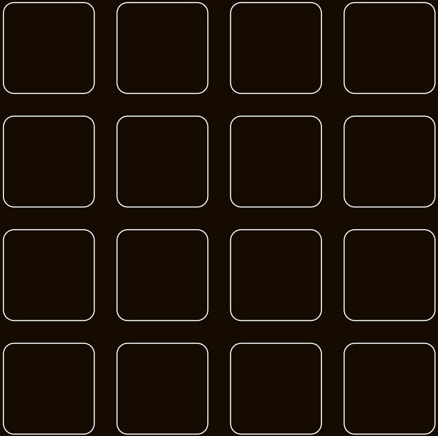
1. Shorter messages,
2. Clusters and thrifty codes,
3. Sparse network,
4. Sparse messages.

Solution

- Global winner-take-all.

Global winner-take-all

Illustration



Idea

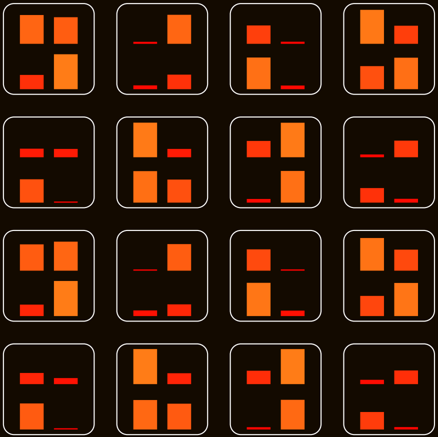
- After global message passing...
- After local maximum selections...
- Global maximum selection.

Interests

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

Global winner-take-all

Illustration



Idea

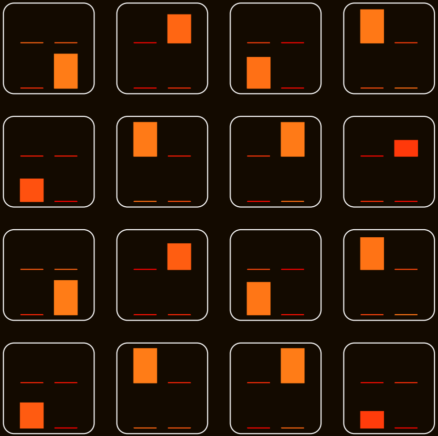
- After global message passing. . .
- After local maximum selections. . .
- Global maximum selection.

Interests

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

Global winner-take-all

Illustration



Idea

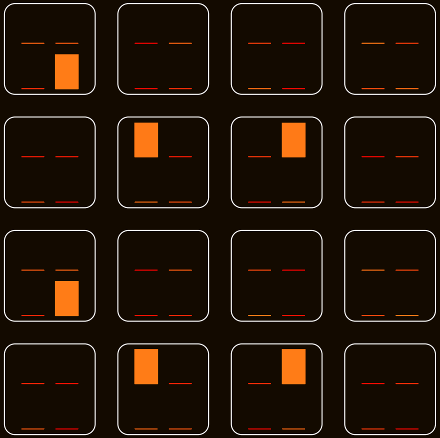
- After global message passing. . .
- After local maximum selections. . .
- Global maximum selection.

Interests

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

Global winner-take-all

Illustration



Idea

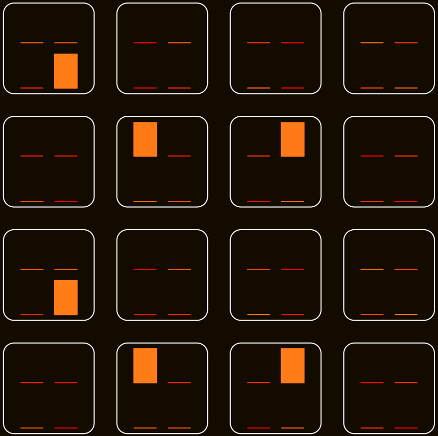
- After global message passing. . .
- After local maximum selections. . .
- Global maximum selection.

