Learning from failure, or/and when surprised: Can a combination of HER and PER work for sparse reward environments?

Gourab Dey
Department of Computer Science, Stony Brook University
gdey@cs.stonybrook.edu

I. INTRODUCTION

Recent works[1] in Offline Reinforcement Learning have shown how to stabilize the training of a value function, represented by a deep neural network, by using experience replay. However, dealing with sparse reward environments is still one of the biggest challenges, where with the absence of a feedback signal, it becomes extremely difficult for the agent to learn effectively. Advanced replay buffer techniques like Hindsight Experience Replay (HER)[2] aim to solve this problem by learning almost as much from achieving an undesired outcome as from the desired one. The pivotal idea behind HER is to replay each episode with a different goal than the one the agent was trying to achieve, e.g. one of the states which was achieved in the episode. However, HER samples transitions from the replay buffer randomly, at the same frequency that they were originally experienced, regardless of their significance, which might lead to longer training time and inefficient learning.

In this paper, I try to reproduce the results in[2] in a single-goal RL setting using the Mountain Car environment. I explore different goal strategies and how HER interacts with reward shaping. More importantly, I propose a new technique Hindsight Prioritized Experience Replay (HPER) which combines HER with Prioritized Experience Replay (PER)[3] to prioritize which transitions to replay from the buffer based on their expected learning progress. I also explore if vanilla PER can be applied to sparse reward settings and evaluate the performance of each approach.

In the following sections, I will first briefly provide details on the literature survey, then illustrate my method, and at last present experimental results and conclusion.

II. LITERATURE SURVEY

A. Hindsight Experience Replay

Hindsight Experience Replay[2] allows sample-efficient learning from rewards which are sparse and binary by re-examining trajectories with a different goal. Given an episode with a state sequence $s_1, ..., s_T$ and a goal $g \neq s_1, ..., s_T$ which implies that the agent received a reward of -1 at every time step - HER harvests this information by using an off-policy RL algorithm and experience replay where they replace g in the replay buffer by s_T . In addition they also replay with the original goal g left intact in the replay buffer. With this modification at least half of the replayed trajectories contain rewards different from -1 and learning becomes much simpler. However, transitions are still replayed from the buffer randomly without assigning any priority.

B. Prioritized Experience Replay

PER investigates how prioritizing which transitions are replayed can make experience replay more efficient and effective than if all transitions are replayed uniformly. Transitions are prioritized based on their expected learning progress measured by the magnitude of their temporal-difference (TD) error. In my work, I also investigate whether PER performs better for sparse reward environments than a vanilla DQN.

III. METHODOLOGY

The DQN consists of 2 fully connected layers of 128 neurons each, with a replay buffer of size 10000 and with a sampling batch of size 32. This configuration performed the best after experimenting with different hyperparameters.

A. Hindsight Experience Replay in a single-goal RL setting

I experimented with different goal strategies, and reward functions to investigate which one worked the best:

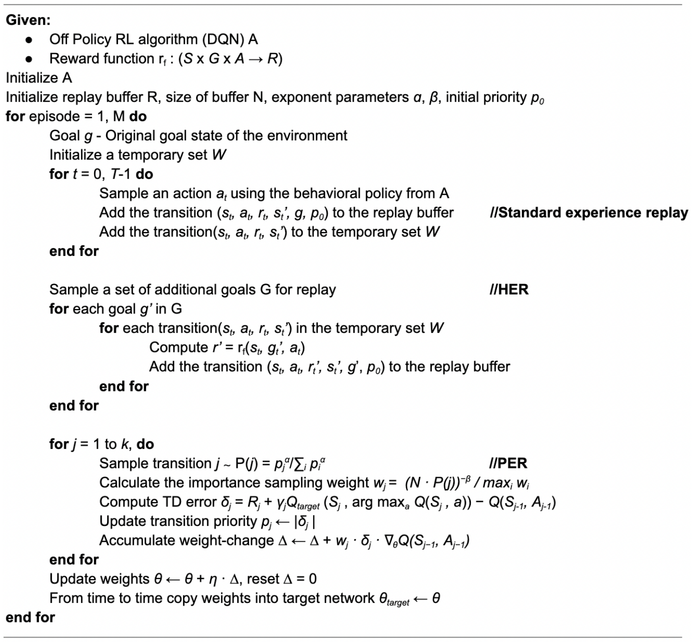
- Experiment 1
  - Goal strategy: $k - future(k = 4)$
  - No reward shaping, $r(s,a) = \begin{cases} 0 & \text{if } s' = g' \\ -1 & otherwise \end{cases}$
- Experiment 2
  - Goal strategy: $final$
  - Reward shaping, $r(s,a) = \begin{cases} 0 & \text{if } s'_x \leq g'_x \\ -1 & otherwise \end{cases}$

