



*De DeepBlue à AlphaGo*  
*Dernières avancées en IA appliquée aux jeux*

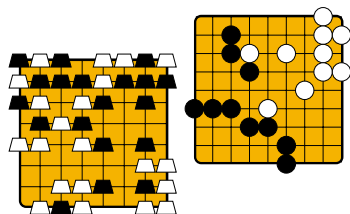
André Fabbri

## Propriétés supplémentaires

- 2 joueurs → *objectifs antagonistes*
- Actions déterministes → *aucun hasard*
- Information parfaite → *aucune information cachée*

## Propriétés supplémentaires

- 2 joueurs → *objectifs antagonistes*
- Actions déterministes → *aucun hasard*
- Information parfaite → *aucune information cachée*

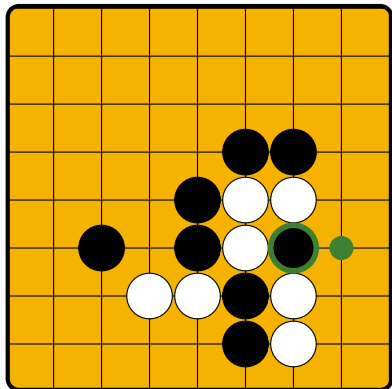


## Exemples

- **Go** *Origine*  $\approx V^e$  av. J.C.
- **Échecs** *Origine*  $\approx IX^e$
- **Clobber** *Origine* 2002

*Jeux de réflexion et pratiqués par l'Homme*

# Règles du jeu de Go



## Objectif

Contrôler le plus grand territoire

→ *Plateau standard 19x19*

## Règle de pose

1 intersection vide du plateau

## Règle de capture

Chaîne sans libertés est capturée

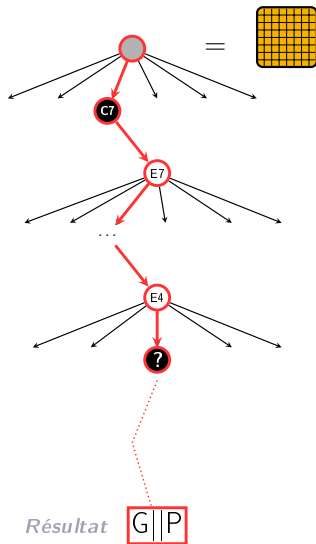
## Lexique :

→ 1 **chaîne** = pierres adjacentes de la même couleur

→ 1 **liberté** = 1 intersection vide adjacente à une chaîne

# Complexités des jeux combinatoires

Tours



Résultat

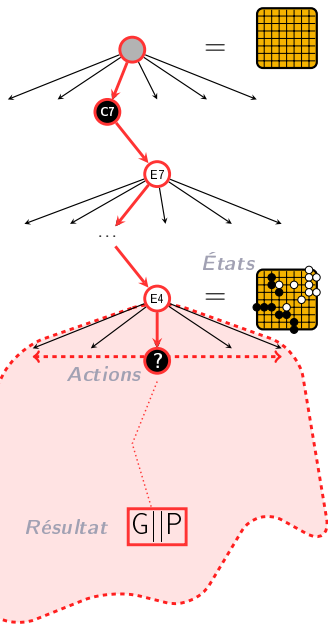
G|P

*G* : joueur courant gagne

*P* : joueur courant perd

# Complexités des jeux combinatoires

Tours



## Espace d'états

Nombre d'états possibles

Échecs  $\simeq 10^{47}$ , Go  $\simeq 10^{171}$

## Facteur de branchement

Nombre d'actions par tour

Échecs  $\simeq 35$ , Go  $\simeq 250$

G : joueur courant gagne

P : joueur courant perd

## Application historique pour le domaine

- Formalisation simple → *Problème bien défini*
- Humainement opposable → *Capacité cognitive*

## Application historique pour le domaine

- Formalisation simple → *Problème bien défini*
- Humainement opposable → *Capacité cognitive*

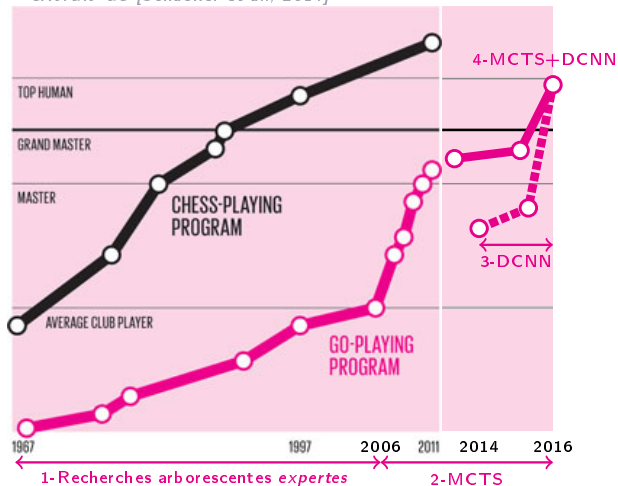
Objectif poursuivi	Exemple
Issue théorique du jeu (exacte)	<i>Jeu de Dames</i> [Schaeffer et al., 2007]
Performances en jeu (approchée)	<i>Deep Blue</i> [Campbell et al., 2002] <i>Alpha-Go</i> [Silver et al., 2016]
Processus cognitifs (modèles)	<i>Perception experte</i> [Harré et al., 2012]

*Problèmes propices à un super calculateur ?*



# Évolution des performances de jeu

extrait de [Schaeffer et al., 2014]



... seulement 20 ans après les échecs !

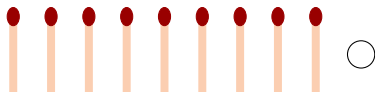
# Première partie I

## Recherche arborescente experte

# Comment décide-t-on dans un jeu ?

## Jeu des allumettes

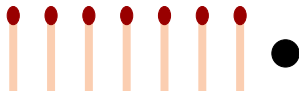
- 1 ou 2 à tour de rôle
- Perdant si aucune action



# Comment décide-t-on dans un jeu ?

## Jeu des allumettes

- 1 ou 2 à tour de rôle
- Perdant si aucune action



# Comment décide-t-on dans un jeu ?

## Jeu des allumettes

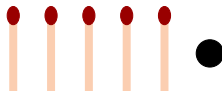
- 1 ou 2 à tour de rôle
- Perdant si aucune action



# Comment décide-t-on dans un jeu ?

## Jeu des allumettes

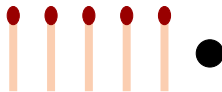
- 1 ou 2 à tour de rôle
- Perdant si aucune action



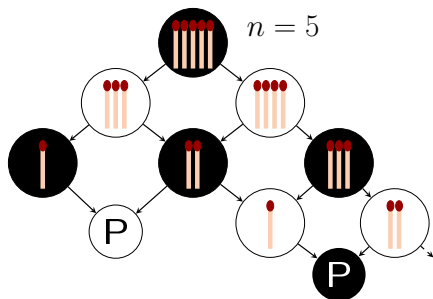
# Comment décide-t-on dans un jeu ?

