Attend our workshop

AMLD EPFL

Deep Reinforcement Leaning for Satellite Constellation Planning and Routing

24 March

9:00 to 17:30

Applied Machine Learning Days

SwissTech Convention Center EPFL, Lausanne, Switzerland

Who are we?

Alexandre Carlhammar

Research in Distributed Space Systems
Founder & President EPFL AI Team
BSc Mechanical Engineering
https://www.linkedin.com/in/alexandre-carlhammar-852844240

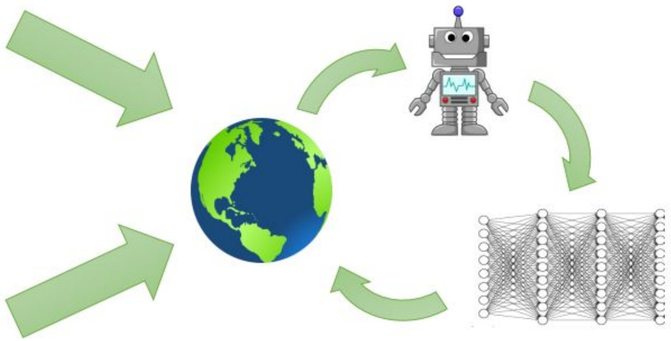
Theo Le Fur

Research in AI
Technical Lead EPFL AI Team
BSc Computer Science
https://www.linkedin.com/in/theo-le-fur-469639265/

Workshop Outline

I. Intro to RL
   A. What is RL? Why is it useful?
   B. Terminology & Notation
   C. Markov Decision Process
   D. Q and V functions
   E. EXERCICES + Coffee Break

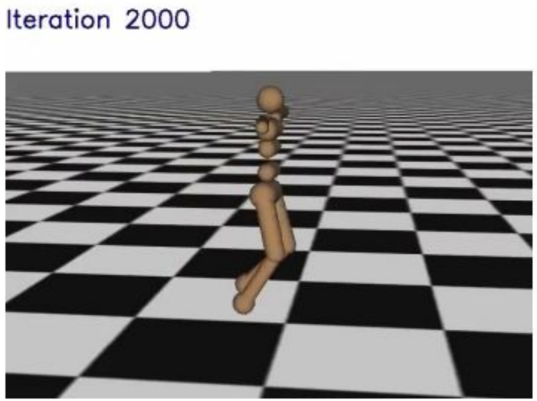
II. Advanced RL
   A. Policy Gradient
   B. Variance and Bias: Baselines
   C. Off policy + Importance Sampling
   D. EXERCICES + Lunch

