AI CAMP

school of ai Algiers

Hello and Welcome to Reinforcement Learning Track

by Amir ALMAMMA, Kamel BROUTHEN and Nazim BENDIB

AI CAMP

school of ai Algiers

Track plan:

1. Introduction to Reinforcement learning ← We are here
2. RL Environment Building
3. Solving RL Problems (Value-Based methods)
4. Solving RL Problems (Policy-Based methods)

AI CAMP

school of ai Algiers

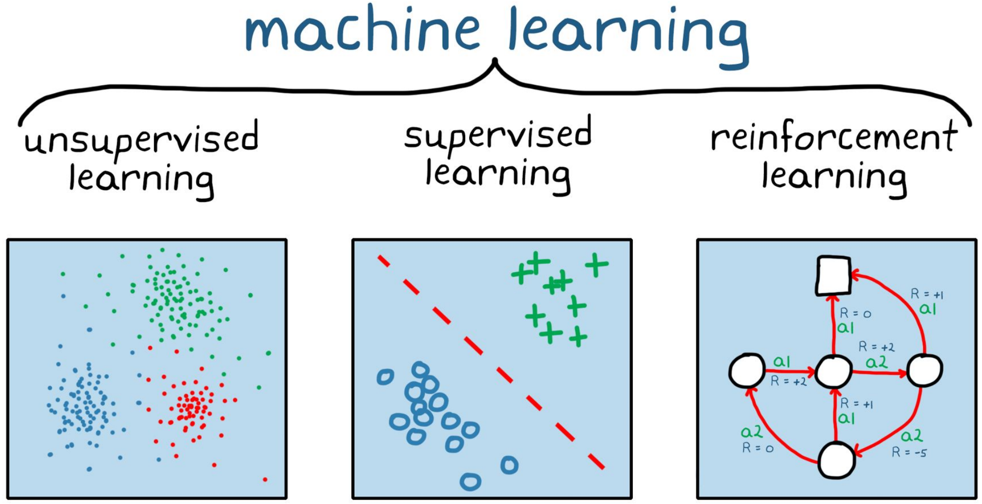
Introduction to Reinforcement learning

Amir ALMAMMA

Workshop Objectives:

- Understanding the motives behind the usage reinforcement learning
- Get an idea about the RL framework
- Learn the basic terminologie
- Get un intuition about RL methods

Intuition



Unlabeled data

Labeled data

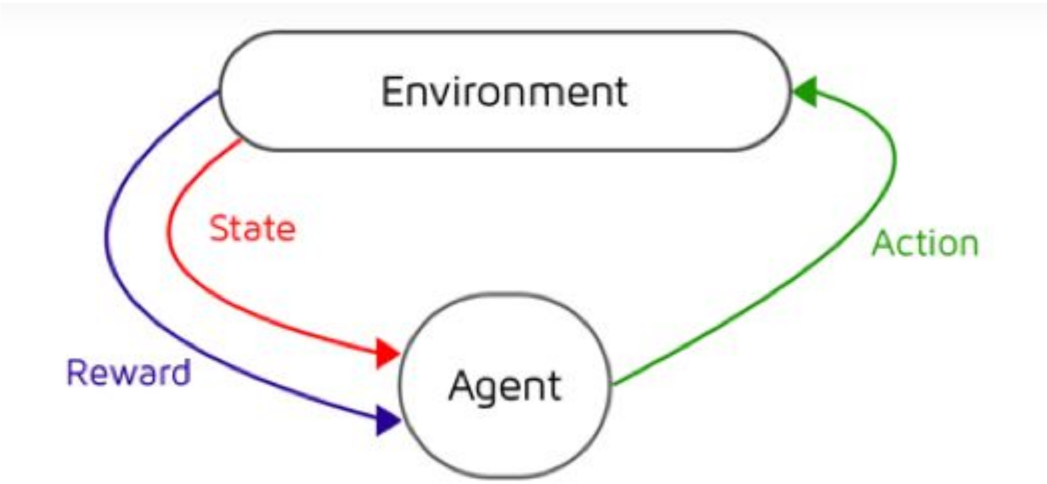
learning the dynamics of a system by interation

