Deep Q-Learning using Redundant Outputs in Visual Doom

Hyunsoo Park, and Kyung-Joong Kim*
Department of Computer Science and Engineering
Sejong University
Seoul, South Korea
rex8312@gmail.com, kimkj@sejong.ac.kr

***Abstract*— Recently, there is a growing interest in applying deep learning in game AI domain. Among them, deep reinforcement learning is the most famous in game AI communities. In this paper, we propose to use redundant outputs in order to adapt training progress in deep reinforcement learning. We compare our method with general ε-greedy in ViZDoom platform. Since AI player should select an action only based on visual input in the platform, it is suitable for deep reinforcement learning research. Experimental results show that our proposed method archives competitive performance to ε-greedy without parameter tuning.**

Keywords—deep reinforcement learning; reinforcement learning; vizdoom; first-person perspective game;

I. INTRODUCTION

In recent years, the deep learning has become famous in various domains. Especially, it shows better performance than conventional methods in handling high dimensional data such as visual inputs. Deep reinforcement learning (Deep Q-Learning; DQL) is the one of representative works in the game AI domain. Basically, it is a combination of deep learning with Q-learning and can learn an AI game player handles raw pixel or text inputs.

Fig. 1. ViZDoom platform (basic example)

Visual Doom (ViZDoom) is the one of AI competition platforms opened recently [1], also OpenAI Gym [2] includes it. It is based on the Doom, the famous classic first person shooter game (Fig. 1). AI players on this platform can only obtain visual input and some variables (eg. Health and armor). The platform does not provide more detailed structured data like map data for navigation or forward models for simulations. It opens new challenge for traditional game AI methods.

The DQL is one of promising solutions to make an AI for ViZDoom. The DQL is one of representative deep learning works in game AI domain. Mnih *et al.* introduce DQL in 2013 [3]. They show DQN can learn how to play various Atari 2600 games. After success of DQL, there have been a lot of works about deep learning in game AI. However, many DQL studies focused on 2-D games unlike ViZDoom. Since ViZDoom provides only first-person view, the player cannot see whole environment (obtains limited information). Furthermore, view angle change makes different visual inputs for the same object.

Exploration and exploitation dilemma is one of important problems in reinforcement learning. It's dilemma between trying new situation in order to get information about environment (exploration) or pursues rewards based on the current knowledge (exploitation). If the player tries the exploration too much or pursues rewards too much, it is possible to reduce total rewards. Because of this reason, the mechanism of how to deal with this dilemma is important.

In many DQL studies, they use ε-greedy algorithm to handle this dilemma. Simply, the ε-greedy selects a random action with ε probability (ε belongs to [0, 1]). Otherwise, it selects the action with the highest rewards based on the current knowledge. Generally, ε value is initially set as high value and gradually reduced at each learning iteration. The ε-greedy is easy to use and shows good performance. But it is not a kind of adaptive learning process and there are some parameters should be determined properly.

In this paper, we propose an algorithm to balance exploration and exploitation. Generally, the number of output nodes in neural network of DQL is the same to the actions that a player can perform. The output of each node is interpreted as an expected Q-value of each action. In our proposed method, we add multiple pair of nodes into the output layer. For instance, if there are two possible actions, then total number of output nodes is 2×10=20. We use these redundant outputs to measure uncertainty of each action's Q-value in the current state. Using this information, proposed method could estimate the progress of learning and use it to balance exploration and exploitation. Osband *et al*. also use redundant output to boost training efficiency [4]. However, our proposed methods are simpler than the previous work.

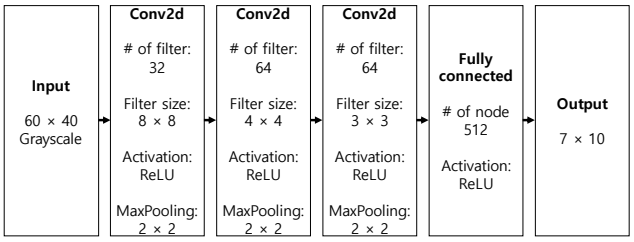
This work was supported by the National Research Foundation of Korea (NRF) grant (2013 R1A2A2A01016589), Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2016.
*: corresponding author

In order to compare performance of our proposed method and ε-greedy, we use ViZDoom's basic example environments. In the experimental result, our proposed method gets similar total rewards eventually without parameter tuning.

