Constrained Control of Multi-Vehicle Systems for Smart Cities and Industry 4.0: from Model Predictive Control to Reinforcement Learning
Part 3 - Reinforcement Learning Review

IEEE Intelligent Vehicles Symposium
Anchorage, Alaska, USA - June 4-7 2023

UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI INGEGNERIA MECCANICA, ENERGETICA E GESTIONALE
DIMEG

Dr. Giuseppe Franzè, June 04 2023

Outline

Motivations

- Recent advances in vehicular networking, communication and computing technologies have facilitated the practical deployment of **autonomous vehicles**
- The increasing number of vehicles requires innovative solutions to deal with **road traffic** issues
- Private mobility within urban road networks is almost always **unsustainable**
- Optimizing routing decisions has a positively impact on **traffic congestion** phenomena
- Future **smart cities** should refer to autonomous mobility systems that may offer a new way to provide equivalent service capabilities at possibly low congestion levels

Aims

Challenging issues -

- Capability to efficiently **take or modify** routing decisions during the on-line operations
- Definition of control architectures in charge to **couple** nominal paths (sequences of routing decisions) with the real dynamics of autonomous vehicles

Objective -

Develop a distributed framework to enjoy

- scalability
- flexibility

by jointly exploiting

1. **reinforcement learning** ideas
2. **model predictive control** philosophy

Reinforcement learning overview

Historical flow -

- Automation of repeated physical solutions

1750-1940 Industrial revolution and Machine Age

- Automation of repeated mental solutions

1950- Digital revolution and information age

- Allow machines to find solutions themselves

1960 - Artificial Intelligence

- It only needs to specify a problem and/or goal

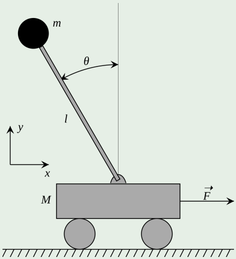
1980 - This requires learning autonomously how to make decisions

What is reinforcement learning?

- People and animals learn by interacting with the environment
- Differ from other classes of learning algorithms
  - Interactions are often sequential - future interactions can depend on earlier ones
- Goal-directed
- Learn without resorting to optimal behaviors
- Optimize some reward signal

Benchmark examples

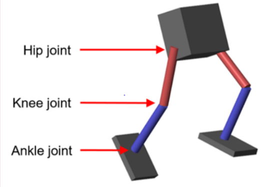
Cart-Pole problem -



- **Objective:** balance a pole on top of a movable cart
- **Measurements:** angle, angular speed, position, horizontal velocity
- **Action:** horizontal force applied on the cart
- **Reward:** good if the pole is upright

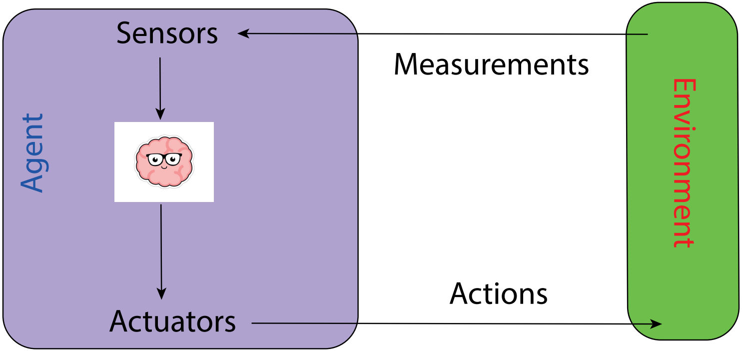
Walking robot -

- **Objective:** make the robot move forward
- **Measurements:** angle and position of the joints
- **Action:** torques applied on joints
- **Reward:** good if upright and forward movement



Definition -

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators



Rational agent -

For each possible measurement sequence, a rational agent should select an action that is expected to maximize a performance criterion based on its built-in knowledge about the environment

Utility function -

A performance criterion is an objective index for evaluating success or failure of an agent's behavior

