IEEE
Student Branch
Ecole Nationale Polytechnique

Introduction à l'apprentissage par renforcement

Houssem MEGHNOUDJ

Sommaire

1. Introduction
2. Environnement
   - Type d'environnements
   - Principales caractéristiques
3. Agent et son rôle dans l'environnement
4. Exploration - exploitation
5. Value function
6. Exemples

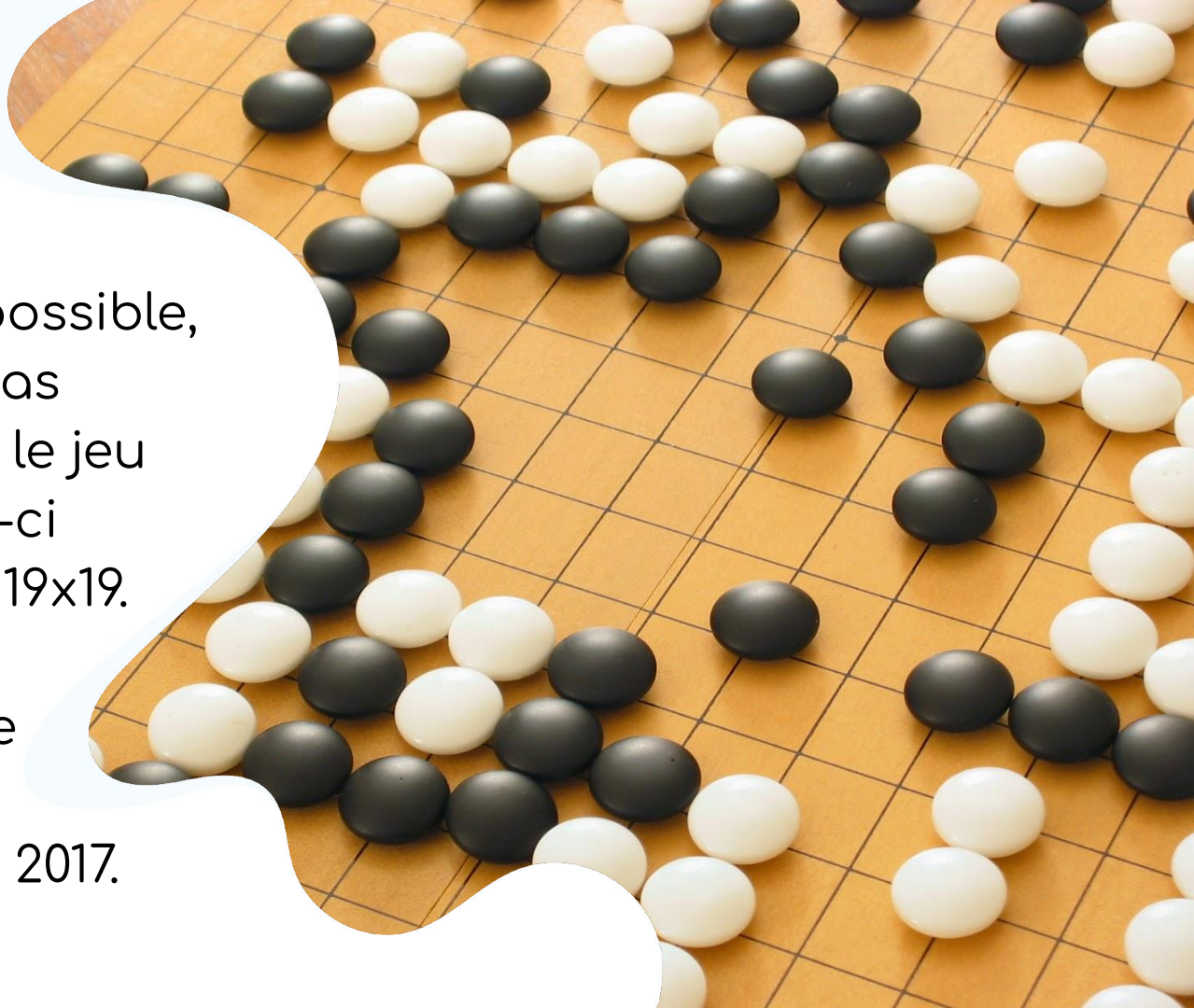
1997 : Gary Kasparov champion mondial d'échecs est détrôné par DeepBlue, une Intelligence Artificielle (IA)

Une grille d'échecs quadrillée 8x8 ne permet qu'un nombre restreint de manœuvres, ce qui facilite l'exploration de toutes les possibilités de jeu. C'est d'ailleurs qui a permis la victoire de la machine.

1. Introduction

A l'opposé d'un jeu d'échecs, il n'est pas commode, voire impossible, d'explorer tous les cas envisageables dans le jeu de go puisque celui-ci dispose d'une grille 19x19.

Raison pour laquelle l'humain a demeuré imbattable jusqu'en 2017.



1. Introduction

2017 : Nouvelle victoire de l'IA contre le champion mondial du jeu de go "Ke Jie" suite à l'utilisation de l'apprentissage par renforcement.

Un exploit puisqu' AlphaGo (nom de l'IA) ne disposait initial-ement que du plateau de jeu et de ses règles.

1. Introduction


OPENAI FIVE BENCHMARK
1. Introduction
Show Fight Recap
OpenAI 3 (Bot)
OpenAI 5 (Bot)
OpenAI 4 (Bot)
OpenAI 1 (Bot)
OpenAI 2 (Bot)
MoonMeander: gg
Blitz: gg
2019 : OpenAi five bat les champions du monde l'équipe OG sur le jeu dota 2

Les jeux présentent un terrain d'expérimentation intéressant de par leur complexité ainsi que le nombre de stratégies existantes.

De ce fait, une intelligence capable de résoudre un jeu, sera forcément en mesure d'appréhender des problèmes réels.

2. Environnement

2. Environnement

On souhaite qu'un robot apprenne par lui-même à marcher et ce par renforcement. Deux cas se présentent alors :

1- L'utilisation d'un environnement réel.

2- L'utilisation d'un environnement simulé.

Environnement réel

L'implémentation de l'algorithme d'apprentissage par renforcement se fait directement dans le robot avant de le laisser apprendre de manière autonome.

Inconvénients :

- Temps d'apprentissage conséquent.
- Danger pour le robot (risque de chute comme pour un bébé qui essaie de marcher).
- Danger pour son entourage.

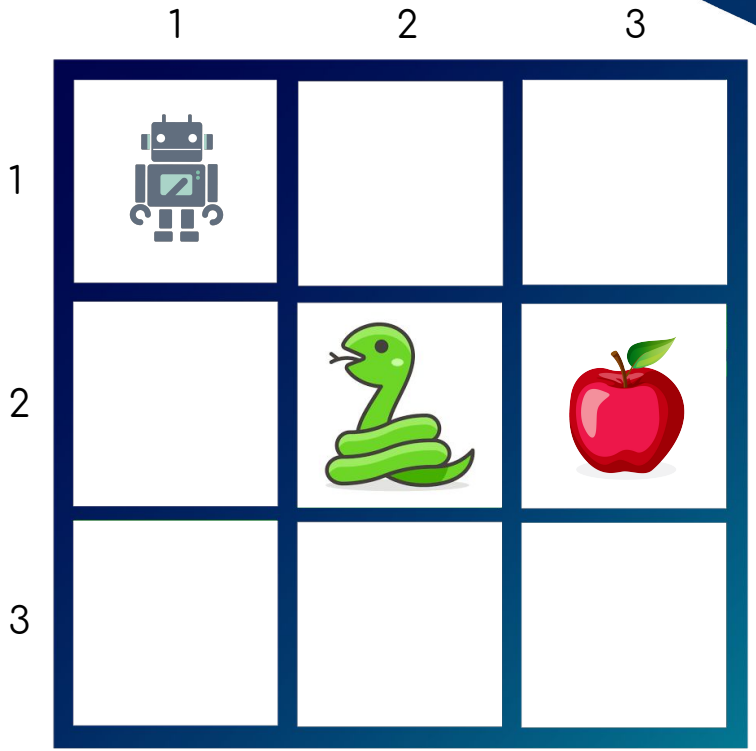
Exemple d'une main de robot qui a appris à résoudre le Rubik's Cube en utilisant le Reinforcement Learning (RL)

Environnement simulé

Créer un environnement virtuel, ou un simulateur, le plus représentatif possible de la réalité pour permettre au robot d'apprendre à marcher, puis transférer l'intelligence développée vers le robot réel une fois la tâche accomplie.

