Journal of KIIT. Vol. 19, No. 7, pp. 27-33, Jul. 31, 2021. pISSN 1598-8619. eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2021.19.7.27

Goal-based Target Network in Deep Q-Network with Hindsight Experience Replay

Chayoung Kim*

Abstract

Handling a sparse reward is one of the most significant challenges in Reinforcement Learning(RL), especially when we achieve human-level performances in a complex domain, such as grasping a moving object of robotic manipulation or preventing a corrupted byte in a cipher text. Deep Q-Network(DQN) with Hindsight Experience Replay(HER) is effective in dealing with sparse rewards, but it is not easy to get the advantages of on-line learning using target network, which has a fixed update. Therefore, in relation to sparse compensation, we propose a method of updating the target network based on the goal of HER in DQN. Since the Goal-based Target Network for HER in DQN proposed in this paper is updated every episode and every goal, it is more frequent and flexible so that the advantages of on-line learning can be utilized a little more. We evaluate the proposed Goal-based Target Network on a bit-flipping environment for preventing Byte-Flipping-Attack. The comparison demonstrates the superiority of our approach, showing that the proposed Goal-based Target Network is a better ingredient to enable the HER in DQN to solve tasks on the domain of the sparse reward.

요 약

희소 보상 처리는 강화 학습(RL)에서 가장 중요한 과제 중 하나이다. 특히 로봇 조작에 있어서 움직이는 물체를 잡으려하거나 암호 텍스트 분야에서 손상된 바이트가 생기지 않도록 하는 것과 같은 분야에서 인간 수준의 성과를 얻고자 할 때 더욱 그렇다. Hindsight Experience Replay(HER)를 사용하는 Deep Q-Network(DQN)은 희소 보상을 처리하는 데에 효과적이지만, 업데이트가 정해진 target network을 사용하므로 온라인 학습의 장점을 얻기 어렵다. 따라서 희소 보상과 관련하여 DQN에서 HER의 목표(Goal) 기반으로 Target Network을 업데이트하는 방법을 제안한다. 본 논문에서 제안하는 Goal-based Target Network은 에피소드와 목표마다 업데이트 하므로, 좀 더 자주 업데이트 하기 때문에 온라인 학습의 장점을 조금 더 이용할 수 있다. 그리고, Byte-Flipping-Attack 방지를 위해, 제안 된 Goal-based Target Network를 bit-flipping 환경에서 평가했다. 비교 결과는 본 논문에서 제안한 Goal-based Target Network가 DQN에서 HER을 사용할 때에 희소 보상 도메인에서 더 나은 요소임을 보여준다.

Keywords
hindsight experience replay, deep Q-network, reinforcement learning, target network, goal-based replay

* Assistant Professor, Div. of General Studies, Kyonggi University
- ORCID: https://orcid.org/0000-0002-4186-5882

· Received: Apr. 18, 2021, Revised: Jun. 21, 2021, Accepted: Jun. 24, 2021
· Corresponding Author: Chayoung Kim
Div. of General Studies, 154-42, Gwanggyosan-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, Korea, 16227.
Tel.: +82-31-249-9509, Email: kimcha0@kgu.ac.kr

Ⅰ. Introduction

Reinforcement learning(RL) algorithms have led to a big range of successes in learning policies for making sequential decision problems to maximize long-term utilities in an environment. A reinforcement learning combined with a deep neural network known as Deep Q-Network(DQN)[1] has been regarded as an effective framework for yielding human-level performance in a complex control problems such as Atari games. There are simulated environments, such as the game of Go[2], a robotic arm moving a puck onto a proper position[3]. In those simulated environment, there is a big challenge for training a RL agent. They need to engineer a complicated reward function that should be carefully shaped[4].

However, it is not easy to apply a complicated engineering because these engineering could hinder the applicability of RL in real-world problems because of RL domain-specific and expertise knowledge. So, it is essential to develop a DQN-variant algorithms which can learn from unshaped and sparse rewards. Recently, Hindsight Experience Replay(HER)[5] has shown a very efficient and effective performance, especially for an off-policy RL in solving goal-based works with sparse and unshaped binary rewards. In terms of off-policy with HER, the RL algorithms use replay buffers to train the agent by retrieving the experience memories with the sample efficiency in a goal setting. However, using replay buffers limits on-line learning, which can enable real-time learning. Moreover, a DQN uses a target network, which can introduce to stabilize training the agents. The target network is a copy of the estimated weights, which can be fixed for some number of steps for serving a stable target. However, the use of the target network can cause training the agent slowly and hinder on-line RL, which is a desirable attribute[6] because of delayed Q-function updates. It means both the use of experience replay and the target network are deviations from on-line learning.