B. Prioritized Experience Replay

To explore how PER can be used for sparse reward environments, I investigate with different hyper-parameters, mainly the exploration rate decay, and the exponent parameters α and β. The motivation to use PER for sparse reward environments is that given a high exploration rate in the initial episodes, the PER algorithm will sample those transitions first which reach the goal state as they will have a high TD error.

C. HPER Algorithm

I present a novel algorithm HPER which combines both HER and PER to incorporate learning from both failure and surprise. The detailed algorithm is listed below:



```
Given:
  • Off Policy RL algorithm (DQN) A
  • Reward function $r_f : (S \times G \times A \rightarrow R)$
Initialize A
Initialize replay buffer R, size of buffer N, exponent parameters $\alpha$, $\beta$, initial priority $p_0$
for episode = 1, M do
    Goal $g$ - Original goal state of the environment
    Initialize a temporary set $W$
    for $t = 0$, $T$-1 do
        Sample an action $a_t$ using the behavioral policy from A
        Add the transition $(s_t, a_t, r_t, s_t', g, p_0)$ to the replay buffer      //Standard experience replay
        Add the transition$(s_t, a_t, r_t, s_t')$ to the temporary set $W$
    end for

    Sample a set of additional goals G for replay                                   //HER
    for each goal $g'$ in G
        for each transition$(s_t, a_t, r_t, s_t')$ in the temporary set $W$
            Compute $r' = r_f(s_t, g_t', a_t)$
            Add the transition $(s_t, a_t, r_t', s_t', g', p_0)$ to the replay buffer
        end for
    end for

    for $j$ = 1 to $k$, do
        Sample transition $j \sim P(j) = p_j^\alpha / \sum_i p_i^\alpha$                  //PER
        Calculate the importance sampling weight $w_j = (N \cdot P(j))^{-\beta} / max_i \, w_i$
        Compute TD error $\delta_j = R_j + \gamma_j Q_{target} (S_j , \arg\max_a Q(S_j , a)) - Q(S_{j-1}, A_{j-1})$
        Update transition priority $p_j \leftarrow |\delta_j|$
        Accumulate weight-change $\Delta \leftarrow \Delta + w_j \cdot \delta_j \cdot \nabla_\theta Q(S_{j-1}, A_{j-1})$
    end for
    Update weights $\theta \leftarrow \theta + \eta \cdot \Delta$, reset $\Delta = 0$
    From time to time copy weights into target network $\theta_{target} \leftarrow \theta$
end for
```

Fig. 1. HPER algorithm

IV. RESULTS AND ANALYSIS

Below are the results and analysis of each algorithm:

- HindSight Experience Replay

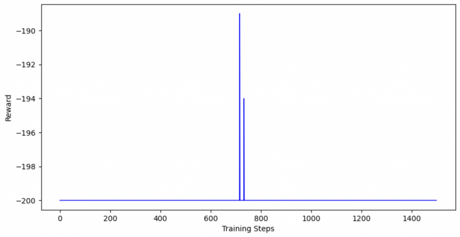


Fig. 2. k-future with no reward shaping

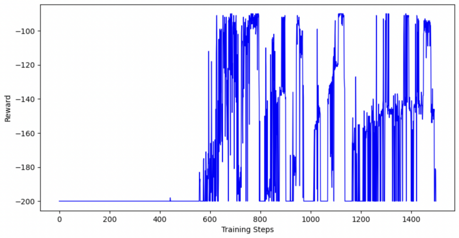


Fig. 3. final with reward shaping

Analysis: Both goal strategy and reward shaping play a major role in HER. The original strategy k-future as suggested in the HER paper performs way worse than the strategy final with reward shaping for this environment.