Avantage :

- Gain de temps grâce aux facteurs suivants :
  - 1- Parallélisme : permettant de créer plusieurs environnements dans lesquels différents robots peuvent apprendre simultanément.
  - 2- Rapidité relative de la simulation.
- Diminution du risque.



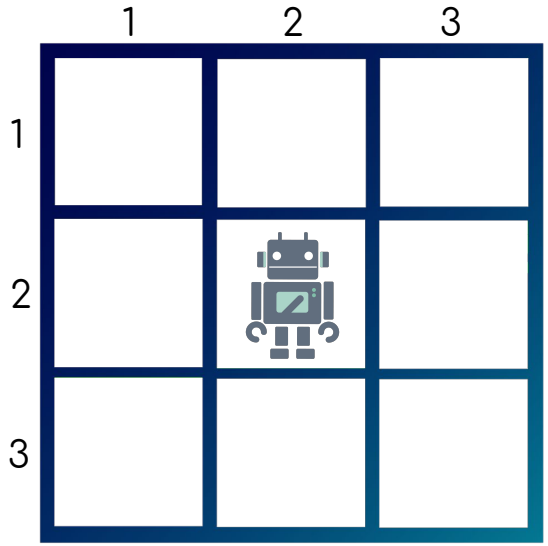
Un environnement est un monde virtuel ou réel où notre intelligence évolue en vue d'accomplir une tâche donnée.

Les états (states)

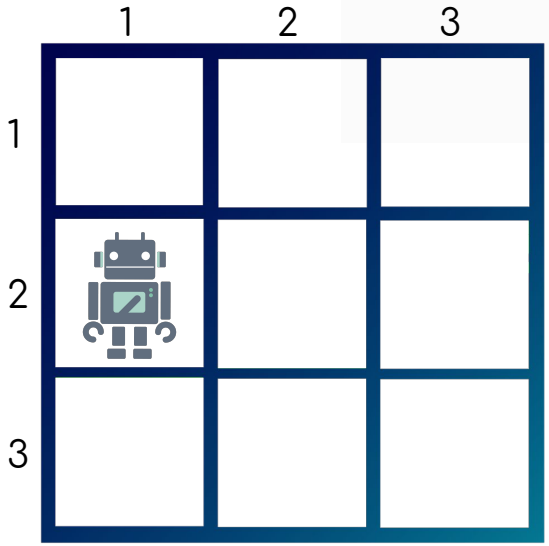
Un état est la situation actuelle dans laquelle se trouve l'environnement.

Dans notre exemple, les états prennent les 9 positions possibles du robot dans la grille.

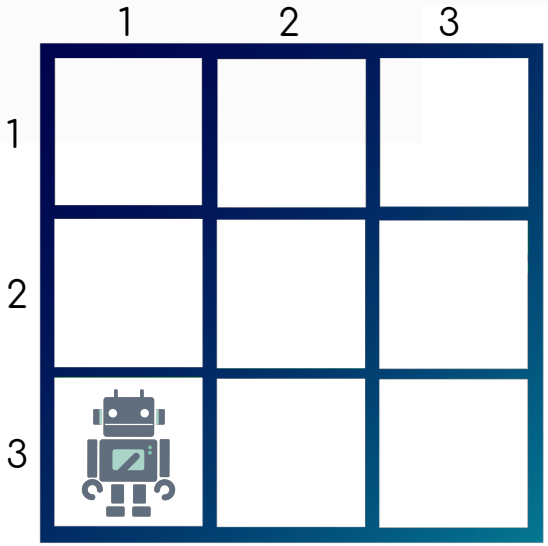
L'environnement du jeu de Pong peut avoir 7 197 120 états différents.



Etat (2, 2)



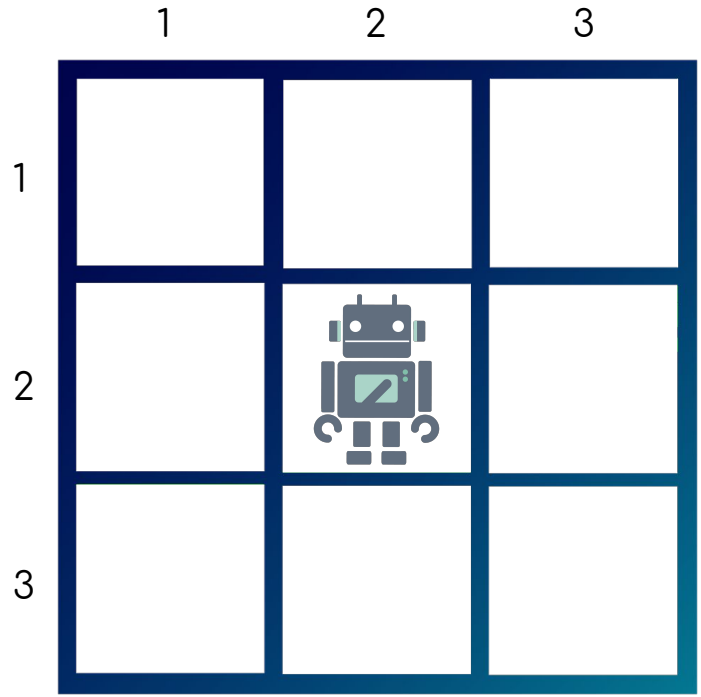
Etat (2, 1)



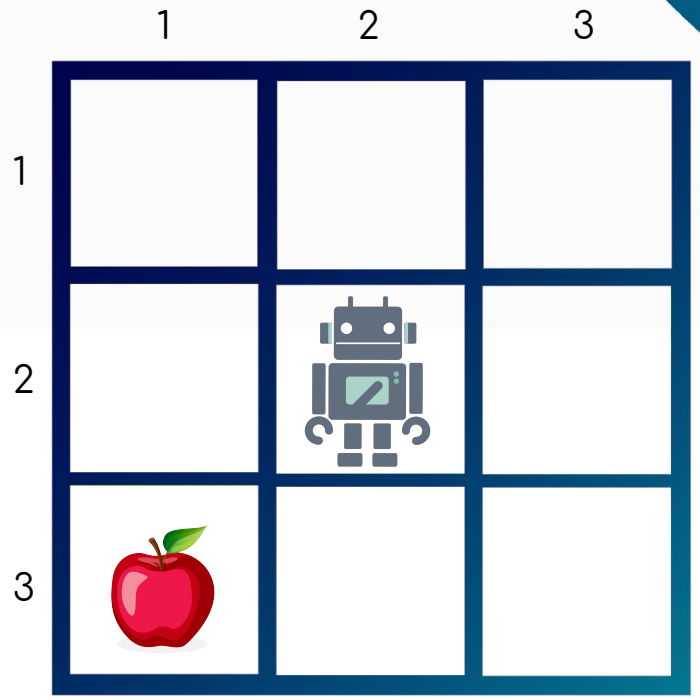
Etat (3, 1)

Les états (states)

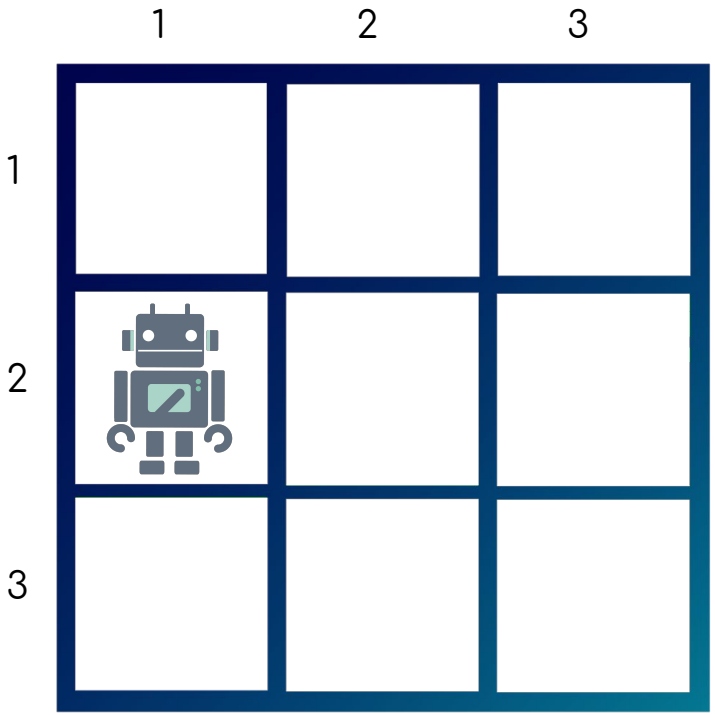
Un état est la situation actuelle dans laquelle se trouve l'environnement.