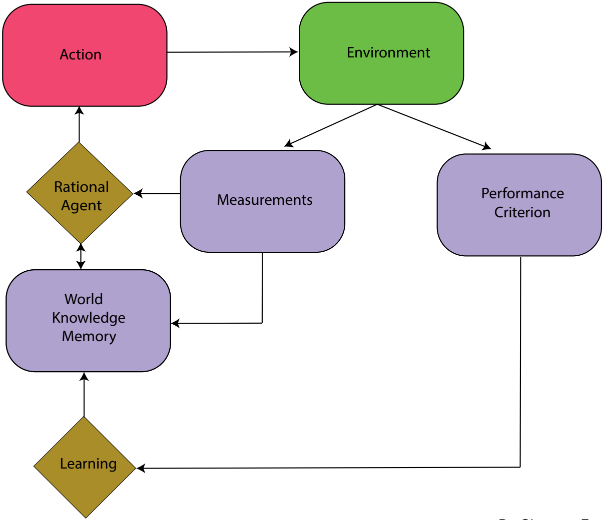
Remarks -

- rational $\neq$ omniscient: measurements may not supply all relevant information
- rational $\neq$ clairvoyant: action outcomes may not be as expected
- rational $\neq$ successful

Rational $\Rightarrow$ exploration, learning and autonomy

Learning -

An agent is learning if it improves its performance after making observations about the world



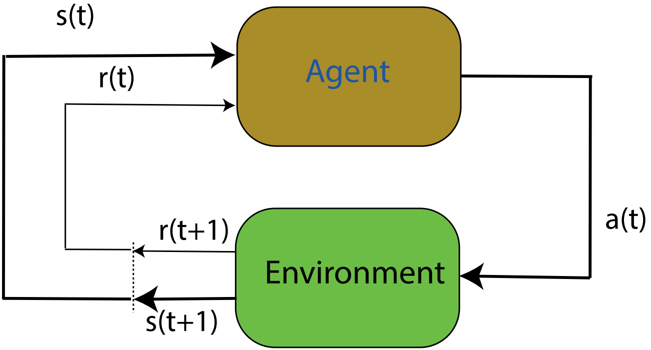
What is Reinforcement Learning?

Idea -

Capability to learn from experience to make good decisions under uncertainty

Ingredients -

- State space $\mathcal{S}$ — states $s \in \mathcal{S}$ (discrete or continuous)
- Action space $\mathcal{A}$ — actions $a \in \mathcal{A}$ (discrete or continuous)
- Reward function — $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$



Environment customization -

1. Observable
2. Stochastic process
3. Markovian transition model:

$$Pr(s(t+1)|s(t), a(t), s(t-1), a(t-1), \ldots, s(0), a(0)) = Pr(s(t+1)|s(t), a(t))$$

Markov decision process -

A Markov decision process is a tuple $\mathcal{M} = \{\mathcal{S}, s(0), \mathcal{A}, \delta\}$ where

- $s(0)$ is the initial state;
- $\delta : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a probabilistic transition function.

Reinforcement Learning functions

Return G -

Starting from the current state $s(t)$, it is the weighted accumulated future rewards

Value function V -

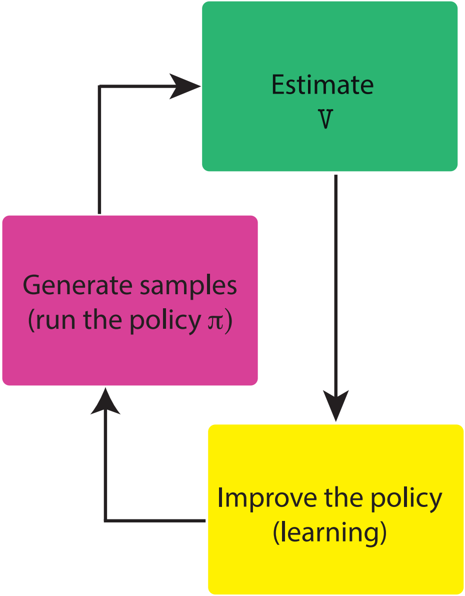
Starting from the current state $s(t)$, it is the total amount of reward an agent can expect to accumulate over the future (expected G)

Policy π -

$$\pi : \mathcal{S} \rightarrow \mathcal{A}$$

It is the core of the framework: it alone is sufficient to determine the agent behavior

Anatomy of the reinforcement learning algorithm



Reinforcement learning algorithms (1/2)

Types of algorithms -

- Policy gradients: directly differentiate the expected return
- Value based: estimate the value function of the optimal policy
- Actor critic: estimate value function of the current policy
- Model based: estimate the transition model

