Planning and Decision Making for Autonomous Robots

Lecture 11: Deep Reinforcement Learning

Lorenzo Fiaschi

Institute for Dynamic Systems and Control
D-MAVT, ETH Zürich
lfiaschi@ethz.ch

October 13, 2026

Limitations of Dynamic Programming

So far in this course:

- **Assume full knowledge** of the state, action, transition model, and reward/cost function
- Do **not** learn from past data/experiences

Limitations of Dynamic Programming

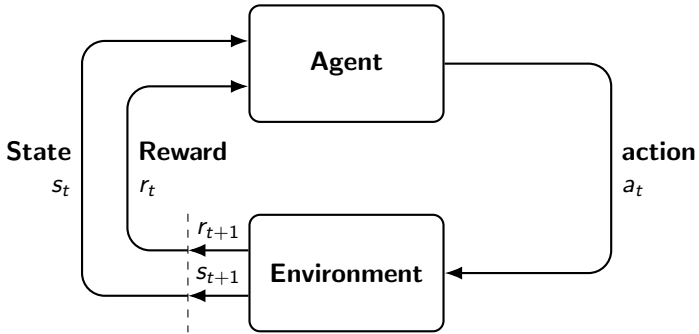
So far in this course:

- **Assume full knowledge** of the state, action, transition model, and reward/cost function
- Do **not** learn from past data/experiences

yet, most of the **modern interesting applications** come with:

- **Unknown/Unstructured** environments in which is hard to model transitions and cost function
- **Computational intractability** of large/continuous state spaces
- **Abundance of data** that can be used to learn from past experiences
- **Multi-agent setups** in which the environment is not stationary

RL paradigm



Context

On-policy vs Off-policy

- On-policy: learn about π **sampling** from π's **experience**
- Off-policy: learn about π from **another** policy μ

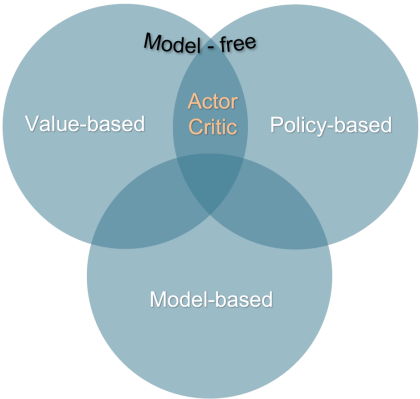
Why off-policy?

- Learn from **imitation**
- **Re-use experience** generated by old policies
- **Learn** the optimal policy while **following** an **exploratory** one
- **Learn multiple** policies while following only one

Lemma (Importance Sampling)

$$\mathbb{E}_{x\sim P}[f(x)] = \sum_x P(x)f(x) = \sum_x Q(x)\frac{P(x)}{Q(x)}f(x) = \mathbb{E}_{x\sim Q}\left[\frac{P(x)}{Q(x)}f(x)\right]$$

Model-free vs Model-based



- **Value-based**
  learn the optimal value function V^*, Q^*
- **Policy-based**
  learn the optimal policy π^*
- **Model-based**
  learn the transition model $P(s'|s,a)$ and reward function $R(s,a,s')$, then plan.

Learning from samples

Exploration vs exploitation tradeoff

- Learning / optimizing from empirical **transitions**
- $\langle s, a, r, s' \rangle$ samples
- Generalized policy iteration: **GPI**
  - $\pi_k \rightarrow Q_{\pi_k}$
  - $Q_{\pi_k} \rightarrow \pi_{k+1}$
  - $k \leftarrow k + 1$ and repeat
- Policy update: $\pi_{k+1}(a|s) = \arg\max_{a \in A} Q_{\pi_k}(s, a)$
- Issue: π_1 is **deterministic**
- **Explotation** is crucial

ε-greedy policy

- ε probability to explore
- 1-ε probability to exploit

$$\pi_k(a|s) = \begin{cases} \frac{\varepsilon_k}{m} + (1 - \varepsilon_k) & \text{if } a = \arg\max_{a \in A} Q_{\pi_k}(s, a) \\ \frac{\varepsilon_k}{m} & \text{otherwise} \end{cases}$$

- m is the number of actions at state s

Proposition (ε-greedy policy improvement)

Let π_{k+1} be the ε-greedy updated policy from π_k. Then, $V_{\pi_{k+1}} \geq V_{\pi_k}$ holds.

