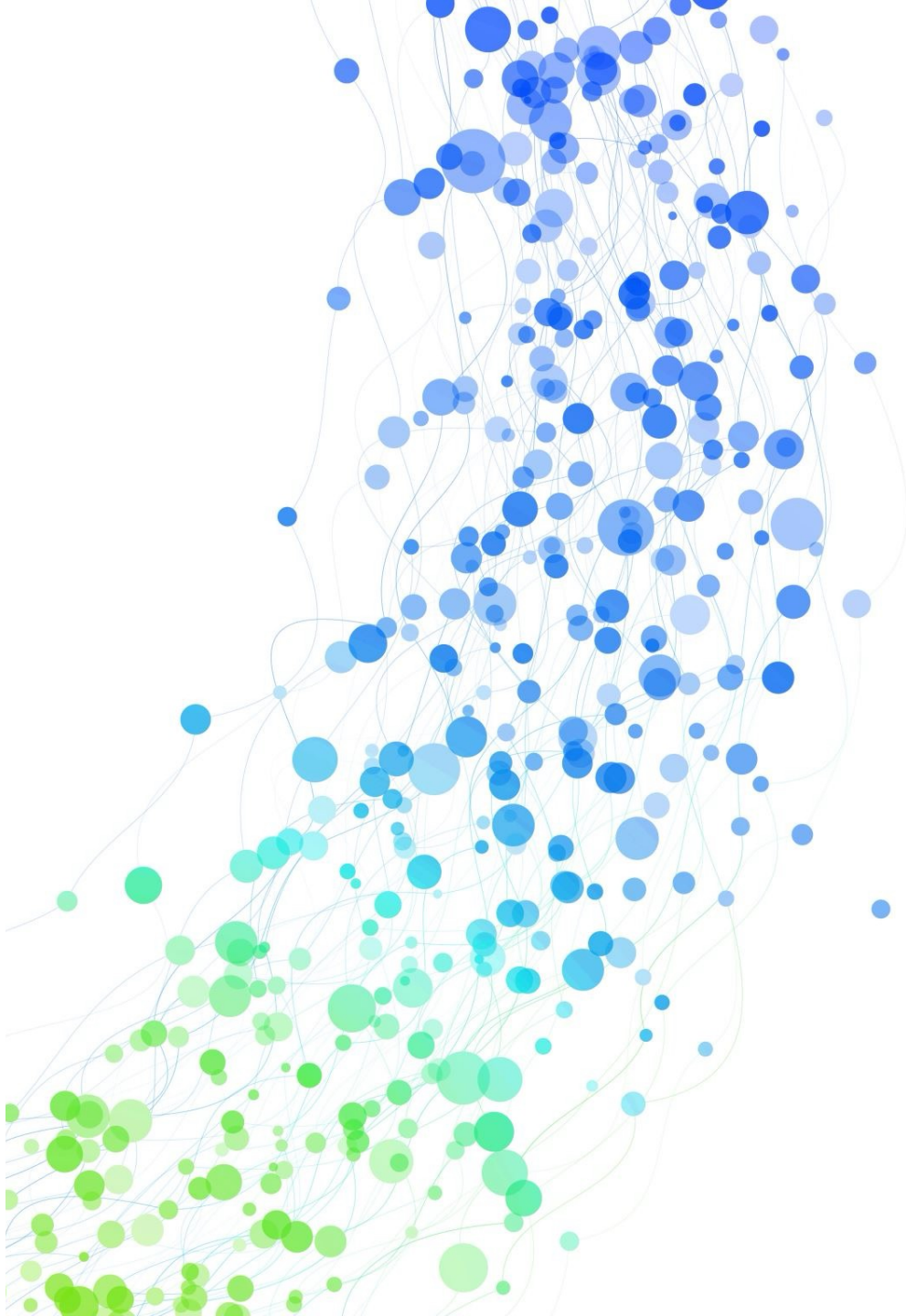
Introduction to Reinforcement Learning – A Tutorial
Barcode



