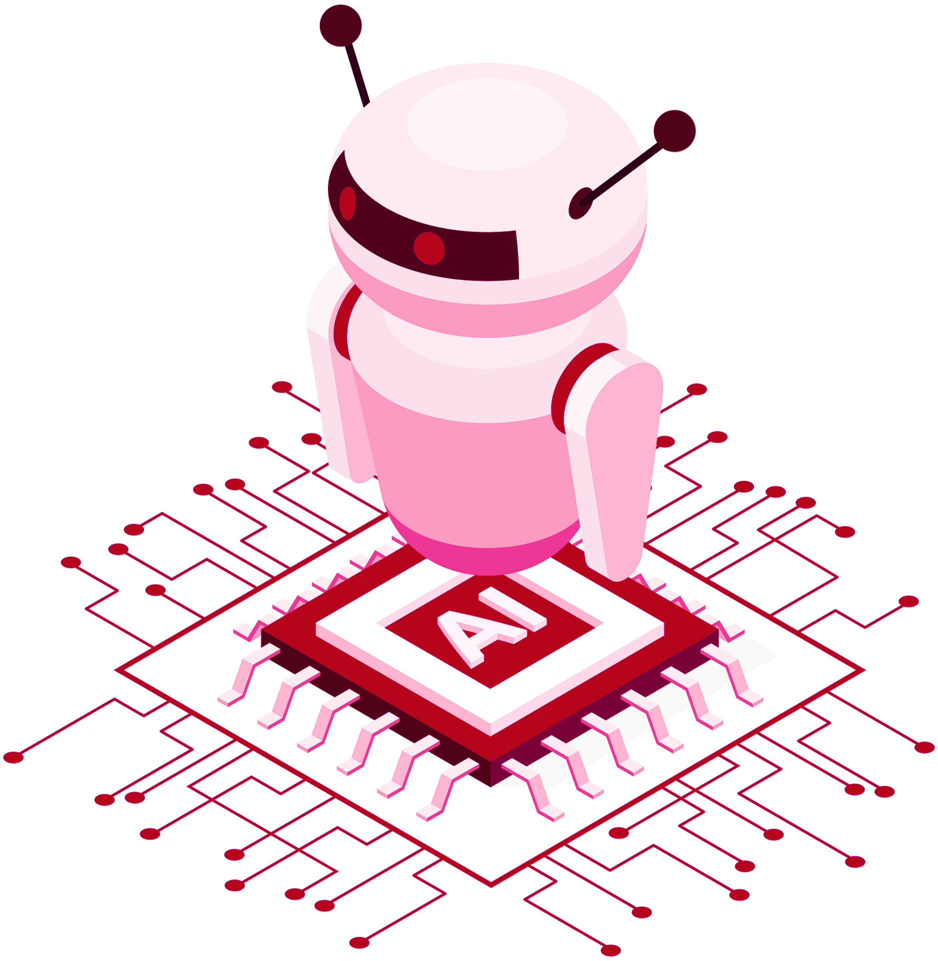

Deep Reinforcement Learning

Professor Mohammad Hossein Rohban

Solution for Homework 13:

Multi-Agent RL

By:

Amir Kooshan Fattah Hesari
401102191

Spring 2025

Grading

The grading will be based on the following criteria, with a total of 110 points:

Task	**Points**
Task 1	50
Task 2	50
Clarity and Quality of Code	5
Clarity and Quality of Report	5
Bonus 1	5
Bonus 2	5

Contents

1 Part 2: Implementing MADDPG/IDDPG

1. In our training loop, the `DDPGLoss` module utilizes `target_policies` to estimate the value of the next state. Explain clearly why employing these slowly-updating target networks, rather than the main policy networks (which change rapidly), is essential for ensuring the stability of the DDPG algorithm. (Hint: Consider what might happen if the critic tried to optimize toward a continuously moving target.)

Answer

The DDPG algorithm trains its critic network through bootstrapping using a target value
$y = r + \gamma Q_{\theta^-}(s', \pi_{\phi^-}(s'))$,
where the critic's loss function is defined as $\mathcal{L}(\theta) = \mathbb{E}\left[(Q_\theta(s,a) - y)^2\right]$.
Should we construct y using the *main* networks rather than the target networks (Q_θ and π_ϕ), both components of the regression target would shift dramatically with each gradient update. This would force the critic to pursue a *constantly shifting target*, violating the (approximate) contraction property of the Bellman operator and establishing a destabilizing positive feedback cycle between the actor and critic components. Practically, this results in explosive/oscillating temporal difference errors, high variance, and training instability—particularly when combined with the problematic trio (function approximation, bootstrapping, off-policy replay).
The target networks $\{\theta^-, \phi^-\}$ address this issue through gradual updates using Polyak averaging
$\theta^- \leftarrow (1-\tau)\theta^- + \tau\theta, \quad \phi^- \leftarrow (1-\tau)\phi^- + \tau\phi,$
where τ is kept small (e.g. $\tau \in [10^{-3}, 5 \times 10^{-3}]$). This approach ensures y remains approximately constant across multiple updates, allowing the critic to learn from a *stable* target; consequently, the actor can follow a more consistent gradient $\nabla_\phi Q_\theta(s, \pi_\phi(s))$, resulting in stable training dynamics.

