DS685 AI for Robotics

ASSIGNMENT 4 Final Exam

YASH PATEL
31702347
yp359

Q. Question 1

1) As per deterministic policy, Bellman eqn should future actions as fixed mapping $a' = \mu(s')$

so action value in varies directly with the chosen action.

Chain rule is used, deterministic policy gradient according to gradient

$$\nabla_a Q^\mu(s,a) \text{ at } a = \mu(s).$$

→ Single action considered for each state, so no integral over action space is required.

→ In contrast, stochastic policy gradient must average over actions sampled from $\pi(a/s)$; this introduces integrals over action space

2) TD(0) stat- update learns only the value of a state $V(s)$, that measures how good state is under a policy but

independent of action.

As a result, it provides no information about which action should be taken or how the action should change.

Now in continuous action spaces, there is no direct way to derive a control command from V(s) without an additional optimization or planning step.

So TD(0) state value update alone cannot directly generate continuous control commands for TurtleBot3.

3. In TB3 now, according to policy smooth linear and angular velocity commands that are sent directly to robot's motors

Deterministic actor establishes a direct and stable mapping from the current state to a single control action, that leads to predictable and ~~actionable~~

smooth behaviour.

- If we gauge how the return changes with respect to the action, the actor can be refined using action gradients allowing small, continuous adjustments to the velocity commands.

- This is deby deterministic actor suited for control problem.

Question 2

i) TD target $y = r + \gamma Q_{\phi'}(s', \mu_\theta(s'))$

ii) In DDPG TD target, depends explicitly on the current policy, and the next action is produced by actor.

In contrast TD(0) uses $y = r + \gamma V(s')$ which evaluates states only and cannot support control.

Here 'Q' learning removes policy dependence by using a max operator, $\max_{a'} Q(s', a')$ but this introduces overestimation bias and unstable in continuous spaces.

So DDPG avoids max operator & instead relies on slowly updated target networks to stabilize TD target y.

2) "max a'" operator in "Q" learning introduces overestimation bias which can amplify value errors and lead to unsafe actions in sim to real collision avoidance

- DDPG improves stability by utilizing slowly updated target networks, which reduce ~~non-static~~ non stationarity in the TD target and prevent unstable feedback between actor and critic.

- TD(0) value prediction is more stable because it avoids action maximization & policy updates. But it only estimates state values, so it cannot by itself define a control policy for real time navigation.

3) In this (S→R) nav., the critic guides how the policy updates its actions, so critic instability can cause abrupt or unsafe control commands on a real robot. While instability

in simulation mainly affects performance, on a real Turtlebot3 it can lead to oscillations or collisions.

Therefore, maintaining a stable critic is essential to ensure smooth and safe real no world navigation.

Question 3

1) Because collisions are heavily penalized, the critic learns steep drops in $Q(s,a)$ near unsafe actions.

In DDPG, the actor follows $\nabla_a Q(s,a)$ so these gradients push the policy away from risky actions, while sparse goal rewards provide weaker gradients toward progress.

2) With weak expl exploration noise, deterministic policies explore only limited actions. In a penalty dominated reward setting, slow or minimal movements avoid negative rewards and recieve higher value, causing the critic to reinforce conservative, low variance behaviours.

3) In stochastic TD methods, randomness is part of the policy $\pi(a|s)$ so different actions are naturally sampled in the same state, leading to broader exploration.
In deterministic policies, exploration comes only from added noise around a fixed action, which is more local and limited.

4) Silver et al. assume exploration is provided externally to the deterministic policy. In real time nav., this allows exploration during training while keeping the

deployed policy deterministic and stable for safe execution.

Question 4

1) Human demonstrations can pretrain the actor by behaviour cloning, learning $\hat{\mu}(s) \approx a$, and pretrain the critic by fitting TD targets on demonstrated transitions

This provides a good initial policy and value estimate before RL fine-tuning.

2) TD updates reduce errors in the critic's value estimates for demonstrated actions. As the critic improves, its action gradient $\nabla_a Q_\phi(s, a)$ guides the actor toward actions with higher predicted return allowing policy to refine and go

beyond the demonstration

3) TD(0) learns only state value $V(s)$, which cannot directly produce continuous actions. Deriving actions from $V(s)$ requires an additional optimization or planning step. In contrast actor-critic methods use $Q(s,a)$ whose $\nabla_a Q(s,a)$ provides direct gradients to refine continuous control actions, which TD(0) alone cannot do.

4) A DDPG-style actor-critic which with demonstrations works better when actions are continuous and rewards are sparse. Demonstrations initialize the policy and the critic's gradients enable efficient refinement, giving better performance and same efficiency than pure TD value learning.

Questions

1) The perception and language encoders are treated as fixed feature extractors, so the state representation s is unchanged.

The policy gradient updates only the action head $\mu_\theta(s)$ w.r.t.

$$\nabla_\theta J(\theta) = E_{s \sim \rho^\mu}[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)]$$

Because the gradient flows only through μ_θ, RL fine tuning adjusts how features are mapped to continuous actions without modifying upstream perception or language representation

2) Deterministic policy gradients are suitable because the VLA model is already well calibrated from supervised pretraining.

DDPG applies small, low variance, reward aligned corrections to the action head via $\nabla_a Q^\mu(s,a)$ unlike

stochastic policies that reshape the entire action distribution.

3) TD critic signals correct systematic biases under distribution shift. Overly conservative actions are pushed toward higher reward behaviours while overly aggressive actions are damped.

The gradient $\nabla_a Q^\mu(s,a)$ directly reshapes the action mapping.

Unlike supervised pretraining, which only imitates demonstrations, RL fine tuning uses TD based critic updates to apply reward driven corrections.