CS533/CS534: Reinforcement Learning

Lab Assignment 4

Project Assignment

Due Date: 24th April 2023 11:59 PM | Max. Marks: 100

Instructions:

1. This lab assignment is for those students who are not doing the term project.
2. This lab assignment needs to be done individually and it carries 50% weightage for CS543 and 20% weightage for CS533.
3. Start by going through the research paper first before you start coding.
4. For queries, you should comment on Google Classroom. Alternatively, you may email the course TAs or course instructors.
5. All code must be written using a Python notebook (preferably using google colab).
6. Submission must be done in Google Classroom. The code as well as the accompanying observations should be made part of the colab notebook.
7. Code Readability is very important. Modularize your code by making use of classes and functions that can be flexibly reused wherever necessary. Also, use self-explanatory variable names and add comments to describe your approach wherever necessary.
8. Students are expected to follow the honor code of the class. Discussions and interactions with your classmates to take help in this assignment is not allowed.
9. Use the following naming convention: Name_rollnumber_lab4.ipynb for submission.

Meta-Learning for Reinforcement Learning

This assignment is based on the following research article:

Clavera, Ignasi, Jonas Rothfuss, John Schulman, Yasuhiro Fujita, Tamim Asfour, and Pieter Abbeel. "Model-based reinforcement learning via meta-policy optimization." In Conference on Robot Learning, pp. 617-629. PMLR, 2018.

Model-based reinforcement learning approaches carry the promise of being data efficient. However, due to challenges in learning dynamics-model that sufficiently match real-world dynamics, they struggle to achieve the same asymptotic performance as model-free methods. The authors propose Model-Based Meta-Policy-Optimization (MB-MPO), an approach that foregoes the strong reliance on accurate learned dynamics models. Using an ensemble of learned dynamic models, MB-MPO meta-learns a policy that can quickly adapt to any model in the ensemble with one policy gradient step. This steers the meta-policy towards internalizing consistent dynamics predictions among the ensemble while shifting the burden of behaving optimally w.r.t. the model discrepancies towards the adaptation step.

The paper and the code are provided as part of this assignment.

- The paper is available at https://arxiv.org/pdf/1809.05214v1.pdf
- The code is available at https://github.com/ray-project/ray/tree/master/rllib (see for MB-MPO)
- https://github.com/jonasrothfuss/model_ensemble_meta_learning

You are allowed to use the available code of this paper to complete this assignment.

The Tasks

Q.1. Reproduce the results given in Section 6.1 of the paper for three gym environments namely, Ant, HalfCheetah and Walker 2D.

Are you able to get the same results as reported by the authors? Comment on the challenges faced. List all the hyper-parameter values that authors didn't mention and you need to make your own choices for the same. Any new insights from the paper will earn additional bonus marks. [**20 marks**]

Q.2. Transfer Learning in RL : [**35 marks**]

Let's consider transferring a policy learned in one environment to another environment with a different MDP. Can we have a single backbone to learn the representations of any policy? Such that a model trained in one environment (say Ant) should train in the other environment (half cheetah) with less number of samples? You need to think in this direction and show if the concepts of MB-MPO can be used to also show policy transfer in RL.

One possible way could be to use the same model parameters/network weights learnt in the Ant environment and run the trained model in the HalfCheetah environment. Which algorithm performs the best, analyze the results using plots and share your insights?

Q.3. Federated RL: [**45 marks**]

Now consider the scenario where there are multiple agents distributed across different machines/servers all trying to learn the same policy for a particular environment without sharing their samples with each other. Such a privacy preserving setup is called Federated Reinforcement Learning (for more details see here https://arxiv.org/abs/1901.08277).

Run multiple copies of the best performing algorithm on the Ant and HalfCheetah environments with slight changes in parameters that are `reset_noise_scale` and `ctrl_cost_weight` for HalfCheetah and `reset_noise_scale`, `ctrl_cost_weight and contact_cost_weight` for Ant.

Can we transform the MB-MPO approach to a privacy preserving Federated setting where the meta policy is learned at the global level without sharing the samples generated by each individual agent? Propose your approach and share your insights on the results.

Note: Many of the training tasks could be very complex and hence you may need to make adjustments in the training process or number of iterations to arrive at some conclusions well before a convergence can occur.

****** All the Best********