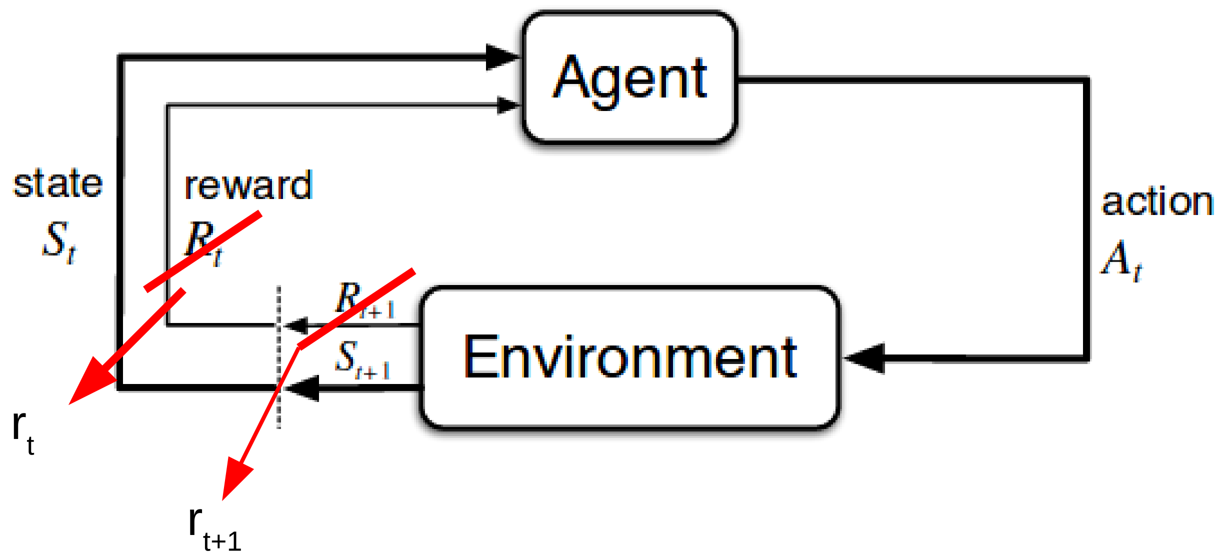
Base class

Classe mère : Agent : contient un code de base + commentaires + structure

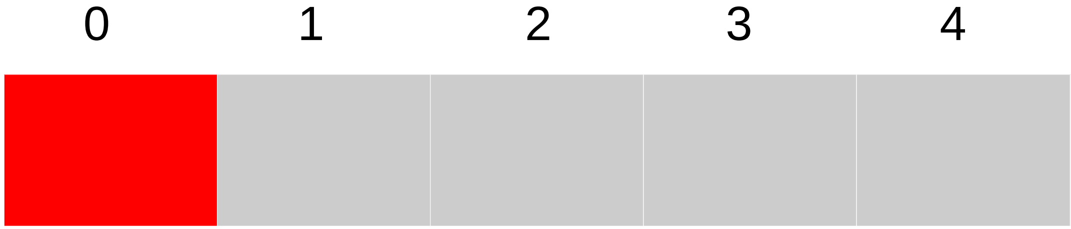
Classes filles : AgentRandom + Agents que vous allez implémenter


Agent
state
S_t
reward
R_t
action
A_t
R_{t+1}
S_{t+1}
Environment
r_t
r_{t+1}

Q learning Monte Carlo

Step by step

Initialization



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

$\pi(s)$ = random(a) with prob Ɛ
$\pi(s) = \text{argmax}_a\ Q(s,a)$ with prob 1- Ɛ

s \ a	left	right
0		
1	[]	[]
2	[]	[]
3	[]	[]
4	[]	[]

