J. Appl. Numer. Optim. 3 (2021), No. 3, pp. 513-519
Available online at http://jano.biemdas.com
https://doi.org/10.23952/jano.3.2021.3.06

IMPROVED SAMPLE EFFICIENCY BY EPISODIC MEMORY HIT RATIO DEEP Q-NETWORKS

RUIYUAN ZHANG, XIANCHAO ZHU, WILLIAM ZHU*

Institute of Fundamental and Frontier Sciences,
University of Electronic Science and Technology of China, Chengdu, China

Abstract. Deep Reinforcement Learning (DRL) has achieved great success in making decisions on some complex tasks. Unfortunately, existing DRL algorithms are usually sample inefficient in that they require a huge amount of interactions with the environment to gain a desirable performance. Recently, Episodic Memory Deep Q-Networks (EMDQN) substantially improves the sample efficiency by episodic memory. However, rewards in episodic memory are delayed because they are obtained after the agent interacts with the environment in a multi-step trial and error manner, which means that EMDQN is sample inefficient to some extent. In this paper, we propose a new algorithm, Episodic Memory Hit Ratio DQN (EMHR-DQN), to improve sample efficiency by reward shaping. Inspired by reward shaping methods, we design a new reward shaping function Episodic Memory Hit Ration (EMHR) to provide additional rewards for the retrieval result of episodic memory. In this way, our method can modify rewards in episodic memory and provide useful supervision for the training of the agent. Experimental results verify the superiority of our method.

Keywords. Episodic memory; Reinforcement learning; Sample efficiency; Reward shaping.

1. INTRODUCTION

Reinforcement Learning (RL) is a major area of interest within the field of machine learning [1]. In RL, the agent learns the behavioural policy to maximize the cumulative reward by interacting with the environment in a trial-and-error manner [2]. In recent years, by adopting Deep Neural Networks (DNN) as function approximators, DRL can learn the behavioural policy faster and better than traditional RL [3]. For example, the pioneering work DQN [4] achieves surpassing human-level performance in playing Atari games. Yet current DRL algorithms still cannot meet the need for the real world, the major reason is sample inefficiency [5].

Recently, Episodic Memory Deep Q-Networks (EMDQN) improves the the sample efficiency by combining episodic memory with DQN for the first time [6]. In EMDQN, DQN is used to approximate the action-value function to improve the generalization, and episodic memory is used to memorize valuable experiences containing high rewards during training and to replay

*Corresponding author.
E-mail addresses: zrynwafu@163.com (R. Zhang), xczhuiffs@163.com (X. Zhu), wfzhu@uestc.edu.cn (W. Zhu).

Received November 10, 2021; Accepted November 27, 2021.

them during evaluation. However, EMDQN obtains rewards by interacting with the environment in a multi-step trial and error manner and stores delayed rewards in the episodic memory, which means that EMDQN is sample inefficiency to some extent.

In this paper, we propose a new algorithm, Episodic Memory Hit Ratio DQN (EMHR-DQN), to improve the sample efficiency by reshaping rewards in episodic memory. Specifically, we design a new reward shaping function Episodic Memory Hit Ratio (EMHR) by calculating the ratio of the number of times that the state-action pair in episodic memory is found to the number of times that episodic memory is retrieved. EMHR regards the historical retrieval information of the episodic memory as the priori and then modifies rewards of episodic memory during the evaluation stage. In this way, our EMHR-DQN algorithm can achieve higher sample efficiency than EMDQN does. Experimental results demonstrate the superiority of our method.

The remainder of our paper is organized as follows. In Section 2, we introduce the background of reinforcement learning and episodic memory. EMDQN is reviewed in Section 3. In Section 4, we propose a new algorithm EMHR-DQN to gain higher sample efficiency than EMDQN does. Then we present the experimental results on Atari games in Section 5. We give the concluding remark in Section 6, the last section.

2. Background

In this section, we introduce the background and related works of reinforcement learning and episodic memory.

