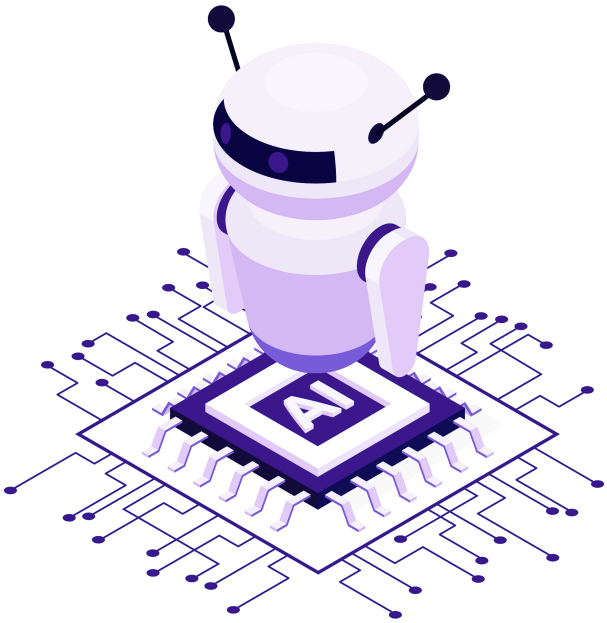

Deep Reinforcement Learning

Professor Mohammad Hossein Rohban

Homework 12:

Offline Methods

Designed By:

Amirabbas Afzali

amir8afzali@gmail.com

Hesam Hosseini

hesam8hosseini@gmail.com

RIML

Spring 2025

Preface

Offline Reinforcement Learning (Offline RL) aims to learn effective policies solely from a fixed dataset of past interactions with the environment, without any additional online data collection. Unlike traditional online RL, where agents learn through active exploration, Offline RL restricts access to the environment during training, making it suitable for scenarios where interaction is costly, risky, or otherwise impractical.

Applications of Offline RL are widespread and critical in domains such as:

- **Healthcare:** Training diagnostic or treatment policies from historical patient data without risking real patients.
- **Autonomous Driving:** Learning driving policies from logged human driving data to avoid dangerous exploration.
- **Robotics:** Improving robot control policies using prior demonstration data, minimizing hardware wear and avoiding unsafe states.
- **Recommendation Systems:** Optimizing recommendation policies based on past user interaction logs without real-time experimentation.
- **Industrial Systems:** Enhancing control strategies in manufacturing or energy systems where online exploration could be costly or disruptive.

Offline RL addresses the unique challenge of learning without new environment interactions, requiring algorithms to be robust against distributional shifts and extrapolation errors due to limited coverage of the state-action space.

Online vs Offline RL:

Online RL In the *Online RL* setting, an agent with policy $\pi \in \Pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$ learns through trial and error. It actively interacts with the environment — including both transition dynamics $\mathcal{T}$ and the reward function $\mathcal{R}$.

At each time step t, an agent observes a state s_t from the environment and selects an action $a_t \sim \pi$. Upon taking the action, the agent receives a reward r_t and transitions to a new state s_{t+1}. The agent's objective is to maximize its expected return:

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{a_t \sim \pi, s_{t+1} \sim \mathcal{T}, s_0 \sim \rho_0} \left[\sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t) \right], \tag{17}$$

We can alternatively denote the trajectory generated by a policy π to be $\tau = \{s_0, a_0 \sim \pi(a_0|s_0), s_1 \sim \mathcal{T}(s_1|s_0, a_0), a_1 \sim \pi(a_1|s_1), \ldots\}$ and denote the trajectory distribution of π as:

$$p_\pi(\tau) = \rho_0 \prod_{t=0}^{T} \pi(a_t|s_t)\mathcal{T}(s_{t+1}|s_t, a_t), \tag{18}$$

where T denotes the length of decision sequences. The learning objective can be expressed as:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=0}^{T} \gamma^t \mathcal{R}(s_t, a_t) \right]. \tag{19}$$

Offline RL In the *Offline RL* setting, interactions with the environment are **strictly forbidden**. The learning problem is no longer online learning but learning from a static dataset of decision logs $\mathcal{D}_{\text{offline}} = \{(s_t^i, a_t^i, s_{t+1}^i, r_t^i)\}$, that is generated by some **unknown behavior policy** π_β.