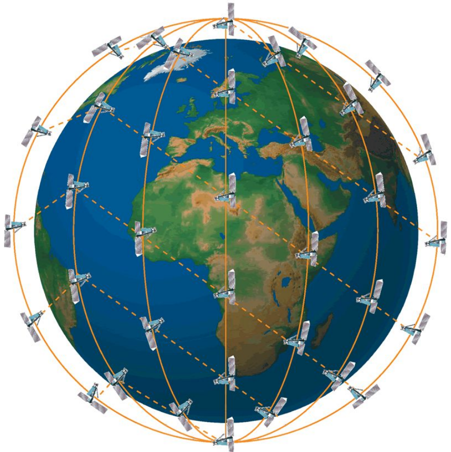


Workshop Outline

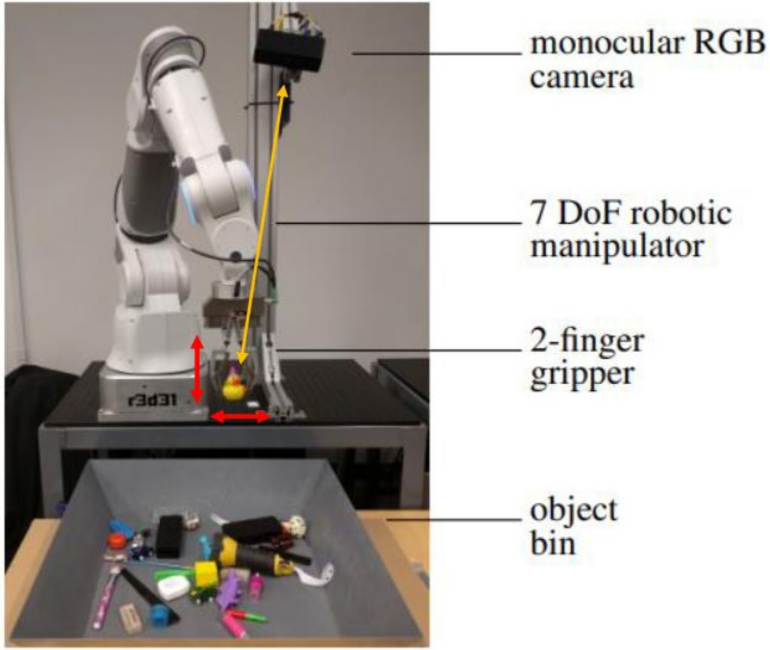
III. Main RL algorithm
   A. Discount Factor
   B. Online & Offline Actor Critic
   C. Generalized Advantage Estimator
   D. Q-Learning
   E. EXERCICES + Coffee Break

IV. Space applications of RL
   A. Intro to space related applications
   B. Past research
   C. Concrete Example
   D. EXERCICES



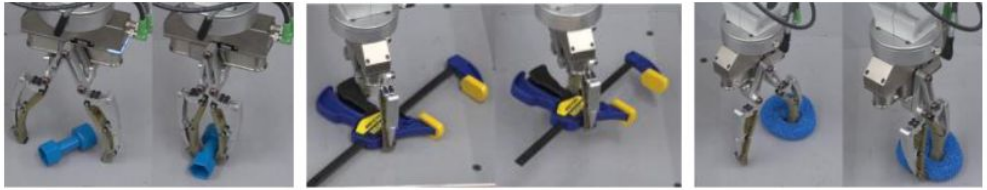


What is RL?



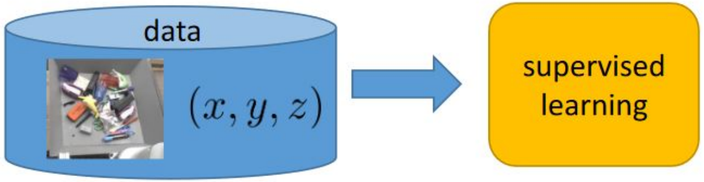
Option 1:

Understand the problem, design a solution

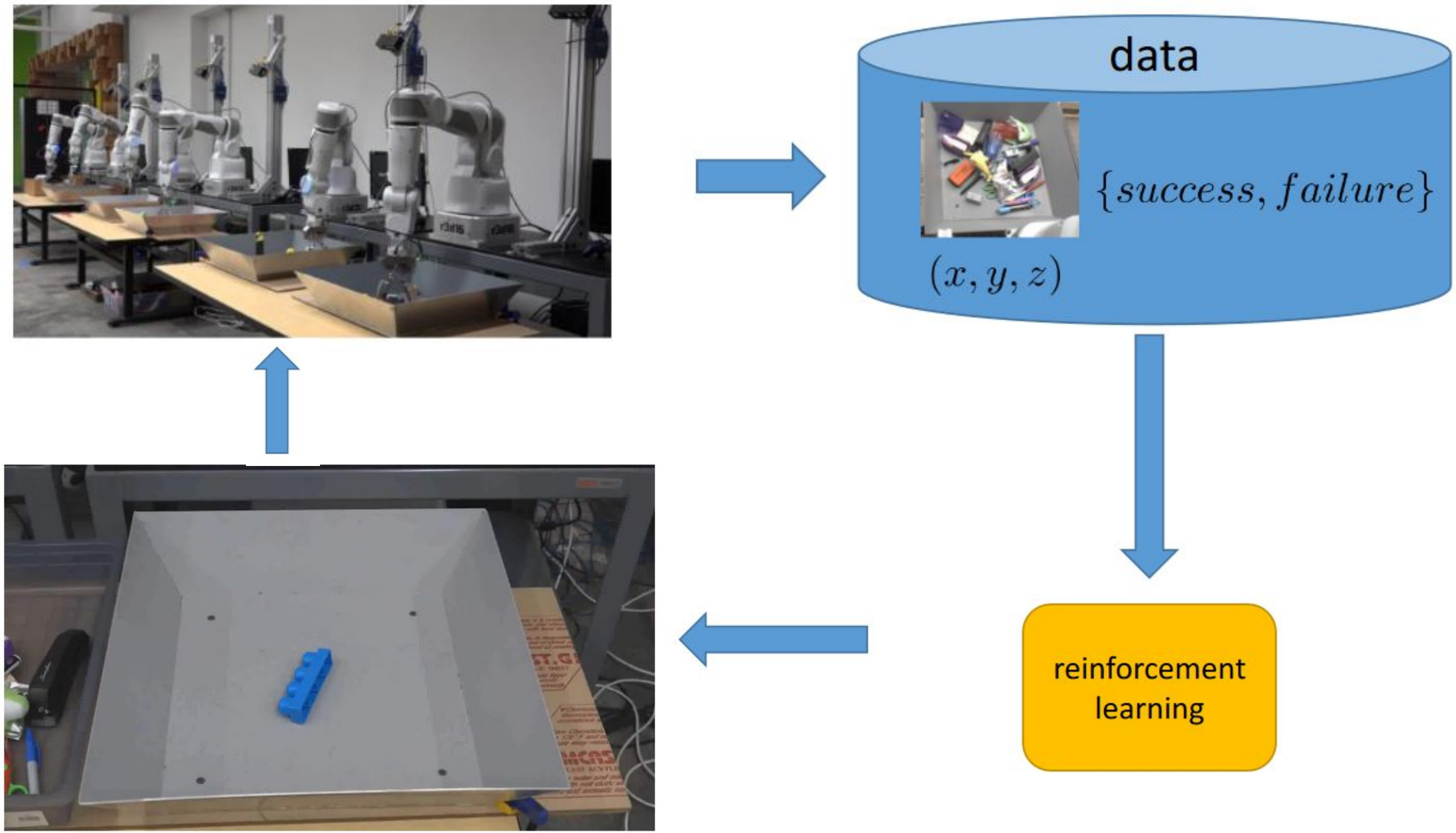


Option 2:

Set it up as a machine learning problem



What is RL?



What is RL?

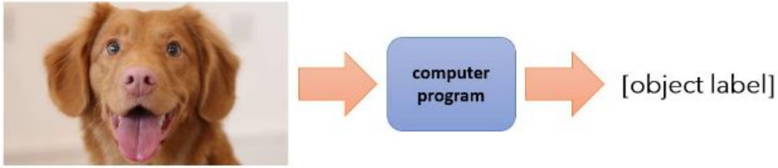
Reinforcement learning is fundamentally 2 things:

- Mathematical formalism for learning based decision making
  → allows to design algorithms

- Approach for learning decision making and control from experience

How does it compare to other methods?