## Jeu des allumettes

- 1 ou 2 à tour de rôle
- Perdant si aucune action



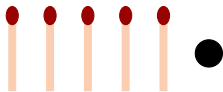
## Recherche dynamique



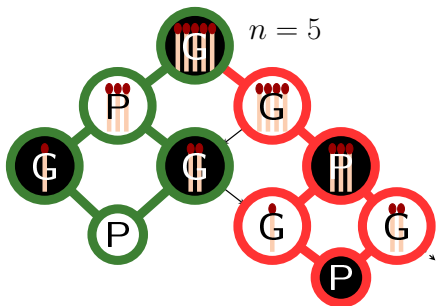
# Comment décide-t-on dans un jeu ?

## Jeu des allumettes

- 1 ou 2 à tour de rôle
- Perdant si aucune action



## Recherche dynamique



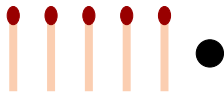
... *rarement praticable*



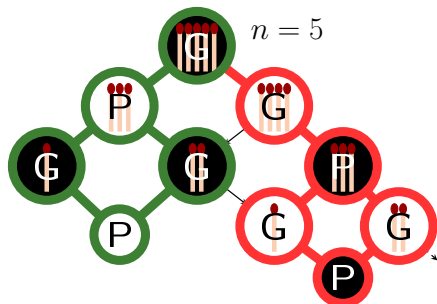
# Comment décide-t-on dans un jeu ?

## Jeu des allumettes

- 1 ou 2 à tour de rôle
- Perdant si aucune action

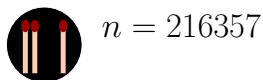


## Recherche dynamique



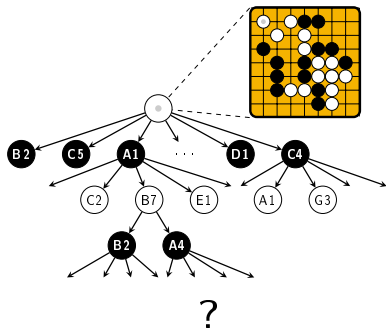
## Évaluation statique

Tas  $n \div 3$  sont P



*... rarement praticable*



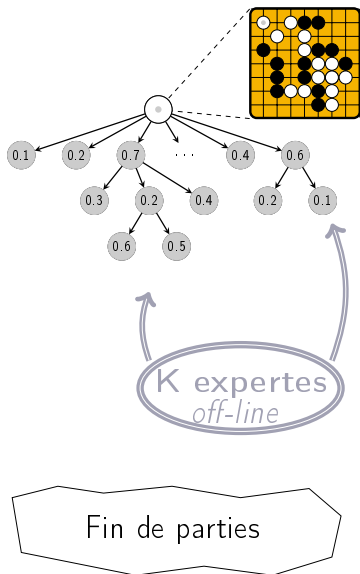


*exemples :  $\alpha - \beta$ , quiescence search*

## Recherche arborescente

- Arbre de recherche
- Exploration progressive

Fin de parties



exemples :  $\alpha - \beta$ , quiescence search

## Recherche arborescente

- Arbre de recherche
- Exploration progressive

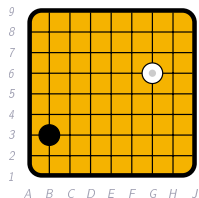
## Fonction d'évaluation experte

- Criticité ou probabilité gain
- Conception par un **expert**

*Recherche guidée par l'évaluation*

# Comment évalue-t-on une position ?

## Évaluation directe de l'état



=

0.6

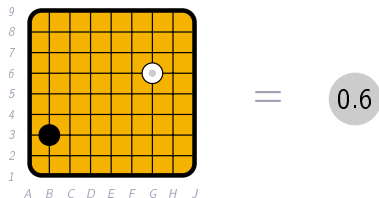
- 1 état = 1 évaluation
- Aucune généralisation

*livre d'ouverture, fin de parties*

***A priori précis mais limité***

# Comment évalue-t-on une position ?

## Évaluation directe de l'état

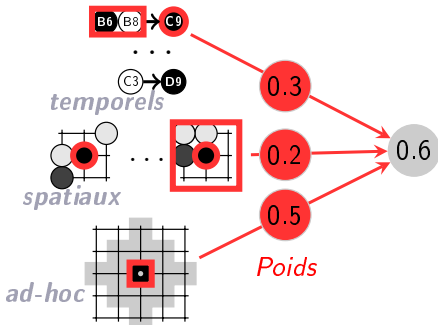


- 1 état = 1 évaluation
- Aucune généralisation

*livre d'ouverture, fin de parties*

***A priori précis mais limité***

Ensemble de *features* locaux  $\simeq$  propriétés : état  $\mapsto$  0 ou 1



- Combinaison pondérée
- Généralisation ambiguë

*évaluation par défaut*

***Quels features ? poids ?***

# Succès et limites des approches expertes



## Deep Blue le super-calculateur

- Vaste recherche arborescente  
→  $10^8$  d'état/s
- Évaluation fortement experte  
→ 8000 features experts
- Hardware spécifique  
→ Chess CPU

# Succès et limites des approches expertes



## Deep Blue le super-calculateur

- Vaste recherche arborescente  
→  $10^8$  d'état/s
- Évaluation fortement experte  
→ 8000 features experts
- Hardware spécifique  
→ Chess CPU

Oui mais pour le jeu de Go ...