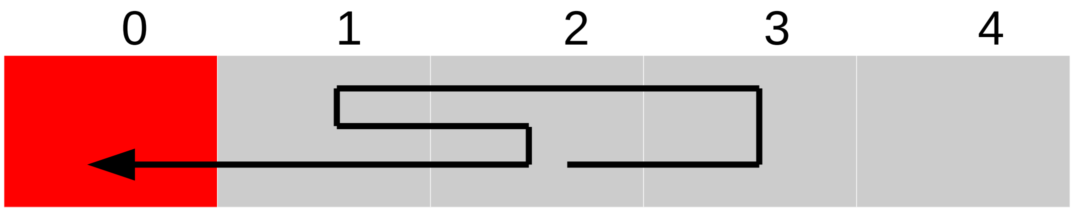
Return table

s \ a	left	right
0		
1	0	0
2	0	0
3	0	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

1st iteration : performs episode



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

$\pi(s)$ = random(a) with prob ε
$\pi(s)$ = argmax_a Q(s,a) with prob 1- ε

2 → 3 : r = -1
3 → 2 : r = -1
2 → 1 : r = -1
1 → 2 : r = -1
2 → 1 : r = -1
1 → 0 : r = 10

R = 5

Note : $\gamma = 1$

s \ a	left	right
0		
1	[]	[]
2	[]	[]
3	[]	[]
4	[]	[]

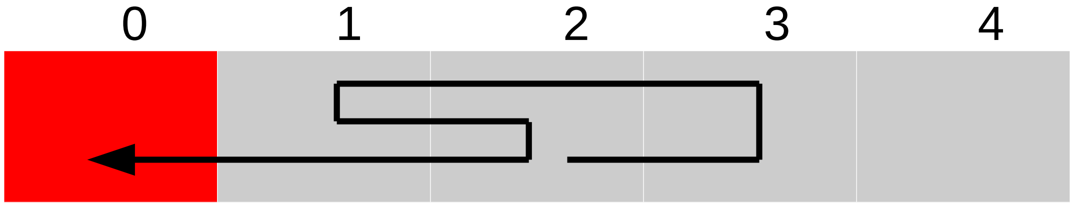
Return table

s \ a	left	right
0		
1	0	0
2	0	0
3	0	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

1st iteration : update tables



Reward function :
-1 every step
10 when in 0

ε-greedy policy :

$\pi(s)$ = random(a) with prob ε
$\pi(s) = \text{argmax}_a\, Q(s,a)$ with prob $1-\varepsilon$

2 → 3 : r = -1
3 → 2 : r = -1
2 → 1 : r = -1
1 → 2 : r = -1
2 → 1 : r = -1
1 → 0 : r = 10

R = 5

Note : $\gamma = 1$

s \ a	left	right
0		
1	[5]	[5]
2	[5]	[5]
3	[5]	[]
4	[]	[]

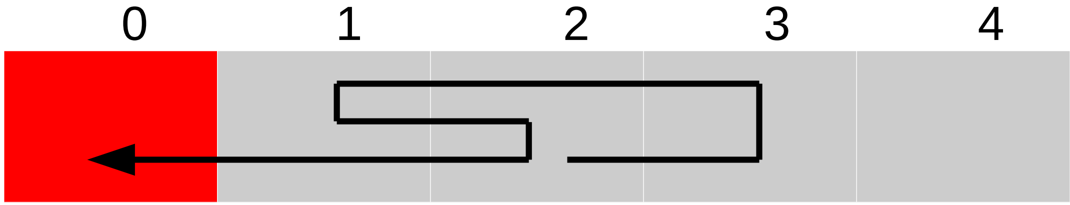
Return table

s \ a	left	right
0		
1	0	0
2	0	0
3	0	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

1^{st} iteration : update tables



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

$\pi(s)$ = random(a) with prob Ɛ
$\pi(s) = \text{argmax}_a\, Q(s,a)$ with prob 1- Ɛ

2 → 3 : r = -1
3 → 2 : r = -1
2 → 1 : r = -1
1 → 2 : r = -1
2 → 1 : r = -1
1 → 0 : r = 10

R = 5

Note : $\gamma = 1$

s \ a	left	right
0		
1	[5]	[5]
2	[5]	[5]
3	[5]	[]
4	[]	[]

