Introduction to Reinforcement Learning

Dr. Mayank S JHA

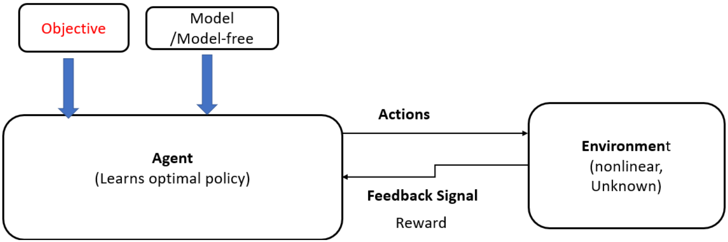
Maitre de Conférences (Associate Professor),
CRAN,
UMR 7039, CNRS,
Polytech Nancy,
Office: C 225,
Université de Lorraine
France.

Table of Contents

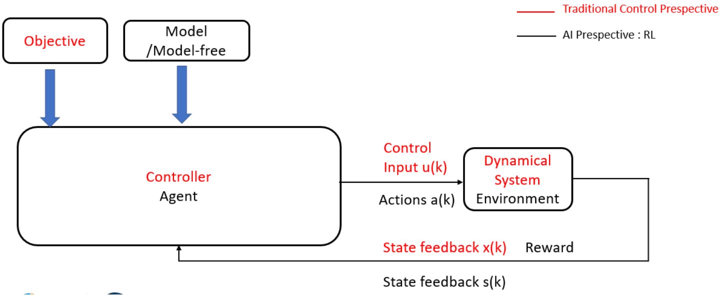
Table of Contents

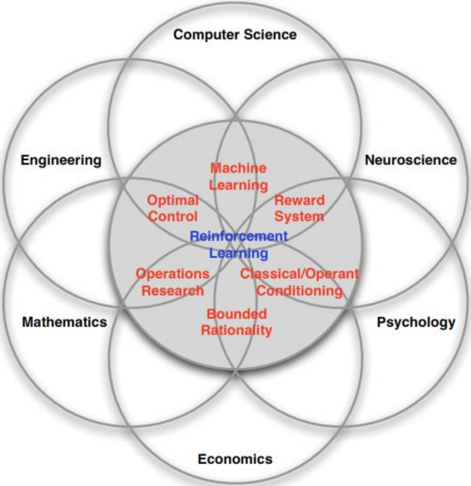
Dr. Mayank S JHA, mayank-shekhar.jha@univ-lorraine.fr
Polytech Nancy, CRAN, University of Lorraine, France

Reinforcement Learning Architecture



Reinforcement Learning: Automatic Control




Computer Science
Engineering
Neuroscience
Machine Learning
Optimal Control
Reward System
Reinforcement Learning
Operations Research
Classical/Operant Conditioning
Bounded Rationality
Mathematics
Psychology
Economics

Motivation

Reinforcement Learning: Towards human level :

control ((Finding the optimal way of doing a given task)

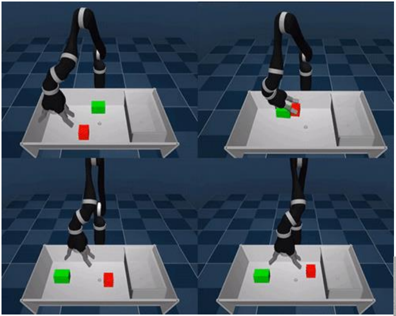
prediction

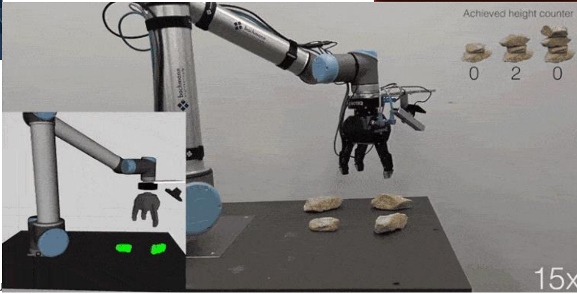
Adaptation (Robots That Can Adapt like Animals, *Nature*)

