Co-training by Experience Replay for Reinforcement Learning

Yuyang Huang

University of Nottingham Ningbo China, 199 Taikang East Road Ningbo Zhejiang, 315100, China

yuyang.huang@nottingham.edu.cn

Abstract. In this paper, to improve the efficiency of the reinforcement learning model to explore the environment and get better results, a new method which involves the co-training process in reinforcement learning by sharing the experience pool of each agent in the training process has been developed. In this method, agents can gain a better understanding of the environment since agents use different policies to make action and explore the environment. At the same time, this paper designed an agent called Hard Memory Collector by modifying the value function and combining this agent and a normal agent for co-training. As an experimental result on the ViZDoom platform, the model achieved better results than the original Duel DQN network in terms of score, steps used per game and loss value.

Keywords: Reinforcement Learning; Co-training; Experience Replay; ViZDoom; Duel DQN.

1. Introduction

Reinforcement learning (RL) combined with neural networks has recently led to a wide range of successes in learning policies for sequential decision-making problems. And some can reach the human level or even better than human performance, such as playing ViZDoom [1], which is a fully customizable, sufficient function, suitable for reinforcement learning 3D platform for research for vision-based reinforcement learning. A typical example is an agent which achieved human performance through reinforcement learning [2]. Another example is the robotic manipulation problem, where reinforcement learning is used to control robotic arms for diverse tasks such as object recognition and physics-based control [3].

However, a common challenge is that the agent needs to fully explore the environment to get enough information to update the policy function, for example, training the model simultaneously to learn game features [4]. Conventional wisdom in problem-solving suggests that the goal of a task can be accomplished in several ways, and this also can be applied to context-dependent reinforcement learning.

A more typical way of learning is to learn from mistakes, because wrong behaviours often contain experiences that can lead to success. This is fully used in the initial training stage of reinforcement learning, but when the agent masters After a certain strategy, the reward is largely deliberately maximized in a single way, rather than actively exploring the environment. At the same time, mistakes are not made deliberately in most cases, because mistakes are often accompanied by loss of profits in real environments. But in reinforcement learning, especially based on virtual environments, mistakes can be made intentionally at no cost to gain experience.

One major question is how different combinations of states and actions affect model learning. This is related to the co-training problem [5], where different feature representations of the same problem enable more effective learning than using only a single representation. Previous methods have usually applied multiple agents with different policies to exploit the environment randomly [6].

This paper proposes a method that combines experience pooling [7–9] and co-training, a model in which multiple agents explore the environment with different policies and experience pools. The experiments of this paper used two agents, one agent (Agent) targets the highest reward, and another agent (Hard Experience Collector) targets the worst behaviour, both learn through a shared experience pool and finally make the Agent get the best result. The Agent partially learns from Hard Experience Collector's misbehaviour to learn from and avoid mistakes. Experiments were carried out based on Duel DQN and obtained higher scores and faster game completion times.

2. Methods

The key in the method is to use different agents with different strategies to explore the environment while recording the process into the experience pool and let each agent share these experiences during the training process to achieve the purpose of effectively exploring the environment and avoiding local optima. The method is based on Duel DQN [10] but involves only an experience pool and policy function to share the experiences of multiple agents, while some agents called Hard Experience Collector (HEC) use a policy function to make the worst action for other agents to learn. Therefore, the method can theoretically also be applied to other off-policy types of reinforcement learning frameworks.

2.1 Co-training

The idea of Co-training is that each agent's action strategy in different states is inconsistent. This inconsistent strategy allows the agent to explore the environment in different ways, which brings a very rich experience pool data. By sharing these pools of memories, specific agents can learn and make decisions in the most efficient efficiency solely on their decision-making pools of limited experience to explore the environment.

The method is shown in Figure 1. During the exploration process, each agent has different strategies and corresponding experience pools. These strategies are manually set with the goal of oblongly obtain much data as possible. After getting the next state and reward, store these experiences into its experience pool.

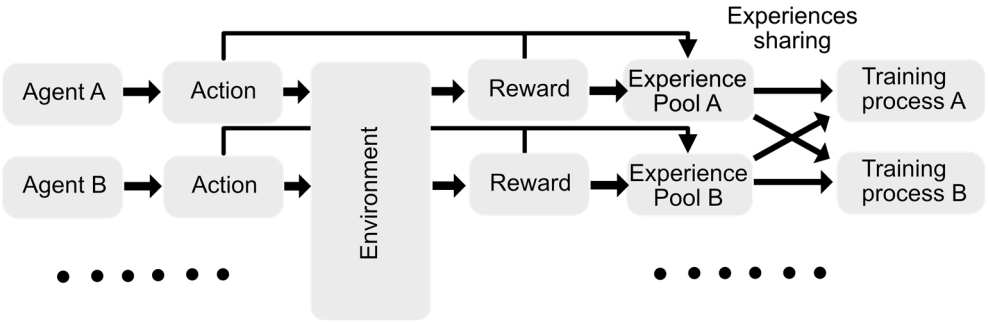

Figure 1. Co-training process

In the training phase, to effectively use the experience pool generated by each agent, it is necessary to have a corresponding strategy to extract samples from all the experience pools. This strategy is a completely random extraction process in the conventional methods, but in this case, most of the experience may be worthless. Therefore, a method for extracting experience is designed in this paper. This method extracts two types of experience based on random extraction, that is, the experience with the maximum reward and the experience with the minimum reward. The method can be summarized as 1) random experience in each experience pool 2) Maximum reward experience 3) Minimum reward experience. To avoid excessive reuse of the maximum and minimum reward experience, the method will randomly remove half of the corresponding experience from the experience pool after each extraction of the maximum and minimum reward experience.

