
HE
IG
VD

Intelligence Artificielle pour les systèmes autonomes (IAA)

Reinforcement Learning (RL)

Prof. Yann Thoma - Prof. Marina Zapater

Février 2024

Basé sur le cours du Prof. A. Geiger

Summary

Today's lesson

→ **Reinforcement learning**

On the previous lecture: supervised techniques

Imitation learning and direct perception

→ Supervised learning

- Using expert demonstration
- Imitation learning: computing the differences between the result of our action and the expert's action (loss function → try to minimize the loss)
- Direct perception: computing the loss on the affordance indicators

→ Today's lecture: reinforcement learning

- Learning models based on the loss that we actually care about
  - Minimize time to get to a location
  - Minimize number of collisions
  - …

Three types of learning

Supervised, unsupervised, reinforcement learning

→ Supervised

- We have a dataset of samples x_i with labels y_i: we want to learn the mapping of $x \rightarrow y$
- Examples: classification (NNs), regression, imitation learning, affordance learning, etc.

→ Unsupervised

- We have a dataset (x_i) and we want to discover the underlying structure
- Examples: clustering

→ Reinforcement Learning (RL)

- Agent interacting with an environment that provides numeric rewards
- Goal: taking actions to maximize the reward
- Example: learning of manipulation and control tasks

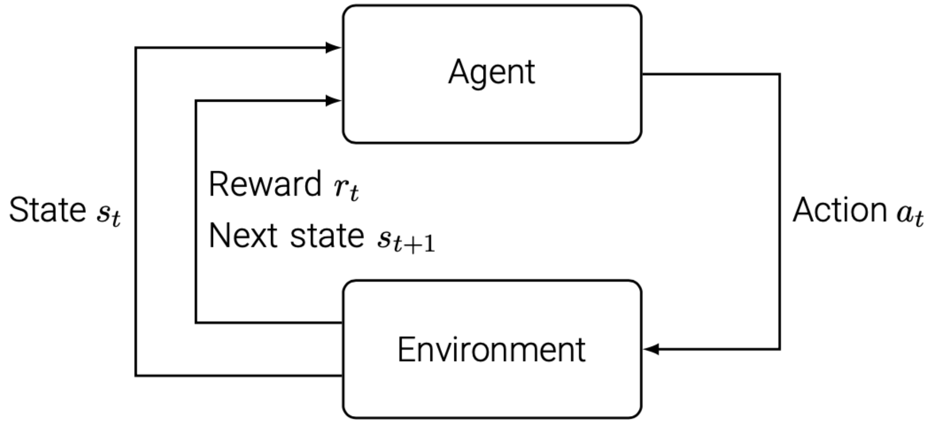
Reinforcement Learning

Overview

→ Agent observes environment state s_t at time t

→ Send an action at at time t to the environment

→ Environment returns the reward and its new state



→ **Goal**: selecting actions that maximize the reward.

→ Knowing that:

- Actions may have long-term consequences
- Reward may be delayed, not instantaneaous
- It might be better to sacrifice immediate reward to gain more long-term reward

Examples

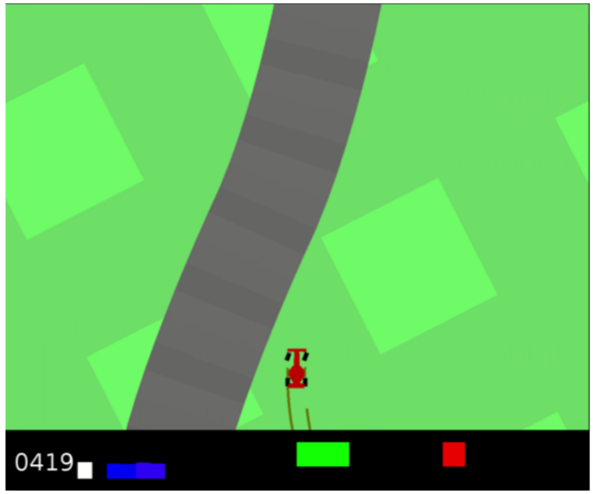
Atari game, AlphaGo, Car racing

https://openai.com/research/gym-retro

- **Objective:** Maximize game score
- **State:** Raw pixels of screen (210x160)
- **Action:** Left, right, up, down
- **Reward:** Score increase/decrease at t