AI CAMP

school of ai Algiers

Intuition

- People and animals learn by **interacting with our environment**
- This differs from certain other types of learning
  - It is **active** rather than passive
  - Interactions are often **sequential** — future interactions can depend on earlier ones
- We are **goal-directed**
- We can learn **without examples** of optimal behaviour
- Instead, we optimise some **reward signal**

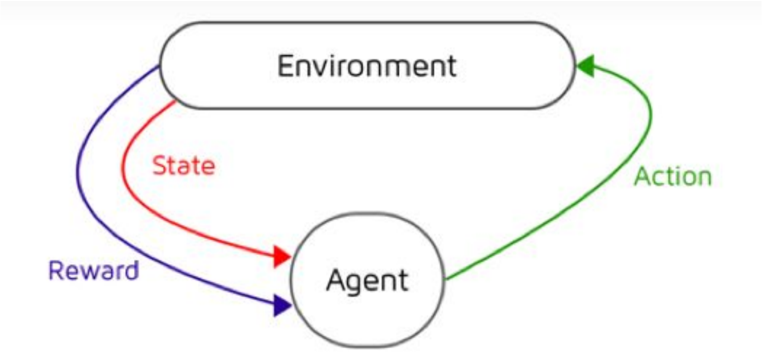


The reinforcement Learning framework

Definition: Reinforcement Learning (RL) is a machine learning approach for an agent to learn making decisions by interacting with an environment.

Objective: The goal of an RL agent is to learn a policy which maximizes the cumulative reward over time.

Reward Hypothese: all goals can be described as the maximization of the expected return (expected cumulative reward).