GLIE properties

- Is GPI **convergence** to optimality still **guaranteed**?
- Greedy in the Limit with Infinite Exploration (**GLIE**) property

Definition (GLIE)

An ε-greedy improvement of π has the GLIE property if and only if

1. $\lim_{k\to\infty} N_k(s,a) = \infty$ [each state-action pair is explored infinitely many times]
2. $\lim_{k\to\infty} \pi_k(a|s) = \mathbb{1}(a; \arg\max_{a'\in A} Q_k(s,a'))$ [policy converges to the greedy one]
3. $\varepsilon_k \in \mathcal{O}(\frac{1}{k})$ [ε converges to zero as $\frac{1}{k}$]

Deep Learning

Parametric function approximators

- **Tabular methods** have some **limitations**
- Difficulties with **non-trivial state spaces**
  - Large spaces
  - Continuous spaces
- **Lack of generalization** properties
  - Incapability to recognize "similar" states
  - Similar states must behave similarly
- **State encoding**
- **Function approximator**
  - Parametric functions:

$$\widehat{V}\colon S \times \Theta \to \mathbb{R}, \qquad \widehat{Q}\colon S \times \Theta \to \mathbb{R}^{|A|}, \qquad \widehat{\pi}\colon S \times \Theta \to A$$

  - Θ: parameter space

Experience replay buffer

- Issue 1: **Non-stationary data**
  - Policy modifies over time
  - The environment may change
- Issue 2: **Non-IID data**
  - Samples are necessarily correlated

Experience replay buffer

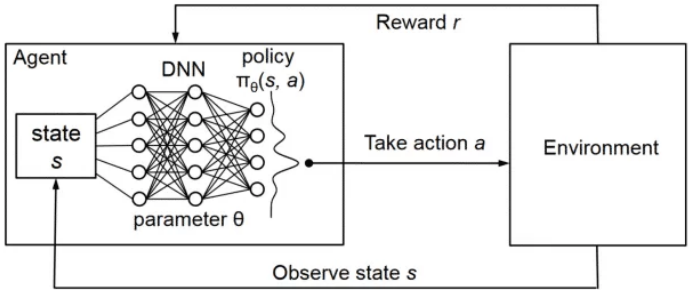
- Issue 1: **Non-stationary data**
  - Policy modifies over time
  - The environment may change
- Issue 2: **Non-IID data**
  - Samples are necessarily correlated

- **Experience replay buffer**
  - Collect agent experience in a dataset
  - Learn by stochastically sampling from it
  - Transitions repetition is allowed
  - Well-posed because data are not correlated
  - Stable due to batch learning
  - Random sampling, Prioritized sampling

Deep Q-learning

Deep Q-network

- **Deep Q-Network** (DQN)
  - DNN to approximate Q
  - Fixed Q-targets to stabilize the learning
  - Targets come from an old, fixed version of the network: θ^-



Deep Q-network

- **Deep Q-Network** (DQN)
  - DNN to approximate Q
  - Fixed Q-targets to stabilize the learning
  - Targets come from an old, fixed version of the network: θ^-

- **Simple DQN loss function**

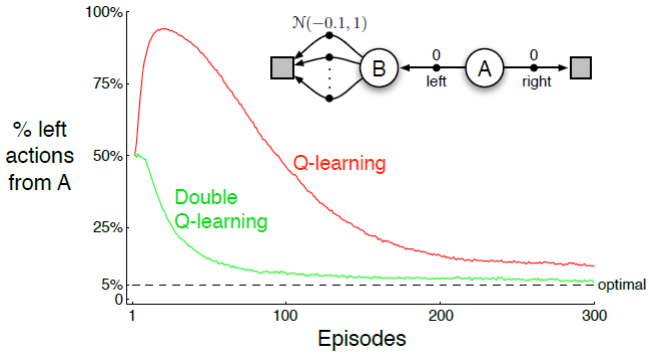
$$L(\theta) = \mathbb{E}_\pi[(r + \gamma \max_{a' \in A} \widehat{Q}_\pi(s', a', \theta^-) - \widehat{Q}_\pi(s, a, \theta))^2]$$

- **Double DQN loss function**

$$L(\theta) = \mathbb{E}_\pi[(r + \gamma \widehat{Q}_\pi(s', \arg\max_{a' \in A} \widehat{Q}_\pi(s', a', \theta), \theta^-) - \widehat{Q}_\pi(s, a, \theta))^2]$$