Martin Lorenz
lorenz@cs.uni-leipzig.de
ScaDS.AI, Leipzig University

Reinforcement Learning - Agenda

RL Basics

- Terminology, MDP, POMDP

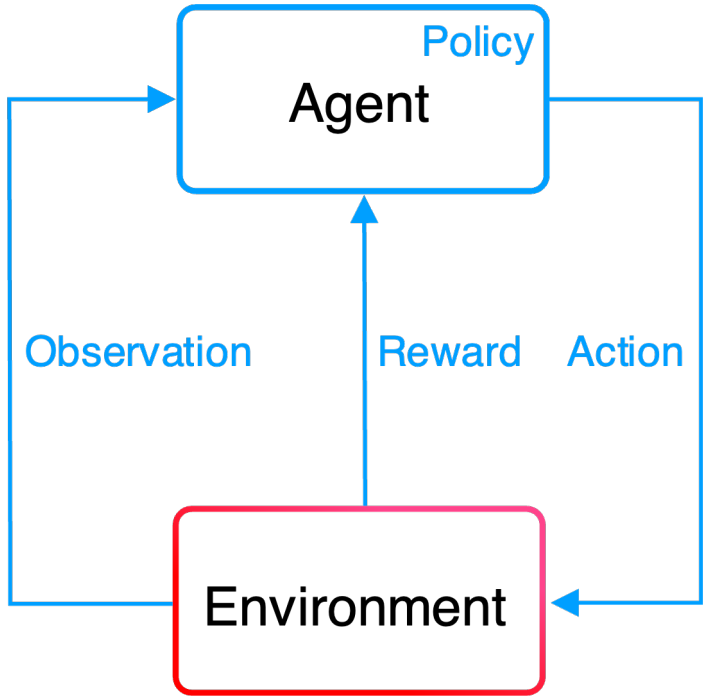
How to implement

- Define Environment, agent
- Train the agent

RL - Basics

Essential Terminology

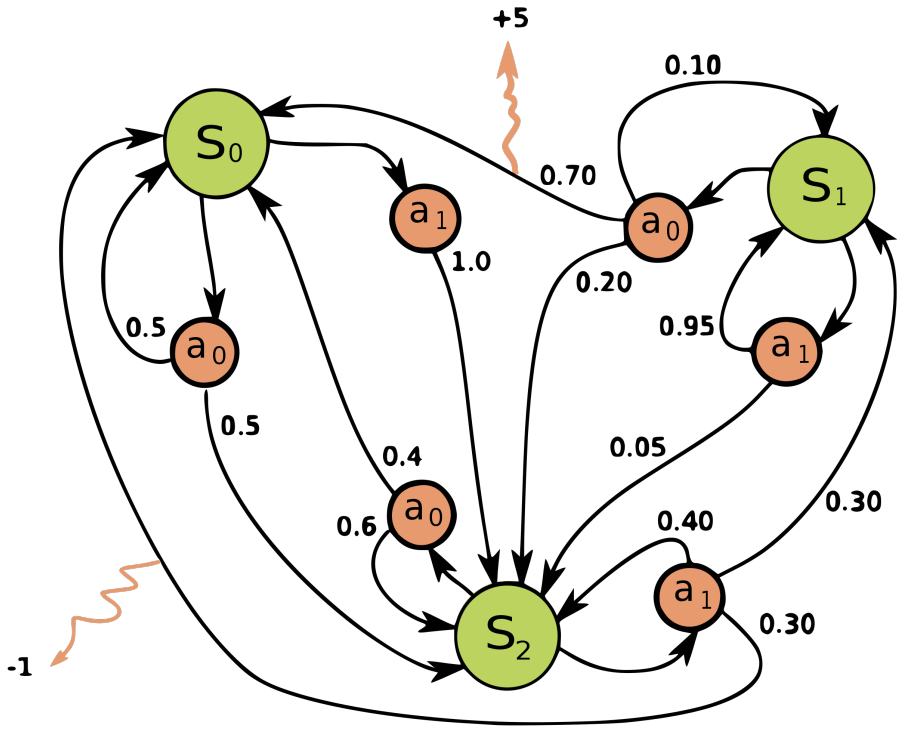
- Agent:
  - The „subject“ (👱 🐕 ✈️ 🤖) that *observes* and *acts* in a given environment
  - Tries to maximize *reward* over time (in our context)
- Environment:
  - World in which agent operates (physical system, simulation)
- State:
  - Situations the agent can encounter during its interactions
  - Snapshot of the environment at time t, denoted as S_t
  - Encapsulates all relevant information for decission making
- Policy (π):
  - The agent's learned strategy to maximize (long-term) reward (π^*)



