Projet Foot
Cours 5
2I013

Nicolas Baskiotis
nicolas.baskiotis@lip6.fr

Université Pierre et Marie Curie (UPMC)
Laboratoire d'Informatique de Paris 6 (LIP6)

S2 (2015-2016)

Plan

Tournoi 1v1

```
psg (luluperet) : 54 (18,0,0) - (133,9)
lob (nihaakey) : 45 (15,0,3) - (133,28)
DZPOWER (Raouf16) : 38 (12,2,4) - (85,27)
team2 (Kabegami) : 37 (12,1,5) - (70,30)
JSK (lounisAmazigh) : 30 (10,0,8) - (66,65)
lalya1 (hmdd) : 28 (9,1,8) - (57,57)
team1 (ad50144124) : 18 (6,0,12) - (42,50)
Warrior (Asparodia) : 8 (2,2,14) - (22,85)
equipe11 (jordanupmc) : 8 (2,2,14) - (35,105)
team1 (3408247) : 0 (0,0,18) - (2,189)
```

Tournoi 2v2

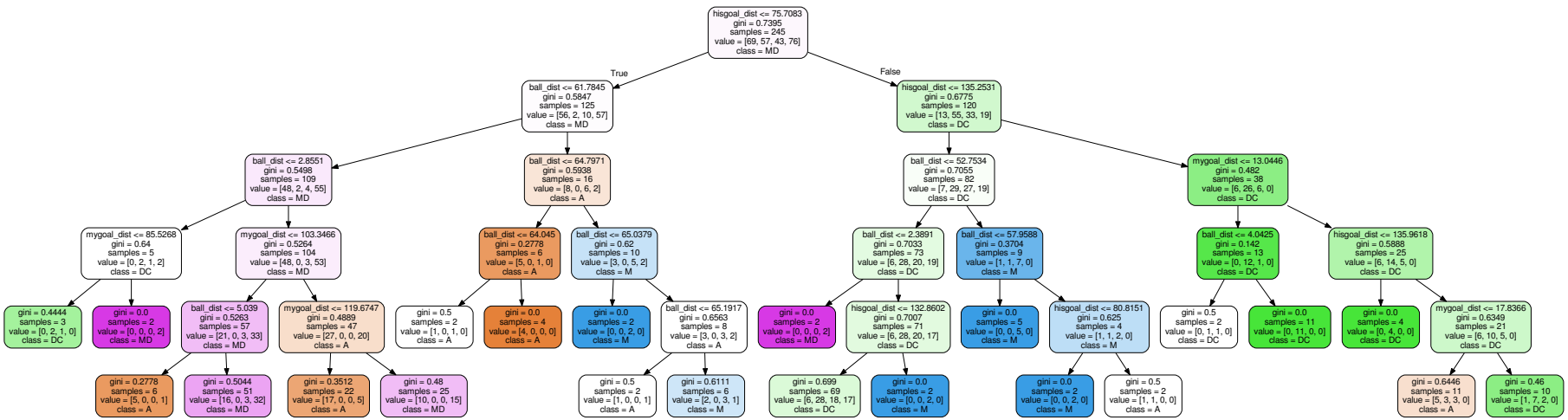
```
***  Resultats pour le tournoi 2 joueurs : ***
lob (nihaakey) : 42 (13,3,2) - (61,11)
JSK (lounisAmazigh) : 40 (13,1,4) - (79,15)
lalya2 (hmdd) : 36 (10,6,2) - (91,14)
team1 (Kabegami) : 33 (8,9,1) - (71,6)
Tremblez! (Asparodia) : 33 (10,3,5) - (79,24)
DZPOWER (Raouf16) : 29 (9,2,7) - (73,45)
psg (luluperet) : 21 (7,0,11) - (66,106)
equipe2 (jordanupmc) : 17 (5,2,11) - (29,36)
team2 (3408247) : 6 (2,0,16) - (20,132)
team1 (ad50144124) : 0 (0,0,18) - (0,180)
```

Tournoi 4v4

```
 ***  Resultats pour le tournoi 4 joueurs : ***
DZPOWER (Raouf16) : 27 (9,0,0) - (62,2)
lob (nihaakey) : 24 (8,0,1) - (43,4)
JSK (lounisAmazigh) : 21 (7,0,2) - (44,10)
team4 (Kabegami) : 16 (5,1,3) - (42,10)
Lel (Asparodia) : 16 (5,1,3) - (33,14)
equipe4 (jordanupmc) : 10 (3,1,5) - (38,12)
team1 (ad50144124) : 10 (3,1,5) - (31,18)
mars (luluperet) : 6 (2,0,7) - (20,83)
team4 (3408247) : 3 (1,0,8) - (10,80)
lalya4 (hmdd) : 0 (0,0,9) - (0,90)
```