Built new moves

Email: *mayank-shekhar.jha [at] univ-lorraine.fr*

Some Applications





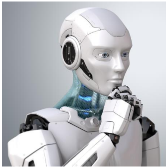
Source : Deep Mind

POLYTECH NANCY
UNIVERSITÉ DE LORRAINE
CRAN

Agent

olervation

O_t



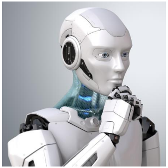
Agent/controller

Action

A_t

Reward R_t

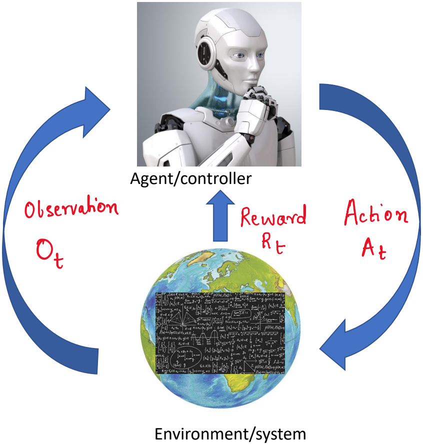
Agent and Environment



Agent/controller

Environment/system

Agent and Environment and Interaction



At each step t:

Agent:

- executes action A_t
- Receives observation O_t
- Receives reward R_t

Environment:

- Receives action A_t
- Emits observation O_{t+1}
- Emits scalar reward R_{t+1}

$t \leftarrow t+1$

Rewards

Reward R_t : scalar feedback signal

: indicates how well an agent does

: Agent maximises the cumulated reward.

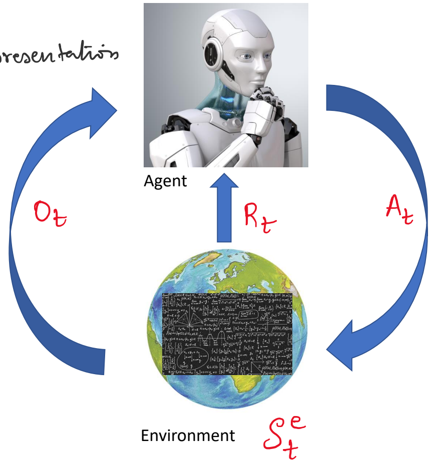
Reinforcement Learning: find the best way (optimal way) to maximize the cumulative reward with respect to a given goal.

Environment State

S_t^e → environment's privet representation

→ Whatever knowledge env uses to emit next observation or reward

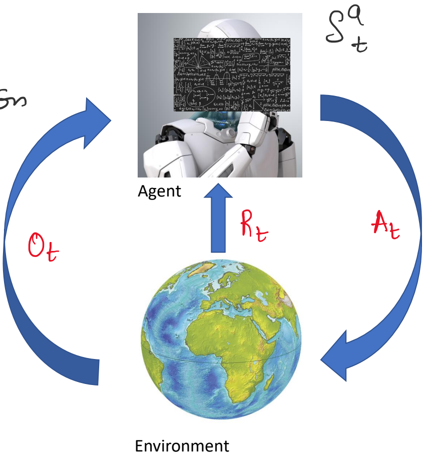
→ state may not be visible



Agent State S_t^a

- $S_t^a \rightarrow$ agent's internal representation
- the 'essential' information to pick next action
- it can also be function of history.

$$S_t^a = f(H_t)$$



Information State

State $S_t \rightarrow$ Markov Property

$$P(S_{t+1} \mid S_t) = P(S_{t+1} \mid S_1, S_2, \dots S_t)$$

- The future is independant of the past given the present.

$$H_{1:t} \longrightarrow S_t \longrightarrow H_{t+1:\infty}$$

- History is known, throw away the history!
- State is fully capable of describing the 'present' situation of the system (agent or Environment)
- S_t^e is Markov, H_t is Markov!

Markov Decision Process:

MDP : 4 tuple (S, A, P_a, R_a)

$S \rightarrow$ set of states

$A \rightarrow$ set of actions

$P_a(s, s') = Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$

$R_a(s, s')$ = immediate reward

Diagram

All processes in this course $\rightarrow$ MDP!!

Environments.

Fully observable (in this course)

$$O_t = s_t^a = s_t^e.$$

When environments is not fully observable,

we have Partially observable Markov Process (POMDPs)

Ex: Robot using camera → position (is not absolute)

Trading agent → observes only prices.

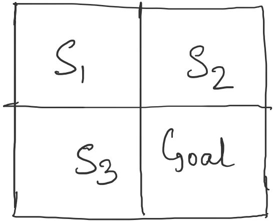
(trends not seen)

Agent:

Agent :

- Policy
- Value Function
- Model.

Grid World Example



What is the environment?