II. Proposed Method

Fig. 2. shows an architecture of neural networks in our experiments. There are six layers including input and output layers. In the input layer, we provide input data as gray scale 60×40 image at each game tick. Next, there are three convolution layer with ReLU activation function. First convolution layer has 32 filters (size: 8×8), second convolution layer has 64 filters (size: 4×4), third convolution layer has 64 filters (3×3). Also, each convolution layer is with max pooling layer (2×2). The next layer is a fully connected layer with ReLU activation function (512 nodes). The final layer is an output layer. In ε-greedy, the number of nodes is equal to the number of actions. But, our proposed method needs 10 times more output nodes than normal. We choose 10 empirically considering diversity of outputs and computation time. Our proposed method needs more computation time then ε-greedy, but it's not significant.

Fig. 2. Neural network architecture



The ViZDoom's basic example environment allows three buttons (Left (L), Right (R) and Shoot (S)). We define 7 possible actions (press L, R, S, L+S, R+S, L+R and nothing). If press two keys in same time, like move (L and R) and S, AI perform two behavior at same time (fire pistol while moving), except press L+R. In this case, AI do nothing. Therefore, the number of output nodes is set as 7×10=70. It has 10 redundant sets, and each set has 7 outputs.

For the balance of exploration and exploitation, this method chooses one set randomly and selects an action with the highest Q-value from the set. As a result, the AI chooses from random actions (exploration) when there are disagreement on the highest Q-value action among the redundant sets. If Q-value across the sets are similar each other, it chooses the best action (exploitation). We update neural net's weights of the action over sets at once. In a testing mode, our method selects an action from the voting of all sets.

III. Experiments

We use ViZDoom's basic example for our experiments (Fig. 1). When starts a new game, there is a small room and a target on random position of opposite side of the room. The goal of this environment is to hit the target as soon as possible. AI player gets 100 points when hits the target, -6 points when misses the target, and -1 point each time nothing happened.

Our proposed method and ε-greedy use the same neural network architecture except the number of output nodes. We use mean squared error as an objective function and RMSProp as an optimizer (learning rate: 10^{-5}). We use replay memory size 10,000 and batch size 32, The e value of ε-greedy starts with 1.0 and gradually reduces during 20,000 training iterations to 0.1 after then it holds 0.1. We test model at each 5,000 iterations. In this time, we run 100 episodes with test mode action selection method.

Fig. 3. Total rewards in training and testing

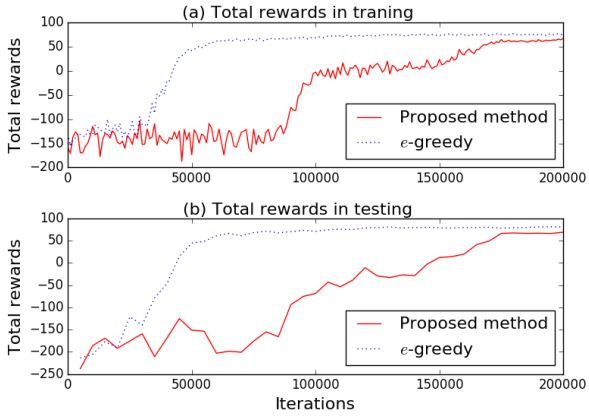


Fig. 3. shows experimental results. It shows total rewards change in training and testing (average of five experiments). According to experimental results, the proposed method gets total rewards similar to ε-greedy's one after enough training. Usually, final trained models of both models perform optimal behavior (move to proper position and shoot accurately). Although it takes more iterations, there are only one parameter (the size of redundancy in output nodes), but ε-greedy needs more (eg. ε's upper/lower bound, and update) parameters.

IV. Conclusion and Future Works

In this paper, we propose using redundant output to explore game environments in DQL. The most general method that can handle exploration and exploitation dilemma in reinforcement learning is the ε-greedy. Our proposed method can archive similar results with enough training iterations. It can adapt training progress with redundant output nodes.

References

[1] 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.

[2] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," arXiv:1606.01540 [cs], Jun. 2016.

[3] 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.

[4] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep Exploration via Bootstrapped DQN," arXiv:1602.04621 [cs, stat], Feb. 2016.