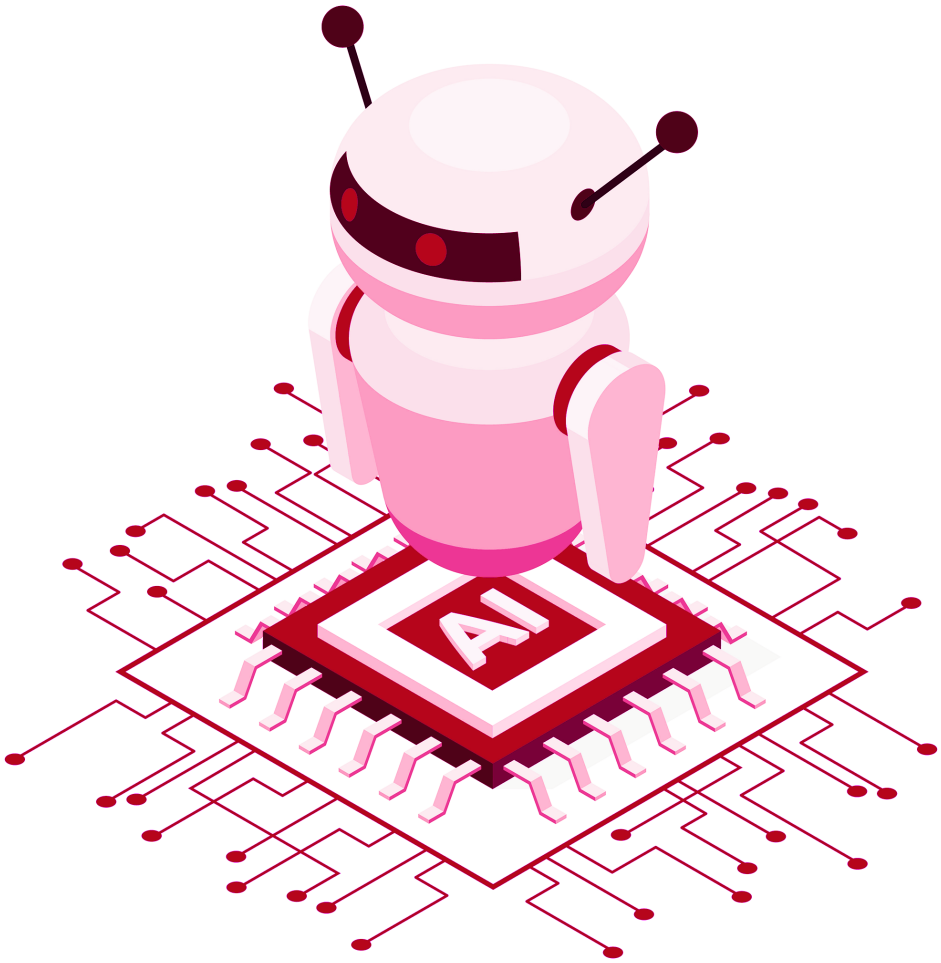

Deep Reinforcement Learning

Professor Mohammad Hossein Rohban

Homework 4:

Advanced Methods in RL

By:

Danial Parnian

401110307

RIML

Spring 2025

Contents

Grading

The grading will be based on the following criteria, with a total of 100 points:

Task	**Points**
Task 1: PPO	25
Task 2: DDPG	20
Task 3: SAC	25
Task 4: Comparison between SAC & DDPG & PPO	20
Clarity and Quality of Code	5
Clarity and Quality of Report	5
Bonus 1: Writing your report in Latex	10

Declaration: Some answers are written with assistance of GPT-4. However, none are directly copied (unless explicitly mentioned) and all the codes and responses are written manually.

1 Task 1: Proximal Policy Optimization (PPO) [25]

1.1 Question 1:

What is the role of the actor and critic networks in PPO, and how do they contribute to policy optimization?

The actor network determines the policy. It takes a state as input and returns probability distribution over actions. In this problem, the action space is continuous so it outputs the mean and std of a normal distribution from which actions are sampled. So its role is to select optimal actions (that maximize the expected reward).
The critic evaluates the value function of a state. It predicts the expected return (reward) from the current state. The critic network is used to compute the advantage function.
PPO uses the actor to explore and select actions, and the critic to provide feedback on the quality of those actions. The algorithm updates actor to maximize the expected advantage while using critic to reduce the variance of the policy gradient estimates, and thats how they contribute to policy optimization.

