

Réseaux de cliques neurales

Vincent Gripon

Télécom Bretagne, Lab-STICC

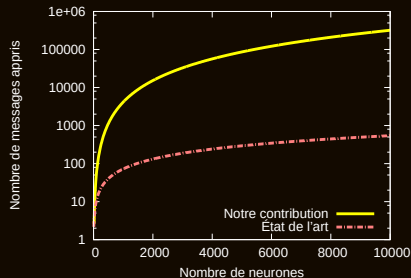
École doctorale SICMA

20 juillet 2011

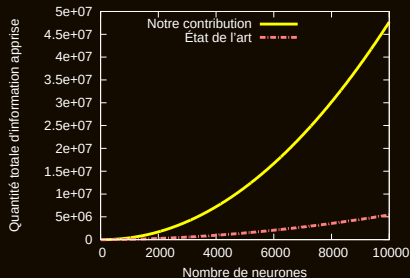
En un mot...

Apprentissage de messages dans des réseaux de neurones récurrents

Diversité d'apprentissage



Capacité d'apprentissage



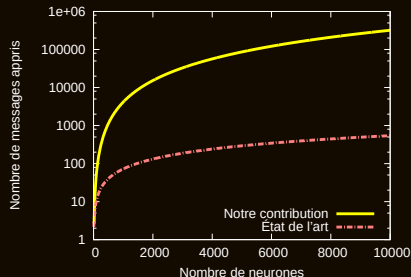
Notre contribution

État de l'art

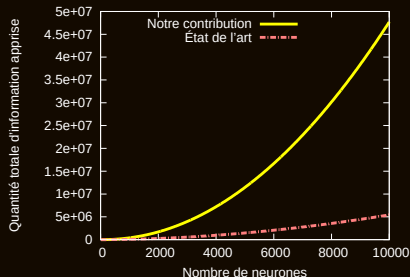
En un mot...

Apprentissage de messages dans des réseaux de neurones récurrents

Diversité d'apprentissage



Capacité d'apprentissage

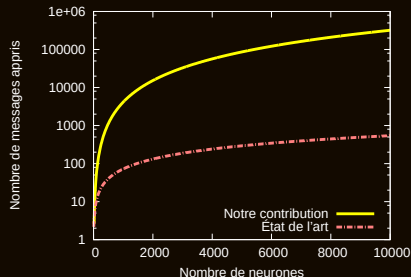


Notre contribution
parcimonie
codage correcteur
État de l'art

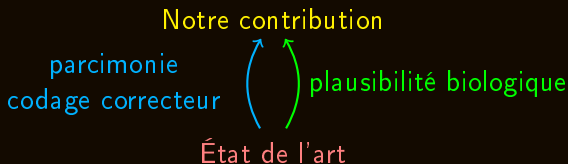
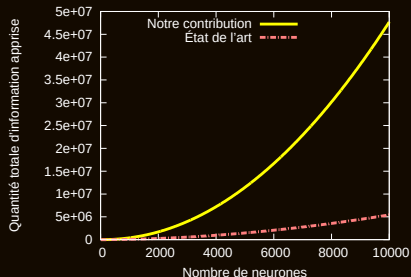
En un mot...

Apprentissage de messages dans des réseaux de neurones récurrents

Diversité d'apprentissage

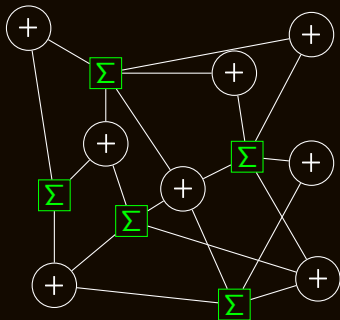


Capacité d'apprentissage



Idée de départ

Décodeur *LDPC*



“Décodeur” néocortical



noeuds Σ = neurones
décodage = remémoration
parité = ?
? = apprentissage

- 1 Mémoires associatives et codage correcteur
 - Mémoires associatives
 - Codage correcteur
 - Codes sur cliques
- 2 Réseau parcimonieux, principe et performances
 - Apprentissage
 - Remémoration
 - Performances
- 3 Développements
 - Messages flous
 - Sources corrélées
 - Messages parcimonieux
 - *Winner-take-all* global
 - Synchronisation temporelle
- 4 Conclusion, ouvertures

- 1 Mémoires associatives et codage correcteur
 - Mémoires associatives
 - Codage correcteur
 - Codes sur cliques
- 2 Réseau parcimonieux, principe et performances
 - Apprentissage
 - Remémoration
 - Performances
- 3 Développements
 - Messages flous
 - Sources corrélées
 - Messages parcimonieux
 - *Winner-take-all* global
 - Synchronisation temporelle
- 4 Conclusion, ouvertures

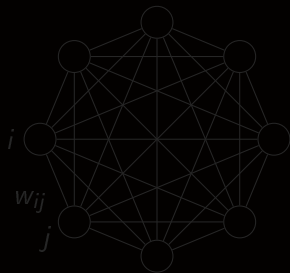
Mémoires associatives et modèle de Hopfield

Mémoire associative

Deux opérations :

- **Apprendre** des messages,
- **Retrouver** un messages appris effacé ou erroné.

L'état de l'art : le modèle de Hopfield



- Apprendre : M messages binaires $\mathbf{d}^m$:

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

- Retrouver : itérer

$$\forall i, v_i \leftarrow \operatorname{sgn}\left(\sum_{j=1}^N v_j w_{ij}\right).$$

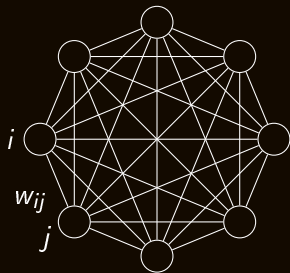
Mémoires associatives et modèle de Hopfield

Mémoire associative

Deux opérations :

- Apprendre des messages,
- Retrouver un messages appris effacé ou erroné.

L'état de l'art : le modèle de Hopfield



- Apprendre : M messages binaires $\mathbf{d}^m$:

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

- Retrouver : itérer
 $\forall i, v_i \leftarrow \text{sgn}\left(\sum_{j \neq i} v_j w_{ij}\right).$

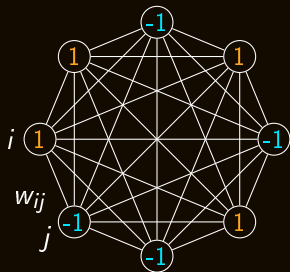
Mémoires associatives et modèle de Hopfield

Mémoire associative

Deux opérations :

- Apprendre des messages,
- Retrouver un messages appris effacé ou erroné.

L'état de l'art : le modèle de Hopfield



- Apprendre : M messages binaires $\mathbf{d}^m$:

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

- Retrouver : itérer

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

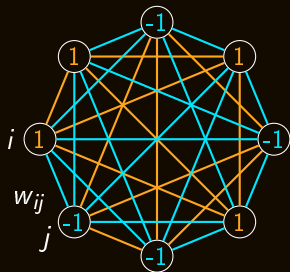
Mémoires associatives et modèle de Hopfield

Mémoire associative

Deux opérations :

- Apprendre des messages,
- Retrouver un messages appris effacé ou erroné.

L'état de l'art : le modèle de Hopfield



- Apprendre : M messages binaires $\mathbf{d}^m$:

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

- Retrouver : itérer
 $\forall i, v_i \leftarrow \text{sgn}\left(\sum_{j \neq i} v_j w_{ij}\right).$

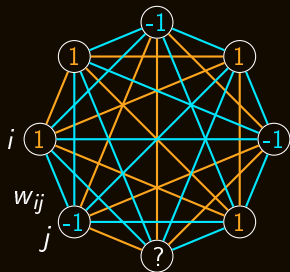
Mémoires associatives et modèle de Hopfield

Mémoire associative

Deux opérations :

- Apprendre des messages,
- Retrouver un messages appris effacé ou erroné.

L'état de l'art : le modèle de Hopfield



- Apprendre : M messages binaires $\mathbf{d}^m$:

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

- Retrouver : itérer
 $\forall i, v_i \leftarrow \text{sgn}\left(\sum_{j \neq i} v_j w_{ij}\right).$

Réseaux de Hopfield (n neurones $\longleftrightarrow$)

- **Diversité** : $M = \frac{n}{2\log(n)}$, $\leftrightarrow$
- **Capacité** : $\frac{n^2}{2\log(n)}$, 
- Quantité d'information binaire requise : $\binom{n}{2} \log_2(M+1)$, 
- $\Rightarrow$ **Efficacité** $\approx \frac{1}{\log(n)\log_2(M+1)}$, 
- Connexions sensibles, longueur des messages = taille du réseau, les messages et leurs opposés sont appris en même temps...

Exemple avec $n = 790$: 

Réseaux de Hopfield (n neurones $\longleftrightarrow$)

- **Diversité** : $M = \frac{n}{2\log(n)}$, $\leftrightarrow$
- **Capacité** : $\frac{n^2}{2\log(n)}$,  = 
- Quantité d'information binaire requise : $\binom{n}{2} \log_2(M + 1)$, 
- $\Rightarrow$ **Efficacité** $\approx \frac{1}{\log(n) \log_2(M+1)}$, 
- Connexions sensibles, longueur des messages = taille du réseau, les messages et leurs opposés sont appris en même temps...

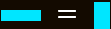
Exemple avec $n = 790$: 

Réseaux de Hopfield (n neurones $\longleftrightarrow$)

- **Diversité** : $M = \frac{n}{2\log(n)}$, $\longleftrightarrow$
- **Capacité** : $\frac{n^2}{2\log(n)}$,  = 
- Quantité d'information binaire requise : $\binom{n}{2} \log_2(M + 1)$, 
- $\Rightarrow$ **Efficacité** $\approx \frac{1}{\log(n) \log_2(M+1)}$, 
- Connexions sensibles, longueur des messages = taille du réseau, les messages et leurs opposés sont appris en même temps...

