Learning to Play Visual Doom using Model-Free Episodic Control

Byeong-Jun Min
Department of Computer Science and Engineering
Sejong University, Seoul, South Korea
okminkr@gmail.com

Kyung-Joong Kim*
Department of Computer Science and Engineering
Sejong University, Seoul, South Korea
kimkj@sejong.ac.kr

***Abstract*— Recently, the deep reinforcement learning has shown successful outcomes in classic video games (e.g., ATARI) and visual doom competition. Although it's very powerful, it suffers from very long learning time to generalize its performance. For example, it takes about 7~15 days to produce a good controller for ATARI games with state-of-the art GPUs. In this work, we propose to speed up the visual-based learning by introducing episodic control into the Visual Doom platform. The episodic control memorizes agent's experience with random projection and selects the next action based on similarity search on the memory. Because it's a model-free learning, it does not require much time to generalize a model and speeds up learning by exploiting previous experience. This is the first time to apply the episodic control into the visual Doom platform. Experimental results show that it converges to the desirable performance faster than the deep Q network in basic environment.**

Keywords— Episodic Control; Reinforcement Learning; Visual Doom; Random Projection;

I. INTRODUCTION

Recently, deep reinforcement learning (DRL) has shown an equal or superior performance to human players in Go, Poker, and classic video games. For example, Deep Q Network (DQN) [1] and Asynchronous Advantage Actor-Critic (A3C) [2] produced human comparable controls for classical Atari games just based on screen inputs without any domain knowledge. Because it learns complex value functions from trail and errors, it takes much time to train it even for simple video games. For example, it requires experience of about 5,000 million frames and physically 7~14 days to finish.

Google DeepMind introduced a model-free episodic control (EC) for Atari games [3]. It works like a K-Nearest Neighbor algorithm by comparing all the scenes stored in the memory with the current game screen. They reported that the approach shows potential to speed up the visual-based learning for the Atari games and Labyrinth environment. It demonstrated that the algorithm is able to play the games quickly than the deep reinforcement learning even there is no exactly same situation during the play.

In this work, we propose to apply the episodic control to the Visual Doom (VizDoom) platform. [4] It's important to know whether the episodic control is useful to other types of games beyond the classical games or navigation/item collection games. This is the first time to test the episodic control in the 3D first person shooting game. Our work shows that the EC can produce successful control more quickly than the conventional DQN.

Fig. 1. VizDoom platform. (basic environment)

II. BACKGROUND

A. Model Free Episodic Control

Humans can very quickly gain high reward in unseen environment. In the brain, such rapid learning is thought to depend on the hippocampus and its capacity for episodic memory. [3] Model-free episodic control is an algorithm that models these things. The episodic controller has a table called Q^{EC} and memorizes AI player's episodic memory and rewards at that time. The episodic controller will make AI player decisions through the Q^{EC} table and play the game, and then update the Q^{EC} table after the game ends. This is similar to that a person relying on their existing experience to solve the problem. And if one's own experience is wrong, it's like revaluation it. Therefore the episodic controller can learn to solve difficult sequential decision-making tasks.

The episodic control is a tabular-RL algorithm. It learns by growing Q^{EC} table. The Q^{EC} table has a buffer for each action and uses state as an index, and its capacity is limited. The state is used as the vector value that the image (observation) is projected to low dimension by embedding function (ϕ). The goal of the episodic control is to learn a policy that maximizes

the expected discounted return. This algorithm is divided into two phases.

$$\widehat{Q^{EC}}(s,a) = \begin{cases} \frac{1}{k}\sum_{i=1}^{k} Q^{EC}\left(S^{(i)},a\right) & if\ (s,a) \notin Q^{EC}, \\ Q^{EC}\ (s,a) & otherwise, \end{cases} \quad (1)$$

One is when the AI player is still playing the game. The episodic controller evaluates the current state through (1). It evaluates the novel state by generalizing it through K similar states. The similar states are searched through the K-nearest neighbor algorithm (KNN). If it is not a novel state, it depends on one's own previous experience.

$$Q^{EC}(s_t,a_t) \leftarrow \begin{cases} R_t & if\ (s_t,a_t) \notin Q^{EC}, \\ \max\{Q^{EC}(s_t,a_t), R_t\} & otherwise, \end{cases} \quad (2)$$

The other is when the game is over. The episodes are organized in reverse order to update the Q^{EC} table. And update via (2). If the buffer is full, the oldest memory is removed and updated. Such forgetting of older, less frequently accessed memories also occurs in the brain.