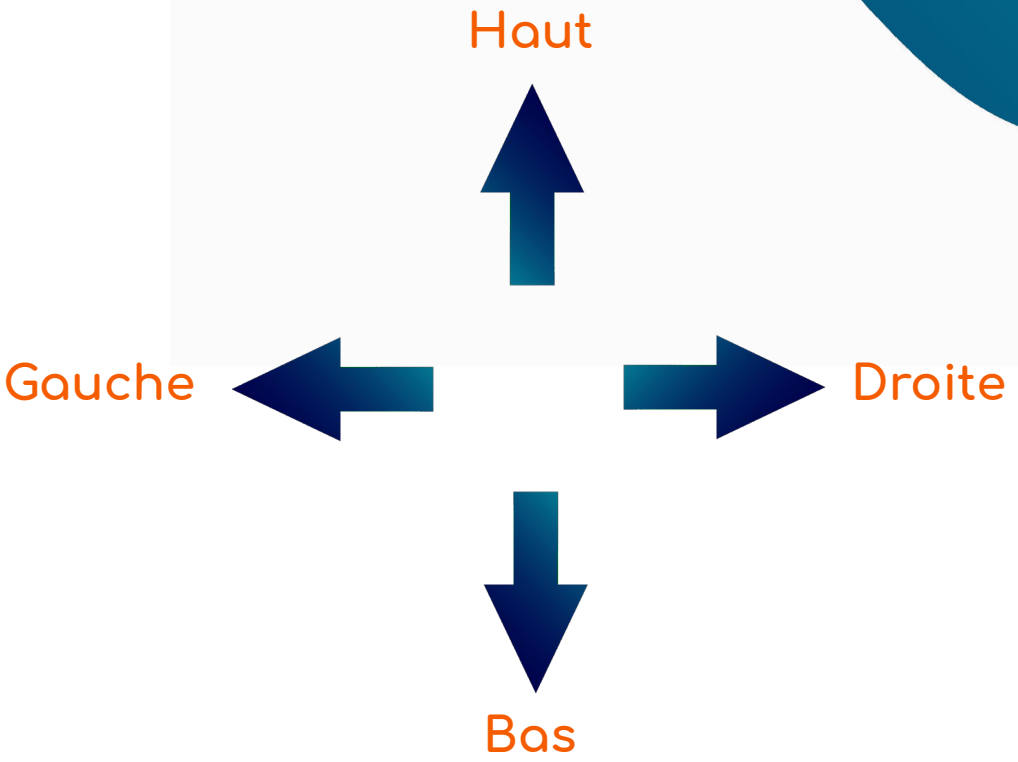
!=



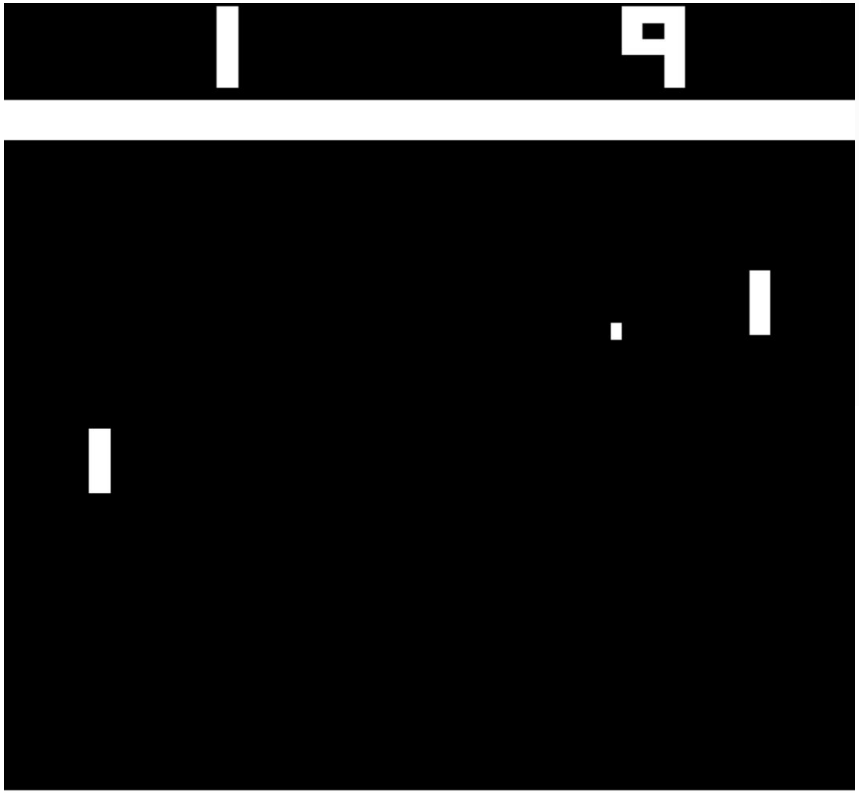
Les actions



Etat (2, 1)



Les actions

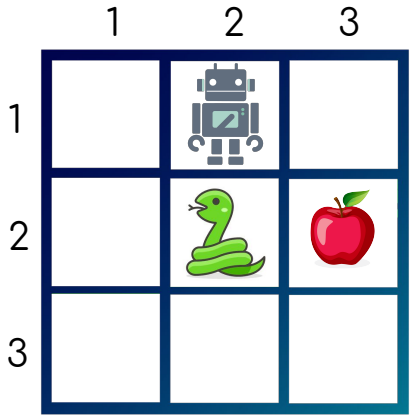


Haut

No-op

Bas

Les récompenses (rewards)

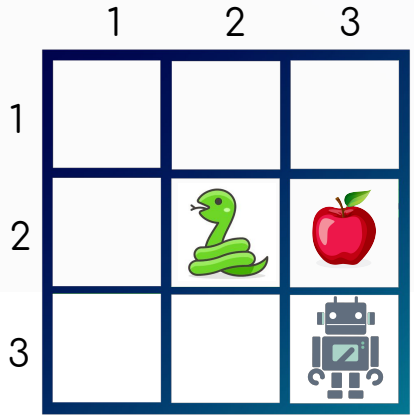


Bas

Reward = -10

—> Si notre IA accomplit une bonne action, elle recevra une récompense positive.

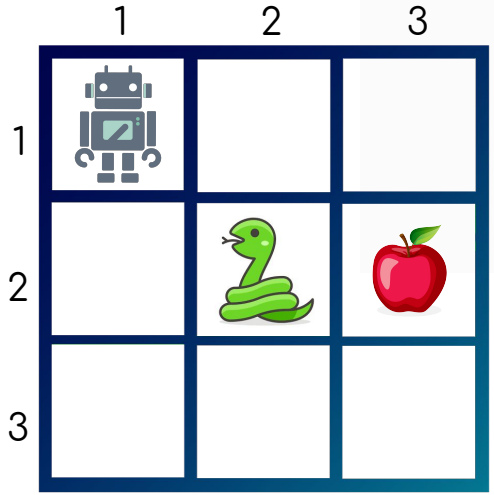
—> Sinon, une récompense négative lui sera attribuée.