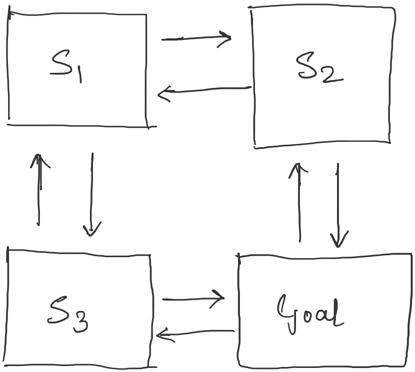
What are the states?

What is agent's role?

A small Grid world $\rightarrow$ Environment

An agent lives in it $\rightarrow$ Agent

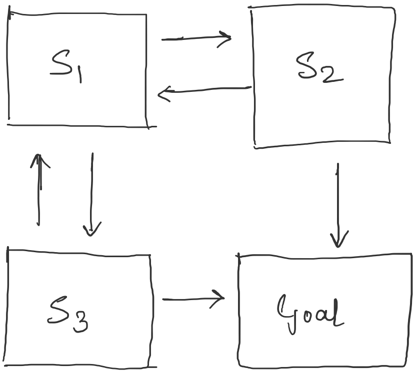
Grid World Example



Agent moves.

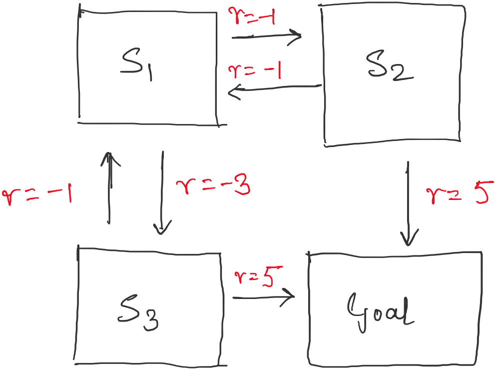
But once reaches goal state, the objective is accomplished!!

Grid World Example



Once, Goal is reached,
No turning back!!

Grid World Example



Rewards are assigned.

Agent knows the MDP

Environment

Rewards

What is the best way to reach goal?

Optimal path?

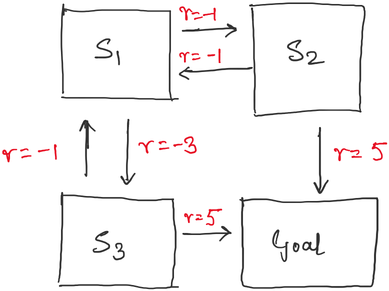
Agent Policy

- Policy → agent's behaviour (dynamics)
- Mathematically, mapping from state to action.

Deterministic Policy : $a = \pi(s)$

Stocastic Policy $\pi(a|s) = P(A_t = a \mid S_t = s)$

Grid World Example



Policy Example. π

Actions are drawn from a uniform distribution.

$$\pi(\text{right} \mid s_1) = 1/2$$
$$\pi(\text{down} \mid s_1) = 1/2$$
$$\pi(\text{left} \mid s_2) = 1/2$$
$$\pi(\text{down} \mid s_2) = 1/2$$
$$\pi(\text{up} \mid s_2) = 1/2$$
$$\pi(\text{right} \mid s_2) = 1/2$$

Actions are based on policy.

Policy depends on some probability

Agent

Value Function.

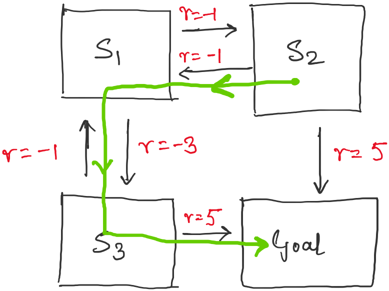
- Assessment of the value of being in a state.
- Prediction of feuture reward.
- Determines the quality of being in that state.
- Helps to make a choice in taking actions while being in a state.

$$V_\pi(s) = \mathbb{E}_\pi\left[\underbrace{R_{t+1} + R_{t+2} + R_{t+3} + \cdots}_{\text{sum of all future rewards.}} \mid S_t = s\right]$$

Value of being in 's' with respect to policy π

Mean over policy π

Grid World Example



$$V_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + R_{t+2} + R_{t+3} + \dots \mid S_t = s\right]$$

Assume : A movement from S2 as:

$$V_\pi(S_{2,t}) = \mathbb{E}_\pi\left(-1 + -3 + 5 \mid S_t = S_2\right)$$

$$= (-1 + -3 + 5) = 1$$

we consider only one π

if, $\pi = \{\pi_1, \pi_2, \pi_3 \dots\}$

then average over all π.

Agent :

Value Function

$$V_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots \mid S_t = s \right]$$

Discount factor.