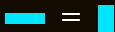
Exemple avec $n = 790$: 

Réseaux de Hopfield (n neurones $\longleftrightarrow$)

- **Diversité** : $M = \frac{n}{2\log(n)}$, $\leftrightarrow$
- **Capacité** : $\frac{n^2}{2\log(n)}$,  = 
- Quantité d'information binaire requise : $\binom{n}{2} \log_2(M + 1)$, 
- $\Rightarrow$ **Efficacité** $\approx \frac{1}{\log(n) \log_2(M+1)}$. 
- Connexions sensibles, longueur des messages = taille du réseau, les messages et leurs opposés sont appris en même temps...

Exemple avec $n = 790$: 

Réseaux de Hopfield (n neurones $\longleftrightarrow$)

- **Diversité** : $M = \frac{n}{2\log(n)}$, $\leftrightarrow$
- **Capacité** : $\frac{n^2}{2\log(n)}$,  = 
- Quantité d'information binaire requise : $\binom{n}{2} \log_2(M + 1)$, 
- $\Rightarrow$ **Efficacité** $\approx \frac{1}{\log(n) \log_2(M+1)}$. 
- Connexions sensibles, longueur des messages = taille du réseau, les messages et leurs opposés sont appris en même temps...

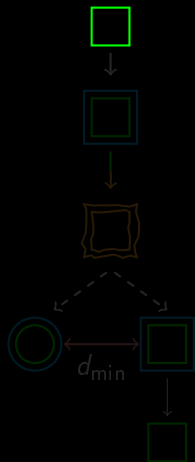
Exemple avec $n = 790$: 

Réseaux de Hopfield (n neurones $\longleftrightarrow$)

- **Diversité** : $M = \frac{n}{2\log(n)}$, $\longleftrightarrow$
- **Capacité** : $\frac{n^2}{2\log(n)}$,  = 
- Quantité d'information binaire requise : $\binom{n}{2} \log_2(M + 1)$, 
- $\Rightarrow$ **Efficacité** $\approx \frac{1}{\log(n)\log_2(M+1)}$. 
- Connexions sensibles, longueur des messages = taille du réseau, les messages et leurs opposés sont appris en même temps...

Exemple avec $n = 790$: 

Codes correcteur d'erreurs

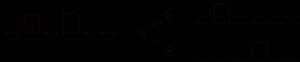


Exemple : le code économe

- Code des mots binaires à un seul "1" :



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

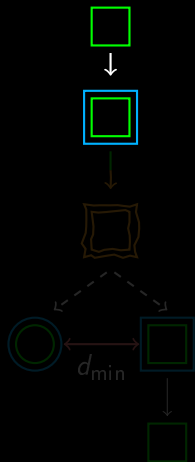


- Mais facile à decoder et minimise l'énergie :



- Ces codes peuvent être associés à la manière des codes distribués...

Codes correcteur d'erreurs

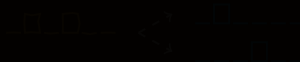


Exemple : le code économe

- Code des mots binaires à un seul "1" :



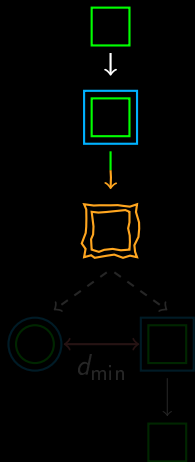
- Handicap : $d_{\min} = 2$:



- Mais facile à decoder et minimise l'énergie :



- Ces codes peuvent être associés à la manière des codes distribués...

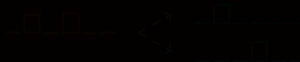


Exemple : le code économe

- Code des mots binaires à un seul "1" :



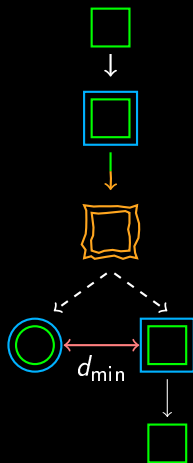
- Handicap : $d_{\min} = 2$:



- Mais facile à decoder et minimise l'énergie :

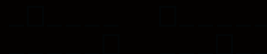


- Ces codes peuvent être associés à la manière des codes distribués...

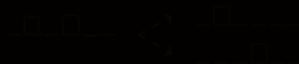


Exemple : le code économe

- Code des mots binaires à un seul "1" :



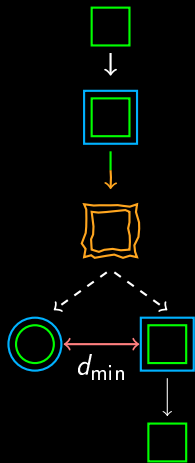
- Handicap : $d_{\min} = 2$:



- Mais facile à décoder et minimise l'énergie :

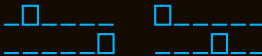


- Ces codes peuvent être associés à la manière des codes distribués...



Exemple : le code économe

- Code des mots binaires à un seul "1" :



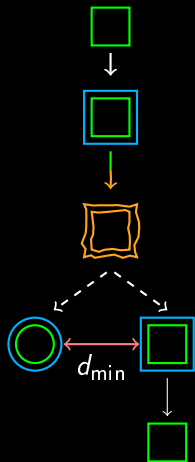
- Handicap : $d_{\min} = 2$:



- Mais facile à décoder et minimise l'énergie :



- Ces codes peuvent être associés à la manière des codes distribués...



Exemple : le code économe

- Code des mots binaires à un seul "1" :



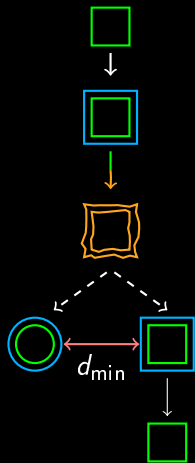
- Handicap : $d_{\min} = 2$:



- Mais facile à décoder et minimise l'énergie :

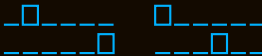


- Ces codes peuvent être associés à la manière des codes distribués...



Exemple : le code économe

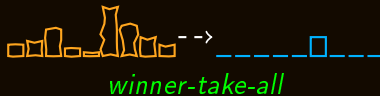
- Code des mots binaires à un seul "1" :



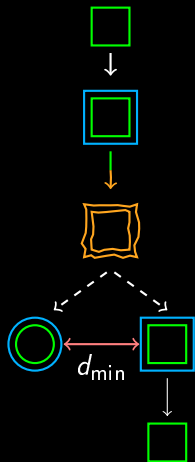
- Handicap : $d_{\min} = 2$:



- Mais facile à décoder et minimise l'énergie :

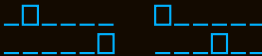


- Ces codes peuvent être associés à la manière des codes distribués...



Exemple : le code économe

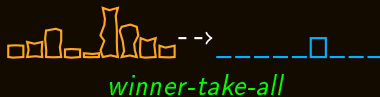
- Code des mots binaires à un seul "1" :



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



- Mais facile à décoder et minimise l'énergie :



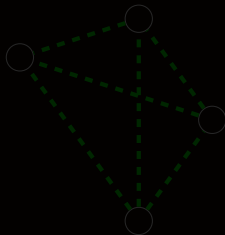
- Ces codes peuvent être associés à la manière des codes distribués...

Des codes sur des cliques de taille constante

Exemple : mots de codes = cliques à 4 noeuds

Clique

Ensemble de noeuds d'un graphe entièrement interconnectés.



2 noeuds différents
 $\Rightarrow d_{\min} = 6$ arêtes

Codes sur cliques de taille $c \ll n$

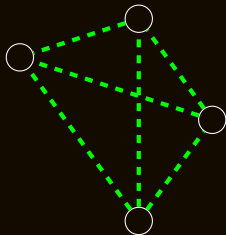
- $d_{\min} = 2(c - 1) \approx 2c$
- $\Rightarrow R = rd_{\min} \approx 2r$
- Les cliques sont les mots de code d'un code correcteur d'erreurs très intéressant

Des codes sur des cliques de taille constante

Clique

Ensemble de
nœuds d'un
graphe
entièrement
interconnectés.

Exemple : mots de codes = cliques à 4 nœuds



Symboles = arêtes

2 nœuds différents
 $\Rightarrow d_{\min} = 6$ arêtes

Codes sur cliques de taille $c \ll n$

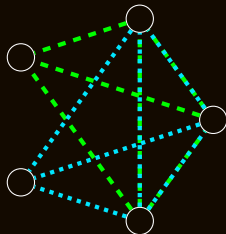
- $d_{\min} = 2(c - 1) \approx 2c$
- $\Rightarrow R = rd_{\min} \approx 2$
- Les cliques sont les mots de code d'un code correcteur d'erreurs très intéressant.

Des codes sur des cliques de taille constante

Clique

Ensemble de
nœuds d'un
graphe
entièrement
interconnectés.

Exemple : mots de codes = cliques à 4 nœuds



Symboles = arêtes

2 nœuds différents
 $\Rightarrow d_{\min} = 6$ arêtes

Codes sur cliques de taille $c \ll n$

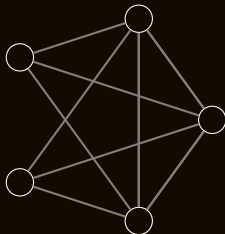
- $d_{\min} = 2(c - 1) \approx 2c$
- $\Rightarrow R = rd_{\min} \approx 2$
- Les cliques sont les mots de code d'un code correcteur d'erreurs très intéressant

Des codes sur des cliques de taille constante

Exemple : mots de codes = cliques à 4 noeuds

Clique

Ensemble de noeuds d'un graphe entièrement interconnectés.



