All the questions are mandatory unless otherwise stated. For submission instructions please refer to the website and the instructions file.

1 Theory

Question 1. [30 points] **Off-policy Reinforcement Learning**

Off-policy learning is an important concept in reinforcement learning. In this setting, an agent follows a *behavior policy* μ to gather data. Using this data, the agent evaluates another policy called the *target policy* π. This setting has practical value as the agent can ideally evaluate any policy by following another policy. More precisely, this idea is critical in many learning frameworks such as options, counterfactual reasoning, etc.

We learned in the class that off-policy methods could be unstable in certain settings. Precisely, when we employ bootstrapping with function approximation, the updates can diverge. This setting is called the *deadly triad* and it was first described in (Baird, 1995) with a counter example.

The reason for the difficulty of off-policy learning is that the behavior policy may take the process to a distribution of states different from that which would be encountered under the target policy, yet the states might appear to be the similar due to function approximation. To correct for the different state distribution, let us introduce importance sampling ratios to re-weight the states encountered. Specifically, we define our updates to the weights θ of the linear function approximator at time t as below,

$$\Delta_t = \alpha(R_t^\lambda - \theta^T \varphi_t)\varphi_t \rho_1 \rho_2 \dots \rho_t \tag{1}$$

where α is the learning rate, R_t^λ is the importance sampling corrected λ returns, φ_t is the feature vector of the state at time t, ρ_i is the importance sampling ratio at time i.

i Show that

$$\mathbb{E}_\mu[\Delta\tilde{\theta}|s_0, a_0] = \mathbb{E}_\pi[\Delta\theta|s_0, a_0], \forall s_0 \in S, a_0 \in A \tag{2}$$

where $\Delta\theta$ and $\Delta\tilde{\theta}$ are the sum of the parameter increments over an episode under on-policy TD(λ) and importance sampled TD(λ) respectively, assuming that the starting weight vector is θ in both cases. S and A are state and action spaces and are finite.

ii Derive an equivalent online algorithm for the update rule in Eq. 2 using traces. You may assume that the updates are done at the end of the episode for this question.

Question 2. [30 points] **Paper Reading** *(Choose one of the following three tracks)*

A. *Experience replay* is a commonly used technique with deep neural networks to stabilize the training. The *experience replay* stores the agent's experience at each time step in a replay memory (or a buffer) that is accessed to perform the weight updates. This is precisely what is done in DQN.

i Highlight and explain in detail what are the advantages of using Q-learning with experience replay.

ii Explain the idea of prioritized experience replay (Schaul et al., 2015).

iii Highlight and compare the pros and cons of using prioritized experience replay over original experience replay.

B. Sutton et al., 1999 show that the gradient of the policy can be estimated from experience and combined with an approximate action-value function. Using this result, they prove the first convergent version of policy iteration with function approximation.

i Explain the gains of having a parametrized policy.

ii Summarize the possible agent objectives considered in the paper, showcasing the differences between them.

iii Explain the difference between REINFORCE and the result from the paper. Explain the motivation for the policy gradient theorem and the major steps in the proof.

iv What is a compatible value function? Explain why the policy gradient theorem works with an approximation of the value function.

v Why is the action-value function approximated actually an advantage function? Discuss the use of baselines and the effect on variance.

vi Explain why policy iteration with function approximation works.

2 Coding

In addition to the instructions provided for each question, please make sure you abide by the following general rules of reproducible and meaningful research:

- When applicable please do a thorough hyperparameter search. A detailed report of the hyperparameter search applicable to the algorithm has to be included in the submission *(Justify the choice of hyperparameters with supporting plots).*
- Each experiment has to be run on different seeds as specified in each individual question. Mean curve along with variance should be highlighted in the plots.

For more information regarding reproducibility and reporting of results, please refer to the instructions file posted along with the assignment.