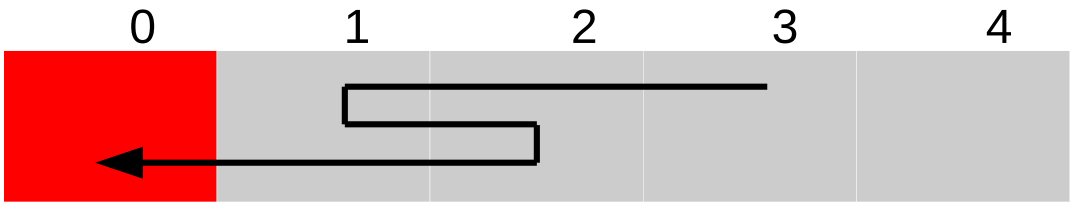
Return table

s \ a	left	right
0		
1	5	5
2	5	5
3	5	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

2nd iteration : perform episode



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

π(s) = random(a) with prob Ɛ
π(s) = argmax_a Q(s,a) with prob 1- Ɛ

3 → 2 : r = -1
2 → 1 : r = -1
1 → 2 : r = -1
2 → 1 : r = -1
1 → 0 : r = 10

R = 6

s \ a	left	right
0		
1	[5]	[5]
2	[5]	[5]
3	[5]	[]
4	[]	[]

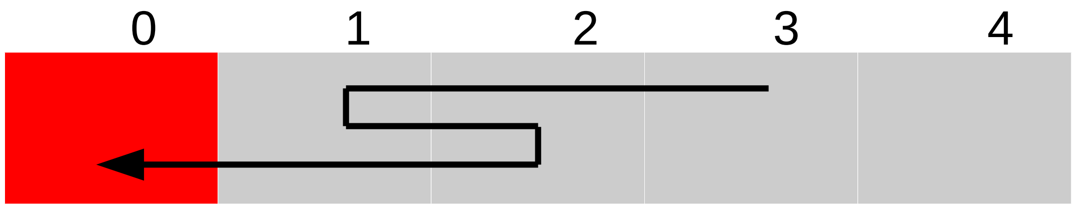
Return table

s \ a	left	right
0		
1	5	5
2	5	5
3	5	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

2nd iteration : update tables



Reward function :

-1 every step
10 when in 0

Ɛ-greedy policy :

$\pi(s)$ = random(a) with prob Ɛ
$\pi(s) = \mathrm{argmax}_a\, Q(s,a)$ with prob 1- Ɛ

3 → 2 : r = -1
2 → 1 : r = -1
1 → 2 : r = -1
2 → 1 : r = -1
1 → 0 : r = 10

R = 6

s \ a	left	right
0		
1	[5,6]	[5,6]
2	[5,6]	[5]
3	[5,6]	[]
4	[]	[]

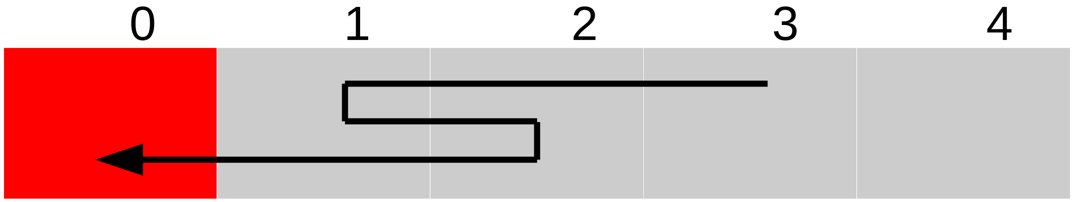
Return table

s \ a	left	right
0		
1	5	5
2	5	5
3	5	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

2^{nd} iteration : update episode



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

π(s) = random(a) with prob Ɛ
π(s) = argmax_a Q(s,a) with prob 1- Ɛ

3 → 2 : r = -1
2 → 1 : r = -1
1 → 2 : r = -1
2 → 1 : r = -1
1 → 0 : r = 10

R = 6

s \ a	left	right
0		
1	[5,6]	[5,6]
2	[5,6]	[5]
3	[5,6]	[]
4	[]	[]