Symboles = arêtes

2 noeuds différents
 $\Rightarrow d_{\min} = 6$ arêtes

Codes sur cliques de taille $c \ll n$

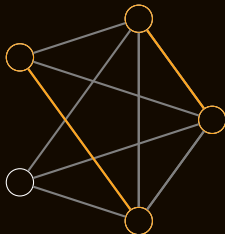
- $d_{\min} = 2(c - 1) \approx 2c$, rendement $r \approx \frac{c}{2}$
- $\Rightarrow F = rd_{\min} \approx 2$,
- Les cliques sont les mots de code d'un code correcteur d'erreurs très intéressant... à étudier.

Des codes sur des cliques de taille constante

Clique

Ensemble de
nœuds d'un
graphe
entièrement
interconnectés.

Exemple : mots de codes = cliques à 4 nœuds



Symboles = arêtes

2 nœuds différents
 $\Rightarrow d_{\min} = 6$ arêtes

Codes sur cliques de taille $c \ll n$

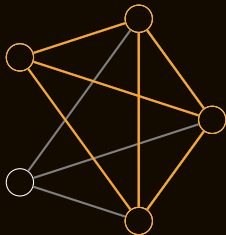
- $d_{\min} = 2(c - 1) \approx 2c$, rendement $r \approx \frac{c}{2} \binom{c}{2}^{-1}$
- $\Rightarrow F = rd_{\min} \approx 2$,
- Les cliques sont les mots de code d'un code correcteur d'erreurs très intéressant... (à étudier)

Des codes sur des cliques de taille constante

Clique

Ensemble de
nœuds d'un
graphe
entièrement
interconnectés.

Exemple : mots de codes = cliques à 4 nœuds



Symboles = arêtes

2 nœuds différents
 $\Rightarrow d_{\min} = 6$ arêtes

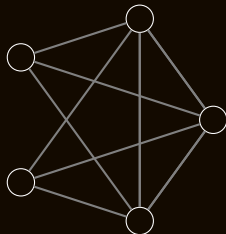
Codes sur cliques de taille $c \ll n$

- $d_{\min} = 2(c - 1) \approx 2c$, rendement $r \approx \frac{c}{2} \binom{c}{2}^{-1}$
- $\Rightarrow F = rd_{\min} \approx 2$,
- Les cliques sont les mots de code d'un code correcteur d'erreurs très intéressant... (à étudier)

Des codes sur des cliques de taille constante

Clique

Ensemble de
nœuds d'un
graphe
entièrement
interconnectés.



Exemple : mots de codes = cliques à 4 nœuds

Symboles = arêtes

2 nœuds différents
 $\Rightarrow d_{\min} = 6$ arêtes

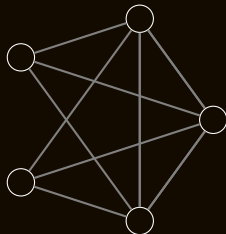
Codes sur cliques de taille $c \ll n$

- $d_{\min} = 2(c - 1) \approx 2c$, rendement $r \approx \frac{c}{2} \binom{c}{2}^{-1}$
- $\Rightarrow F = rd_{\min} \approx 2$,
- Les cliques sont les mots de code d'un code correcteur d'erreurs très intéressant...

Des codes sur des cliques de taille constante

Clique

Ensemble de noeuds d'un graphe entièrement interconnectés.



Exemple : mots de codes = cliques à 4 noeuds

Symboles = arêtes

2 noeuds différents
 $\Rightarrow d_{\min} = 6$ arêtes

Codes sur cliques de taille $c \ll n$

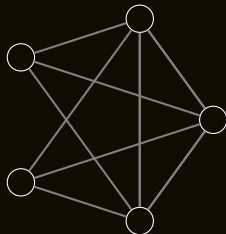
- $d_{\min} = 2(c - 1) \approx 2c$, rendement $r \approx \frac{c}{2} \binom{c}{2}^{-1}$
- $\Rightarrow F = rd_{\min} \approx 2$,
- Les cliques sont les mots de code d'un code correcteur d'erreurs très intéressant...et gratuit!

Des codes sur des cliques de taille constante

Exemple : mots de codes = cliques à 4 noeuds

Clique

Ensemble de noeuds d'un graphe entièrement interconnectés.



Symboles = arêtes

2 noeuds différents
 $\Rightarrow d_{\min} = 6$ arêtes

Codes sur cliques de taille $c \ll n$

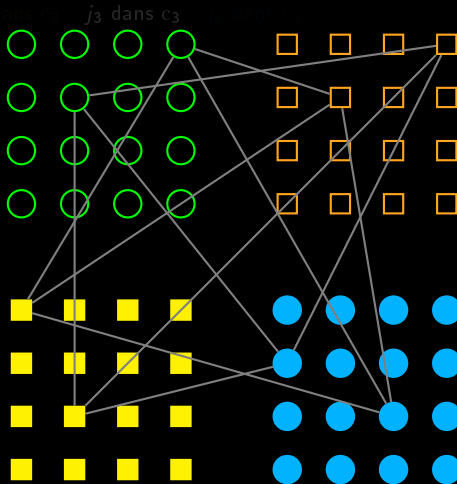
- $d_{\min} = 2(c - 1) \approx 2c$, rendement $r \approx \frac{c}{2} \binom{c}{2}^{-1}$
- $\Rightarrow F = rd_{\min} \approx 2$,
- Les cliques sont les mots de code d'un code correcteur d'erreurs très intéressant...**et gratuit!**

- 1 Mémoires associatives et codage correcteur
 - Mémoires associatives
 - Codage correcteur
 - Codes sur cliques
- 2 Réseau parcimonieux, principe et performances
 - Apprentissage
 - Remémoration
 - Performances
- 3 Développements
 - Messages flous
 - Sources corrélées
 - Messages parcimonieux
 - *Winner-take-all* global
 - Synchronisation temporelle
- 4 Conclusion, ouvertures

Notre modèle : apprentissage

- Exemple : $c = 4$ clusters de $l = 16$ neurones chacun,

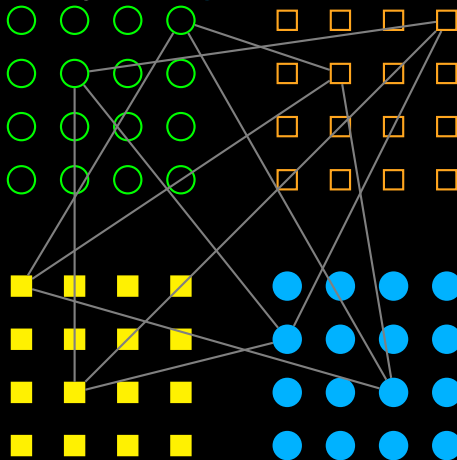
- $\underbrace{1000}_{j_1 \text{ dans } c_1} = 8$ $\underbrace{0011}_{j_2 \text{ dans } c_2} = 3$ $\underbrace{0010}_{j_3 \text{ dans } c_3} = 2$ $\underbrace{1001}_{j_4 \text{ dans } c_4} = 9$,



Notre modèle : apprentissage

- Exemple : $c = 4$ clusters de $l = 16$ neurones chacun,

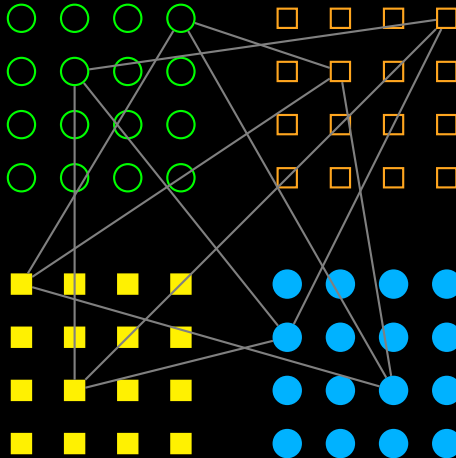
- $\underbrace{1000}_{j_1 \text{ dans } c_1} = 8 \quad \underbrace{0011}_{j_2 \text{ dans } c_2} = 3 \quad \underbrace{0010}_{j_3 \text{ dans } c_3} = 2 \quad \underbrace{1001}_{j_4 \text{ dans } c_4} = 9,$



Notre modèle : apprentissage

- Exemple : $c = 4$ clusters de $l = 16$ neurones chacun,

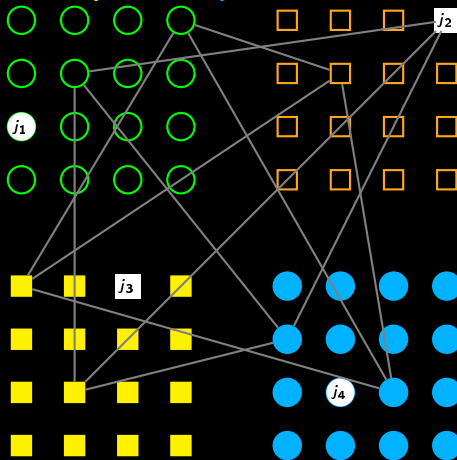
- $\underbrace{1000}_{j_1 \text{ dans } c_1} = 8$ $\underbrace{0011}_{j_2 \text{ dans } c_2} = 3$ $\underbrace{0010}_{j_3 \text{ dans } c_3} = 2$ $\underbrace{1001}_{j_4 \text{ dans } c_4} = 9$,