B. Random Projection

Random Projection (RP) [5] is one of the dimension reduction techniques. Higher vectors can be projected to the appropriate lower dimension, and the distance between the projected vectors can be maintained. The original d-dimensional data is projected to a k-dimensional ($k << d$) subspace, using a random k x d - dimensional matrix R. The random matrix R can be generated using a Gaussian distribution.

III. Proposed System

The ViZDoom basic environment allows three actions: Left, Right, Shot. We have restricted the AI player's parallel behavior (Left + Shot, Right + Shot) to reduce GPU memory requirements. Because each action has a buffer, the memory requirement is proportional to the number of actions. This means that there is still a possibility of improving AI player performance.

The episodic control uses KNN algorithms to find similar situations. It slows down as the Q^{EC} buffer size grows. To solve this problem, we use the GPU. The larger the buffer size, the better GPU processing than CPU processing. It guarantees the real-time play of AI player. It is advantageous to use the GPU for further experiments. For Example, In VizDoom Health-Gathering, which is provided in an open Gym, the maximum length of the game is six times that of the VizDoom basic. However, it is not recommended if the GPU buffer size is small.

There are other dimension reduction algorithms, but we used 64 Random Projection as an embedding function. Random Projection is very effective in reducing size because we do not want to find special properties in the image. Simply, episodic controller compares the similarities of the images themselves.

The episodic control varies greatly in performance depending on the k value. Therefore, it is very important to choose a good k value. When considering the k value, we also considered the case that game screen is same but UI of game is different.

Fig. 2. Episodic controller Configurations

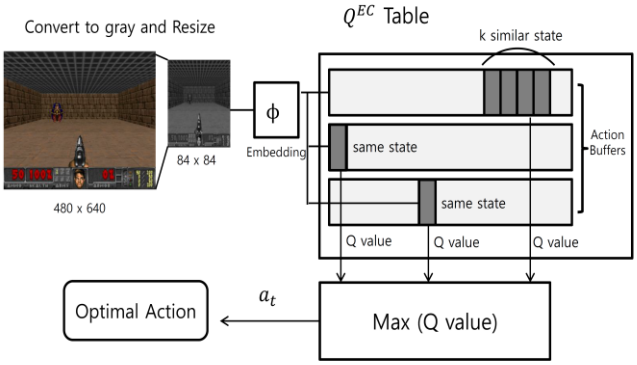

Fig. 2 shows the configuration of the episodic controller. The image (observation) is projected into a low dimensional vector through a series of transformations. The projected image is defined state (s_t). The Q^{EC} table determines the optimal action (a_t) through each action buffer.

IV. Experiments

The rule of VizDoom basic is to hit a random target on the opposite wall in a small room. If the AI player hit the target, AI player will get 100 points and if the AI player misses the target will get -5 points. If nothing happens, it gets -1 point every 0.028 seconds. If 350 frame passes, it automatically goes to the next episode

We set parameters: the capacity of each action buffer was limited to 250,000, the image resized to 84x84 and converted to gray-scale, the discount factor $\gamma = 0.99$, $k = 4$, frame skip = 8, and the initial $\epsilon = 1.0$ started to decay during 50000 frame, finishing the decay at $\epsilon = 0.05$ at 2600 episodes.

TABLE I. Development Environment

CPU	Intel® Core™ i7-3770
RAM	32.0GB
GPU	Geforce GTX 960 2GB
Language	Python 2.7
Libraries	Tensorflow, Numpy

Experiments were performed in the VizDoom basic Environment within the Open AI Gym platform. The AI player has trained a total of 500,000 frames. Experimental results are compared with DQN and another episodic controller. Table. 1 shows the development environment.