Therefore, we propose a Goal-based Target Network in DQN with HER that increase the advantage of on-line learning in spite of the use of the target network, while still ensuring stable learning in sparse and binary reward domains. We compare the performances of the Goal-based Target Network with the original DQN with HER in an environment of bit-flipping for preventing Byte-Flipping-Attack[7]. Our empirical results show that the proposed Goal-based Target Network can achieve more interact-ability than the original DQN with HER.

The rest of the paper is organized as follows. In Section 2, there are background methods. And Section 3 describes the proposed algorithm, Goal-based Target Network in DQN with HER and the comparisons with the original DQN with HER. Finally, we conclude this work in Section 4.

Ⅱ. Background

In a bit-flipping environment in Fig. 1, there are the states, $S = \{0, 1\}^n$ and the actions, $A=\{0, 1, ..., n-1\}$ for an arbitrary integer n where executing the i^{th} action flips the i^{th} element of the state. In every execution, we sample not only an initial state but also a target state with uniformly random distribution and a policy can get -1 reward if so not in the target state.

HER in Fig. 2[5] is an efficient and effective technique of exploiting the replay buffer, especially in off-policy RL algorithms such as a DQN.

The idea of HER uses experience replay in an off-policy RL where the goal g can be replaced in the replay buffer. The replacement of the goal g helps the replayed trajectories receive rewards different from -1 and training an agent becomes simpler. So, every transition $s_t \rightarrow s_{t+1}$ along with the replaced goal for the episode is stored in the replay buffer. The failed experience with HER can be modified and stored in the replay buffer with the modified goal.

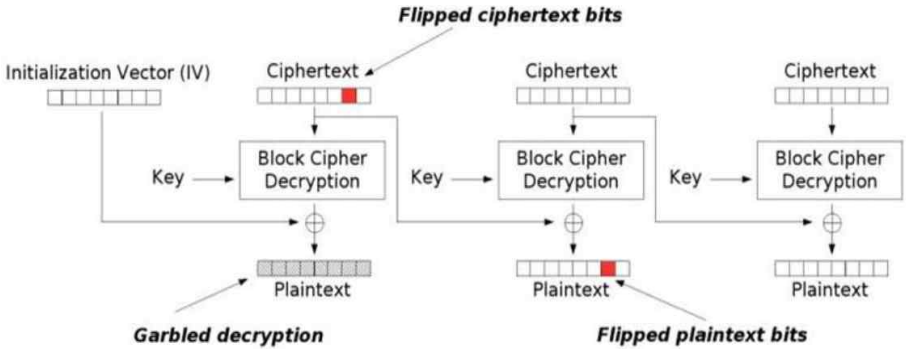


Fig. 1. An example of bit-flipping attack[7]

Algorithm 1 Hindsight Experience Replay (HER)

Given:

- an off-policy RL algorithm $\mathbb{A}$, ▷ e.g. DQN, DDPG, NAF, SDQN
- a strategy $\mathbb{S}$ for sampling goals for replay, ▷ e.g. $\mathbb{S}(s_0, \ldots, s_T) = m(s_T)$
- a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$. ▷ e.g. $r(s, a, g) = -[f_g(s) = 0]$

Initialize $\mathbb{A}$ ▷ e.g. initialize neural networks
Initialize replay buffer R
for episode = 1, M **do**
  Sample a goal g and an initial state s_0.
  for $t = 0, T-1$ **do**
    Sample an action a_t using the behavioral policy from $\mathbb{A}$:
      $a_t \leftarrow \pi_b(s_t||g)$ ▷ || denotes concatenation
    Execute the action a_t and observe a new state s_{t+1}
  end for
  for $t = 0, T-1$ **do**
    $r_t := r(s_t, a_t, g)$
    Store the transition $(s_t||g, a_t, r_t, s_{t+1}||g)$ in R ▷ standard experience replay
    Sample a set of additional goals for replay $G := \mathbb{S}$(**current episode**)
    for $g' \in G$ **do**
      $r' := r(s_t, a_t, g')$
      Store the transition $(s_t||g', a_t, r', s_{t+1}||g')$ in R ▷ HER
    end for
  end for
  for $t = 1, N$ **do**
    Sample a minibatch B from the replay buffer R
    Perform one step of optimization using $\mathbb{A}$ and minibatch B
  end for
end for

Fig. 2. Hindsight experience replay(HER)[5]

The behind idea is to replace the original goal with the visited state of a failed episode. This modified goals hint the agent how to achieve the newly modified goal.

