

# **Une toute petite (et incomplète) histoire de l'IA**



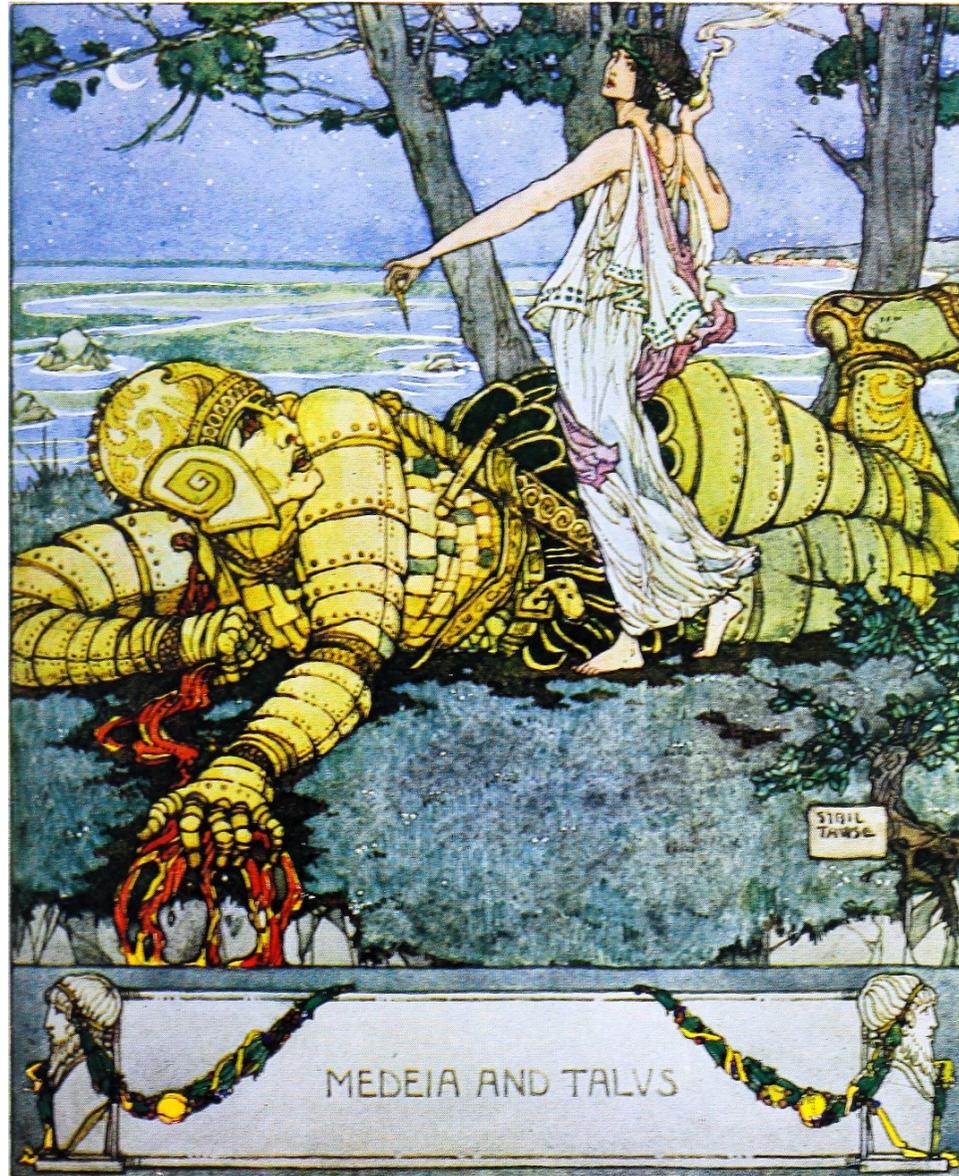


## Homère (L'Iliade 850BC)

- ⌘ Hephaistos (Vulcain), dieu du feu et des forgerons, crée des apprentis mécaniques qui sont intelligents et doués du pouvoir de la parole.
- ⌘ Il créera aussi un homme de bronze, Talos, qui surveille les rives de la Crète et repousse les envahisseurs.



# Homère (L'Iliade 850BC)





## Egypte hellénique (200BC)

- ⌘ Les prêtres égyptiens créent de véritables automates, soit mus par des systèmes hydrauliques, soit mus par des systèmes complexes de leviers et de cordes.
- ⌘ Ils sont utilisés durant des cérémonies religieuses.



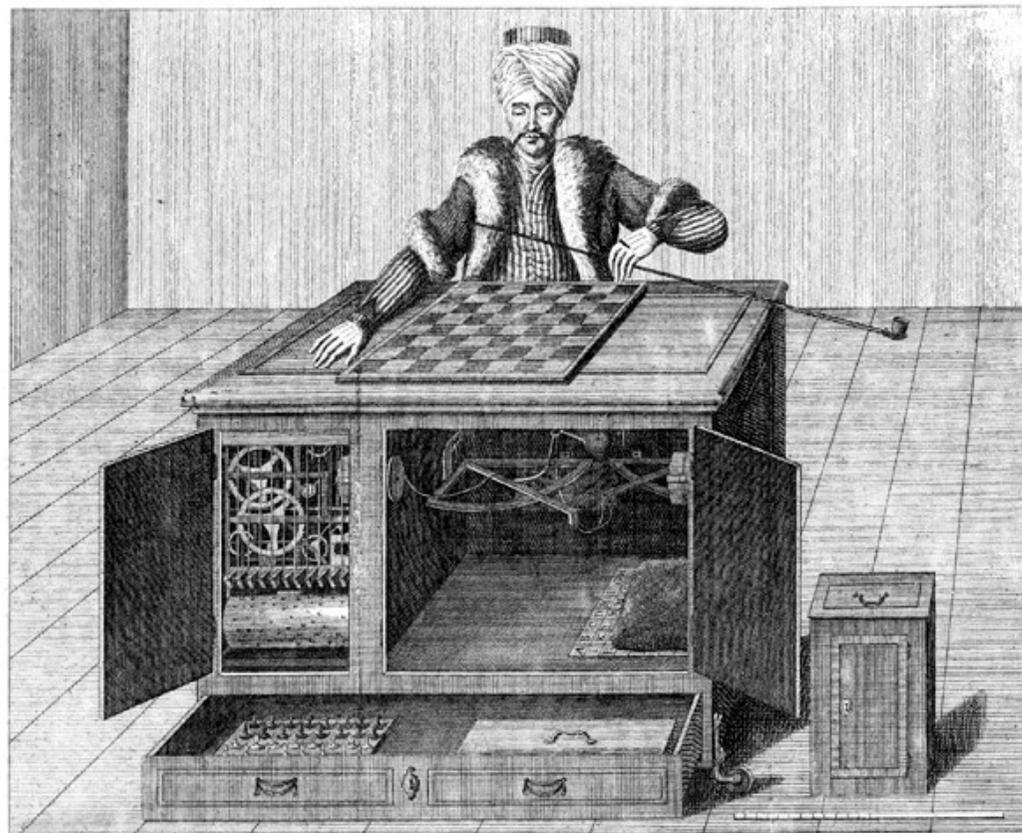
## Le pape Sylvestre II (945-1000)

⌘ Est supposé avoir construit un automate qui pouvait dire « oui » ou « non » lorsqu 'on lui posait une question.



## Le Turk

⌘ Construit en 1770 par Wolfgang von Kempelen



*W. de Kempelen del.*

*Ch. à Mechel, excusé. Basilea.*

*P.G. Pintz, fecit.*

*Der Schach-Spieler, wie er vor dem Spiele gezeiget wird von vorne. Le Joueur d'Échecs, tel qu'on le montre avant le jeu, par devant.*



## Les jeux à la base de l'IA

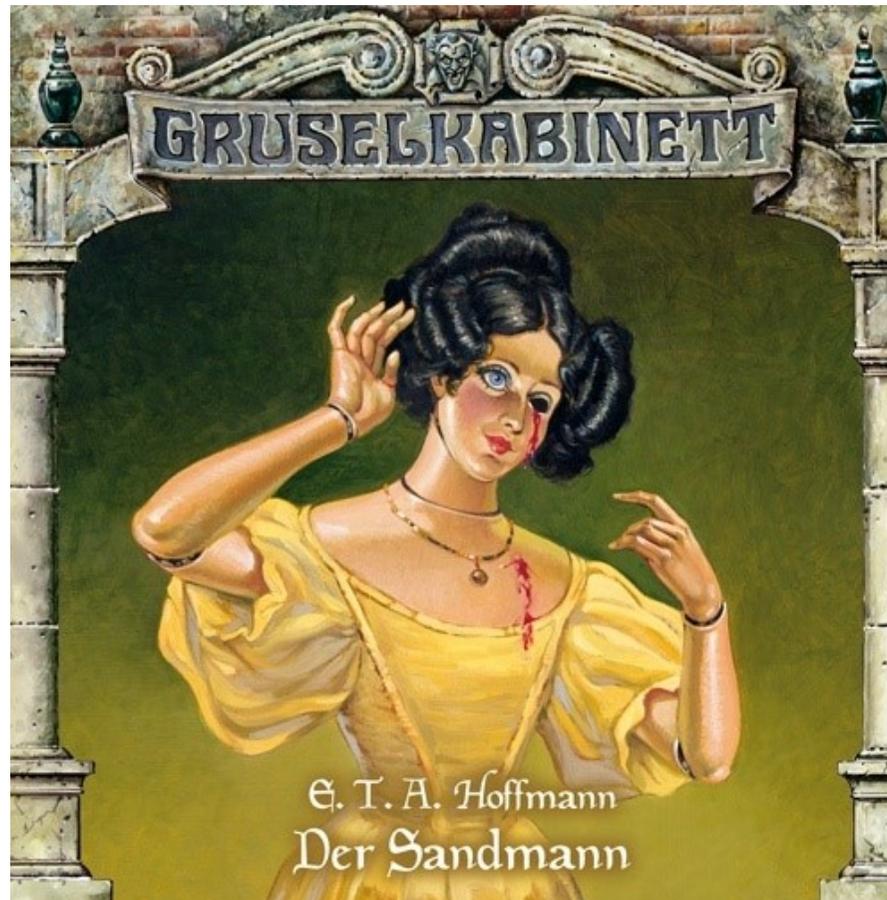
- ⌘ Domaine complètement spécifiable et à information totale
- ⌘ Résultats facilement lisibles: victoire ou défaite
- ⌘ Domaine « représentatif » de l'intelligence humaine
- ⌘ Domaine « populaire » qui attire l'attention...et les financements
- ⌘ Domaine riche et complexe



## Der Sandman (1817)

E.T.A Hoffman

⌘ Met en scène Olimpia, automate créé par son père, professeur de physique





# Frankenstein (1818) Mary Shelley

⌘ Pose le mythe de la création se retournant contre son maître





## L'Eve future (1886)

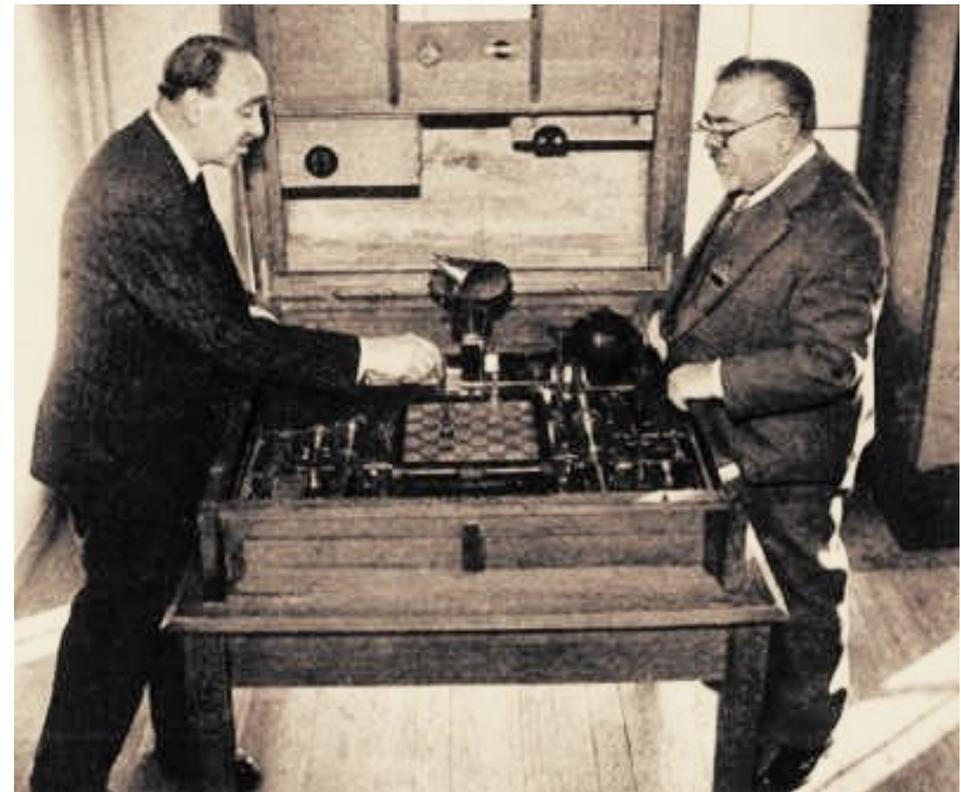
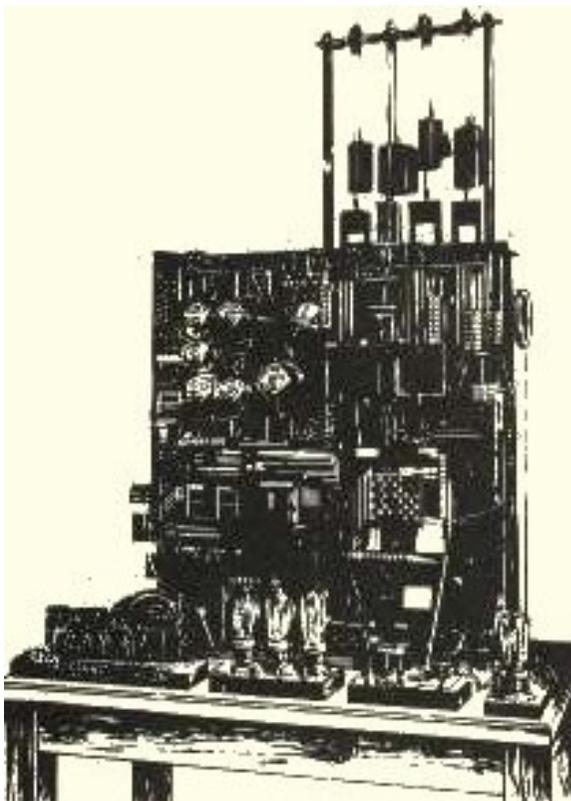
- ⌘ Roman écrit par le comte de Villiers de l'Isle-Adam
- ⌘ Met en scène un andréïde féminin, nommée Eve





## El Ajedrecista (1910)

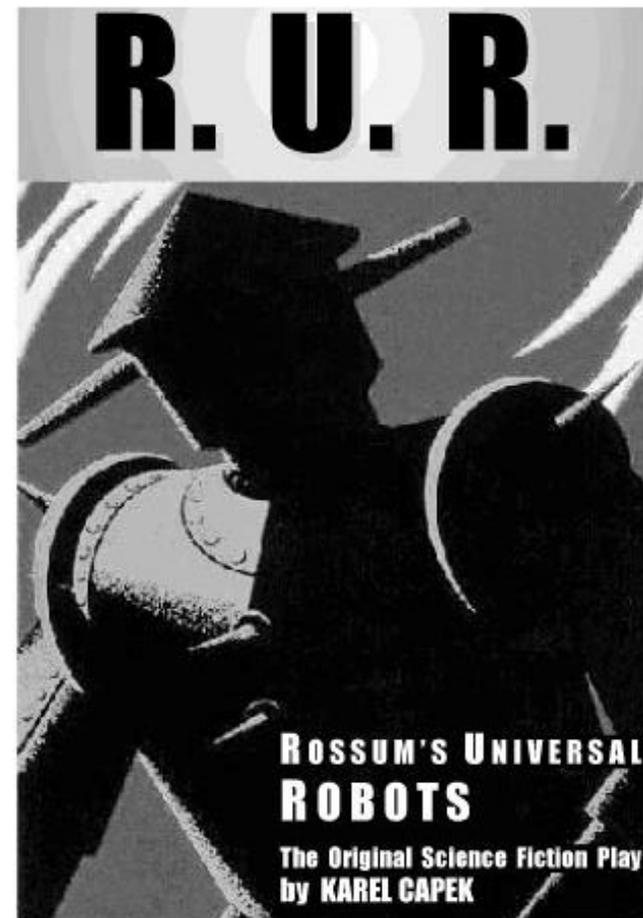
⌘ Leonardo Torres y Quevedo. Jue la finale Roi vs Roi+Tour. En 1920, version magnétique.





## Les robots (1920)

⌘ Karel Kapec  
(tchèque) publie le  
roman Rossum 's  
Universal Robots  
(R.U.R). Robota  
signifie « corvée »  
en tchèque.





## Metropolis (1926)

⌘ Film de Fritz Lang dont un des personnages principaux est une androïde.



## Un vieil argument anti IA

- ⌘ « Les machines sont limitées dans ce qu'elles peuvent calculer, c'est Turing qui l'a montré ».
- ⌘ Ceux qui l'emploient n'ont pas compris grand-chose à la notion de calculabilité, comme nous allons le voir...



## Un peu de théorie: l'infini..

- ⌘ 1638: Galilée remarque que « les nombres peuvent-être associés avec leur carré »
- ⌘ Y a-t-il « autant » de carrés que d'entiers?
- ⌘ Que veut dire « autant »?



## Un peu de théorie: les ensembles

- ⌘ Fin du XIXème siècle: Cantor formalise la notion d 'ensemble, de bijection et d 'équipotence
- ⌘ « Autant » signifie « peut-être mis en relation 1-1 (bijection) avec »
- ⌘ Il reste à trouver les grandes relations d 'équipotence



## Un peu de théorie: dénombrabilité

- ⌘ Premier ensemble étudié: l'ensemble des entiers naturels **N**
- ⌘ Il est équipotent à l'ensemble des entiers relatifs, aux nombres algébriques, aux fractions rationnelles, etc

- ⌘ On peut numérotter les fractions:

	<b>1</b>	<b>2</b>	<b>3</b>
<b>1</b>	1/1	2/1	3/1
<b>2</b>	1/2	2/2	3/2
<b>3</b>	1/3	2/3	3/3



## Un peu de théorie

- ⌘ **N** ainsi que tous les ensembles « numérotables » est dit « dénombrable »
- ⌘ L'ensemble des fonctions de **N** dans **N** est « plus grand » que **N** (on va le voir)
- ⌘ Pour information:
  - ⊞ L'ensemble des fonctions de **N** dans **N** est équipotent à **R** (les nombres réels)
  - ⊞ Existe-t-il des ensembles « entre » **N** et **R** ? C'est une proposition indécidable (dans l'axiomatique classique Zermelo-Frankel de **N** [Cohen 1963])



## Un peu de théorie

- ⌘ Y a-t-il des ensembles plus grand que  $\mathbf{N}$  ?
- ⌘ Supposons que l'on peut numéroter les fonctions de  $\mathbf{N}$  dans  $\mathbf{N}$  :  $f_1, f_2, \dots$
- ⌘ Soit la fonction  $g$  définie par:  $g(i) = f_i(i) + 1$
- ⌘  $g$  est une fonction de  $\mathbf{N}$  dans  $\mathbf{N}$  donc il existe  $n$  tel que:
  - ⌘  $f_n(i) = g(i)$  pour tout  $i$
- ⌘ Pour  $i = n$ , on a  $f_n(n) = g(n) = f_n(n) + 1$
- ⌘ Absurde: cet ensemble n'est pas dénombrable



## Les machines de Turing

- ⌘ Une bande de papier infinie divisée en cases
- ⌘ Un pointeur, qui indique la case active
- ⌘ Un indicateur d'états qui garde en mémoire l'état de la machine
- ⌘ Un programme sous la forme d'un tableau à deux entrées (voir exemple suivant)
- ⌘ Les machines de Turing sont dénombrables (simple à démontrer)



## Les machines de Turing

- ⌘ Programme de la machine effectuant  $f(x)=x+1$
- ⌘ Ex: Si la machine est dans l'état  $z_0$  et que la case contient un 0, elle reste dans l'état  $z_0$ , écrit un 0 et va à droite.