2.1. Reinforcement Learning.

As an important subfield of machine learning, RL has achieved great success in decision-making tasks [7, 8]. In RL, the agent learns the policy by interacting with environment. This process can be modeled as a Markov Decision Process (MDP), which is denoted as a tuple $M \triangleq (S, A, T, R, \gamma)$, where S is the set of states, A is the set of actions, $\mathrm{T}(s_{t+1}|s_t, a_t)$ is the transition probability to state $s_{t+1} \in S$ starting from time t and state $s_t \in S$ with action $a_t \in \mathrm{A}$, $\mathrm{R}(s_t, a_t, s_{t+1})$ is the reward function, and $\gamma \in [0,1)$ is the discount factor indicating how much to give in the short term reward. The goal of RL is to find an optimal policy $\pi^* : S \to A$ that maximizes the expected accumulated reward.

Q-learning is a classical method to address the above task [9]. It uses a table to record states and actions and then calculates the Q-function. With the increase in scale, the application of Q-learning will be limited. Afterwards, Deep Q-network successfully approximates the Q-function by DNN to solve complicated and high dimension problems, such as Atria games. The goal of DQN is to minimize the loss

$$(r_t + \gamma \max_{a'} Q_{\bar{\theta}}(s_{t+1}, a') - Q_{\theta}(s_t, a_t))^2,$$

where $\bar{\theta}$ is parameters of the target network, θ represents parameters of primary network, r_t is the reward at time t, and a' is the action selected greedily.

Recently, many works were proposed to improve DQN. For instance, Double-DQN (DDQN) [10] solved the overestimation problem by decorrelating the noises in action selection and value evaluation with two different networks; Dueling DQN [11] decomposed the action-value function Q into state-value function V and the advantage function A to improve robustness. Although these algorithms achieve some success, they still suffer from low sample efficiency because the experiences are absorbed into parameters slowly [12].

2.2. **Episodic Memory.** Some works show that the use of episodic memory can prominently improve the sample efficiency of DRL algorithms [13]. These algorithms use episodic memory to memorize the best episodic experiences (state, action, and reward) during training and to replay the experiences sequences during evaluation, which helps the agent latch on historical actions quickly obtaining high values. Model-Free Episodic Control (MFEC) [14] was proposed to learn policies faster by using non-parametric memory. Another representative method is Neural Episodic Control (NEC) [15], which uses a differentiable neural dictionary and the search based on context to select actions. Although these algorithms can improve the sample efficiency to some extent, they lack generalization.

3. Episodic Memory Deep Q-Networks

To solve the above problem, Episodic Memory Deep Q-Networks(EMDQN), which combines the fast-converging property of episodic memory and good generalization of DQN, gains high sample efficiency [6]. EMDQN leverages two learning targets to simulate striatum and hippocampus respectively [16]. One target is denoted as S, used for providing the inference to simulate striatum; EMDQN chooses the one-step bootstrapped target as the inference target S:

$$S(s_i,a_i) = r_i + \Upsilon \max_{a_{i+1}} Q_\theta(s_{i+1},a_{i+1}).$$

The other is denoted by H, simulating hippocampus by providing the memory target; the best-memorized return is chosen as the memory target H:

$$H(s_i,a_i) = \max_t R_t(s_i,a_i), t \in \{1,2,...,E\},$$

where E represents the number of episodes that the agent has been trained, and $R_t(s_i,a_i)$ represents the future return when the agent takes action a under state s in t-th episode. The loss function combining the two targets is proposed:

$$L = (Q_\theta - S)^2 + \alpha(Q_\theta - H)^2,$$

where Q_θ represents the DNN approximation of the value function with parameter θ, and α is the weight for adjusting the importance of S and H. By integrating the rapid convergence of episodic memory into DNN, EMDQN accelerates the training process and considerably improves the sample efficiency. However, EMDQN gains rewards by multi-step trial and error with the environment, which causes that delayed rewards are stored in experiences. Consequently, EMDQN is sample inefficient to some extent.