Notre modèle : apprentissage

- Exemple : $c = 4$ clusters de $l = 16$ neurones chacun,

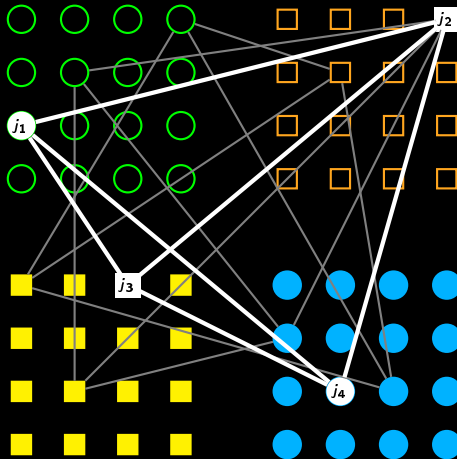
- $\underbrace{1000}_{j_1 \text{ dans } c_1} = 8$ $\underbrace{0011}_{j_2 \text{ dans } c_2} = 3$ $\underbrace{0010}_{j_3 \text{ dans } c_3} = 2$ $\underbrace{1001}_{j_4 \text{ dans } c_4} = 9$,



Notre modèle : apprentissage

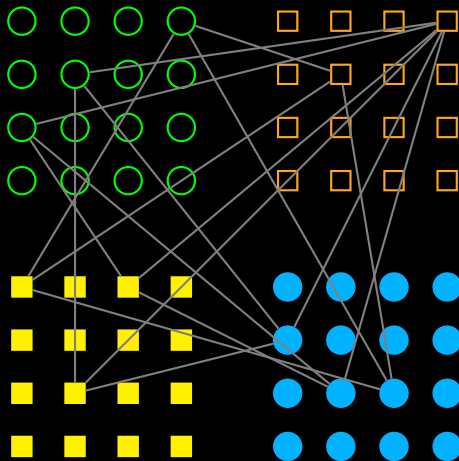
- Exemple : $c = 4$ clusters de $l = 16$ neurones chacun,

- $\underbrace{1000}_{j_1 \text{ dans } c_1} = 8$ $\underbrace{0011}_{j_2 \text{ dans } c_2} = 3$ $\underbrace{0010}_{j_3 \text{ dans } c_3} = 2$ $\underbrace{1001}_{j_4 \text{ dans } c_4} = 9$,



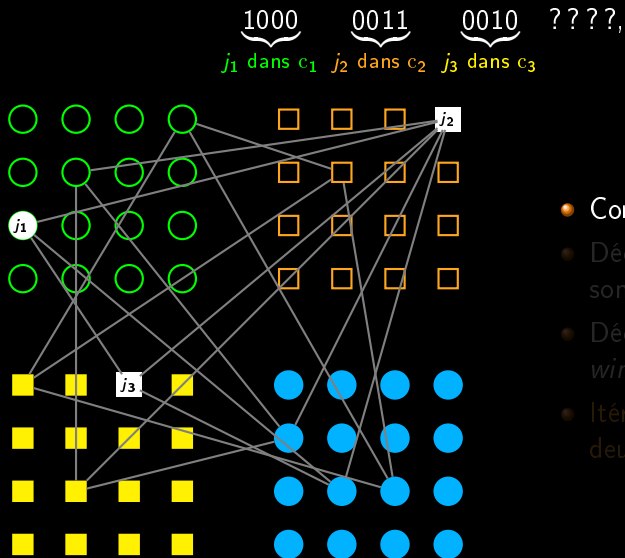
Notre modèle : remémoration

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



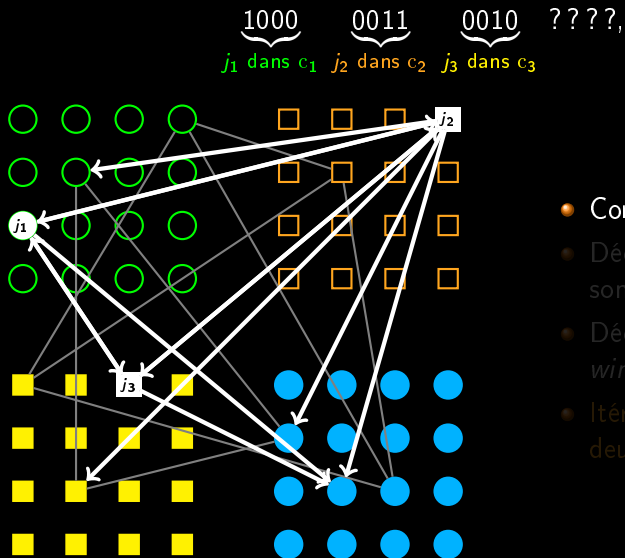
- Correspondance locale,
- Décodage global :
somme,
- Décodage local :
winner-take-all,
- Itération possible des
deux décodages.

Notre modèle : remémoration



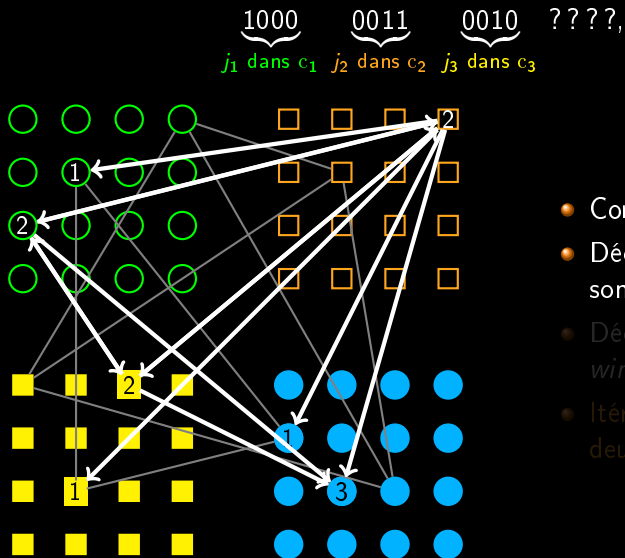
- Correspondance locale,
- Décodage global :
somme,
- Décodage local :
winner-take-all,
- Itération possible des
deux décodages.

Notre modèle : remémoration



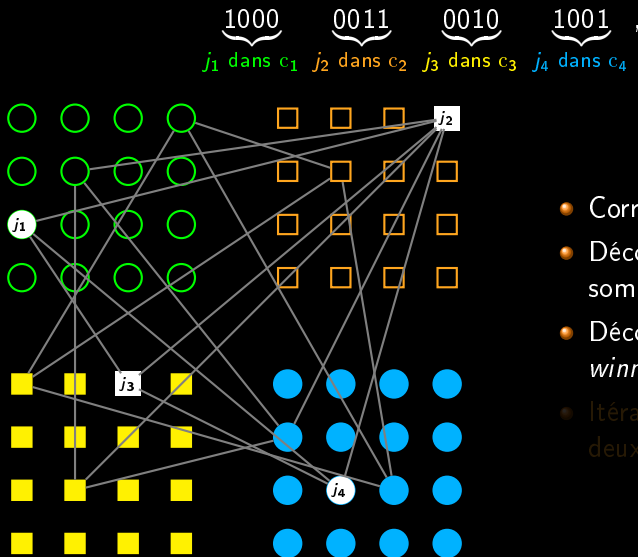
- Correspondance locale,
- Décodage global :
somme,
- Décodage local :
winner-take-all,
- Itération possible des
deux décodages.

Notre modèle : remémoration



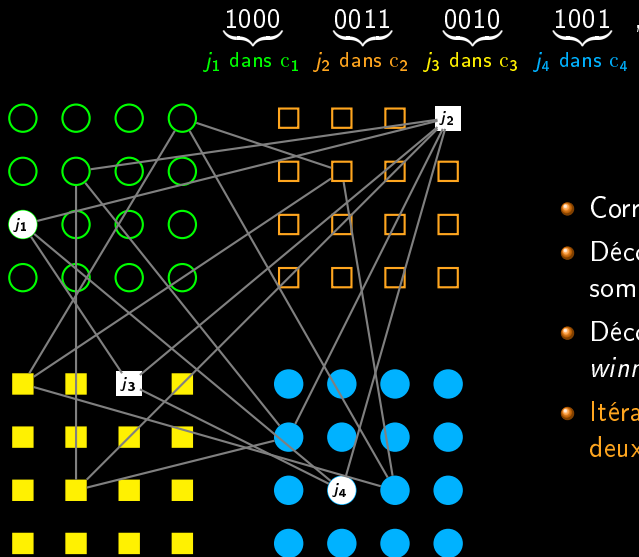
- Correspondance locale,
- Décodage global :
somme,
- Décodage local :
winner-take-all,
- Itération possible des
deux décodages.

Notre modèle : remémoration



- Correspondance locale,
- Décodage global :
somme,
- Décodage local :
winner-take-all,
- Itération possible des
deux décodages.

Notre modèle : remémoration



- Correspondance locale,
- Décodage global :
somme,
- Décodage local :
winner-take-all,
- Itération possible des
deux décodages.

Introduction d'un nouveau paramètre

- La densité d est le rapport entre le nombre de connexions utilisées et disponibles au total,
- Si messages i.i.d. : $d \approx 1 - \left(1 - \frac{1}{I^2}\right)^M$.

Courbes

Remarques

- $d = 1$: plus de distinction possible entre messages appris ou non,
- $d = f(I, M)$,
indépendant de c ,
- $d \approx \frac{M}{I^2}$, pour $M \ll I^2$.

Introduction d'un nouveau paramètre

- La densité d est le rapport entre le nombre de connexions utilisées et disponibles au total,
- Si messages i.i.d. : $d \approx 1 - \left(1 - \frac{1}{P^2}\right)^M$.

Courbes

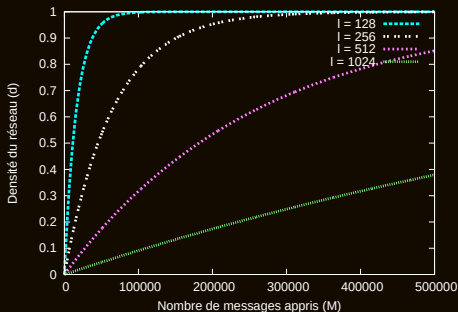
Remarques

- $d = 1$: plus de distinction possible entre messages appris ou non,
- $d = f(I, M)$,
indépendant de c ,
- $d \approx \frac{M}{P^2}$, pour $M \ll P^2$.