	B	0	1
$z_0$	$(z_1, B, G)$	$(z_0, 0, D)$	$(z_0, 1, D)$
$z_1$	$(z_h, 1, I)$	$(z_h, 1, I)$	$(z_1, 0, G)$



## Thèse de Church-Turing

- ⌘ Une fonction est calculable au sens intuitif du terme si elle est calculable par une machine de Turing.
- ⌘ Cette notion de calculabilité est équivalente à celle définie en utilisant les fonctions récursives générales de Church.



## Argument anti IA

- ⌘ Les machines de Turing sont dénombrables
- ⌘ L'ensemble des fonctions de  $\mathbb{N}$  dans  $\mathbb{N}$  n'est pas dénombrables
- ⌘ Donc un calculateur ne peut pas tout faire
- ⌘ .....
- ⌘ Sauf qu'un cerveau humain ne fait pas mieux...



## Un exemple de problème non calculable

- ⌘ Le problème de Post:
- ⌘ Soit les deux listes suivantes:
  - ⌘ (aabb, ab, aab)
  - ⌘ (bb, abaa, b)
- ⌘ Peut-on construire une liste d'indices telle que la concaténation des éléments de la première liste est égale à la concaténation des éléments de la seconde?



## Solution sur un cas particulier du problème de Post

⌘ Reprenons les deux listes précédentes:

☒ (aabb, ab, aab)

☒ (bb, abaa, b)

⌘ La solution est (2,3,2,1)

☒ Liste 1:  $ab(2) aab(3) ab(2) aabb(1) \Rightarrow abaababaabb$

☒ Liste 2:  $abaa(2) b(3) abaa(2) bb(1) \Rightarrow abaababaabb$

⌘ On peut prouver qu'il est impossible de trouver un algorithme résolvant le problème de Post dans le cas général



## Le problème de Post: quelques enseignements

⌘ Le problème de Post ne peut se résoudre (dans le cas général) car:

- ☑ Si le problème a une solution, l'algorithme la trouvera bien en un temps fini (il suffit d'imaginer un algorithme énumérant toutes les listes possibles en commençant par les listes de taille 1, puis 2, etc...)
- ☑ En revanche si le problème n'a pas de solution, l'algorithme est incapable de dire en un temps fini qu'il n'y a pas de solution (ce résultat est démontrable).



## Le problème de Post: quelques enseignements

- ⌘ Le problème de l'arrêt du calcul n'est pas lié à la structure d'un calculateur. Un être humain aura exactement le même type de difficulté
- ⌘ L'argument de la « dénombrabilité » des machines de Turing ne tient pas pour réfuter la possibilité d'une IA...



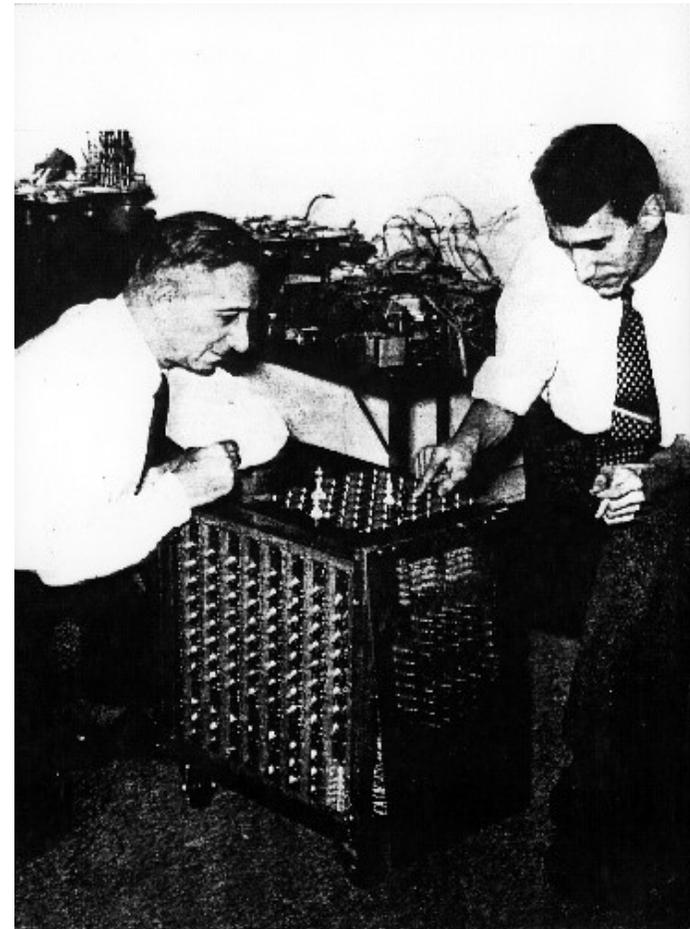
## Les débuts de l'IA

- ⌘ 1945-1955: la préhistoire
- ⌘ De nombreux spécialistes du décryptage se lancent dans la traduction automatique
- ⌘ Débuts enthousiastes
- ⌘ Prédications plus qu'optimistes...
- ⌘ Echecs cuisants



## La machine de Shannon (1950)

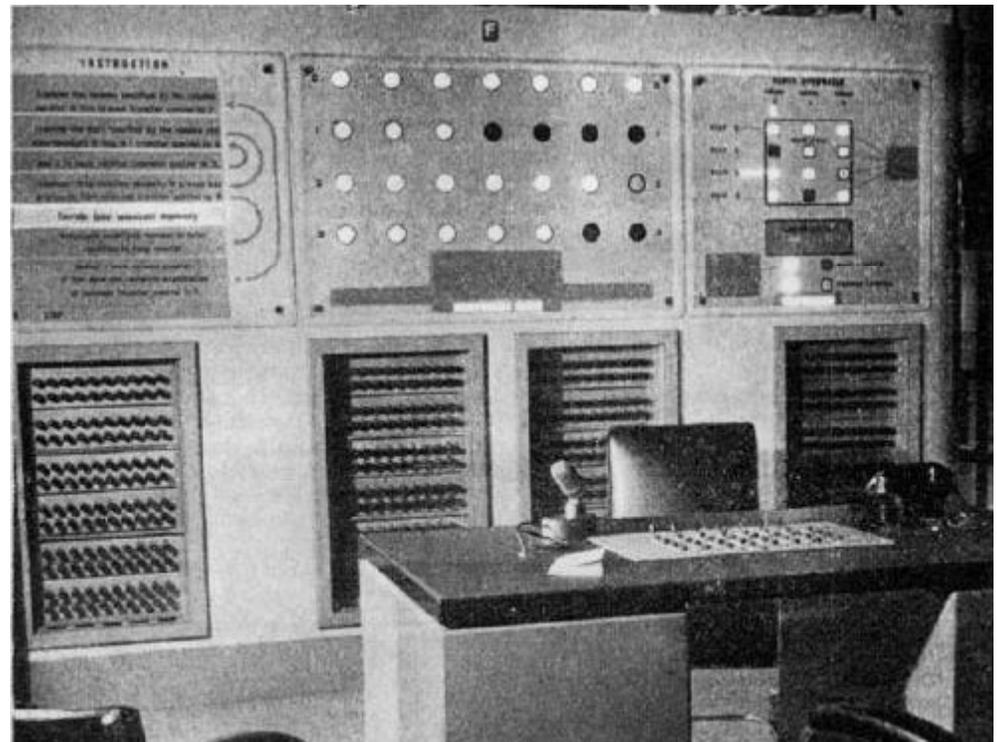
⌘ Construite par Claude Shannon, elle est capable de jouer des fin de parties comportant 6 pièces.





## Le nimrod (1951)

- ⌘ Construit par Ferranti pour une exposition.
- ⌘ Joue parfaitement au jeu de Nim.
- ⌘ 6Kw, 40m<sup>3</sup>





## Le turochamp (1952)

- ⌘ En 1952, Alan Turing construit un jeu d'instructions (le turochamp) qui constitue le premier « programme » d'échecs existant. Faute de machine capable de l'exécuter, le programme sera simulé par Turing dans une partie contre un étudiant, Alick Glennie.



## Le « test de Turing »

- ⌘ Test supposé définir « l'intelligence ».
- ⌘ Un ordinateur peut-il se faire passer pour un être humain en dialoguant avec lui à travers un écran et un clavier?
- ⌘ The « chinese room argument » (John Searle, 1980): la simulation de l'intelligence est-elle de l'intelligence?



## Qu'est ce que l'intelligence?

- ⌘ Le règne animal est d'une incroyable continuité, de l'amibe aux primates.
- ⌘ Nous sommes (presque) tous d'accord pour dire qu'un chimpanzé capable de communiquer par la langue des signes est « intelligent », alors qu'une amibe ne l'est pas.
- ⌘ => Où commence l'intelligence?



## Qu'est ce que l'intelligence?

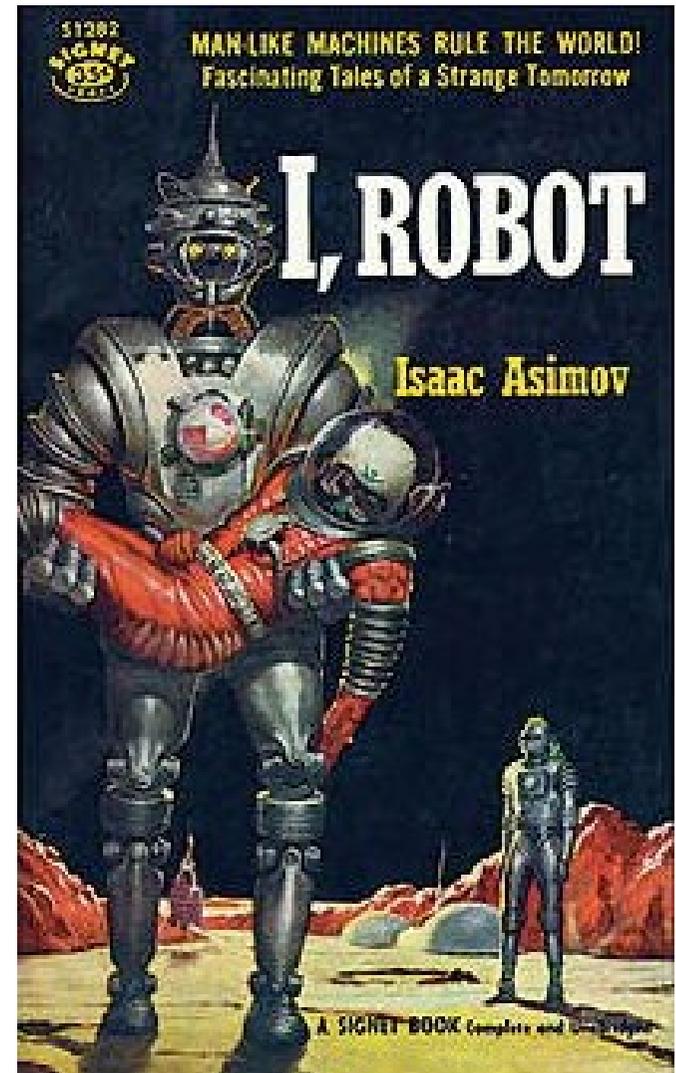
- ⌘ « Are we smart enough to know how smart animals are? » (Frans de Waal)
- ⌘ Certains animaux exhibent des capacités cognitives supérieures aux nôtres dans certains domaines.
- ⌘ Sont-ils plus intelligents que nous?
- ⌘ => La définition de l'intelligence n'est pas « unimodale », on ne peut pas définir un « ordre total »

# I, Robot

⌘ Parution en 1950 de *I, Robot* d'Isaac Asimov, recueil de nouvelles (1939-1950)

⌘ Fixe les « 3 lois de la robotique »

- ☒ A robot may not injure a human being or, through inaction, allow a human being to come to harm.
- ☒ A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.
- ☒ A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.





**Robby the Robot**

⌘ 1956: Forbidden Planet



## Semi Automatic Ground Environment (SAGE)





## IA: les vrais débuts (55-70)

### ⌘ 1956: Dartmouth College Conference

#### ☑ Participants prévus:

☒ Marvin Minsky, Julian Bigelow, D.M. Mackay, Ray Solomonoff, John Holland, John McCarthy, Claude Shannon, Nathaniel Rochester, Oliver Selfridge, Allen Newell, Herbert Simon.

#### ☑ D'autres viendront dont:

☒ John Nash, Mc Culloch



⌘ “We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves.”



## IA: les vrais débuts (55-70)

- ⌘ 1956: the Logic Theorist (Newell, Simon, Shaw): démonstrateur de théorèmes
- ⌘ 1957: General Problem Solver (Newell, Simon, Shaw)
- ⌘ 1958: Checkers : jeu de « dames » (Samuel)
- ⌘ 1965: DENDRAL: système expert de chimie organique (Feigenbaum, Buchanan, Lederberg, Djerassi)



## IA: les vrais débuts

⌘ Simon déclare (1958) qu'avant dix ans:

- ☑ Un programme d'échecs sera champion du monde
- ☑ Un programme de démonstration automatique démontrera un théorème important
- ☑ « Les machines seront capables d'ici vingt ans de faire le travail de n'importe quel humain. »



## IA: les vrais débuts

⌘ Marvin Minski (1959):

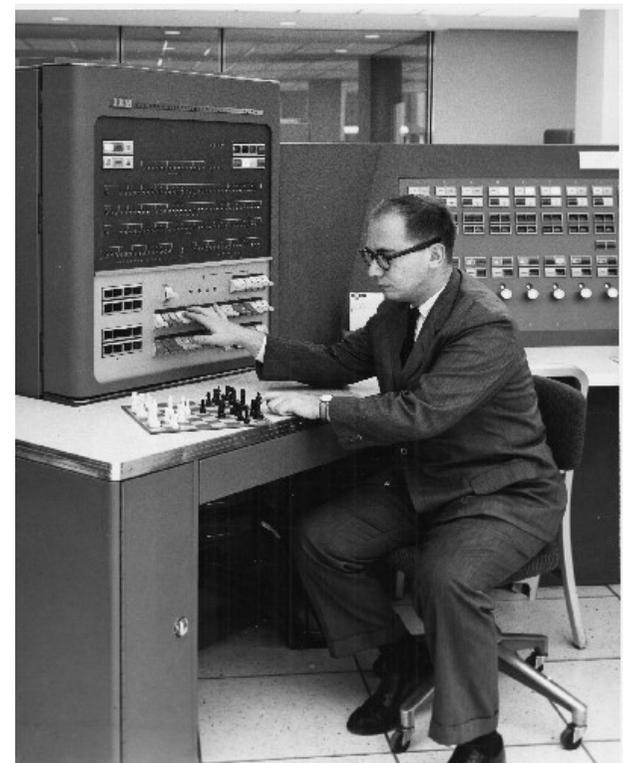
☑ « D'ici une génération, le problème de la création d'une intelligence artificielle sera résolu. »

⌘ Arthur Samuel reste pessimiste...



## The Bernstein Chess Program (1957)

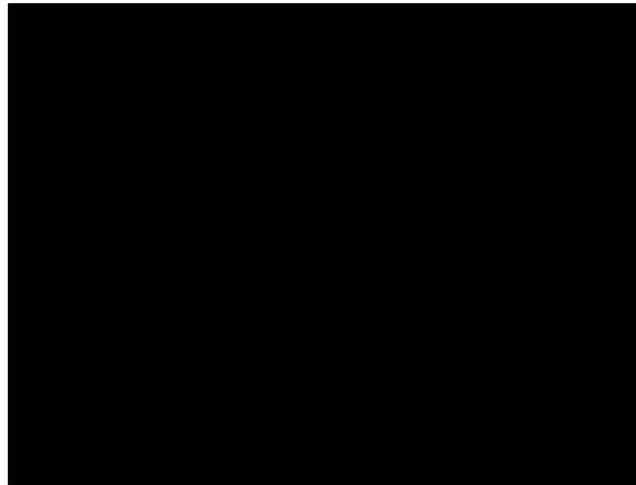
- ⌘ Écrit par Alex Bernstein sur un IBM 704
- ⌘ Premier vrai programme d'échecs
- ⌘ Calcule 4 niveaux en minimax en 8 minutes, avec de l'élagage (Shannon B)



Alex Bernstein



# The Bernstein Chess Program





## NSS (Newell, Simon and Shaw) 1958

- ⌘ « Goal directed »
- ⌘ Reste extrêmement faible
- ⌘ Shannon type B





## Daisy Bell by The IBM 7094 (1961)

⌘ Premier exemple de synthèse vocale



## IA: « connexionisme » et « apprentissage »

- ⌘ Années 40/50: McCulloch, Pitts et Hebb posent les premières idées de réseaux de neurones artificiels et les premières règles d'apprentissage, dans le sillage de la cybernétique inventée par Norbert Wiener.
- ⌘ Rosenblatt (1957) introduit le perceptron, premier système artificiel qui apprend même si certains exemples sont erronés.



## Les réseaux de neurones

⌘ Warren McCulloch and Walter Pitts: *A Logical Calculus of Ideas Immanent in Nervous Activity (1942)*:

☒ "Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means or nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes."

⌘ D. O. Hebb : *The Organization of Behavior (1949)*:

☒ « Let us assume that the persistence or repetition of a reverberatory activity (or "trace") tends to induce lasting cellular changes that add to its stability.[...] When an axon of cell *A* is near enough to excite a cell *B* and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that *A*'s efficiency, as one of the cells firing *B*, is increased."