In RL[8], an agent interacts with an environment by Table. 1.

Table 1. Elements of RL

Symbol	Meaning	Usage
S	States	s
A	Actions	a
T	Functions	T(s, a, s'), s →s' by a
R	Rewards	R(s, a, s')
ɣ	Discount rate	[0,1] discounted long-term

In the sample of (s, a, r, s'), Q-function is defined as follows:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \quad (1)$$

When Q uses a parameterized function approximation, such as a deep neural network, that is a DQN. It can be parameterized by weights Θ, which is defined as follows:

$$\theta = \theta + \alpha(r + \gamma \max_{a'} Q(s',a';\theta) - Q(s,a;\theta))\nabla_\theta Q(s,a,\theta) \quad (2)$$

The DQN[1] has experience replay memory and the target network for stabilizing learning and improving performance. Experience replay memory has the tuple (s, a, r, s'), samples randomly and performs the update based on the randomly sampled tuples. The target network has a separate weight vector θ^-. So the update can be as follows:

$$\theta = \theta + \alpha(r + \gamma \max_{a'} Q_T(s', a'; \theta^-) - Q(s, a; \theta)) \nabla_\theta Q(s, a, \theta) \quad (3)$$

where the weight vector θ^- can be synchronized with θ after some period of time based on a parameters. In this paper, it is the number of goals.

III. The Proposed Algorithm

3.1 The algorithm Description

We introduce a Goal-based Target Network in DQN with HER. Some previous works[9] showed that Q-learning in deep reinforcement learning can suffer from an over-estimation, which might cause a highly biased training because of $E[\max(Q^\pi)] \geq \max E[Q^\pi]$, where $E[\max(Q^\pi)]$ is the expected maximum Q^π, $\max E[Q^\pi]$ is the maximum operation of the expected Q^π, and π is a policy. Q-learning might over-estimate the target Q-value because of the estimator Q^π. In practice, this differences between the real Q-value and the target Q-value are quite large. So, many works uses a separate target network for stabilizing the estimated Q-value. However, it can cause the hindrance of on-line earning. Moreover, the use of replay buffers for reducing the correlation between consecutive samples makes it worse in terms of on-line learning[10].

When the environment has a HER, which can store every transition $s_t \rightarrow s_{t+1}$ not only with the original goal but also with a subset of other goals, we can experience some episode s_0, s_1, ..., s_T in the normal replay buffer and replay each trajectory with an arbitrary goal in a DQN. The standard target network maintains its own separate target function Q_T and then copy Q_T every C step. So, our purpose is making a strategy of copying the target network, Q_T should be changed.

Therefore we attempt to change the strategy of copying Q_T based on the use of HER with a subset of other goals. The update frequency of the target network is crucial factor in our technique. In a DQN, although the estimator Q^π is updated every iteration, the target network Q_T is updated every C step. The update frequency C=1 means the target network Q_T is copied from the estimator Q^π immediately. While C > 1, the target network Q_T is updated with a delay. In the previous research[10], they use the hyper-parameter ω, which is tuned based on each application domain.

They empirically found the hyper-parameter ω by using a grid search method. So, the hyper-parameter ω varies with the domains for the reasonable performance range. However in our research, we attempt to find out the update frequency based on HER with a subset of other goals because the number of goals is selected based on the HER and there is no needs for another method, such as the grid search method. In Fig. 3, there is a formal description of the proposed algorithm, which is from Line 20 to Line 30 are related to update the target network based on HER with the number of goals in DQN. This part is different from the original HER in Fig. 2.

We can update the target network every episode and every goal. However, the original HER can update the target network only every episode. Since HER updates the target network's weight vector θ^- with the original Q-Network's weight θ for every episode, the $\{y_j\text{-}Q(\Phi_j, a_j, \theta)\}$ value of Line 28 in "perform gradient descent" is updated only when the episode changes. However, according to the algorithm proposed in this paper, the $y_j\text{-}Q(\Phi_j, a_j, \theta)\}$ value of Line 28 is updated every episode and every goal. So, the proposed algorithm can perform more detailed gradient descent.