4. Episodic Memory Hit Ratio Deep Q-Networks

In this section, we propose a new algorithm, Episodic Memory Hit Ratio DQN (EMHR-DQN), to achieve higher sample efficiency than EMDQN does. First, we review reward shaping. Then, we will introduce a new reward shaping function Episodic Memory Hit Ratio (EMHR) and illustrate the framework of our algorithm.

4.1. **Reward Shaping.** Reward shaping is usually used for solving delayed reward problems, which designs a new reward function $M : S \times A \times S \to R$ that provides useful prior knowledge in the form of additional localized rewards [17]. Ng *et al.* proved that the potential-based reward shaping (PBRS) can reduce the learning time and do not change the optimality of the original

optimal policy [18]. Wiewiora *et al.* extended the PBRS by potential-based advice (PBA) to include actions [19]. The generic form of reward shaping is as follows:

$$\mathrm{R}'(\mathrm{s},\mathrm{a},\mathrm{s}') = \mathrm{R}(\mathrm{s},\mathrm{a},\mathrm{s}') + \mathrm{M}(\mathrm{s},\mathrm{a},\mathrm{s}'),$$

where $R(s,a,s')$ represents the original reward function, $M(s,a,s')$ represents the additional reward function and the result of reward shaping is $R'(s,a,s')$.

4.2. **Episodic Memory Hit Ratio Deep Q-Networks.** In this subsection, we introduce our algorithm EMHR-DQN to obtain high sample efficiency by reshaping rewards in experiences that episodic memory stores. First, we define an indicator function $I(s_i,a_i)$ to represent whether the target experience is found in episodic memory. The formula is expressed as follows:

$$I(s_i,a_i) = \begin{cases} 1, & if\,(s_i,a_i) \in H, \\ 0, & otherwise. \end{cases}$$

If the state-action pair is found in episodic memory, then the value of $I(s_i,a_i)$ is set to 1; otherwise, the value of $I(s_i,a_i)$ is set to 0. Based on the above formula, we then propose a new reward function Episodic Memory Hit Ratio (EMHR), $E(s_i,a_i)$, by calculating the ratio of the number of times that the state-action pair in episodic memory is found to the number of times that episodic memory is retrieved:

$$E(s_i,a_i) = \frac{1}{mc} * \sum_{i=1}^{m}\sum_{j=1}^{c} I(s_i,a_i),$$

where m represents the current iteration number, c is the size of a batch, and $1/mc$ represents the total number of retrieval. EMHR regards the probability of the state-action pair found in episodic memory during evaluation as the priori. And it can modify rewards stored in experiences according to the use of episodic memory during the whole evaluation stage. Furthermore, we propose the new memory target $F(s_i,a_i)$ to provide regularization for $Q_\theta(s_i,a_i)$.

$$F(s_i,a_i) = H(s_i,a_i) - \lambda E(s_i,a_i),$$

where λ is a hyper-parameter. In summary, we obtain a new loss function:

$$\begin{aligned} L_{EMHR} = \min_{\theta} \sum_{(s_i,a_i,r_i,s_{i+1}\in D)} & [(Q_\theta(s_i,a_i) - S(s_i,a_i))^2 \\ & + \beta(Q_\theta(s_i,a_i) - F(s_i,a_i))^2], \end{aligned} \tag{4.1}$$

where D is a mini-batch of experiences, and β represents a hyper-parameter.

Figure 1 shows the architecture of our algorithm EMHR-DQN. In the training phase, the state s, represented by four frames of game images, is processed by convolutional neural network and full connected layer to figure out $\mathrm{Q}_\theta(s,a)$; the vector h, the value of $\Phi(s)$, is the product of state s and a random matrix and is used for retrieving the episodic memory; the episodic memory is made use of storing good experiences (including states, actions and rewards), which is constructed by kd-tree with the advantages of simple storage and quick retrieval. In the evaluation phase, (1) we calculate EMHR based on formula (4.2); (2) we use $(\Phi(s),a)$ to retrieve the memory and obtain the search result $F(s_i,a_i)$ based on formula (4.2); (3) we use stochastic gradient descent (SGD) to optimize the loss function (4.1) and update parameter θ via backpropagation.