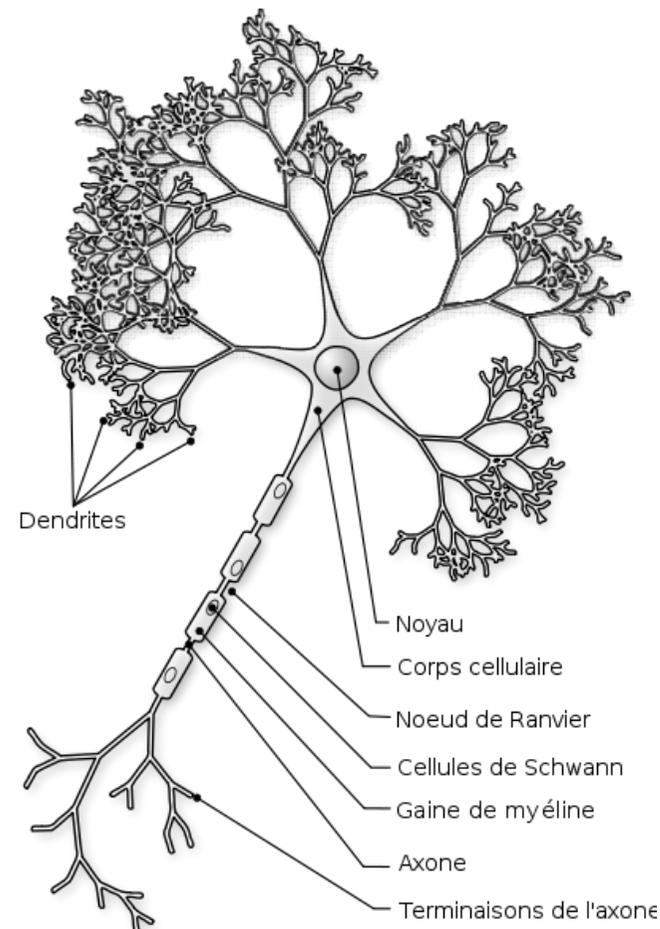
⌘ F. Rosenblatt : *The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain (1958)*

⌘ P. Werbos : *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences (1974)*

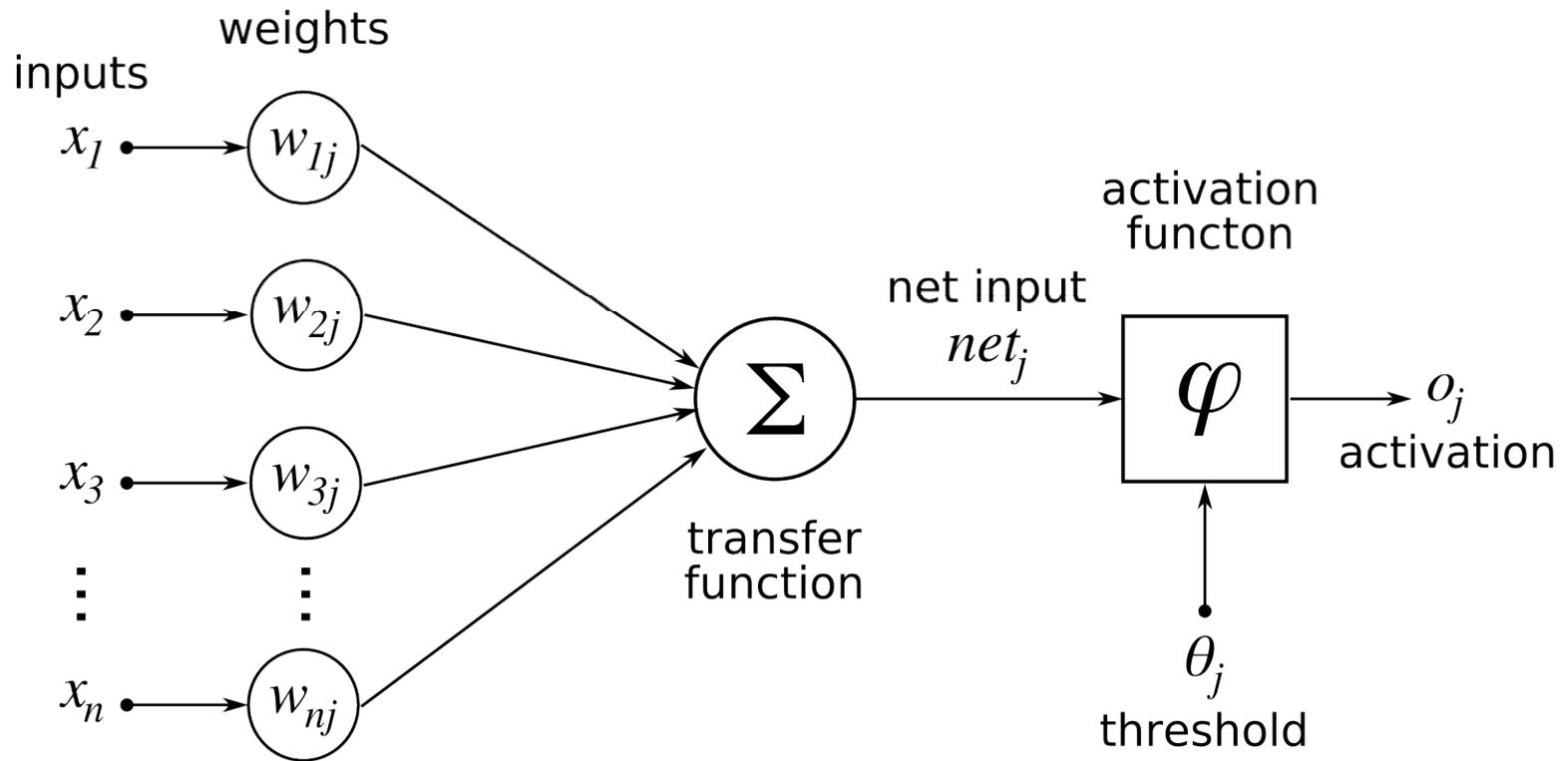
☒ Backpropagation Algorithm

## Le modèle physiologique du neurone

- Les dendrites véhiculent l'information vers le neurone
- Les synapses sont les zones de liaison entre deux neurones (entre l'axone de l'émetteur et la dendrite du récepteur).
- Les synapses peuvent être de nature électriques ou chimiques.
- Les synapses peuvent être excitatrices ou inhibitrices.
- La réponse du neurone est de la forme « tout ou rien » relativement à la valeur de ses entrées par rapport à un seuil.



# Les réseaux de neurones





## Règle de propagation

⌘ La propagation est la composition d'une multiplication matricielle et d'une fonction d'activation:

$$\boxtimes N_j = \sum W_{ij} X_i$$

$$\boxtimes O_j = f(N_j + \theta_j)$$

⌘ Dans le perceptron de Rosenblatt, la fonction  $f$  est de la forme:

$$\boxtimes f(x) = 1 \text{ si } x > 0$$

$$\boxtimes f(x) = 0 \text{ sinon}$$

⌘ Rosenblatt développe une règle simple permettant d'apprendre automatiquement les poids



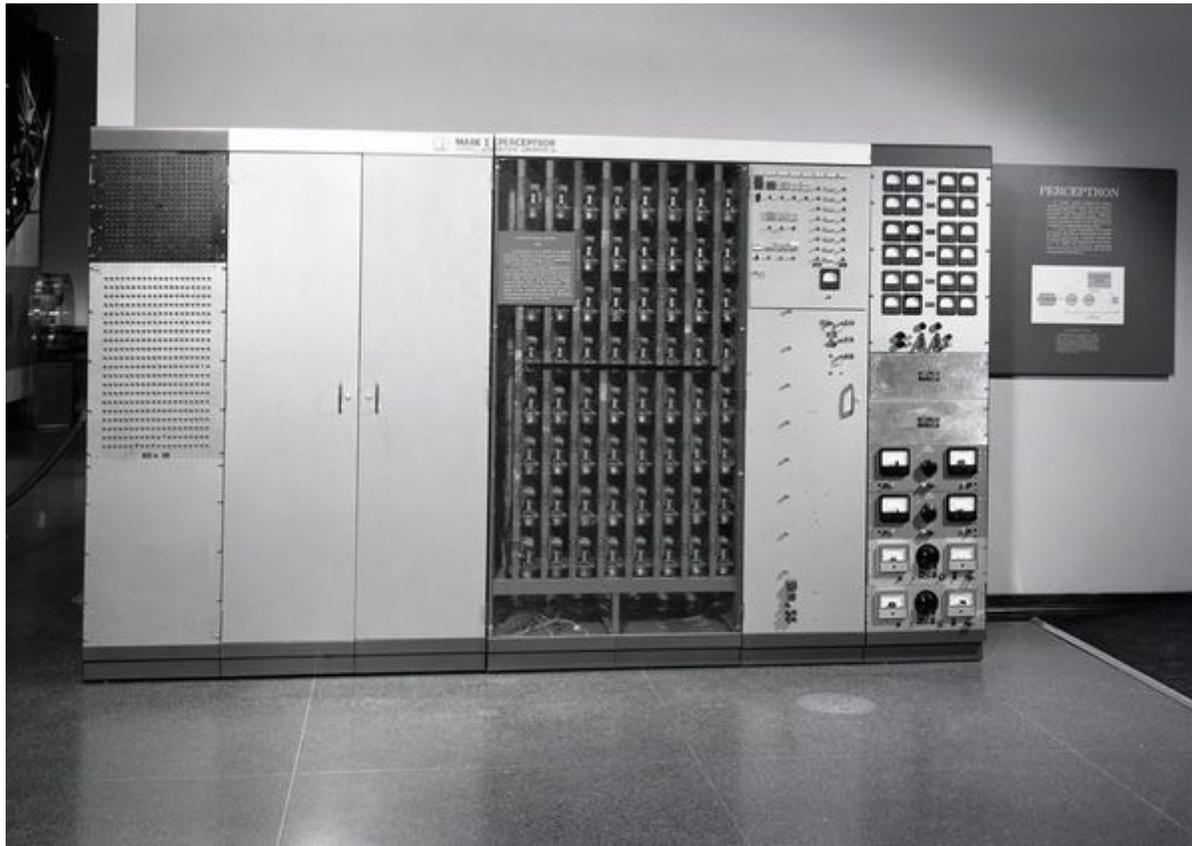
## Apprentissage supervisé

- ⌘ Le réseau est entraîné sur un certain nombre d'exemples
- ⌘ Les sorties calculées par le réseau sont comparées avec les sorties attendues.
- ⌘ Les poids du réseau sont modifiés de façon à rapprocher les sorties calculées des sorties attendues
- ⌘ Mais le perceptron est limité dans ce qu'il peut apprendre.



## The Mark I perceptron machine

- ⌘ Le perceptron est une machine « physique » connecté à 400 photo-récepteurs pour faire de la reconnaissance d'images:





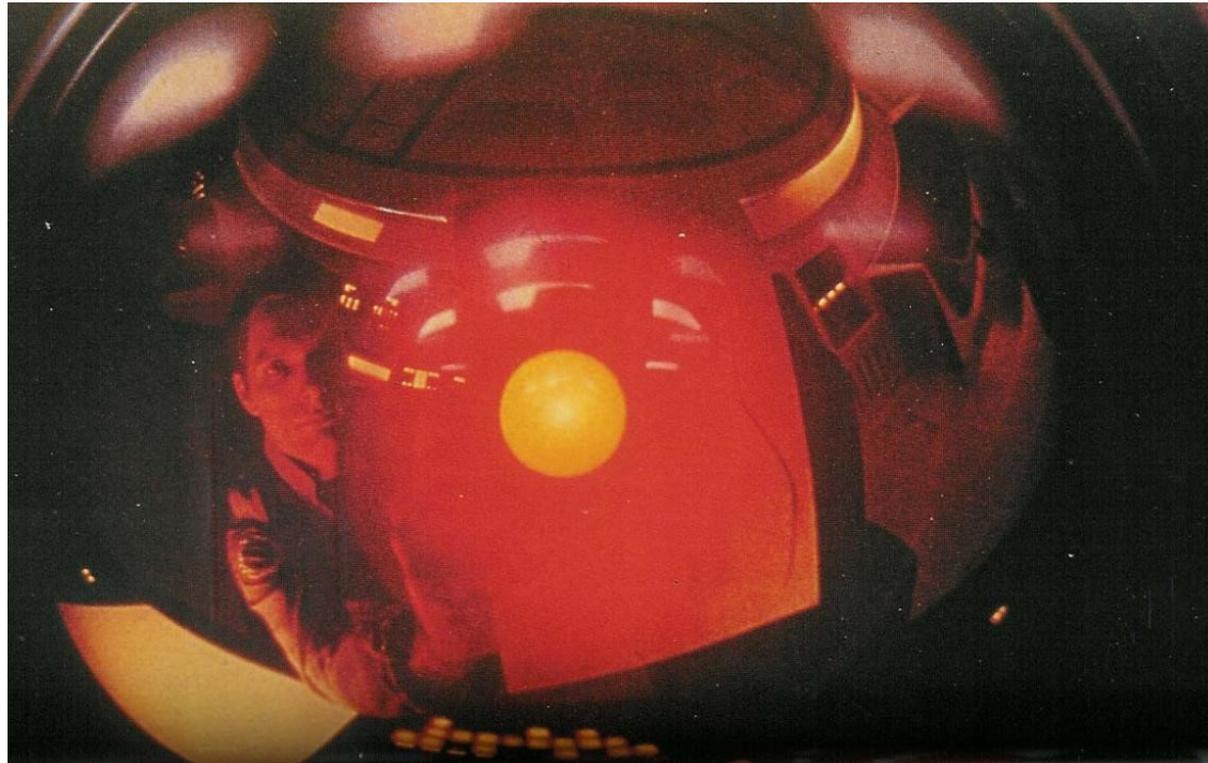
## IA: cognitifs et connexionnistes

- ⌘ L'approche connexionniste est une approche « boîte noire ».
- ⌘ Elle déplait fortement à la communauté « cognitive ».
- ⌘ En 1969, Minsky et Pappert montrent les limites du perceptron linéaire simple couche, et massacrent la recherche dans ce domaine pour vingt ans.
- ⌘ Les problèmes personnels entre Wiener d'une part et McCulloch et Pitts d'autre part affaiblissent le domaine. Pitts meurt en 1969, comme McCulloch.
- ⌘ Les techniques d'apprentissage ne referont surface que vers la fin des années 80.



## **2001, A Space Odyssey (Kubrick, 1968)**

Film très documenté qui reprend beaucoup de ce que certains scientifiques pensaient à l'époque





## 2001, the HAL computer

⌘ H : Heuristically programmed

⌘ AL: ALgorithmic computer

⌘ Référence (involontaire?, hum...) à IBM

⌘ (H->I, A->B, L->M)



## IA: 1970-1980

- ⌘ Au milieu des années 70, de nombreuses critiques vont s'élever contre l'IA « forte ».
- ⌘ Peu de résultats concrets.
- ⌘ Les financements s'effondrent.
- ⌘ C'est le premier « AI winter »



## IA: spécialisation / « professionalisation » 70-80

⌘ Séparation de l'IA en plusieurs branches (qui couvrent quasiment la totalité de l'informatique...):

- ☑ Compréhension du langage
- ☑ Modélisation et résolution logique
- ☑ Démonstration automatique
- ☑ Jeux
- ☑ Systèmes experts
- ☑ Perception
- ☑ Apprentissage
- ☑ Etc, etc...



## IA: la spécialisation

- ⌘ Les années 70-80 voient le début de la controverse pragmatiste/cognitif
- ⌘ Les pragmatistes s'intéressent avant tout au résultat et se moquent de la méthode (la force brute en est un exemple).
- ⌘ Les cognitifs veulent reproduire le raisonnement humain
- ⌘ Cette guerre de religion fera énormément de dégâts.



## IA 80-90: les années « système expert »

- ⌘ Arrivée des japonais et du projet « 5ème génération »
- ⌘ L'IA « cognitive » est à la mode:
  - ☑ Systèmes experts
  - ☑ Logique floue (des appareils photos aux machines à laver...)
  - ☑ Nombreuses équipes de recherche
- ⌘ Les approches liés à l'apprentissage et au connexionnisme restent rejetées au second plan.



**IA: 1992-2010**

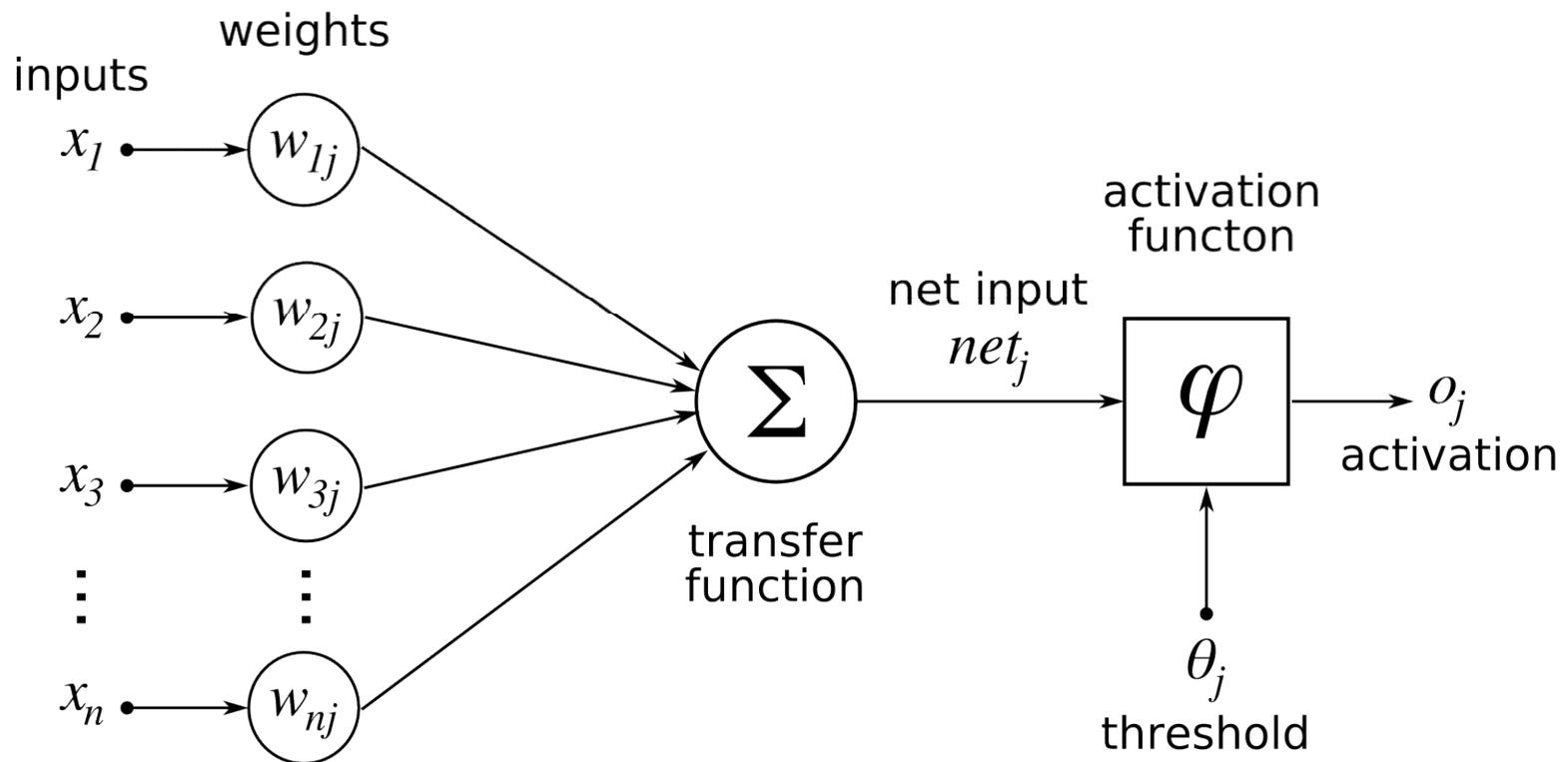
## ⌘ 92-: le retour de balancier

- ☒ Deuxième «AI winter »
- ☒ L'IA cognitive s'est discréditée à nouveau par des déclarations pompeuses et bien trop optimistes
- ☒ Le projet 5ème génération se termine en 92 dans la confusion
- ☒ Le terme d 'IA lui-même commence à disparaître
- ☒ Malheureusement, de nombreuses excellentes équipes d'IA « formelle » sont entraînées dans la « chute », en raison de la malhonnêteté et du sectarisme d'une partie de la communauté cognitive lors des années précédentes



- ⌘ Paradoxalement, certaines grandes réalisations de l'IA se produisent dans les années 90, mais pas grâce aux techniques d'IA « cognitives », et on évite de parler d'IA...
- ⌘ L'approche basée sur l'apprentissage et les méthodes connexionnistes et évolutionnaires émerge après 25 ans passée dans l'obscurité:
  - ⌘ P. Werbos : Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences (1974) => algorithme de rétropropagation du gradient.
  - ⌘ 1982: Hopfield (réseaux récurrents)
  - ⌘ 1986: Rumelhart, Werbos, LeCunn (perceptrons multi-couches)

Les réseaux de neurones « modernes »





## Règle de propagation

⌘ La propagation est toujours la composition d'une multiplication matricielle et d'une fonction d'activation:

