Deep Reinforcement Learning (Sp25)

Instructor: Dr. Mohammad Hossein Rohban

Quiz Solutions - Lecture 10
Solution by : Arshia Gharooni

Q1

Consider a system where, at each time step, there is a prediction error of magnitude ε. If these errors accumulate linearly over T steps in an open-loop rollout, determine the order of the total accumulated error as a function of T and ε.

(a) $O(T\varepsilon)$
(b) $O(\sqrt{T}\varepsilon)$
(c) $O(T^2\varepsilon)$
(d) $O(\varepsilon)$

Correct Answers: C

At every step, the system incurs an error of ε. Since the errors accumulate linearly, the total error after T steps is given by

$$\varepsilon + \varepsilon + \cdots + \varepsilon \quad (\text{with } T \text{ terms}) = T\varepsilon.$$

Thus, the error scales linearly with T, which can be expressed in Big-O notation as:

$$O(T\varepsilon).$$

Q2

Consider a node s_i in an MCTS tree with total accumulated reward $Q(s_i) = 80$ and visited $N(s_i) = 8$ times. Its parent node has been visited $N(\text{parent}) = 50$ times. Using an exploration constant $C = 2$, the UCB score is given by:

$$\text{UCB}(s_i) = \frac{Q(s_i)}{N(s_i)} + 2 \cdot \sqrt{\frac{\ln\big(N(\text{parent})\big)}{N(s_i)}}.$$

Which of the following is *closest* to the UCB score for node s_i?

a) 11.41
b) 14.00
c) 11.73
d) 11.40

Correct Answers: D

We are given the UCB formula:

$$\text{UCB}(s_i) = \frac{Q(s_i)}{N(s_i)} + C \cdot \sqrt{\frac{\ln\big(N(\text{parent})\big)}{N(s_i)}}$$

with the following values:

$$Q(s_i) = 80, \quad N(s_i) = 8, \quad N(\text{parent}) = 50, \quad C = 2.$$

Step 1: Compute the Exploitation Term

$$\frac{Q(s_i)}{N(s_i)} = \frac{80}{8} = 10.$$

Step 2: Compute the Exploration Term

1. Compute the logarithm:
$$\ln\big(N(\text{parent})\big) = \ln(50) \approx 3.912.$$
2. Divide by $N(s_i)$:
$$\frac{\ln(50)}{N(s_i)} = \frac{3.912}{8} \approx 0.489.$$
3. Take the square root:
$$\sqrt{0.489} \approx 0.699 \quad (\text{approximately } 0.7).$$
4. Multiply by the constant C:
$$C \cdot \sqrt{\frac{\ln(50)}{8}} \approx 2 \times 0.699 \approx 1.398 \quad (\text{approximately } 1.4).$$

$$\text{UCB}(s_i) \approx 10 + 1.4 = 11.4.$$

Q3

In Monte Carlo Tree Search (MCTS), what role does the exploration term in the UCB score serve?

a) It penalizes nodes with high cumulative reward to balance exploration.
b) It increases the score for nodes with fewer visits to encourage exploration.
c) It discounts rewards based on the depth of the node.
d) It forces the algorithm to always select the node with the highest average reward.

Correct Answers: B

In Monte Carlo Tree Search (MCTS), a common strategy for node selection is based on the *Upper Confidence Bound (UCB)* formula. A typical form of the UCB formula for a child node i is given by:

$$\text{UCB}_i = \underbrace{\frac{Q_i}{n_i}}_{\text{exploitation term}} + c\sqrt{\frac{\ln N}{n_i}},$$

where:

- Q_i is the total (or average) reward from child node i.
- n_i is the number of times child node i has been visited.
- N is the total number of visits to the parent node.
- c is a constant that controls the balance between exploration and exploitation.

The first term, $\frac{Q_i}{n_i}$, represents the exploitation of the information gathered so far, favoring nodes with higher average rewards. The second term, $c\sqrt{\frac{\ln N}{n_i}}$, serves as an exploration bonus that increases the score for nodes with fewer visits (i.e., lower n_i). This bonus encourages the algorithm to explore less-visited nodes to gather more information, preventing the search from focusing too heavily on nodes that already appear promising based solely on current rewards.

Thus, the exploration term *increases the score for nodes with fewer visits*, ensuring that the algorithm maintains a balance between exploring new nodes and exploiting known rewarding nodes.