- Prioritized Experience Replay

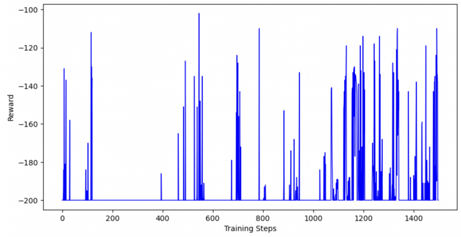


Fig. 4. Exploration decay = 0.999, α = 0.6, β = 0.4

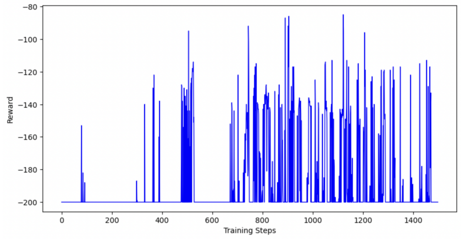


Fig. 5. Exploration decay = 0.9999, α = 0.6, β = 0.4

Analysis: PER performs better when the exploration rate is higher in initial episodes, the agent is able to perform better and accumulate better rewards later.

- HPER

  Below are the results of the HPER algorithm for different initialization seeds.

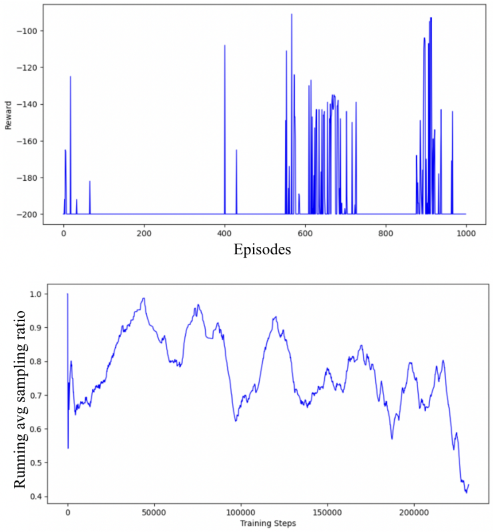


Fig. 6. Seed = 0, Rewards and the running mean of actual goals sampled

Analysis: The initial results of HPER were promising as shown in Fig 6, the agent starts accumulating high

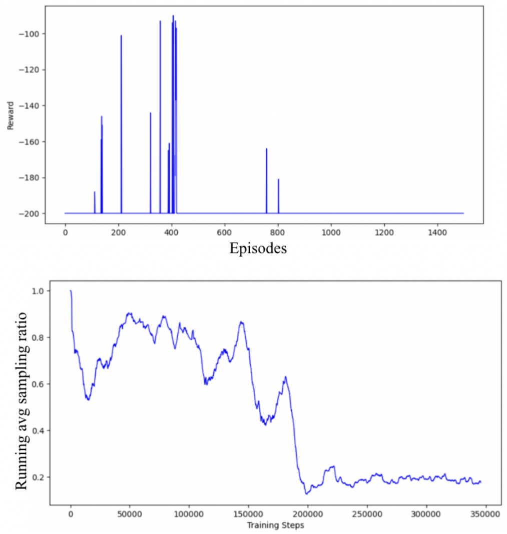


Fig. 7. Seed = 42, Rewards and the running mean of actual goals sampled

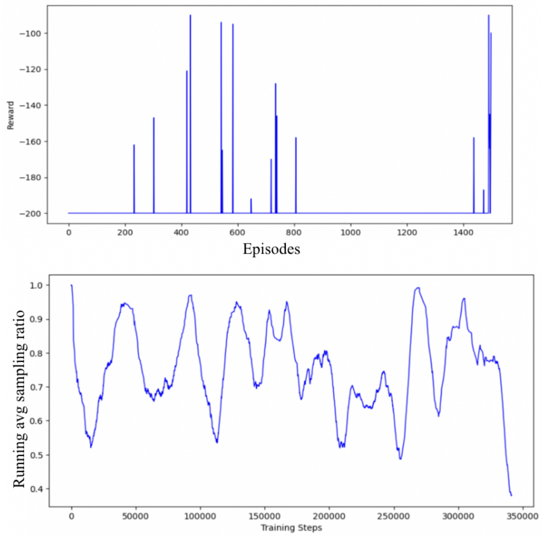


Fig. 8. Seed = 100, Rewards and the running mean of actual goals sampled

rewards within a few episodes. However, it seems that HPER is pretty unstable as shown in Fig 7 and Fig 8, and it seems that it is highly dependent on the percentage of actual goals sampled from the buffer. The replay buffer consists of both the actual goals and the HER goals now, and as seen in Fig 7, when the actual goals sampled are high (from episode 0 to episode 800), the agent learns effectively. However as the fraction of actual goals drop, the agent starts performing worse (episode 800 to episode 1500). This leads us to HPER 2.0 where I have used the sampling ratio (s_f), defined by:

$$s_f = \frac{\text{(Actual goals)}}{\text{(HER goals)}}$$

as a parameter while sampling transitions from the buffer.