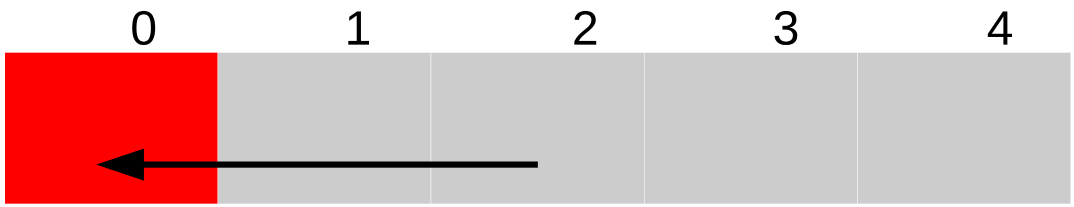
Return table

s \ a	left	right
0		
1	5.5	5.5
2	5.5	5
3	5.5	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

3rd iteration : perform episode



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

$\pi(s)$ = random(a) with prob Ɛ
$\pi(s)$ = argmax_a Q(s,a) with prob 1- Ɛ

2 → 1 : r = -1
1 → 0 : r = 10

R = 9

s \ a	left	right
0		
1	[5,6]	[5,6]
2	[5,6]	[5]
3	[5,6]	[]
4	[]	[]

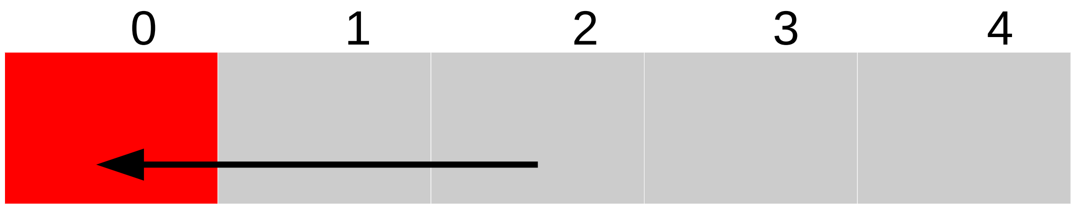
Return table

s \ a	left	right
0		
1	5.5	5.5
2	5.5	5
3	5.5	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

3rd iteration : update tables



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

$\pi(s)$ = random(a) with prob Ɛ
$\pi(s)$ = argmax$_a$ Q(s,a) with prob 1- Ɛ

2 → 1 : r = -1
1 → 0 : r = 10

R = 9

s \ a	left	right
0		
1	[5,6,9]	[5,6]
2	[5,6,9]	[5]
3	[5,6]	[]
4	[]	[]

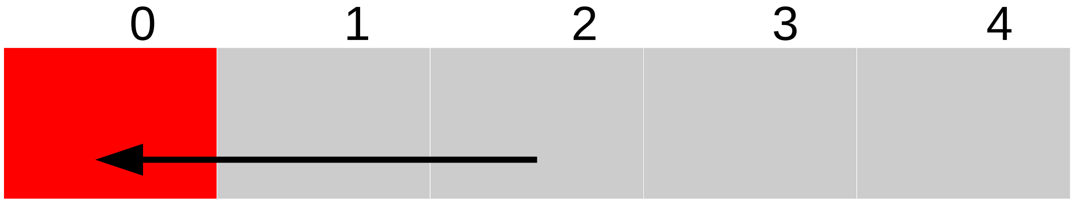
Return table

s \ a	left	right
0		
1	5.5	5.5
2	5.5	5
3	5.5	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

3^{rd} iteration : update tables



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

π(s) = random(a) with prob Ɛ
π(s) = argmax_a Q(s,a) with prob 1- Ɛ

2 → 1 : r = -1
1 → 0 : r = 10

R = 9

s \ a	left	right
0		
1	[5,6,9]	[5,6]
2	[5,6,9]	[5]
3	[5,6]	[]
4	[]	[]