1.2 Question 2:

PPO is known for maintaining a balance between exploration and exploitation during training. How does the stochastic nature of the actor network and the entropy term in the objective function contribute to this balance?

Actor network returns a probability distribution of the action, rather than a specific action itself. Sampling from a distribution leads to more randomness and exploration of different actions. Also the entropy term encourages the entropy of distribution to be higher, which forces it to remain more random and cover wider range of actions. This prevents exploitation too early and encourages exploring more actions.

1.3 Question 3:

When analyzing the training results, what key indicators should be monitored to evaluate the performance of the PPO agent?

Total reward, actor and critic losses, and entropy should be monitored, since these are the most important indicators. Total reward is the primary objective we try to maximize, actor and critic losses indicate how well the policy is, and how well the value function estimates are. Also as mentioned above, entropy is important and we could monitor that too.

2 Task 2: Deep Deterministic Policy Gradient (DDPG) [20]

2.1 Question 1:

What are the different types of noise used in DDPG for exploration, and how do they differ in terms of their behavior and impact on the learning process?

The two common types of noise used are Ornstein-Uhlenbeck (OU) noise and Gaussian noise. In our implementation we used a small Gaussian noise with mean=0 and std=0.1 and added that to the predicted optimal action. This method is simple and works well in many tasks, but it may not be as effective in environments that require smooth and consistent action changes.
OU noise is a temporally correlated noise process, meaning the noise added to actions evolves over time rather than being completely random at each step. This makes it useful in environments with momentum, where smoother exploration is beneficial. However, it can sometimes lead to biased exploration if not tuned properly. Overall, OU noise encourages more structured exploration, while Gaussian noise provides more diverse but less correlated exploration. The choice between them depends on the environment and the stability required in action selection. [1]

2.2 Question 2:

What is the difference between PPO and DDPG regarding the use of past experiences?

PPO is an on-policy algorithm, and it uses the current policy to gather data and update the networks a little using them, and then forget them completely. Therefore it's data inefficient and relies on recent experience.
In contrast, DDPG is off-policy and stores past experiences in a replay buffer. It can use old transitions multiple times, which leads to more data efficiency.

3 Task 3: Soft Actor-Critic (SAC) [25]

3.1 Question 1:

Why do we use two Q-networks to estimate Q-values?

We use two Q-networks to resolve the overestimation bias that may occur with a single Q-network. By having two Q-networks, the algorithm takes the minimum of the two estimates, which helps prevent overshooting and improves the stability and performance of the learning process.

3.2 Question 2:

What is the temperature parameter(α), and what is the benefit of using a dynamic α in SAC?

This parameter controls the trade off between exploration and exploitation by scaling the entropy term in the policy's objective function. Using a dynamic α allows the algorithm to automatically adjust exploration during training. SAC uses automatic entropy tuning, but DDPG doesn't, which could be one of the reasons SAC performs better (along with stochastic policy).

3.3 Question 3:

What is the difference between evaluation mode and training mode in SAC?

It is in how actions are selected. During training, actions are sampled from the stochastic policy (for more exploration). In evaluation mode, actions are selected deterministically without added noise, which ensures we use the optimal learned policy (we just use the mean of predicted action distribution).

4 Task 4: Comparison between SAC & DDPG & PPO [20]

4.1 Question 1:

Which algorithm performs better in the `HalfCheetah` environment? Why?
Compare the performance of the PPO, DDPG, and SAC agents in terms of training stability, convergence speed, and overall accumulated reward. Based on your observations, which algorithm achieves better results in this environment?

As you can see in the plots, SAC reaches the best overall reward (over 10,000). After that, DDPG is the second best by little gap and then there is PPO which performs much weaker than SAC and DDPG. Also SAC and DDPG are much more stable than PPO. in terms of convergence speed, DDPG and SAC dont show meaningful difference but are still beter than PPO (all the models were trained for about 2 hours)

