Question 1:

The correct answer is:

True

Explanation:
In policy gradient methods, subtracting a suitable baseline can help in reducing the variance of the gradient estimates without introducing any bias. This means that while the expected value of the gradient remains unchanged, the variability or noise in the gradient estimates is reduced. Commonly used baselines include the average return or the value function estimate of a state. By reducing the variance, the learning process can become more stable and potentially converge faster. Therefore, the statement is true: decreasing bias by subtracting a suitable baseline in policy gradient estimation produces less noisy gradient estimates.

Question 2:

The correct answer is:

True

Explanation:
Actor-critic methods consist of two main components: the actor and the critic. The actor is responsible for selecting actions based on a learned policy, and the critic evaluates the chosen actions by estimating the value function (either state-value or action-value). The actor updates its policy based on feedback from the critic. In actor-critic methods, the policy (actor) is explicitly represented and learned, often parameterized by neural networks or other function approximators. Therefore, the statement is true: Actor-critic methods use learned policy functions.

Question 3:

The correct answer is:

True

Explanation:
Actor-critic methods are a type of reinforcement learning algorithm that combines both value-based and policy-based approaches. In actor-critic methods:

- The "actor" is responsible for selecting actions based on a learned policy.
- The "critic" evaluates the chosen actions by estimating the value function (either the state-value function $V(s)$ or the action-value function $Q(s, a)$).

The critic's value function estimate is used to provide feedback to the actor, guiding the updates to the policy. Therefore, the statement is true: Actor-critic methods estimate a value function.

Question 4:

The correct answer is:

True

Explanation:
Actor-critic methods can indeed handle continuous action spaces. In actor-critic architectures, the "actor" outputs a continuous action or a distribution over continuous actions, often parameterized by means and variances in the case of Gaussian distributions. Algorithms like Deep Deterministic Policy Gradient (DDPG) and Soft Actor-Critic (SAC) are examples of actor-critic methods designed specifically for continuous action spaces. In these methods, the actor network directly outputs continuous-valued actions, making them suitable for tasks with continuous action domains. Therefore, the statement is true: Actor-critic methods can be used to produce actions for continuous action spaces.

Question 5:

Advantage estimation is used to determine how much better or worse an action is compared to the average action in a given state. The advantage function $A(s, a)$ is defined as the difference between the action-value function $Q(s, a)$ and the state-value function $V(s)$:

$$A(s, a) = Q(s, a) - V(s)$$

Using the advantage function in policy gradient updates can help in reducing the variance of the gradient estimates, which can lead to more stable learning. However, the primary purpose of advantage estimation is not to address potential divergence due to large updates (large step size).

The step size (or learning rate) determines the magnitude of the updates. If the step size is too large, it can lead to divergence regardless of whether advantage estimation is used. Techniques like trust region methods or clipped objective functions (as used in PPO - Proximal Policy Optimization) are more directly aimed at addressing potential divergence due to large policy updates.

So, you are correct. The statement "Advantage estimation helps to address potential divergence of policy updates due to large updates (large step size)" is False.

That's incorrect. Advantage estimation as such helps to address the noise (uncertainty) in the gradient estimates, which potentially speeds up convergence, but does not help in setting the step size.

The correct answer is 'False'.

Question 6:

The correct answer is:

False

Explanation:
The advantage function $A(s, a)$ is defined as the difference between the action-value function $Q(s, a)$ and the state-value function $V(s)$:

$$A(s, a) = Q(s, a) - V(s)$$

The advantage function measures how much better or worse an action a is in state s compared to the average action in that state. If the action a is just as good as the average action, the advantage will be zero. If the action a is better than the average action, the advantage will be positive. However, if the action a is worse than the average action, the advantage will be negative.

Therefore, the advantage function can be positive, negative, or zero, depending on the quality of the action relative to the average action in that state. The statement "The advantage function is always positive" is false.

Question 7:

The correct answer is:

True

Explanation:
Yes, it is possible to approximate the advantage function $A(s, a)$ using only a function approximator for the state value function $V(s)$ without directly approximating $Q(s, a)$. One common approach to do this is by using the Temporal Difference (TD) error.

Given a transition (s, a, r, s'), the TD error δ is defined as:
$\delta = r + \gamma V(s') - V(s)$

The TD error represents the difference between the estimated value of the current state $V(s)$ and the estimated value of the next state $V(s')$ (discounted by γ) plus the immediate reward r. This difference can be used as an estimate of the advantage $A(s, a)$.

Thus, the advantage function can be approximated as:
$A(s, a) \approx \delta$

By using this approach, one can estimate the advantage function without explicitly modeling or approximating the action-value function $Q(s, a)$. Therefore, the statement is true.

Question 8:

The correct answer is:

True

Explanation:
Importance sampling is a technique used to estimate the expected value of a random variable under one distribution, given samples from another distribution. In the context of reinforcement learning, importance sampling can be used to estimate the performance (expected return) of a target policy π using data generated by a behavior policy π_0.

The importance sampling ratio is given by:
$\rho_t = \frac{\pi(A_t|S_t)}{\pi_0(A_t|S_t)}$

Where A_t is the action taken at time t, S_t is the state at time t, π is the target policy, and π_0 is the behavior policy.

By weighting the returns by the importance sampling ratio, one can correct for the difference in action probabilities between the target policy and the behavior policy, allowing for an unbiased estimate of the expected return under the target policy π using data from the behavior policy π_0.

Therefore, the statement is true: Importance sampling can be used to estimate the performance of a policy π given data from another policy π_0.

Question 9:

The correct answer is:

False

Explanation:
Model-based reinforcement learning (MBRL) aims to increase the data efficiency of reinforcement learning by learning a model of the environment's dynamics, not the action value function. Specifically, MBRL learns a model that predicts the next state and reward given the current state and action. This model can then be used to simulate experiences, allowing the agent to plan and make decisions without interacting with the actual environment, thereby improving data efficiency.

In contrast, model-free reinforcement learning methods, such as Q-learning or SARSA, directly learn the action value function (or policy) without explicitly modeling the environment's dynamics.

Therefore, the statement is false: Model-based reinforcement learning aims to increase the data efficiency of reinforcement learning by learning a model of the environment's dynamics, not the action value function.

Question 10:

The correct answer is:

True

Explanation:
Dyna-Q is indeed an approach for model-based reinforcement learning. The Dyna-Q algorithm combines both direct learning from real interactions with the environment (model-free learning) and simulated learning from a learned model of the environment (model-based learning).

In Dyna-Q:

1. The agent interacts with the environment and updates its action-value function (Q-values) using a model-free method like Q-learning.
2. The agent also learns a model of the environment's dynamics, which predicts the next state and reward given the current state and action.
3. The agent then performs "planning" by simulating experiences using the learned model. For each simulated experience, the agent updates its Q-values as if the experience was real.
4. This combination of real and simulated experiences accelerates learning by allowing the agent to benefit from both actual interactions and imagined ones based on the model.

Therefore, the statement is true: Dyna-Q is an approach for model-based reinforcement learning.