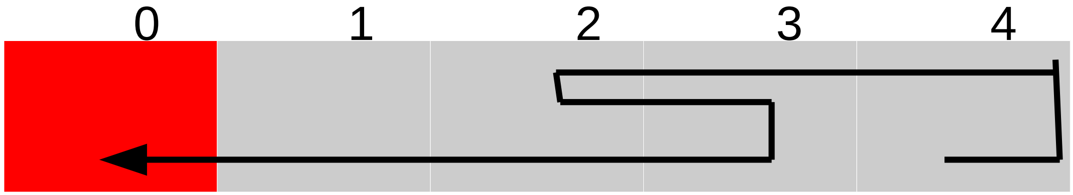
Return table

s \ a	left	right
0		
1	6.6	5.5
2	6.6	5
3	5.5	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

4th iteration : perform episode



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

$\pi(s) = \mathrm{random}(a)$ with prob ε
$\pi(s) = \mathrm{argmax}_a\, Q(s,a)$ with prob $1- \varepsilon$

4 → 4 : r = -1
4 → 3 : r = -1
3 → 2 : r = -1
2 → 3 : r = -1
3 → 2 : r = -1
2 → 1 : r = -1
1 → 0 : r = 10

R = 4

s \ a	left	right
0		
1	[5,6,9]	[5,6]
2	[5,6,9]	[5]
3	[5,6]	[]
4	[]	[]

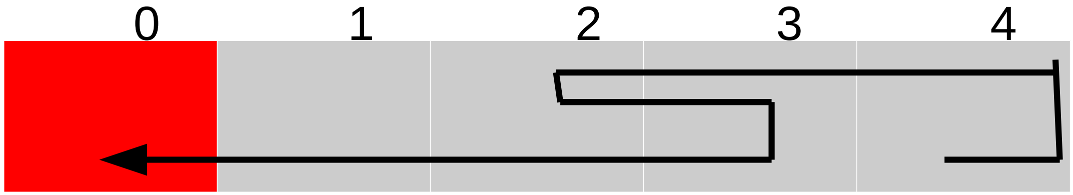
Return table

s \ a	left	right
0		
1	6.6	5.5
2	6.6	5
3	5.5	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

4th iteration : update tables



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

$\pi(s)$ = random(a) with prob Ɛ
$\pi(s) = \text{argmax}_a\ Q(s,a)$ with prob 1- Ɛ

4 → 4 : r = -1
4 → 3 : r = -1
3 → 2 : r = -1
2 → 3 : r = -1
3 → 2 : r = -1
2 → 1 : r = -1
1 → 0 : r = 10

R = 4

s \ a	left	right
0		
1	[5,6,9,4]	[5,6]
2	[5,6,9,4]	[5,4]
3	[5,6,4]	[]
4	[4]	[4]

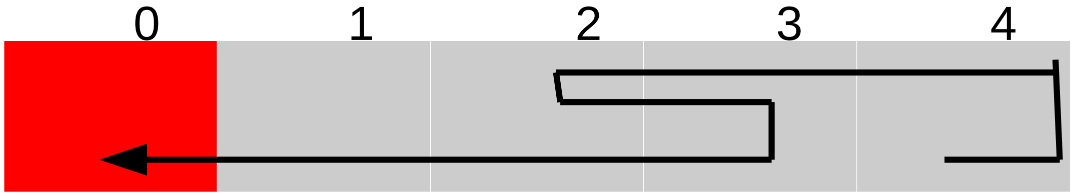
Return table

s \ a	left	right
0		
1	6.6	5.5
2	6.6	5
3	5.5	0
4	0	0

Q table

Q(a,s) = mean R(a,s)

4th iteration : update tables



Reward function :
-1 every step
10 when in 0

Ɛ-greedy policy :

$\pi(s)$ = random(a) with prob Ɛ
$\pi(s) = \text{argmax}_a Q(s,a)$ with prob 1- Ɛ

4 → 4 : r = -1
4 → 3 : r = -1
3 → 2 : r = -1
2 → 3 : r = -1
3 → 2 : r = -1
2 → 1 : r = -1
1 → 0 : r = 10

R = 4

s \ a	left	right
0		
1	[5,6,9,4]	[5,6]
2	[5,6,9,4]	[5,4]
3	[5,6,4]	[]
4	[4]	[4]

Return table

s \ a	left	right
0		
1	6	5.5
2	6	4.5
3	5	0
4	4	4

Q table

Q(a,s) = mean R(a,s)