Introduction d'un nouveau paramètre

- La densité d est le rapport entre le nombre de connexions utilisées et disponibles au total,
- Si messages i.i.d. : $d \approx 1 - \left(1 - \frac{1}{l^2}\right)^M$.

Courbes



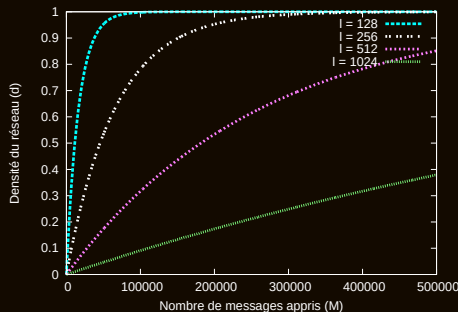
Remarques

- $d = 1$: plus de distinction possible entre messages appris ou non,
- $d = f(l, M)$,
indépendant de c ,
- $d \approx \frac{M}{l^2}$, pour $M \ll l^2$.

Introduction d'un nouveau paramètre

- La densité d est le rapport entre le nombre de connexions utilisées et disponibles au total,
- Si messages i.i.d. : $d \approx 1 - \left(1 - \frac{1}{l^2}\right)^M$.

Courbes



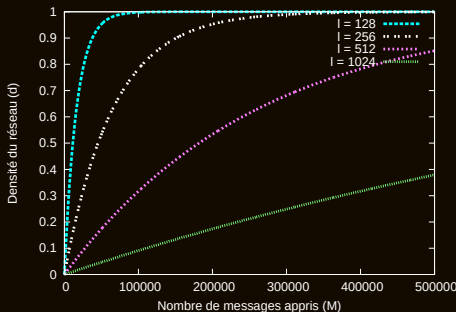
Remarques

- $d = 1$: plus de distinction possible entre messages appris ou non,
- $d = f(l, M)$,
indépendant de c ,
- $d \approx \frac{M}{l^2}$, pour $M \ll l^2$.

Introduction d'un nouveau paramètre

- La densité d est le rapport entre le nombre de connexions utilisées et disponibles au total,
- Si messages i.i.d. : $d \approx 1 - \left(1 - \frac{1}{l^2}\right)^M$.

Courbes



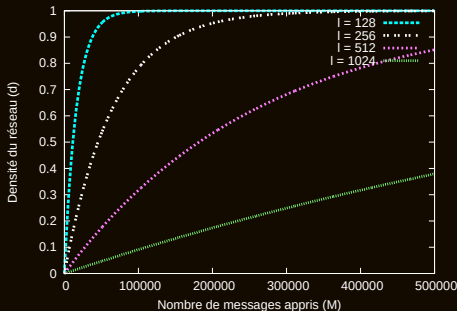
Remarques

- $d = 1$: plus de distinction possible entre messages appris ou non,
- $d = f(l, M)$, indépendant de c ,
- $d \approx \frac{M}{l^2}$, pour $M \ll l^2$.

Introduction d'un nouveau paramètre

- La densité d est le rapport entre le nombre de connexions utilisées et disponibles au total,
- Si messages i.i.d. : $d \approx 1 - \left(1 - \frac{1}{l^2}\right)^M$.

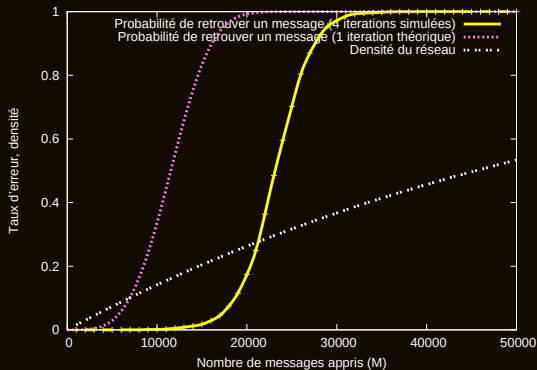
Courbes



Remarques

- $d = 1$: plus de distinction possible entre messages appris ou non,
- $d = f(l, M)$, indépendant de c ,
- $d \approx \frac{M}{l^2}$, pour $M \ll l^2$.

Mémoire associative



$c = 8$ clusters de $l = 256$ neurones ($\sim$ messages de longueur 64 bits),
Probabilité d'erreur lors de la remémoration de messages effacés à moitié.

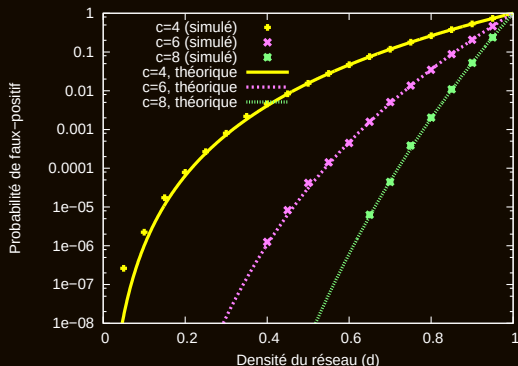
Réseau de Hopfield ($n = 790$)



Notre réseau



Classification

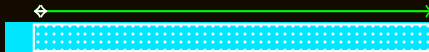


Probabilité de faux-positif pour divers nombres de clusters c et $l = 512$ neurones par cluster.

Réseau de Hopfield ($n = 740$)

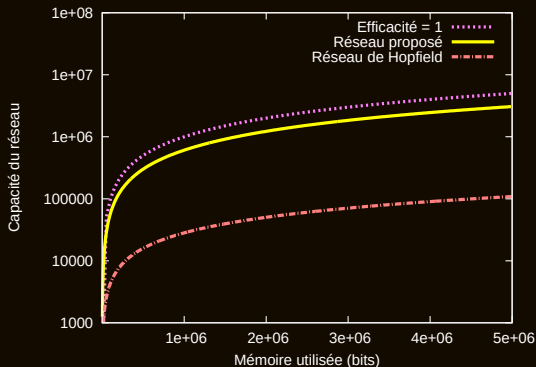


Notre réseau



Comparaison des capacités de notre réseau et de celui de Hopfield

Performances (3/3)



Comparaison des capacités du modèle de Hopfield et de nos réseaux (dans le cas de mémoire associative) pour une même quantité d'information utilisée.

Un mot sur la plausibilité

Analogies

Notre réseau		Littérature neuroscientifique
Cliques de neurones	↔	Cliques neurales
Décodage local	↔	<i>Winner-take-all</i>
Clusters	↔	Colonnes néocorticales
Code économe	↔	Neurones spécifiques

Limites

- Nécessité de posséder une information parfaite (bien que partielle).
- Les messages ne doivent pas être corrélés.
- Les clusters doivent être grands et peu nombreux.
- Taille fixe des messages appris.
- Utilisation systématique de tous les clusters.

Un mot sur la plausibilité

Analogies

Notre réseau		Littérature neuroscientifique
Cliques de neurones	↔	Cliques neurales
Décodage local	↔	<i>Winner-take-all</i>
Clusters	↔	Colonnes néocorticales
Code économe	↔	Neurones spécifiques

Limites

- Nécessité de posséder une information parfaite (bien que partielle),
- Les messages ne doivent pas être corrélés,
- Les clusters doivent être grands et peu nombreux,
- Taille fixe des messages appris,
- Utilisation systématique de tous les clusters.

Un mot sur la plausibilité

Analogies

Notre réseau		Littérature neuroscientifique
Cliques de neurones	↔	Cliques neurales
Décodage local	↔	<i>Winner-take-all</i>
Clusters	↔	Colonnes néocorticales
Code économe	↔	Neurones spécifiques

Limites

- Nécessité de posséder une information parfaite (bien que partielle),
- Les messages ne doivent pas être corrélés,
- Les clusters doivent être grands et peu nombreux,
- Taille fixe des messages appris,
- Utilisation systématique de tous les clusters.

Un mot sur la plausibilité

Analogies

Notre réseau		Littérature neuroscientifique
Cliques de neurones	↔	Cliques neurales
Décodage local	↔	<i>Winner-take-all</i>
Clusters	↔	Colonnes néocorticales
Code économe	↔	Neurones spécifiques

Limites

- Nécessité de posséder une information parfaite (bien que partielle),
- Les messages ne doivent pas être corrélés,
- Les clusters doivent être grands et peu nombreux,
- Taille fixe des messages appris,
- Utilisation systématique de tous les clusters.

Un mot sur la plausibilité

Analogies

Notre réseau		Littérature neuroscientifique
Cliques de neurones	↔	Cliques neurales
Décodage local	↔	<i>Winner-take-all</i>
Clusters	↔	Colonnes néocorticales
Code économe	↔	Neurones spécifiques

Limites

- Nécessité de posséder une information parfaite (bien que partielle),
- Les messages ne doivent pas être corrélés,
- Les clusters doivent être grands et peu nombreux,
- Taille fixe des messages appris,
- Utilisation systématique de tous les clusters.

Un mot sur la plausibilité

Analogies

Notre réseau		Littérature neuroscientifique
Cliques de neurones	↔	Cliques neurales
Décodage local	↔	<i>Winner-take-all</i>
Clusters	↔	Colonnes néocorticales
Code économe	↔	Neurones spécifiques

Limites

- Nécessité de posséder une information parfaite (bien que partielle),
- Les messages ne doivent pas être corrélés,
- Les clusters doivent être grands et peu nombreux,
- Taille fixe des messages appris,
- Utilisation systématique de tous les clusters.

