A Study of Experience Replay in Reinforcement Learning

Masoumeh Bakhtiariziabari, Tarun Krishna, Dhruba Pujary, Thomas A. Unger

1. Research Question

In Reinforcement Learning, a neural network agent can learn the most rewarding actions by sampling transitions (i.e., interactions) from the environment. However, **successive transitions are correlated**. This violates the i.i.d. assumption and causes the "**catastrophic forgetting**" of earlier interactions.

With **Experience Replay** (**ER**), the i.i.d. assumption is revalidated by storing interactions in a **memory buffer** and training the neural network with random samples from this buffer.

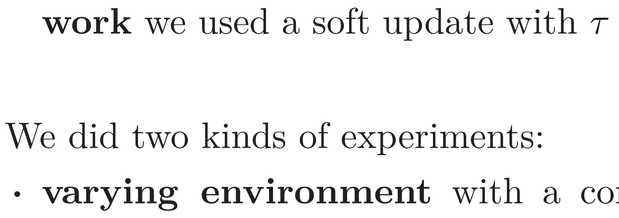
We study, *what is the effect of different types of ER and different memory buffer sizes in different OpenAI Gym environments?*

3. Experimental Setup

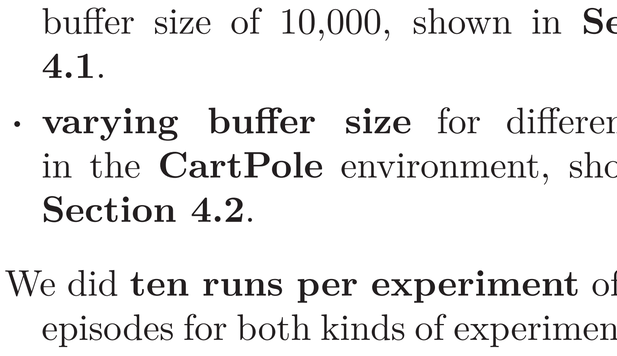
- For environments, we used **CartPole-v1**, **MountainCar-v0** and **LunarLander-v2** from the OpenAI Gym.
- We used a **DQN** with two hidden layers and ReLU non-linearities. For the **target network** we used a soft update with $\tau = 0.1$.
- We did two kinds of experiments:
  - **varying environment** with a constant buffer size of 10,000, shown in **Section 4.1**.
  - **varying buffer size** for different ER in the **CartPole** environment, shown in **Section 4.2**.
- We did **ten runs per experiment** of 1,000 episodes for both kinds of experiments and plot their means and standard deviation.

4.1 Environment Experiments

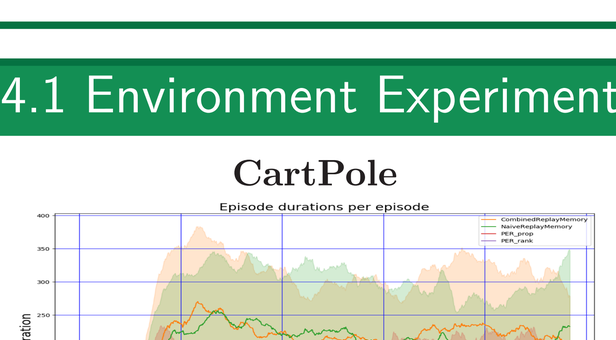
CartPole



MountainCar



LunarLander



Note: for the LunarLander environment, the y-axis corresponds to rewards instead of episode lengths.

2. Types of Experience Replay

- **Naive ER** (used as **baseline**) samples a mini batch from the memory buffer **uniformly**.
- **Prioritized ER** (**PER**)*[1] samples a mini batch from the memory buffer with a **bias towards** samples where the agent makes a **large prediction error** δ. This is meant to make learning more efficient. The probability of sampling transition i is given by:

$$P_i = \frac{p_i^a}{\sum_k p_k^a}$$

where $p_i > 0$ is the priority of the transition and α determines how much prioritization is used. There are two variants of PER:

Proportional[1],[2] PER, where the sampling probability of sample i is proportional to: $\mathbf{p_i = |\delta_i| + \epsilon}$, where δ_i is the sample's TD error and ϵ prevents the edge-case of transitions not being revisited once their error is zero.