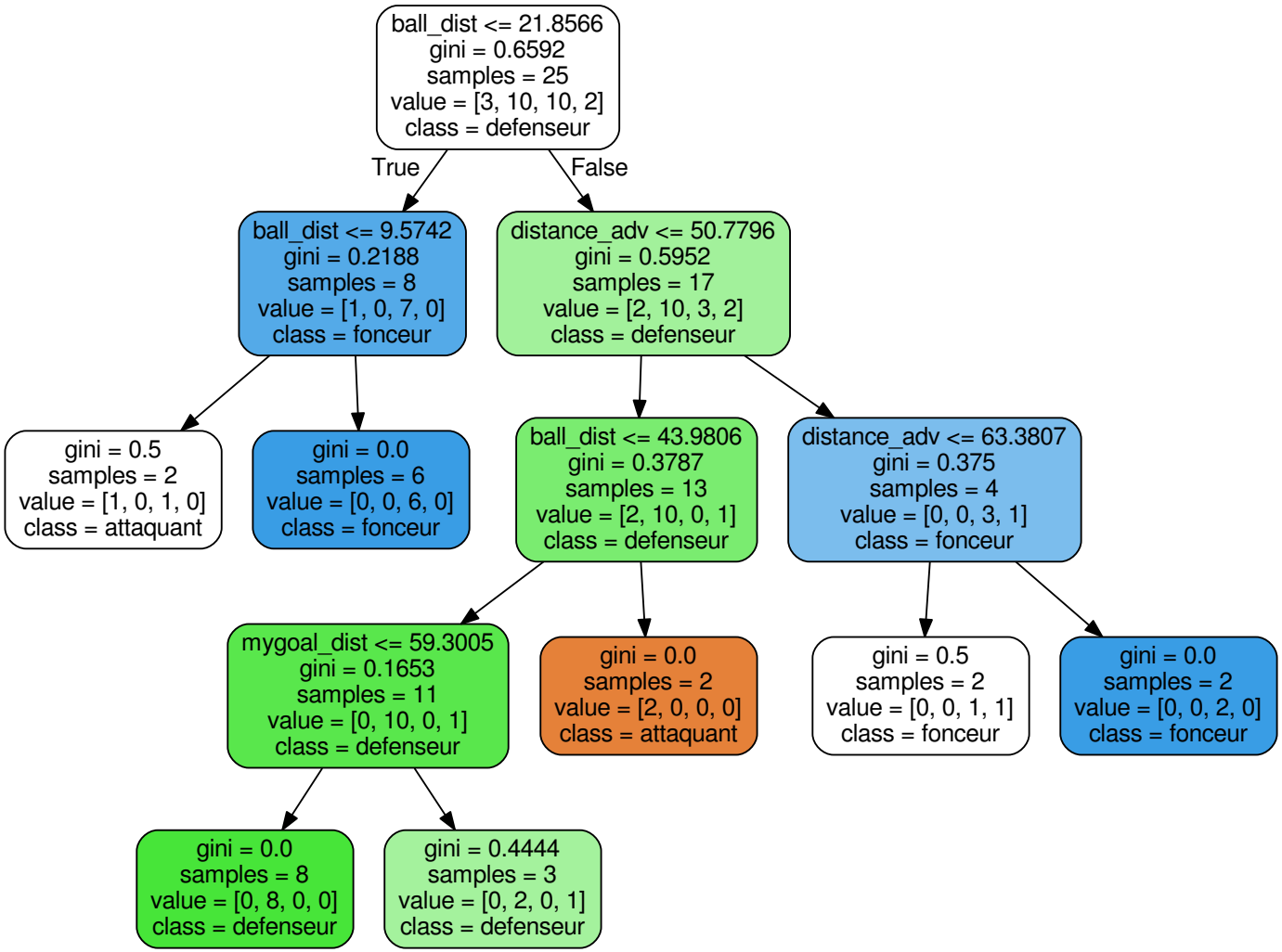
Un peu de recule sur les arbres de décisions

- Peut-on tout apprendre ?
- De quelle forme sont les règles apprises ?



Un peu de recule sur les arbres de décisions

- Peut-on tout apprendre ?
- De quelle forme sont les règles apprises ?



Plan

Inspiration biologique

Le cerveau

- Robuste, tolérant aux fautes
- Flexible, sait s'adapter
- Gère les informations incomplètes
- Capable d'apprendre

Composé de neurones !

- 10^{11} neurones dans un cerveau humain
- 10^{4} connexions par neurones
- Potentiel d'action, neuro-transmetteurs, période réfractaire
- Signaux excitateurs / inhibiteurs

Problèmes

- Opacité des raisonnements
- Opacité des résultats

Inspiration biologique

Le cerveau

- Robuste, tolérant aux fautes
- Flexible, sait s'adapter
- Gère les informations incomplètes
- Capable d'apprendre

Composé de neurones !

- 10^{11} neurones dans un cerveau humain
- 10^{4} connexions par neurones
- Potentiel d'action, neuro-transmetteurs, période réfractaire
- Signaux excitateurs / inhibiteurs

Problèmes

- Opacité des raisonnements
- Opacité des résultats

Historique

Prémisses

- Mc Cullch et Pitts (1943) : 1er modèle de neurone formel. Base de l'IA
- Règle de Hebb (1949) : apprentissage par renforcement du couplage synaptique

Premières réalisations

- Adaline (Widrow-Hoff, 1960)
- Perceptron (Rosenblatt, 1958-1962)
- Analyse de Minsky et Papert (1969)

Développement

- Réseau bouclé (Hopfield 1982)
- Réseau multi-couches (1985)

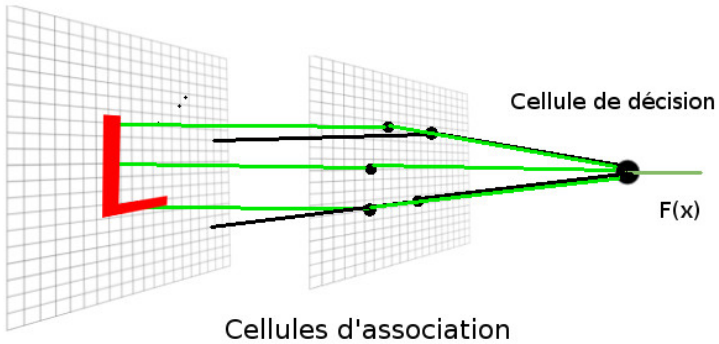
Deuxième renaissance

- Réseaux profonds (2000-)

Le perceptron de Rosenblatt (1960)