Goal-based Target Network in DQN with HER

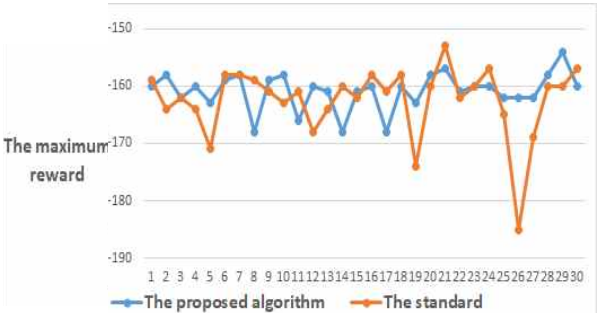
1. Initialize replay memory buffer R
2. Initialize Q with random weights θ and Q_T with $\theta^- = \theta$
   # Q_T is Tartget Q-Network
3. For episode =1 , M do
4. Initialize state s_0, sample a goal g and $\phi_0=\phi(s_0, g)$
5. for t = 0, T-1 do
6. Select an action a_t based on $\max_a Q^\pi(\phi_t, a; \theta)$
7. Or a random with probability ε
8. $a_t \leftarrow \pi(s_t||g)$ # || is concatenation
9. Execute a_t, observe reward r_t , a new state s_{t+1} and $\phi_{t+1}=\phi(s_{t+1}, g)$
10. end for
11. for t = 0, T-1 do
12. $r_t = r(s_t, a_t, g)$
13. Store the transition$(s_t||g, a_t, r_t, s_{t+1}||g)$ in R for a standard replay
14. Sample a set of additional goals for replay G = S
    # S is current episode
15. for $g' \in$ G do
16. $r' = r(s_t, a_t, g')$
17. Store the transition$(s_t||g', a_t, r', s_{t+1}||g')$ in R for HER
18. end for
19. end for
20. for t = 0, N-1 do
21. for g =0 , K-1 do # K is the number of goals
22. Sample a minibatch $(s_t||g', a_t, r', s_{t+1}||g')$ from R
23. if episode terminates at step j then
24. set $y_j = r_j$
25. else
26. set $y_j = r_j + \gamma \max Q_T(\phi_{j+1}, a'; \theta^-)$
    # Goal-based Target Network
27. endif
28. Perform a gradient descent on $\{y_j - Q(\phi_j, a_j; \theta)\}^2$
29. Copy Q_T=Q
    # Update the Target Network based on the number of the goals
30. end for
31. end for
32. end for

Fig. 3. Proposed algorithm

3.2 The comparison evaluation

The proposed algorithm and the standard one to learn Q^π are implemented using TensorFlow[11]. The proposed model and the standard one take actions for an arbitrary integer n where executing the i^{th} action flips like[7]. Moreover, a deep learning network and some hyper-parameters are the same like[12] for both the proposed model and the standard one because our comparisons are not related to the deep learning model and the hyper-parameters but related to the number of goals. Fig. 4 shows comparison between the proposed Goal-based Target Network in DQN with HER and the standard DQN with HER with the fixed update frequency of the target network in the bit-flipping environment.

(a) How many the improved rewards(the quantity)



(b) What is the maximum reward(the quality)

Fig. 4. Comparison results

The maximum number of iterations is 500, the number of episodes is 16, and the size of the bits is 15 as follows the research[7]. The research[5] suggests that in all cases, when the parameter K is 4 or 8, it performs well. Based on the research[5][13][14], we used the k=4. Fig. 4(a) shows the quantity of the comparison, “how many improved rewards are reached in 30 iterations.” Fig 4(b) shows the quality of the comparison, “what is the maximum reward in 30 iterations.”

In terms of the quality comparison of Fig. 4(b), both the proposed algorithm and the standard one with the fixed update frequency are similar. Both could not reach the topmost reward. However, it shows the fluctuated patterns of the rewards of the proposed algorithm is a little bit smaller than the standard one. In terms of the quantity comparison of Fig. 4(a), the proposed algorithm is much reached to the better reward. So, we know whenever the quality of the proposed algorithm is better, the quantity of this is also better.

Table 2 describes the differences between the worst and the best of the proposed algorithm and the

standard one with the fixed update frequency. The difference of the standard algorithm is bigger than the proposed one, especially in terms of the quantity.