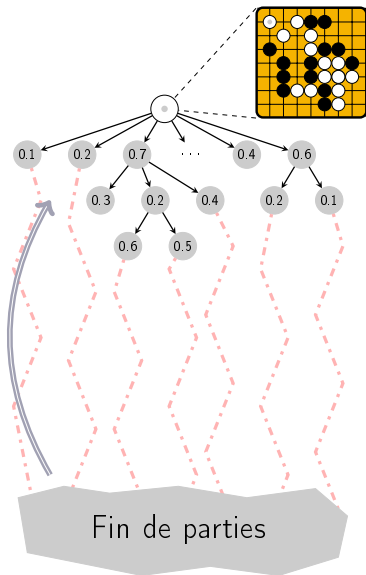
- 1 **Comment évaluer ? Comment définir les features ?**  
→ Toutes les pièces sont identiques au Go
- 2 **Comment tenir le changement d'échelle ?**  
→ Facteur de branchement de 250 au Go (35 aux échecs)



## Deuxième partie II

Recherche arborescente de Monte Carlo

# Évaluation des états par échantillonnage (Monte Carlo)

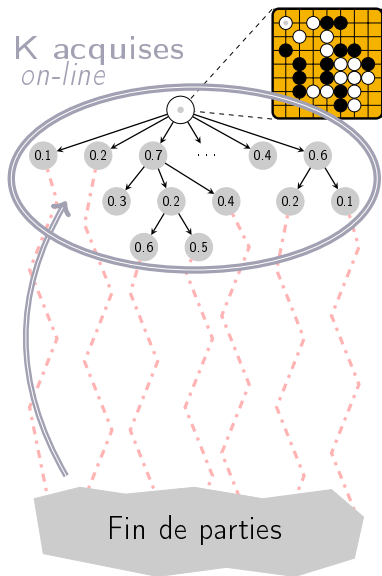


Moyenne des résultats  
de simulations aléatoires

## Approximation statistique

1 simulation = 1 issue possible  
1 résultat = 1 estimation

# Évaluation des états par échantillonnage (Monte Carlo)



Moyenne des résultats  
de simulations aléatoires

## Approximation statistique

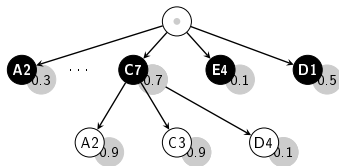
1 simulation = 1 issue possible  
1 résultat = 1 estimation

## Apprentissage par renforcement

1 simulation = 1 expérience  
1 résultat = 1 renforcement

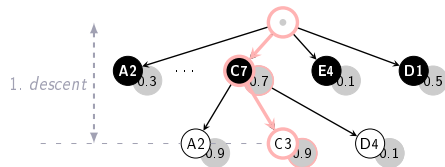
*Aucune feature experte  
nécessaire*

# Une nouvelle méthode : *Monte Carlo Tree Search* (MCTS)



Fin de parties

# Une nouvelle méthode : *Monte Carlo Tree Search* (MCTS)

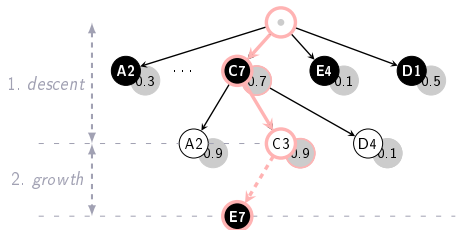


1 itération

- 1 descente

Fin de parties

# Une nouvelle méthode : *Monte Carlo Tree Search* (MCTS)

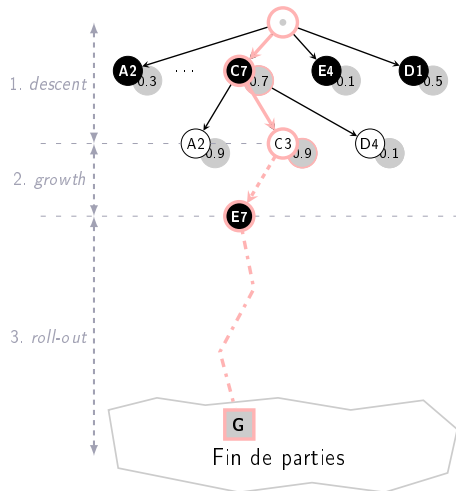


## 1 itération

- 1 descente
- 1 nouvel état

Fin de parties

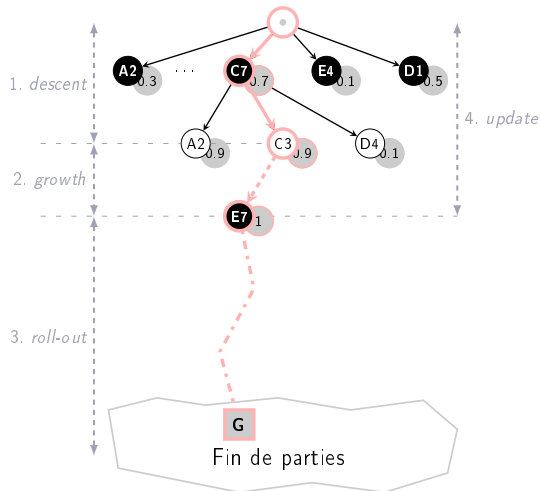
# Une nouvelle méthode : *Monte Carlo Tree Search* (MCTS)