⊞  $N_j = \sum W_{ij} X_i$

⊞  $O_j = f(N_j + \theta_j)$

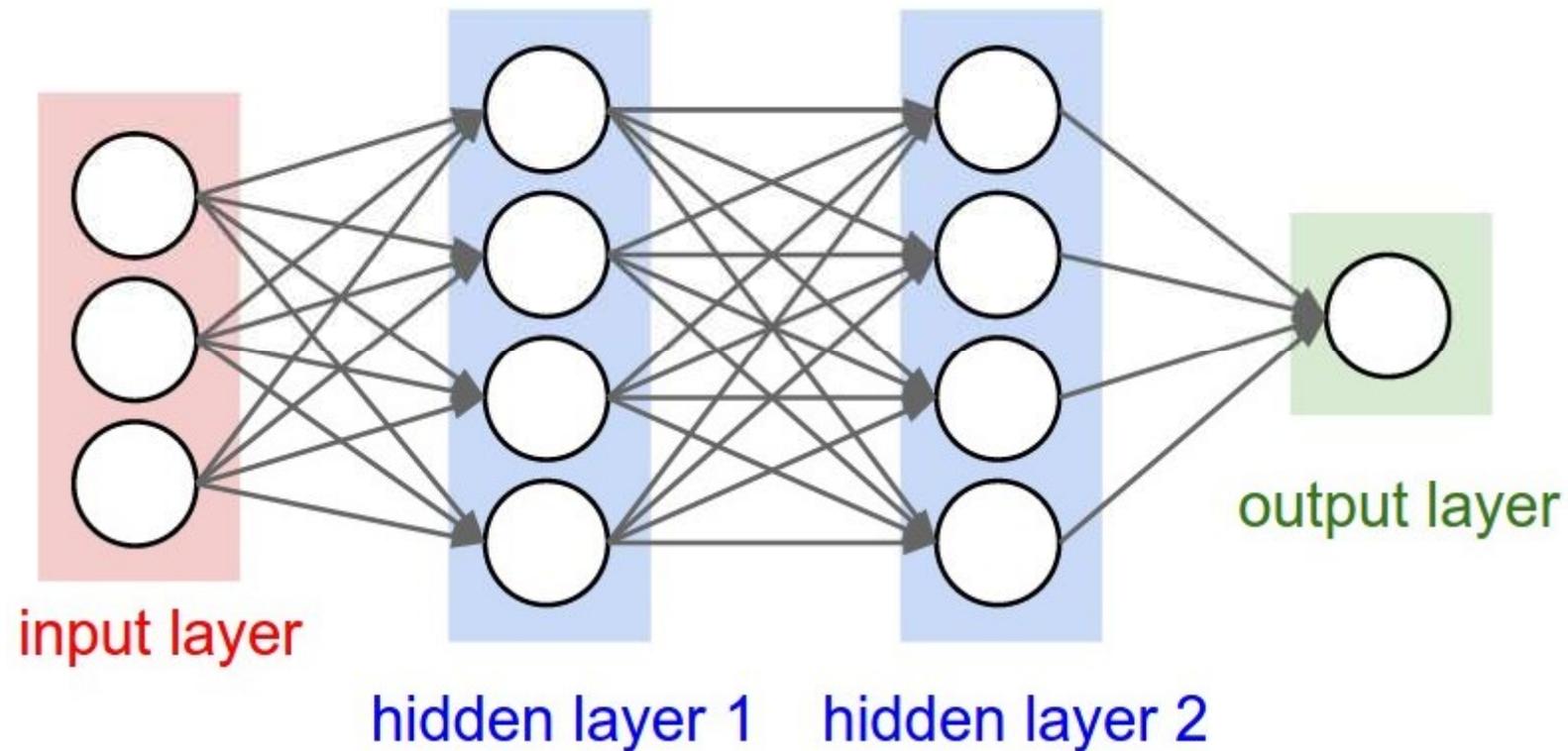
⌘ Mais maintenant la fonction  $f$  est de la forme:

⊞  $f(x) = \text{th}(x)$

⊞ Plus tard:  $f(x) = \text{Relu}(x)$

⊞ ....

Et on met plusieurs couches...



- Apprentissage supervisé: backprop
- Question: comment organiser les couches intermédiaires?



## Règle d'apprentissage (Werbos 1974)

⌘ La règle d'apprentissage la plus classique est connue sous le nom de « rétropropagation du gradient »:

$$\Delta W_{ji} = \eta \delta_j O_i$$

$$\delta_j = (T_j - O_j) f'_j(N_j)$$

$$\delta_j = f'_j(N_j) \sum_k \delta_k W_{kj}$$



**D'immenses progrès sont aussi faits dans le domaine des jeux**

⌘ Les échecs

⌘ Othello

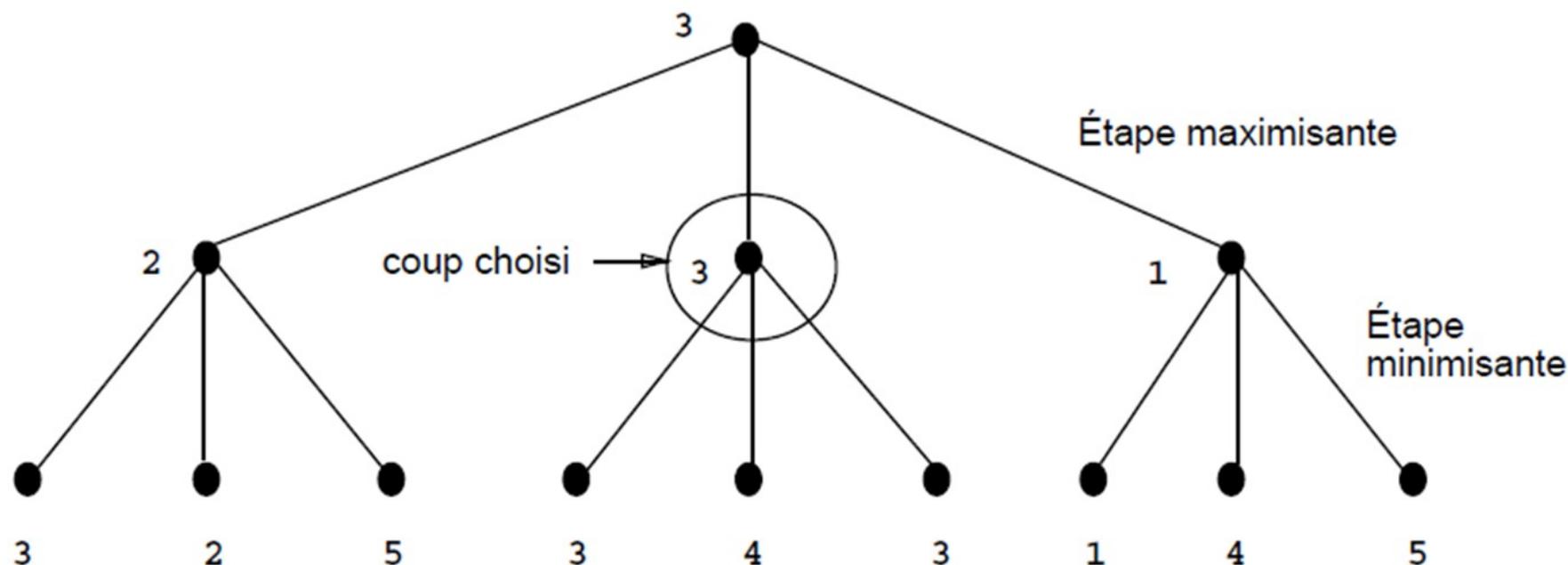
⌘ Les checkers

⌘ Le bridge

⌘ En fait, tout sauf le Go...

## Minimax

- En théorie des jeux, les algorithmes « standards » utilisés sont dits minimax (avec toutes leurs variantes). Ces algorithmes se sont raffinés depuis le début des années 70, mais restent basés sur la force brute, des heuristiques d'élagage, et une fonction d'évaluation qui calcule la valeur de chaque position.





## DeepBlue (1997)

- ⌘ Supercalculateur fabriqué « sur mesure » par IBM
- ⌘ 200 millions de positions par seconde (force brute)
- ⌘ Premier ordinateur à battre un champion du monde humain avec un contrôle de temps standard sur 6 parties





## « Chess in the Stratosphere »

⌘ Les meilleurs programmes actuels:

☑ Stockfish: open source

☑ Komodo: close source

⌘ ELO >? 3300 sur un Q6600 (quad core)  
2.4GHz

⌘ ELO sur un processeur plus récent  
probablement >? 3400

# IRIT Othello

- ⌘ Les programmes sont aujourd'hui beaucoup plus forts que les meilleurs joueurs humains
  - ☑ 1997: Logistello bat Takeshi Murakami 6-0
- ⌘ Les développements nouveaux ont quasiment cessés.





## Awele (Awari, Awale)

⌘ 2002: Awele is solved (Romein-Bal). La partie est nulle en cas de jeu parfait.



# IRIT Les checkers

- ⌘ 2007: l'équipe de Schaeffer et Chinook prouvent que le jeu de checkers est nul en cas de jeu parfait.
- ⌘ Les checkers sont résolus



# IRIT Le bridge

- ⌘ Il est difficile d'évaluer le niveau d'un programme de façon objective.
- ⌘ Le meilleur programme actuel (GIB/Matthew Ginsberg) est sans doute moins fort que les meilleurs humains, mais il est meilleur que l'immense majorité des joueurs.
- ⌘ Il utilise des méthodes de force brute (Double Dummy Solver et Monte-Carlo)





## IA 2010: il nous reste le Go

⌘ Considéré comme inaccessible aux ordinateurs  
car:

- ☒ Facteur de branchement trop important
- ☒ Notion d'influence difficile à traduire informatiquement
- ☒ En 2010, les meilleurs programmes sont (très) loins du niveau des professionnels

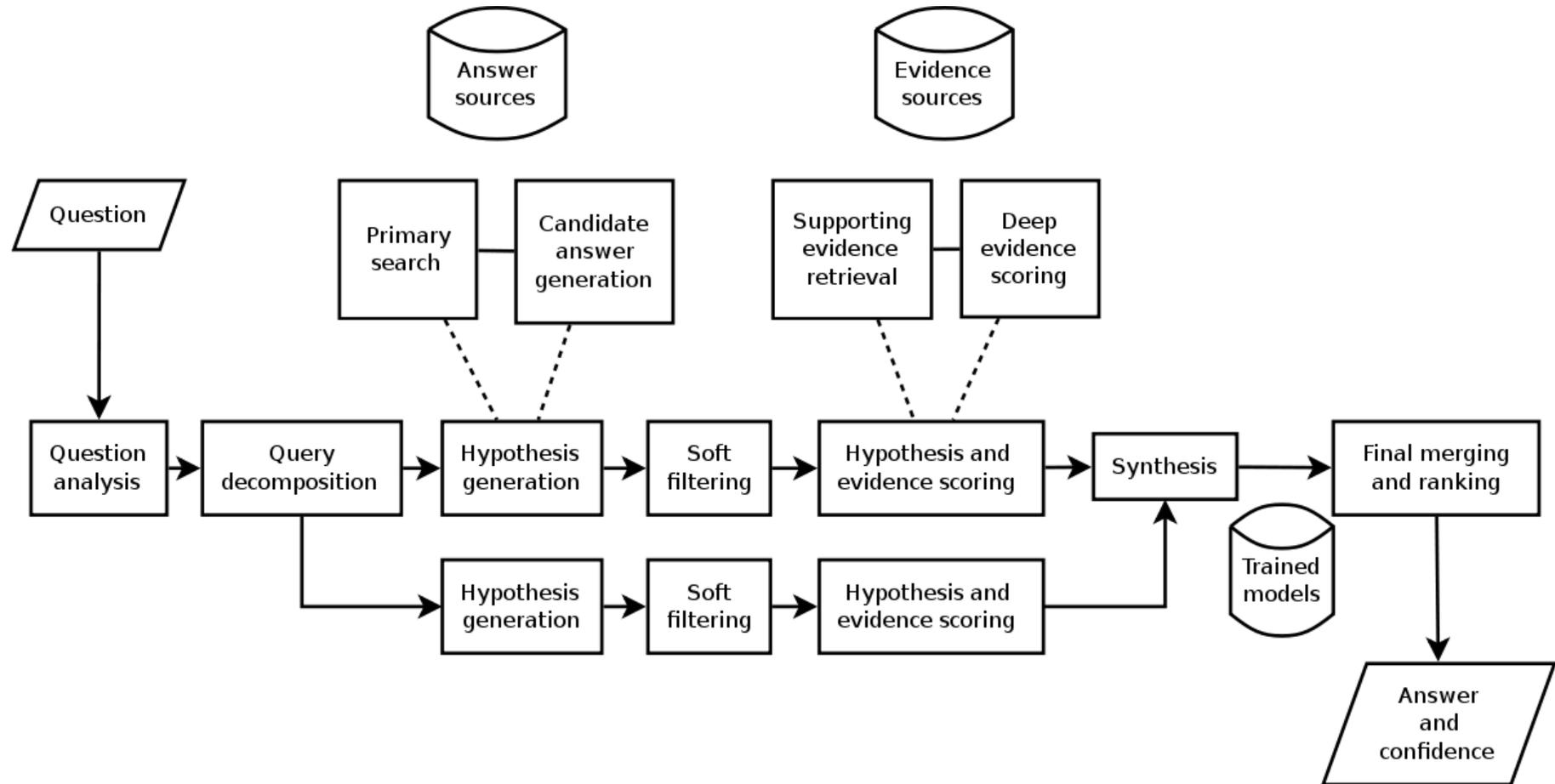


## IBM Watson (2011)

- ⌘ Système spécialisé dans l'analyse de corpus en langage naturel, et capable de répondre en langage naturel à des questions posées en langage naturel.
- ⌘ En 2011, Watson remporte un match exceptionnel en 3 manches de Jeopardy contre les meilleurs humains.



# IBM Watson





- ⌘ 2012: « Utilization Management » pour le traitement du cancer du poumon (Memorial Sloan Kettering Cancer Center, WellPoint)
- ⌘ 2016: collaboration avec Manipal Hospitals en Inde comme "expertise" sur le traitement du cancer
- ⌘ Nombreuses autres applications annoncées, mais les résultats sont peu clairs
- ⌘ Watson est plus un "framework" qu'un véritable système



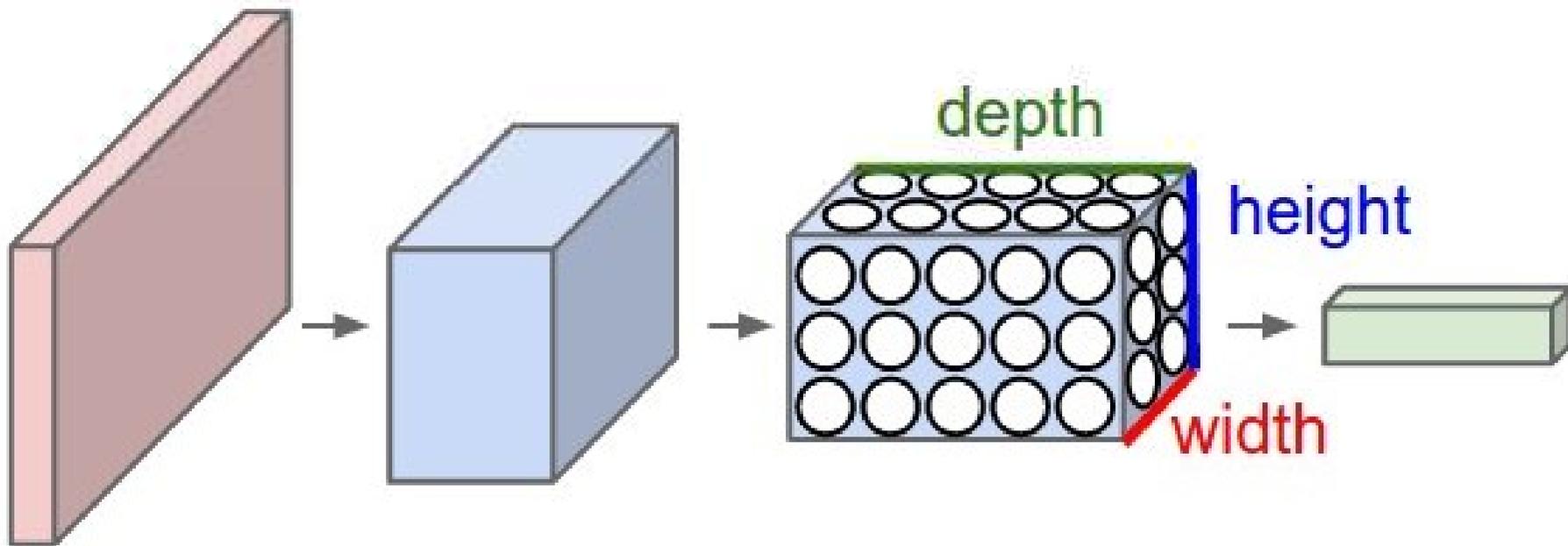
## DNN et apprentissage supervisé

- La construction du réseau est un art qui se perfectionne
  - 1998: LeNet par LeCun/Bottou/Bagio : CONV layer
  - Large Scale Visual Recognition Challenger
    - 2012: AlexNet : multiple CONV layers
    - 2014: GoogLeNet: Inception modules
      - Doing each convolution in parallel and concatenating the resulting feature
    - 2015: Residual Networks (ResNet)
      - Skip Layer
      - Batch normalization
      - No Fully Connected Layers
    - etc.....
  - L'apprentissage supervisé est un domaine actif déjà largement fouillé avec des résultats excellents



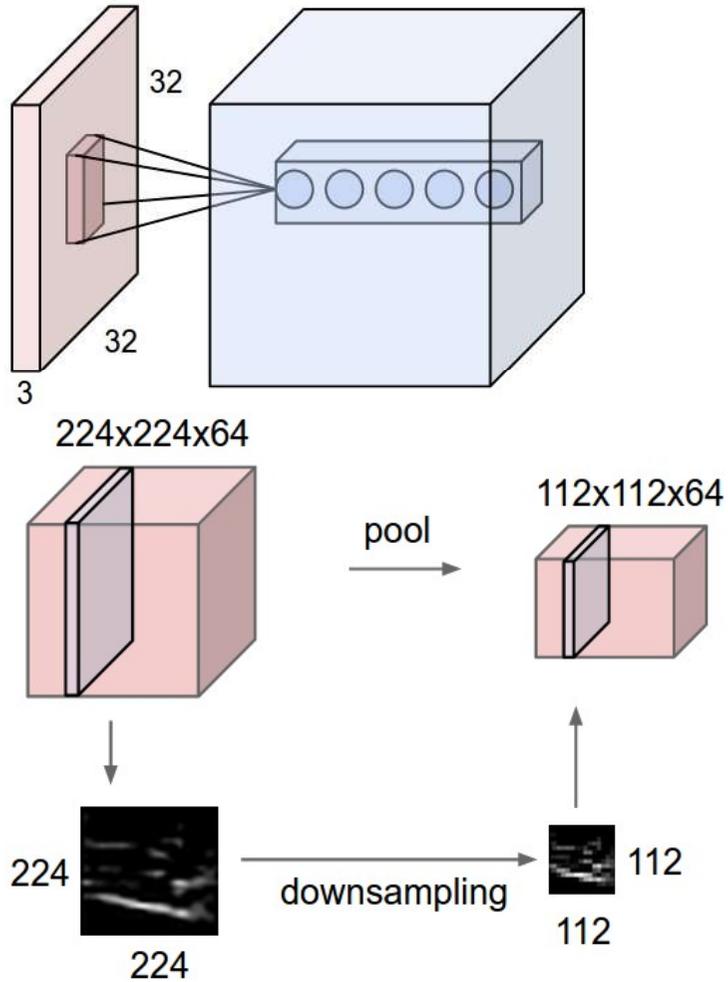
DLVS

## Convolutional Neural Networks (CNN)





Couches



CONVolution  
layer

POOLing layer





## IA: 2010-

- ⌘ Les DNN ont atteint un niveau de maturité exceptionnel dans la reconnaissance d'images.
- ⌘ Les applications sont multiples, dans de très nombreux domaines.
- ⌘ On ne parle cependant pas d'Intelligence Artificielle...



## IA: 2010-

- ⌘ Le balancier va pourtant bientôt repartir (à nouveau) dans l'autre sens...
- ⌘ 2010: fondation de la société DeepMind, championne de l'approche connexionniste et de l'apprentissage
- ⌘ 2013: développement d'un DNN en apprentissage par renforcement pour jouer à Space Invaders et Breakout
- ⌘ 2014: Google achète DeepMind



# Apprentissage par renforcement et DNN

Qu'est-ce que c'est ?