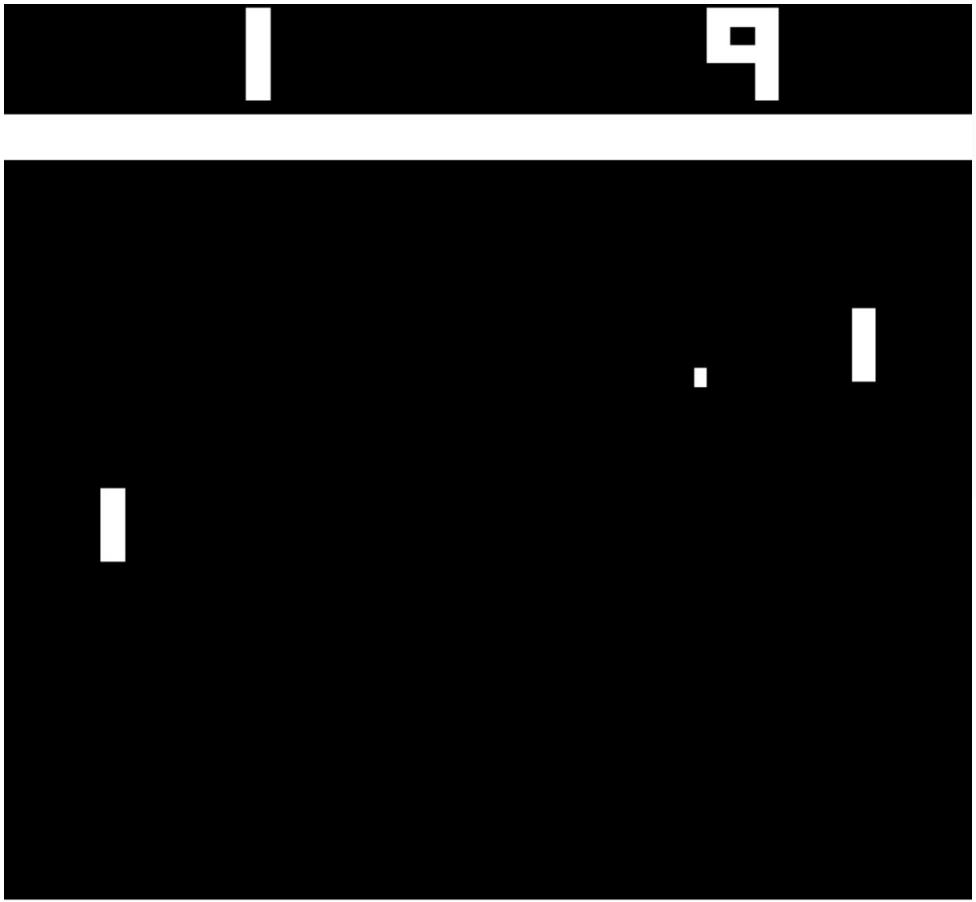
Haut

Reward = +10

La récompense sera définie suivant la manière avec laquelle une tâche souhaité est accomplie.

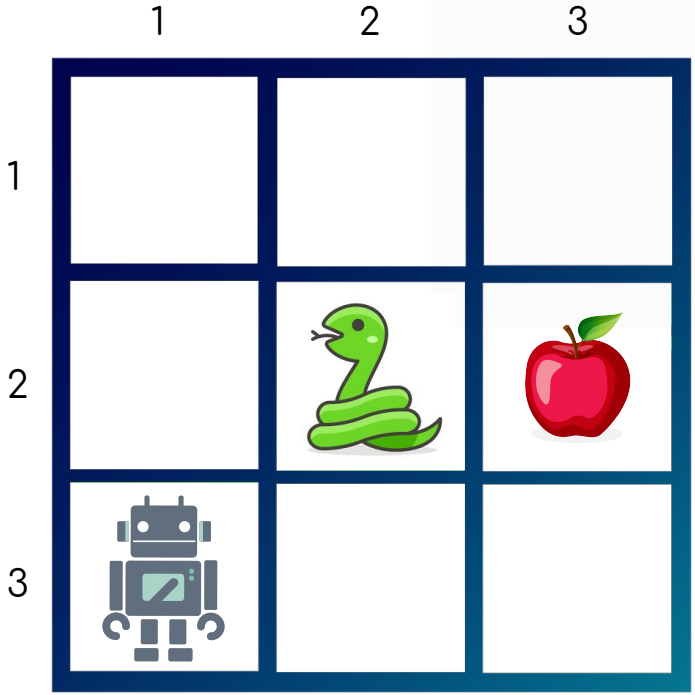


Total reward pour le chemin du haut = -1 -1 +10 = +8
Total reward pour le chemin du bas = -1 -1 -1 -1 +10 = +6

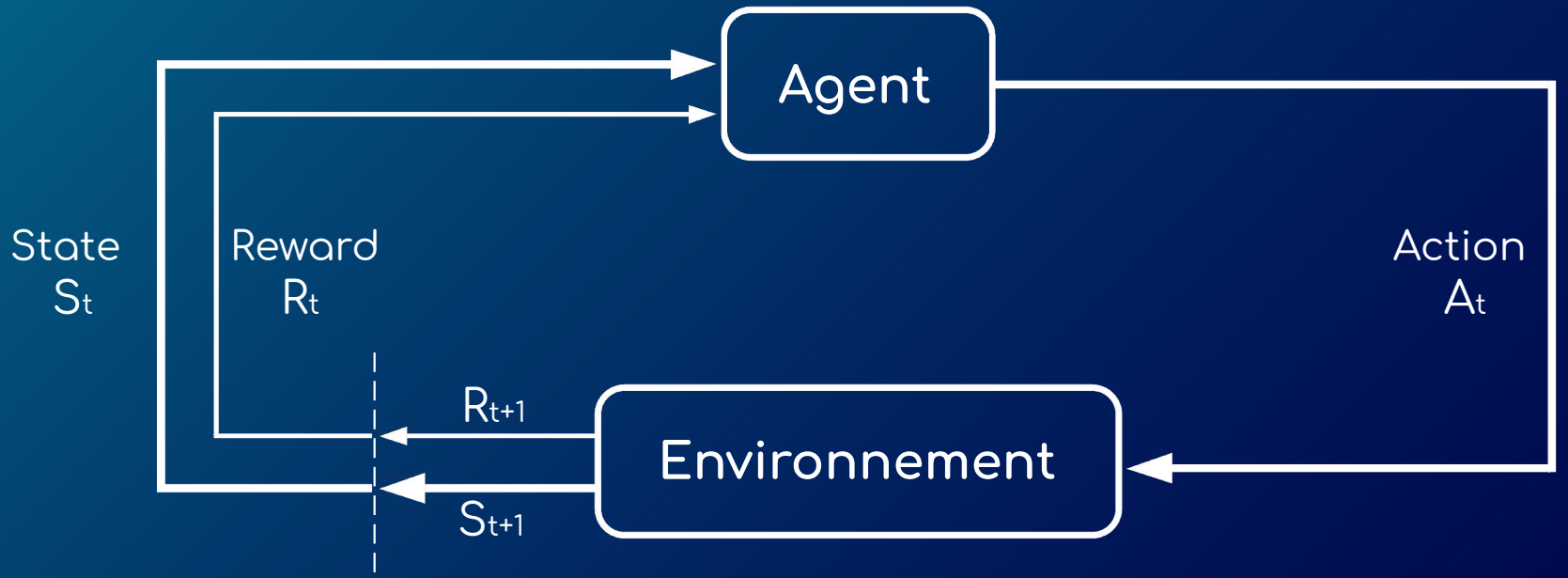


Récompenses pour PONG

L'intelligence placée dans l'environnement sera appelée agent.



Relation agent environnement



Rôle de l'agent dans l'environnement

Le rôle d'un agent dans l'environnement est de trouver un comportement, ou une politique, qui maximise les récompenses dans son environnement.

L'apprentissage par renforcement vise l'optimisation du comportement d'un agent dans un environnement donné.

Comment maximiser les récompenses ?