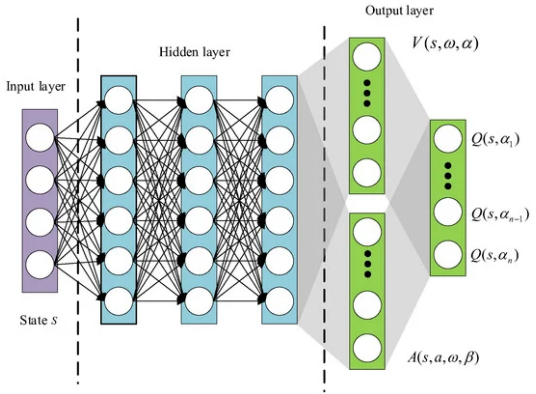
Maximization bias

- Use of the maximum estimate as the estimate of the maximum
- A is the starting point
- Suppose to experience something positive going left from A
- The learning is biased towards B



Duelling DQN

$$\widehat{Q}_\pi(s,a,\theta) = \widehat{V}_\pi(s,\theta_V) + \left(\widehat{A}_\pi(s,a,\theta_A) - \max_{a' \in A} \widehat{A}_\pi(s,a',\theta_A)\right)$$



Policy Gradient Methods

Policy Gradient Methods

- π^* approximation **converges faster** than V^* one
- Even if it gets more easily stuck in local optima
- Idea: **approximate** it **directly**
- Policy approximation function: $\widehat{\pi}(a|s, \theta)$
- **Previously**, π^* approximation was **implicit**
- This enables:
  - learning in **continuous** action spaces
  - learning of a **stochastic** policy
- Now, $Q\colon S \times A \times \Omega \to \mathbb{R}$

Policy gradient theorem

Theorem (Policy Gradient)

Let $\widehat{\pi} \in C^1(\Theta)$ *and* $J(\theta) \propto \mathbb{E}_{\widehat{\pi}}[V_{\widehat{\pi}}(s)]$. *Then,*

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\widehat{\pi}} \left[\nabla_\theta \ln \widehat{\pi}(A|S,\theta) Q_{\widehat{\pi}}(S,A) \mid d_{\widehat{\pi}}(S,A) \right]$$

- Empirical estimation:

$$\nabla_\theta J(\theta) \propto \mathbb{E} \left[\sum_{e=1}^{E} \sum_{t=0}^{T_e} \nabla_\theta \log \widehat{\pi}(a_t^e|s_t^e,\theta) G_t^e \right]$$

- Unbiased but high variance estimator
- To reduce it, use a state-only-dependent baseline $b(s)$:

$$\nabla_\theta J(\theta) \propto \mathbb{E} \left[\sum_{e=1}^{E} \sum_{t=0}^{T_e} \nabla_\theta \log \widehat{\pi}(a_t^e|s_t^e,\theta) \left(G_t^e - b(s_t) \right) \right]$$

Actor-Critic policy gradient

- To further reduce the variance we can learn $\widehat{Q}$ as well
- Two entities now
  - **Actor**: $\widehat{\pi}$ generates experience and updating directions
  - **Critic**: $\widehat{Q}$ evaluates the quality of the actions taken
- Does the policy gradient theorem still hold?

Theorem (Compatible function approximation)

In case of actor-critic approach, if

1. $\nabla_\omega \widehat{Q}(s, a, \omega) = \nabla_\theta \ln \widehat{\pi}(a|s, \theta)$ *[compatibility]*
2. $\mathbb{E}_{\widehat{\pi}}[(Q_{\widehat{\pi}} - \widehat{Q})^2] \xrightarrow{t \to \infty} 0$ *[unbiased property]*

then the Policy Gradient Theorem still holds true.

- Thus,

$$\widehat{Q}(s, a, \omega) = \nabla_\theta \ln \widehat{\pi}(a|s, \theta)^T \omega$$

Natural policy gradient

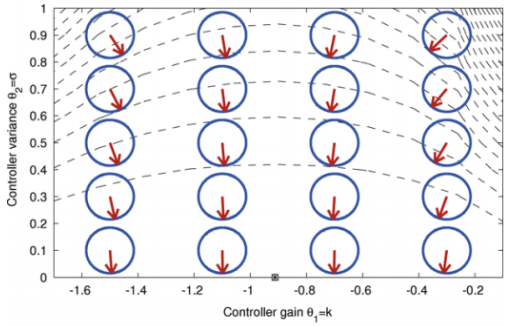
- Policy space is ill-shaped w.r.t. the parameters: **plateaus** and **discontinuities** are frequent
- **Slow** and **unstable** convergence
- Solution: make **changes** in θ proportional to **policy variation**
- Possible metric: **KL-divergence** between $\widehat{\pi}(\theta_t)$ and $\widehat{\pi}(\theta_{t+1})$

$$KL(\widehat{\pi}(\theta_t) \,||\, \widehat{\pi}(\theta_{t+1})) = \int_{S \times A} \widehat{\pi}(a|s, \theta_t) \ln \left(\frac{\widehat{\pi}(a|s, \theta_t)}{\widehat{\pi}(a|s, \theta_{t+1})} \right) \mathrm{d}(s, a)$$

