Reproducibility in Evaluating Reinforcement Learning Algorithms

Khimya Khetarpal* Zafarali Ahmed* Andre Cianflone Riashat Islam Joelle Pineau
Reasoning and Learning Lab Mila, McGill University

TLDR: We highlight challenges in comparing RL algorithms in terms of evaluation and propose an evaluation pipeline decoupled from training code.

Why is comparing results in reinforcement learning difficult?

Implementation Details

- Libraries have different quirks for implementing.
- Details can cause massive performance differences [2, 3].
- Differ from algorithm description.
- Optimization algorithm and policy coupled together.

Training Details

- Compute Power: Different labs have access to different amount of computer power.
- Number of rollouts used per iteration for updates. These can skew the learning curves that measure efficiency and rewards.

Evaluation Details

- Score / Discounted Return / Reward: Inconsistent measures of performance between results.
- Sample Efficiency: Sample efficiency is not a good measure of how good an algorithm performs unless training conditions are constant.
- Top Seeds / Best Seeds: Only reporting the best seeds found can skew results in your favour. [4]
- Stochasticity of policy: Explicitly stating if the policy used was stochastic or not.
- Environment start states: Some labs may not have access to the conditions of the environment that make evaluations unfair.

Moving toward standard evaluation pipelines

Algorithm

- A2C
- IMPALA
- REINFORCE
- DON
- A3C
- RAINBOW

Trained Agent

- .act(o)

Environment

Config Script

Training

Evaluation

Evaluation pipeline is detailed by the environment or a third party. [5]

Ensures consistency in:
- Number of and value of seeds [7]
- Metric to record

This allows papers to compare results on the evaluation phase in a fair way.

[1] Agent image from Wikimedia Commons

Paper available at:
https://openreview.net/forum?id=HJgAmITcgm
Example code available at:
https://github.com/kkhetarpal/prototype4evaluation