Croisement entre deux univers:

- L'apprentissage par renforcement: on est capable d'évaluer le résultat et de l'utiliser pour améliorer le système mais on n'a pas de base d'exemples pour faire de l'apprentissage supervisé.
- Les DNN



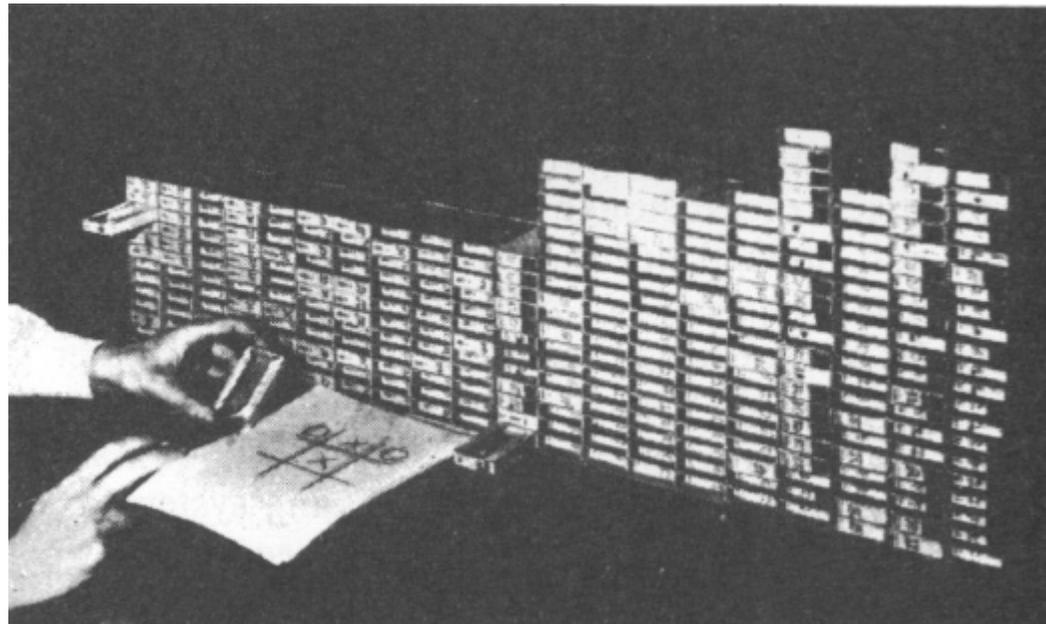
## L'apprentissage par renforcement

### ⌘ *Turing:*

- ☒ « Instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child brain is something like a notebook as one buys it from the stationers. Rather little mechanism, and lots of blank sheets.”
- ☒ “The use of punishments and rewards can at best be a part of the teaching process. Roughly speaking, if the teacher has no other means of communicating to the pupil, the amount of information which can reach him does not exceed the total number of rewards and punishments applied. By the time a child has learnt to repeat “Casabianca” he would probably feel very sore indeed, if the text could only be discovered by a “Twenty Questions” technique, every “NO” taking the form of a blow.”

## L'apprentissage par renforcement

- ⌘ Arthur Samuel (1959): « Some Studies in Machine Learning Using the Game of Checkers »
  - ☑ Memoization
  - ☑ Ajustement automatique de paramètres par régression.
- ⌘ Donald Michie (1962): MENACE (*Matchbox Educable Noughts And Crosses Engine*)





## L'apprentissage par renforcement

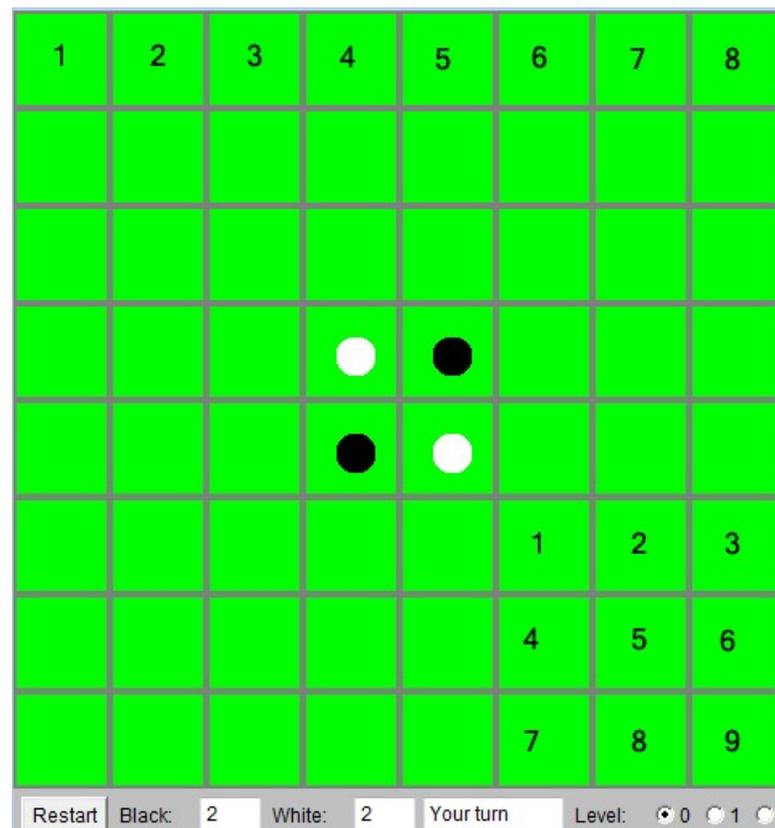
⌘ *MENACE permet de comprendre de façon « visuelle » de nombreux problèmes que l'on rencontre dans l'apprentissage:*

- ☒ *La représentation de la connaissance (front end processing)*
- ☒ *Le nombre élevé de paramètres*
- ☒ *La façon d'initialiser le système*
- ☒ *La façon de récompenser / punir: comment identifier le coup « responsable » en cas de victoire ou de défaite*
- ☒ *La façon de constituer la base d'apprentissage*
- ☒ *La façon de présenter la base d'apprentissage*
- ☒ *Le nombre élevé d'exemples nécessaires*



## L'apprentissage par renforcement

- La méthode, aussi simpliste qu'elle paraisse, fonctionne même dans des cas beaucoup plus complexes
- OTAGE, programme d'Othello que j'ai écrit dans les années 90 fonctionne exactement sur le même principe, a figuré dans les 10 premiers mondiaux





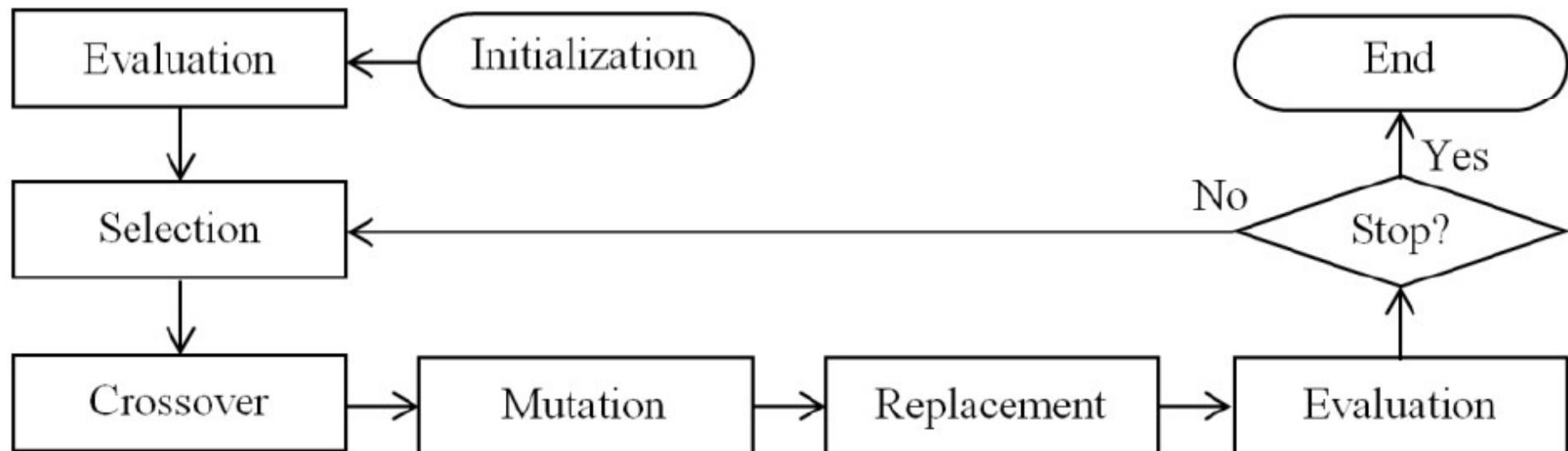
## Réseaux de neurones et renforcement

Comment faire lorsque l'on ne dispose pas de base d'exemples mais que l'on sait évaluer la qualité du résultat?

- Utiliser des techniques de type génétique pour sélectionner et faire évoluer les réseaux de neurones (AG+NN) ou les fonctions d'évaluation (GP)
- Sutton (1988): « Learning to predict by the methods of temporal differences »
  - Algorithme TD-lambda (exemple: TD-gammon)
- Watkins (1989): « Learning from Delayed Rewards »
  - Algorithme de « Q-learning »
  - Algorithme SARSA
  - Deep Q learning
  - Double Q learning
  - etc....

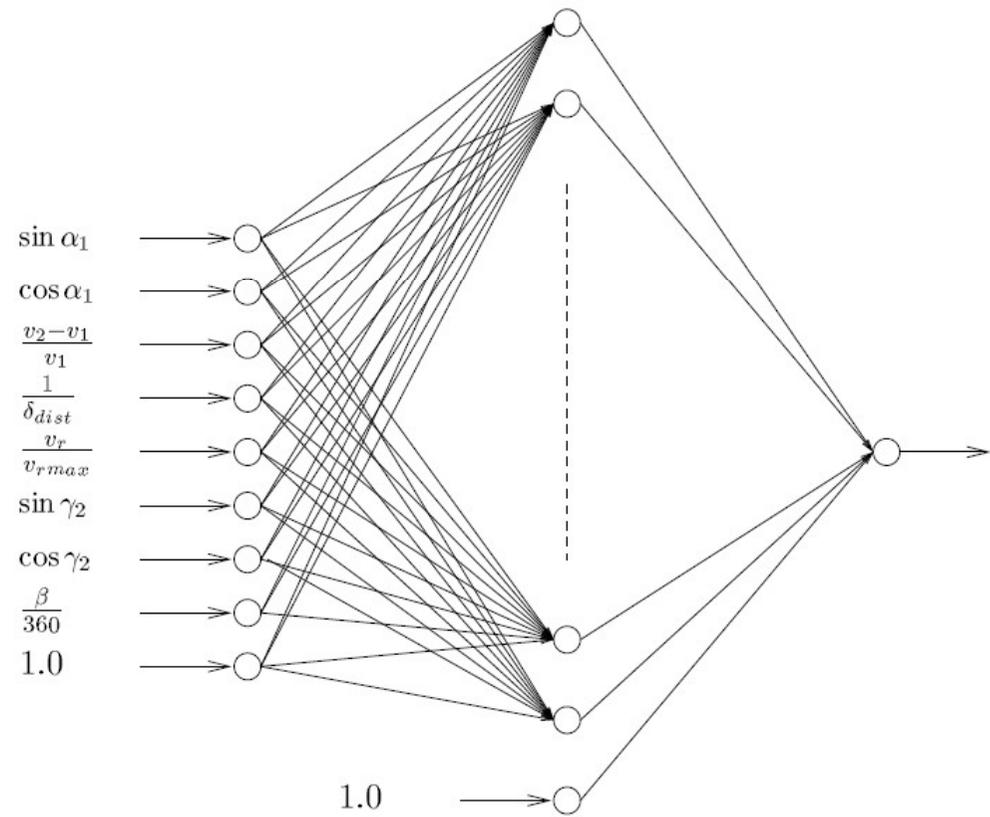
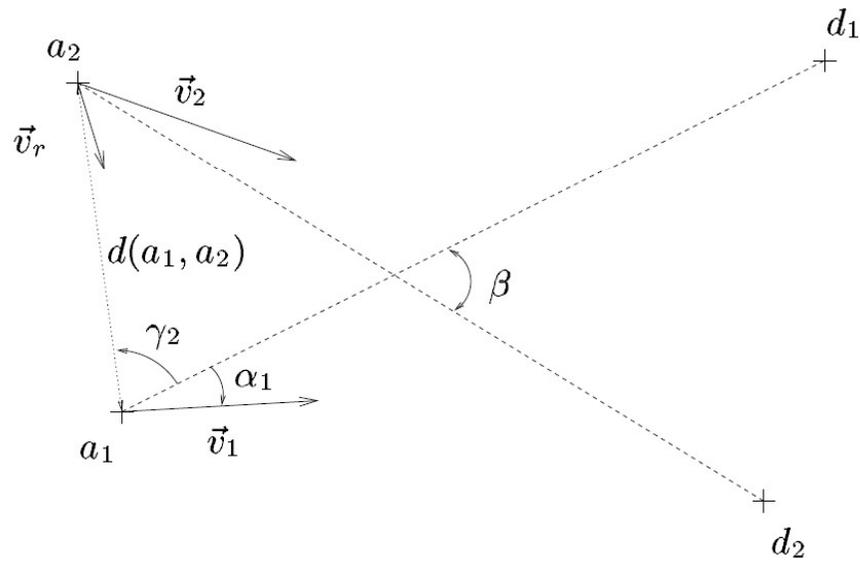


## Algorithmes génétiques





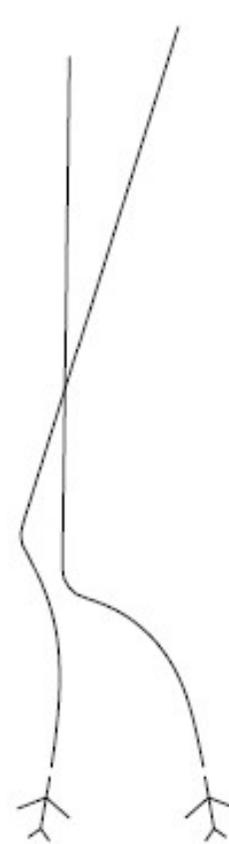
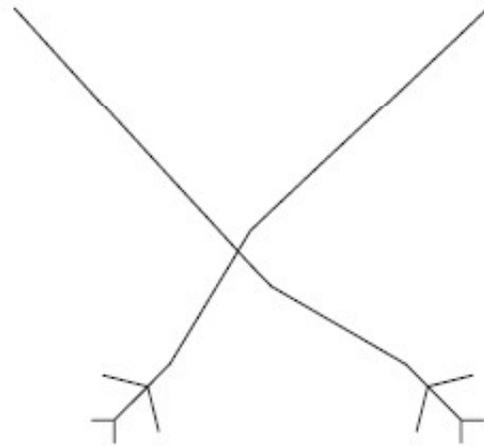
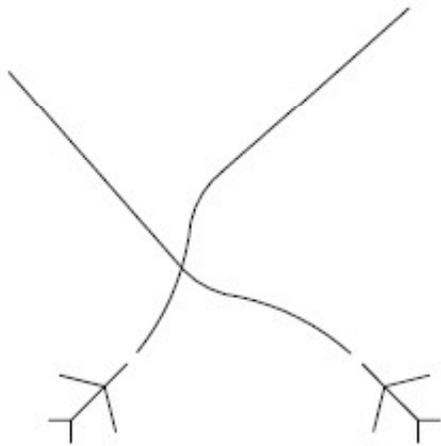
Résolution de conflits (Alliot-Durand-Médioni 1996)





**DLVS**

## Résolution de conflits (RN+AG)





## Learning from delayed rewards

Famille de problèmes qui ont tous en commun la même difficulté:

- Le problème des bandits manchots
- Les algorithmes de type « Monte Carlo Tree Search »
- L'apprentissage d'un RN en fonction du résultat

## Les bandits manchots

- ⌘ On dispose de  $n$  « bandits manchots », quelle est la meilleure stratégie d'exploration pour maximiser le gain? (*exploitation-exploration dilemma*)
  - ⊞ L'exploration permet de mieux identifier les « meilleurs » bandits manchots, mais fait chuter le gain
  - ⊞ Se focaliser rapidement sur un bandit manchot peut amener à manquer le "bon" et donc à faire également chuter le gain.
- ⌘ Upper Confidence Bound (UCB) algorithm:
  - ⊞  $\bar{X}_j$ : moyenne des résultats pour le bras  $j$
  - ⊞  $n_j$ : nombre de fois où le bras  $j$  a été utilisé
  - ⊞  $n$ : nombre total de coups joués
  - ⊞ Prendre le  $j$  qui maximise:

$$\text{UCB1} = \bar{X}_j + \sqrt{\frac{2 \ln n}{n_j}}$$

## Q learning

- On dispose d'un ensemble  $S$  d'états et un ensemble  $A$  d'actions. A chaque instant  $t$ , dans l'état  $s_t$ , on peut exécuter une ou plusieurs actions  $a_t$  qui permettent de passer dans un nouvel état  $s_{t+1}$ . Une action amène un gain immédiat  $r_t$ , et le but est de maximiser la somme des gains jusqu'à l'état final en partant de l'état initial.
- Problème dont on « voit » la proximité avec le bandit manchot, mais aussi avec le problème de l'apprentissage par renforcement.
- On construit une fonction  $Q(s_t, a_t)$  qui va estimer la « valeur » de l'exécution de l'action  $a_t$  dans l'état  $s_t$

$$Q(s_t, a_t) \leftarrow (1 - \alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left( \underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)$$

learned value

- ⌘  $Q(s, a)$  est normalement une fonction discrète stockée sous forme de table. Si le nombre d'états et/ou d'actions deviennent grands, on peut estimer  $Q$  en utilisant un DNN => algorithme Deep Neural Q Learning (DQN, puis DDQN)
- ⌘ 2013: Deepmind développe un programme basé sur un DQN pour jouer parfaitement à certains jeux d'Atari

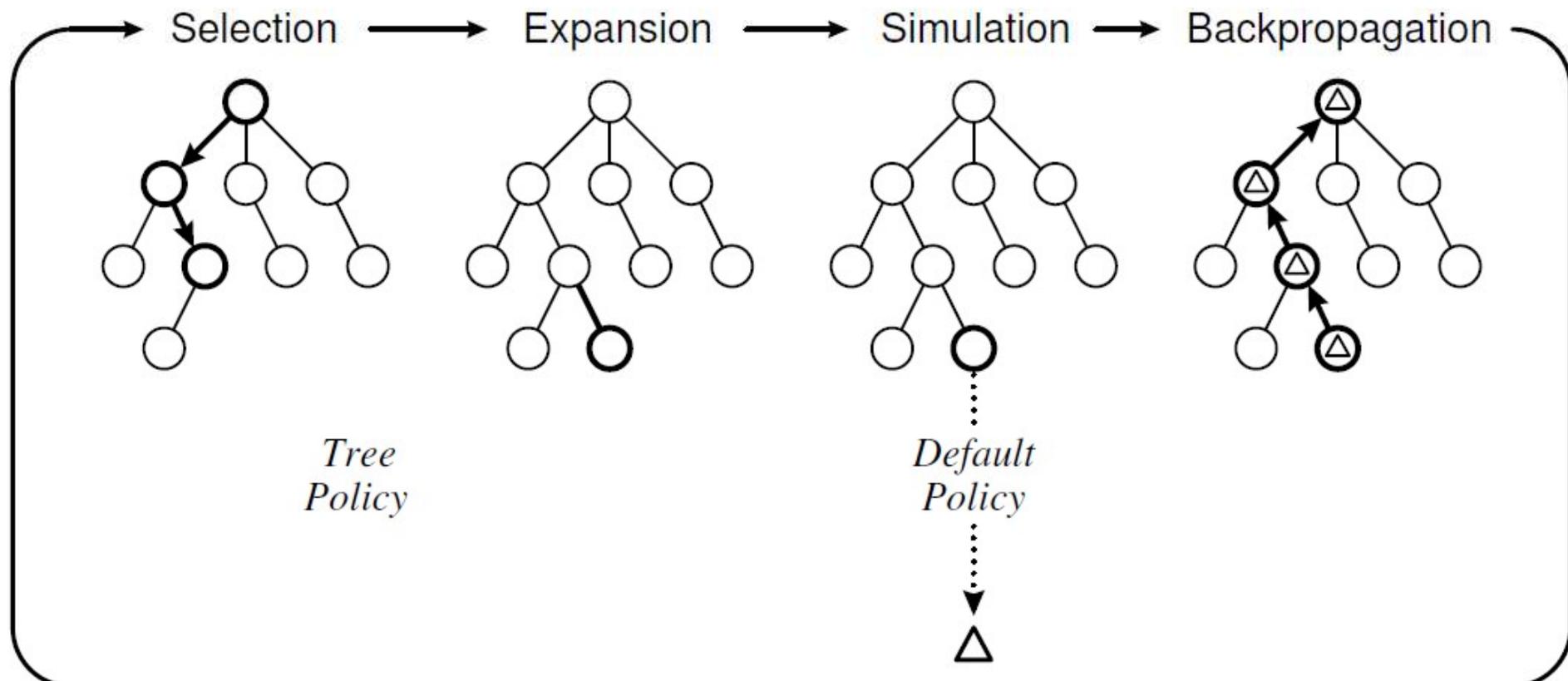


## Minimax

- En théorie des jeux, les algorithmes « standards » utilisés sont de la forme minimax (avec toutes leurs variantes) à profondeur fixe.
- L'idée (très grossièrement) est de basculer sur des algorithmes de parcours d'arbre légèrement différents (MCTS) et de remplacer la fonction d'évaluation par un réseau de neurones.