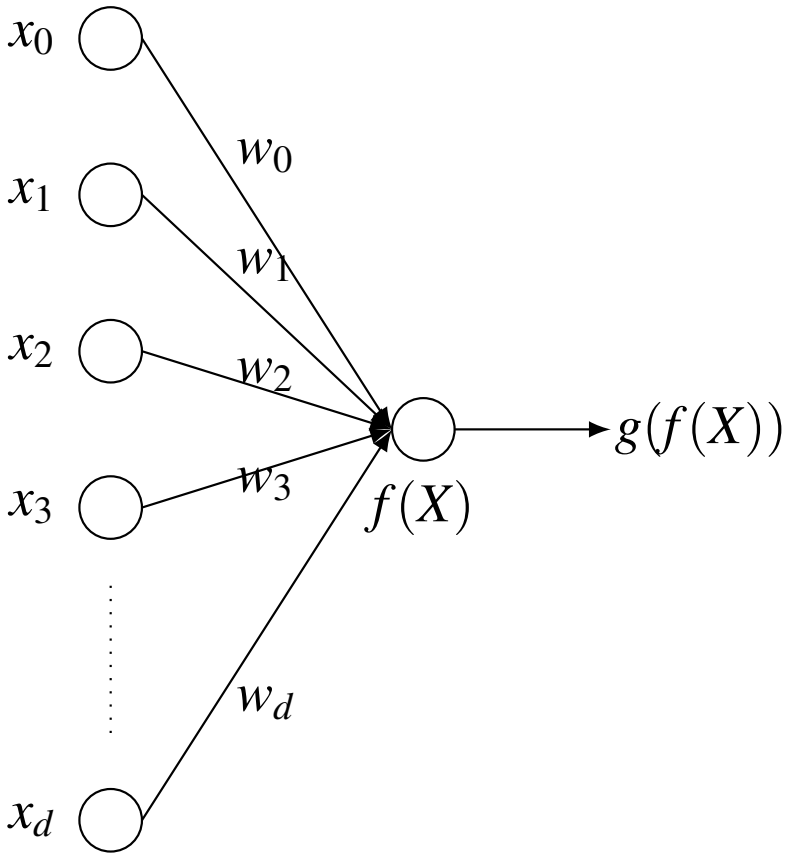
L'idée

- Reconaissance de forme (*pattern*) entre deux classes
- Inspirée cortex visuel



- Chaque cellule d'association produit une sortie $f_i(S)$ en fonction d'un stimulus
- La cellule de décision répond selon une fonction seuil $f_d(\sum w_i f_i(S_i))$

Formalisation



Le perceptron considère

- $f(\mathbf{x}) =< \mathbf{x}, \mathbf{w} >= \sum_{i=1}^{d} x_i w_i$
- Fonction de décision :
  $g(x) = sign(x)$

→ Sortie :
$g(f(\mathbf{x})) = sign(< \mathbf{x}, \mathbf{w} >)$

Considérations géométriques

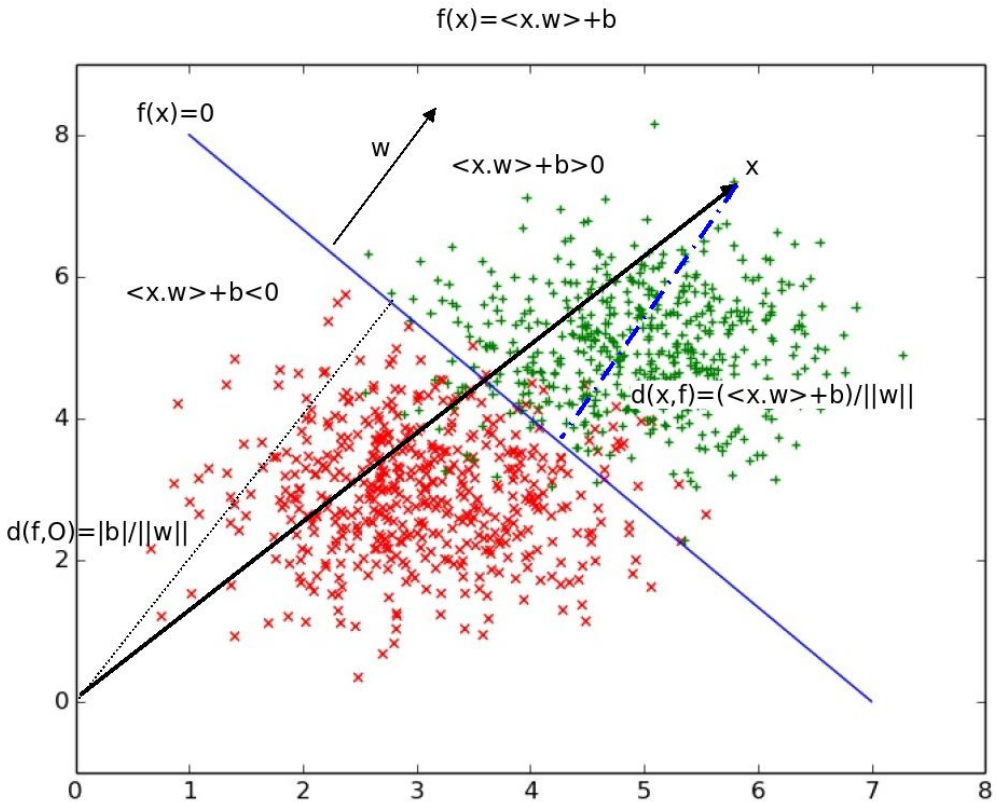
Soit $y(x)$ la sortie attendue :

- Que représente $\mathbf{w}$ par rapport à la séparatrice ?
- Que représente $< \mathbf{wx} >$?
- Que représente $y(x) < \mathbf{wx} >$?
- A quoi correspond la règle de mise à jour :
  - Si $(y(x) < \mathbf{w}.\mathbf{x} >) > 0$ ne rien faire
  - Si $(y(x) < \mathbf{w}.\mathbf{x} >) < 0$ corriger $\mathbf{w} = \mathbf{w} + y(x)x$?