Episode -

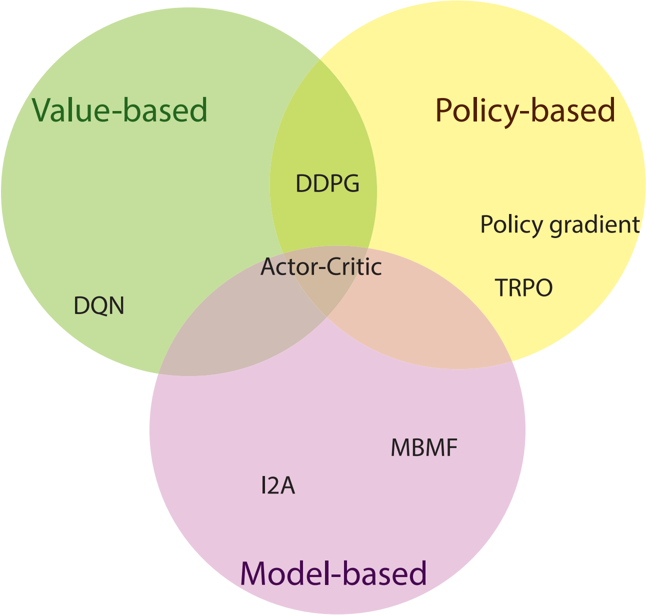
Let $s(t_{start})$ and $s(t_{end})$ be two given environment states. An episode ep is defined as:

$$ep := [t_{start} \ \ t_{end}]$$

Reset function -

Re-initialize the environment for successive episodes

Reinforcement learning algorithms (2/2)



Definition -

- It is a value based reinforcement learning algorithm for agents in Markovian domains
- It is an incremental method for dynamic programming with limited computational demands
- It successively improves the evaluation of the quality of particular actions at specific states
- It finds an optimal policy by maximizing the expected value of the total reward over the future

State-action value function

Q-function : $$Q(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t r(t+1)\right]$$

- $E[\cdot]$: the expected value operator
- $\gamma \in (0,1)$: the discount factor

Q-learning policy

Greedy action -

Maximize the Q-function:

$$a^* := arg \max_{a \in \mathcal{A}_p(s)} Q(s, a)$$

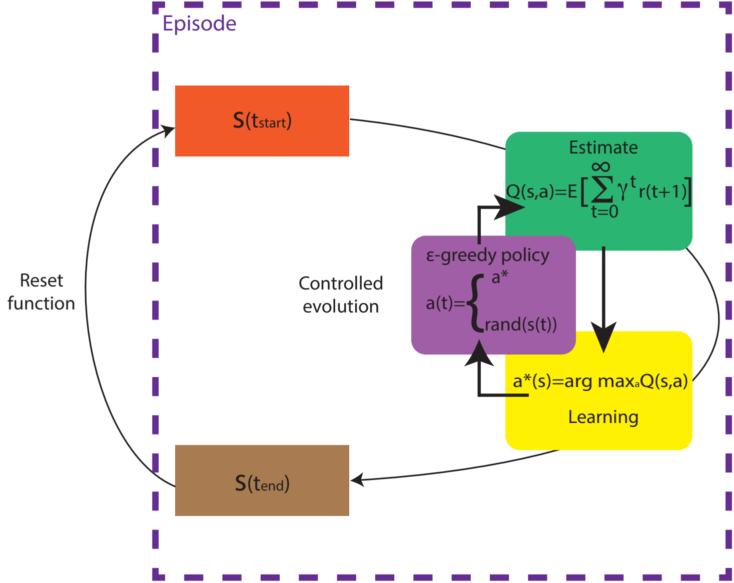
where $\mathcal{A}_p(s) \subseteq \mathcal{A}$ is the set of possible actions on $s \in \mathcal{S}$

$\epsilon-$greedy policy -

$$a(t) = \begin{cases} \text{rand}(\mathcal{A}_p(s)), & \text{probability } \epsilon(t) \\ a^*, & \text{otherwise} \end{cases}$$

where $\epsilon(t) \in (0, 1)$ is a monotonically decreasing function of time

Q-learning loop



Deep Q-Learning (DQL)

Definition -

Deep Q-learning belongs to the class of Q-learning algorithms and makes use of a deep neural network to approximate the state action value function

Deep Q-function -

A deep neural network in charge to compute the optimal value of $Q(s,a)$, namely $Q^*(s,a)$, is a Deep Q-network $Q(s,a;\theta)$ with θ the neural network weights