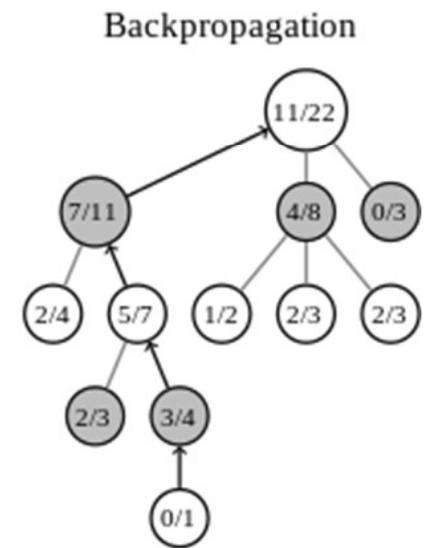
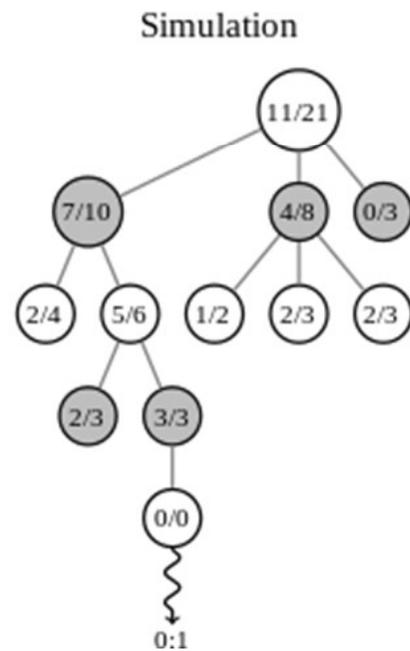
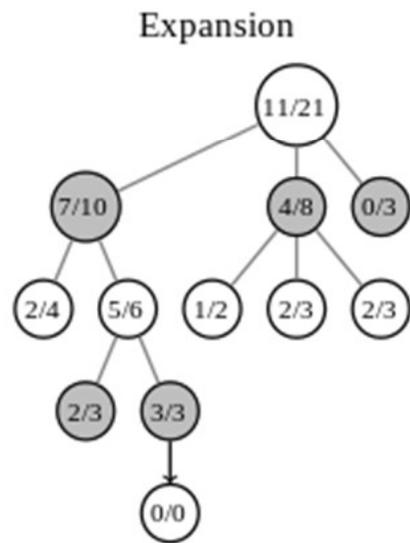
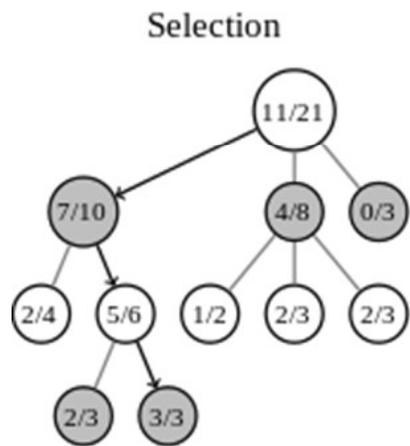
## Monte-Carlo Tree Search (MCTS)

- L'algorithme et le terme MCTS ont été définis par Remi Coulom (un des directeurs de thèse d'Aja Huang et l'auteur de CrazyStone) en 2006 (l'idée remonte à Abramson en 1987). L'algorithme est séparé en 4 phases et étend l'arbre de jeu de façon « inégale ».
- La « tree policy » utilisée est souvent l'algorithme UCB.
- MCTS est généralement moins efficace que l'alpha-béta sauf...



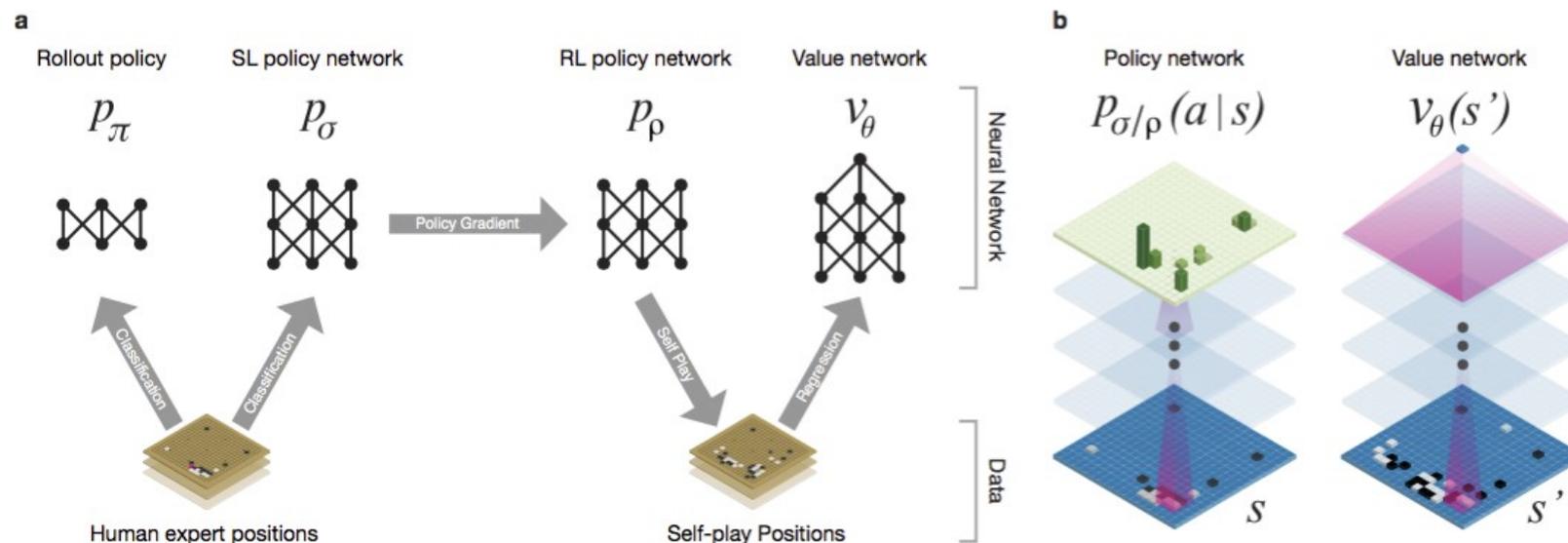


# Monte-Carlo Tree Search (MCTS)



## AlphaGo

- 2 DNN: un « policy network » et un « value network »
- Etape 1: apprentissage supervisé de  $p_\pi$  et  $p_\sigma$  à partir de 30 millions de positions récupérées sur le serveur KGS. La sortie est un vecteur de probabilités évaluant l'ensemble des coups possibles à partir de la position courante.
- Etape 2: Construction de  $p_\rho$  à partir de  $p_\sigma$  en faisant jouer le programme contre lui-même.
- Etape 3: Construction de  $v_\theta$  (en utilisant  $p_\rho$ ) qui retourne la probabilité de victoire pour une position donnée.
- Etape 4: on utilise un algorithme de type MCTS avec une « tree policy » proche du Q learning





## AlphaGo: détail de l'algorithme MCTS

- Une action pour un nœud est choisie suivant l'équation (proche UCB):

$$a_t = \operatorname{argmax}_a \left( Q(s_t, a) + u(s_t, a) \right) \quad u(s, a) \propto \frac{P(s, a)}{1 + N(s, a)}$$

- Tous les  $P(s, a)$  sont initialisés par le vecteur fourni par le réseau  $p_\sigma$  (et non  $p_\rho$ ) lors de la première visite du nœud ( $p_\rho$  fonctionne moins bien !).
- $N(s, a)$  est le nombre de fois où l'action  $a$  a été exécutée pour le nœud  $s$
- Lorsque l'on atteint un nœud terminal  $s_L$  sa valeur est évaluée en utilisant la combinaison linéaire du réseau  $v_\theta$  et d'une simulation terminale utilisant la politique de choix rapide  $p_\pi$

$$V(s_L) = (1 - \lambda)v_\theta(s_L) + \lambda z_L$$

- Comme dans un MCTS standard, les valeurs  $Q(s, a)$  impliquant les nœuds  $s$  traversés sont mises à jour suivant l'équation:

$$N(s, a) = \sum_{i=1}^n \mathbf{1}(s, a, i) \quad Q(s, a) = \frac{1}{N(s, a)} \sum_{i=1}^n \mathbf{1}(s, a, i) V(s_L^i)$$



- ⌘ 03/2013: CrazyStone (Remi Coulom) bat Yoshio Ishida (9d) avec 4 de handicap
- ⌘ 09/2015: AlphaGo (DeepMind) bat FanHui
- ⌘ 03/2016: AlphaGo bat Lee Sedol (2<sup>ème</sup> mondial) 4-1
- ⌘ Durant la fin de l'année 2016, AlphaGo « updated » joue en ligne contre les meilleurs joueurs mondiaux et gagne 60-0
- ⌘ "After humanity spent thousands of years improving our tactics, computers tell us that humans are completely wrong... I would go as far as to say not a single human has touched the edge of the truth of Go."  
(Ke Jie)



## AlphaZero

- AlphaGo contre LeeSedol (9-15 mars 2016): 4-1
- AlphaGoMaster (4 TPUv2):
  - 60-0 dans des matchs rapides en ligne contre les meilleurs humains
  - 5-0 dans « The future of Go Summit » dont 3-0 contre KeJie
- AlphaGoZero est une version d'AlphaGo qui n'a construit sa connaissance qu'à partir de parties jouées contre lui-même (pas d'apprentissage supervisé à partir des positions de KGS). Il est légèrement plus fort qu'AlphaGoMaster.
- AlphaZero est un algorithme générique utilisé à la fois pour le Go, les échecs et pour le shogi qui reprend l'architecture générale d'AlphaGoZero avec quelques modifications. Il utilise 64 TPUv2 pour l'apprentissage et 4 TPUv2 pour jouer.
  - AlphaZero est légèrement plus fort qu'AlphaGoZero: 60-40
  - AlphaZero bat Stockfish (64 threads, 1Gb de hash): 25-72-3



## AlphaZero

- Les résultats contre Stockfish sont « discutables »
- AlphaZero est indiscutablement plus fort, mais cela ne signifie pas la fin des algorithmes minimax car:
  - Stockfish est privé de ses bibliothèques d'ouverture et de finale
  - Stockfish n'utilise que 1 Gb de table de Hash
  - Stockfish utilise 64 CPU:
    - On est très loin de la puissance des 4 TPUv2 de Google
    - Stockfish est un programme développé par des « amateurs » qui ne disposent pas de machines massivement parallèles. Le programme n'est pas optimisé pour le parallélisme massif (la recherche sur la parallélisation des algorithmes de type alpha-béta est peu active).



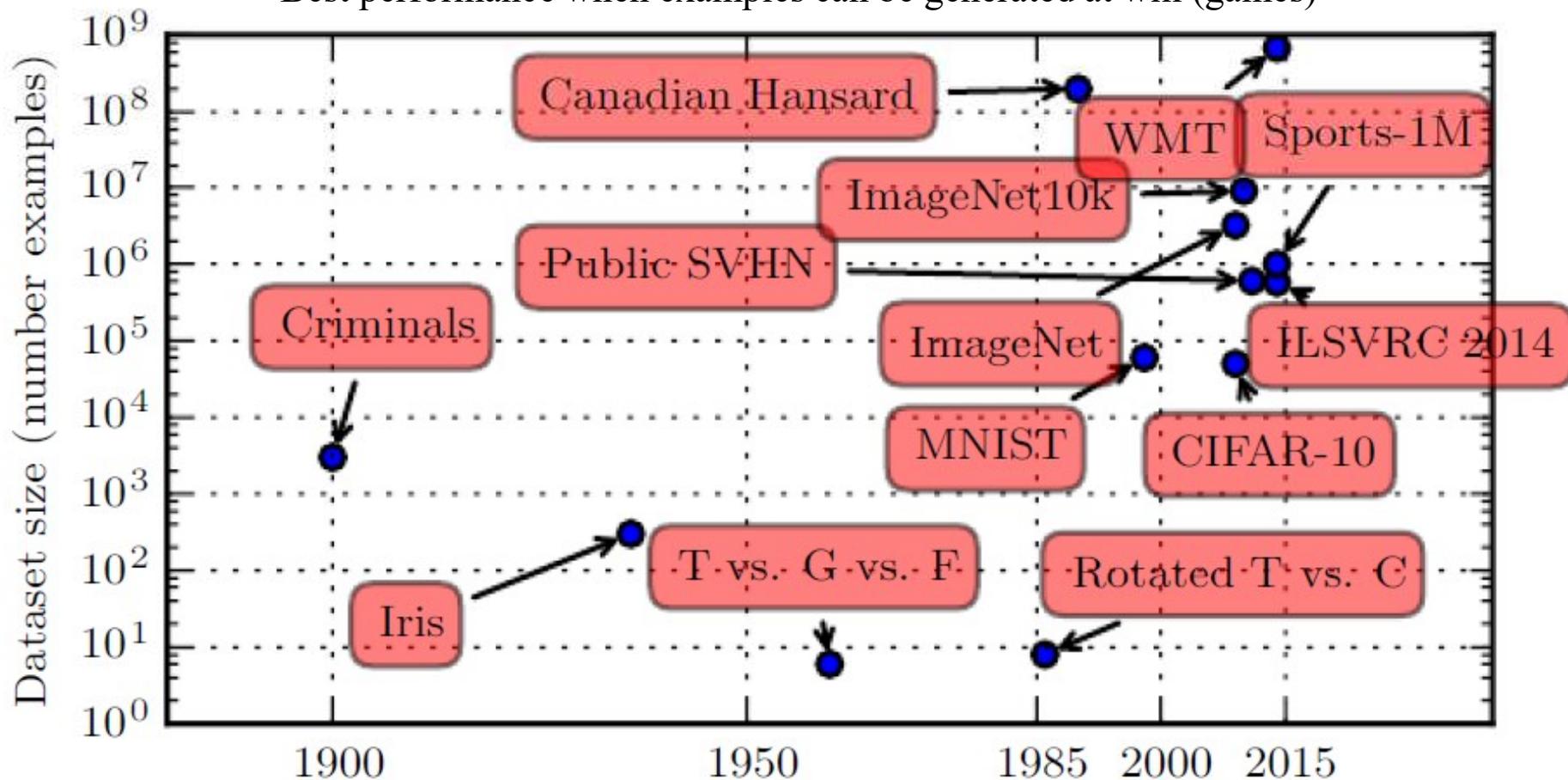
## Facteur 1: GPU et TPU

- L'explosion des DNN est largement liée à l'arrivée de processeurs SIMD (Single Instruction Multiple Data) massifs
- La progression de vitesse des CPU a considérablement ralenti depuis plusieurs années, alors que la puissance des GPU continue d'augmenter régulièrement :
  - Riva TNT (1998) : 180 Mop/s
  - GTX 1080 (2016) : 8 Tflop/s
- Google a développé pour ses DNN un processeur particulier, le TPU (Tensor Processing Unit) qui est spécialisé dans la multiplication matricielle. La version 2 développe 180 Tflop/s et 64Gb de mémoire pour 200Watts de consommation (Google loue ses TPU en bêta-test pour 6.5 dollars de l'heure à quelques chercheurs).
- Nvidia Tesla V100 (2017): TPU speed: 120 Tflop/s, 16Gb, 300W, 10000\$
- Nvidia RTX 2080Ti (2018): TPU speed: 108 Tflop/s, 11Gb, 250W, 1000\$

## Facteur 2: le temps du « « Big Data » »

Bagio (2017): for supervised learning :

- Acceptable performance: 5000 examples
- Expert performance: 10 millions examples
- Best performance when examples can be generated at will (games)



- ⌘ Les approches basées sur l'apprentissage et le data mining (fouille de données) se généralisent. C'est la revanche absolue de l'approche « boîte noire ».
- ⌘ Succès dans bien d'autres domaines que le Go...
  - ☒ Traduction (nouvelle version en 2016 du système google de traduction)
  - ☒ Evolution des moteurs de recherche
  - ☒ Médecine: ...



- ⌘ Juillet 2016: partenariat avec *Moorfields eyes hospital* pour l'analyse de scans rétiniens afin de détecter les signes précurseurs de perte de la vue.
- ⌘ Aout 2016: partenariat avec *l'University College London Hospital* pour la détection de tissus cancéreux dans la nuque et la tête.
- ⌘ Novembre 2017: partenariat avec le *cancer research UK* à Imperial College pour l'analyse de mammographie afin de détecter le cancer du sein
- ⌘ Et bien d'autres....



**IA 2015:**

**la fin de l'humanité arrive...**

- ⌘ Stephen Hawking: "Artificial intelligence could spell the end of the human race."
- ⌘ Elon Musk: "I think we should be very careful about artificial intelligence. If I were to guess like what our biggest existential threat is, it's probably that."
- ⌘ Bill Gates: « First the machines will do a lot of jobs for us and not be super intelligent. That should be positive if we manage it well. A few decades after that though the intelligence is strong enough to be a concern."

