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

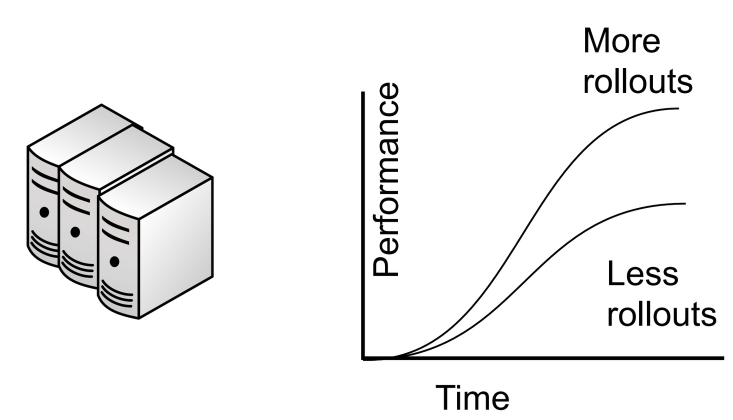
PyTorch TensorFlow Caffe

- Libraries have different quirks for implementing.
- details can cause massive performance differences [2, 3]
- differs 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.

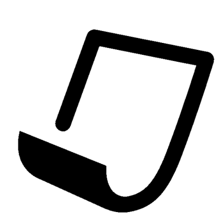
Sample Efficiency
Sample efficiency is not a good measure of how good an algorithm performs unless training conditions are constant.

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



Config Script

Environment

Training

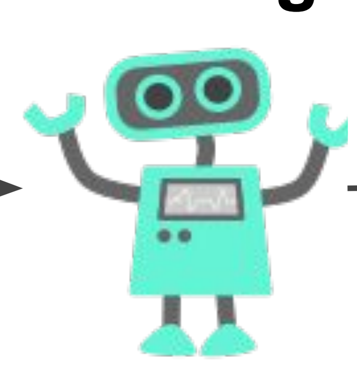
Evaluation

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

Algorithm

A2C IMPALA
REINFORCE DQN
A3C RAINBOW

Trained Agent



.act(o)

Evaluation Pipeline

Hyperparameters are released to allow fair training and comparison. [6]

Agent exposes an "act" function which takes an observation and returns an action in a **framework independent** way!

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
[2] Henderson et al. "Deep reinforcement learning that matters". 2018
[3] Tucker et al. "The Mirage of Action-Dependent Baselines in Reinforcement Learning". 2018
[4] Shimon et al. "Protecting against evaluation overfitting in empirical reinforcement learning." 2011.
[5] Bellemare et al. "The arcade learning environment: An evaluation platform for general agents." 2013
[6] Riedmiller et al. "Evaluation of policy gradient methods and variants on the cart-pole benchmark." 2007.
[7] Zhang et al. "A Study on Overfitting in Deep Reinforcement Learning." 2018

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