The most obvious difficulty in the offline RL setting is that such a setting **prohibits** exploration — hence it hinders the improvement of policy learning to be improved over the demonstration data.

Grading

The grading will be based on the following criteria, with a total of 160 points:

Task	**Points**
Task 1: CQL Theoretical	60
Task 2: Conservative Q-Learning	100
Clarity and Quality of Code	5
Clarity and Quality of Report	5

Submission

The deadline for this homework is 1404/04/13 (July 4th 2025) at 11:59 PM.

Please submit your work by following the instructions below:

- Place your solution alongside the Jupyter notebook(s).
  - Your written solution must be a single PDF file named `HW12_Solution.pdf`.
  - If there is more than one Jupyter notebook, put them in a folder named `Notebooks`.
- Zip all the files together with the following naming format:

  `DRL_HW12_[StudentNumber]_[FullName].zip`

  - Replace `[FullName]` and `[StudentNumber]` with your full name and student number, respectively. Your `[FullName]` must be in CamelCase with no spaces.
- Submit the zip file through Quera in the appropriate section.
- We provided this LaTeX template for writing your homework solution. There is a 5-point bonus for writing your solution in LaTeX using this template and including your LaTeX source code in your submission, named `HW12_Solution.zip`.
- If you have any questions about this homework, please ask them in the Homework section of our Telegram Group.
- If you are using any references to write your answers, consulting anyone, or using AI, please mention them in the appropriate section. In general, you must adhere to all the rules mentioned here and here by registering for this course.

Keep up the great work and best of luck with your submission!

Contents

1 Part 1: CQL Theoretical

1. Considering the Bellman update, explain with reasoning why value estimation suffers from overestimation in the offline framework.

$$Q(s,a) \leftarrow r(s,a) + \mathbb{E}_{a' \sim r_{new}}[Q(s',a')]$$

2. One of the solutions to address the overestimation problem in the offline framework is CQL, whose objective function for computing the value is given below. Explain the role of each of the four terms in this objective function.

$$\begin{aligned}\hat{Q}^T = \arg\min_Q \max_\mu & \alpha \mathbb{E}_{s\sim D, a\sim\mu(a|s)}[Q(s,a)] \\ & - \alpha \mathbb{E}_{(s,a)\sim D}[Q(s,a)] \\ & - \mathbb{E}_{s\sim D}[\mathcal{H}(\mu(\cdot|s))] \\ & + \mathbb{E}_{(s,a,s')\sim D}\left[(Q(s,a) - (r(s,a) + \mathbb{E}[Q(s',a')]))^2\right]\end{aligned}$$

3. Rewrite the optimization problem from part 3 as a minimization-only problem.
4. To apply this method in model-based reinforcement learning, what changes are needed in the objective function? Rewrite the new objective function.

2 Part 2: Conservative Q-Learning

2.1 Introduction

In this assignment, we explore Offline Reinforcement Learning (Offline RL), where agents must learn solely from a fixed dataset without further interaction with the environment. Unlike Online RL, Offline RL is crucial when real-time exploration is expensive, risky, or impractical.

This assignment focuses on Conservative Q-Learning (CQL)[2], a robust offline RL algorithm designed to mitigate overestimation errors and improve policy learning from static datasets.

2.2 Theoretical Insights

Conservative Q-Learning (CQL) addresses the overestimation issue in offline RL by learning a Q-function that conservatively estimates value functions, ensuring the expected return under the learned policy is a lower bound of the true return.

2.2.1 Preliminaries

Notations:

- $T(s'|s,a)$: Transition probability.
- $\mathcal{D}$: Dataset collected from the behavior policy π_β.
- $r(s,a)$: Reward function, γ: Discount factor.