Determines how important is any future reward wrt the present reward.

(if $\gamma = 0.1$;
then $\gamma^{10} = (0.1)^{10}$

$\gamma^{10} << \gamma$

so $\gamma^{10} \cdot R_{t+11}$
is very less important !!

Model

- Model predicts what environment does next.

  Ex: If you have model of robot → predict next step.

  But exact models are rarely available.

  Exact models do not exist!!

- $P_a(s, s') = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$ → P predicts the next state

- $R_a(s) = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$ → R predicts the next (immediate) reward.

RL Agents.

- Value Based : Finding best value leads to best policy.
- Policy: Finding better policy through better value functions iteratively.

Learning Paradigm

- Model free
  - Policy / Value Function
- Model based
  - Policy / Value Function

Exploration and Exploitation.

- RL → a trail and error learning
- The agent's current policy should be able to discover a better policy if it exists.
- The agent current policy should be able to 'explore' different / better ways of doing the same task.

Exploration and Exploitation

Exploration : Find more info about the environment.

Exploitation : Utilize the acquired info to obtain the best policy.

An agent should Explore & Exploit.

What is the best way ??

Table of Contents

MDP Definition

Consider the MDP(X, U, P, R) where:

- X is a set of states and U is a set of actions or controls.
- The transition probabilities $P : X \times U \times X \to [0, 1]$ describe, for each state $x \in X$ and action $u \in U$, the conditional probability $P^u_{x,x'} = \Pr\{x' \mid x, u\}$ of transitioning to state $x' \in X$ given the MDP is in state x and takes action u.
- The cost function $R : X \times U \times X \to R$ is the expected immediate cost $R^u_{xx'}$, paid after transition to state $x' \in X$ given that the MDP starts in state $x \in X$ and takes action $u \in U$.

Note: The Markov property refers to the fact that transition probabilities $P^u_{x,x'}$ depend only on the current state x and not on the history of how the MDP attained that state.

MDP

Control law/ Policy

The basic problem for MDP is to find a mapping $\pi : X \times U \to [0,1]$ that gives, for each state x and action u, the conditional probability $\pi(x,u) = \Pr\{u \mid x\}$ of taking action u given that the MDP is in state x.

- Such a mapping is referred to as a closed-loop control or action strategy or policy. The strategy or policy $\pi(x,u) = \Pr\{u \mid x\}$ is called stochastic or mixed if there is a nonzero probability of selecting more than one control when in state x.

MDP

Control law/ Policy

The basic problem for MDP is to find a mapping $\pi : X \times U \to [0, 1]$ that gives, for each state x and action u, the conditional probability $\pi(x, u) = \Pr\{u \mid x\}$ of taking action u given that the MDP is in state x.

- If the mapping $\pi : X \times U \to [0, 1]$ admits only one control, with probability one, when in every state x, the mapping is called a deterministic policy. Then, $\pi(x, u) = \Pr\{u \mid x\}$ corresponds to a function mapping states into controls $\mu(x) : X \to U$.

Optimal Sequential Decision Problems

Stage Cost

Define a stage cost at time k by $r_k = r_k\left(x_k, u_k, x_{k+1}\right)$.
Then $R^u_{xx'} = E\left\{r_k \mid x_k = x, u_k = u, x_{k+1} = x'\right\}$, with $E\{\cdot\}$ as the expected value operator.
Define a performance index as the sum of future costs over the time interval $[k, k+T]$,

$$J_{k,T} = \sum_{i=0}^{T} \gamma^i r_{k+i} = \sum_{i=k}^{k+T} \gamma^{i-k} r_i,$$

where $0 \leq \gamma < 1$ is a discount factor that reduces the weight of costs incurred further in the future.

Optimal Sequential Decision Problems

Stage Cost

Define a stage cost at time k by $r_k = r_k\left(x_k, u_k, x_{k+1}\right)$.
Then $R^u_{xx'} = E\left\{r_k \mid x_k = x, u_k = u, x_{k+1} = x'\right\}$, with $E\{\cdot\}$ as the expected value operator.

Performance Index

Define a performance index as the sum of future costs over the time interval $[k, k+T]$,

$$J_{k,T} = \sum_{i=0}^{T} \gamma^i r_{k+i} = \sum_{i=k}^{k+T} \gamma^{i-k} r_i,$$

where $0 \leq \gamma < 1$ is a discount factor that reduces the weight of costs incurred further in the future.