Exploration VS Exploitation

Bon restaurant habituel

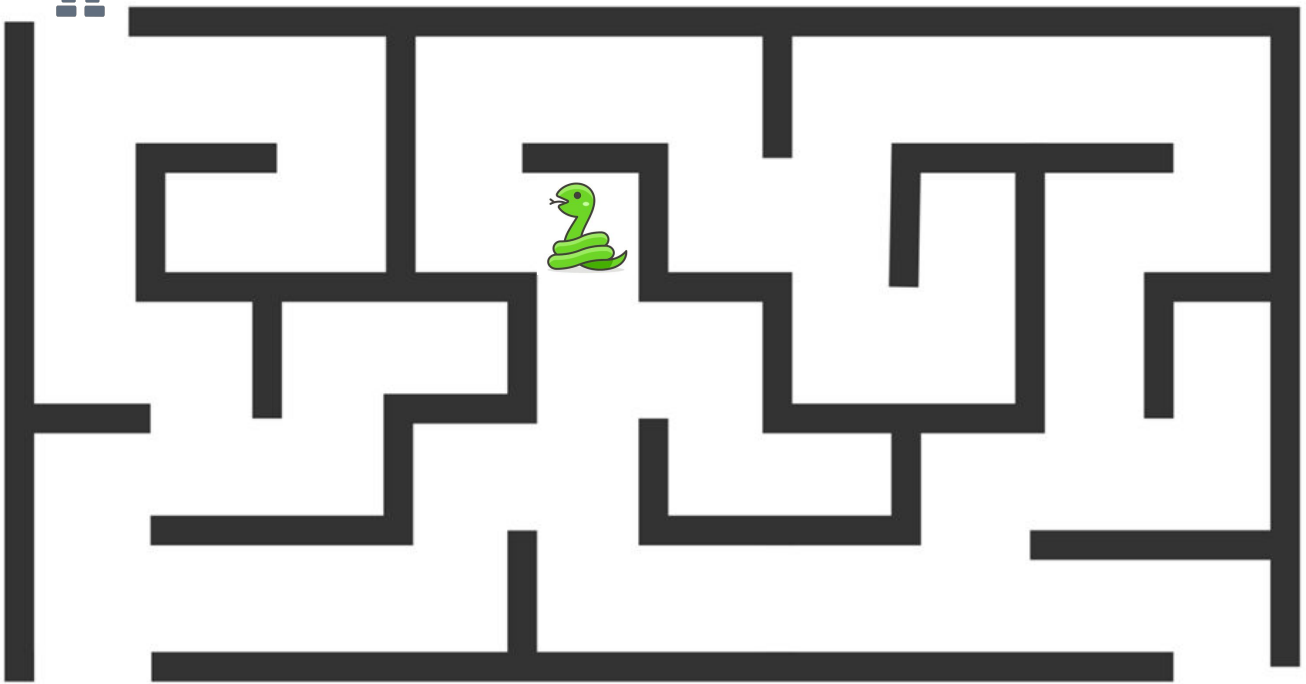
Nouveau restaurant

Exploration VS Exploitation

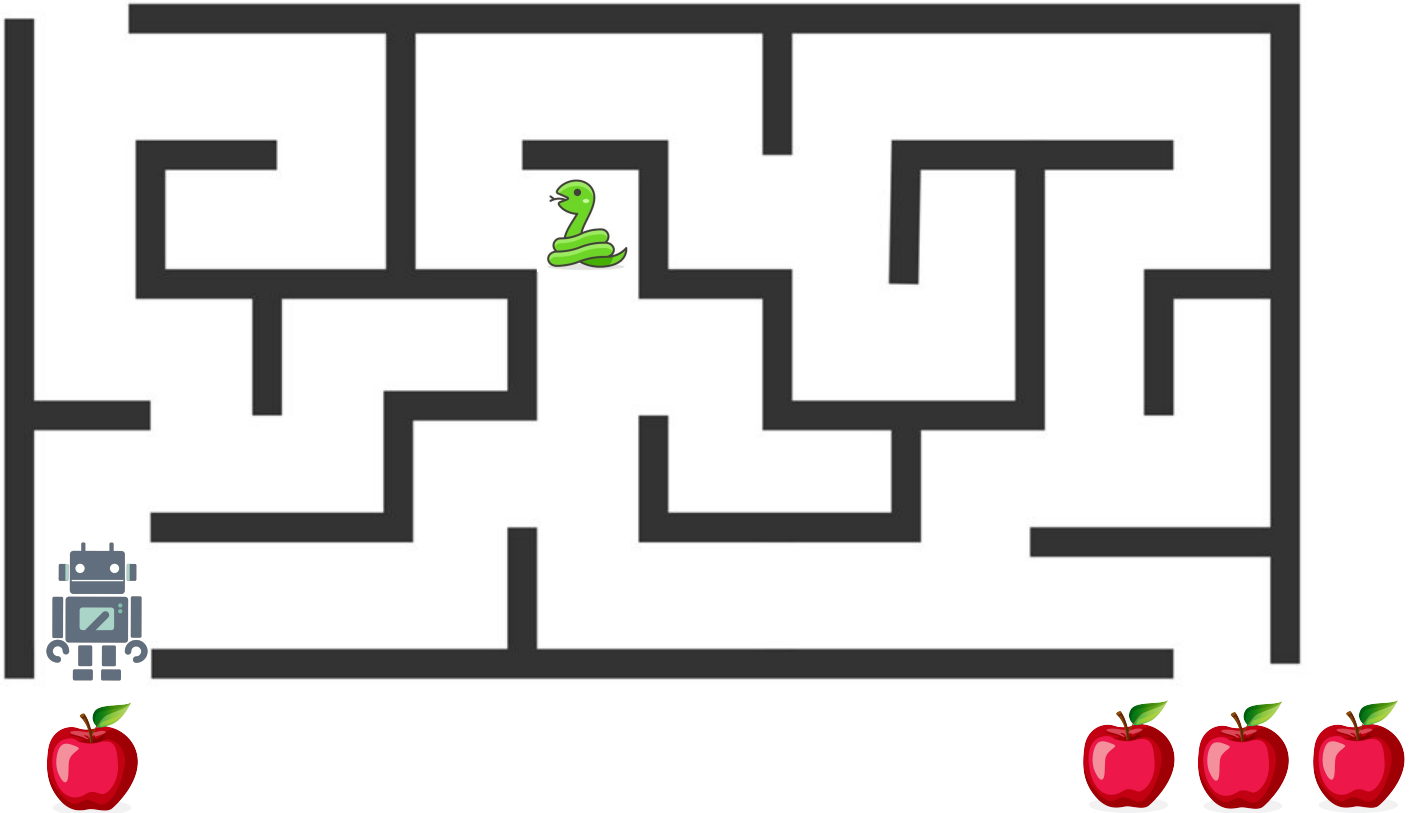
Un bébé prend tout et n'importe quoi
et le met dans sa bouche —> il fait de l'exploration.

Une personne âgée entreprend les
actions qu'elle pense être optimales
—> elle fait de l'exploitation.

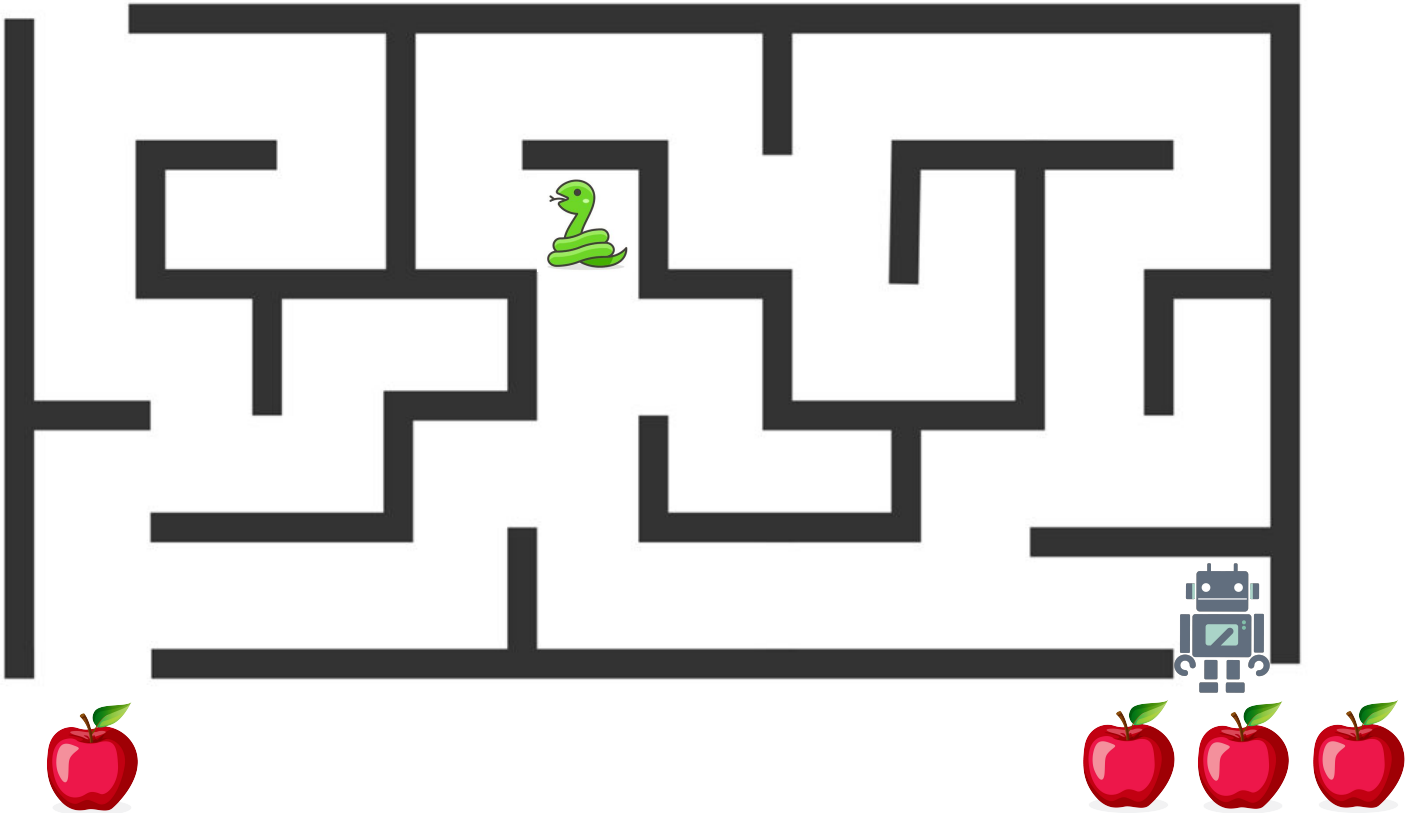
Vu que l'environnement est inconnu on commencera à l'explorer en faisant des actions aléatoires.