## 1 itération

- 1 descente
- 1 nouvel état
- 1 simulation

# Une nouvelle méthode : *Monte Carlo Tree Search* (MCTS)

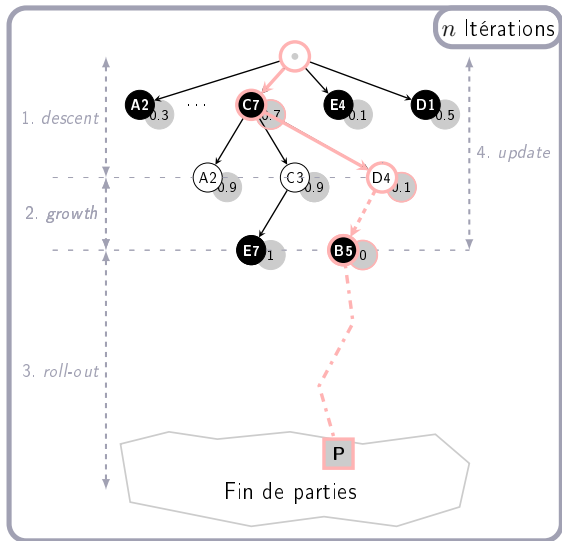


## 1 itération

- 1 descente
- 1 nouvel état
- 1 simulation
- 1 rétroaction



# Une nouvelle méthode : *Monte Carlo Tree Search* (MCTS)

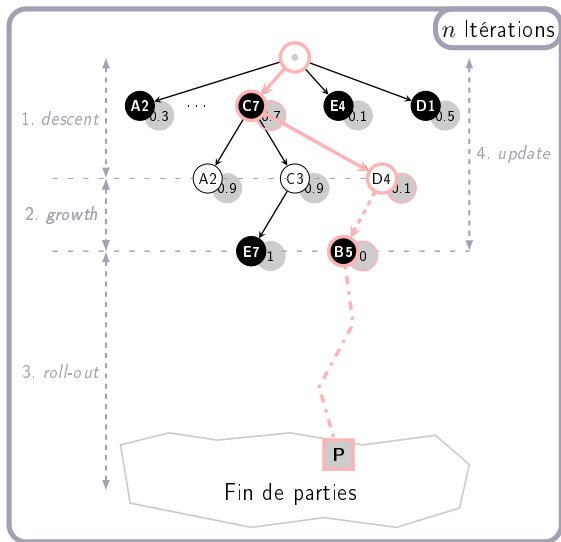


## 1 itération

- 1 descente
- 1 nouvel état
- 1 simulation
- 1 rétroaction

1 tour =  $n$  iterations

# Une nouvelle méthode : *Monte Carlo Tree Search* (MCTS)



## 1 itération

- 1 descente
- 1 nouvel état
- 1 simulation
- 1 rétroaction

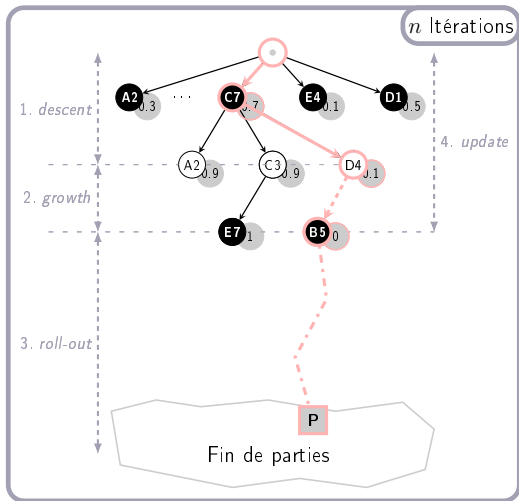
1 tour =  $n$  iterations

## Dilemme

*Exploration/Exploitation*

*Co-construction recherche et évaluation*

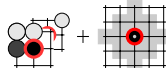
# Un apprentissage facilité par des connaissances expertes



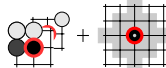
# Un apprentissage facilité par des connaissances expertes

## A-priori experts

Amorce experte



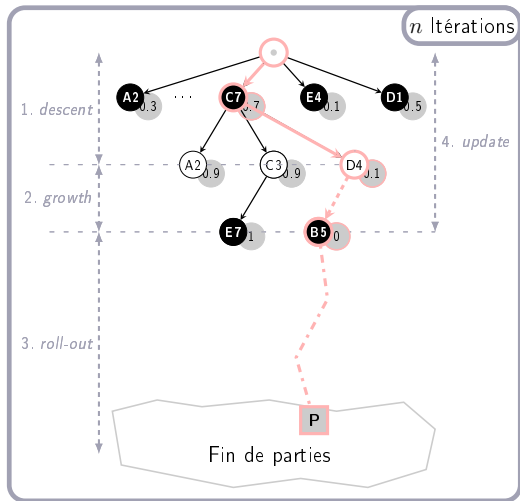
Simulation experte



ex : Mogo [Gelly and Silver, 2008]

+

+



*Premiers programmes de niveau professionnel... en 9x9*

# Un passage à l'échelle limité

## Évaluation évolutive

*Plus de simulations*



*Meilleures estimations*

## Historique MC

*Gobble (1993)  $\approx 10^1$  sim/s*

*Fuego (2010)  $\approx 10^4$  sim/s*

# Un passage à l'échelle limité

## Évaluation évolutive

Plus de simulations



Meilleures estimations

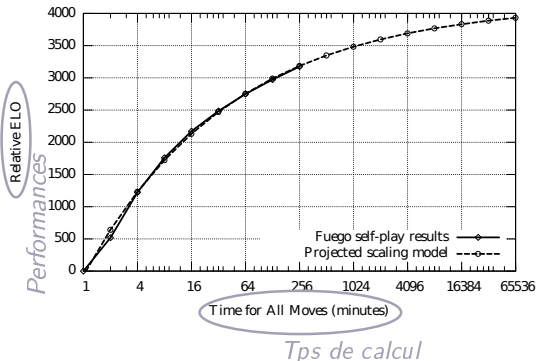
## Historique MC

Gobble (1993)  $\approx 10^1$  sim/s

Fuego (2010)  $\approx 10^4$  sim/s

## Les performances stagnent

extrait de [Segal, 2011]



L'évaluation directe des états ne suffit pas...

à la généralisation ?

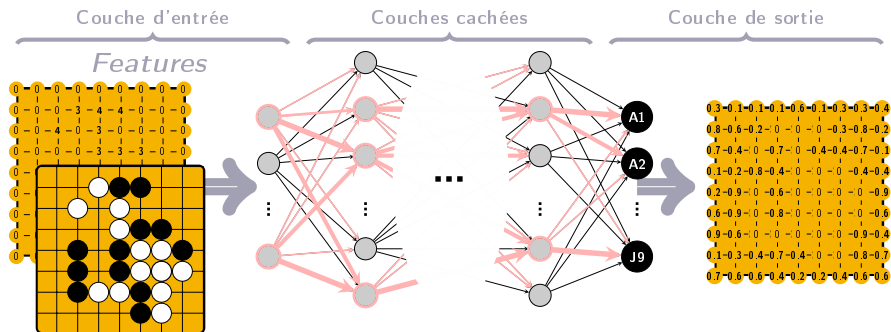
## Troisième partie III

# Réseau de neurones convolutif profond





# Du cerveau au réseau artificiel



## Réseau de neurones artificiel

- Neurones connectés en couches → *ordre de progression*
- Sorties de  $C_j \rightarrow$  Entrées de  $C_{j+1}$  → *combinaisons de features*
- Poids paramétrables → *apprentissage possible*