Interests

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

Global winner-take-all

Illustration



Idea

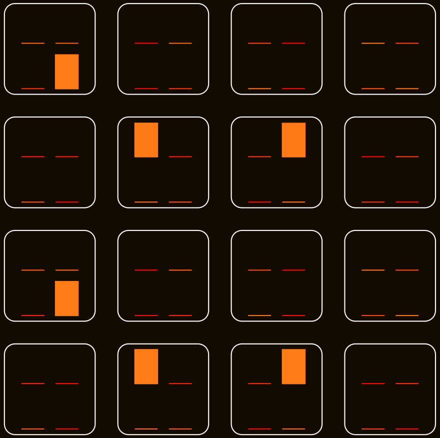
- After global message passing. . .
- After local maximum selections. . .
- Global maximum selection.

Interests

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

Global winner-take-all

Illustration



Idea

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

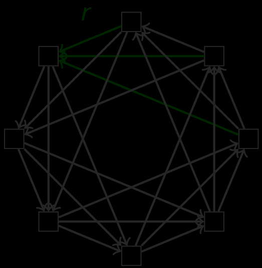
Interests

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

Tournament chains and unidirectional connections

Problem

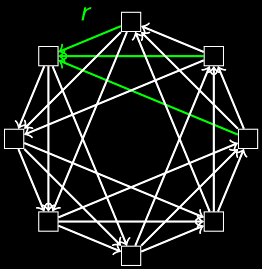
Bidirectional connections and full inter-connectivity.



Tournament chains and unidirectional connections

Problem

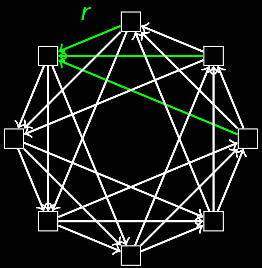
Bidirectional connections and full inter-connectivity.

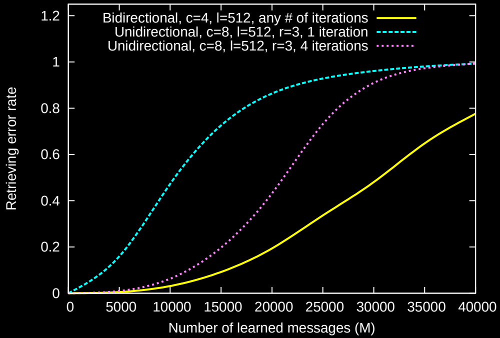


Tournament chains and unidirectional connections

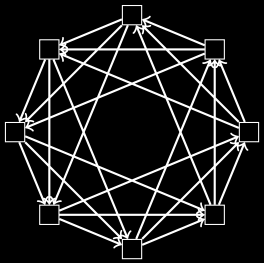
Problem

Bidirectional connections and full inter-connectivity.