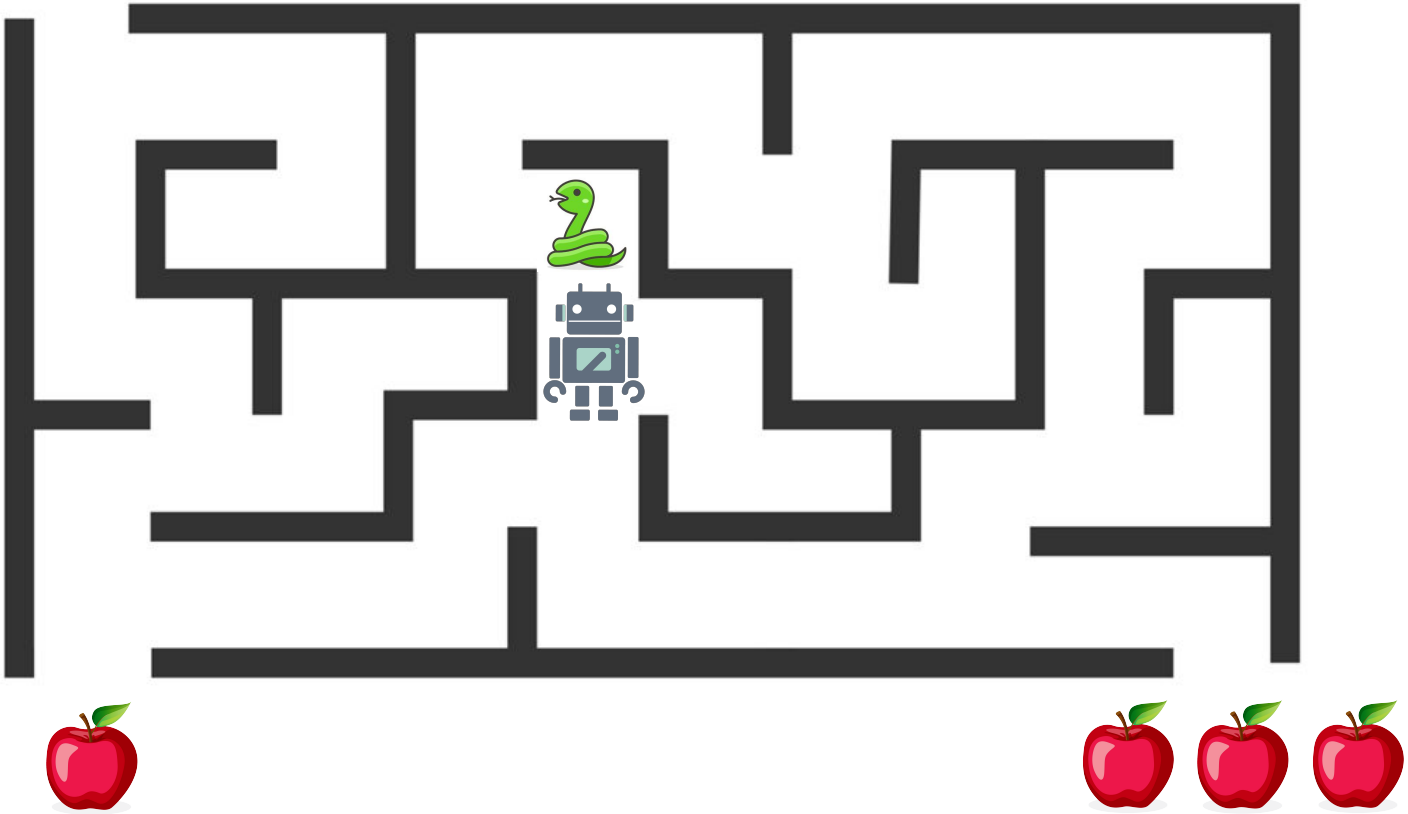
Une fois qu'une récompenses a été trouvée par hasard, on va essayer de l'obtenir à nouveau en faisant de l'exploitation.



Mais attention à ne pas faire trop d'exploitation car il se peut que l'on passe à côté d'une plus grande récompense.

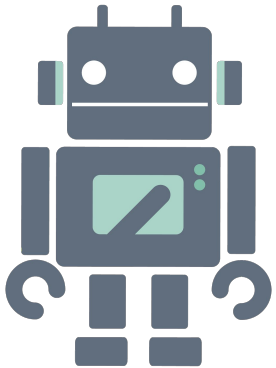


Il faudra aussi faire en sorte d'éviter les récompenses négatives



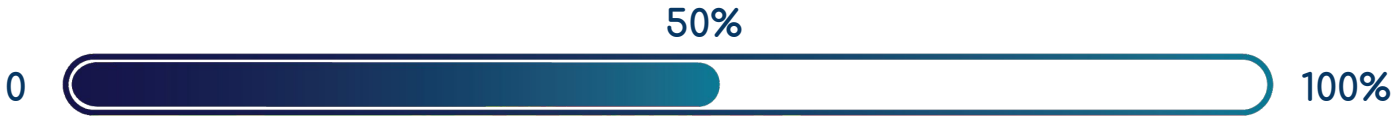
Pour résumer, au début il faudra faire beaucoup d'exploration pour que le robot puisse découvrir la majorité des états et récompenses de l'environnement.

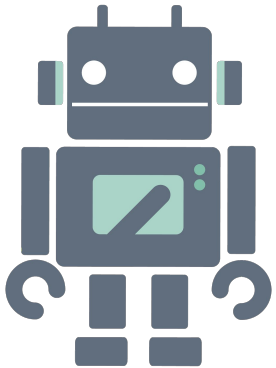
Par la suite, le taux d'exploration est réduit graduellement tandis que le taux d'exploitation est augmenté



Pour résumer, au début il faudra faire beaucoup d'exploration pour que le robot puisse découvrir la majorité des états et récompenses de l'environnement.

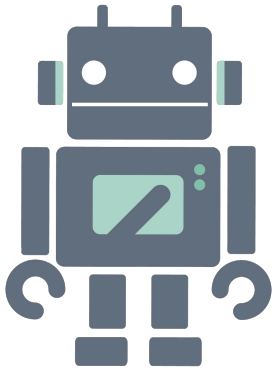
Par la suite, le taux d'exploration est réduit graduellement tandis que le taux d'exploitation est augmenté





Pour résumer, au début il faudra faire beaucoup d'exploration pour que le robot puisse découvrir la majorité des états et récompenses de l'environnement.

Par la suite, le taux d'exploration est réduit graduellement tandis que le taux d'exploitation est augmenté



Value function

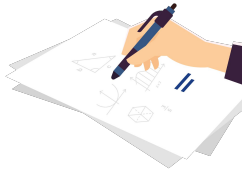
r = 0

10/20

r = +6

16/20

r = -5



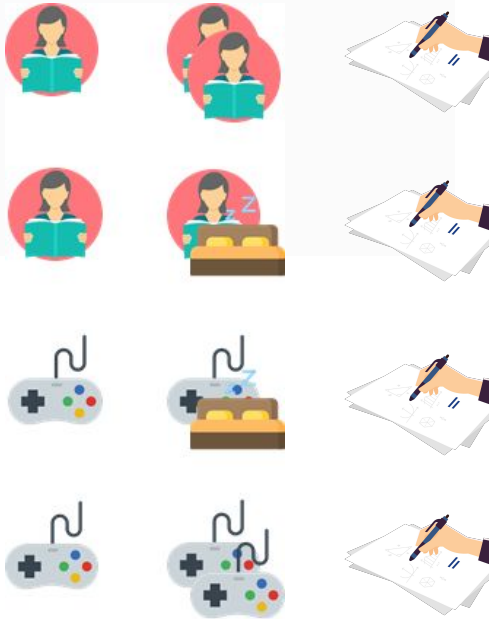
5/20

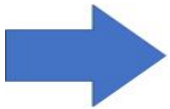
r = -1

5/20

Value function

On va se mettre à la place d'un agent qui est nouveau dans l'environnement et qui ne sait pas quelles actions prendre. Il va donc explorer l'environnement pour aboutir au schéma suivant :





(s :  , s' :  , r=0)

(s :  , s' :  , r=0)

(s :  , s' :  , r=0)

(s :  , s' :  , r=-1)

(s :  , s' :  , r=0)

(s :  , s' :  , r=0)

(s :  , s' :  , r=+6)