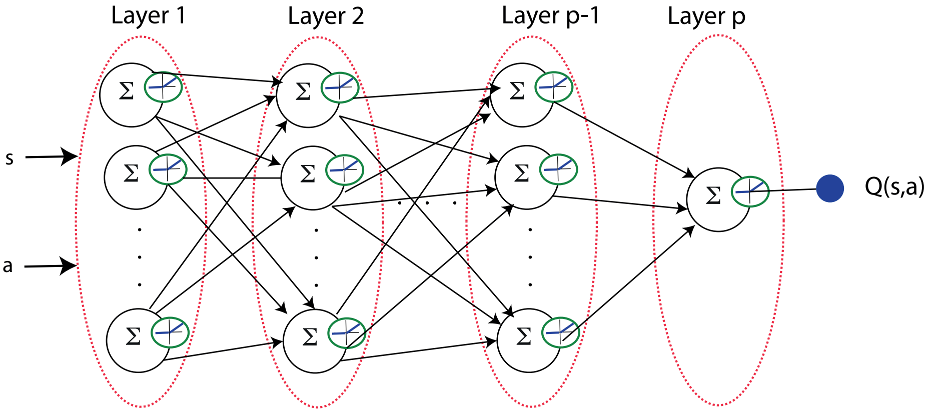
Aim-

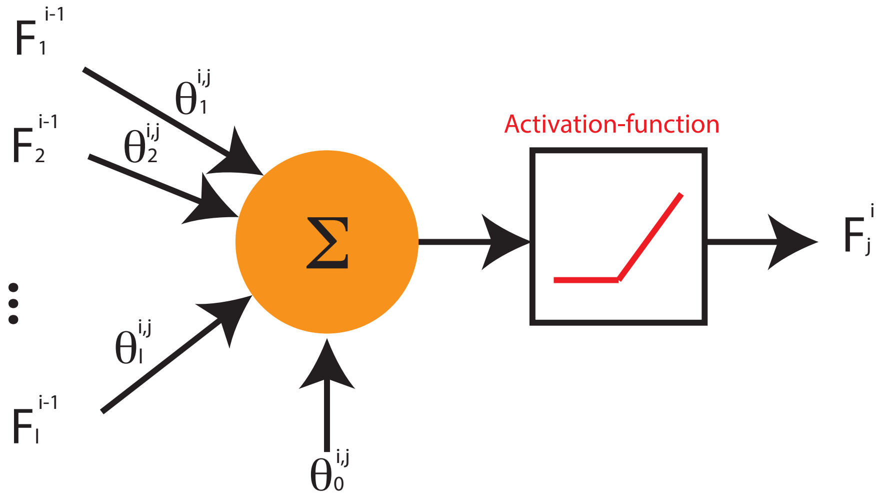
$$\theta^* = arg\min_{\theta} |Q^*(s,a) - Q(s,a;\theta)| \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}_p(s)$$

Drawback -

The optimal value $Q^*(s,a)$ is not available

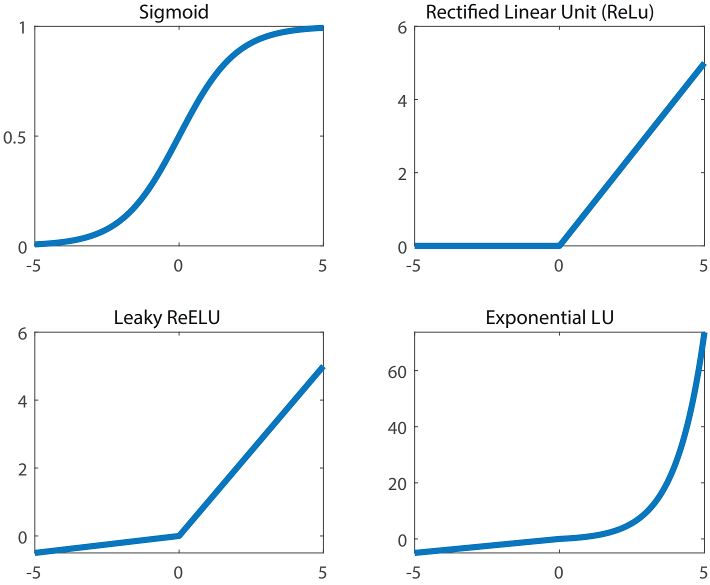
Fully connected feed-forward Deep Q-Network





F_1^{i-1}
F_2^{i-1}
F_l^{i-1}
$\theta_1^{i,j}$
$\theta_2^{i,j}$
$\theta_l^{i,j}$
Σ
$\theta_0^{i,j}$
Activation-function
F_j^{i}

Activation functions



Neural network training

Definition -

It is the process of teaching a neural network to perform a task

Procedure -

1. For each tupla $\{s(t), a(t), r(t+1), s(t+1)\}$ compute the expected reward:

$$\hat{r}(t+1;\theta) = Q(s(t), a(t);\theta) - \gamma \max_{a \in \mathcal{A}_p(s(t+1))} Q(s(t+1), a;\theta)$$