- $\hat{\pi}_\beta(a|s)$: Empirical behavior policy from dataset counts.
- $d^{\pi_\beta}(s)$: Discounted marginal state distribution.
- $\mu(s,a)$: Generic distribution over state-action pairs.

Assumptions:

- Rewards are bounded: $|r(s,a)| \leq R_{\max}$.

2.2.2 Standard Q-Learning and Its Limitations

In standard Q-learning, the Bellman operator is defined as:

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

where the update targets maximizing the expected return. In practice, offline RL faces challenges due to distribution shifts between the behavior policy and the learned policy, leading to overestimation for out-of-distribution (OOD) actions.

2.2.3 Conservative Q-Learning (CQL)

CQL mitigates overestimation by penalizing Q-values for unseen or unlikely actions. The key idea is to add a conservative penalty during Q-learning updates:

$$\hat{Q}^{k+1} \leftarrow \arg\min_Q \alpha \mathbb{E}_{s,a\sim\mu}[Q(s,a)] + \frac{1}{2}\mathbb{E}_{s,a,s'}\left[\left(Q(s,a) - \hat{\mathcal{B}}^\pi \hat{Q}^k(s,a)\right)^2\right].$$

This formulation ensures that the learned Q-values remain pessimistic, avoiding the overestimation of OOD actions.

Further improvements can be made by introducing an additional term to promote higher Q-values for dataset actions:

$$\hat{Q}^{k+1} \leftarrow \arg\min_Q \alpha\left(\mathbb{E}_{s,a\sim\mu}[Q(s,a)] - \mathbb{E}_{s,a\sim\hat{\pi}_\beta}[Q(s,a)]\right) + \frac{1}{2}\mathbb{E}_{s,a,s'}\left[\left(Q(s,a) - \hat{\mathcal{B}}^\pi \hat{Q}^k(s,a)\right)^2\right].$$

2.2.4 CQL Objective Variants

CQL can be further generalized by defining different regularizers $\mathcal{R}(\mu)$. One popular choice is maximum entropy regularization, leading to the CQL($\mathcal{H}$) variant:

$$\text{CQL}(\mathcal{H}) = \min_Q \alpha \mathbb{E}_s\left[\log \sum_a \exp(Q(s,a)) - \mathbb{E}_{a\sim\hat{\pi}_\beta}[Q(s,a)]\right] + \frac{1}{2}\mathbb{E}_{s,a,s'}\left[\left(Q(s,a) - \hat{\mathcal{B}}^\pi \hat{Q}^k(s,a)\right)^2\right].$$

Other variants, such as using a KL-divergence regularizer, are also discussed in the original CQL paper.

2.3 Problem Setup & Implementation

A large offline dataset was generated using a random policy, simulating unstructured exploration. Additionally, a small number of new samples were collected via an SAC agent for evaluation purposes, helping to track the agent's ability to generalize beyond the dataset.

We trained a CQL-SAC agent on the offline data. The agent was evaluated on newly gathered states by measuring the average return, allowing us to monitor learning progress and performance improvement.

Further details of the assignment, including TODOs, evaluation metrics, and implementation details, are provided in the notebook.

2.4 Conclusion

In this assignment, we trained a SAC agent using a relatively large offline dataset collected from a random policy, supplemented by a limited number of online interactions gathered during training. These online samples were used both for training and for evaluating the agent's performance in newly observed states, highlighting improvements over time. While this hybrid setup enabled effective learning, it is important to note that in real-world offline RL scenarios, online interactions are often prohibited, making the problem significantly more challenging. In such cases, greater amounts of offline data and longer training periods are necessary. Through this process, we gained a deeper understanding of the intuition behind Conservative Q-Learning (CQL), its practical implementation, and why it is considered a stable and popular method for offline reinforcement learning.

References

[1] Cover image designed by freepik

[2] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. *Conservative Q-Learning for Offline Reinforcement Learning*. arXiv preprint arXiv:2006.04779, 2020. Available at: https://arxiv.org/abs/2006.04779.