The reinforcement Learning framework

Example #1

agent: baby

environment: box game

state: the position of the baby and the pieces next to him

action: moving a piece

observation: what the baby can see and touch

reward: clapping when he gets a piece right

Example #2: maze

agent: a robot

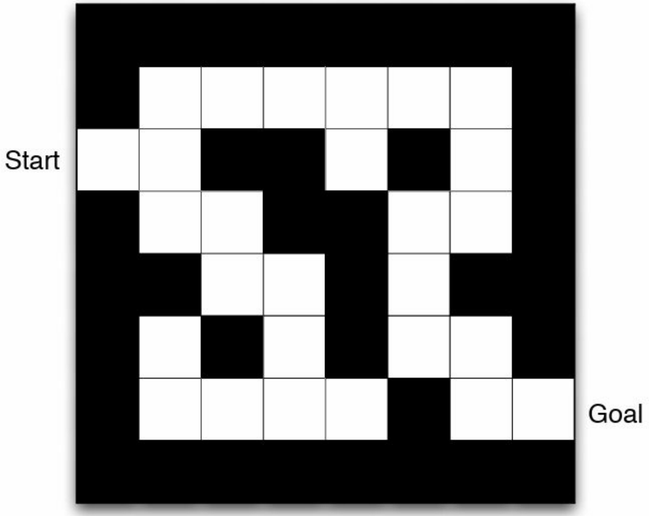
environment: the maze

state: the position of the robot

action: moving up, down, right or left

observation: the neighbourhood of the robot

reward: 0 if goal achieved else -1



Exploration VS Exploitation trade-off

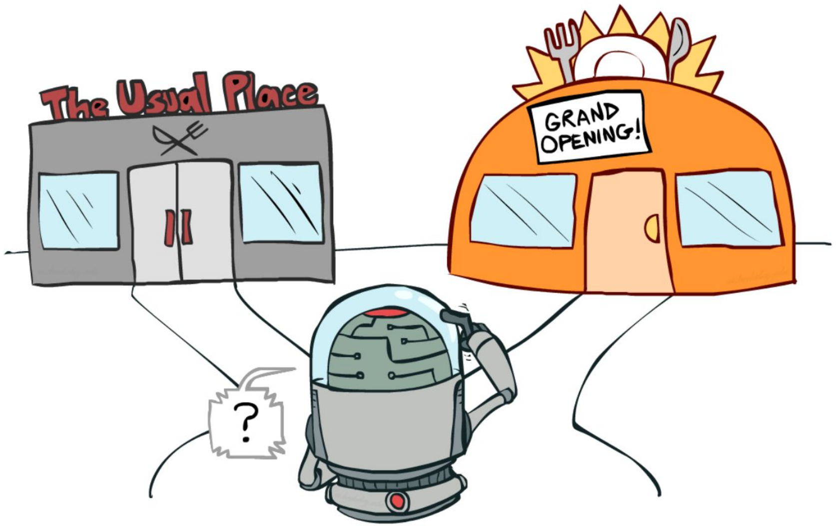
Explained Intuitively

AI CAMP

school of ai Algiers

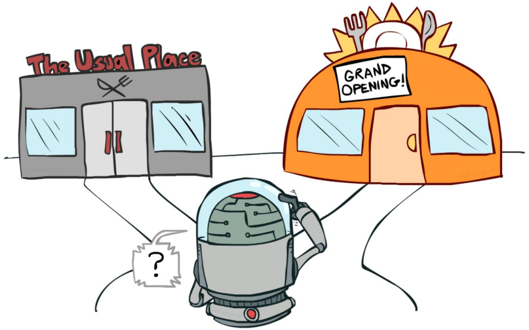
Exploitation VS Exploration

the value of the usual place is well known, but if we keep sampling from it we may never know how bad/good the new place is.



the value of the new place is unknown, if we choose it we are not using what we learned, so our action can be bad, but our policy might get better because we know more about the environment.

Exploitation VS Exploration

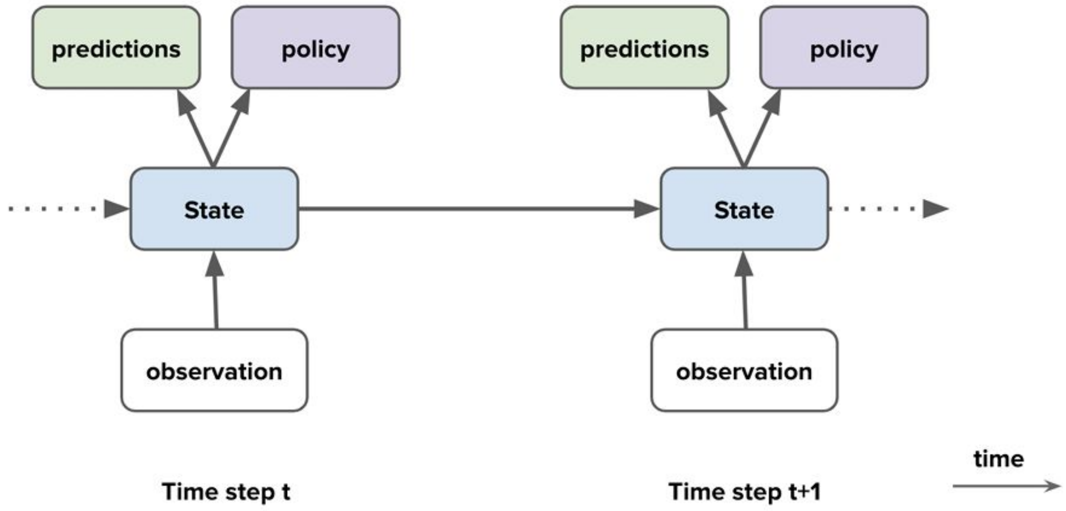


- Do we start by exploring, and exploit more as we get more experience?
- How do we know if we accumulated enough experience to start exploiting only?
- What we fall to a local optimum?
- In the literature, we have ideas, and ways to implement them, like e-greedy, e-decay, UCB..

how does the agent learn?