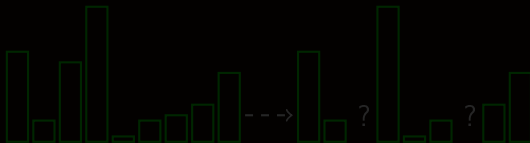
- 1 Mémoires associatives et codage correcteur
 - Mémoires associatives
 - Codage correcteur
 - Codes sur cliques
- 2 Réseau parcimonieux, principe et performances
 - Apprentissage
 - Remémoration
 - Performances
- 3 **Développements**
 - Messages flous
 - Sources corrélées
 - Messages parcimonieux
 - *Winner-take-all* global
 - Synchronisation temporelle
- 4 Conclusion, ouvertures

Messages flous

Problème

Les messages partiels doivent comporter des valeurs correctes et précises.

Modèle du bruit



Décodage *soft*

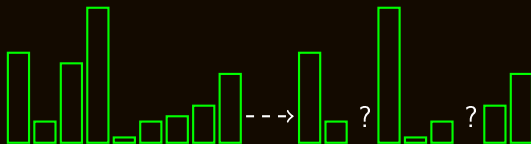


Messages flous

Problème

Les messages partiels doivent comporter des valeurs correctes et précises.

Modèle du bruit



Décodage *soft*

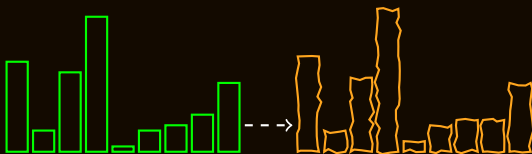


Messages flous

Problème

Les messages partiels doivent comporter des valeurs correctes et précises.

Modèle du bruit



Décodage *soft*

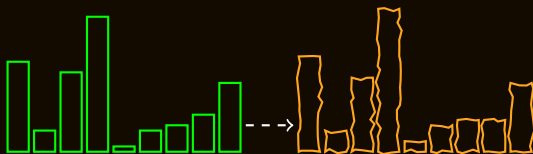


Messages flous

Problème

Les messages partiels doivent comporter des valeurs correctes et précises.

Modèle du bruit



Décodage *soft*

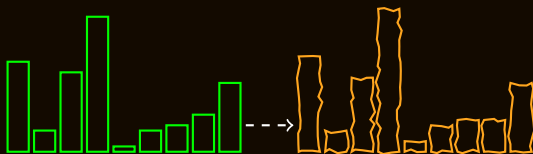


Messages flous

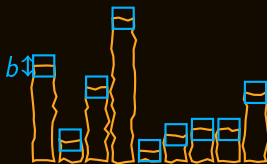
Problème

Les messages partiels doivent comporter des valeurs correctes et précises.

Modèle du bruit



Décodage *soft*

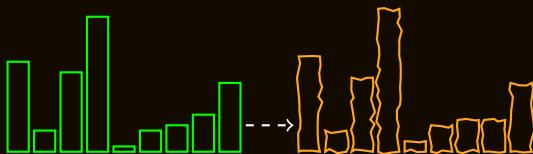


Messages flous

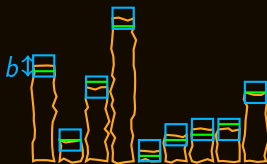
Problème

Les messages partiels doivent comporter des valeurs correctes et précises.

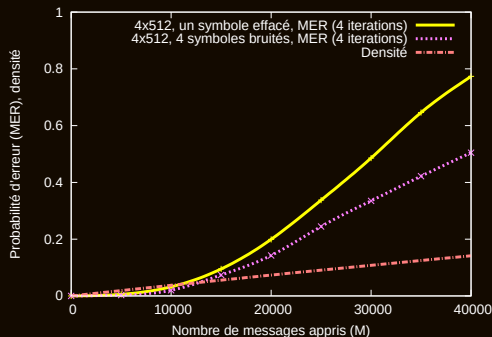
Modèle du bruit



Décodage *soft*



Simulations

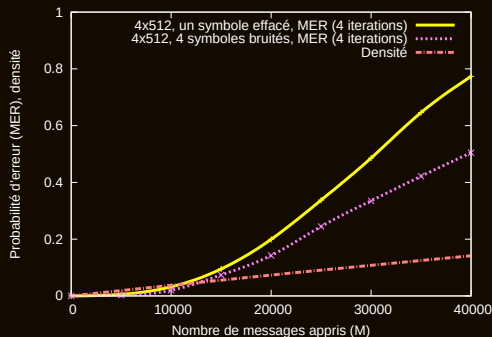


Comparaison des performances quand les messages sont partiellement effacés et lorsqu'ils sont bruités ($b = 5$).

Raison des bonnes performances

- Effacement : $\searrow$ cliques concurrentes ($\approx l$) $\nearrow$ probabilité ($\approx d^{c-1}$),
- Bruit : $\nearrow$ cliques concurrentes ($\approx b^c$) $\searrow$ probabilité ($\approx d^{\frac{c(c-1)}{2}}$).

Simulations



Comparaison des performances quand les messages sont partiellement effacés et lorsqu'ils sont bruités ($b = 5$).

Raison des bonnes performances

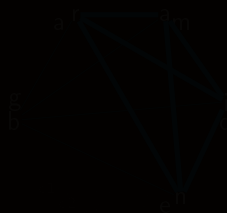
- Effacement : ↘ cliques concurrentes ($\approx l$) ↗ probabilité ($\approx d^{c-1}$),
- Bruit : ↗ cliques concurrentes ($\approx b^c$) ↘ probabilité ($\approx d^{\frac{c(c-1)}{2}}$).

Problème

La corrélation des messages accentue l'apparition de fausses cliques.

Combattre la corrélation par l'ajout de redondance aléatoire

- Il y a deux effets de la corrélation :
 - Un effet incontournable : *malin* et *matin* appris $\rightarrow$ *ma * in*?
 - Un effet venant du réseau :

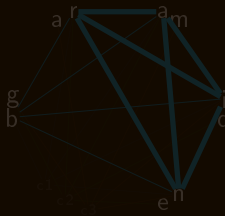


Problème

La corrélation des messages accentue l'apparition de fausses cliques.

Combattre la corrélation par l'ajout de redondance aléatoire

- Il y a deux effets de la corrélation :
 - Un effet incontournable : *malin* et *matin* appris $\rightarrow$ *ma * in*?
 - Un effet venant du réseau :



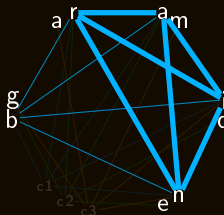
Problème

La corrélation des messages accentue l'apparition de fausses cliques.

Combattre la corrélation par l'ajout de redondance aléatoire

- Il y a deux effets de la corrélation :
 - Un effet incontournable : *malin* et *matin* appris $\rightarrow$ *ma * in*?
 - Un effet venant du réseau :

brain



Problème

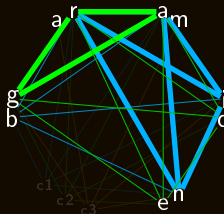
La corrélation des messages accentue l'apparition de fausses cliques.

Combattre la corrélation par l'ajout de redondance aléatoire

- Il y a deux effets de la corrélation :
 - Un effet incontournable : *malin* et *matin* appris $\rightarrow$ *ma * in*?
 - Un effet venant du réseau :

brain

grade



Messages corrélés

Problème

La corrélation des messages accentue l'apparition de fausses cliques.

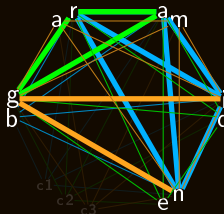
Combattre la corrélation par l'ajout de redondance aléatoire

- Il y a deux effets de la corrélation :
 - Un effet incontournable : *malin* et *matin* appris $\rightarrow$ *ma * in*?
 - Un effet venant du réseau :

brain

grade

gamin



Messages corrélés

Problème

La corrélation des messages accentue l'apparition de fausses cliques.

Combattre la corrélation par l'ajout de redondance aléatoire

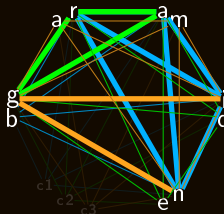
- Il y a deux effets de la corrélation :
 - Un effet incontournable : *malin* et *matin* appris $\rightarrow$ *ma * in*?
 - Un effet venant du réseau :

brain

grade

gamin

grain



Messages corrélés

Problème

La corrélation des messages accentue l'apparition de fausses cliques.

Combattre la corrélation par l'ajout de redondance aléatoire

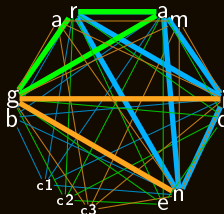
- Il y a deux effets de la corrélation :
 - Un effet incontournable : *malin* et *matin* appris $\rightarrow$ *ma * in*?
 - Un effet venant du réseau :

brain +c1

grade +c2

gamin +c3

grain +c ?



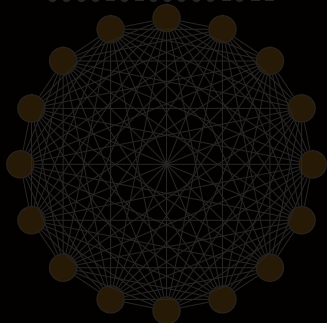
Vers un quatrième niveau de parcimonie

Problèmes

- Les cliques doivent rester peu nombreuses et grandes,
- Les messages appris sont tous de même taille.

Illustration

0000101000001011



Idée

- 1 Messages plus courts,
- 2 Clusters et codes économiques,
- 3 Réseau creux,
- 4 Messages parcimonieux.

Solutions

- *Winner-take-all* global,
- Synchronisation temporelle.

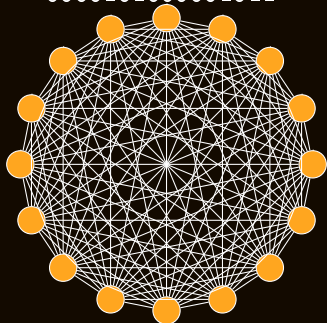
Vers un quatrième niveau de parcimonie

Problèmes

- Les cliques doivent rester peu nombreuses et grandes,
- Les messages appris sont tous de même taille.