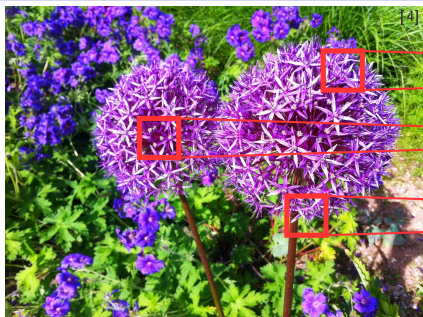
***Fonction d'évaluation avec features recombinaibles***

# Un réseau... convolutif?

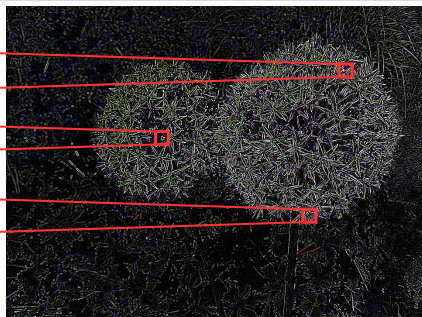


*Image initiale*

# Un réseau... convolutif?



*Image initiale*



*Image convoluée*

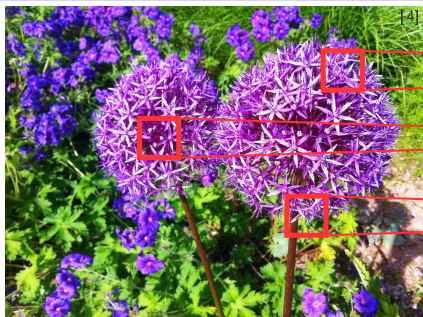
## Convolution d'une image par un noyau

- Filtre local de poids → *masque (noyau)*
- Calcul relatif à chaque pixel → *nouveau pixel*
- Invariant sur toute l'image → *nouvelle image*

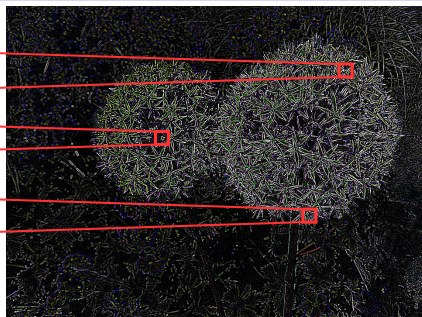
Filtre 3x3  
*Détecteurs de bords*

-1	-1	-1
-1	8	-1
-1	-1	-1

# Un réseau... convolutif?



*Image initiale*



*Image convoluée*

## Convolution d'une image par un noyau

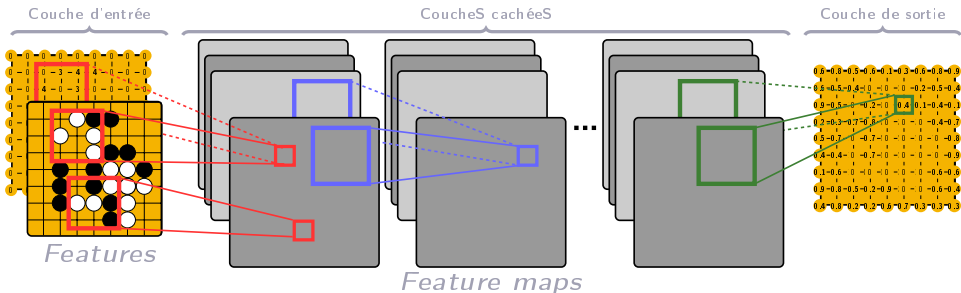
- Filtre local de poids → *masque (noyau)*
- Calcul relatif à chaque pixel → *nouveau pixel*
- Invariant sur toute l'image → *nouvelle image*

Filtre 3x3  
Détecteurs de bords

-1	-1	-1
-1	8	-1
-1	-1	-1

***Généralisation des entrées par convolutions successives***

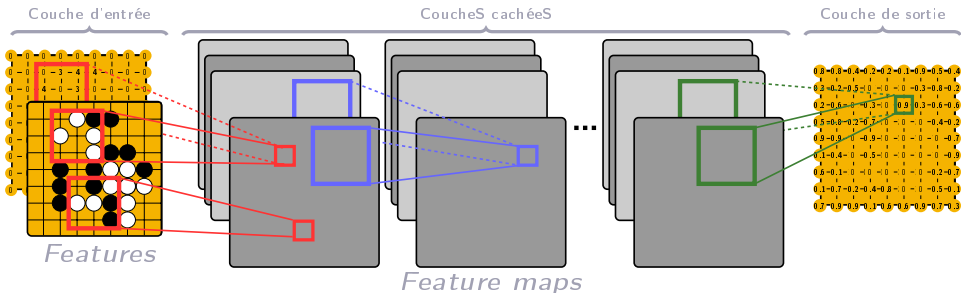
# Deep Convolutional Neural Network (DCNN)