Table 3 and Table 4 show the average and the standard deviation of Fig. 4(a) the improved reward and Fig. 4(b) the maximum reward, respectively. Both results show that the proposed algorithm is a little bit better than the standard one in terms of the average. However, the standard deviation of the proposed algorithm is more even than those of the standard one. So, it shows the proposed algorithm is more stable than the standard one.

Table 2. Differences the worst and the best

Proposed / standard	Quatity	Quality
Worst	167 / 31	-168 / -185
Best	337 / 323	-154 / -153

Table 3. Average and standard deviation of Fig. 4(a) the improved reward of the proposed and the standard

Improved reward	Proposed	Standard
Average	282.86	263.9
Standard deviation	41.81	63.57

Table 4. Average and standard deviation of Fig. 4(b) the maximum reward of the proposed and the standard

Maximum reward	Proposed	Standard
Average	-160.87	-162.43
Standard deviation	3.27	6.13

IV. Conclusion

We have suggested a Goal-based Target Network in DQN with HER for increasing the advantage of on-line learning in spite of the use of the target network and replay memories because the use of target network does not work with the fixed update frequency. The proposed algorithm still attempt to ensure the stable learning in sparse and binary reward of high-dimensional domains. We compare the performances of the Goal-based Target Network with the original DQN with HER with the fixed update frequency of the target network in the bit-flipping environment. Our empirical results show that the proposed algorithm can achieve more interact-ability than the original DQN with HER. And we find out that there is a room for considerations in terms of higher rewards. Our future work is to find the quite optimal number of goals based on each application domain such as OpenAI gym.

References

[1] V. Mnih, K.y Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Belle⊏mare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wier⊏stra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning", Nature, Vol. 518, pp. 529-533, Feb. 2015. https://doi.org/10.1038/nature14236.

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, and M. Lanctot, "Mastering the game of go with deep neural networks and tree search", Nature, Vol. 529, No. 7587, pp. 484-489, Jan. 2016. https://doi.org/10.1038/nature16961.

[3] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies", Journal of Machine Learning Research, Vol. 17, No. 1, pp. 1-40, Jan. 2016.

[4] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping", In Proceedings of the 16th ICML, San Francisco, CA, United States, pp. 278-287, Jun. 1999.

[5] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. "Hindsight experience replay", In Advances in Neural Information Processing Systems(NIPS 2017), LONG BEACH CA, pp. 5048-5058, Jul. 2017.

[6] R. S. Sutton and A. G. Barto, "Reinforcement learning", An introduction, MIT press Cambridge, Vol. 1, 1998.

[7] Byte-Flipping-Attack, https://resources.infosecinstitute.com/topic/cbc-byte-flipping-attack-101-approach/ [accessed: Apr. 01, 2021]

[8] R. S. Sutton, "Temporal credit assignment in reinforcement learning", Doctoral Dissertation, Jan. 1984.

[9] H. v. Hasselt. "Double Q-learning", In Advances in Neural Information Processing Systems(NIPS 2010), Vancouver CANADA, pp. 2613–2621, 2010.

[10] S. Kim, K. Asadi, M. Littman, and G. Konidaris, "DeepMellow: Removing the Need for a Target Network in Deep Q-Learning", In IJCAI 2019, pp. 2733-2739. https://doi.org/10.24963/ijcai.2019/379

[11] Tensorflow, https://github.com/tensorflow/tensorflow [accessed: Mar. 31, 2021]

[12] Tidy Reinforcement Learning with Tensorflow, https://github.com/babyapple/tidy-rl [accessed: Mar. 31, 2021]

[13] M. Fang, C. Zhou, B. Shi, B. Gong, J. Xu, and T. Zhang, "DHER: Hindsight experience replay for dynamic goals", In International Conference on Learning Representations(ICLR 2019), New Orleans LA, pp.1-12, May 2019.

[14] Q. He, L. Zhuang, and H. Li, "Soft Hindsight Experience Replay", In IJCAI 2020, Yokohama, arXiv:2002.02089, Feb. 2020.

Author

Chayoung Kim

Feb., 2006 : Ph.D., Computer Science, Korea University

Nov., 2005. ~ Dec., 2008. : Senior Project Researcher, KISTI

Mar., 2009. ~ Feb., 2018. : Adjunct Professor, Dept. of Computer Science, Kyonggi University

Mar. 2018. ~ present : Assistant Professor, Div. of General Studies, Kyonggi University

Reserach Interest : Big-Data, Machine Learning, Deep Learning, Reinforcement Learning