https://deepmind.google/technologies/alphago/

- **Objective:** Winning the game
- **State:** Position of all pieces
- **Action:** Location of next piece
- **Reward:** 1 if game won, 0 otherwise



- **Objective:** Lane Following
- **State:** Image (96x96)
- **Action:** Acceleration, Steering

How do we know which actions to take

A simple grid example

→ Goal: reaching one of the terminal states (marked with *) in the least number of actions possible

→ Penalty: negative “reward” given for every transition made

actions = {
1. right →
2. left ←
3. up ↑
4. down ↓

}

states

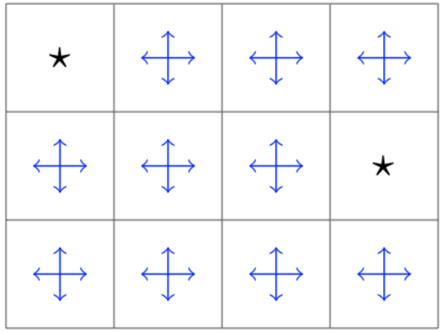
$\star$			
			$\star$

reward: $r = -1$ for each transition

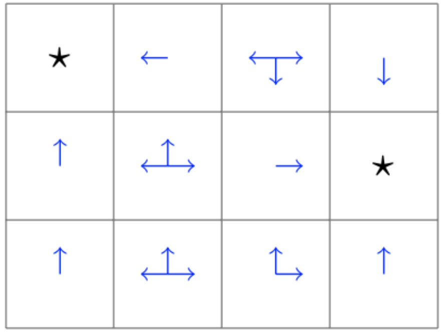
The simple grid example

Random policy vs optimal policy

→ Arrows indicate equal probability of moving into each direction



Random Policy



Optimal Policy

→ How do we choose how to cover the possible space of actions in RL?

Q-learning

A model-free algorithm for RL for learning the value of an action

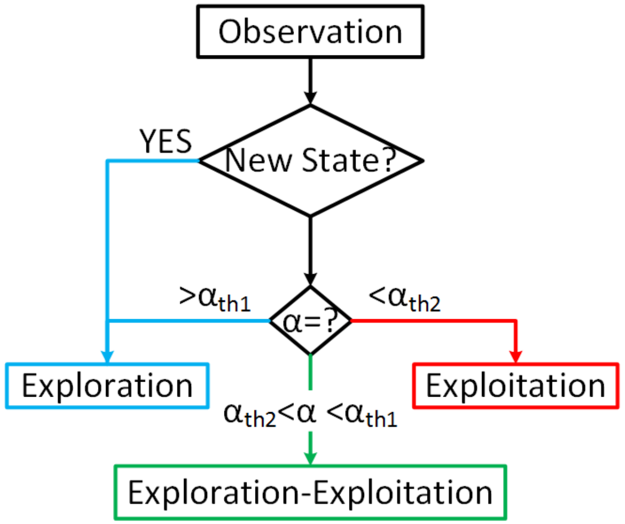
→ The algorithm calculates the Quality of a state-action combination

- Trying to maximize total reward over any and all successive steps

→ Builds and updates the Quality of a sequence of actions iteratively

→ Implementation:

- Initialize Q-table and initial state randomly
- Repeat:
  - Observe state, choose action using epsilon-greedy strategy
  - Observe reward and next state
  - Compute error
  - Update Q-tables
- Explore new states until a certain threshold is achieved, then exploit what has been learned



$$Q^{new}(S_t, A_t) \leftarrow (1 - \underbrace{\alpha}_{\text{learning rate}}) \cdot \underbrace{Q(S_t, A_t)}_{\text{current value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{R_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(S_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{new value (temporal difference target)}}$$

Q-learning example

The mouse finding the cheese in the maze

Problem

Q-table

	←	→	↑	↓
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0

https://huggingface.co/deep-rl-course/unit2/q-learning-example?fw=pt

What’s the main issue of Q-learning?