2.2 Hard Experience Collector

The main goal of HEC is deliberately making the worst action, and to provide these experiences to the agent which can take the right action to learn. The target of the original Duel DQN is calculated by the following equation [10]:

$$y_i^{DDQN} = r + \gamma Q\left(s', \arg\max_{a'} Q(s', a'; \theta_i); \theta^-\right) \quad (1)$$

To make the worst action, this paper modifies it to the following formula

$$y_i^{HEC} = r + \gamma Q\left(s', \arg\min_{a'} Q(s', a'; \theta_i); \theta^-\right) \quad (2)$$

This allows HEC to act against the optimal behaviour, thus obtaining a pool of experience of the agent that is completely different from the final training target.

3. Experiments

This paper uses the ViZDoom platform as an environment to conduct experiments and get results. The experimental environment is the official basic scenario. In this scenario, the agent can make three actions: move left, move right and attack. The action moves the monsters in the game relate to the centre of the field of vision and then attack to complete the game. The game will restart after each game is completed.

3.1 Hyperparameters

All data use the MSELoss algorithm and a minibatch size of 512. The network weights will be updated at each step after the experience pool size of all agents is larger than the minibatch, and the target network of Duel DQN will be updated every 2000 steps. Such an update will be performed a total of 60 times, that is, the training contains a total of 120,000 steps, The discount factor was set to 0.99 and the replay memory contained the 10,000 most recent frames. An ε-greedy policy [11] has been used during the training phase, where ε was linearly decreased from 1 to 0.1 over the first million steps, and then fixed to 0.1. And use grayscale images and a 4/3 resolution of 640x480 which is resized to 30x45.

3.2 Experience Pool and Experience Sharing Strategy

The experience of each experience pool comes from the records of the current state, next state, reward, and action generated by the corresponding agent in the exploration environment. These experiences are used in the training of the agent by sampling these experiences. This article sets different experience pool sampling strategies for two different agents, as shown in the figure, where the agent is the training object as the final result, the goal is to get the optimal result, and the agent samples the random part from its experience pool and The maximum part, sampling the worst part from the HEC experience pool so that the agent can learn from the worst results; HEC samples the random part from its experience pool, and samples the maximum part from both experience pools at the same time, which encourages HEC Take the worst behavior that is contrary to the best experience.

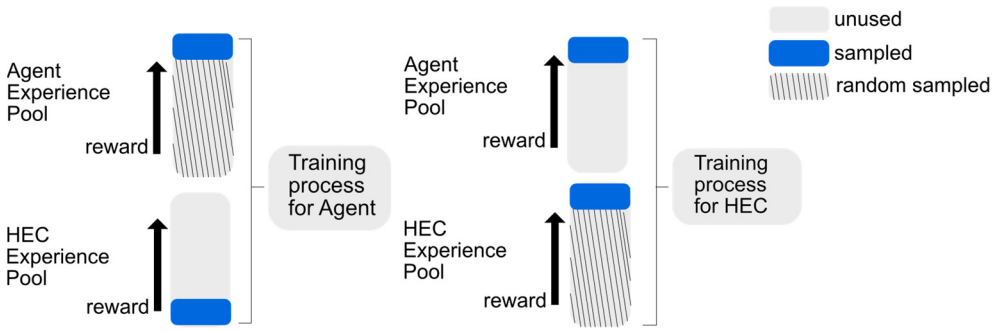

Figure 2. Experience Sharing Strategy

3.3 Results

This paper used the step number and scores obtained in each game and the loss of the network as the evaluation criteria. As shown in Figure 3, it is shown that the method is more efficient than a single agent in terms of convergence speed and results.