Markov Decision Process (MDP)

The Foundation of RL

- Markov Decision Process (MDP) (S, A, P, R, γ)
  - S … state space
  - A … action space
  - P … transition function
    - $P(s'|s, a)$ probability of transitioning from state s to state s' when taking action a
  - $R: S \times A \rightarrow \mathbb{R}$ … reward function
    - $R(s, a, s')$ immediate reward when transitioning from state s to s' taking action a
  - γ … discount factor
    - Agent's preference for immediate vs. future rewards
  - Policy π is a (probabilistic) mapping from state space S to action space A

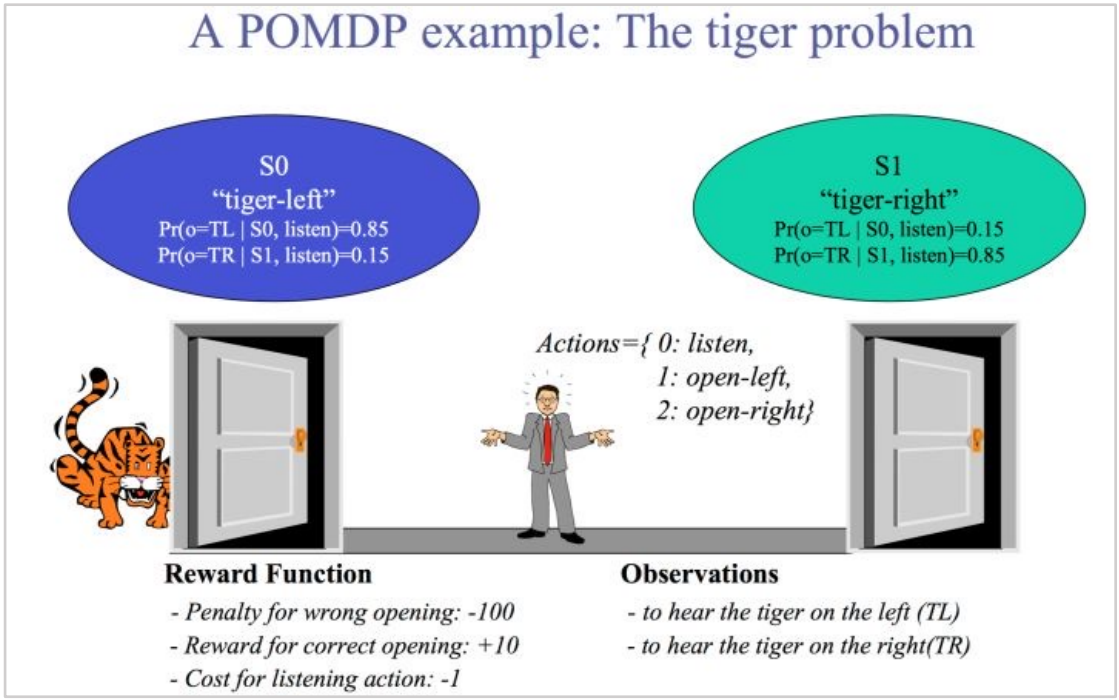


Source: https://commons.wikimedia.org/wiki/File:Markov_Decision_Process.svg#/media/File:Markov_Decision_Process.svg

Partially Observable Markov Decision Process (POMDP)