Q4

Consider a **parameterized** policy $\pi_\theta(a_l|s_l)$ and a **differentiable** learned dynamics model $\hat{f}_\phi$. Suppose we define a loss function that directly tries to maximize the sum of rewards for a fixed horizon T by **backpropagating** through the model:

$$\mathcal{L}(\theta) = \sum_{l=0}^{T-1} r(s_l, a_l), \quad \text{with} \quad a_l \sim \pi_\theta(a_l|s_l), \quad s_{l+1} = \hat{f}_\phi(s_l, a_l).$$

What is the form of the gradient *w.r.t.* the policy parameters θ, assuming we ignore any gradient *w.r.t.* ϕ (the dynamics model parameters)?

a)
$$\nabla_\theta \mathcal{L}(\theta) = \sum_{l=0}^{T-1} \nabla_\theta \log \pi_\theta(a_l|s_l)\, r(s_l, a_l).$$

b)
$$\nabla_\theta \mathcal{L}(\theta) = \sum_{l=0}^{T-1} \left(\nabla_\theta \pi_\theta(a_l|s_l)\right)^\top \nabla_{a_l} r(s_l, a_l).$$

c)
$$\nabla_\theta \mathcal{L}(\theta) = \sum_{l=0}^{T-1} \left(\nabla_{a_l} r(s_l, a_l) + \nabla_{a_l} r(s_{l+1}, a_{l+1}) \cdot \frac{\partial \hat{f}_\phi}{\partial a_l}\right) \nabla_\theta \pi_\theta(a_l|s_l).$$

d)
$$\nabla_\theta \mathcal{L}(\theta) = \sum_{l=0}^{T-1} \left(\nabla_{a_l} r(s_l, a_l) + \frac{\partial s_{l+1}}{\partial a_l} \cdot \nabla_{s_{l+1}} r(\cdots)\right) \frac{\partial a_l}{\partial \theta}.$$

Correct Answers: D

Consider the loss function

$$\mathcal{L}(\theta) = \sum_{l=0}^{T-1} r(s_l, a_l),$$

where the actions a_l are drawn from the parameterized policy $\pi_\theta(a_l|s_l)$ and the state transitions are given by the differentiable dynamics model

$$s_{l+1} = \hat{f}_\phi(s_l, a_l).$$

Since we ignore gradients with respect to ϕ, the loss depends on θ both directly (through a_l) and indirectly (through the state s_{l+1} affecting future rewards).
Applying the chain rule, the gradient of the loss with respect to θ becomes:

$$\nabla_\theta \mathcal{L}(\theta) = \sum_{l=0}^{T-1} \left(\nabla_{a_l} r(s_l, a_l) + \frac{\partial s_{l+1}}{\partial a_l} \cdot \nabla_{s_{l+1}} r(\cdots)\right) \frac{\partial a_l}{\partial \theta}.$$

Here:

- $\nabla_{a_l} r(s_l, a_l)$ captures the direct effect of a_l on the immediate reward.
- $\frac{\partial s_{l+1}}{\partial a_l} \cdot \nabla_{s_{l+1}} r(\cdots)$ captures the indirect effect on future rewards via the state transition.

Q5

What is the rationale behind combining short-horizon model-based rollouts with frequent re-planning in model-based RL?

a) It maximizes exploration by reducing the action space.
b) It minimizes computational costs while training ensemble models.
c) It reduces the compounding of model errors and enables corrective feedback from actual environment steps.
d) It improves the accuracy of gradient estimates during backpropagation.

Correct Answers: D

In model-based RL, one typically uses a learned dynamics model (or ensemble of models) to simulate future states and rewards. However, no learned model is perfect, and small inaccuracies can grow exponentially if you simulate too far into the future (a phenomenon known as **compounding model errors**).

Short-horizon rollouts help mitigate this problem:

1. **Shorter Predictions:** By planning only a few steps ahead, the simulation doesn't rely on the model's predictions for too long, limiting how much any single model error can multiply.
2. **Frequent Re-Planning:** After executing those short-horizon actions in the real environment, you collect fresh data, update (or correct) the model, and re-plan. This allows continuous **corrective feedback** from the environment to keep the learned model accurate.

Thus, combining short-horizon rollouts with frequent re-planning is mainly motivated by reducing the impact of accumulating errors in the model and maintaining an up-to-date representation of the environment's dynamics.