Control policy

- Control policy $\rightarrow \pi_k(x_k, u_k)$ that is used at each stage k of the MDP.
- *Stationary policies,* where the conditional probabilities $\pi_k(x_k, u_k)$ are independent of k.
  - Then $\pi_k(x, u) = \pi(x, u) = \Pr\{u \mid x\}$, for all k.

Note:

Nonstationary deterministic policies have the form $\pi = \{\mu_0, \mu_1, \cdots\}$, where each entry is a function $\mu_k(x) : X \rightarrow U; k = 0, 1, \ldots$.
Stationary deterministic policies are independent of time, that is, have the form $\pi = \{\mu, \mu, \cdots\}$.

Select a fixed stationary policy $\pi(x, u) = \Pr\{u \mid x\}$.

Control policy

- Select a fixed stationary policy $\pi(x, u) = \Pr\{u \mid x\}$.
- Then the "closed-loop" MDP reduces to a Markov chain with state space X.
- That is, the transition probabilities between states are fixed with no further freedom of choice of actions. The transition probabilities of this Markov chain are given by

$$p_{x,x'} \equiv P^{\pi}_{x,x'} = \sum_{u} \Pr\left\{x' \mid x, u\right\} \Pr\{u \mid x\} = \sum_{u} \pi(x, u) P^{u}_{x,x'}$$

where the Chapman-Kolmogorov identity is used.

Some properties

- A Markov chain is ergodic if all states are positive recurrent and aperiodic.
- Under the assumption that the Markov chain corresponding to each policy, with transition probabilities being ergodic, it can be shown that every MDP has a stationary deterministic optimal policy.
- Then, for a given policy, there exists a stationary distribution $p_\pi(x)$ over X that gives the steady-state probability the Markov chain is in state x.

Value of a Policy

Value

The value of a policy is defined as the conditional expected value of future cost when starting in state x at time k and following policy $\pi(x, u)$ thereafter,

$$V_k^\pi(x) = E_\pi \{J_{k,T} \mid x_k = x\} = E_\pi \left\{ \sum_{i=k}^{k+T} \gamma^{i-k} r_i \mid x_k = x \right\},$$

where $E_\pi\{\}$ is the expected value given that the agent follows policy $\pi(x, u)$, and $V^\pi(x)$ is known as the value function for policy $\pi(x, u)$, which is the value of being in state x given that the policy is $\pi(x, u)$.

Objective

The main objective of MDP is to determine a policy $\pi(x, u)$ to minimize the expected future cost

Optimal policy

$$\begin{aligned}\pi^*(x,u) &= \arg\min_{\pi} V_k^{\pi}(s)\\ &= \arg\min_{\pi} E_{\pi}\left\{\sum_{i=k}^{k+T} \gamma^{i-k} r_i \mid x_k = x\right\}.\end{aligned}$$

This policy is termed the optimal policy.

Objective

The corresponding optimal value is given as

[

Optimal cost]

$$V_k^*(x) = \min_\pi V_k^\pi(x) = \min_\pi E_\pi \left\{ \sum_{i=k}^{k+T} \gamma^{i-k} r_i \mid x_k = x \right\}.$$

Table of Contents

Recursive relationship for *Value function*

The value of the policy $\pi(x, u)$ can be written as

$$V_k^{\pi}(x) = E_{\pi}\left\{J_k \mid x_k = x\right\} = E_{\pi}\left\{\sum_{i=k}^{k+T} \gamma^{i-k} r_i \mid x_k = x\right\},$$

$$V_k^{\pi}(x) = E_{\pi}\left\{r_k + \gamma \sum_{i=k+1}^{k+T} \gamma^{i-(k+1)} r_i \mid x_k = x\right\},$$

$$V_k^{\pi}(x) = \sum_u \pi(x,u) \sum_{x'} P_{xx'}^{u} \Big[R_{xx'}^{u} + \gamma E_{\pi}\left\{\sum_{i=k+1}^{k+T} \gamma^{i-(k+1)} r_i \mid x_{k+1} = x'\right\}\Big].$$

Recursive relationship for *Value function*

Recursive Relationship

Therefore the value function for the policy $\pi(x, u)$ satisfies

$$V_k^{\pi}(x) = \sum_u \pi(x, u) \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_{k+1}^{\pi}\left(x'\right) \right]$$

This equation provides a backward recursion for the value at time k in terms of the value at time $k+1$.

Attention!

Attention!

In computational intelligence and economics, the interest is in utilities and rewards, and the interest is in **maximizing the expected performance index.**

References I

Lewis, F. L., Vrabie, D., & Vamvoudakis, K. G. (2012). Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, *32*(6), 76-105.