Rank[1] based PER, which ranks samples by δ_i and uses their rank to determine their sampling probabilities:

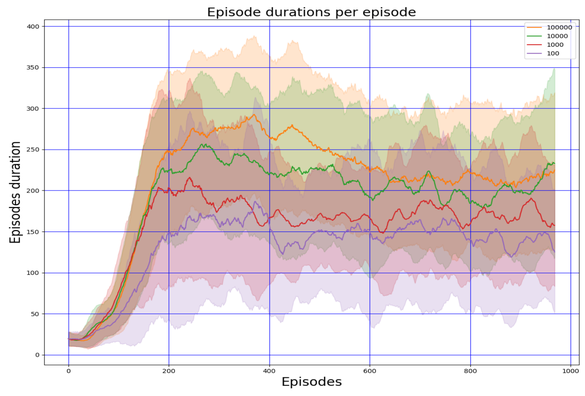
$$p_i = \frac{1}{\text{rank}(i)}, \text{ where rank}(i) = \text{rank of transition } i \text{ according to } |\delta|$$

- **Combined ER** (**CER**)[3] samples from the memory buffer uniformly, but **guarantees** the inclusion of the **latest transition**. This is meant to remedy the negative influence of a large replay buffer, where it may take a long time to learn from newer situations.

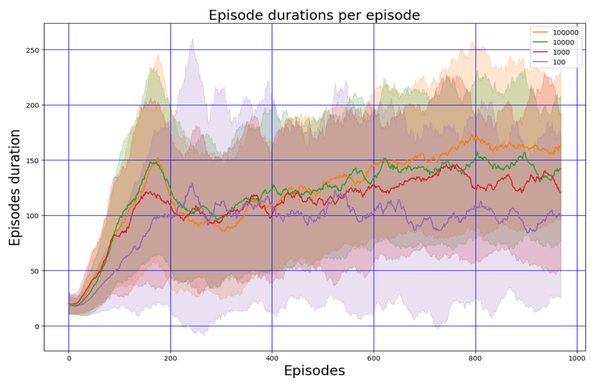
*__Note:__ For PER we used importance sampling to overcome bias and for stability reasons we normalized weights by scaling the updates downward.

4.2 Buffer Experiments

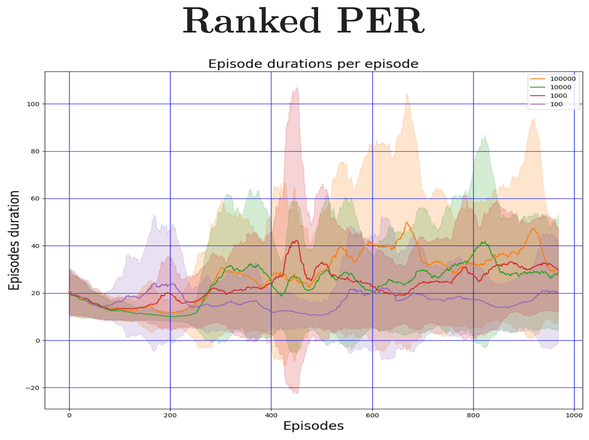
Naive ER



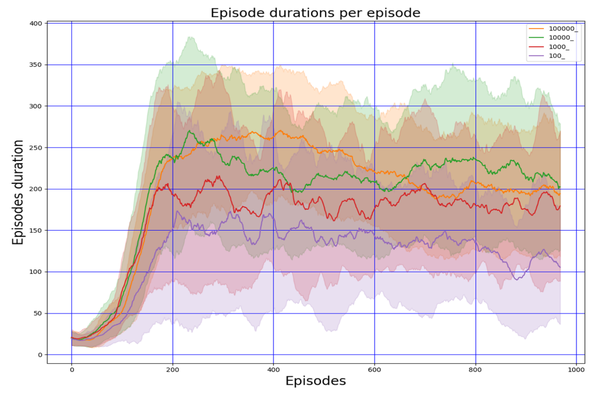
Proportional PER



Ranked PER



CER



Results of experiments with various buffer sizes in the CartPole environment.

5. Conclusions

- In the **CartPole** environment:
  - Any advantage of a larger memory buffer is small. A **larger memory buffer** tends to lead to **better results**, but the variability is relatively large.
  - Naive ER and CER perform about equally.
  - Both variations of PER consistently underperform.
- In the **MountainCar** environment, there is no consistent difference between Naive ER, CER and proportional PER.
- In the **LunarLander** environment, both CER and Naive ER perform better than both types of PER not only on average, but also in terms of variability.
- There is **not one type of ER that is consistently better** than the others. However, in all the environments where we tested it, **ranked PER consistently under performed** compared to the others.

6. References

[1] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *CoRR*, abs/1511.05952, 2015.

[2] `https://github.com/wotmd5731/dqn/blob/master/memory.py`.

[3] Shangtong Zhang and Richard S. Sutton. A deeper look at experience replay. *CoRR*, abs/1712.01275, 2017.

[4] Ruishan Liu and James Zou. The effects of memory replay in reinforcement learning. *arXiv preprint arXiv:1710.06574*, 2017.