1 entrée = 1 image

1 couche cachée =  $k$  feature maps

# Deep Convolutional Neural Network (DCNN)



1 entrée = 1 image

1 couche cachée =  $k$  feature maps

## Feature map

- 1 nappe de neurones
- 1 unique filtre
- Invariance spatiale

→ image convoluée

→ mini-feature

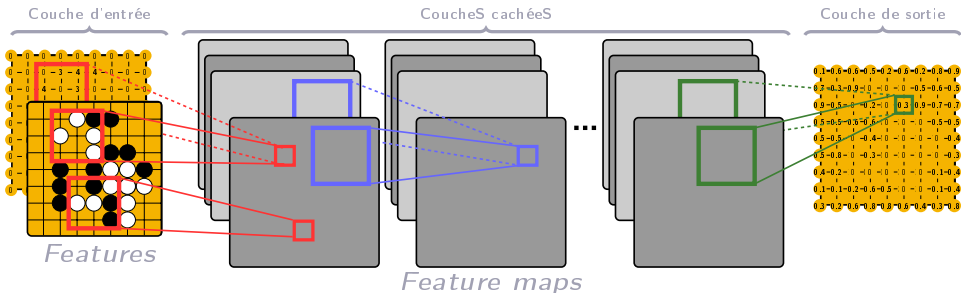
→ poids partagés

## Filtre 3x3

À apprendre

$W_{0,2}$	$W_{1,2}$	$W_{2,2}$
$W_{0,1}$	$W_{1,1}$	$W_{2,1}$
$W_{0,0}$	$W_{1,0}$	$W_{2,0}$

# Deep Convolutional Neural Network (DCNN)



1 entrée = 1 image

1 couche cachée =  $k$  feature maps

## Feature map

- 1 nappe de neurones
- 1 unique filtre
- Invariance spatiale

→ image convoluée

→ mini-feature

→ poids partagés

## Filtre 3x3

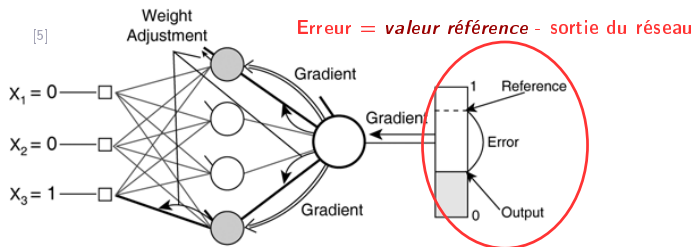
À apprendre

$W_{0,2}$	$W_{1,2}$	$W_{2,2}$
$W_{0,1}$	$W_{1,1}$	$W_{2,1}$
$W_{0,0}$	$W_{1,0}$	$W_{2,0}$

*Qualité des features induits dépend du nombre de couches*

# Apprentissage des poids du réseau *off-line*

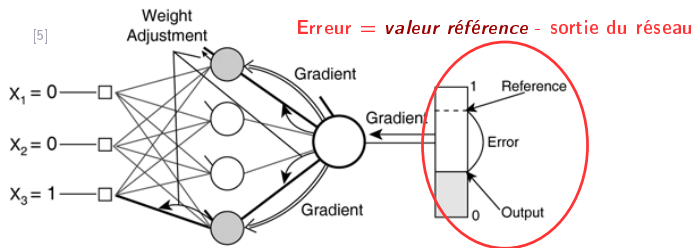
## Algorithme de rétropropagation de l'erreur [Rumelhart et al., 1988]





# Apprentissage des poids du réseau *off-line*

## Algorithme de rétropropagation de l'erreur [Rumelhart et al., 1988]



### Apprentissage supervisé

#### **Référence experte**

- Base d'apprentissage
- Réponses "attendues"

### Apprend par mimétisme

### Apprentissage par renforcement

#### **Récompense + sortie suivante**

- Essais et erreurs
- *Curse of dimensionality*

### Apprend de lui-même

## Performances évolutives ?

*Plus de couches cachées*



*Meilleures généralisations*

## Historique CNN

[Schraudolph et al., 1994]  $\approx 10^3$  poids /  $10^4$  données

AlphaGo [Silver et al., 2016]  $\approx 10^6$  poids /  $10^6$  données

# Réapparition des réseaux de neurones pour le Go

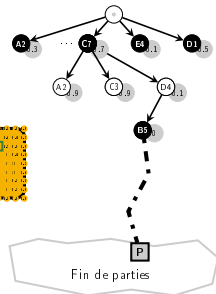
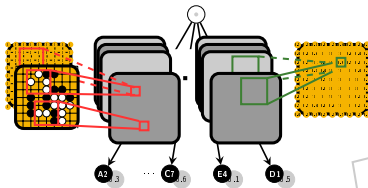
## Performances évolutives ?

Plus de couches cachées



Meilleures généralisations

DCNN > MCTS  
(AlphaGo) (Pachi 100k)



## Historique CNN

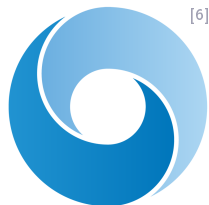
[Schraudolph et al., 1994]  $\approx 10^3$  poids /  $10^4$  données

AlphaGo [Silver et al., 2016]  $\approx 10^6$  poids /  $10^6$  données

**Un réseau de neurone SEUL évalue 1 UNIQUE position !**

## Quatrième partie IV

AlphaGo



[6]

## Historique DeepMind

- 2010 : Création
- 2014 : Rachat Google - début du projet AlphaGo
- 2015 : AlphaGo contre Fan Hui (2 Dan Pro)
- 2016 : AlphaGo contre Lee Sedol (9 Dan Pro)

### parmi les membres du projet...

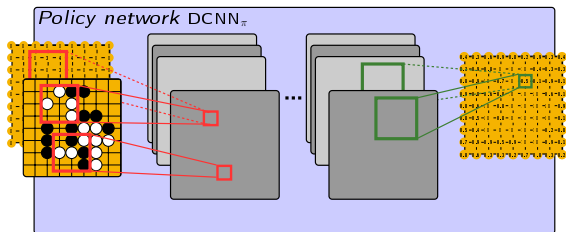
- David Silver → *Spécialiste MCTS appliqué au Go*
- Aja Huang → *Spécialiste MCTS appliqué au Go*
- Ilya Sutskever → *Spécialiste CNN (appliqué au Go)*

***Compétence de spécialistes avec la puissance de Google***

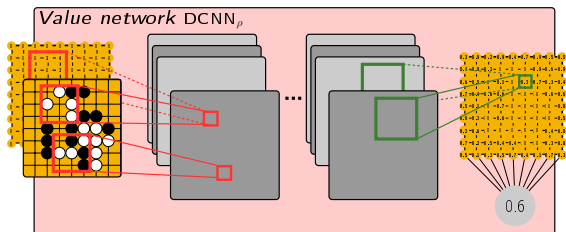




# Apprentissage mixte du réseau de neurone



Self-play + Régession



## Policy network (DCNN $_{\pi}$ )

- App. supervisé
- 30 M positions (*KGS*)  
→ 3 semaines sur 50 GPU

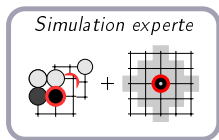
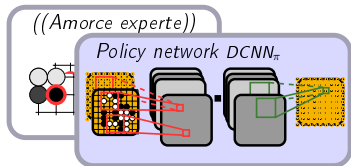
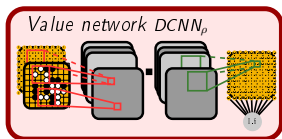
## Value network (DCNN $_{\rho}$ )