Considérations géométriques

Soit $y(x)$ la sortie attendue :

- Que représente $\mathbf{w}$ par rapport à la séparatrice ?
- Que représente $< \mathbf{wx} >$?
- Que représente $y(x) < \mathbf{wx} >$?
- A quoi correspond la règle de mise à jour :
  - Si $(y(x) < \mathbf{w}.\mathbf{x} >) > 0$ ne rien faire
  - Si $(y(x) < \mathbf{w}.\mathbf{x} >) < 0$ corriger $\mathbf{w} = \mathbf{w} + y(x)x$?



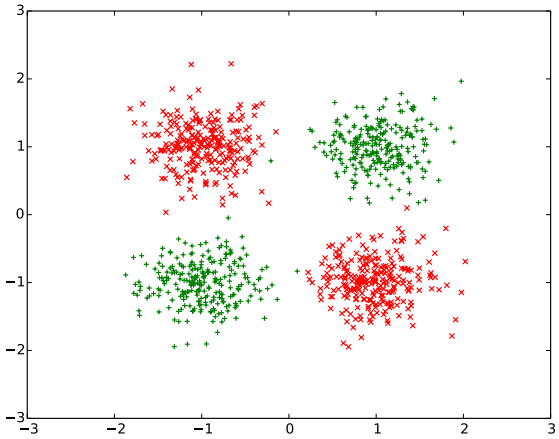
Algorithme de résolution

Algorithme du perceptron

- Initialiser au hasard $\mathbf{w}$
- Tant qu'il n'y a pas convergence :
  - pour tous les exemples (x^i, y^i) :
    - si $(y^i < \mathbf{w}.\mathbf{x}^i >) < 0$ alors $\mathbf{w} = \mathbf{w} + \epsilon y^i x^i$
- Décision : $f(x) = sign(< \mathbf{wx} >)$

Problèmes “durs”

Non linéairement séparable

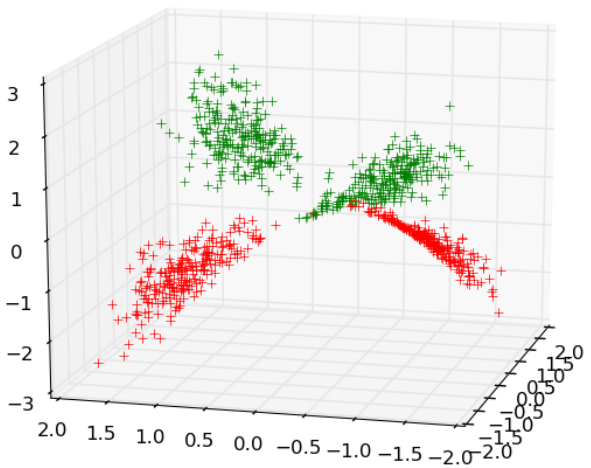


Que faire ?

Problèmes “durs”

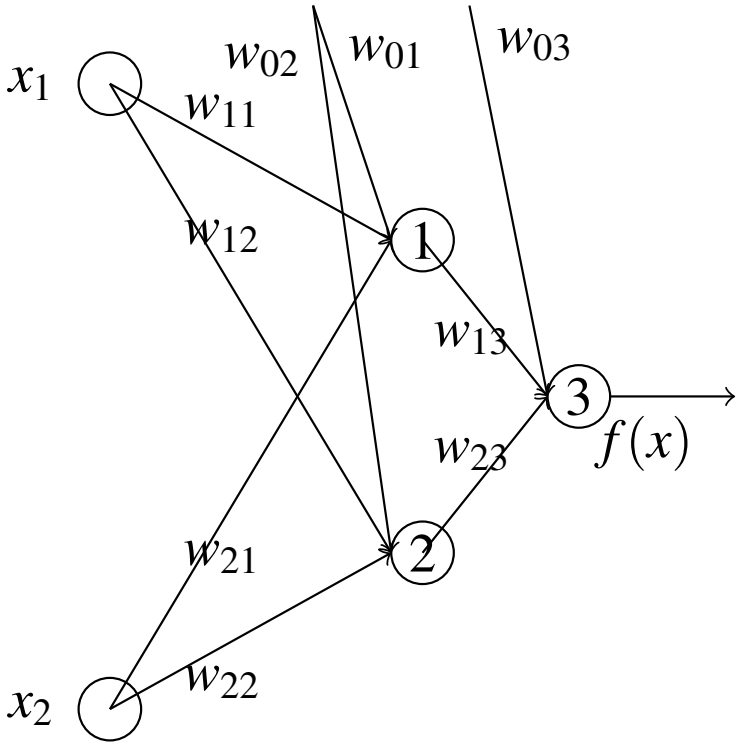
Transformation de la représentation

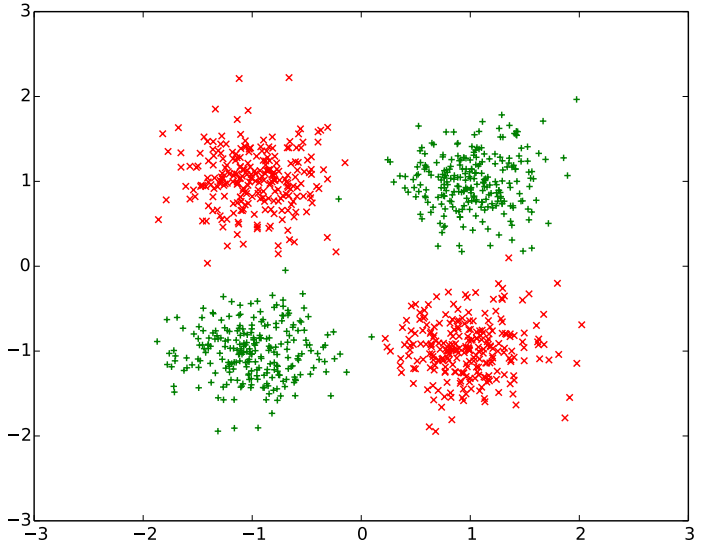
- On augmente d’une dimension : $(x_1, x_2) \rightarrow (x_1, x_2, x_1 x_2)$