- HPER 2.0

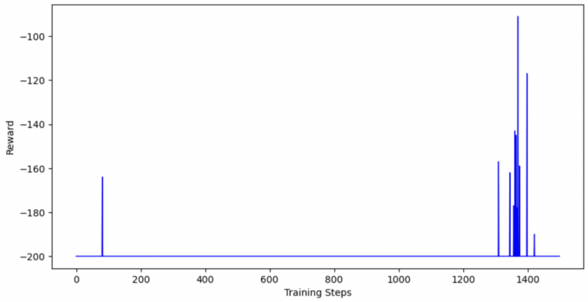


Fig. 9. Rewards accumulated over time

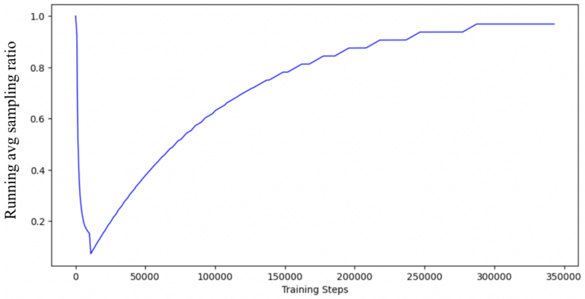


Fig. 10. Running mean of actual goals sampled

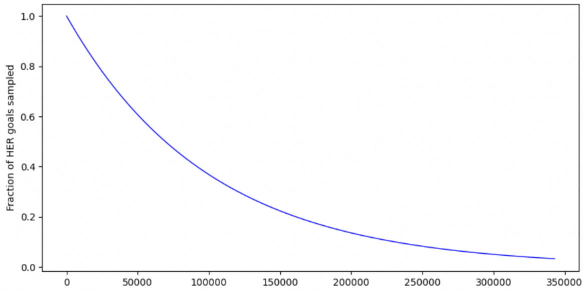


Fig. 11. HER goals sampled over time = $1/s_f$

Analysis: Preliminary results indicate that there is a clear correlation between s_f and the training performance of the algorithm (as the fraction of actual goals increase over time in Fig 10, the agent starts learning effectively as shown in Fig 9). The main idea behind sampling HER goals initially was, given that HER performs a type of curriculum learning implicitly, the agent would be able to learn from HER goals at first to reach the actual goal, and then PER would sample those transitions which have a high expected learning progress towards the actual goal. Further analysis, as to how to vary s_f needs to be done to comment more on the stability of HPER 2.0.

- Performance Analysis of each algorithm

Agent	Episodes for convergence	Rewards
Vanilla DQN	1500	~(-120)
DQN_HER	600	~(-100)
DQN_PER	800	~(-110)
DQN_HPER	800(Unstable)	~(-110)(Unstable)
DQN_HPER2.0	1300	~(-110)

V. Conclusion

In this paper, I analysed if advanced replay buffer techniques could be used in sparse reward environments. PER can be applied to sparse reward settings and it outperforms vanilla DQN implementation wrt. time required for convergence and rewards accumulated. HER outperforms both PER and vanilla DQN even with a very low exploration rate in initial episodes. However, care should be taken while devising the goal strategy and reward shaping. HPER is highly dependent on the sampling ratio s_f and future work will focus on identifying as to how to vary this parameter to learn efficiently.

References

[1] Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, Petersen, Stig, Beattie, Charles, Sadik, Amir, Antonoglou, Ioannis, King, Helen, Kumaran, Dharshan, Wierstra, Daan, Legg, Shane, and Hassabis, Demis. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

[2] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, Wojciech Zaremba. Hindsight Experience Replay. In Arxiv: https://arxiv.org/abs/1707.01495, 2018

[3] Tom Schaul, John Quan, Ioannis Antonoglou, David Silver. Prioritized Experience Replay. https://arxiv.org/abs/1511.05952, 2016

[4] Tom Schaul, Dan Horgan, Karol Gregor, David Silver. Universal Value Function Approximators. https://proceedings.mlr.press/v37/schaul15.pdf, 2015

[5] Egor Rotinov. Reverse Experience Replay. https://arxiv.org/abs/1910.08780, 2019