Experience Replay Optimization
Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou and Xia Hu
Department of Computer Science and Engineering, Texas A&M University
{daochen.zha, khlai037, zkxiong, xiahu}@tamu.edu

Introduction

- Experience replay is a memory buffer that stores past transitions (experiences) which are replayed for later use.
- It is a key technique behind off-policy RL algorithms. It greatly stabilizes the training and improves the sample efficiency.
- Uniform sampling is usually adopted. However, not all transitions are of equal importance: the agent can learn more efficiently from some experiences than from others.
- Some rule-based replay strategies have been studied. However, they may not be able to adapt to different tasks and algorithms.

Motivation

- Humans tend to replay the memories that will lead to the most rewarding future decisions.
- We are motivated to use the feedback from the environment as a rewarding signal to adjust the replay strategy.

Methodology

Sampling with Replay Policy: the replay policy is described as a priority score function \( \phi(B; \theta') \in (0, 1) \), in which higher value indicates higher probability of a transition being replayed. We maintain a vector \( \lambda \) to store these scores:

\[ \lambda = \{ \phi(B; \theta') | B \in \mathcal{B} \} \in \mathbb{R}^N. \tag{1} \]

In training, we then sample a subset \( \mathcal{B}' \) according to

\[ I \sim \text{Bernoulli}(\lambda), \]
\[ \mathcal{B}' = \{ B | B \in \mathcal{B} \wedge I = 1 \}. \tag{2} \]

Then \( \mathcal{B}' \) is used to update the agent with standard procedures.

Training with Policy Gradient: The reward-reward is defined as the improvement of the cumulative reward:

\[ r' = r_x - r_{t'} \]

By using the REINFORCE trick, we can calculate the gradient of the improvement \( \mathcal{J} \) w.r.t \( \theta' \):

\[ \nabla_{\theta'} \mathcal{J} = \nabla_{\theta'} \mathbb{E}[r'] = \mathbb{E}[r' \nabla_{\theta'} \log \mathbb{P}(I|\phi)]. \tag{4} \]

The resulting policy gradient can be written as:

\[ \nabla_{\phi} \mathcal{J} \approx \frac{1}{N} \sum_{i=1}^{N} \nabla_{\phi} [I \log \phi + (1 - I) \log(1 - \phi)]. \tag{5} \]

Experiments

Performance Comparison

Efficiency Evaluation

Replay Policy Analysis

Observations

- The learned replay policy of ERO samples more transitions with low TD errors in HalfCheetah. More studies are needed to understand this aspect in the future work.
- ERO samples more recent transitions than Vanilla-DDPG. This suggests that recent transitions may be more helpful in this specific task.

Acknowledgements

The work is, in part, supported by National Science Foundation (#IIS-1718840 and #IIS-1750074).