Illustration

0000101000001011



Idée

- ➊ Messages plus courts,
- ➋ Clusters et codes économiques,
- ➌ Réseau creux,
- ➍ Messages parcimonieux.

Solutions

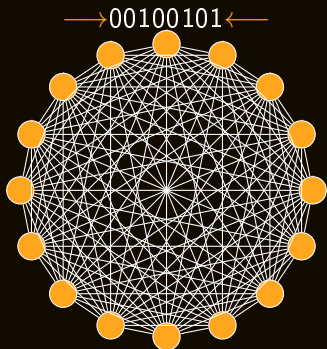
- *Winner-take-all* global,
- Synchronisation temporelle.

Vers un quatrième niveau de parcimonie

Problèmes

- Les cliques doivent rester peu nombreuses et grandes,
- Les messages appris sont tous de même taille.

Illustration



Idée

- 1 Messages plus courts,
- 2 Clusters et codes économes,
- 3 Réseau creux,
- 4 Messages parcimonieux.

Solutions

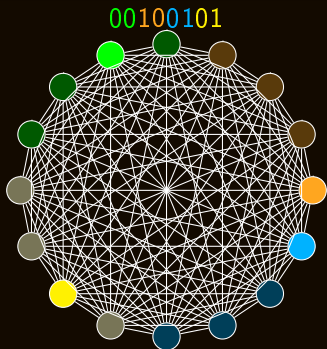
- Winner-take-all global,
- Synchronisation temporelle.

Vers un quatrième niveau de parcimonie

Problèmes

- Les cliques doivent rester peu nombreuses et grandes,
- Les messages appris sont tous de même taille.

Illustration



Idée

- 1 Messages plus courts,
- 2 Clusters et codes économes,
- 3 Réseau creux,
- 4 Messages parcimonieux.

Solutions

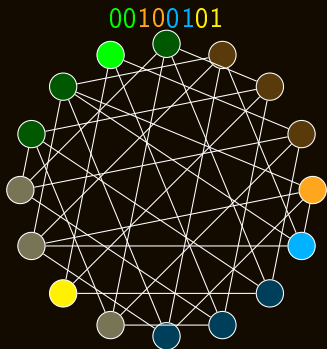
- Winner-take-all global,
- Synchronisation temporelle.

Vers un quatrième niveau de parcimonie

Problèmes

- Les cliques doivent rester peu nombreuses et grandes,
- Les messages appris sont tous de même taille.

Illustration



Idée

- 1 Messages plus courts,
- 2 Clusters et codes économes,
- 3 Réseau creux,
- 4 Messages parcimonieux.

Solutions

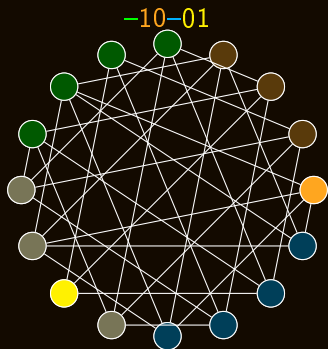
- Winner-take-all global,
- Synchronisation temporelle.

Vers un quatrième niveau de parcimonie

Problèmes

- Les cliques doivent rester peu nombreuses et grandes,
- Les messages appris sont tous de même taille.

Illustration



Idée

- 1 Messages plus courts,
- 2 Clusters et codes économes,
- 3 Réseau creux,
- 4 Messages parcimonieux.

Solutions

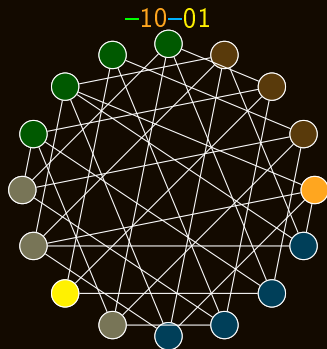
- Winner-take-all global,
- Synchronisation temporelle.

Vers un quatrième niveau de parcimonie

Problèmes

- Les cliques doivent rester peu nombreuses et grandes,
- Les messages appris sont tous de même taille.

Illustration



Idée

- 1 Messages plus courts,
- 2 Clusters et codes économes,
- 3 Réseau creux,
- 4 Messages parcimonieux.

Solutions

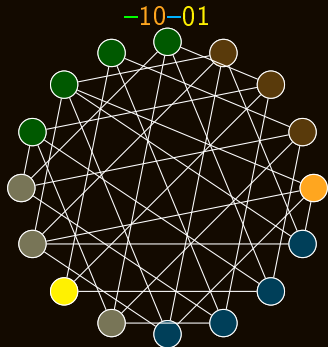
- *Winner-take-all* global,
- Synchronisation temporelle.

Vers un quatrième niveau de parcimonie

Problèmes

- Les cliques doivent rester peu nombreuses et grandes,
- Les messages appris sont tous de même taille.

Illustration



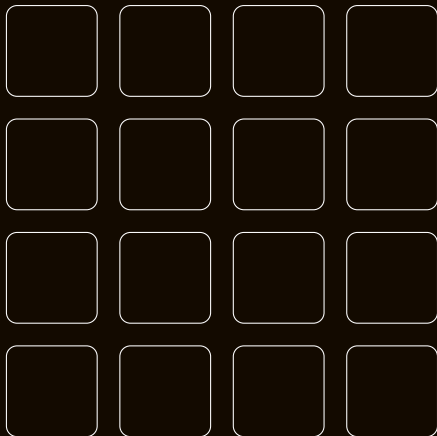
Idée

- 1 Messages plus courts,
- 2 Clusters et codes économes,
- 3 Réseau creux,
- 4 Messages parcimonieux.

Solutions

- *Winner-take-all* global,
- Synchronisation temporelle.

Illustration



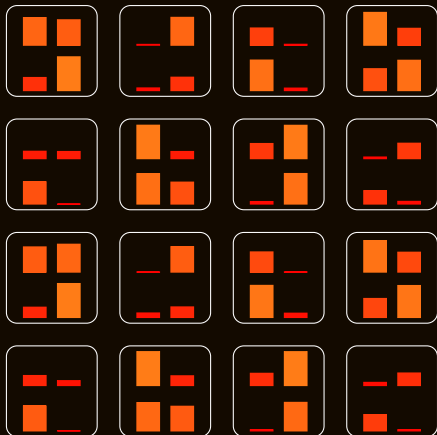
Idée

- Après passage de messages globaux...
- Après sélection des gagnants locaux...
- Sélection des gagnants globaux.

Intérêts

- Diversité x 4
- Taille des messages appris variable.

Illustration



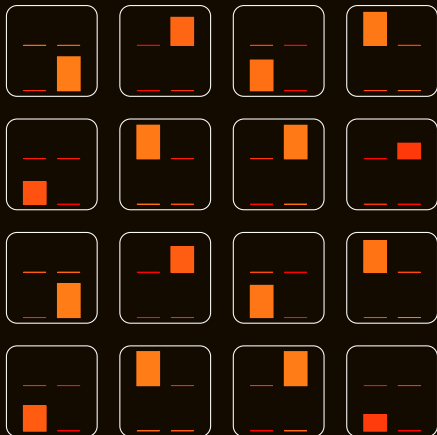
Idée

- Après passage de messages globaux...
- Après sélection des gagnants locaux...
- Sélection des gagnants globaux.

Intérêts

- Diversité x 4
- Taille des messages appris variable.

Illustration



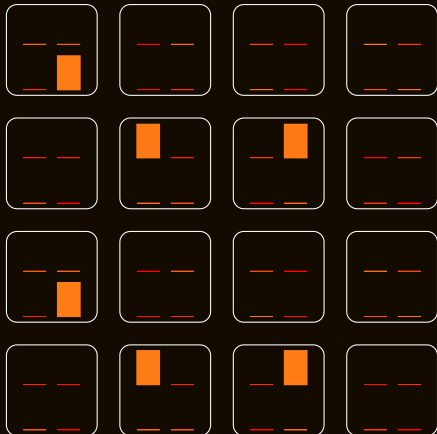
Idée

- Après passage de messages globaux...
- Après sélection des gagnants locaux...
- Sélection des gagnants globaux.

Intérêts

- Diversité x 4
- Taille des messages appris variable.

Illustration



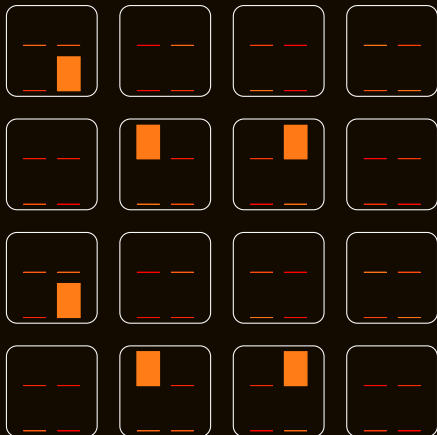
Idée

- Après passage de messages globaux...
- Après sélection des gagnants locaux...
- Sélection des gagnants globaux.

Intérêts

- Diversité
- Taille des messages appris variable.

Illustration



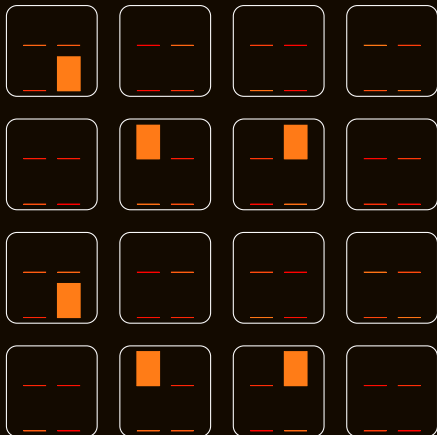
Idée

- Après passage de messages globaux...
- Après sélection des gagnants locaux...
- Sélection des gagnants globaux.