Scalability, and how to solve it

→ Scalability: tables don’t scale to high-dimensional state/action pairs

→ Solution:
- Multi-agent reinforcement learning: splitting the environment, agents collaborate/compete observing parts of it
- Deep Q-learning: use a function approximator (a neural network) to represent Q(s,a)

→ Deep Q-learning:
- Very popular today, but has shortcomings
- Long training times, simplistic exploration strategy, action space limited

“Learn to drive in a day”

Real world RL demo for self-driving by Wayve

→ Input: single camera image

→ Action: steering and speed

→ Reward: distance travelled without safety driver taking control

→ No maps / localization required

→ 4 Conv layers / 2 FC layers

→ Only 35 training episodes

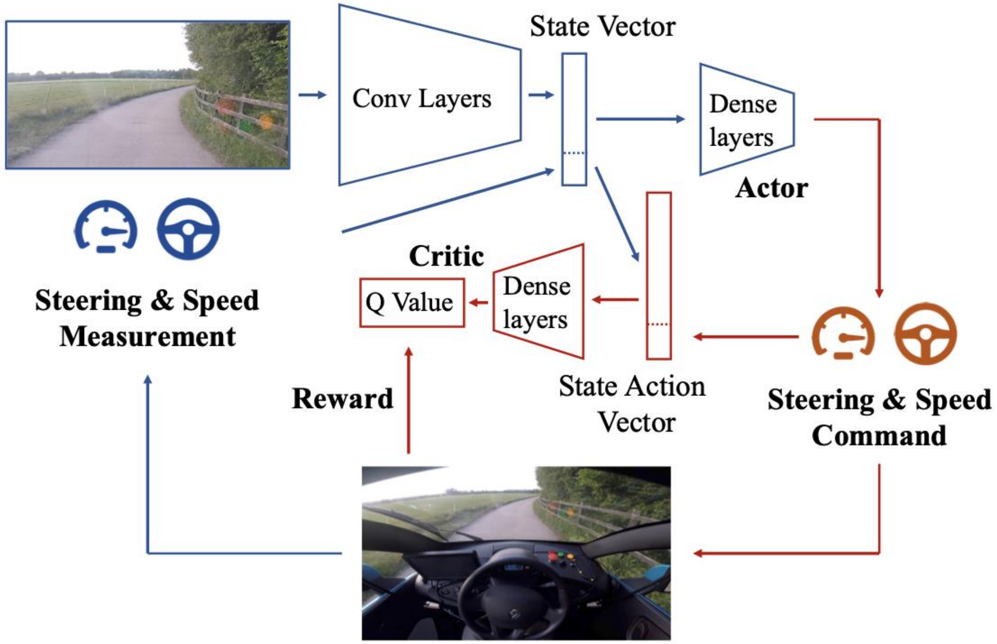


Fig. 1: We design a deep reinforcement learning algorithm for autonomous driving. This figure illustrates the actor-critic algorithm which we use to learn a policy and value function for driving. Our agent maximises the reward of distance travelled before intervention by a safety driver.

https://wayve.ai/thinking/learning-to-drive-in-a-day/
https://arxiv.org/pdf/1807.00412.pdf

“Learn to drive in a day”

