Week 7 Prerequisite: Introduction to Q-Learning

And complete the Pre-Quiz on Gradescope.

Week 7 Tuesday Paper 1: Double DQN

Explained by HuggingFace

Supplemental: The Original DQN paper

1. What is the overestimation problem in Q-Learning? What is the role of Theorem 1 in the paper?

Ans: The overestimation problem in Q-learning arises due to the maximization step in the algorithm, where estimated action values are used both to select and to evaluate an action. This double usage tends to favor overestimated values, leading to overoptimistic value estimates. In practice, especially with large-scale problems or deep neural networks like DQN, this can lead to substantial overestimations in some games. The role of Theorem 1 is to show that even if value estimates are on average correct, estimation errors can still drive the estimates up, away from the true optimal values. Specifically, Theorem 1 provides a mathematical explanation of how and why these overestimations occur due to the maximization step in Q-learning, illustrating that overestimations can be more common and severe than previously acknowledged.

2. What modifications has Double DQN done to the original Q-Learning framework? What is the purpose of these changes?

Ans: Double DQN modifies the original Q-learning framework by decoupling the action selection from the action evaluation, addressing the overestimation problem. In Double DQN, two sets of weights-online network weights and target network weights are used: one set to determine the greedy policy and the other to evaluate its value. The purpose of these changes is to reduce the overoptimistic value estimates seen in standard DQN, leading to more accurate value estimates and, consequently, better policy performance.

3. What do the experiments suggest about the difference in quality and stability of results of Double DQN vs. DQN?

Ans: The experiments suggest that Double DQN reduces overestimations compared to standard DQN and results in better policy performance. In games where DQN shows substantial overestimations, Double DQN provides more accurate value estimates and higher scores, indicating that the original overestimations were leading to poorer policies. Double DQN's learning is also more stable, suggesting that reducing overestimations benefits the stability and reliability of learning, ultimately leading to the development of better and more robust policies.

4. What are the main hyperparameters in Double DQN training? What purposes do they serve?

Ans: The main hyperparameters in Double DQN training include the discount factor, the learning rate, the update frequency of the target network (τ), the size of the experience replay memory, the batch size for updates, and the exploration policy. The discount factor balances the importance of immediate versus future rewards, learning rate controls the step size of the weight updates, τ determines how often the target network is updated, the replay memory and batch size affect the stability and efficiency of learning, and the exploration policy guides the trade-off between exploring new actions and exploiting known good actions.

Week 7 Tuesday Paper 2: DDPG

Explained by: Deep Deterministic Policy Gradient — Spinning Up documentation

Supplemental: A comprehensive review of offline RL

1. What is the fundamental problem about continuous spaces that the paper is trying to solve? How are they different from discrete spaces?

Ans: The paper addresses the challenge that traditional deep reinforcement learning algorithms like DQN are not directly applicable to problems with continuous (real-valued) and high-dimensional action spaces. In continuous spaces, finding the action that maximizes the action-value function requires iterative optimization at every step, which is impractical with large, unconstrained function approximators and nontrivial action spaces. This issue is in contrast to discrete spaces where actions can be easily enumerated and the max operation is straightforward.

2. What is the Actor-Critic architecture used here? How does it solve the problem identified?

Ans: The paper employs a model-free, off-policy actor-critic algorithm based on the deterministic policy gradient. The actor-critic approach used integrates insights from Deep Q-Learning, including the use of replay buffers and target networks to stabilize training. The actor deterministically maps states to actions, while the critic estimates the expected return of taking such actions. This architecture solves the identified problem by avoiding the need for iterative optimization processes in action selection, making the approach viable for continuous action spaces.

3. Explain Replay Buffers and Target Networks. Why are they important in DDPG?

Ans: Replay buffers decouple the time correlation of the observed data by mixing past and present experiences, thus allowing the algorithm to benefit from learning across a set of uncorrelated transitions. Target networks help stabilize learning by providing a slowly updating target value, preventing the updates from being too tied to a constantly moving reference point. These elements are crucial in the DDPG algorithm to mitigate the issues of sample correlation and non-stationarity of targets, which are prominent challenges in using deep learning for reinforcement learning

4. How does the DDPG algorithm balance exploration and exploitation?

Ans: DDPG handles the balance between exploration and exploitation by adding noise to the policy's actions. This approach allows the agent to explore the action space sufficiently. Specifically, the paper uses an Ornstein-Uhlenbeck process to generate temporally correlated noise, which is beneficial for exploration in environments with inertia, providing a method to explore efficiently while still exploiting learned policies

5. What do the experimental results suggest about the quality and stability of DDPG?

Ans: The experimental results suggest that the DDPG algorithm can robustly solve a variety of challenging physical control problems involving complex movements and dynamics. The algorithm was able to learn competitive policies across more than 20 simulated physics tasks using the same architecture and hyperparameters. Some learned policies even exceeded the performance of a planning algorithm with full access to the dynamics. The results also indicate that DDPG can learn effectively from both low-dimensional state inputs and raw pixel inputs

6. From your understanding, which is a stronger driver of the learning in DDPG: Q-Learning or Policy Learning?

Ans: From the paper, it's clear that both Q-learning and policy learning play crucial roles in the learning process of DDPG. However, the use of the deterministic policy gradient, the actor-critic structure, and the emphasis on learning and updating policy directly from the gradient of the action-value function suggest that policy learning is a strong driver of the learning process in DDPG. However, Q-learning is also integral in providing the necessary value estimates that guide policy improvement. Therefore, both components are crucial, but the architecture leans slightly more on the side of policy learning due to its foundational role in the actor-critic setup