Question 1. [20 points] **Baird's counterexample.** Consider the episodic seven-state, two-action MDP shown in Figure 11.1 in Sutton and Barto. The MDP consists of 6 non-terminal states and 1 terminal state. The dashed action takes the system to one of the six upper states with equal probability, whereas the solid action takes the system to the seventh state.

The behavior policy b selects the dashed and solid actions with probabilities 6/7 and 1/7 , so that the next-state distribution under it is uniform (the same for all non-terminal states), which is also the starting distribution for each episode. The target policy π always takes the solid action, and so the on-policy distribution (for π) is concentrated in the seventh state. The reward is zero on all transitions. The discount rate γ is 0.99. Consider estimating the state-value under the linear parameterization indicated by the expression shown in each state circle in Fig 11.1 in Sutton and Barto.

- Implement the Semi-gradient TD(0) algorithm for this MDP. The goal here is to perform policy evaluation: that is, to estimate the value function of the policy being followed. The value of each state is to be approximated as a linear combination of features.
- Plot the evolution of the components of the parameter vector w of the Semi-gradient Off-Policy TD(0) algorithm . The step size is to be set to 0.01, and the initial weights are to considered $w = (1, 1, 1, 1, 1, 1, 10, 1)^T$. You should see a figure like Fig 11.2. In essence, in this question you are asked to reproduce the demonstration of instability on Baird's counterexample.
- What do you observe? Write a brief summary of your observations.

Question 2. [20 points] **Policy Gradient/Actor-Critic Methods** *(Choose one of the following two tracks)*

A. Use the cartpole environment from Open AI gym (Brockman et al., 2016). Set the discount factor γ to 0.9.

- Implement the REINFORCE algorithm (Williams, 1992) with a two layered MLP to learn the task.
- Present your results in a plot, showing on the x-axis the episode count and on the y-axis the episode length (train until convergence). Run on 5 different seeds/runs. The resultant plots will be averaged over these 5 independent seeds. Also show the confidence interval (using the standard deviation obtained from the seeds/runs - you may use fill_between from matplotlib).
- Implement actor-critic method. Use critic network along with the policy network to learn the optimal behaviour. In critic network, the value function is learnt through bootstrapping. Compare the actor-critic performance against REINFORCE in a single plot.
- What do you observe? Write a brief summary of your observations.

B. Use the mountain car and cartpole environments from Open AI gym (Brockman et al., 2016). Set the discount factor γ to 0.9.

- Implement the Advantage Actor Critic (A2C) algorithm (Mnih et al., 2016) with a two layered Neural Network to learn the task. Use the A2C version, explained here (A2C), instead of the original asynchronous one.
- Present your results in a plot showing on the x-axis should the episode count and on the y-axis the episode length (train until convergence). Run on 5 different seeds/runs. The resultant plots will be averaged over these 5 independent seeds. Also show the confidence interval (using the standard deviation obtained from the seeds/runs - you may use fill_between from matplotlib).
- Write a brief summary of your observations.

References

(A2C). URL: %5Curl%7Bhttps://openai.com/blog/baselines-acktr-a2c/%7D.

Baird, Leemon (1995). "Residual algorithms: Reinforcement learning with function approximation". In: *Machine Learning Proceedings 1995*. Elsevier, pp. 30–37.

Brockman, Greg et al. (2016). *OpenAI Gym*. eprint: arXiv:1606.01540.

Mnih, Volodymyr et al. (2016). "Asynchronous Methods for Deep Reinforcement Learning". In: *CoRR* abs/1602.01783. arXiv: 1602.01783. URL: http://arxiv.org/abs/1602.01783.

Schaul, Tom et al. (2015). "Prioritized experience replay". In: *arXiv preprint arXiv:1511.05952*.

Sutton, Richard S. et al. (1999). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. NIPS'99. Denver, CO: MIT Press, pp. 1057–1063.

Williams, Ronald J. (May 1992). "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning". In: *Mach. Learn.* 8.3–4, pp. 229–256. ISSN: 0885-6125. DOI: 10.1007/BF00992696. URL: https://doi.org/10.1007/BF00992696.