2. Evaluate the loss function:

$$\mathcal{L}(t+1;\theta) := (r(t+1) - \hat{r}(t+1;\theta))^2$$

3. Update the weights vector θ :

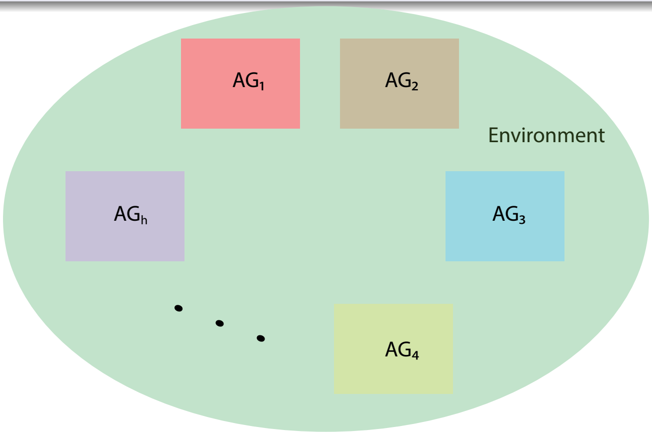
$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$$

with $\alpha \in (0,1)$ the learning rate

Multi-agent RL model

Multi-agent system -

A multi-agent system is a group of interacting rational agents sharing a common environment



Stochastic game -

It is the generalization of the Markov decision process to the multi-agent scenario

$$< V, \mathcal{S}, \mathcal{A}, \{r_1, \ldots, r_h\} >, \;\; V = \{AG_1, \ldots, AG_h\}$$

Reward functions

Global reward -

$$\Phi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$$

It is computed by the environment and shared in broadcast with all the agents $AG_i,\ i = 1, \ldots, h$

Local reward -

It exploits the information coming from environment and neighbors

$$r_i \triangleq \Phi + \phi_i$$

- $\phi_i : \mathcal{O}_i \times \mathcal{A}_i \times \mathcal{O}_i \to \mathbb{R}$ a heuristic function accounting for the task of the agent AG_i
- $\mathcal{O}_i$ the observation space

Multi-agent Deep Q-learning Algorithm

Ingredients -

- deep Q-networks: $Q_i(s, a; \theta_i)$, $i = 1, \ldots, h$
- greedy actions: $a_i^*(t) := \arg\max_{a \in \mathcal{A}_i} Q_i(o(t), a; \theta_i)$, with $o(t) \in \mathcal{O}_i$
- ϵ-greedy policy:

$$a_i(t) = \begin{cases} \text{rand}(\mathcal{A}_{p_i}(s)), & \text{probability } \epsilon(t) \\ a_i^*(t), & \text{otherwise} \end{cases}$$

- Loss function: $\mathcal{L}(t+1; \theta_i) = (r_i(t+1) - \hat{r}_i(t+1; \theta_i))^2$

References

- **Intelligent agents-**
  - S. Russell and P. Norvig,“Artificial intelligence a modern approach,” *Pearson Education, Inc.*, 2021.
  - M. Wooldridge,“An Introduction to Multi-Agent Systems,” *John Wiley and Sons*, 2002.
- **Reinforcement learning-**
  - R. Sutton and A. Barto, “Reinforcement learning: An introduction,” *MIT press*, 2018.
  - K. Leslie Pack, M. Littman and A. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research,* Pp. 237-285, 1996.
- **Q-learning-**
  - J. Fan *et al.*, “A theoretical analysis of deep Q-learning,” *Learning for Dynamics and Control*, 2020.
  - T. Hester *et al.*, “Deep q-learning from demonstrations,” *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, No. 1, 2018.
- **Neural networks-**
  - A. Martin, P. Bartlett and P. Bartlett, “Neural network learning: Theoretical foundations,” *Cambridge university press*, 1999.
  - K. Murphy, “Probabilistic machine learning: an introduction,” *MIT press*, 2022.
- **Multi-agent deep Q-learning-**
  - B. Lucian, R. Babuska, and B. De Schutter, “Multi-agent reinforcement learning: An overview,” *Innovations in multi-agent systems and applications*, Pp. 183-221, 2010.
  - R. Wang*et al.*, “Multi-agent reinforcement learning for edge information sharing in vehicular networks,” *Digital Communications and Networks*, Vol. 8, No. 3, Pp. 267-277, 2022.

THANKS FOR THE ATTENTION!!