MDPs for Real World Problems

- POMDP $(S, A, P, R, \Omega, O, \gamma)$
  - S … state space
  - A … action space
  - P … transition function
    - $P(s'|s,a)$ probability of transitioning from state s to state s' when taking action a
  - $R: S \times A \rightarrow \mathbb{R}$ … reward function
    - $R(s,a,s')$ immediate reward when transitioning from state s to s' taking action a
  - Ω … observation space
  - O … conditional observation probabilities
    - $O(o|s',a)$ probability to observe o when transitioning to s' with action a
  - γ … discount factor



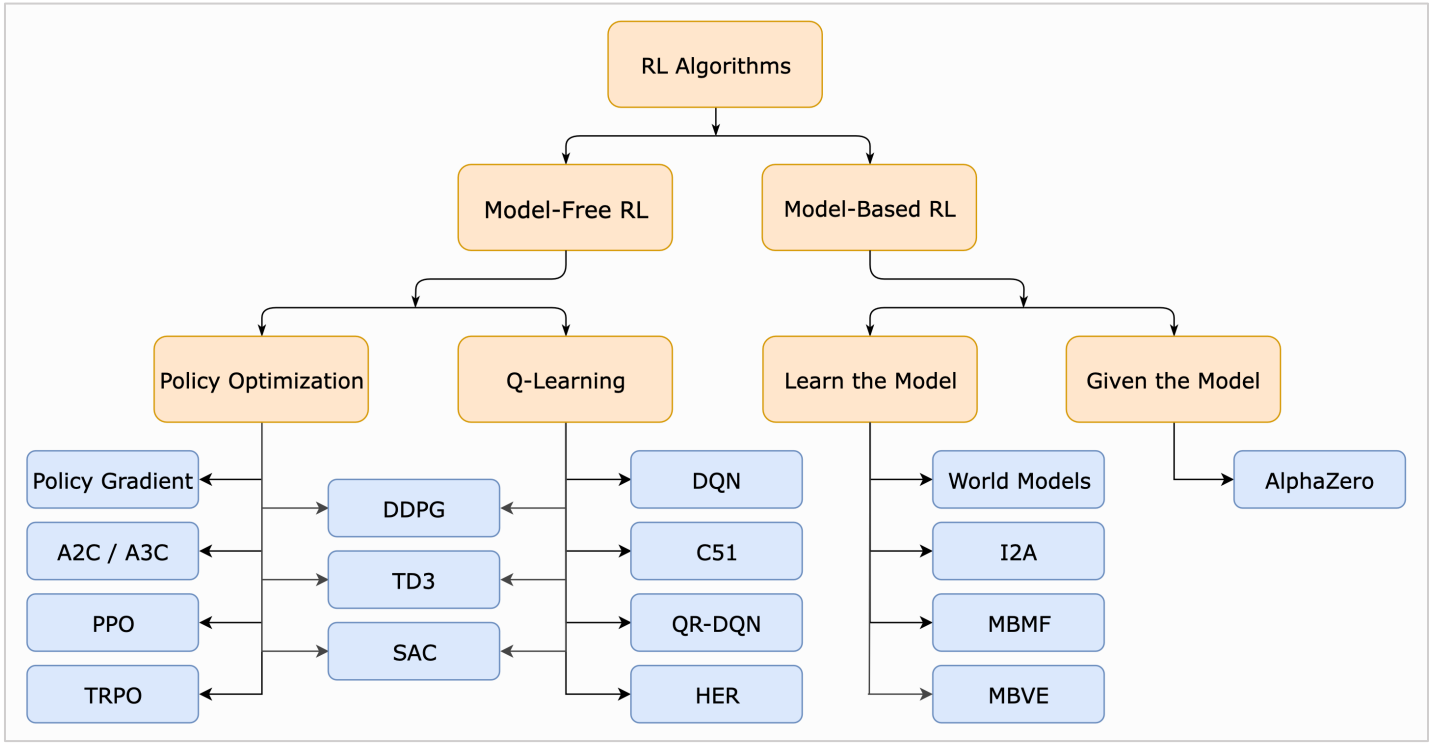
Source: https://user-images.githubusercontent.com/33338567/60923303-8f087100-a296-11e9-8952-c46abcaab698.jpg

How to solve it?