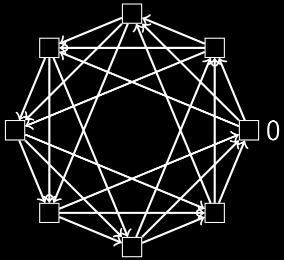
Learning arbitrarily long sequences



Performance

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

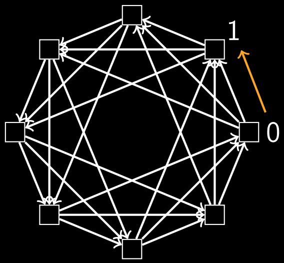
Learning arbitrarily long sequences



Performance

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

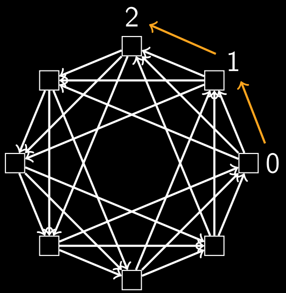
Learning arbitrarily long sequences



Performance

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

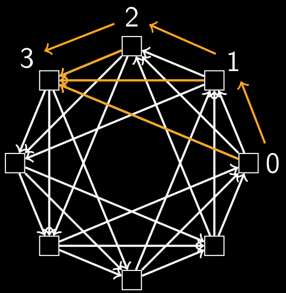
Learning arbitrarily long sequences



Performance

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

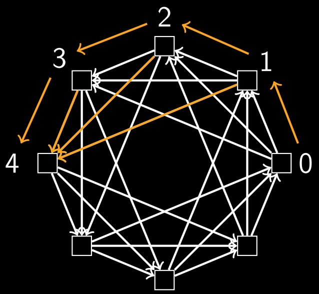
Learning arbitrarily long sequences



Performance

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

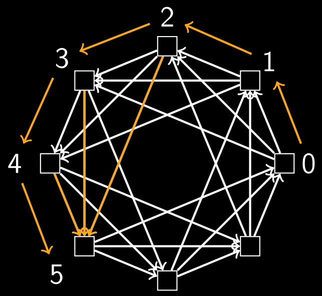
Learning arbitrarily long sequences



Performance

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

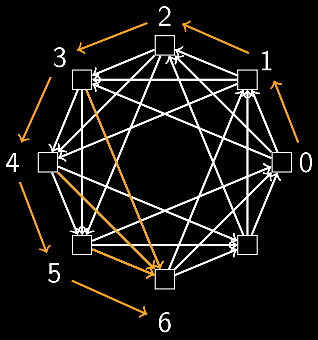
Learning arbitrarily long sequences



Performance

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

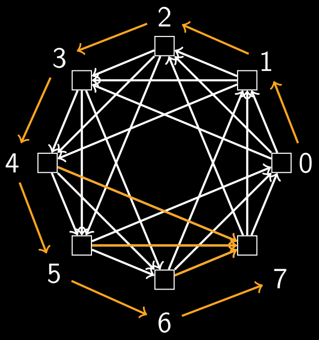
Learning arbitrarily long sequences



Performance

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

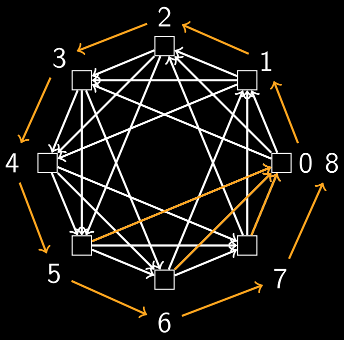
Learning arbitrarily long sequences



Performance

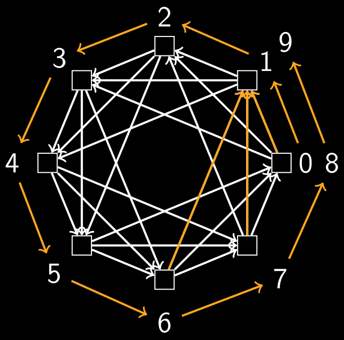
- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in messages,
- $m = 1823$ learned messages,
- $P_e \leq 0.01$.

Learning arbitrarily long sequences



Performance

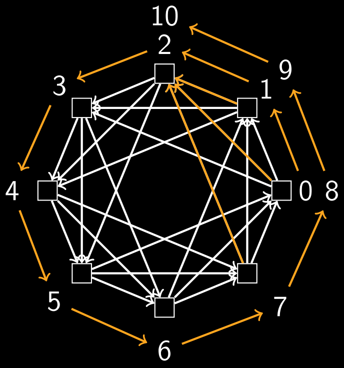
- $c = 50$ clusters,
- $l = 256$ neurons/cluster,
- $L = 1000$ symbols in messages,
- $m = 1823$ learned messages,
- $P_e \leq 0.01$.