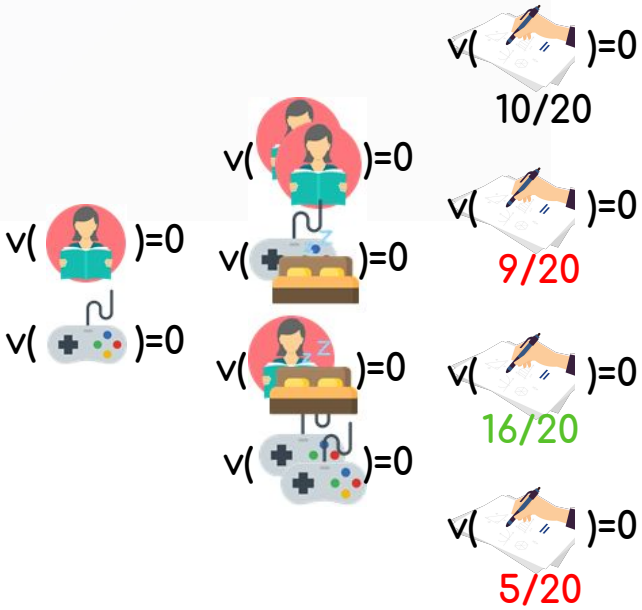
(s :  , s' :  , r=-5)

V(s) = V(s) + learning_rate * (V(s') – V(s))

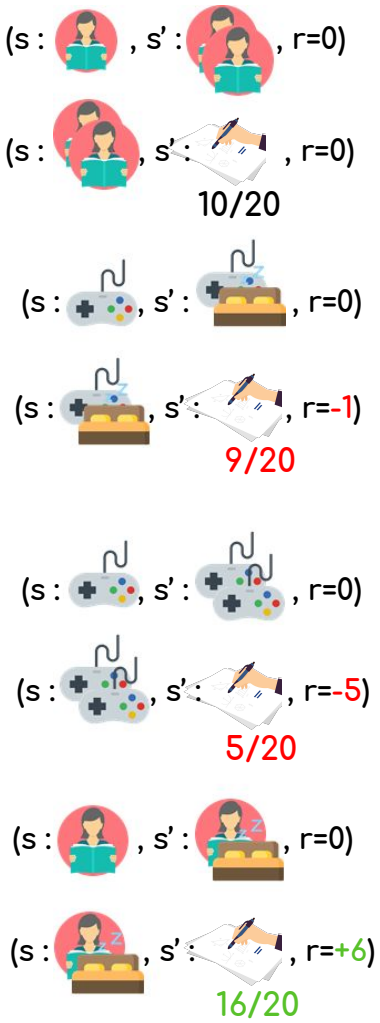
$$V(s) = V(s) + \alpha(V(s') - V(s))$$

Value function

$$V(s) = V(s) + \alpha(V(s') - V(s))$$



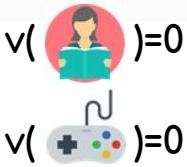
Backtracking

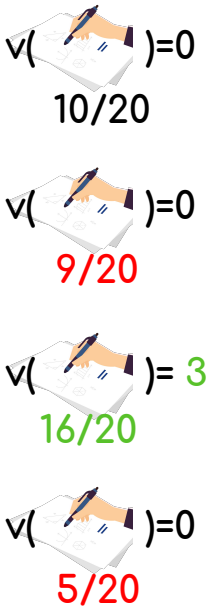


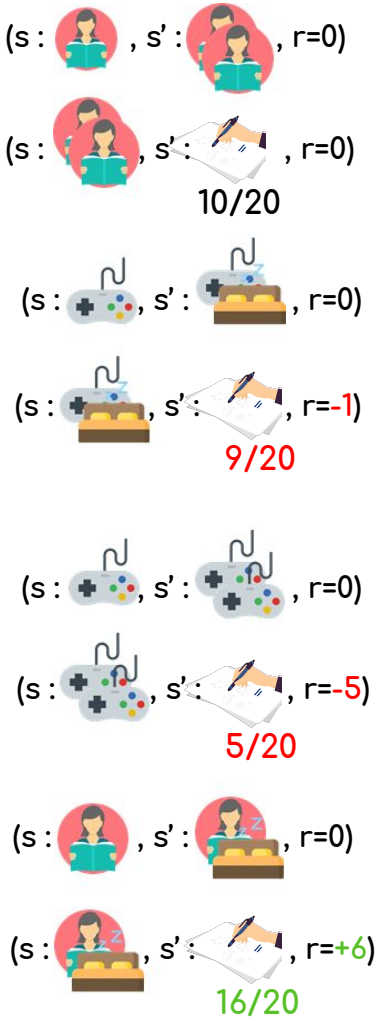
Value function

Learning_rate = 0.5

$V(s) = V(s) + 0.5 * (V(s') - V(s))$







$0 + 0.5 * (6 - 0) = 3$

Value function

Learning_rate = 0.5

$V(s) = V(s) + 0.5 * (V(s') - V(s))$

v( )=0

v( )=0

v( )=0

v( )=0

v( )=0

v( )=0

10/20

v( )= -0.5

9/20

v( )= 3

16/20

v( )= -2.5

5/20

(s : , s' : , r=0)

(s : , s' : , r=0)

10/20

(s : , s' : , r=0)

(s : , s' : , r=-1)

9/20

(s : , s' : , r=0)

(s : , s' : , r=-5)

5/20

(s : , s' : , r=0)

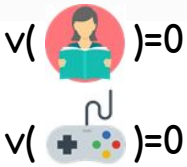
(s : , s' : , r=+6)

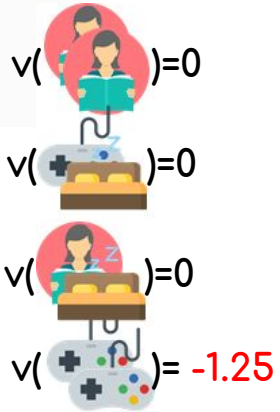
16/20

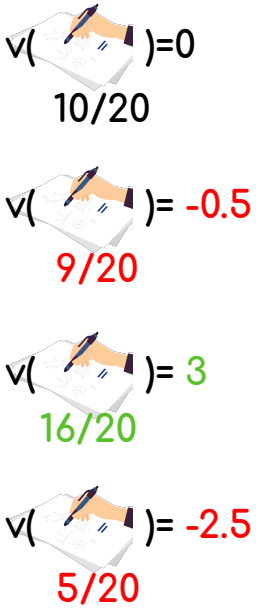
Value function

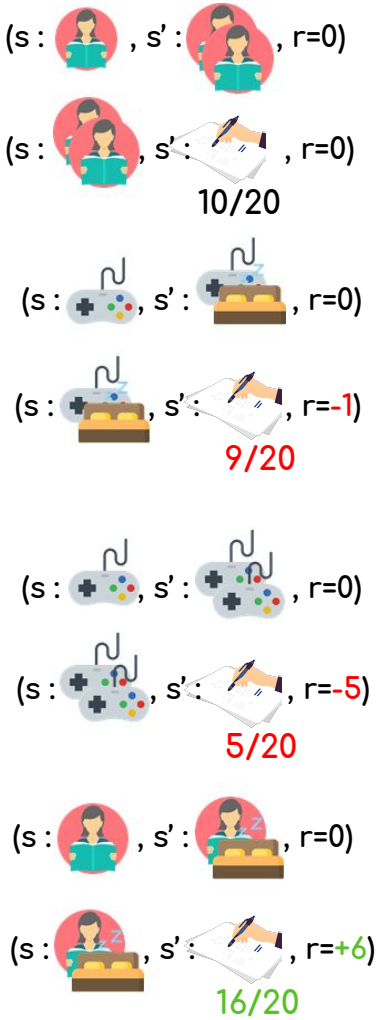
Learning_rate = 0.5

$V(s) = V(s) + 0.5 * (V(s') - V(s))$









$0 + 0.5 * (-2.5 - 0) = -1.25$

Value function

Learning_rate = 0.5

$V(s) = V(s) + 0.5 * (V(s') - V(s))$

v()=0

v()=0

v()=0

v()= -0.25

v()= 1.5

v()= -1.25

v()=0

10/20

v()= -0.5

9/20

v()= 3

16/20

v()= -2.5

5/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=0)

10/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=-1)

9/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=-5)

5/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=+6)

16/20

Value function

Learning_rate = 0.5

$V(s) = V(s) + 0.5 * (V(s') - V(s))$

v()=0

v()= -0.625

v()=0

v()= -0.25

v()= 1.5

v()= -1.25

v()=0

10/20

v()= -0.5

9/20

v()= 3

16/20

v()= -2.5

5/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=0)