2. (bonus) Consider the training plot shown in Figure 1, which resulted from modifying a single scalar hyper-parameter in the training script.

   (a) Describe the issue with the learning process depicted in the plot.

   Answer

   Problem identified. The learning trajectories demonstrate significant variability and recurring abrupt negative drops, sluggish initial progress (postponed enhancement until approximately 200–300 training cycles), and diminished final performance relative to the properly configured experiment. These characteristics indicate compromised training stability and decelerated convergence rates.

   (b) Identify which hyper-parameter you believe was changed, and explain the role of this parameter within the MADDPG algorithm.

Answer

Probable hyperparameter modification (and its impact). This observed pattern most closely aligns with employing an elevated target update coefficient τ (reduced smoothing effect). Within MADDPG/DDPG frameworks, τ determines the velocity at which target networks follow the main networks:

$$\theta^{-} \leftarrow (1-\tau)\theta^{-} + \tau\theta.$$

When τ becomes excessively high, target networks adapt to the continuously updating primary networks with insufficient lag, recreating the mobile-target dilemma and producing the erratic, oscillating learning behavior evident in the experimental data. Alternatively, an overly conservative τ would normally result in enhanced stability accompanied by reduced learning velocity. Reverting τ to an optimal setting around 0.005 typically decreases volatility and boosts sample utilization effectiveness.

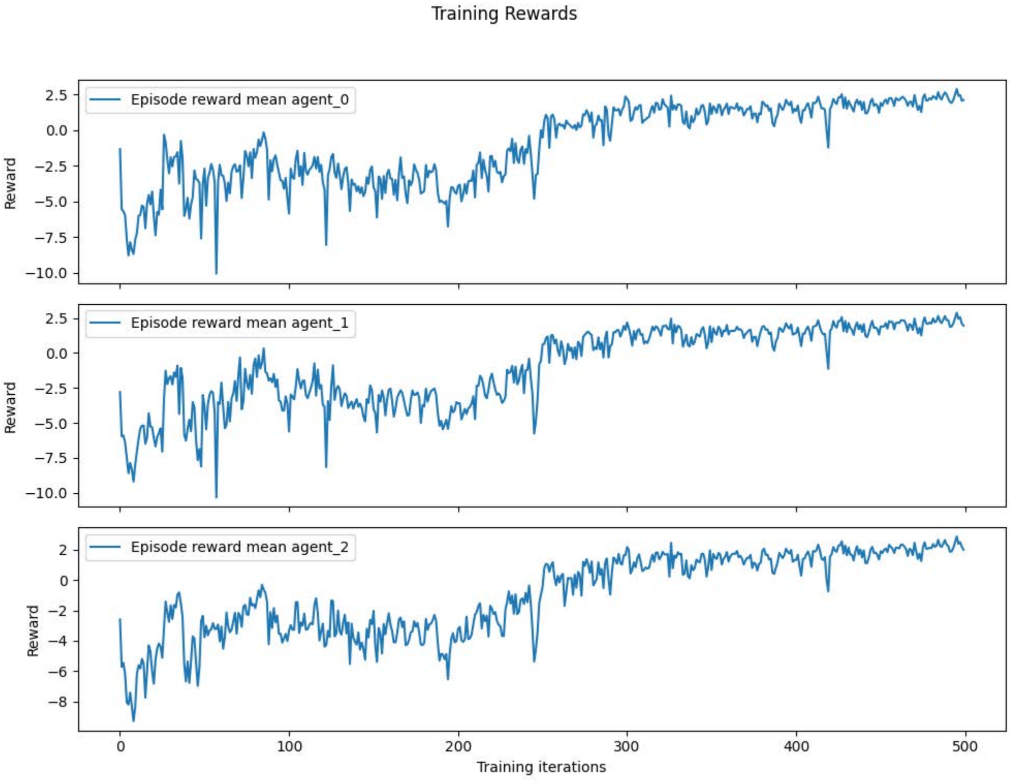

Figure 1: Agents performance after modifying a scalar hyper-parameter.

References

[1] Cover image designed by freepik

[2] Ryan Lowe, Yi I. Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. arXiv:1706.02275