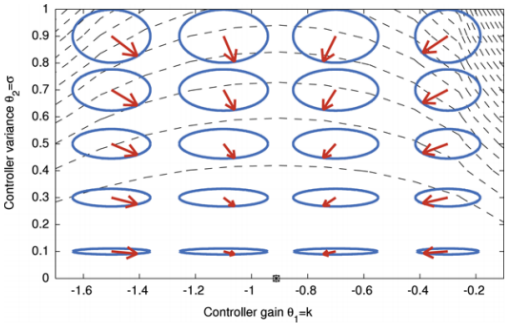
- **Problem reformulation**

$$\begin{aligned} \max_{\theta_t} \quad & J(\theta_t) \\ \text{s.t.} \quad & KL(\widehat{\pi}(\theta_t) || \, \widehat{\pi}(\theta_{t+1}) \leq \varepsilon \end{aligned}$$

Natural policy gradient



(a) Vanilla policy gradient.



(b) Natural policy gradient.

Theorem (Natural policy gradient)

Let $\widehat{\pi} \in C^1(\Theta)$, $J(\theta) \propto \mathbb{E}_{\widehat{\pi}}[V_{\widehat{\pi}}(s)]$, *and* $\widehat{Q}_\omega$ *compatible with* $\widehat{\pi}_\theta$. *Then,*

$$\nabla_\theta J(\theta) \propto \omega$$

Deterministic policy gradient

Theorem (Deterministic Policy Gradient Theorem)

Let $\widehat{\pi} \in C^1(\Theta)$, $J(\theta) \propto \mathbb{E}_{\widehat{\pi}}[V_{\widehat{\pi}}(s)]$, *then*

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\pi_\theta}[\nabla_\theta \pi_\theta(s) \nabla_a Q(s,a)|_{a=\pi_\theta(s)} \,|\, s \sim d(S)]$$

Theorem (Deterministic compatible function approximation)

Let $Q \in C^1(A)$. *Then, the Deterministic Policy Gradient Theorem continues to hold if*

- $\nabla_a \widehat{Q}(s,a,\omega)|_{a=\pi_\theta(s)} = \nabla_\theta \widehat{\pi}(s,\theta)^T \omega$
- $\mathbb{E}_{\widehat{\pi}}[(\nabla_a \widehat{Q}(s,a,\omega) - \nabla_a Q(s,a)|_{a=\pi_\theta(s)})^2] \xrightarrow{t\to\infty} 0$

- Thus,

$$\widehat{Q}(s,a,\omega) = a^T \nabla_\theta \widehat{\pi}(s,\theta)^T \omega$$

Proximal Policy Optimization (PPO)

- Alternative way to create a trustregion for the policy gradient
- Naiver but effective

$$J(\theta_k) = \mathbb{E}_t \left[\min \left(\frac{\pi_{\theta_k}(a_t|s_t)}{\pi_{\theta_{k-1}}(a_t|s_t)}, \, g(\varepsilon, A_t) \right) A_t \right]$$

where

$$g(\varepsilon, A) \coloneqq \begin{cases} (1+\varepsilon) & \text{if} \quad A > 0 \\ (1-\varepsilon) & \text{otherwise} \end{cases}$$

PPO in the gokart simulator

From theory to practice video

Credits to A. Zanardi

Final considerations

RL caveats

- **Hyperparameter tuning:** The learning is very sensitive to the choice of the hyperparameters such as network architecture, learning rate, etc.
- **Generalization:** Sim2Real gap + never seen scenarios.
- **Safety:** no guarantees a priori, need to be induced somehow.

A glimpse beyond

- Soft actor-critic
- Model-based RL
- Hierarchical RL
- Distributional RL
- Inverse RL
- Imitation learning
- Constraints learning
- RL from human feedbacks

References

[1] Sutton R., Barto A. G., "Reinforcement Learning: an introduction"

[2] Wang Z. et al. , Dueling Network Architectures for Deep Reinforcement Learning 2016

[3] Silver D. et al., Deterministic Policy Gradient Algorithms, 2014

[4] Peters J., Schaal S., Reinforcement learning of motor skills with policy gradients, 2008

[5] Schulman et al., Proximal Policy Optimization Algorithms, 2017

[6] Ghasemi M. et al., Comprehensive Survey of Reinforcement Learning: From Algorithms to Practical Challenges, 2025

[7] OpenAI Spinning Up

[8] Foundations of Reinforcement Learning by Prof. Niao He [ETHz course]

[9] Feel free to ask me for more references!