Intérêts

- Diversité $\propto c^2$,
- Taille des messages appris variable.

Illustration



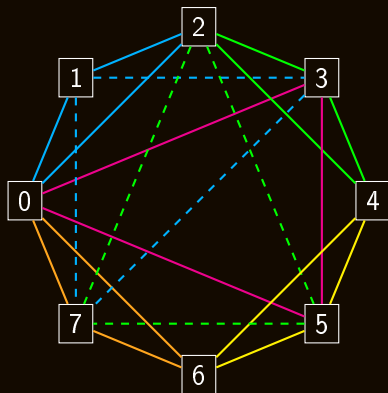
Idée

- Après passage de messages globaux...
- Après sélection des gagnants locaux...
- Sélection des gagnants globaux.

Intérêts

- Diversité $\propto c^2$,
- Taille des messages appris variable.

Illustration



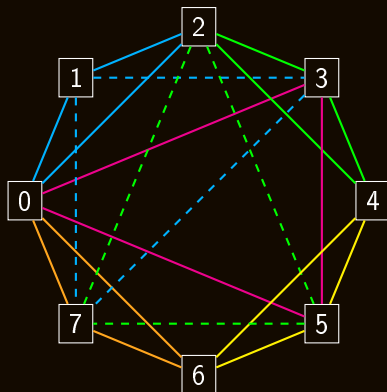
Idée

- Sous-réseaux indépendants,
- Localement : agrégation et *winner-take-all*,
- Globalement : détection de coïncidences.

Intérêt

Rapprochement possible vers la modélisation de pathologies liées à des défauts de synchronisation.

Illustration



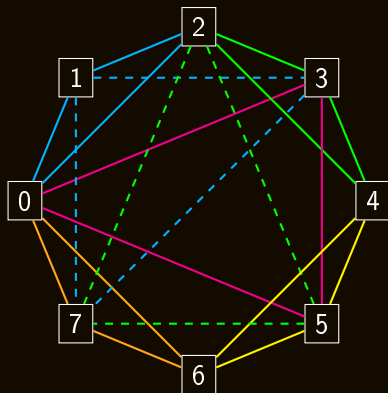
Idée

- Sous-réseaux indépendants,
- Localement : agrégation et *winner-take-all*,
- Globalement : détection de coïncidences.

Intérêt

Rapprochement possible vers la modélisation de pathologies liées à des défauts de synchronisation.

Illustration



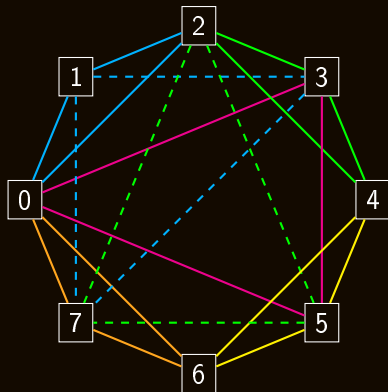
Idée

- Sous-réseaux indépendants,
- Localement : agrégation et *winner-take-all*,
- Globalement : détection de coïncidences.

Intérêt

Rapprochement possible vers la modélisation de pathologies liées à des défauts de synchronisation.

Illustration



Idée

- Sous-réseaux indépendants,
- Localement : agrégation et *winner-take-all*,
- Globalement : détection de coïncidences.

Intérêt

Rapprochement possible vers la modélisation de pathologies liées à des défauts de synchronisation.

- 1 Mémoires associatives et codage correcteur
 - Mémoires associatives
 - Codage correcteur
 - Codes sur cliques
- 2 Réseau parcimonieux, principe et performances
 - Apprentissage
 - Remémoration
 - Performances
- 3 Développements
 - Messages flous
 - Sources corrélées
 - Messages parcimonieux
 - *Winner-take-all* global
 - Synchronisation temporelle
- 4 Conclusion, ouvertures

Conclusion

Démarche

élaboration d'une mémoire associative →

↑
parcimonie

↑
codes distribués

architecture néocorticale

Résultats

- Capacités quasi-optimales, diversités considérables,
- Architecture massivement parallèle,
- Analogues avec les architectures et fonctionnements neurobiologiques,
- Robustesse, résilience, synchronisation,
- Degrés de liberté : inhibitions, temps, poids,
- Aucun compromis nécessaire entre performances et plausibilité.

Conclusion

Démarche

élaboration d'une mémoire associative →

↑
parcimonie

↑
codes distribués

architecture néocorticale

Résultats

- Capacités quasi-optimales, diversités considérables,
- Architecture massivement parallèle,
- Analogies avec les architectures et fonctionnements neurobiologiques,
- Robustesse, résilience, synchronisation...
- Degrés de liberté : inhibitions, temps, poids,
- Aucun compromis nécessaire entre performances et plausibilité.

Conclusion

Démarche

élaboration d'une mémoire associative →

↑
parcimonie

↑
codes distribués

architecture néocorticale

Résultats

- Capacités quasi-optimales, diversités considérables,
- Architecture massivement parallèle,
- Analogies avec les architectures et fonctionnements neurobiologiques,
- Robustesse, résilience, synchronisation...
- Degrés de liberté : inhibitions, temps, poids,
- Aucun compromis nécessaire entre performances et plausibilité.

Conclusion

Démarche

élaboration d'une mémoire associative →

↑
parcimonie

↑
codes distribués

architecture néocorticale

Résultats

- Capacités quasi-optimales, diversités considérables,
- Architecture massivement parallèle,
- Analogies avec les architectures et fonctionnements neurobiologiques,
- Robustesse, résilience, synchronisation...
- Degrés de liberté : inhibitions, temps, poids,
- Aucun compromis nécessaire entre performances et plausibilité.

Conclusion

Démarche

élaboration d'une mémoire associative →

architecture néocorticale

↑
parcimonie

↑
codes distribués

Résultats

- Capacités quasi-optimales, diversités considérables,
- Architecture massivement parallèle,
- Analogies avec les architectures et fonctionnements neurobiologiques,
- Robustesse, résilience, synchronisation...
- Degrés de liberté : inhibitions, temps, poids,
- Aucun compromis nécessaire entre performances et plausibilité.

Conclusion

Démarche

élaboration d'une mémoire associative →

↑
parcimonie

↑
codes distribués

architecture néocorticale

Résultats

- Capacités quasi-optimales, diversités considérables,
- Architecture massivement parallèle,
- Analogies avec les architectures et fonctionnements neurobiologiques,
- Robustesse, résilience, synchronisation...
- Degrés de liberté : inhibitions, temps, poids,
- Aucun compromis nécessaire entre performances et plausibilité.

Conclusion

Démarche

élaboration d'une mémoire associative →

↑
parcimonie

↑
codes distribués

architecture néocorticale

Résultats

- Capacités quasi-optimales, diversités considérables,
- Architecture massivement parallèle,
- Analogies avec les architectures et fonctionnements neurobiologiques,
- Robustesse, résilience, synchronisation...
- Degrés de liberté : inhibitions, temps, poids,
- Aucun compromis nécessaire entre performances et plausibilité.

Journal

- Un article dans *IEEE Transactions on neural networks*.

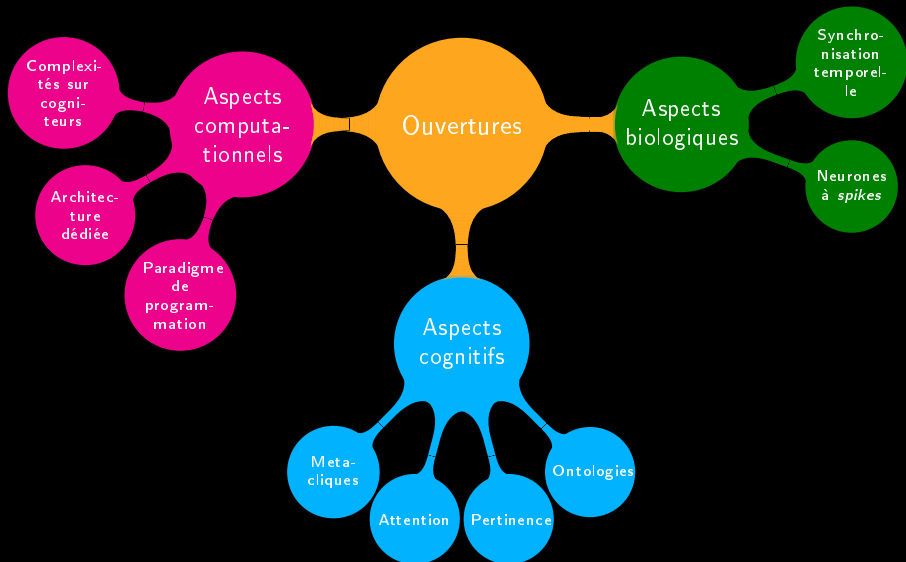
Conférences

- Une communication dans *Proc. of 6th Int'l Symposium on Turbo Codes and Iterative Information Processing*,
- Une communication dans *Proc. of IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain*.

Brevets

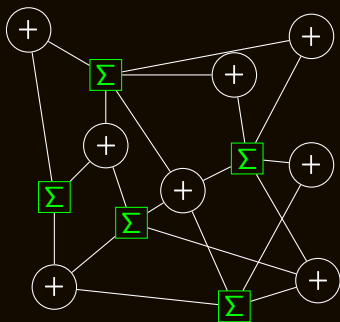
- Un premier brevet déposé en 2010 : réseau présenté,
- Un second en cours de dépôt : apprentissage de séquences.

Ouvertures



Merci pour votre attention. Je suis à votre disposition pour toutes questions complémentaires.

Décodeur *LDPC*



“Décodeur” néocortical