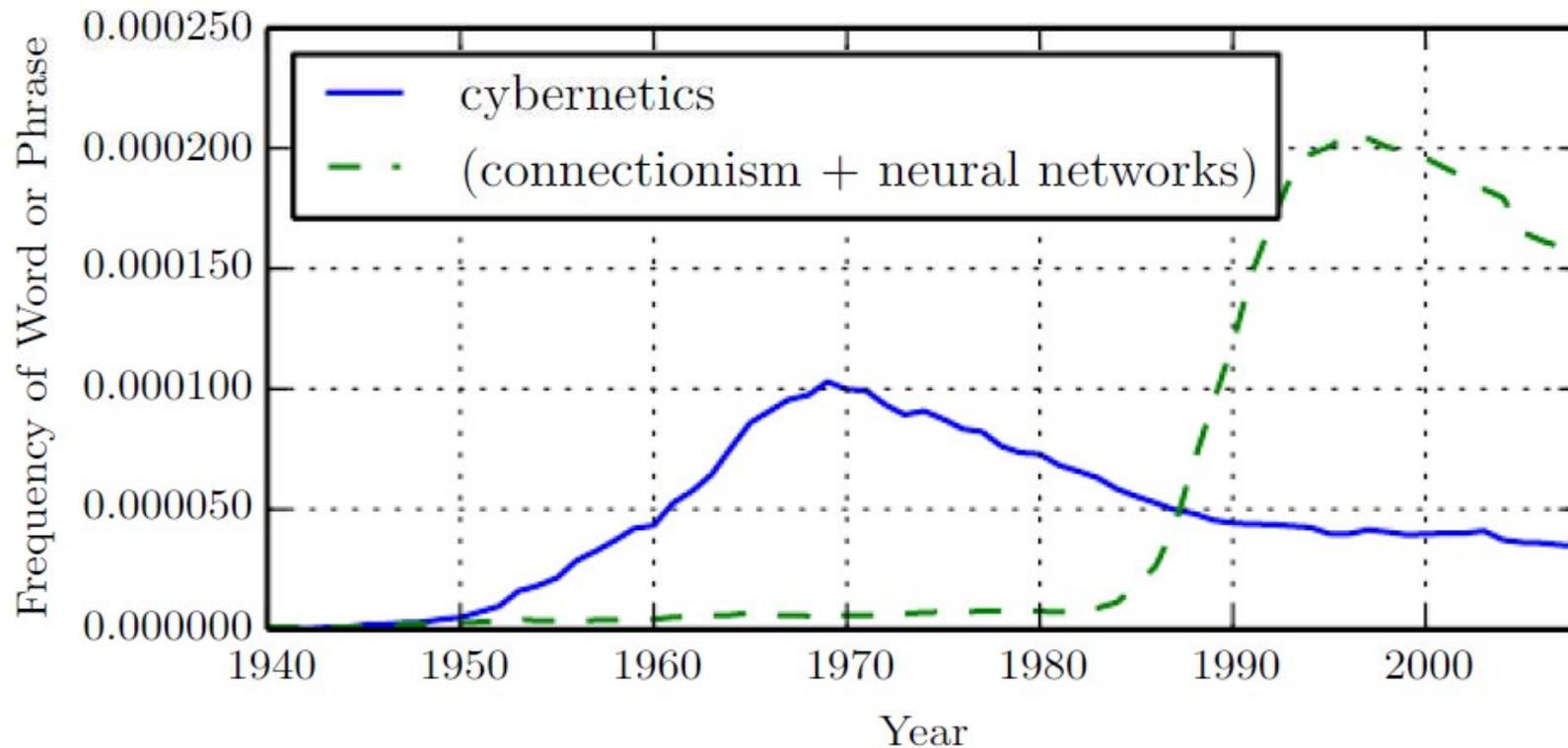


## The hype

- ⌘ H. L. Mencken: « The whole aim of practical politics is to keep the populace alarmed (and hence clamorous to be led to safety) by menacing it with an endless series of hobgoblins, all of them imaginary.”
- ⌘ Traduction IA très personnelle: “La base de toute activité nécessitant des fonds publics est de faire peur (ou parfois envie) pour obtenir des financements.”

**N'oublions pas!**

- Le Deep Learning et les Deep Neural Networks sont les anciens réseaux de neurones et les anciennes méthodes connexionnistes
- Attention aux effets de mode: l'IA en a suffisamment souffert!!



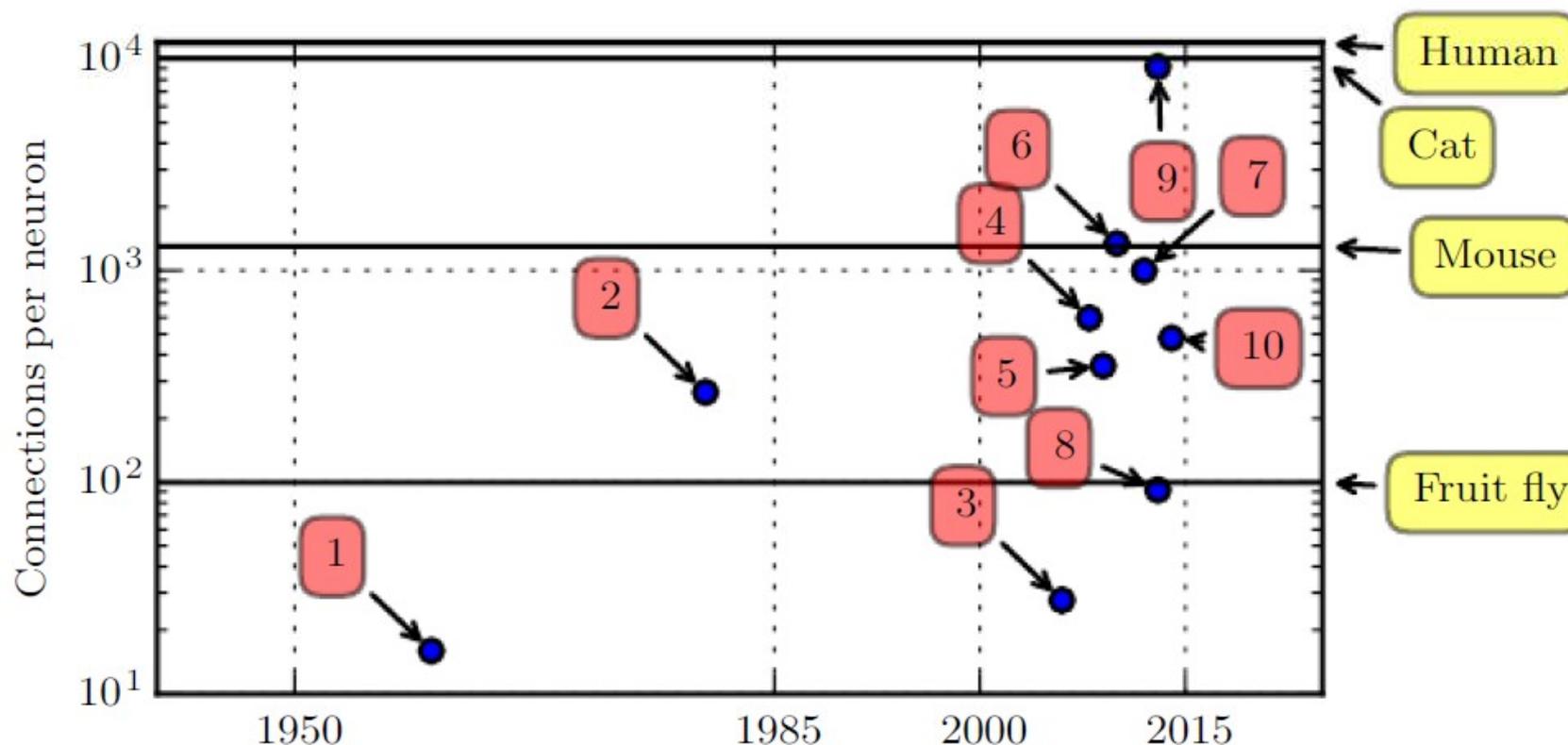


## Some wisdom...

- ⌘ *Rodney Brooks (1991): Critical paper about the then-mainstream line in AI (cognitive AI and expert systems):*
  - ⌘ "There is a bandwagon effect in Artificial Intelligence Research, and many lines of research have become goals of pursuit in their own right, with little recall of the reasons for pursuing those lines"
- ⌘ *Rodney Brooks: Machine Learning Explained (28 Aout 2017):*
  - ⌘ <https://rodneybrooks.com/forai-machine-learning-explained/>
  - ⌘ "Vast numbers of new recruits to AI/ML have jumped aboard after recent successes of Machine Learning, and are running with particular versions of it as fast as they can. They have neither any understanding of how their tiny little narrow technical field fits into a bigger picture of intelligent systems, nor do they care. They think that the current little hype niche is all that matters, are blind to its limitations, and are uninterested in deeper questions."
  - ⌘ "The papers in conferences fall into two categories. One is mathematical results showing that yet another slight variation of a technique is optimal under some carefully constrained definition of optimality. A second type of paper takes a well known learning algorithm, and some new problem area, designs the mapping from the problem to a data representation, and show the results of how well that problem area can be learned."

## Wisdom?

*Le problème à résoudre est un problème de «hardware»: le cerveau humain, avec son incroyable complexité physiologique, ne peut pas se simuler sur les architectures limitées des calculateurs actuels.*





## Quand les DNN font n'importe quoi...

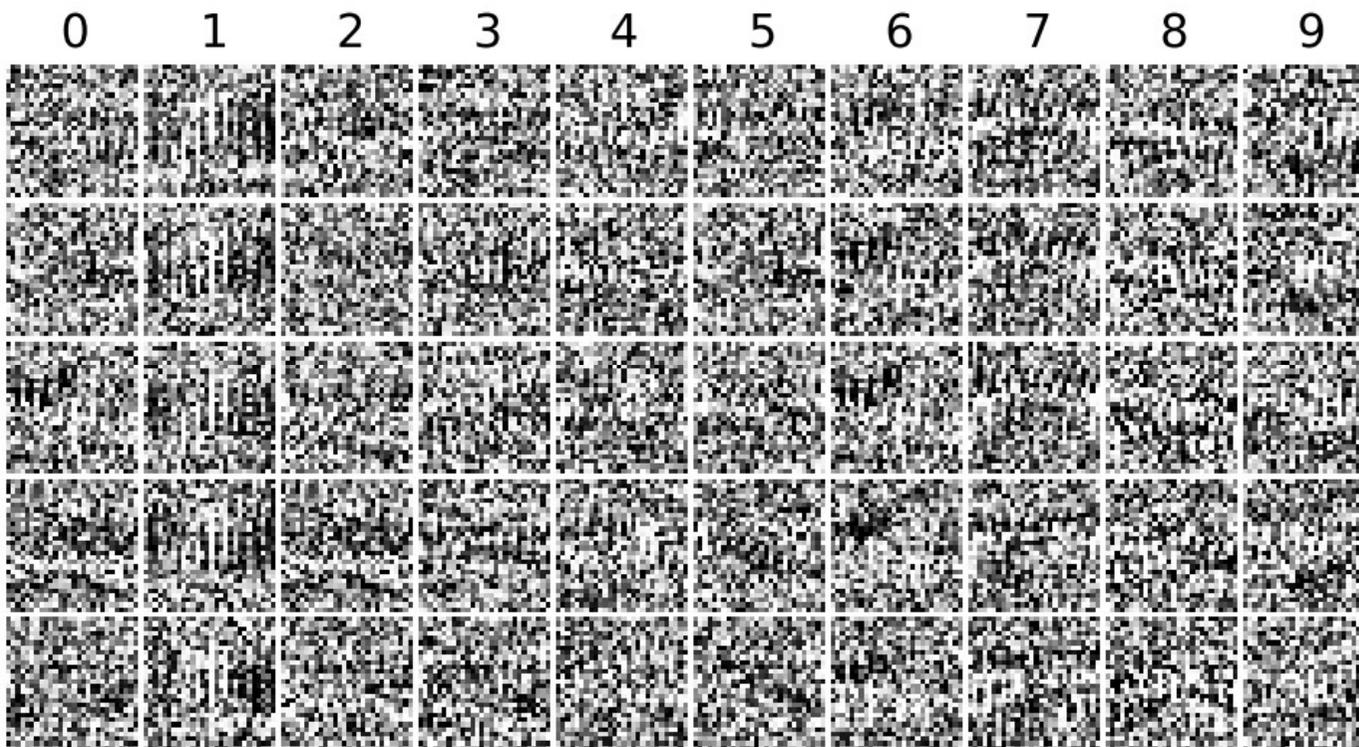


Figure 4. Directly encoded, thus irregular, images that MNIST DNNs believe with 99.99% confidence are digits 0-9. Each column is a digit class, and each row is the result after 200 generations of a randomly selected, independent run of evolution.



## Quand les DNN font n'importe quoi...

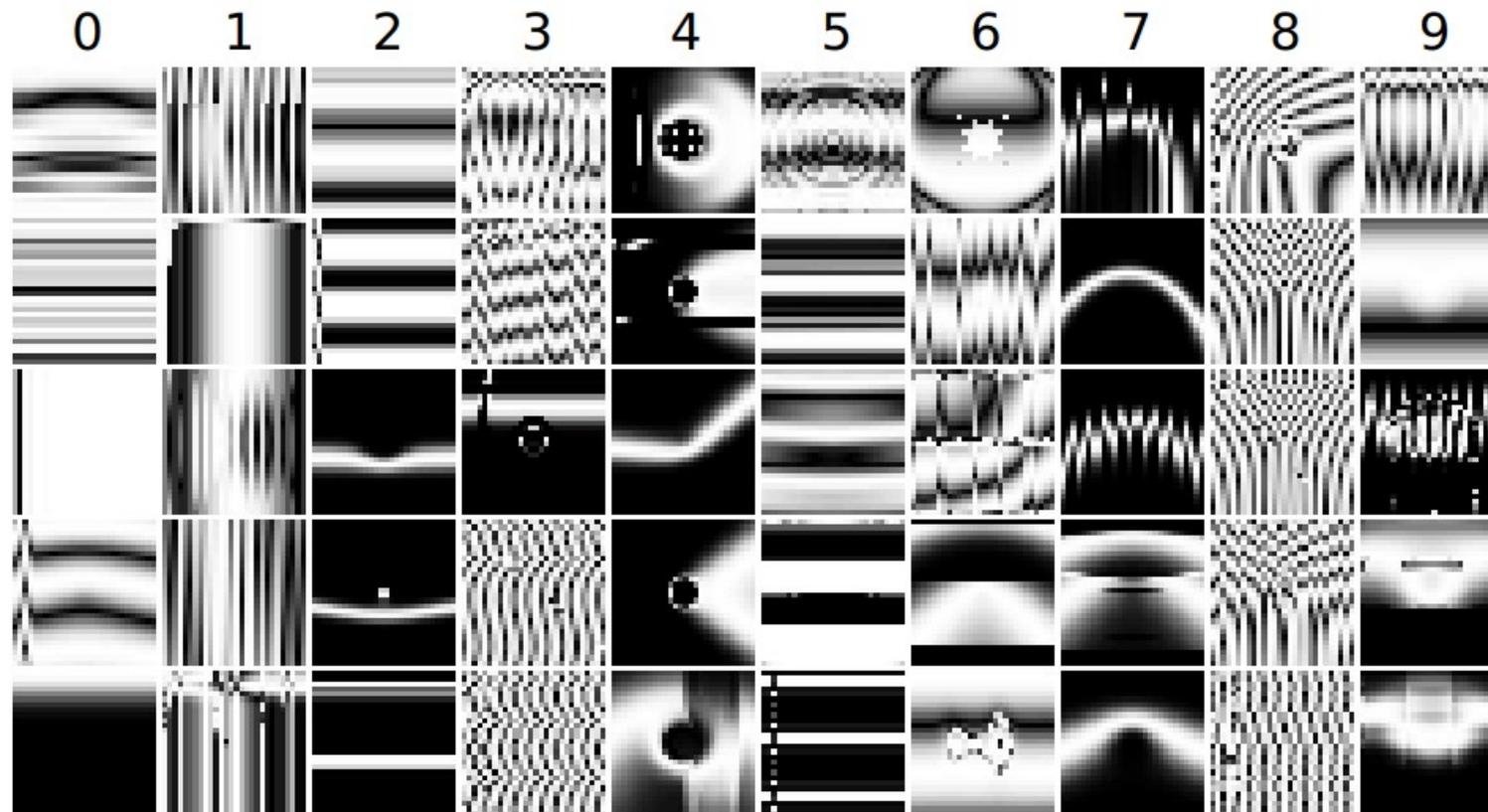


Figure 5. Indirectly encoded, thus regular, images that MNIST DNNs believe with 99.99% confidence are digits 0-9. The column and row descriptions are the same as for Fig. 4.



# Quand les DNN font n'importe quoi...

	CaffeNet	VGG-F	GoogLeNet
Original	 <p>“rapeseed” 99.9% confidence</p>	 <p>“jay” 99.9% confidence</p>	 <p>“bell pepper” 99.8% confidence</p>
Perturbed	 <p>“cardigan” 89.7% confidence</p>	 <p>“mask” 81.8% confidence</p>	 <p>“strainer” 86.5% confidence</p>



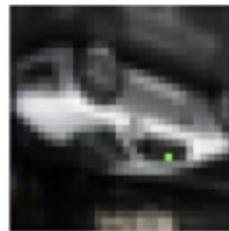
# Quand les DNN font n'importe quoi...



Airplane (Dog)



Automobile (Dog)



Automobile  
(Airplane)



Cat (Dog)



Dog (Ship)



Deer (Dog)



Frog (Dog)



Frog (Truck)



Dog (Cat)



Bird (Airplane)



Horse (Cat)



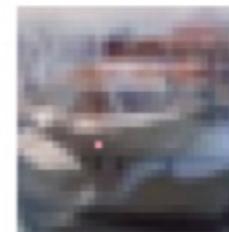
Ship (Truck)



Horse



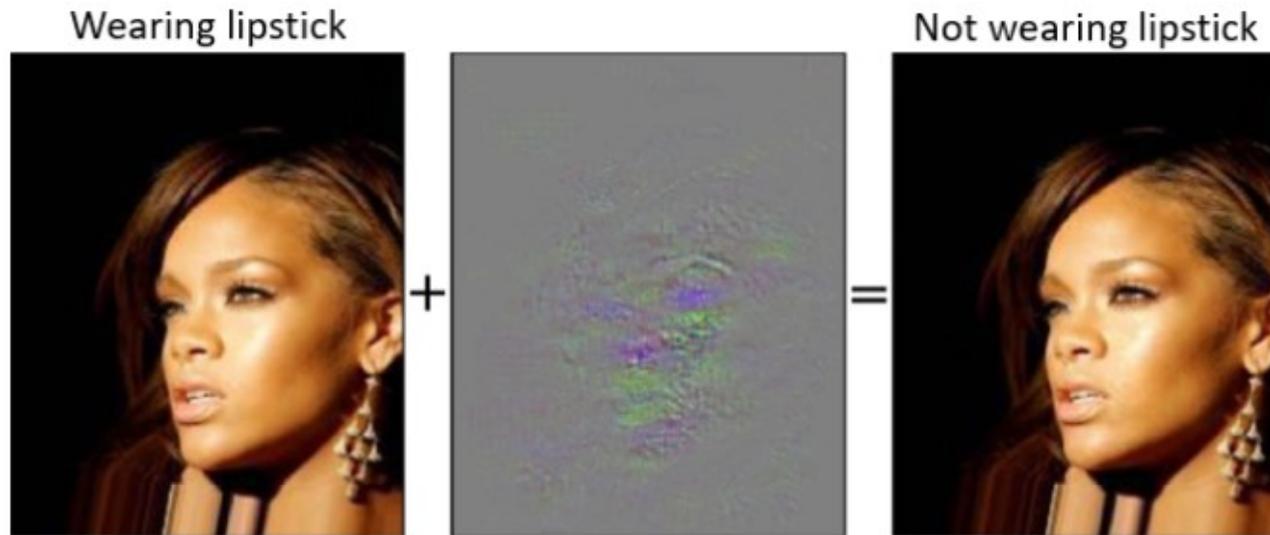
Dog (Horse)



Ship (Truck)



# Quand les DNN font n'importe quoi...



Clean

Perturbation

Adversarial



Female

Male



**2015->? : vive le « Deep Learning » (?!)**

- ⌘ La mode connexionniste s'empare du monde de l'IA, puis du monde de l'informatique, puis du monde tout court...
- ⌘ Attention au 3<sup>ème</sup> « AI Winter »?

