Can prioritized experience replay, hindsight experience replay outperform normal experience replay in different environments?

Jiun-Han Chen (11649712), Changxin Miao (11853018), Michael Mo (10770518), Yachu Yang (11571276)

Universiteit van Amsterdam

Motivation

Human beings develop their intelligence by learning from experience. Some experiences could be more valuable than others. Experience with unsuccessful outcome might be as significant as successful ones. Similarly, the learning process of the reinforcement learning algorithms could be facilitated with different experience replay strategies.

Research Question

Can prioritized experience replay, hindsight experience replay outperform normal experience replay in different environments?

Normal Experience Replay

Advantage
- Add randomization in the sampling phase and satisfy i.i.d assumption.

Disadvantage
- Consider all experiences are equivalently important.
- Unable to deal with sparse reward and require shaped rewards for efficient learning.

Prioritized Experience Replay

Not all experience replay are equally important. Instead of randomly sampling, setting priorities for experiences by TD-error, i.e., large TD-error with higher priority, and sampling memory by their priority can make the learning process more efficient.

1. Store replay memory with their **priority** $p_i = |i| + \varepsilon$. δ_i is **TD-error** and ε is small constant.
2. Sample k replay experiences for a minibatch according to their **stochastic prioritization: Transition j ~ P(j)** = $p_j^\alpha / \sum_i p_j^\alpha$ (α = 0 is equal to uniform sampling).
3. Compute **importance-sampling** weight w_j, to anneal the bias caused by PER: $w_j = (\frac{1}{N} \cdot \frac{1}{p_j})^\beta$ (β=1 fully compensates for the non-uniform P(j)).
4. Compute TD error δ_i, and update the priority in replay memory.
5. Accumulate weight-change in a minibatch: $\Delta \leftarrow \Delta\, w_j \delta_j \nabla_\theta Q(S, A)$
6. After iterating over a minibatch, update parameter weights: $\theta \leftarrow \theta + \eta\nabla$

Hindsight Experience Replay

Failure is just “not success yet”, and it can be our teacher. Setting and learning from some existing failed case can help you to achieve success.

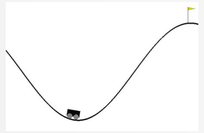
1. A **goal** g is set before each episode begins.
2. Define the reward function to make rewards **dependent on the goal** $r(s, a, g) = -[f_g(s) = 0]$
3. **Concatenate the state and goal** $(s||g)$, map it to an action, a $\leftarrow \pi(s||g)$
4. The episode with goal g, transitions and rewards $(s||g, a, r_g, s'||s)$are stored in the replay memory.
5. Furthermore, the same experience is now also used to “replay” virtual episodes **with different goals** $(r'_g = r(s, a, g'))$, thereby adding more additional transitions rewards $\left(s \middle|\middle| g, a, r_g, s' \middle|\middle| g\right)$ into the replay memory.

Goal selection strategies:
- Final: Choose the final state of the experienced episode as a goal.
- Future: Replay with k random states in the same episode after the transition.
- Episode: Replay with k random states in the same episode as the transition being replayed.
- Random: Replay with k random states encountered in the whole training procedure.

Experiment 1

Environment setups

MountainCar-V0:
- **Explicit goal** exists in the state space
- **Sparse reward**
- Reward is -1 each time step
- Reward is 0 and episode stops when goal is reached

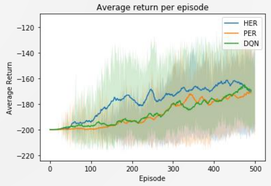


Model: Off-policy learning DQN

Parameters:
- 2 hidden layers with ReLU and 30 hidden neurons for each hidden layer.
  - Learning rate = 0.001
  - Memory size = 10000
  - ε starts from 1 and decreases at a rate of 0.95 until 0.05
- Retrain the model 30 times to get the average return, upper and lower bound per episode. 500 episodes in each training.

Experiment Result

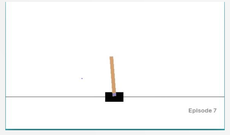
- No huge performance difference between PER and normal experience replay.
- Due to the sparse reward density, TD error and priority of most of states is similar to each other, which makes PER not significantly different with uniform sampling.
- HER demonstrates significant better performance. After 200 episodes, the average return is already around -170, while PER and DQN are still around -190.
- The car could receive relatively higher reward by passing the goal states, consequently leading to higher Q values.



Experiment 2

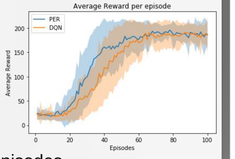
Experiment setups

Cartpole-V0
- **High** frequent reward
- Reward is +1 for each timestep



Model: Off-policy learning DQN

Parameters:
- 1 hidden layer with ReLU and 128 hidden neurons.
  - Same learning rate and ε as experiment 1.
- Discount factor is 0.8.
- Retrain the model 30 times to get the average return per episode.
  - 100 episodes in each training.



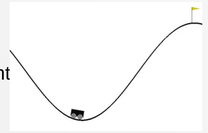
Experiment Result

- The influence of high variance is diminished with average.
- Curves are smooth and close to monotonically increase.
- To obtain the same reward, DQN with PER outperform NER with 10 episodes..
- PER is able to accelerate the training when the reward is easy to obtain.

Experiment 3

Environment setups

Same environment and model parameterization as in experiment
- Experimented memory size 10k / 60k



Different experience replay strategies

Goal selection strategies:
- Standard final, random, episode.
- Max: Goal is highest position achieved.
- Random, episode with k=4 goals.
- Left-Right: Goals are most left and most right positions achieved.

Experiment Result

- Different goal selection strategies
  - The figure depicts the average returns gained when applying different goal selection strategies with the memory size of 10k.
  - Various t strategies achieved similar results except for random goal selection strategies.
- Different memory sizes
  - No difference exists between DQN with memory size of 10k and 60k.
  - Only Random-4 strategy shows some difference.
  - Small memory size might cause experience to get lost from the memory more quickly, which is more obvious when multiple goals are replayed and stored in the buffer.

Conclusion

- PER and HER demonstrate an advantage on sparse rewards in paper but not in our experiment.
- PER still have potential to be applied to various environments to accelerate the training attributed to its limited constraint. The guidance form TD-error improves the quality of samples and importance sampling compensates the bias introduced by preferable sampling methods.
- When an ultimate goal is hard to define, PER outperforms HER and NER
- HER can let an agent learn more efficiently in an environment with really sparse reward than normal experience replay.
- HER gives better performance than DQN with standard ER in all used goal selection strategies. The goal selected for each HER should be chosen to maximize the outcome of that episode.
- Randomly selecting goals is thus likely to give less performance.

References
Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). Prioritized experience replay. arXiv preprint arXiv:1511.05952.
Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., ... & Zaremba, W. (2017). Hindsight experience replay. In Advances in Neural Information Processing Systems (pp. 5048-5058).

The project code is based on:
https://github.com/rlcode/per
https://github.com/YunqiuXu/HER_pytorch