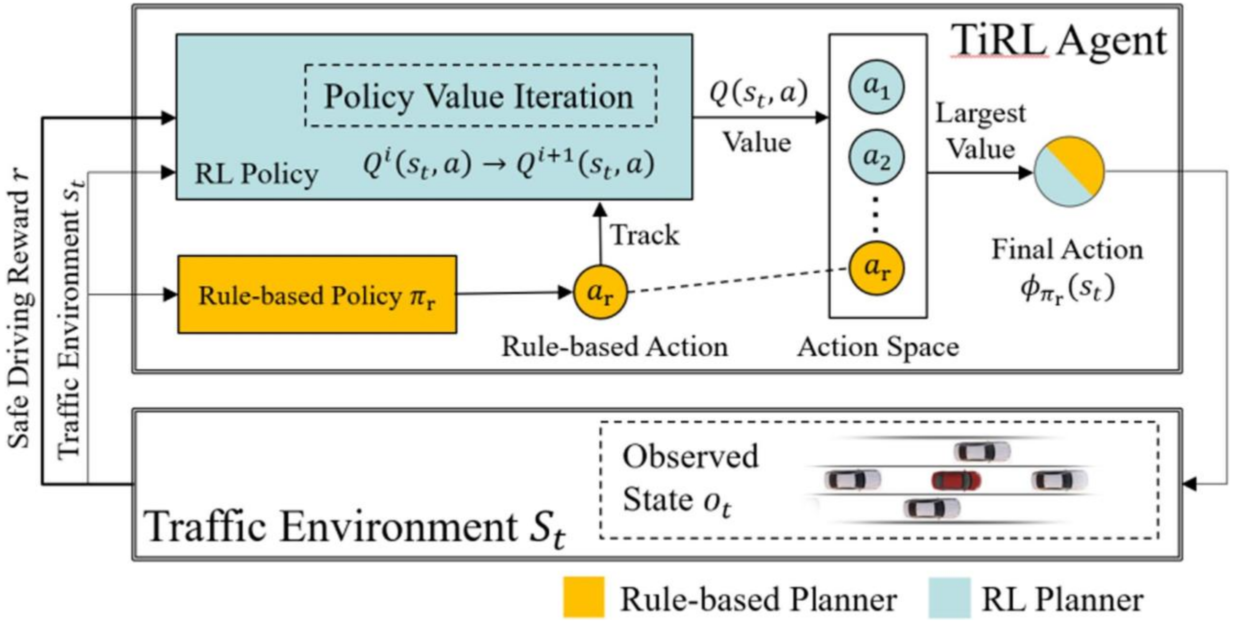
Youtube video

https://www.youtube.com/watch?v=eRwTbRtnT1I&t=5s

Expert and Reinforcement learning

Take the best of the two worlds

Trustworthy improvement RL (TiRL)



https://www.sciencedirect.com/science/article/pii/S0968090X22000997

Reinforcement learning for safety validation

→ Finding corner cases and safety-critical events is hard

→ Use reinforcement learning to generate such events

→ https://www.nature.com/articles/s41586-023-05732-2

TODO’s for last lecture

Exercises

1. Analysis of the Dronet running on the Crazyflie (papers provided on Cyberlearn)

2. Conditional Imitation Learning

3. Conditional Affordance Learning

HE VD
IG

TODO’s for today

Exercises

1. Understand the mouse and cheese problem

2. Checking the Wayve paper and demo
(”Learn to drive in a day”)

3. RL using Deep Q-Learning (Deep Q Networks)
Using OpenAI Gym (Car Racing environment)

    - https://github.com/andywu0913/OpenAI-GYM-CarRacing-DQN
    - https://towardsdatascience.com/applying-a-deep-q-network-for-openais-car-racing-game-a642daf58fc9
    - https://scientific-python.readthedocs.io/en/latest/notebooks_rst/6_Machine_Learning/04_Exercices/02_Practical_Work/02_RL_2_CarRacing.html

4. Take a look at the RL Zoo
https://github.com/DLR-RM/rl-baselines3-zoo

REDS
Institut
Reconfigurable
and Embedded
Digital Systems

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale