Reinforcement Learning: An Introduction

"Smart Traffic Lights" Project

Vladislav Savinov

Mathematics, Algorithms & Data Science
St Petersburg State University

April 21, 2022

Summary

What is Reinforcement Learning?

Definition

*Reinforcement Learning (RL) is an area of machine learning concerned with how **intelligent agents** ought to take **actions** in an **environment** in order to maximize the notion of cumulative **reward**.*

An approximate dynamic programming or neuro-dynamic programming

Definition

*Reinforcement Learning (RL) is an area of machine learning concerned with how **intelligent agents** ought to take **actions** in an **environment** in order to maximize the notion of cumulative **reward**.*

An approximate dynamic programming or neuro-dynamic programming

Definition, Intelligence

To be able to learn to make decisions to achieve goals

Key concepts

1. People and animals learn by **interacting with our environment**
2. This differs from certain types of learning:
   - 2.1 It is **active** rather than passive
   - 2.2 Interactions are often **sequential** - future interactions can depend on earlier ones
3. We are **goal-directed**
4. We can learn **without examples** of optimal behavior

What is Reinforcement Learning?

1. Science and framework of learning to make decisions from interaction
2. The purpose is to learn an "optimal" policy in terms of maximizing the "reward function"
3. This requires us to think about
   - 3.1 long-term consequences of actions
   - 3.2 actively gathering experience
   - 3.3 predicting the future
   - 3.4 dealing with uncertainty

RL Scenario

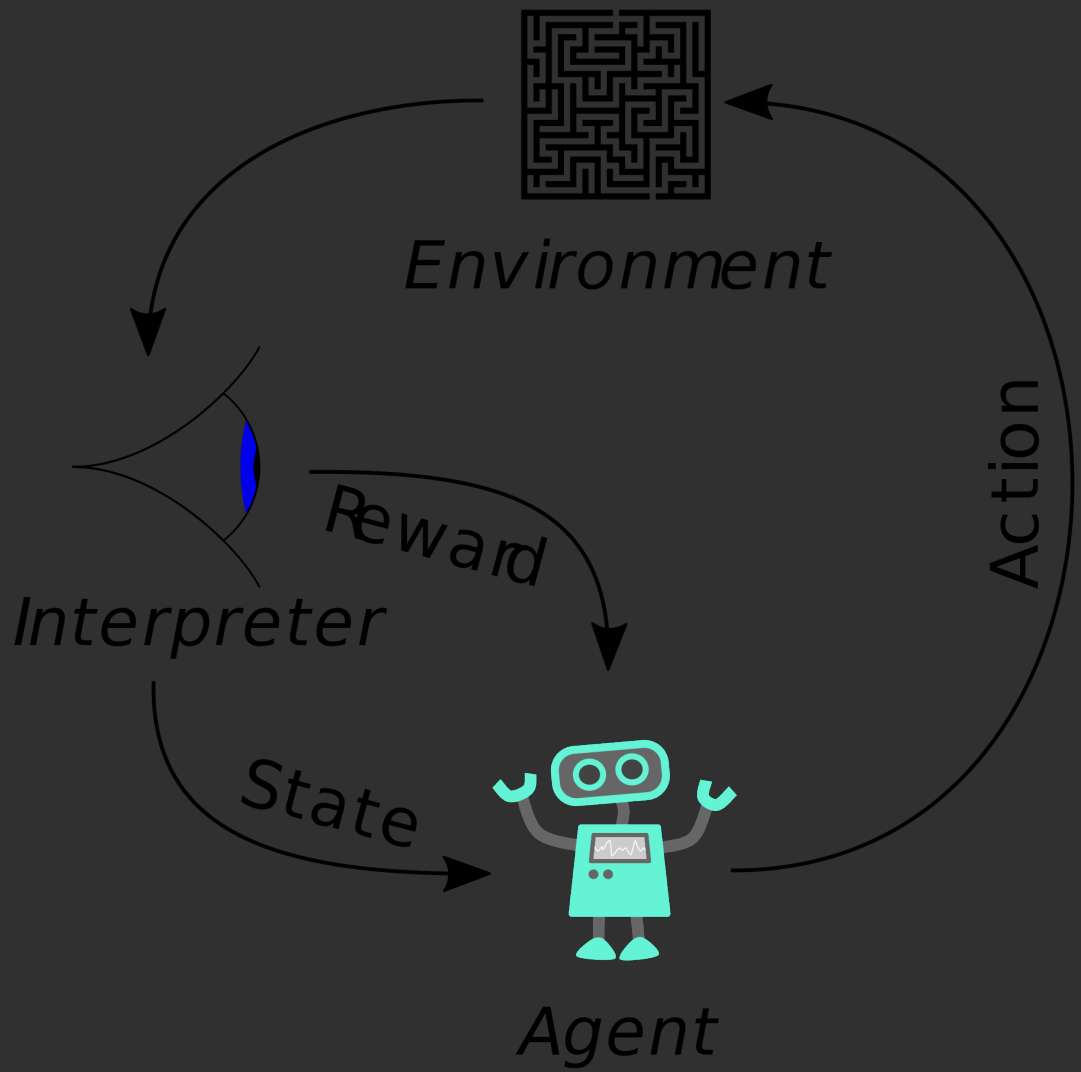


Figure: The typical framing of an RL scenario

Definition, Agent

An agent is a system that receives at time t an observation O_t and outputs an action A_t.

More formally, the agent is a system that selects an action $A_t = \alpha(H_t)$ at time t given its experience history $H_t = O_1, A_1, O_2, \ldots, A_{t-1}, O_t$.

Definition, Agent

An agent is a system that receives at time t an observation O_t and outputs an action A_t.

More formally, the agent is a system that selects an action $A_t = \alpha(H_t)$ at time t given its experience history $H_t = O_1, A_1, O_2, \ldots, A_{t-1}, O_t$.

Definition, Environment

An environment is a system that receives action A_t at time t and responds with an observation O_{t+1} at the next time step

More formally, an environment is a system $O_{t+1} = \varepsilon(H_t, A_t, \eta_t)$ that determines the next observation O_{t+1} that the agent will receive from the environment, given experience history H_t, the latest action A_t, and potentially a source of randomness η_t.

Definition, Environment

An environment is a system that receives action A_t at time t and responds with an observation O_{t+1} at the next time step

More formally, an environment is a system $O_{t+1} = \varepsilon(H_t, A_t, \eta_t)$ that determines the next observation O_{t+1} that the agent will receive from the environment, given experience history H_t, the latest action A_t, and potentially a source of randomness η_t.

Definition, Environment

Simulation, the most common ones are provided by OpenAI through the Gym library.

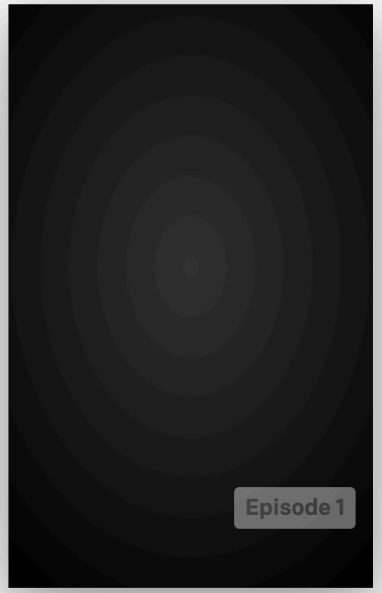
> *Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.*

Definition, Environment

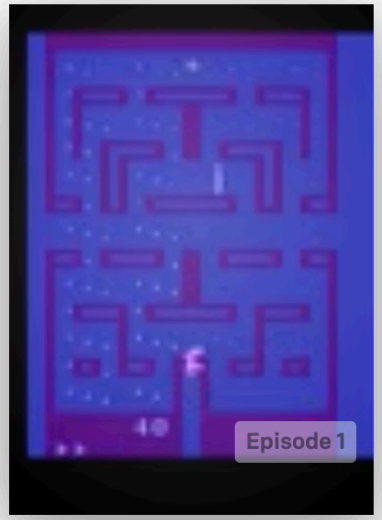
Atari

Reach high scores in Atari 2600 games.

AirRaid-ram-v0
Maximize score in the game AirRaid, with RAM as input



AirRaid-v0
Maximize score in the game AirRaid, with screen images as input



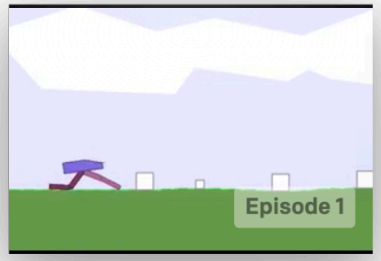
Alien-ram-v0
Maximize score in the game Alien, with RAM as input

Definition, Environment

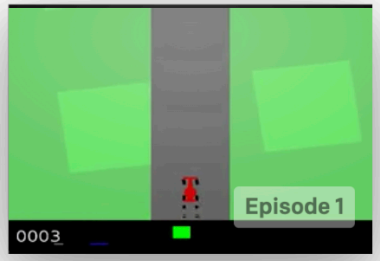
Box2D

Continuous control tasks in the Box2D simulator.

BipedalWalker-v2
Train a bipedal robot to walk.



BipedalWalkerHardcore-v2
Train a bipedal robot to walk over rough terrain.



CarRacing-v0
Race a car around a track.

LunarLander-v2
Navigate a lander to its landing pad.

LunarLanderContinuous-v2
Navigate a lander to its landing pad.

Definition, Environment

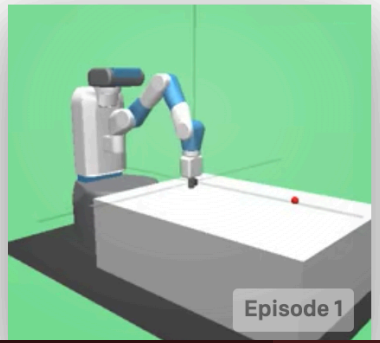
Robotics

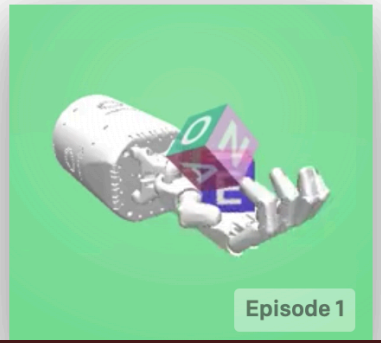
Simulated goal-based tasks for the Fetch and ShadowHand robots.

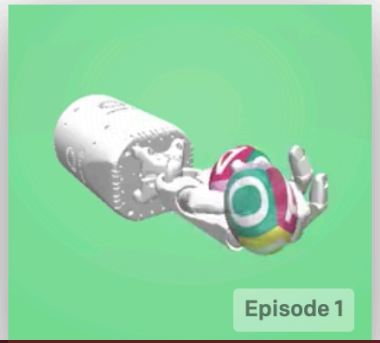
FetchPickAndPlace-v1
Lift a block into the air.

FetchPush-v1
Push a block to a goal position.

FetchReach-v1
Move Fetch to a goal position.







Definition, Reward

Theorem (Reward hypothesis)

Any goal can be formalized as the outcome of maximizing a cumulative reward

A reward is a special scalar observation R_t, emitted at every time-step t by a reward signal in the environment, that provides an instantaneous measurement of progress towards a goal.

Definition, Reward

Theorem (Reward hypothesis)

Any goal can be formalized as the outcome of maximizing a cumulative reward

A reward is a special scalar observation R_t, emitted at every time-step t by a reward signal in the environment, that provides an instantaneous measurement of progress towards a goal.

Reward Examples

1. Manage an investment portfolio - gains, gains minus risk
2. Make a robot walk - distance, speed
3. Fly a helicopter - air time
4. Play a video game - win

Definition, Return

Return: $R = \sum_{t=0}^{\infty} \gamma^t r_t$, can be defined as the discounted sum of **future** rewards, γ is the discount-rate, $\gamma \in [0, 1]$

Central Optimization Problem

$$J(\pi) = \int_{\tau} P(\tau \mid \pi) R(\tau) = \mathbb{E}_{\tau \sim \pi}[R(\tau)]$$

The central optimization problem is to learn an **optimal policy** π^*

$$\pi^* = \arg\max_{\pi} J(\pi)$$

Definition, Value Functions

On-policy Value function:

$$V^{\pi}(s) = \mathbb{E}_{\tau \sim \pi}[R(\tau) \mid s_0 = s],$$

expected return starting with the state s

Optimal Value Function:

$$V^{*}(s) = \max_{\pi} \mathbb{E}_{\tau \sim \pi}[R(\tau) \mid s_0 = s],$$

which gives the expected return if you start in state s and always act according to the optimal policy in the environment

Definition, Value Functions

On-policy Action-Value function:

$$Q^{\pi}(s, a) = \mathbb{E}_{\tau \sim \pi}[R(\tau) \mid s_0 = s, a_0 = a],$$

which gives the expected return if you start in state s, take an arbitrary action a (which may not have come from the policy), and then forever after act according to policy π

Optimal Action-Value Function

$$Q^{*}(s, a) = \max_{\pi} \mathbb{E}_{\tau \sim \pi}[R(\tau) \mid s_0 = s, a_0 = a],$$

which gives the expected return if you start in state s, take an arbitrary action a, and then forever after act according to the optimal policy in the environment

Definition, Bellman Equations

All four of the value functions obey special equations called Bellman equations.

$$V^{\pi}(s) = \mathbb{E}_{a\sim\pi, s'\sim P}[r(s,a) + \gamma V^{\pi}(s')]$$

$$Q^{\pi}(s,a) = \mathbb{E}_{s'\sim P}[r(s,a) + \gamma \mathbb{E}_{a'\sim\pi}[Q^{\pi}(s',a')]],$$

the next state s' is sampled from the environment's transition rules.

The crucial difference between the Bellman equations for the on-policy value functions and the optimal value functions, is the absence or presence of the $\max$ over actions.

Its inclusion reflects the fact that whenever the agent gets to choose its action, in order to act optimally, it has to pick whichever action leads to the highest value.

Real world applications

Similarity to the real world

1. Learning
2. Perception
   - 2.1 image segmentation to avoid falling off a cliff
   - 2.2 object recognition to classify healthy and poisonous foods
3. Social Intelligence, interaction with other agents (game theory)
4. Language
5. Generalization, transferring solution from one problem to another (from paintings to photographs etc.)
6. General Intelligence (the ability to flexibly achieve a variety of goals in different contexts?)

Similarity to the real world

1. Learning
2. Perception
   - 2.1 image segmentation to avoid falling off a cliff
   - 2.2 object recognition to classify healthy and poisonous foods
3. Social Intelligence, interaction with other agents (game theory)
4. Language
5. Generalization, transferring solution from one problem to another (from paintings to photographs etc.)
6. General Intelligence (the ability to flexibly achieve a variety of goals in different contexts?)

Similarity to the real world

1. Learning
2. Perception
   - 2.1 image segmentation to avoid falling off a cliff
   - 2.2 object recognition to classify healthy and poisonous foods
3. Social Intelligence, interaction with other agents (game theory)
4. Language
5. Generalization, transferring solution from one problem to another (from paintings to photographs etc.)
6. General Intelligence (the ability to flexibly achieve a variety of goals in different contexts?)

Similarity to the real world

1. Learning
2. Perception
   - 2.1 image segmentation to avoid falling off a cliff
   - 2.2 object recognition to classify healthy and poisonous foods
3. Social Intelligence, interaction with other agents (game theory)
4. Language
5. Generalization, transferring solution from one problem to another (from paintings to photographs etc.)
6. General Intelligence (the ability to flexibly achieve a variety of goals in different contexts?)

Similarity to the real world

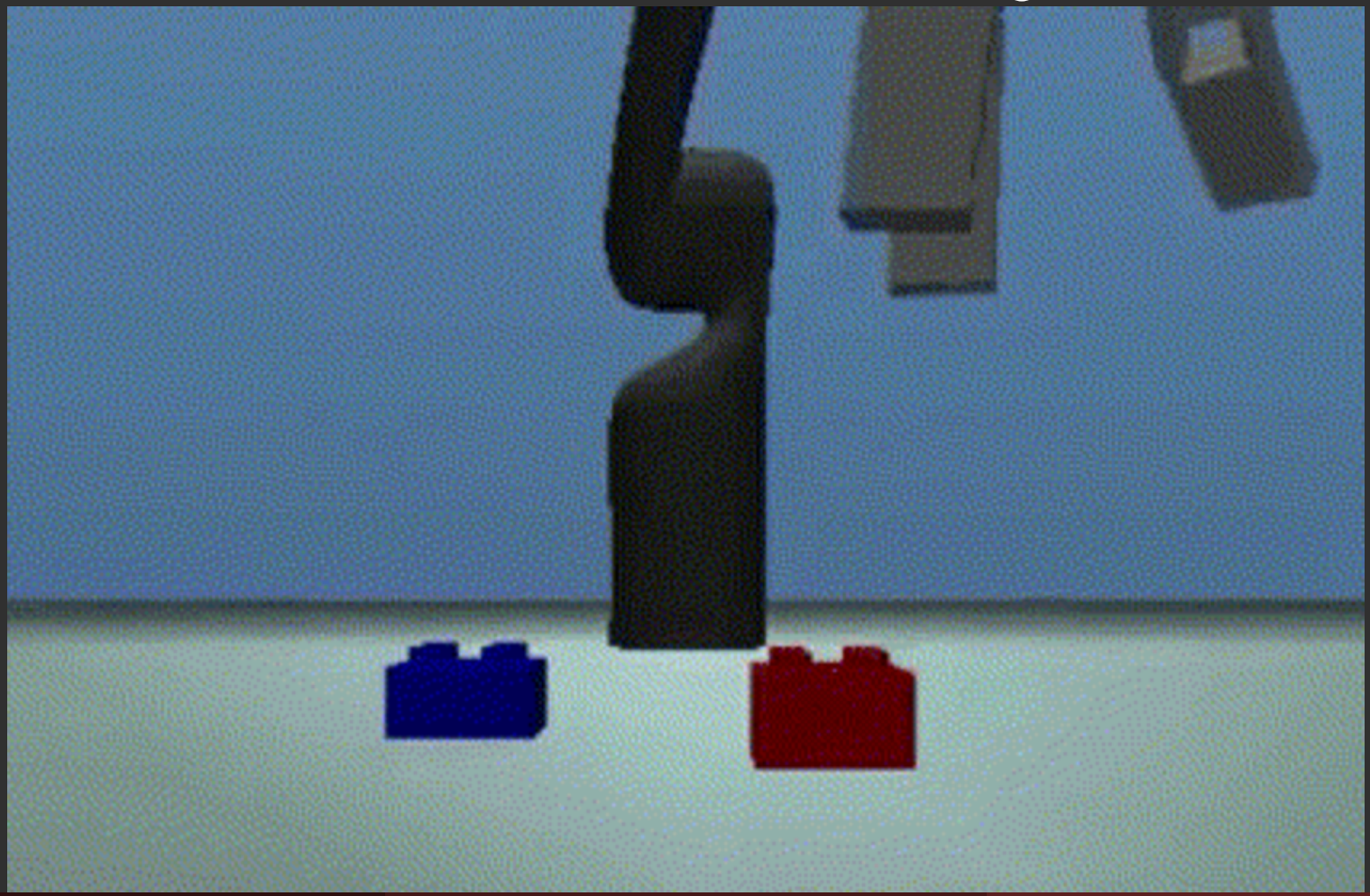
1. Learning
2. Perception
   - 2.1 image segmentation to avoid falling off a cliff
   - 2.2 object recognition to classify healthy and poisonous foods
3. Social Intelligence, interaction with other agents (game theory)
4. Language
5. Generalization, transferring solution from one problem to another (from paintings to photographs etc.)
6. General Intelligence (the ability to flexibly achieve a variety of goals in different contexts?)