- Agent State
- Policy (how to act)
- Value function (how good are the next options)
- Model (what do we understand about the environment dynamics)

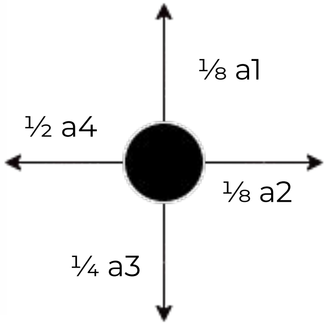
1- State:



the agent changes state after each action, moving in the action space, it's policy is a function of state

2- Policy:

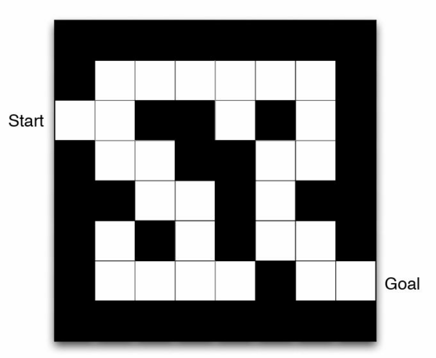
- A policy defines the agent's behaviour
- It is a map from agent state to action
- Deterministic policy: $A = \pi(S)$
- Stochastic policy: $\pi(A|S) = p(A|S)$



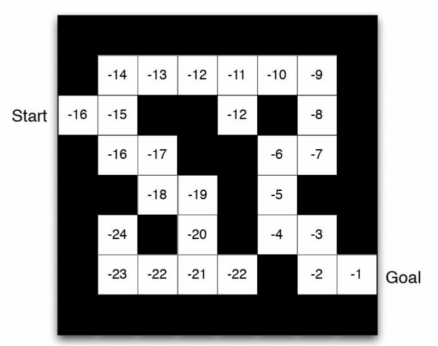
stochastic policy

3- Value function:

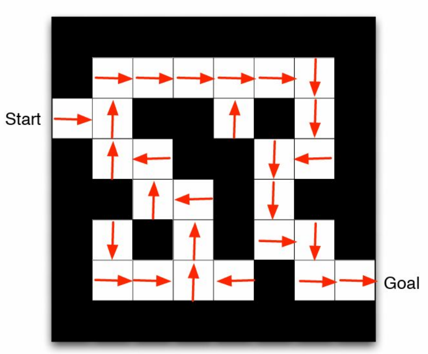
(state)-> estimation of the return that we get after being on this state



initial naive value function (all 0)



value function after learning



greedy policy on the value function (choosing to move to the state that has maximum value)

4- Model:

- A **model** predicts what the environment will do next
- E.g., $\mathcal{P}$ predicts the next state

$$\mathcal{P}(s, a, s') \approx p(S_{t+1} = s' \mid S_t = s, A_t = a)$$

- E.g., $\mathcal{R}$ predicts the next (immediate) reward

$$\mathcal{R}(s, a) \approx \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$$

- A model does not immediately give us a good policy - we would still need to plan
- We could also consider **stochastic** (**generative**) models

MDPs and POMDPs

Markov desicion processes and
partially observable markov decision processes

AI CAMP

school of ai
Algiers

Markov decision processes (MDPs) are a useful mathematical framework

Definition

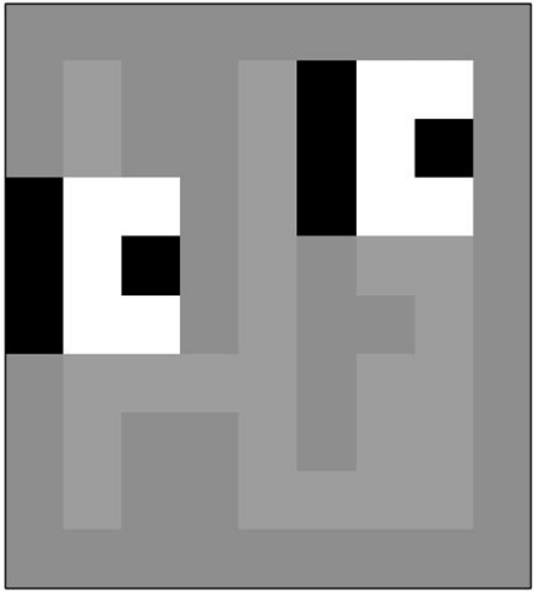
A decision process is Markov if

$$p(r, s \mid S_t, A_t) = p(r, s \mid \mathcal{H}_t, A_t)$$

- This means that the state contains all we need to know from the history
- Doesn't mean it contains everything, just that adding more history doesn't help
- $\Longrightarrow$ Once the state is known, the history may be thrown away
  - The full environment + agent state is Markov (but large)
  - The full history $\mathcal{H}_t$ is Markov (but keeps growing)
- Typically, the agent state S_t is some compression of $\mathcal{H}_t$
- Note: we use S_t to denote the **agent state**, not the **environment state**