- Le problème est de nouveau séparable linéairement !
- Autre solution ?

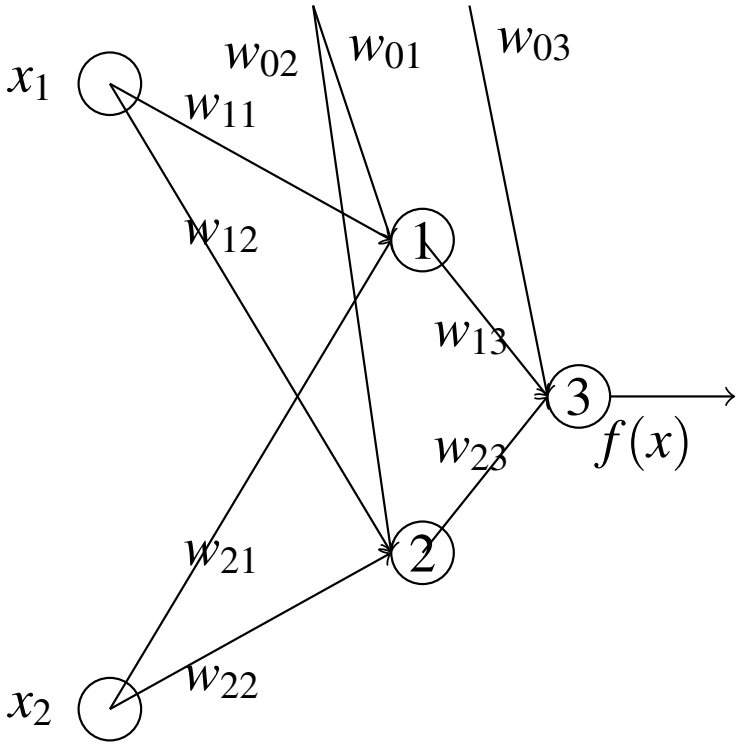
Deux neurones

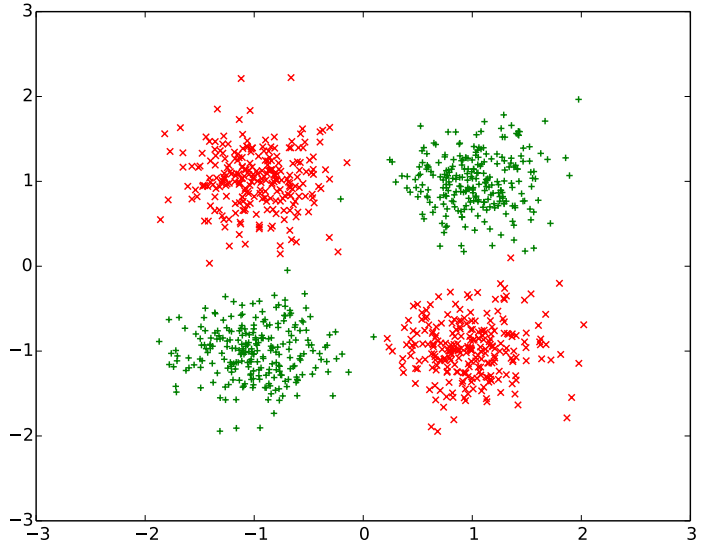




- Combiner des neurones → augmente l'expressivité
- Création de dimensions nouvelles, de nouveaux features