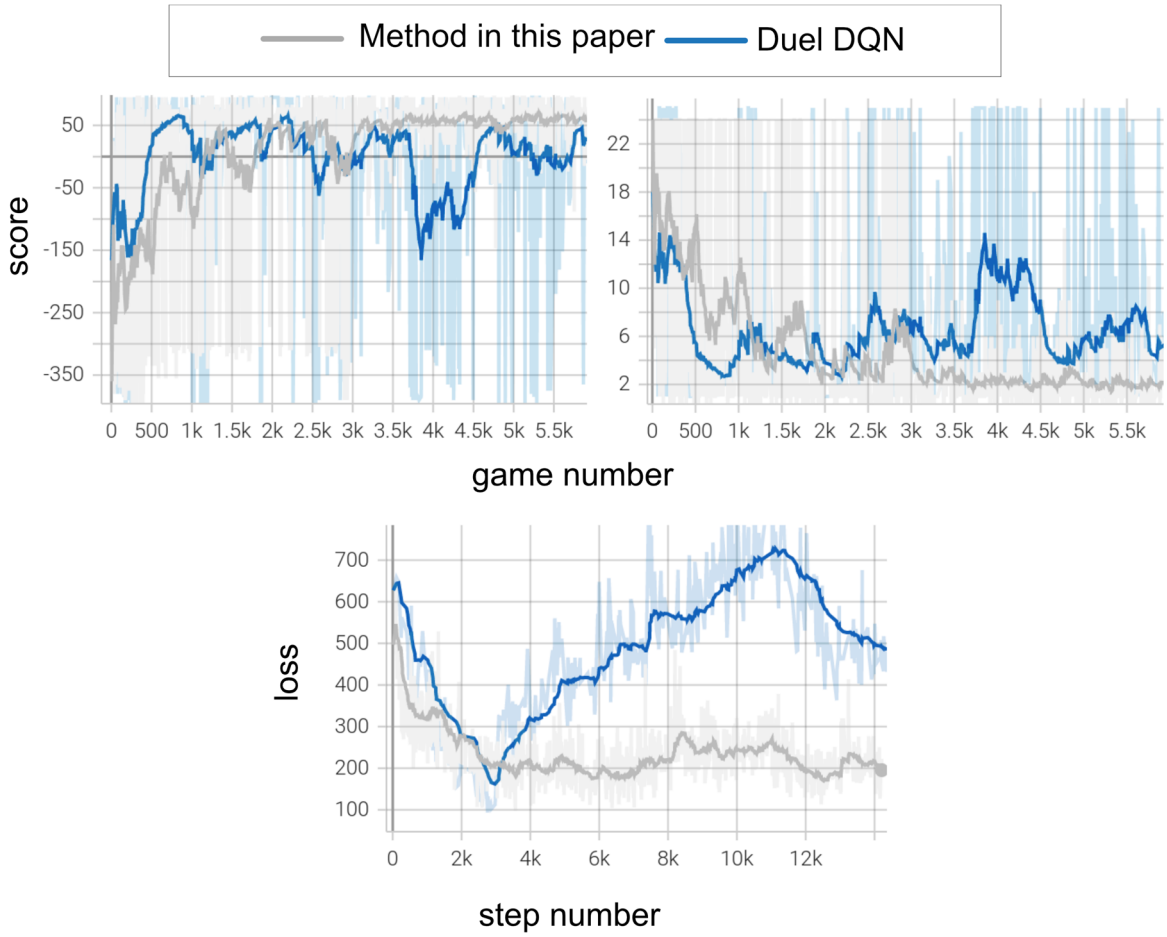

Figure 3. Experiment results

4. Conclusion

A new method that combines co-training and reinforcement learning has been introduced in this paper, it allows multiple agents to explore the environment by different policies and share their replay memory. Compared with conventional Duel DQN, experimental results on the ViZDoom platform show that this method can successfully achieve better performance.

References

[1] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaskowski, "ViZDoom: A Doom-based AI research platform for visual reinforcement learning," in 2016 IEEE Conference on Computational Intelligence and Games (CIG), Santorini, Greece, Sep. 2016, pp. 1–8. doi: 10.1109/CIG.2016.7860433.

[2] M. Wydmuch, M. Kempka, and W. Jaśkowski, "ViZDoom Competitions: Playing Doom from Pixels," IEEE Trans. Games, vol. 11, no. 3, pp. 248–259, Sep. 2019, doi: 10.1109/TG.2018.2877047.

[3] H. Guan, "Analysis on Deep Reinforcement Learning in Industrial Robotic Arm," in 2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), Sanya, China, Dec. 2020, pp. 426–430. doi: 10.1109/ICHCI51889.2020.00094.

[4] G. Lample and D. S. Chaplot, "Playing FPS Games with Deep Reinforcement Learning," AAAI, vol. 31, no. 1, Feb. 2017, doi: 10.1609/aaai. v31i1.10827.

[5] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training y," p. 10.

[6] J. Song, R. Lanka, Y. Yue, and M. Ono, "Co-training for Policy Learning," p. 11.

[7] M. Andrychowicz et al., "Hindsight Experience Replay." arXiv, Feb. 23, 2018. Accessed: Aug. 07, 2022. [Online]. Available: http://arxiv.org/abs/1707.01495.

[8] J. Foerster et al., "Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning," p. 10.

[9] R. Liu and J. Zou, "The Effects of Memory Replay in Reinforcement Learning." arXiv, Oct. 17, 2017. Accessed: Aug. 28, 2022. [Online]. Available: http://arxiv.org/abs/1710.06574.

[10] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," p. 9.

[11] V. Mnih et al., "Playing Atari with Deep Reinforcement Learning." arXiv, Dec. 19, 2013. Accessed: Aug. 28, 2022. [Online]. Available: http://arxiv.org/abs/1312.5602.