How to find π^*?

- Depends on:
  - MDP / POMDP
  - Model free / Model based

- Bunch of algorithms and architectures, e.g.:
  - Q-Learning
  - Deep Q-Learning / Network (DQN)
  - Soft Actor-Critic (SAC) [Haarnoja et al.]
  - Proximal Policy Optimization (PPO) [Schulman et al.]



Source: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

How to implement?

Frameworks and Tools

- Gymnasium API (Farama Foundation)
  - "An API standard for reinforcement learning with a diverse collection of reference environments"
  - Based on PyTorch
  - https://gymnasium.farama.org

- StableBaselines3 (SB3)
  - "SB3 is a set of reliable implementations of reinforcement learning algorithms in PyTorch. "
  - https://stable-baselines3.readthedocs.io/en/master/

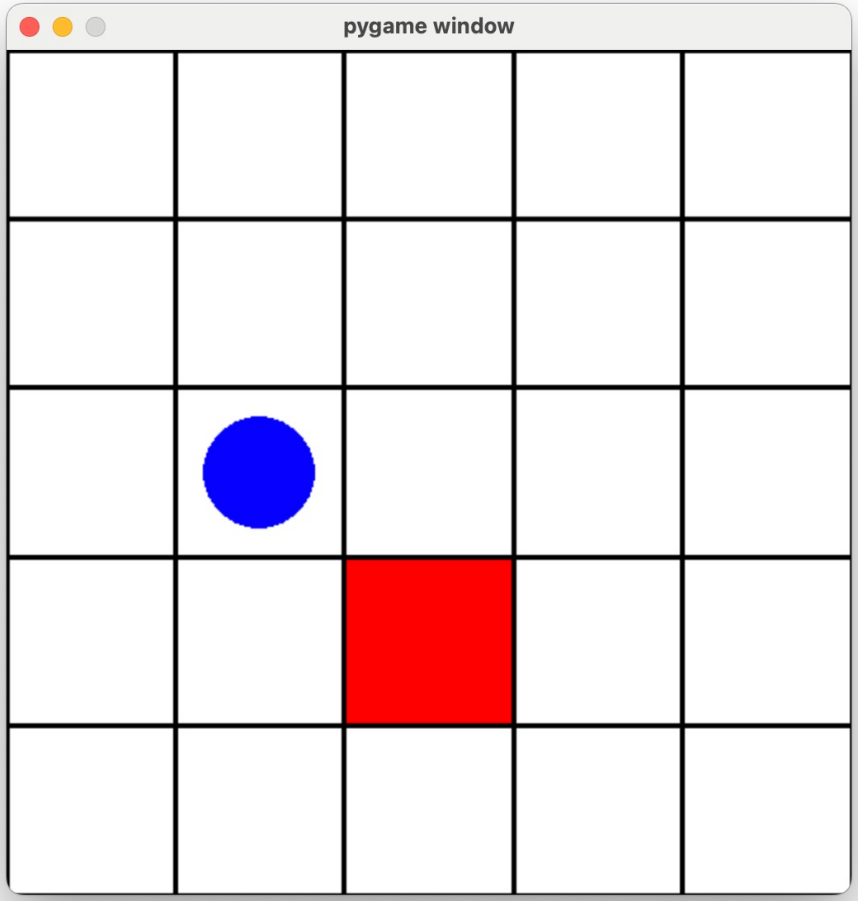
- Ray Rllib
  - "offer[s] support for production-level, highly distributed RL workloads while maintaining unified and simple APIs for a large variety of industry applications. "
  - Supports multi-agent setup
  - https://docs.ray.io/en/latest/rllib/index.html