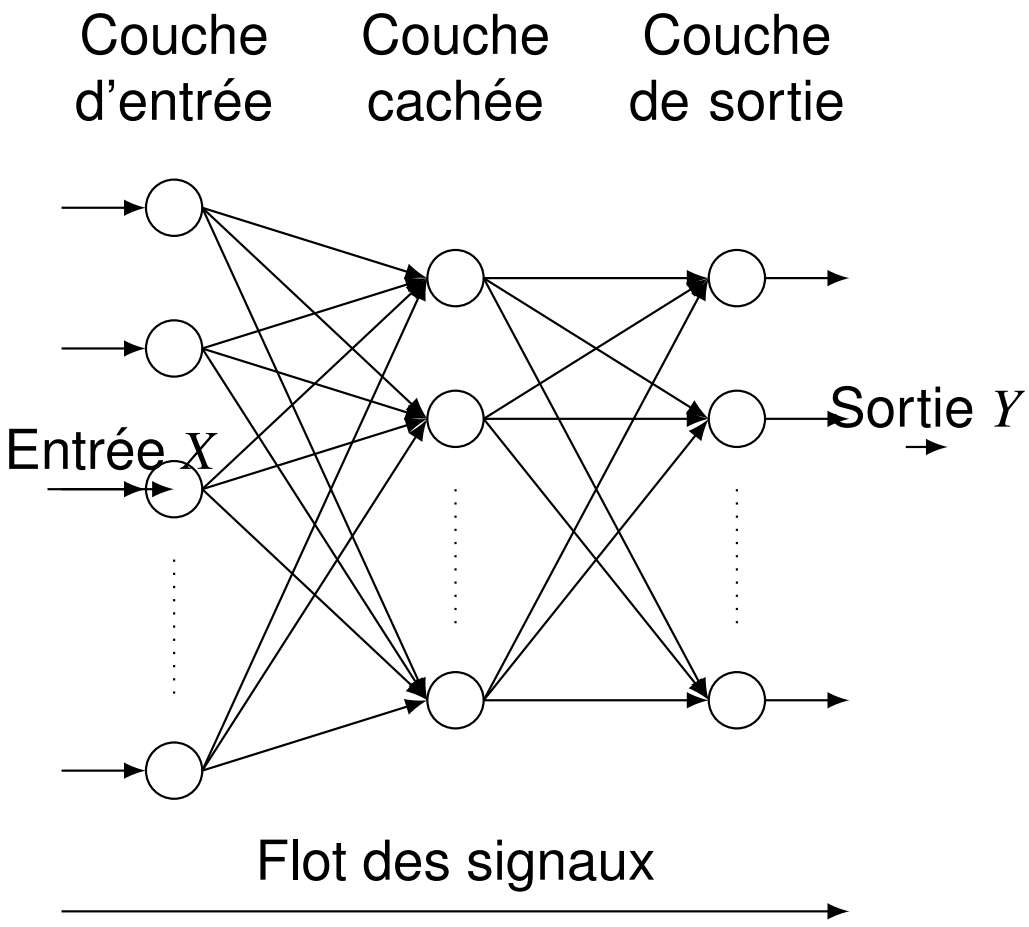
Deux neurones





- Combiner des neurones $\rightarrow$ augmente l'expressivité
- Création de dimensions nouvelles, de nouveaux features

Topologie typique



Pour chaque neurone :

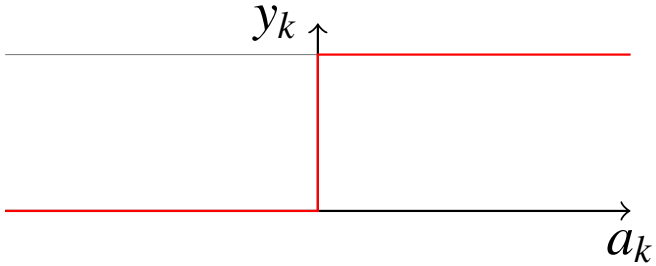
$$y_k = g\left(\sum_{j=0}^{d} w_{j,k}\phi_j\right) = g\left(a_k\right)$$

où

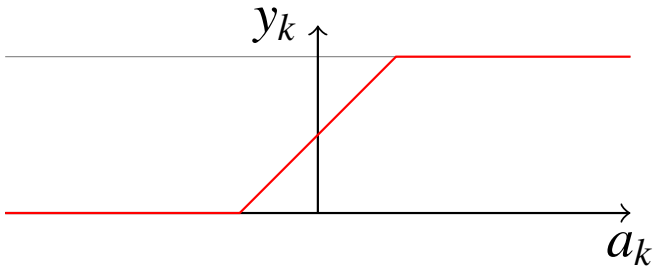
- $w_{j,k}$: poids de la connexion de la cellule j à la cellule k
- a_k : activation de la cellule k
- g : fonction d'activation