Similarity to the real world

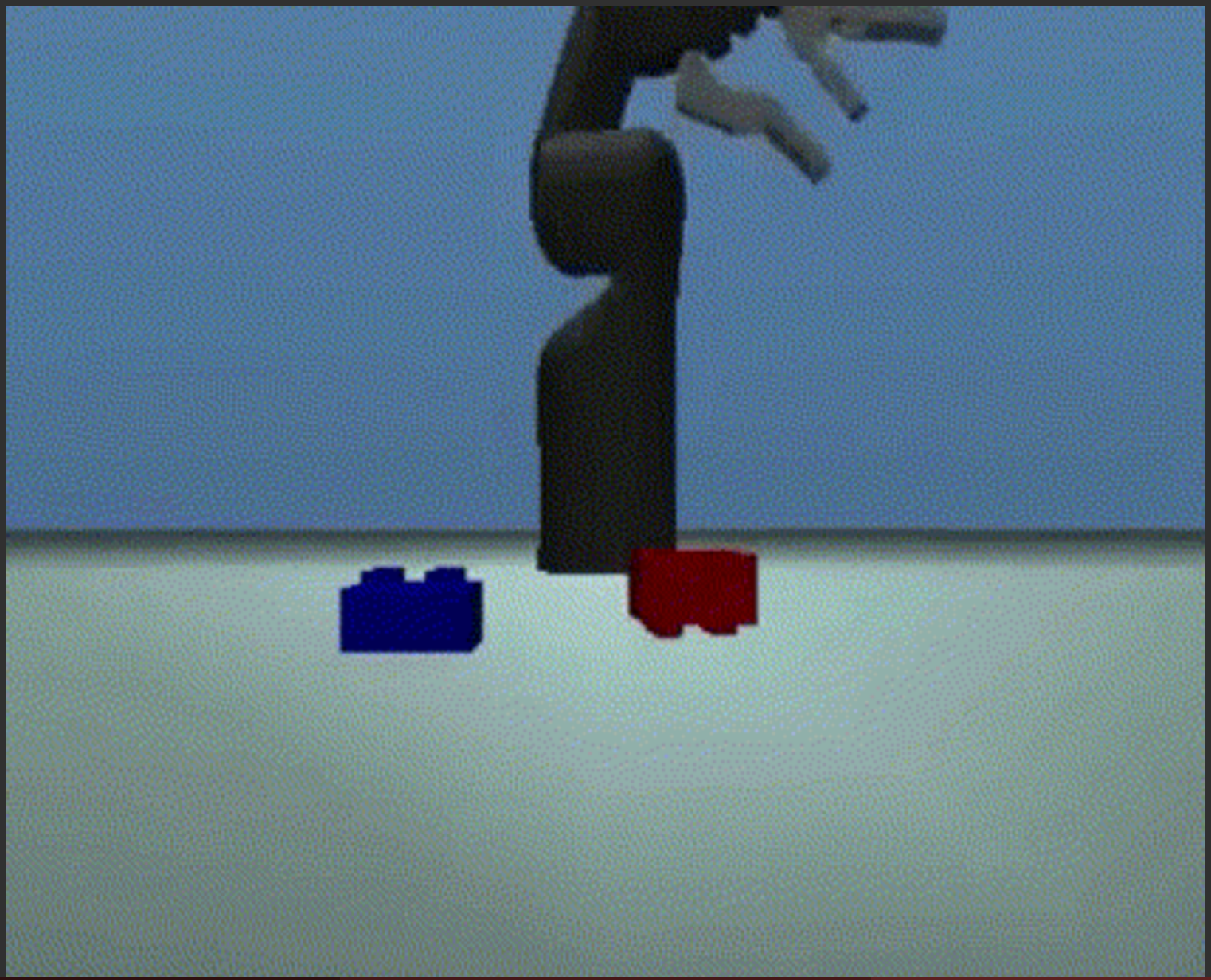
1. Learning
2. Perception
   - 2.1 image segmentation to avoid falling off a cliff
   - 2.2 object recognition to classify healthy and poisonous foods
3. Social Intelligence, interaction with other agents (game theory)
4. Language
5. Generalization, transferring solution from one problem to another (from paintings to photographs etc.)
6. General Intelligence (the ability to flexibly achieve a variety of goals in different contexts?)

Concerns

In a Lego stacking task, the desired outcome was for a red block to end up on top of a blue block. The agent was rewarded for the height of the bottom face of the red block when it is not touching the block.



Concerns



Smart Traffic Lights

About

The goal is to research RL methods in application to optimal traffic control problem.

Collaboration with Russian Academy of Sciences and Moscow Center of Road Traffic Optimization.