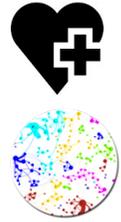


## Mon opinion personnelle (si elle intéresse quelqu'un...)

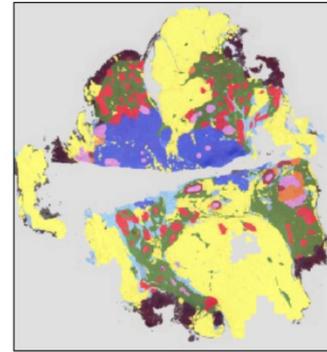
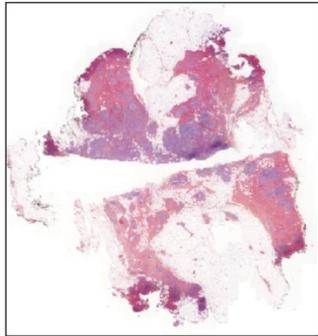
- ⌘ Ne retombons pas dans les erreurs du passé et arrêtons de dire n'importe quoi.
- ⌘ Jouer au Go, ce n'est pas être maître du monde. AlphaGo est un programme très simple et très limité.
- ⌘ Les DNNs restent extrêmement fragiles même sur des tâches aussi élémentaires que la reconnaissance d'image.
- ⌘ L'intelligence artificielle est (presque) aussi loin aujourd'hui qu'il y a 30 ans.



# Deep learning pour la classification d'images complètes



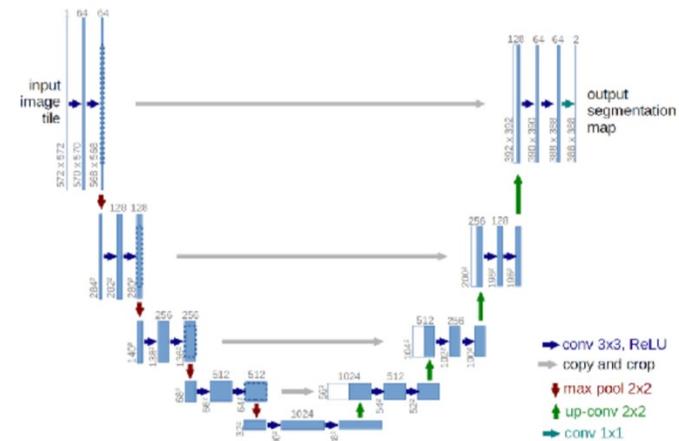
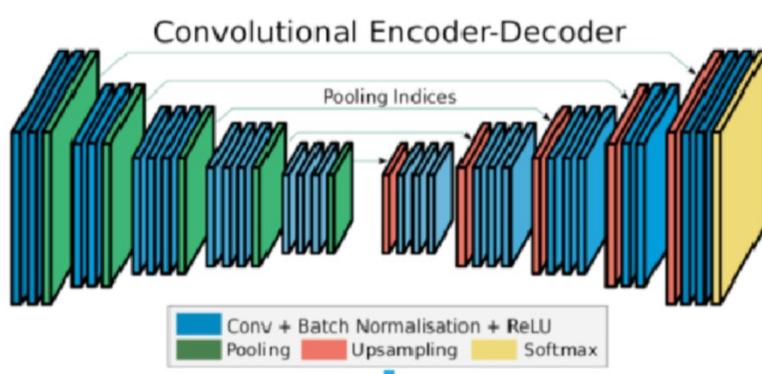
Collaboration IUCT- Pierre Brousset

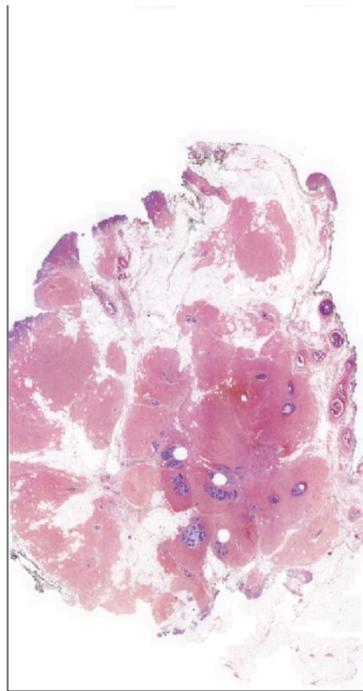


- ⌘ Annotation manuelle du set d'entraînement (6h/image HR)
- ⌘ 2 architectures d'apprentissage

- **Segnet** A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation [Badrinarayanan, Kendall, Cipolla, IEEE 2016]

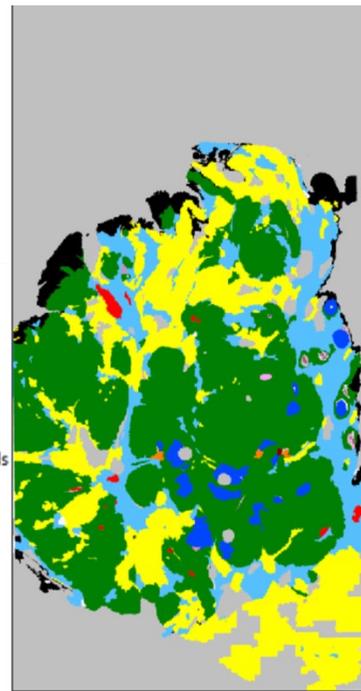
- **UNET** Convolutional Networks for Biomedical Image Segmentation [Olaf Ronneberger, Philipp Fischer, 2015]



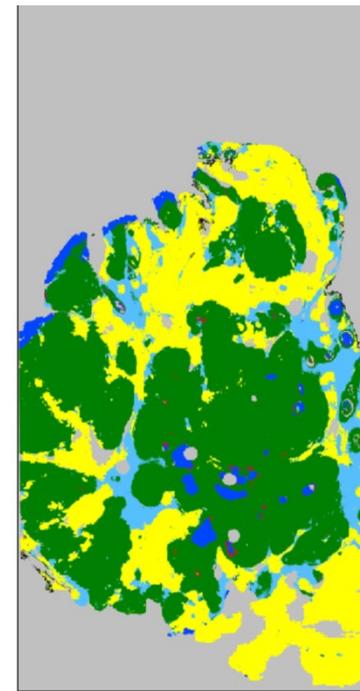


Test image

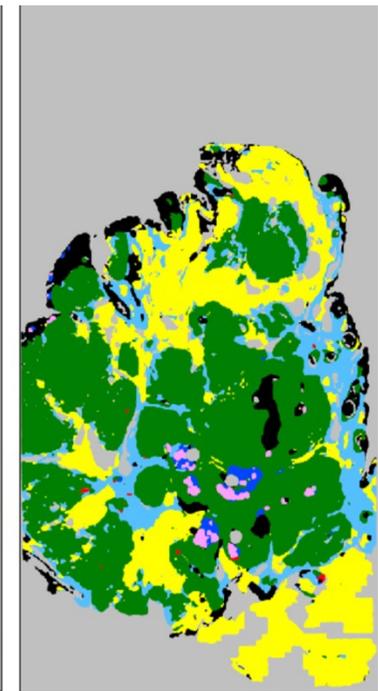
- classes
- Adipose
- Artifacts
- Background
- Carcinoma in situ
- Fibrous
- Fibrous & adipose
- Non carcinomatous mammary glands
- invasive carcinoma & stroma
- skin



Ground truth mask



Predicted mask UNET



predicted mask SEGNET

- NB patches = 5000/image
- Size patch = 288\*288
- 9 train / 2 test
- Data augmentaiton = rotation

<b>Jaccard coef</b>	0,881	0,873
---------------------	-------	-------

$$\text{Jaccard Coefficient} = \frac{\text{True positives}}{\text{True positives} + \text{False positives} + \text{False negatives}}$$

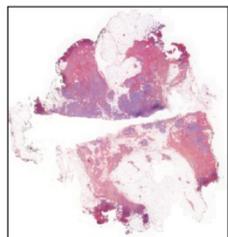
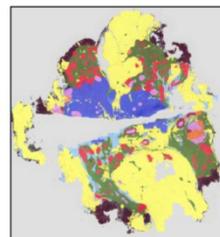


Image Analysis



Machine Learning

Segmentation

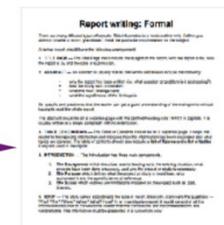


Segmented image



Machine Learning

Text/Image deep mining

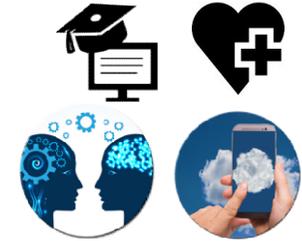


Textual report

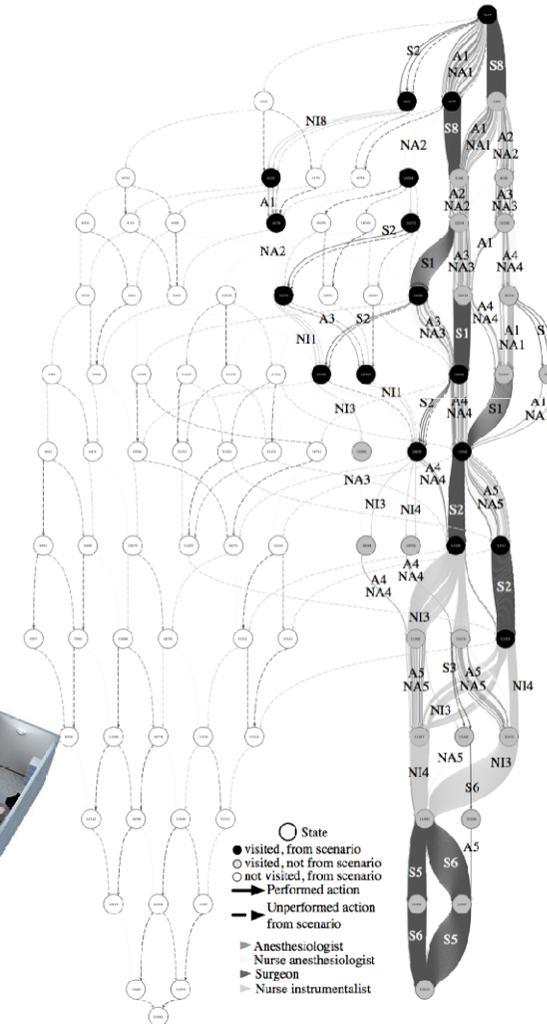


# 3DVOR: Virtual Operating Room

PIA 1 – Pr. Lubrano et Minville (CHU Toulouse)



- ⌘ Serious game dans le domaine médical
  - ☒ Entrainement pour la coordination des équipes
  - ☒ Simulation de salles d'opérations et de salles de réveil
- ⌘ Apprentissage des comportements à l'aide de MCTS
  - ☒ Exploration des chemins comportementaux possibles
  - ☒ Objectif de pouvoir remplacer dynamiquement n'importe quel corps de métier

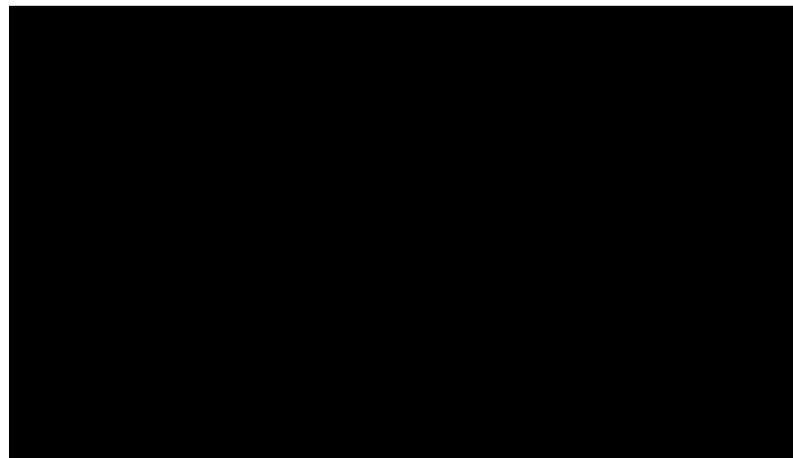
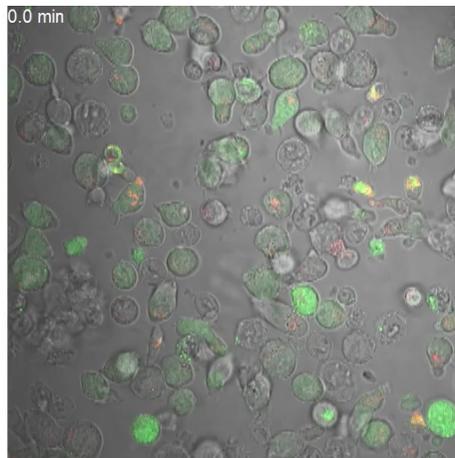




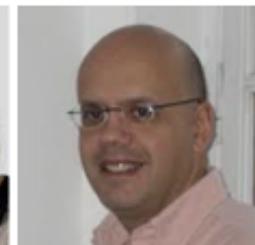
# Simulation pour l'immunologie



- ⌘ Modélisation des interactions entre les lymphocytes T (CTL) et les cellules cancéreuses (mélanomes)
- ⌘ Evaluation in-silico des causes de l'immuno-résistance
- ⌘ Prédiction et amélioration des protocoles de traitement



R. Kazhen



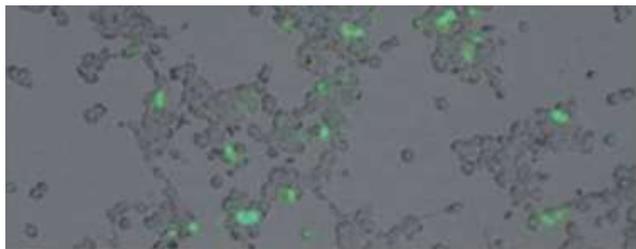
S. Valitutti



# Simulation pour la virothérapie



- ⌘ Modélisation de l'infection contrôlée de tumeurs pancréatiques
- ⌘ Synthèse de la production virale dans les cellules tumorales
- ⌘ Fort impact de la phase du cycle cellulaire sur le traitement



pmaxGFP

**B** 12  
8 10



P. Garcin



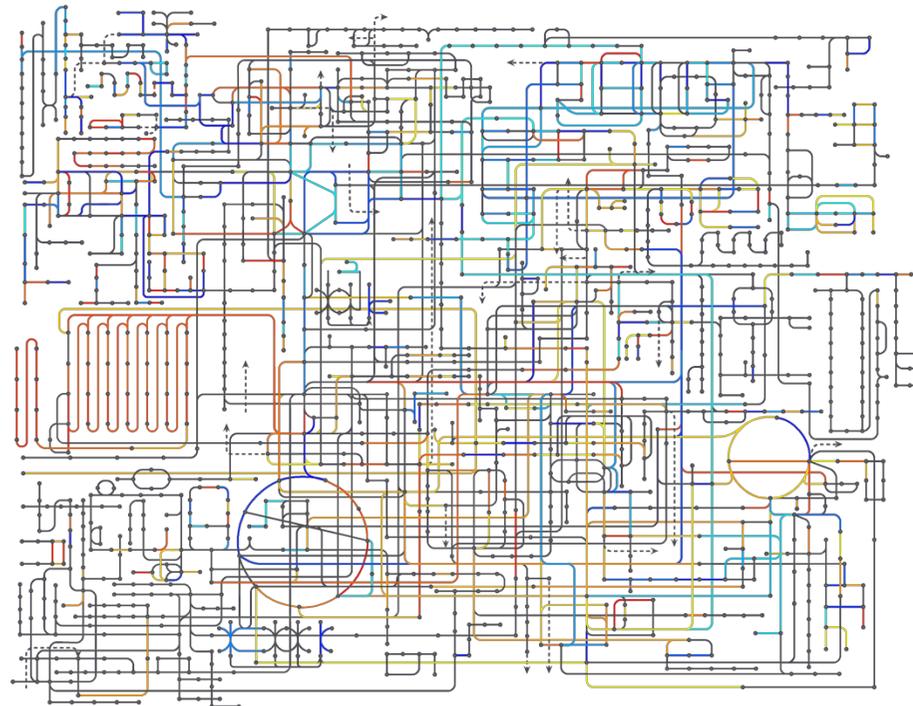
P. Cordelier

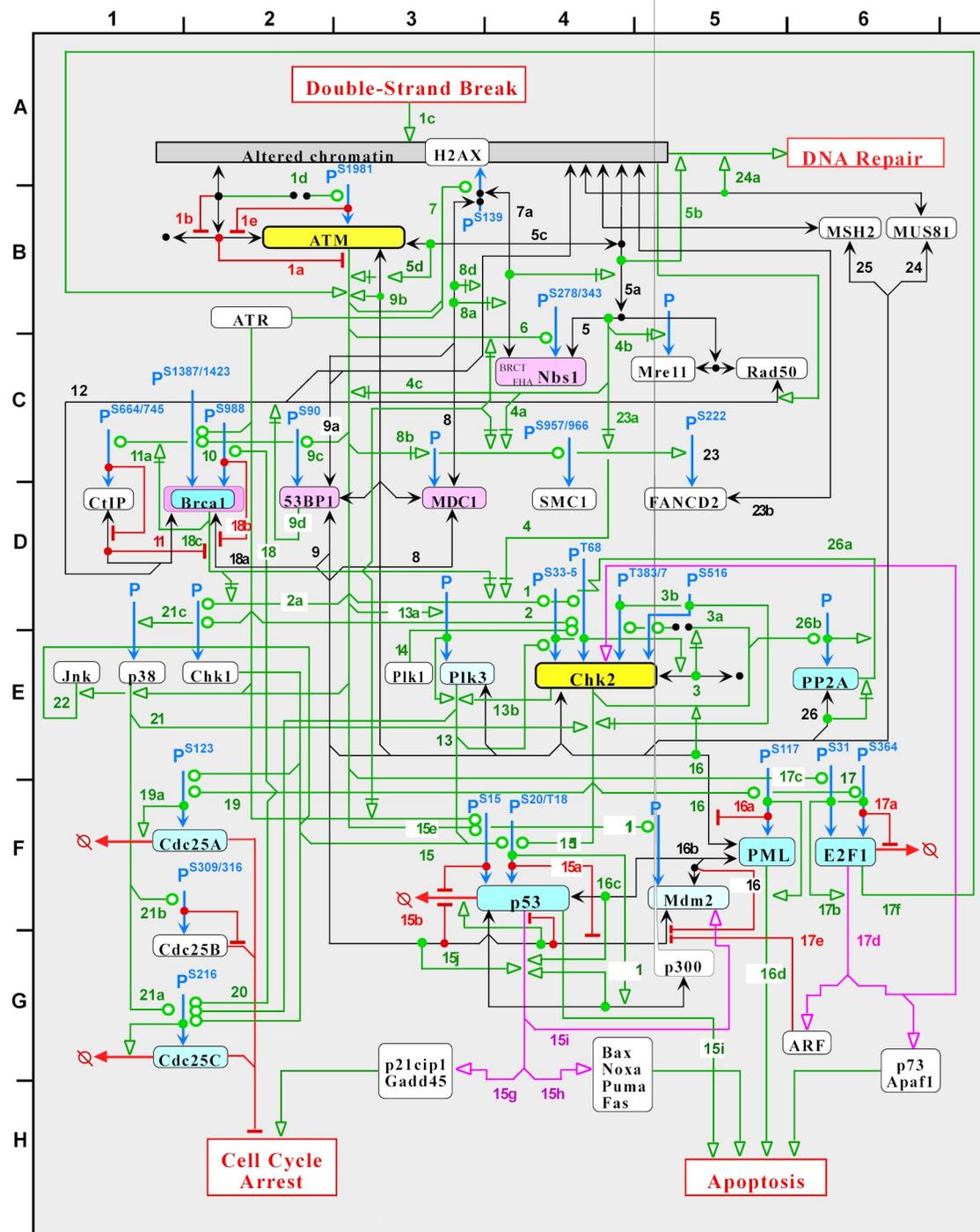


# Exploration de réseaux métaboliques



- ⌘ Recherche de pathways permettant de contourner une mutation délétère
- ⌘ Prédiction de mutation génétique à réaliser pour déclencher la croissance de bactéries







# Modélisation de réseaux métaboliques

Gilles Favre, Jean-Charles Faye, Olivier Sordet (CRCT) / JM Alliot, Luis Farinas (IRIT)