Fonction d'activation

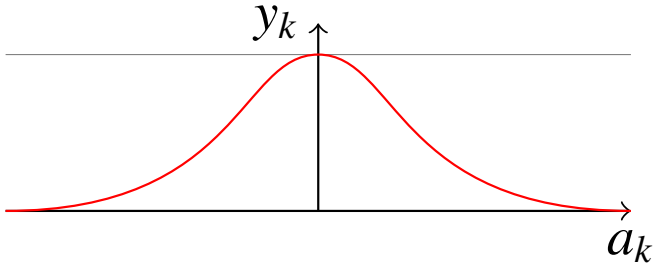
- Fonction à seuil



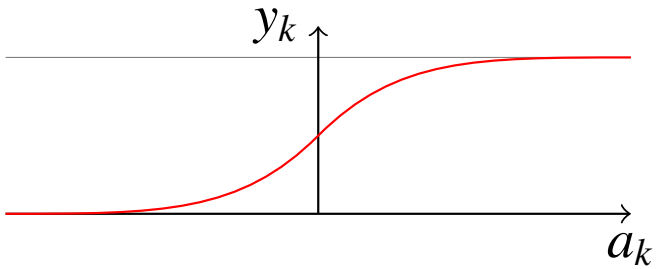
- Fonction à rampe



- Fonction radiale

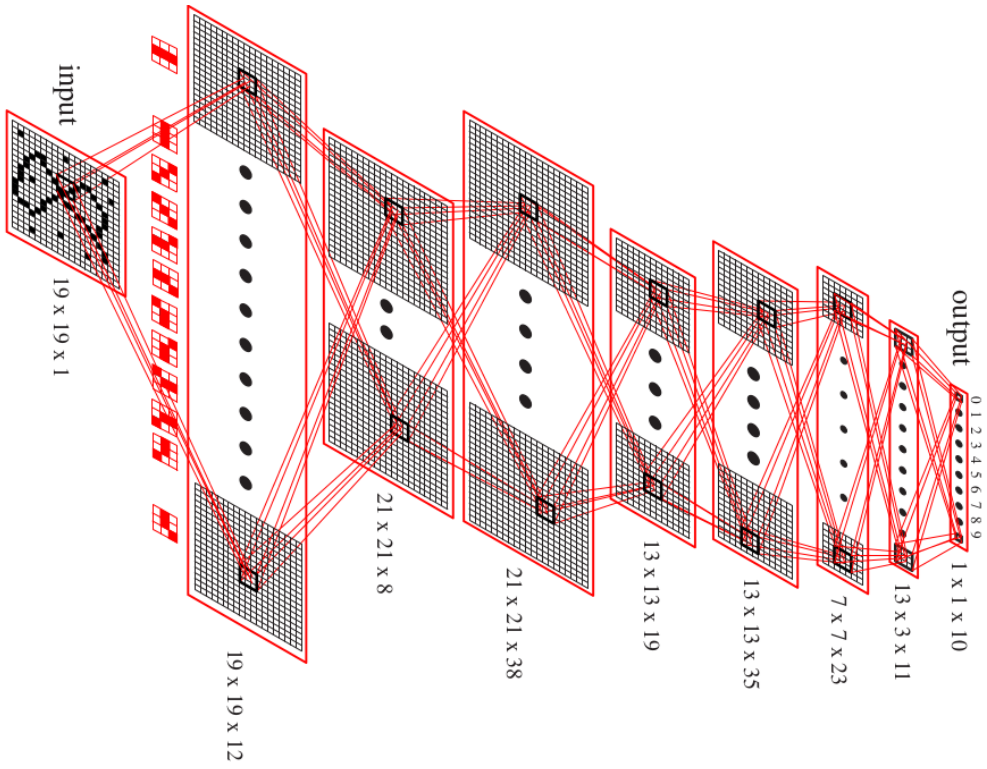


- Fonction sigmoïde



  - $g(a) = \frac{1}{1+\exp(-a)}$
  - $g'(a) = g(a)(1 - g(a))$

Réseaux de neurones



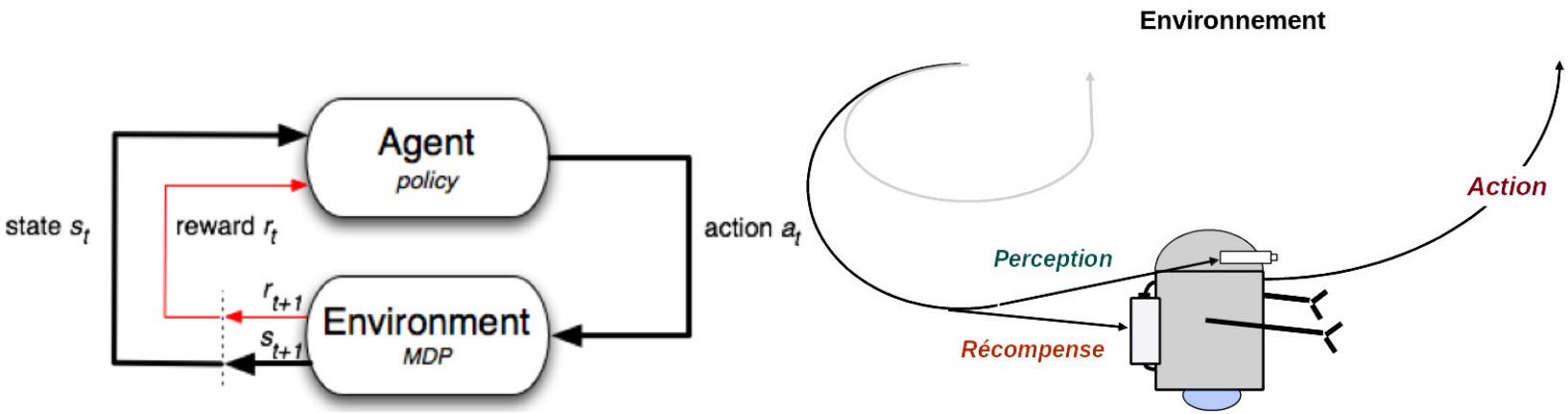
- Pouvoir expressif très grand
- Plus difficile à apprendre
- Résaux profonds (*Deep Learning*) : ont révolutionné la reconnaissance d'image et beaucoup d'autres domaines depuis une dizaine d'année.

Plan

Modélisation agent

- Environnement : tout ce qui est extérieur à l'agent
- Etat : ce que perçoit l'agent
- Action : ce que peut décider l'agent
- Récompense : donnée par l'environnement de l'agent

$\Rightarrow$ Objectif : maximiser les récompenses



Formalisation

Modélisation

- L'environnement : un ensemble d'observations $\mathcal{O} = \{O_1, \ldots, O_o\}$
- Les états : $\mathcal{S} = \{s_1, \ldots, s_s\} \Rightarrow$ un vecteur de $\mathbb{R}^d$, la sortie de la fonction `gen_feature`
- Les actions : $\mathcal{A} = \{a_1, \ldots, a_a\} \Rightarrow$ les stratégies "simples"
- Les récompenses dans $\mathbb{R}$.

Une séquence de jeu $\mathcal{T}$

- c'est une séquence de triplets (observation,action,récompense) : $\{(o_{t_1}, a_{t_1}, r_{t_1}), (o_{t_2}, a_{t_2}, r_{t_2}), \ldots, (o_{t_T}, a_{t_T}, r_{t_T})\}$
- La récompense : $R = r_{t_1} + r_{t_2} + \ldots r_{t_T}$
- A noter : $s_i =$ gen_feature(o_t), l'état "vu" par l'agent est une transformation de l'observation.