POMDPs:

- **Partial observability**: The observations are not Markovian
  - A robot with camera vision isn't told its absolute location
  - A poker playing agent only observes public cards
- Now using the observation as state would not be Markovian
- This is called a **partially observable Markov decision process** (POMDP)
- The **environment state** can still be Markov, but the agent does not know it
- We might still be able to construct a Markov agent state



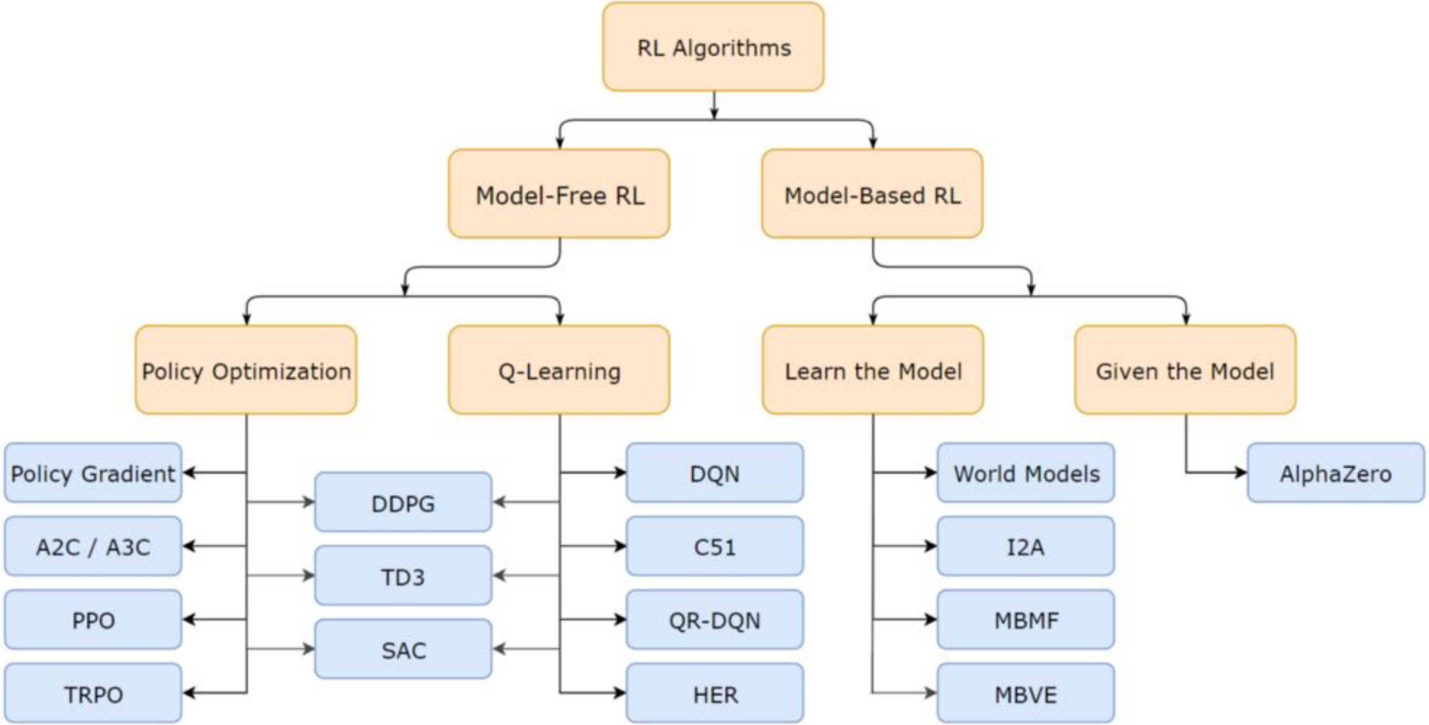
these 2 states are not markovian because the do not capture the full information to make a decision, no matter how good the policy gets, it will not learn to make the best decision in each state

RL methods and types of agents

AI CAMP

school of ai Algiers

RL methods



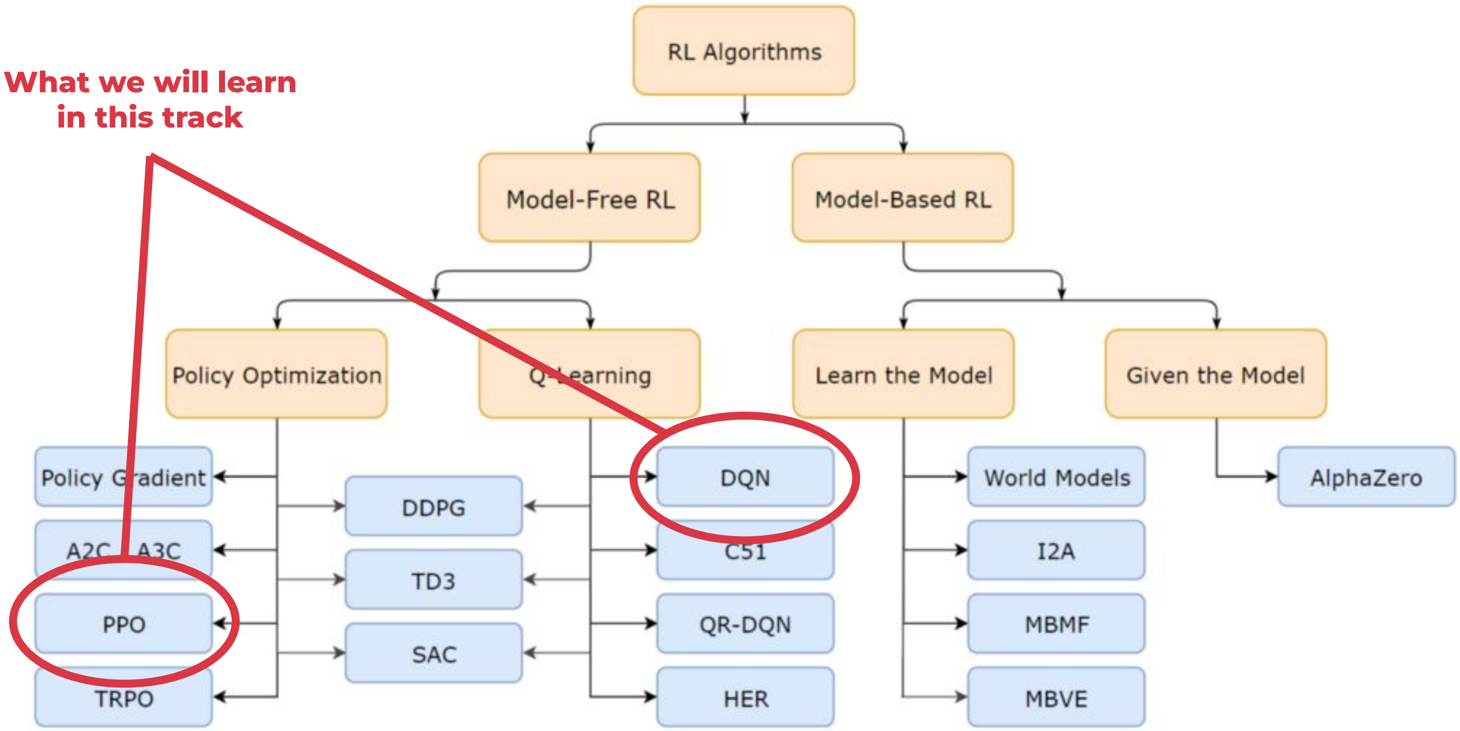
stochastic policy

AI CAMP

school of ai Algiers

RL methods

What we will learn in this track



AI CAMP

school of ai Algiers

AI CAMP

school of ai Algiers

Thank you for attending

any questions ?