CS6700 : Reinforcement Learning

Programming Assignment #2

Deadline: 11:59 PM - 6 April, 2024

1 Environments

In this programming task, you'll utilize the following Gymnasium environments for training and evaluating your policies:

- Acrobot-v1: The system consists of two links connected linearly to form a chain, with one end of the chain fixed. The joint between the two links is actuated. The goal is to apply torques on the actuated joint to swing the free end of the linear chain above a given height while starting from the initial state of hanging downwards.
- CartPole-v1: A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The pendulum is placed upright on the cart and the goal is to balance the pole by applying forces in the left and right direction on the cart.

2 Algorithms

You are tasked with training two variants of each Dueling-DQN and Monte-Carlo REINFORCE and assessing their comparative performance.

2.1 Dueling-DQN

Dueling DQN is an extension of the DQN algorithm, designed to improve learning efficiency by decomposing the Q-value function into two separate streams: one estimating the state value and the other estimating the advantage of each action. The update equation for the dueling network is:

$$Q(s,a;\theta) = V(s;\theta) + \left(A(s,a;\theta) - \frac{1}{|\mathcal{A}|} \sum_{a' \in |\mathcal{A}|} A(s,a';\theta) \right) \tag{Type-1}$$

Where $Q(s,a;\theta)$ represents the dueling Q-function with parameters θ.

Following is another way to estimate the Q-values:

$$Q(s,a;\theta) = V(s;\theta) + \left(A(s,a;\theta) - \max_{a' \in |\mathcal{A}|} A(s,a';\theta) \right) \tag{Type-2}$$

Implement both update rules (Type-1) **&** (Type-2) **and compare their performance in both the environments.**

2.2 Monte-Carlo REINFORCE

The MC-REINFORCE (Chapter 13) algorithm utilizes Monte Carlo sampling to estimate gradients for policy optimization. The update equation of its policy parameter θ is given by

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \alpha G_t \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})} \qquad \text{(w/o Baseline)}$$

In the presence of baseline, $V(\cdot; \boldsymbol{\Phi})$, the update equation is given by

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \alpha(G_t - V(S_t; \boldsymbol{\Phi})) \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})} \qquad \text{(w/ Baseline)}$$

The baseline $V(\cdot; \boldsymbol{\Phi})$ is updated by TD(0) method.

Implement MC REINFORCE with both update methods ((w/o Baseline) **&** (w/ Baseline)**) and compare their performance in both the environments.**

3 Instructions

We expect a comprehensive report that involves the following details:

- Snippets of the important parts of the code
- Four results plots (2 environments $\times$ 2 algorithms) *(Ex. Plot 1 should compare* (Type-1) *&* (Type-2) *Dueling DQN in Acrobot environment)*
- Inferences and conjectures from all your experiments and results
- Github link to the code

You are required to compare each algorithm with it's own variant ((Type-1) vs (Type-2)) and not with the other algorithm. Please adhere strictly to the following instructions.

- Use $\gamma = 0.99$ for all experiments
- Tune the hyper-parameter to minimize the regret in all experiments
- To account for stochasticity, use the average of 5 random seeds for each experiment/plot

- Plot the episodic return versus episodic number for every experiment
- The plots should consist the mean and variance across the 5 runs/seeds (Sample plot 3)

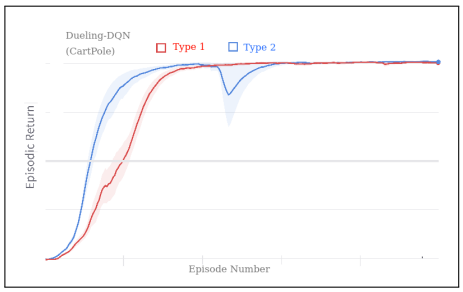

Figure 1: Sample plot of Dueling-DQN in CartPole. We expect a total of 4 such plots, one for each comparision

- Upload the corresponding code to a private repository in Github and attach the link in the report
- **Please strictly follow the academic code of conduct. Plagiarism will be penalized**

Please refer to the report-outline in Gradescope for the report structure. We expect one submission per group of 2 members.