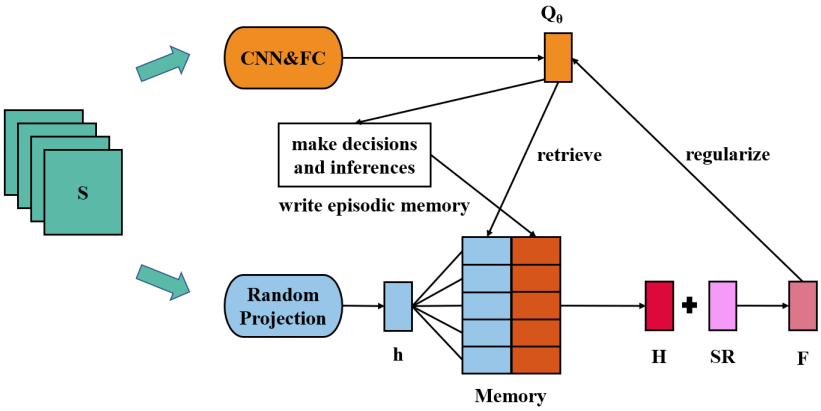


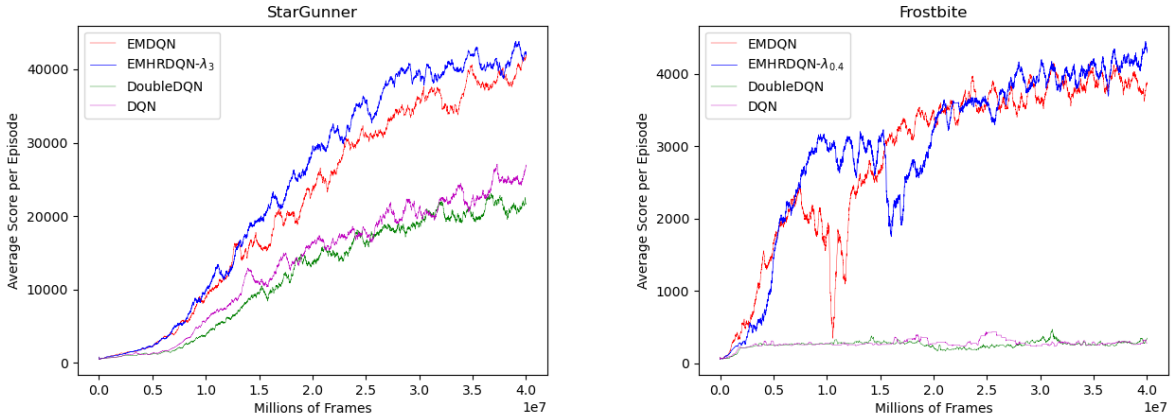
FIGURE 1. The architecture of EMHR-DQN

$$F(s_i,a_i)=\begin{cases}\max\{H(s_i,a_i),R(s_i,a_i)\}-\lambda E(s_i,a_i), & if\ (s_i,a_i)\in H,\\ R(s_i,a_i)-\lambda E(s_i,a_i), & otherwise,\end{cases} \tag{4.2}$$

where $R(s_i,a_i)$ represents the Monte-Carlo return of the episode.

5. EXPERIMENTS

We use Arcade Learning Environment (ALE) [20] to verify the effectiveness of our method. Specifically, we select two classical games (StarGunner and Frostbite) as experimental environments, which contains delayed rewards and different magnitudes of scores across games. Both games are relatively simple visually but require complex and precise policies to achieve the high expected reward [21].



(a) StarGunner (b) Frostbite

FIGURE 2. Training curves of StarGunner and Frostbite on 40M frames.

We follow all of the hyper-parameter and networks settings as EMDQN. Figure 2 shows training curves on two games for 40 epochs (each containing 1M frames). All learning curves

show the average performance over several different initial random seeds. And it is clear that our algorithm obtains higher sample efficiency than other compared algorithms.