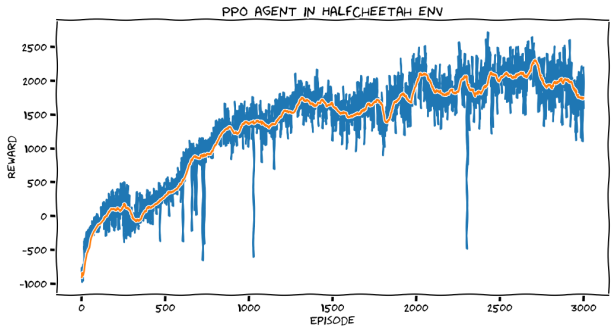

Figure 1: PPO rewards over time in HalfCheetah

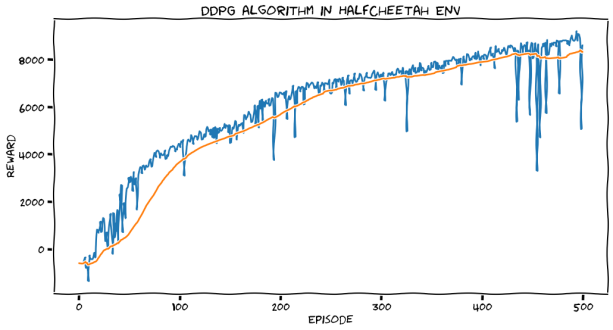

Figure 2: DDPG rewards over time in HalfCheetah

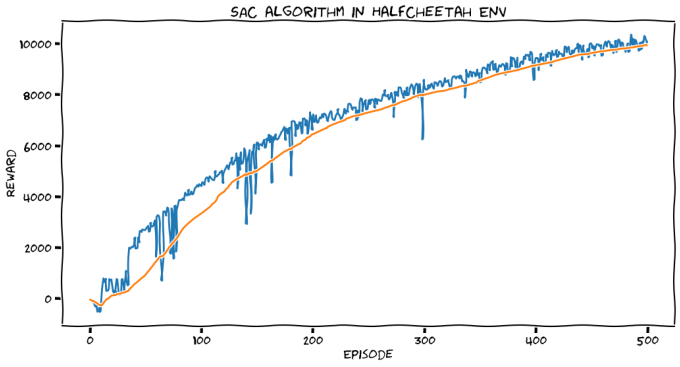

Figure 3: SAC rewards over time in HalfCheetah

4.2 Question 2:

How do the exploration strategies differ between PPO, DDPG, and SAC?
Compare the exploration mechanisms used by each algorithm, such as deterministic vs. stochastic policies, entropy regularization, and noise injection. How do these strategies impact learning in environments with continuous action spaces?

PPO uses stochastic policies (sampling actions from a normal distribution) and entropy regularization to encourage exploration by adding randomness to the actions. DDPG uses deterministic policies with noise injection, such as OU or Gaussian noise. SAC uses stochastic policies alongside with entropy maximization (including entropy in the objective of RL). In continuous action spaces, all these strategies are helpful but SAC usually performs the best.

4.3 Question 3:

What are the key advantages and disadvantages of each algorithm in terms of sample efficiency and stability?
Discuss how PPO, DDPG, and SAC handle sample efficiency and training stability. Which algorithm is more sample-efficient, and which one is more stable during training? What trade-offs exist between these properties?

PPO is stable due to its clipped objective function, but it is sample inefficient since it's on-policy and does not use past data. DDPG is more sample efficient as it is off-policy and reuses past experiences stored in the replay buffer, but it might suffer from instability due to its deterministic policy. SAC is both sample-efficient and stable, and it uses a stochastic policy to further improve DDPG algorithm.

4.4 Question 4:

Which reinforcement learning algorithm—PPO, DDPG, or SAC—is the easiest to tune, and what are the most critical hyperparameters for ensuring stable training for each agent?
How sensitive are PPO, DDPG, and SAC to hyperparameter choices, and which parameters have the most significant impact on stability? What common tuning strategies can help improve performance and prevent instability in each algorithm?

PPO is the easiest to tune, due to its robustness and fewer hyperparameters. The important hyperparameters for PPO are the clipping parameter and learning rate. DDPG is more sensitive to hyperparameter choices and usually hard to tune. SAC is more stable than DDPG, but still requires careful tuning of the entropy coefficient (alpha), learning rate and target update interval.
Common tuning strategies include using grid search or random search for hyperparameter optimization, ensuring a large enough replay buffer for DDPG and SAC, and adjusting the learning rate and batch size to balance stability and performance. Regular evaluation and monitoring of training curves can also help in identifying and addressing instability issues. [1]

References

[1] This paragraph is generated by ChatGPT.