10/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=-1)

9/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=-5)

5/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=+6)

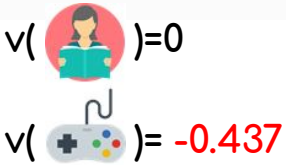
16/20

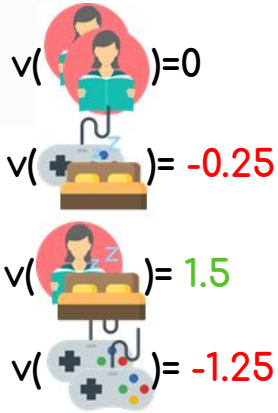
0 + 0.5 * (-1.25 – 0) = -0.625

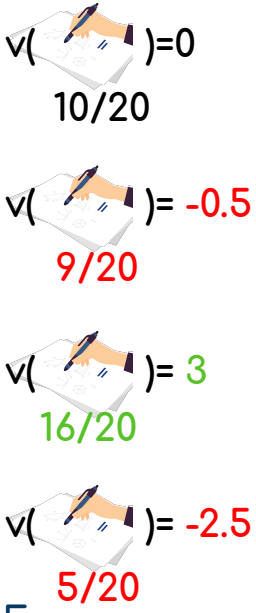
Value function

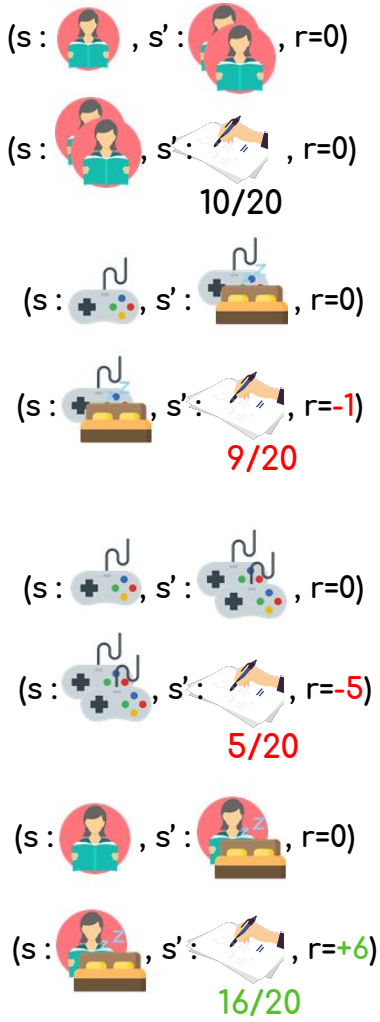
Learning_rate = 0.5

$V(s) = V(s) + 0.5 * (V(s') - V(s))$









-0.625 + 0.5 * (-0.25 – (-0.625)) = -0.4375

Value function

Learning_rate = 0.5

$V(s) = V(s) + 0.5 * (V(s') - V(s))$

v()= 0.75

v()= -0.437

v()=0

v()= -0.25

v()= 1.5

v()= -1.25

v()=0

10/20

v()= -0.5

9/20

v()= 3

16/20

v()= -2.5

5/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=0)

10/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=-1)

9/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=-5)

5/20

(s :  , s' :  , r=0)

(s :  , s' :  , r=+6)

16/20

Value function

V(s) = 0.75

V(s) = -0.43

V(s) = 0

V(s) = 1.5

V(s) = -1.25

V(s) = -0.25

r = 0

r = +6

r = -5

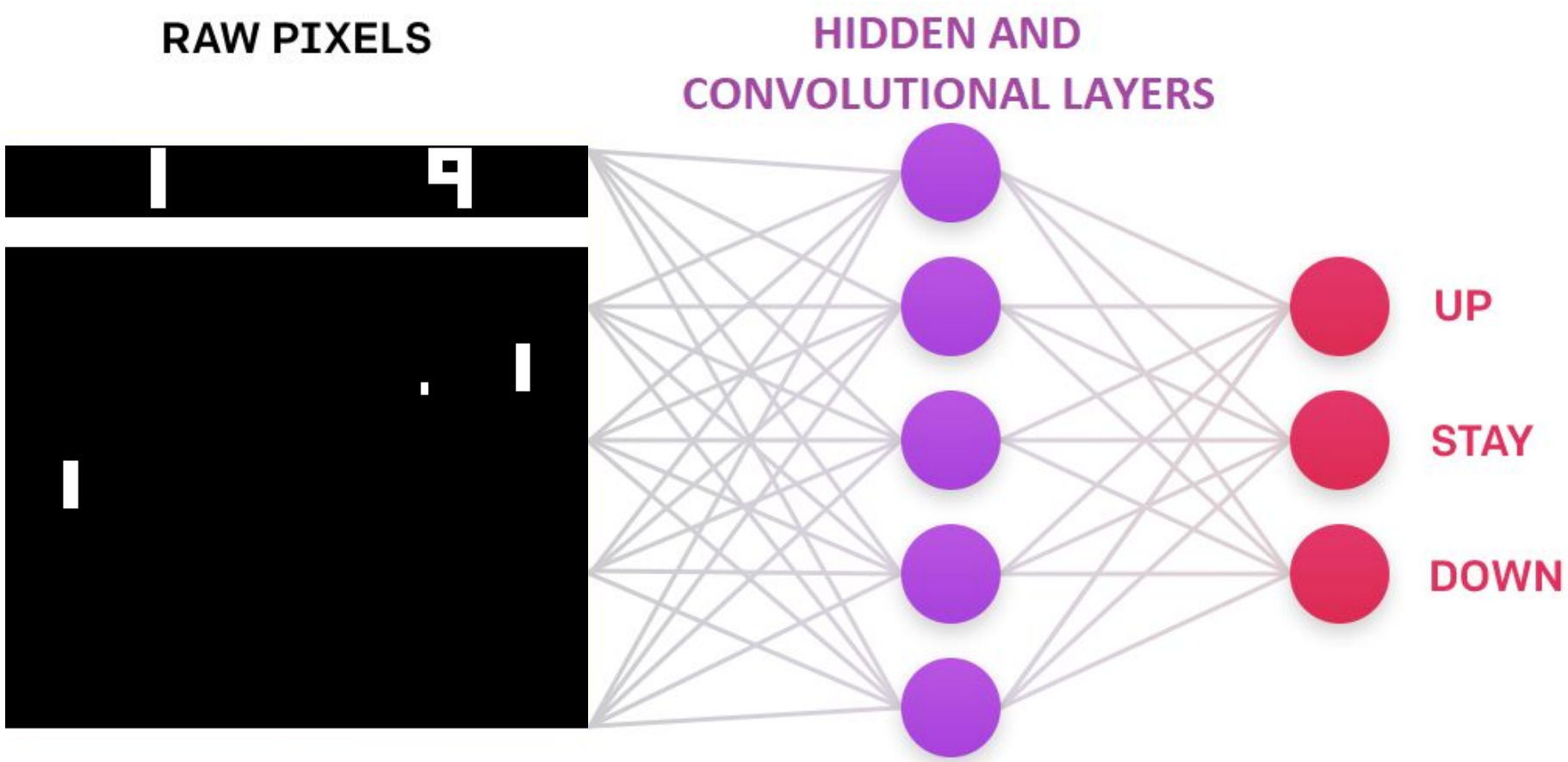
r = -1

10/20

16/20

5/20

5/20


RAW PIXELS
HIDDEN AND
CONVOLUTIONAL LAYERS
UP
STAY
DOWN

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \qquad 0 < \gamma < 1$$

$$\mathcal{P}^a_{ss'} = Pr(s_{t+1} = s' | s_t = s, a_t = a) \qquad \mathcal{R}^a_{ss'} = \mathbb{E}[r_{t+1} | s_t = s, s_{t+1} = s', a_t = a]$$

$$V^\pi(s) = \mathbb{E}_\pi \left[R_t | s_t = s \right]$$

$$V^\pi(s) = \sum_a \pi(s,a) \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s') \right]$$

$$Q^\pi(s,a) = \mathbb{E}_\pi \left[R_t | s_t = s, a_t = a \right] \qquad Q^\pi(s,a) = \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma \sum_{a'} \pi(s',a') Q^\pi(s',a') \right]$$

$$NewQ(s,a) = Q(s,a) + \alpha[R(s,a) + \gamma max\, Q'(s',a') - Q(s,a)]$$

New Q value for that state and that action

Current Q value

Learning Rate

Reward for taking that action at that state

Discount rate

Maximum expected future reward **given the new s' and all possible actions at that new state**

Don’t worry about it if you don’t understand

– *Andrew Ng*