About

Technologies:

1. **S**imulation of **U**rban **Mo**bility - SUMO
2. FLOW, developed by University of California, Berkeley. Python library that allows one to integrate RL algorithms into SUMO
3. CityFlow, open source framework for large scale simulation

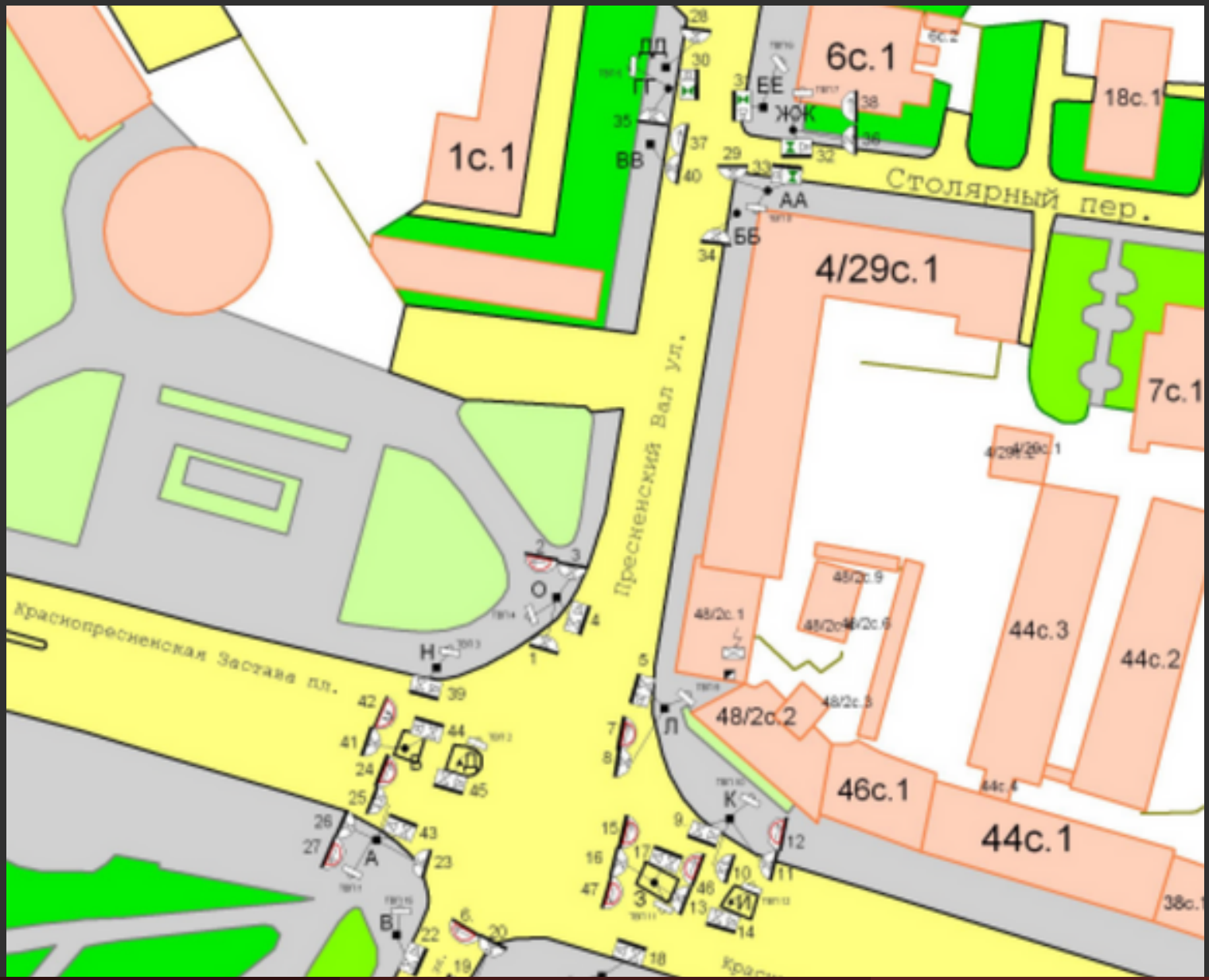
Relation to RL

1. Environment = SUMO
2. Agent = Traffic Light

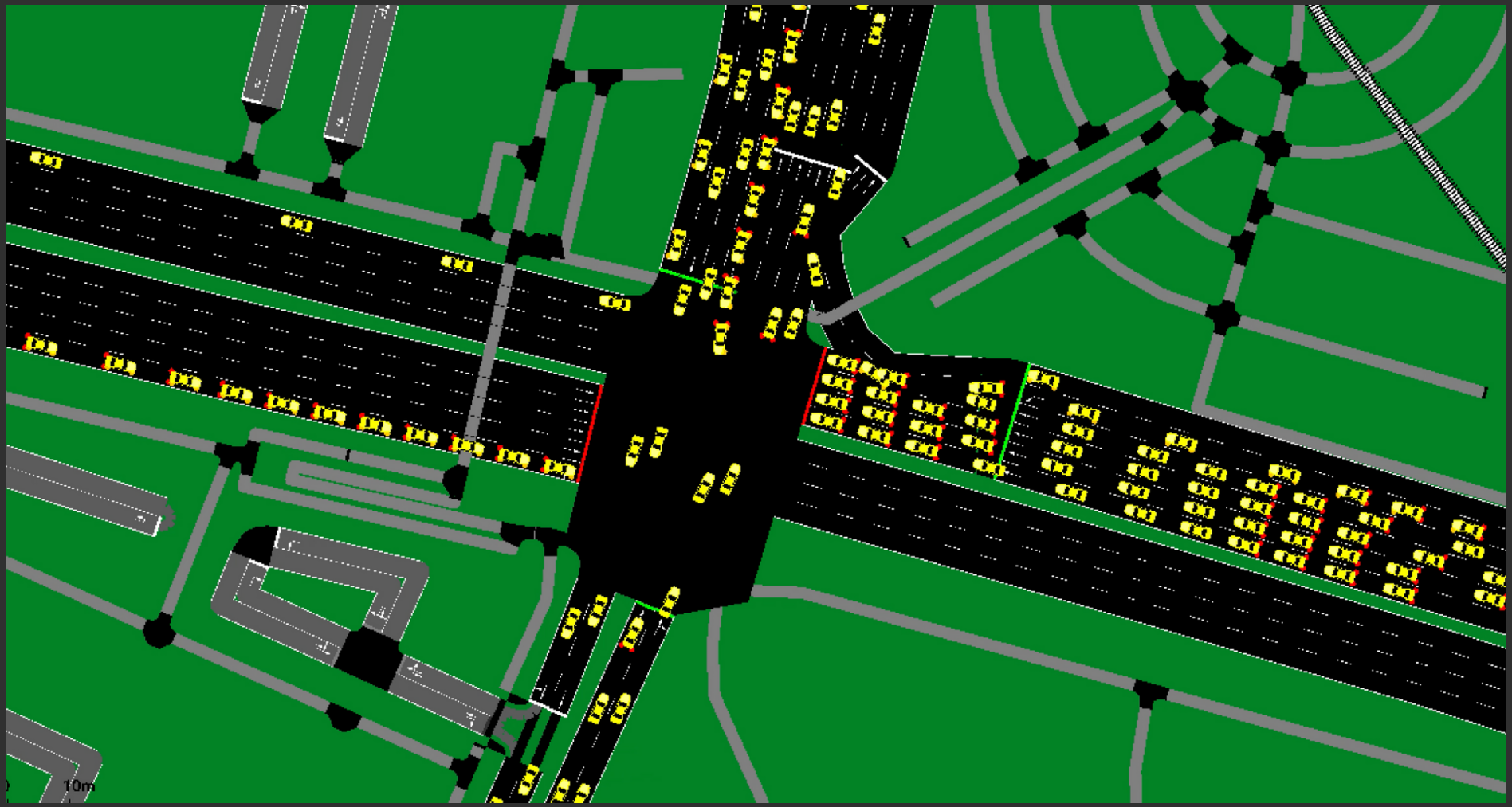
Progress

1. Literature overview
2. Setup, installation of libraries, searching for frameworks
3. Network creation and training
4. Data acquisition
5. Real intersection modeling

Simulation


1с.1
6с.1
18с.1
4/29с.1
7с.1
44с.3
44с.2
48/2с.2
46с.1
44с.1
Столярный пер.
Пресненский Вал ул.
Краснопресненская Застава пл.

Simulation



Video: https://www.youtube.com/watch?v=C8arR-oWw78

Useful Links

Resources

1. OpenAI, introduction to RL: https://spinningup.openai.com/en/latest/user/introduction.html
2. Reinforcement Learning: An Introduction, http://www.incompleteideas.net/book/the-book.html
3. Reward Is Enough, paper: https://doi.org/10.1016/j.artint.2021.103535

Thank you for your attention