supervised learning



input: $\mathbf{x}$

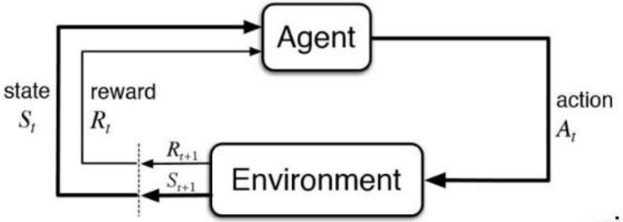
output: $\mathbf{y}$

data: $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}$

goal: $f_\theta(\mathbf{x}_i) \approx \mathbf{y}_i$

someone gives this to you

reinforcement learning



pick your own actions

input: $\mathbf{s}_t$ *at each time step*

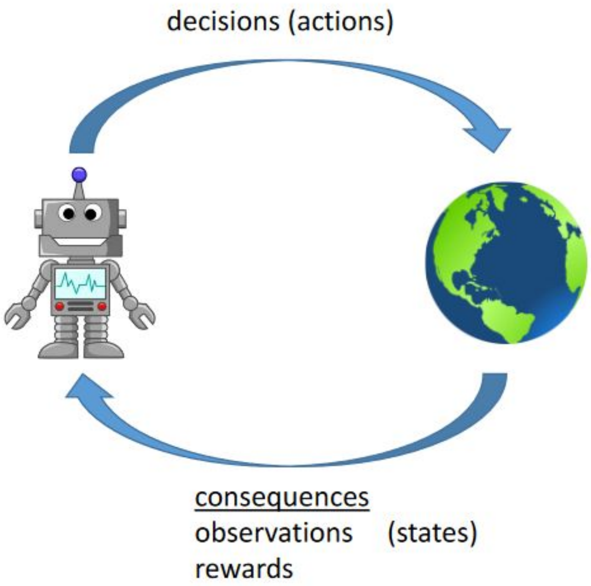
output: $\mathbf{a}_t$ *at each time step*

data: $(\mathbf{s}_1, \mathbf{a}_1, r_1, \ldots, \mathbf{s}_T, \mathbf{a}_T, r_T)$

goal: learn $\pi_\theta : \mathbf{s}_t \rightarrow \mathbf{a}_t$

to maximize $\sum_t r_t$

Examples….



Actions: muscle contractions
Observations: sight, smell
Rewards: food

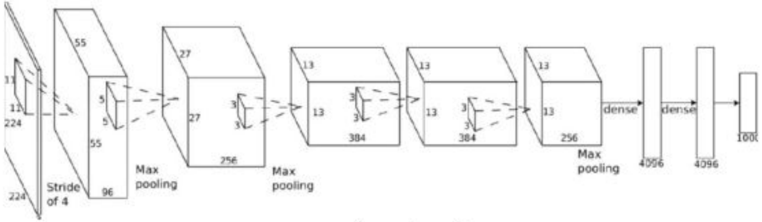
Actions: motor current or torque
Observations: camera images
Rewards: task success measure (e.g., running speed)

Actions: what to purchase
Observations: inventory levels
Rewards: profit

Examples….

Examples….

Terminology & Notation: States & Actions

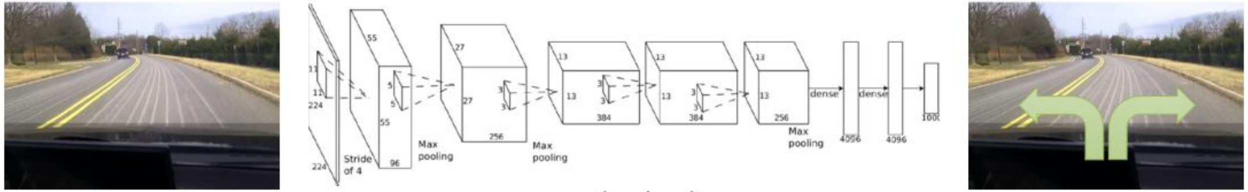


$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

Terminology & Notation: Reward Function



$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

which action is better or worse?

$r(\mathbf{s}, \mathbf{a})$: reward function

tells us which states and actions are better

$\mathbf{s}$, $\mathbf{a}$, $r(\mathbf{s}, \mathbf{a})$, and $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ define Markov decision process

high reward

low reward

Terminology & Notation: Markov Chain

Definitions

Andrey Markov

Markov chain

$\mathcal{M} = \{\mathcal{S}, \mathcal{T}\}$

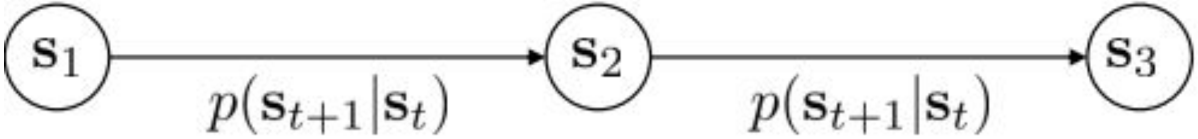
$\mathcal{S}$ – state space states $s \in \mathcal{S}$ (discrete or continuous)

$\mathcal{T}$ – transition operator $p(s_{t+1}|s_t)$

why "operator"? let $\mu_{t,i} = p(s_t = i)$ $\vec{\mu}_t$ is a vector of probabilit

let $\mathcal{T}_{i,j} = p(s_{t+1} = i | s_t = j)$ then $\vec{\mu}_{t+1} = \mathcal{T}\vec{\mu}_t$

Terminology & Notation: Markov Property



Markov Property:
independent of s_{t-1}!!

WHAT'S MISSING?

Terminology & Notation: Markov Decision Process

Definitions

Markov decision process $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, r\}$

$\mathcal{S}$ – state space states $s \in \mathcal{S}$ (discrete or continuous)

$\mathcal{A}$ – action space actions $a \in \mathcal{A}$ (discrete or continuous)

$\mathcal{T}$ – transition operator (now a tensor!)

r – reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

$r(s_t, a_t)$ – reward

Richard Bellman

Terminology & Notation: Partially Observable MDP

Definitions

partially observed Markov decision process $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{E}, r\}$

$\mathcal{S}$ – state space — states $s \in \mathcal{S}$ (discrete or continuous)

$\mathcal{A}$ – action space — actions $a \in \mathcal{A}$ (discrete or continuous)

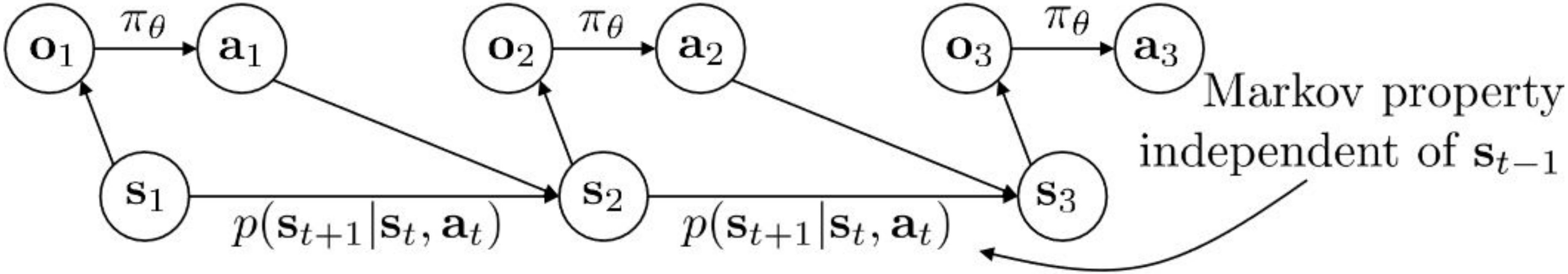
$\mathcal{O}$ – observation space — observations $o \in \mathcal{O}$ (discrete or continuous)

$\mathcal{T}$ – transition operator (like before)

$\mathcal{E}$ – emission probability $p(o_t|s_t)$

r – reward function — $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

Terminology & Notation



Exercice Session

- 3 exercise sessions in total
- work in group as much as you can!!!
- corrections published when ¾ of the allocated time for exercises has passed
- we will provide detailed correction presentation at the end of each session for critical algorithm implementation and/or if we notice you have many similar questions

Exercice Session 1 - 1H

- Assignment 1.1
  - basic Numpy and Torch review
  - advanced Torch review
  - implement a policy parametrized by a gaussian distribution as a neural network

- Assignment 1.2
  - implement a basic "Grid World" environment that mimics the functionality of OpenAI Gym environments