Let's code

PyCharm

The Project

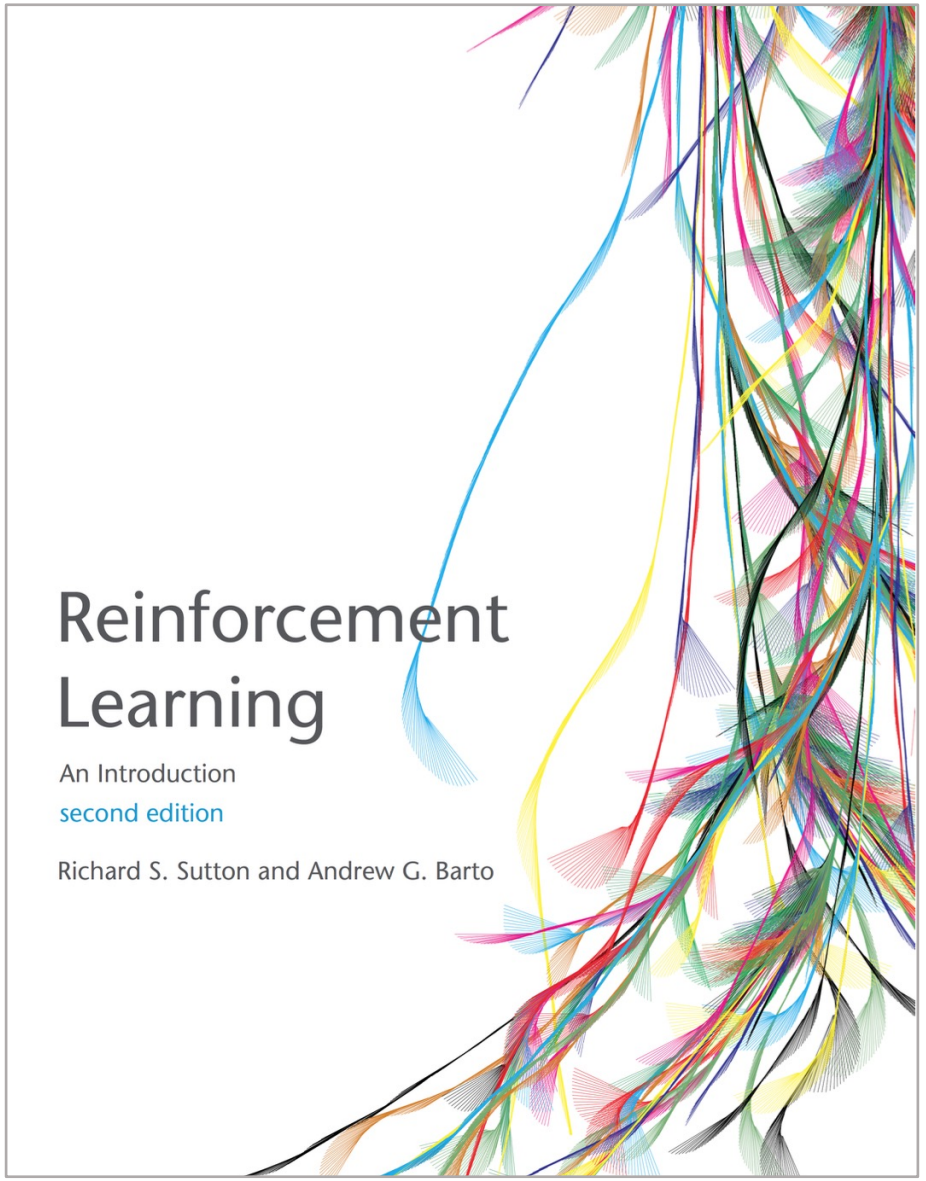
- Scenario:
  - Grid World ($n \times n$)
  - Agent has to find target (constant, random position)
- Actions:
  - Up, Down, Left, Right
- Observation:
  - Current agent's position
  - (distance to target)
- Reward:
  - Step: -1
  - Finding the target: +10
- `git clone https://github.com/ciao-group/RL-Tutorial.git`



Summary

- Intuitive Introduction to Reinforcement Learning
  - Agent, Environemnt, Action, Observation, Reward, Policy
- MDP vs. POMDP
- Implementation with Gymnasium
- Training with Stable Baselines 3

Further reading



http://incompleteideas.net/book/RLbook2020.pdf

Thank you very much!