6. Conclusion

In this paper, we proposed a new algorithm, Episodic Memory Hit Ratio DQN, to improve the sample efficiency by reshaping rewards in episodic memory. Inspired by reward shaping methods, we designed a new reward shaping function Episodic Memory Hit Ration (EMHR) to provide additional rewards for the episodic memory. In this way, our EMHR-DQN algorithm can achieve higher sample efficiency than EMDQN does. Experimental results demonstrated the superiority of our method. In the future, we plan to improve the exploration ability of our algorithm by combining intrinsic rewards.

Acknowledgments

The authors are grateful to the referees for useful suggestions which improved the contents of this paper. This work was supported in part by The National Nature Science Foundation of China under Grant Nos. 61772120.

References

[1] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, The MIT Press, Cambridge, 1998.
[2] S.D. Whitehead, A complexity analysis of cooperative mechanisms in reinforcement learning, Proceedings of the AAAI Conference on Artificial Intelligence, pp. 607-613, 1991.
[3] P. Henderson, R. Islam, P. Bachman, et al., Deep reinforcement learning that matters, Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[4] V. Mnih, K. Kavukcuoglu, D. Silver, et al., Human-level control through deep reinforcement learning, Nature, 2015, 518 (2015), 529-533.
[5] S.Y. Lee, C. Sungik, S.Y. Chung, Sample-efficient deep reinforcement learning via episodic backward update, Adv. Neural Info. Process. Sys. 32 (2019), 2112-2121.
[6] Z. Lin, T. Zhao, G. Yang, et al, Episodic memory deep Q-networks, Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 2433-2439, 2018.
[7] R. Furuta, N. Inoue, T. Yamasaki, Fully convolutional network with multi-step reinforcement learning for image processing, Proceedings of the AAAI Conference on Artificial Intelligence, 33 (2019), 3598-3605.
[8] Y.D. Jiang, S.S. Gu, K.P. Murphy, et al, Language as an abstraction for hierarchical deep reinforcement learning, Adv. Neural Info. Process. Sys. 32 (2019), 9419-9431.
[9] C.J.C. H. Watkins, P. Dayan, Q-learning, Machine Learning, 8 (1992), 279-292.
[10] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double q-learning, Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016.
[11] Z. Wang, T. Schaul, M. Hessel, et al, Dueling network architectures for deep reinforcement learning, International Conference on Machine Learning, pp. 1995-2003, 2016.
[12] A. Zakharov, M. Crosby, Z. Fountas, Episodic memory for subjective-timescale models, International Conference on Machine Learning 2021 Workshop on Unsupervised Reinforcement Learning, 2021.
[13] M. Lengyel, P. Dayan, Hippocampal contributions to control: the third way, Adv. Neural Info. Process. Sys. 20 (2007), 889-896.
[14] C. Blundell, B. Uria, A. Pritzel, et al, Model-free episodic control, arXiv preprint arXiv:1606.04460, 2016.
[15] A. Pritzel, B. Uria, S. Srinivasan, et al, Neural episodic control, International Conference on Machine Learning, pp. 2827-2836, 2017.
[16] C.M.A. Pennartz, R. Ito, P. Verschure, et al, The hippocampal–striatal axis in learning, prediction and goal-directed behavior, Trends in Neurosciences, 34 (2011), 548-559.

[17] Y. Hu, W. Wang, H. Jia, et al, Learning to utilize shaping rewards: A new approach of reward shaping, 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, 2020.

[18] A.Y. Ng, D. Harada, S. Russell, Policy invariance under reward transformations: Theory and application to reward shaping, International Conference on Machine Learning, 99 (1999), 278-287.

[19] E. Wiewiora, G.W. Cottrell, C. Elkan, Principled methods for advising reinforcement learning agents, International Conference on Machine Learning, pp. 792-799, 2003.

[20] J. Schrittwieser, I. Antonoglou, T. Hubert, et al, Mastering atari, go, chess and shogi by planning with a learned model, Nature, 588 (2020), 604-609.

[21] G. Farquhar, K. Baumli, Z. Marinho, et al, Self-consistent models and values, 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia, 2021.