Fig. 3. Comparison Result with DQN (video available from **)

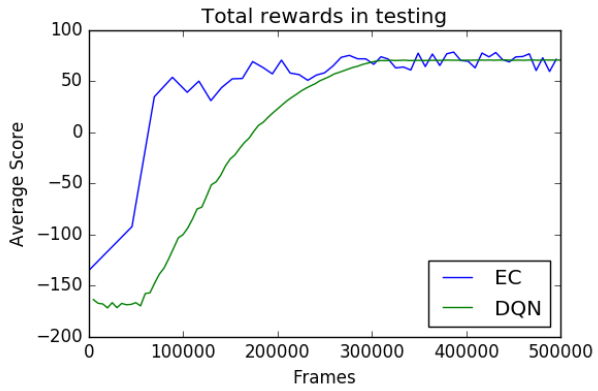

Fig. 3 shows experimental results. It shows the total rewards change in testing (average score of 50 runs). The AI player quickly earned a high score as soon as ϵ-greedy reached its minimum value. The time it took to achieve 50 points was 200,000 frames faster than DQN. At the end of the learning, the left and right action buffer was much smaller than the shot action buffer.

Fig. 4. Comparison Result of k parameter in Episodic control

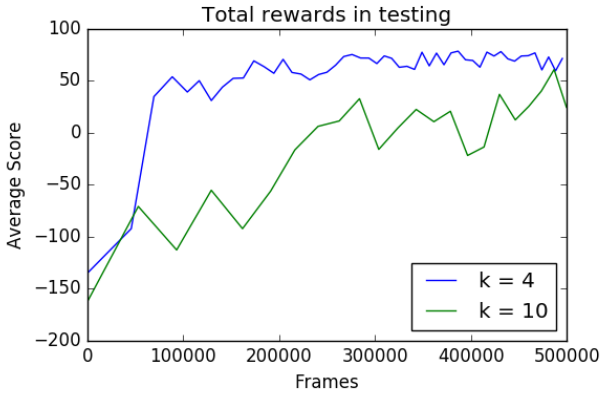

Fig. 4 shows the results of two episodic controllers using different k parameters. The episodic controller with k = 10 reached 50 points after learning 500,000 frame. It is late 300,000 frames compared to k = 4. The size of the Q^{EC} table was 200,000 larger than using a k = 4, but performance was poor. It shows how the k value affects the performance of the episodic controller.

V. Conclusions and Future Works

In this paper, we propose to speed up the visual-based learning by introducing episodic control into the visual Doom platform. We have confirmed that the episodic controller achieves high scores quickly in the VizDoom basic environment as expected.

In this experiment, episodic controller was not overtaken by DQN, but in an environment such as Health-Gathering it will be overtaken, because the size of the Q^{EC} table cannot be larger than the depth of the game. Therefore it is necessary to research the combination of episodic control and DQN in the future. In addition, we will research how to effectively reduce the data in the Q^{EC} table. Because not using parallel behavior is not a fundamental solution.

Acknowledgement

This research was supported by basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(2017R1A2B4002164). *: corresponding author

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing Atari with Deep Reinforcement Learning,” arXiv:1312.5602 [cs], Dec 2013

[2] V. MNIH, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. Lillicrap, D. Silver, K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," In: International Conference on Machine Learning, p. 1928-1937, June 2016.

[3] C. Bluedell, B. Uria, A. Pritzel, Y. Li, A. Ruderman, J.Z. Leibo, J. Rae, D, Wierstra, D. Hassabis “Model-Free Episodic Control,” arXiv:1606.04460, Jun 2016

[4] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski, “ViZDoom: A Doom-based AI Research Platform for Visual Reinforcement Learning,” arXiv:1605.02097 [cs], May 2016.

[5] E. Bingham, H. Mannila. "Random projection in dimensionality reduction: applications to image and text data," Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 245-250, August 2001.

** https://youtu.be/j-e5ELaXbnc