- App. par renforcement
- 30 M positions (*self-play*)  
→ 1 semaine sur 50 GPU

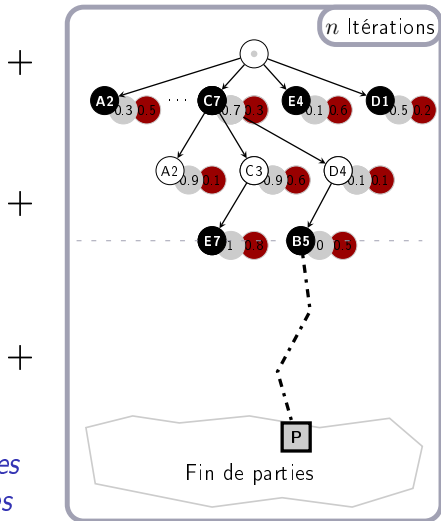
*Amorce supervisée complétée par renforcement*



# Une évaluation complémentaire à MCTS



- 1  $DCNN_\pi$  actions intéressantes
- 2 MCTS séquences probables
- 3  $DCNN_p$  évaluations précises



## Match contre Fan Hui (2 Dan Pro)



### AlphaGo V13

- 5-0 parties longues
- 3-2 parties courtes

## Match contre Fan Hui (2 Dan Pro)



### AlphaGo V13

- 5-0 parties longues
- 3-2 parties courtes

Comparatif	AlphaGo [Silver et al., 2016]	DeepBlue [Campbell et al., 2002]
Hardware	1202 CPU 176 GPU	30 CPU 480 chess CPU
Recherche	$\approx 10^5$ états/s	$10^8$ états/s
Features	48 plans + $10^6$	8000

*Évaluation AlphaGo > Recherche DeepBlue*

# Match contre Lee Sedol (9 Dan Pro)



## AlphaGo V18

- 4-1 parties longues
- 1 M USDollard

## Match contre Lee Sedol (9 Dan Pro)



### AlphaGo V18

- 4-1 parties longues
- 1 M USDollard

### Différences avec AlphaGo V13

- Configuration matérielle semblable
- App. par renforcement complémentaire
- Fan Hui en consultant

*(GPU → TPU)*

*(lequel?)*

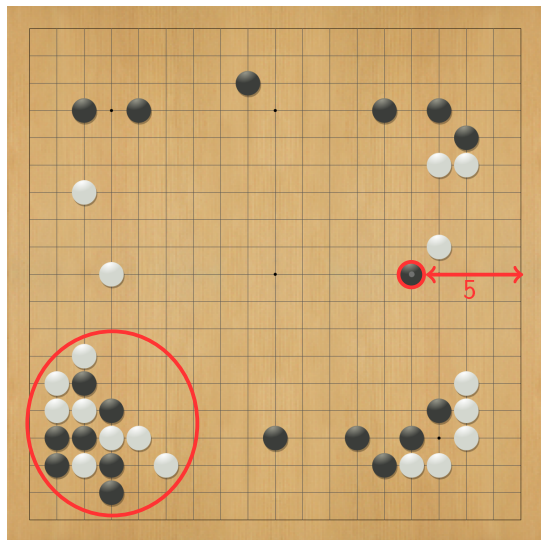
*(secret professionnel)*

**AlphaGo V18 » AlphaGo V13**

## 2ème partie - coup 37 : un coup innovant ?

AlphaGo (Noir)

Lee Sedol (Blanc)



... D'après *Google*

1 / 10000 (humain)

[Hassabis, 2016]

Coup inattendu

App. par Renforc.

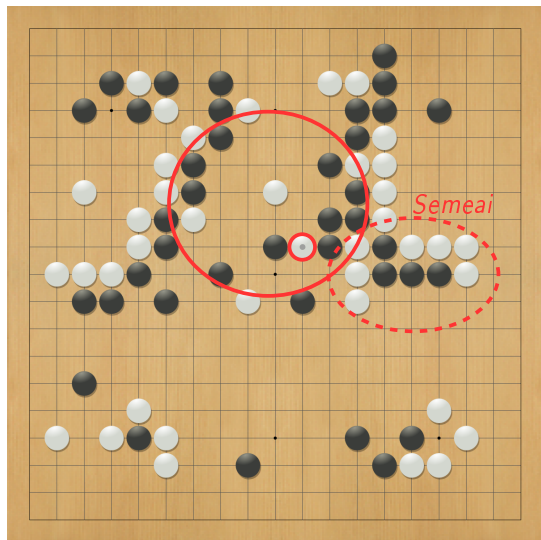
→ *TD-Gammon*

[Tesauro, 1995]

## 4ème partie - coup 78 : une faiblesse ?

AlphaGo (Noir)

Lee Sedol (Blanc)



... D'après *Google*

1 / 10000 (humain)

[Hassabis, 2016]

Position complexe

Sous-Problèmes

→ *Faible MCTS ?*

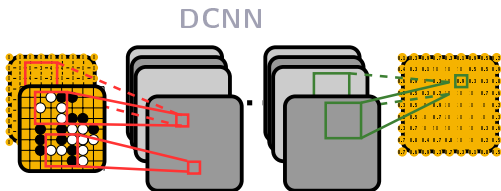
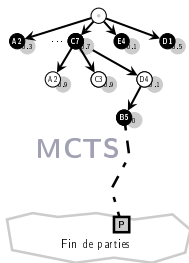
[Huang and Müller, 2013]

## Cinquième partie V

Conclusion



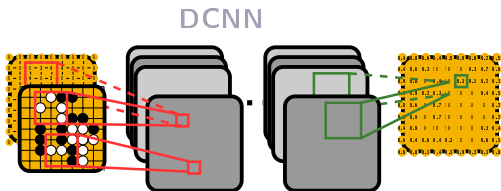
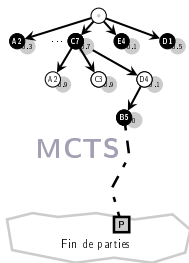
# Pour résumer ...



