Deep Reinforcement Learning (Sp25)

Instructor: Dr. Mohammad Hossein Rohban

Quiz Solutions - Lecture 19

Solution By: Behnia Soleymani

Q1

Why are tasks like Montezuma's Revenge considered "hard exploration problems" for standard RL agents?

A) Because the state space is infinitely large.
B) Because the rewards are very dense, confusing the agent.
C) Because rewards are sparse and require long, specific sequences of actions to achieve.
D) Because the discount factor is always set too low.

Correct Answer: C
Explanation: hard exploration problems often involve sparse rewards (like the key) that only come after long, specific sequences of actions, making it hard for agents to stumble upon them randomly.

Q2

What is the key difference in exploration behavior within an episode when using Bootstrapped DQN compared to simple epsilon-greedy exploration?

A) Bootstrapped DQN takes more random actions than epsilon-greedy.
B) Bootstrapped DQN selects a random Q-function head and acts consistently according to it for the whole episode, while epsilon-greedy adds randomness at each action step.
C) Epsilon-greedy uses multiple heads, while Bootstrapped DQN only uses one.
D) Bootstrapped DQN only explores at the very beginning of training, while epsilon-greedy explores continuously.

Correct Answer: B
Explanation: The Bootstrapped DQN achieves "consistent exploration" by picking one head (sampled strategy) and sticking to it for the episode, contrasting this with the action-level randomness ("dithering") of epsilon-greedy.

Q3

Explain why the epsilon-greedy exploration strategy often fails in tasks that require long sequences of specific actions to obtain a reward.

Explanation: Epsilon-greedy explores by taking random actions with a small probability (ϵ) at each step. For tasks requiring a long sequence of N specific actions, the probability of executing that exact sequence using random exploration steps is extremely low (roughly proportional to ϵ^N if many steps need to be random, or dependent on $(1-\epsilon)$ for exploiting known parts and ϵ for exploring new parts, which still diminishes rapidly). The agent is unlikely to stumble upon the correct, long path purely by chance and often gets "stuck" or "dithers" without making meaningful progress towards the distant reward.

Q4

The Bootstrapped DQN method relies on training multiple Q-function "heads". Explain how bootstrapping (re-sampling the dataset D with replacement) helps the exploration in this method.

Explanation: Bootstrapping involves creating multiple training datasets ($D_1, \ldots, D_N$) by sampling with replacement from the original replay buffer D. Each dataset D_i will be slightly different – some transitions will be duplicated, others omitted. When each Q-function head (Q_k) is trained primarily on its corresponding dataset D_k, these small variations in the training data lead the heads to learn slightly different value estimates and, consequently, potentially different optimal policies (strategies). This data-driven variation is key to generating the diversity among the heads needed for effective Thompson Sampling-like exploration.

Q5

An RL agent uses an epsilon-greedy strategy with $\epsilon = 0.09$. It faces a task requiring it to first successfully execute a sequence of $k = 6$ specific actions by exploiting its current best policy, followed immediately by taking one specific necessary exploratory action. Calculate the chance of the right exploration in this setting using epsilon greedy.

Solution: To determine the probability of the agent successfully executing the required sequence (6 specific 'exploit' actions followed by 1 specific 'explore' action) using an epsilon-greedy strategy with $\epsilon = 0.09$, we calculate the product of the probabilities for each required action. The probability of selecting the correct action during an 'exploit' step is $(1 - \epsilon) = 1 - 0.09 = 0.91$. The probability of selecting the specific required action during the 'explore' step is $\epsilon = 0.09$. Therefore, the probability of the entire 7-step sequence occurring correctly is $P(\text{exploit})^6 \times P(\text{explore})^1 = (0.91)^6 \times (0.09)^1$. Calculating this yields approximately $0.57 \times 0.09 \approx 0.0513$. Thus, the chance of performing the right exploration step immediately after the required exploitation sequence is approximately 0.0513, or 5.13%.