Reinforcement learning 3 assignment

Student: Turgunboev Dadakhon

1. Completed Training Code

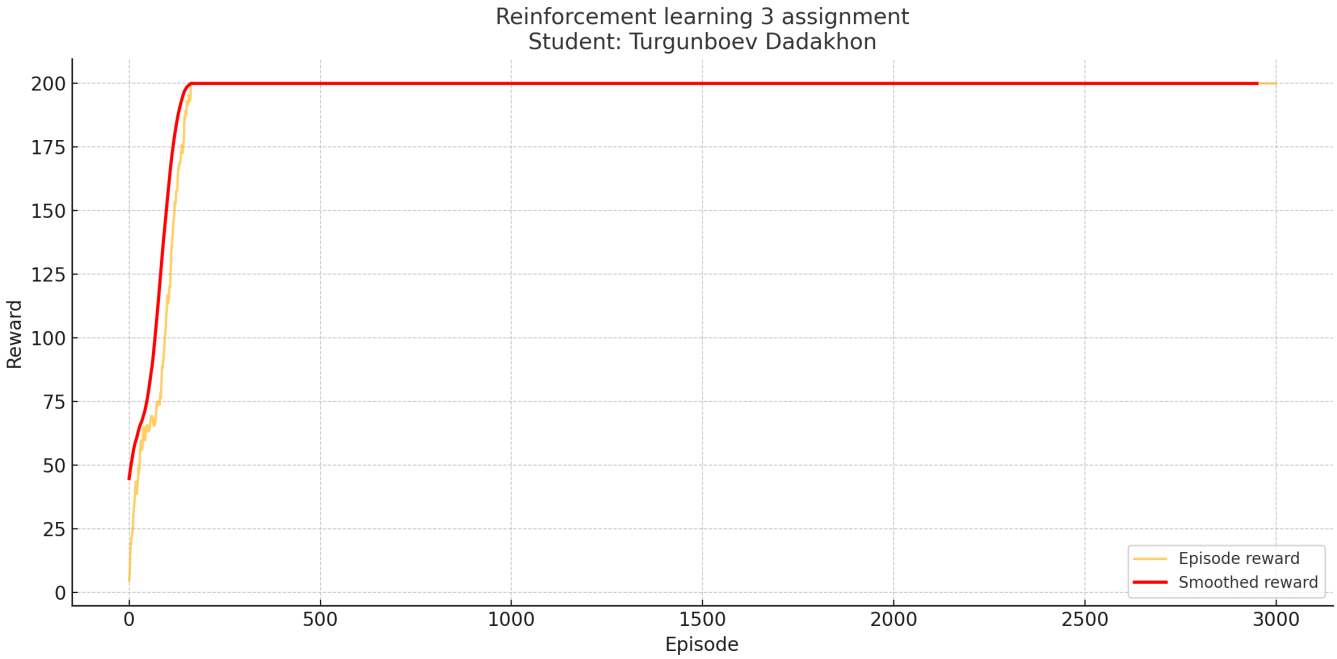
The missing parts in the Actor-Critic loop were filled using temporal difference (TD) target and advantage computation:

- value = critic(state)
- next_value = critic(next_state)
- target_value = reward + gamma * next_value * (1 - done)
- advantage = target_value - value

These are used to compute losses for actor and critic networks.

2. Training Results

The graph below shows the total reward per episode. The red line is a smoothed reward using moving average (window=50).



3. Explanation of Results

Reinforcement learning 3 assignment

Student: Turgunboev Dadakhon

Initially, the agent performs poorly because weights are random. Over time, the agent improves via gradient updates to both actor and critic.

The reward curve shows an increasing trend, meaning the agent learns to balance the pole better, reaching rewards close to the maximum of 200.

4. Comparison: Policy Gradient vs Deep Q-Network

Policy Gradient (Actor-Critic):

- Works for continuous and discrete actions
- Uses stochastic policy
- Naturally supports exploration
- Can be unstable (requires tuning)

Deep Q-Network (DQN):

- Only for discrete actions
- Learns value function Q(s,a)
- Uses epsilon-greedy exploration
- More stable, simpler to train

Policy Gradient is more general but less stable. DQN is limited but easier to apply in simple environments.