## Progrès dans l'IA appliquée aux jeux

- Recherche → MCTS
- Évaluation → DCNN et MCTS

## Pour résumer ...



## Progrès dans l'IA appliquée aux jeux

- Recherche → MCTS
- Évaluation → DCNN et MCTS

## Succès de l'apprentissage

- Supervisé (DCNN) et par renforcement (MCTS et DCNN)
- *Off-line* (DCNN) et *on-line* (MCTS)

## Évolutions dans l'IA appliquée aux jeux

- DCNN comme nouveau paradigme → *Go, Atari*
- Intérêt sur d'autres jeux → *GGP, Starcraft*

## Évolutions dans l'IA appliquée aux jeux

- DCNN comme nouveau paradigme → *Go, Atari*
- Intérêt sur d'autres jeux → *GGP, Starcraft*

## Pistes futures pour l'apprentissage

- Aucun *feature* expert → *Uniquement la position des pierres*
- *Transfert learning* → *Capitaliser pour d'autres problèmes*
- Extraction de *feature* → *Que voient les DCNN ?*
- Apprentissage hiérarchique → *Résoudre des objectifs locaux*

## Évolutions dans l'IA appliquée aux jeux

- DCNN comme nouveau paradigme → *Go, Atari*
- Intérêt sur d'autres jeux → *GGP, Starcraft*

## Pistes futures pour l'apprentissage

- Aucun *feature* expert → *Uniquement la position des pierres*
- *Transfert learning* → *Capitaliser pour d'autres problèmes*
- Extraction de *feature* → *Que voient les DCNN ?*
- Apprentissage hiérarchique → *Résoudre des objectifs locaux*

***Merci pour votre attention !***

# Références Bibliographiques I



Allis, L. V. (1994).

*Searching for Solutions in Games and Artificial Intelligence.*  
PhD thesis, Limburg University, Maastricht, Netherland.



Baudiš, P. and Gailly, J. (2012).

**Pachi** : State of the art open source Go program.  
In *Advances in Computer Games*, pages 24–38. Springer-Verlag.



Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012a).  
**A Survey of Monte Carlo Tree Search Methods.**  
*Computational Intelligence and AI in Games, IEEE Transactions on*, 4(1) :1–43.



Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012b).  
**A Survey of Monte Carlo Tree Search Methods.**  
*IEEE Transactions on Computational Intelligence and AI in Games*, 4(1) :1–43.



Brügmann, B. (1993).

**Monte carlo go.**  
In *AAAI Symposium on Games : Playing, Planning, and Learning*.



Campbell, M., Hoane, A. J., and Hsu, F.-h. (2002).

**Deep blue.**  
*Artificial intelligence*, 134(1) :57–83.

# Références Bibliographiques II



Enzenberger, M., Muller, M., Arneson, B., and Segal, R. (2009).

FUEGO - an Open-Source Framework for Board Games and Go Engine Based on Monte-Carlo Tree Search.

*IEEE Transactions on Computational Intelligence and AI in Games*, 2(4) :259–270.



Gelly, S. and Silver, D. (2008).

Achieving master level play in  $9 \times 9$  computer go.

In *Proceedings of AAAI*, pages 1537–1540.



Harré, M., Bossomaier, T., and Snyder, A. (2012).

The Perceptual Cues that Reshape Expert Reasoning.

*Scientific Reports*, 2.



Hassabis, D. (2016).

What we learned in Seoul with AlphaGo.

<https://googleblog.blogspot.fr/2016/03/what-we-learned-in-seoul-with-alphago.html>.



Huang, A. and Müller, M. (2013).

Investigating the Limits of Monte Carlo Tree Search Methods in Computer go.

In *The 8th international conference on Computers and Games (CG2013)*.



Jouppi, N. (2016).

Google supercharges machine learning tasks with TPU custom chip.

<https://cloudplatform.googleblog.com/2016/05/>

[Google-supercharges-machine-learning-tasks-with-custom-chip.html](#).

# Références Bibliographiques III



Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988).

Learning representations by back-propagating errors.

*Cognitive modeling*, 5(3) :1.



Schaeffer, J., Burch, N., Björnsson, Y., Kishimoto, A., Müller, M., Lake, R., Lu, P., and Sutphen, S. (2007).

Checkers Is Solved.

*Science*, 317(5844) :1518–1522.



Schaeffer, J., Müller, M., and Kishimoto, A. (2014).

Als Have Mastered Chess. Will Go Be Next ?



Schraudolph, N. N., Dayan, P., and Sejnowski, T. J. (1994).

Temporal Difference Learning of Position Evaluation in the Game of Go.

*Advances in Neural Information Processing Systems*, pages 817–817.



Segal, R. B. (2011).

On the scalability of parallel UCT.

In *Computers and Games*, pages 36–47. Springer.



Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016).

Mastering the game of Go with deep neural networks and tree search.

*Nature*, 529(7587) :484–489.



Tesauro, G. (1995).

Temporal Difference Learning and TD-Gammon.

*Commun. ACM*, 38(3) :58–68.



# Crédits photographiques I



[1] Lee Jin Man / Associated Press, *extraite de la page*  
<http://www.cbc.ca/news/technology/alphago-ai-lee-sedol-1.3481767>  
(consultée le 10/06/16)



[2] Adam Nadel / Associated Press, *extraite de la page*  
<http://www.wired.com/2012/09/deep-blue-computer-bug/>  
(consultée le 10/06/16)



[3] LadyofHats, *accessible à la page*  
[https://fr.wikipedia.org/wiki/Neurone#/media/File:Complete\\_neuron\\_cell\\_diagram\\_fr.svg](https://fr.wikipedia.org/wiki/Neurone#/media/File:Complete_neuron_cell_diagram_fr.svg)  
(consultée le 10/06/16 - Licence CC0)



[4] danny appel, *accessible à la page*  
<https://pixabay.com/fr/fleur-prairie-violet-printemps-209777/>  
(consultée le 10/06/16 - Licence CC0)



[5] Yaldex.com, *extraite de la page*  
[http://www.yaldex.com/game-development/1592730043\\_ch20lev1sec5.html](http://www.yaldex.com/game-development/1592730043_ch20lev1sec5.html)  
(consultée le 10/06/16)



[6] DeepMind, *accessible de la page*  
<https://deepmind.com/css/images/opengraph/deepmind-logo.png>  
(consultée le 10/06/16 - Tous droits réservés)



[7] Deep Mind Youtube, *extraite de la page*  
<http://electronicdesign.com/memory/defeating-top-go-player-artificial-intelligence-narrows-learning-gap-humans/>  
(consultée le 10/06/16 - Licence Youtube Standard)

# Crédits photographiques II



[8] Lee Jin-Man /Associated Press, *extraite de la page*

[http://www.lemonde.fr/pixels/article/2016/03/12/jeu-de-go-victoire-decisive-de-l-intelligence-artificielle-contre-lee-sedol\\_4881624\\_4408996.html](http://www.lemonde.fr/pixels/article/2016/03/12/jeu-de-go-victoire-decisive-de-l-intelligence-artificielle-contre-lee-sedol_4881624_4408996.html)

(consultée le 10/06/16)