Apprentissage par renforcement

Principe

- apprendre une *politique* pour réagir à l'environnement

$\Rightarrow$ une fonction $\pi(s)$ renvoyant pour chaque état une action ou une distribution de probabilité sur les actions

- Objectif : trouver la politique qui maximise l'espérance des récompenses

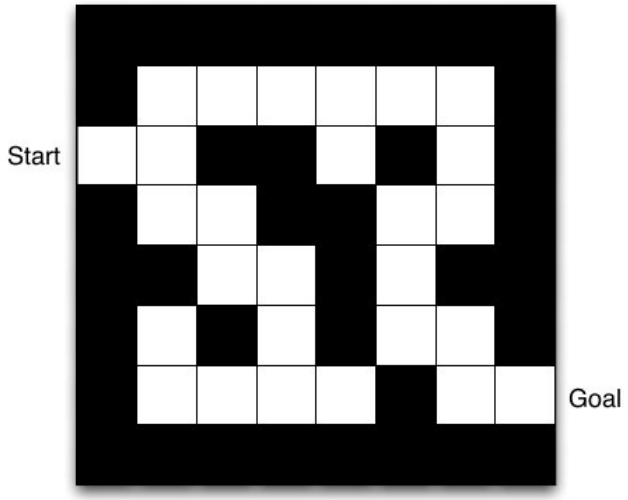
Notion fondamentale : fonctions (équivalentes)

- de valeur d'état
  $v_\pi(s) = \mathbb{E}_\pi(r_{t+1} + \gamma r_{t+2} + \gamma^3 r_{t+3} + \ldots + \gamma^T r_T | s_t = s)$ prédit le score d'un état
- de valeur d'action $Q_\pi(s,a) = r_t + \gamma \sum_{s'} P_\pi(s'|s) v_\pi(s)$ prédit le score d'une action entreprise dans un état donné.

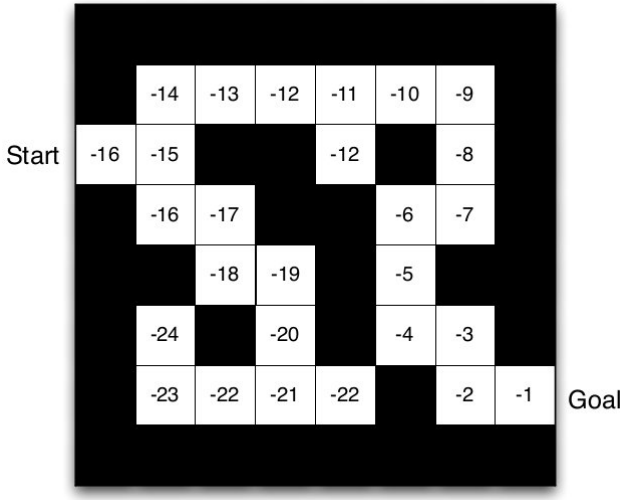
Pour évaluer ses fonctions, besoin de simulations (beaucoup) !

Exemple : labyrinthe

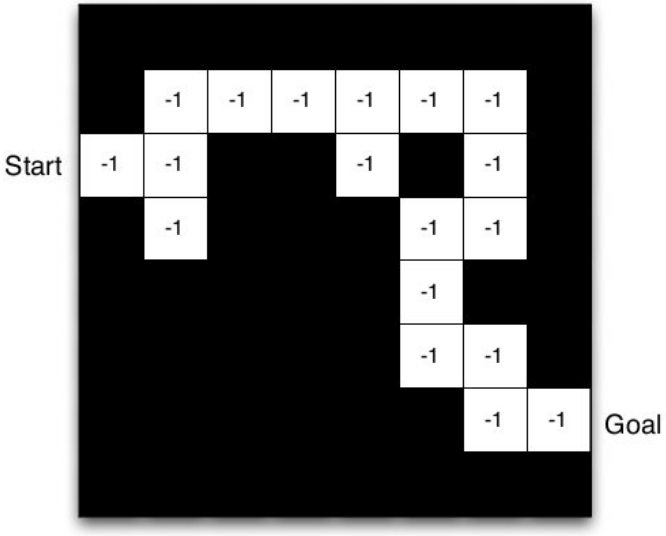
environnement



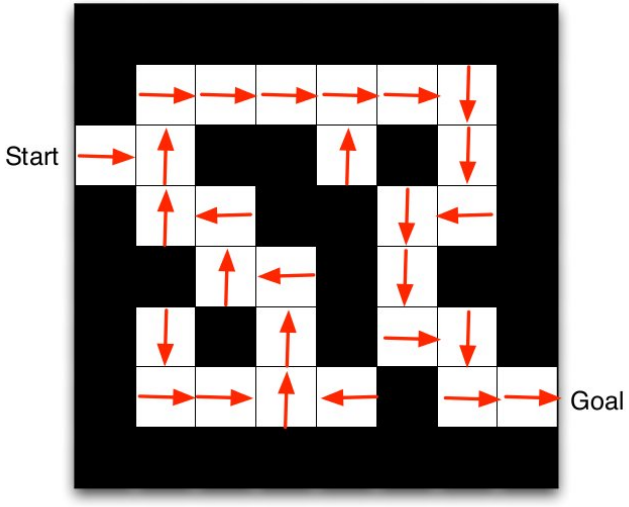
valeur d'état



récompense



meilleur action



Apprentissage par renforcement

Beaucoup de variantes possibles

- Value based : pas d'estimation de la politique, uniquement de la fonction de valeur
- Policy based : pas de fonction de valeur, estimation de la politique
- Actor critic : mixe des deux
- Model based : estime le modèle de transition entre états, et les récompenses attendues
- Model free : pas de modèle à estimer

Exploration vs exploitation : essai-erreur

- Au tout début, on ne connait rien !
- la politique initiale : au hasard
- Besoin de tester de nouveaux choix pour certains états, voir si ca améliore le score global $\Rightarrow$ Exploration
- Mais pas trop ! se guider des bons choix actuels $\Rightarrow$ Exploitation

Q-learning

Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (for each step of episode):
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$
    $S \leftarrow S'$;
  until S is terminal