Performance

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

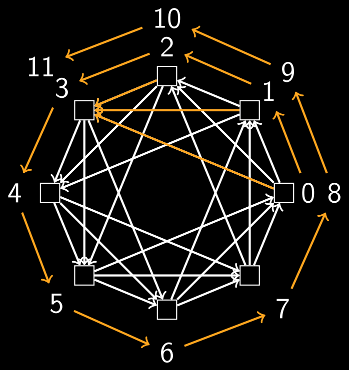
Learning arbitrarily long sequences



Performance

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

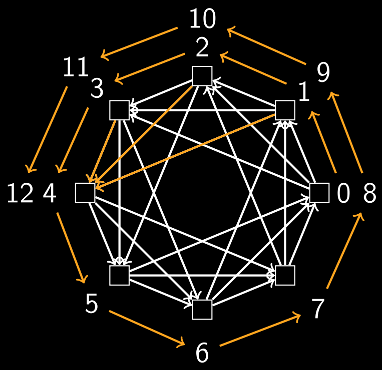
Learning arbitrarily long sequences



Performance

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

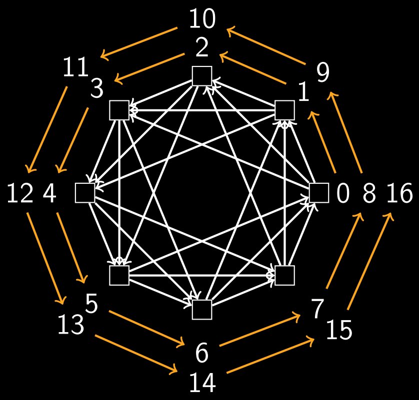
Learning arbitrarily long sequences



Performance

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

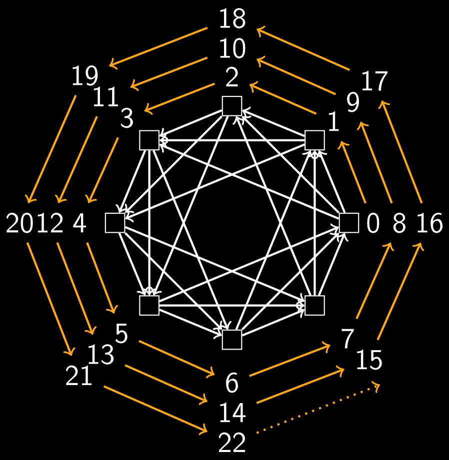
Learning arbitrarily long sequences



Performance

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

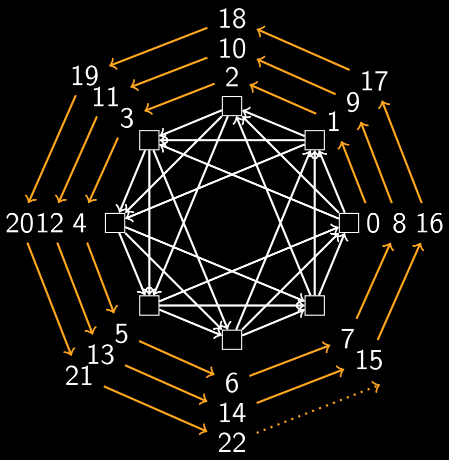
Learning arbitrarily long sequences



Performance

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

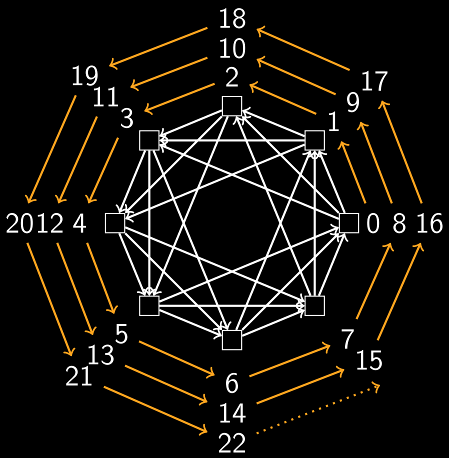
Learning arbitrarily long sequences



Performance

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

Learning arbitrarily long sequences



Performance

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

Plan

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

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, synchronization. . . ,
- 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, synchronization. . . ,
- 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, synchronization. . . ,
- 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, synchronization. . . ,
- 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, synchronization. . . ,
- 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, synchronization. . . ,
- Degrees of freedom: inhibitions, time, weights,
- No trade off required between performance and plausibility.

Conclusion

Approach

Designing an associative memory → neocortical architecture

sparsity

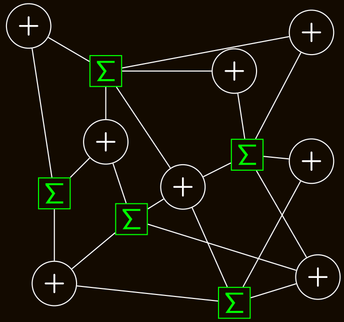
distributed codes

Results

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

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

LDPC decoder



Neocortical "decoder"