Deep Q Networks for Visual Fighting Game AI

Seonghun Yoon
Department of Computer Science and Engineering
Sejong University, Seoul, South Korea
kiboyz2@naver.com

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

***Abstract*— Recently, the introduction of vision-based deep Q learning demonstrated successful results in Atari, and Visual Doom AI platform. Unlike the previous study, the fighting game assumes two players with a relatively large number of actions. In this study, we propose to use deep Q Networks (DQN) for the visual fighting game AI competitions. The number of actions was reduced to 11 and the sensitivity of several control parameters was tested using the visual fighting platform. The experimental results show the potential of the DQN approach for the two-player real-time fighting game.**

Keywords—Deep Q-Learning; Fighting Game; Combination key; Reinforcement learning

I. INTRODUCTION

Fighting Game is an international artificial intelligence competition platform with matches between two agents. In the competition, Monte-Carlo Tree Search (MCTS) technique was successfully adopted in the top-ranked entries. Recently, the number of visual-based real time game AI has increased over the years. It has potential to employ the visual-based approach in the domain of the fighting game competition to defeat the MCTS-based agents. Especially, we adopted the Deep Q-learning Network (DQN) successfully demonstrated in Atari Games, and Visual Doom AI competitions [2]

Fig. 1. Picture of Fighting Game

Compared to previous studies on DQN, the fighting game AI platform suffers from the relatively large number of actions. For example, Atari platform usually has 4~8 actions per each game and the Visual Doom platform requires left, right, shoot, and change weapons. In the fighting game platform, it defines around 41 actions for each character. Several actions are defined for different states (Ground, In Air, Crouched, and Down). Because it's challenging to train agents with relatively large number of actions, we propose to reduce the actions by considering only 11 actions.

II. PROPOSED METHOD AND RESULTS

A. Deep Neural Network Architecture

- The DQN network model has two convolutional layers followed by two fully connected layers. In the input layer, there are 4 channels. Each channel inputs one of the sequential frames.
- In the first convolutional layer, there are 16 filters. The filter size is 8x8 with stride 4. In the second convolutional layer, there are 32 filters with filter size 4x4 with stride 2. After second convolutional layer, there is a fully connected layer with 256 hidden units. The output dimension is equal to the number of actions in the Fighting Game DQN agent.
- In Fighting Game, the total number of actions for each character is 41. We eliminate redundant actions and reduce to 11 actions. We use actions except for duplication of combination key between the GROUND state the AIR state.

  - Three of all actions is the direction. (Up, Down, and Right)

  - Two of them is the Punch and Kick.

  - Remaining actions are defined as a combo (combination), a set of actions pressed in sequence.

B. Parameter Setting

In this paper, the opponent was a simple static agent that stands without actions (named as None). The None agent was the idle state. The DQN agent's character was LUD, and the None agent's character is ZEN. The reward function was set by the difference between DQN agents hit points and the opponent agents hit point.

- The resolution of raw image is 960 x 640 pixels. It's rescaled into a small gray-scale image (e.g., 96 x 64).

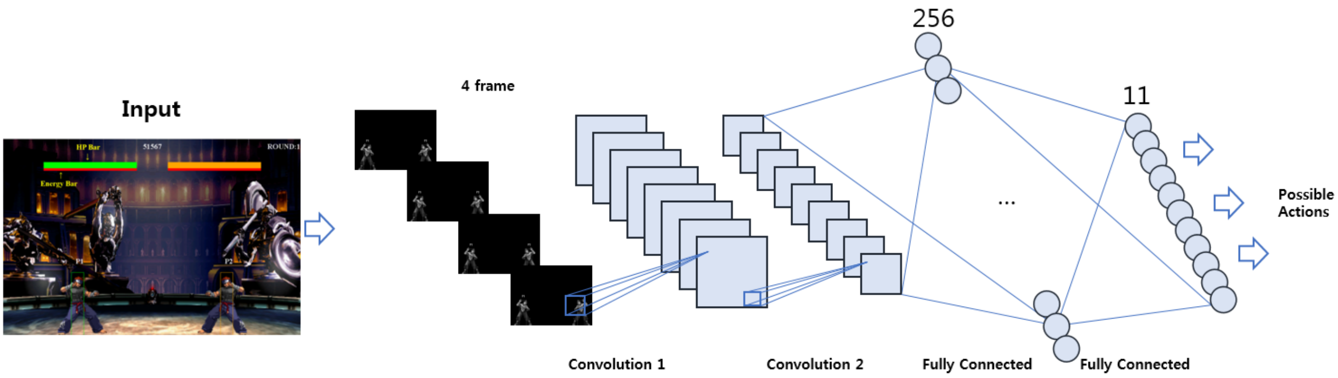

Fig. 2. The architecture of the neural network used for the experiment

- DQN agent produces one action using 4 frame inputs. Using frame skipping technique, we can speed up learning by ignoring multiple frames.
- The actions are Up, Down, Right, Punch, Kick, and six combination keys.
  - 'Down' + 'Down & Right' + 'Right & Punch'
  - 'Down' + 'Down & Right' + 'Right & Kick'
  - 'Right' + Down' + 'Right & Down & Punch'
  - 'Right' + 'Down' + 'Right & Down & Kick'
  - 'Down' + 'Left & Down' + 'Left & Punch'
  - 'Down' + 'Left & Down' + 'Left & Kick'
- We run training with 32-sized minibatch, 50,000-sized replay memory and 10^{-6} learning rate per each episode.

We run 1440 epochs for 5 times. At first, it selects an action randomly for the initial 3,500 frames. The epsilon was gradually decreased from 0.5 to 0.1 during 1,000,000 frames. All other parameters are determined similarly with the work on ATARI games [3]. All experiments run on Windows 10, Intel i7-3930K, Python 3.6 with NVIDIA GeForce GTX Titan Black.

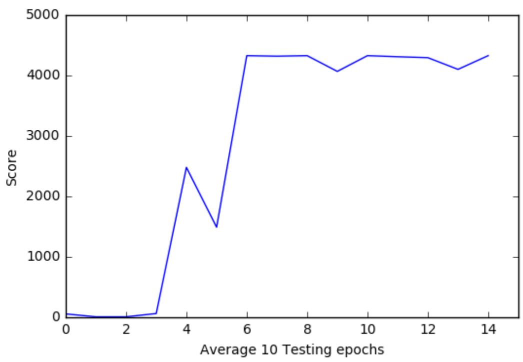

Fig. 3. Fighting Game learning progress. (DQN agent vs None agent)

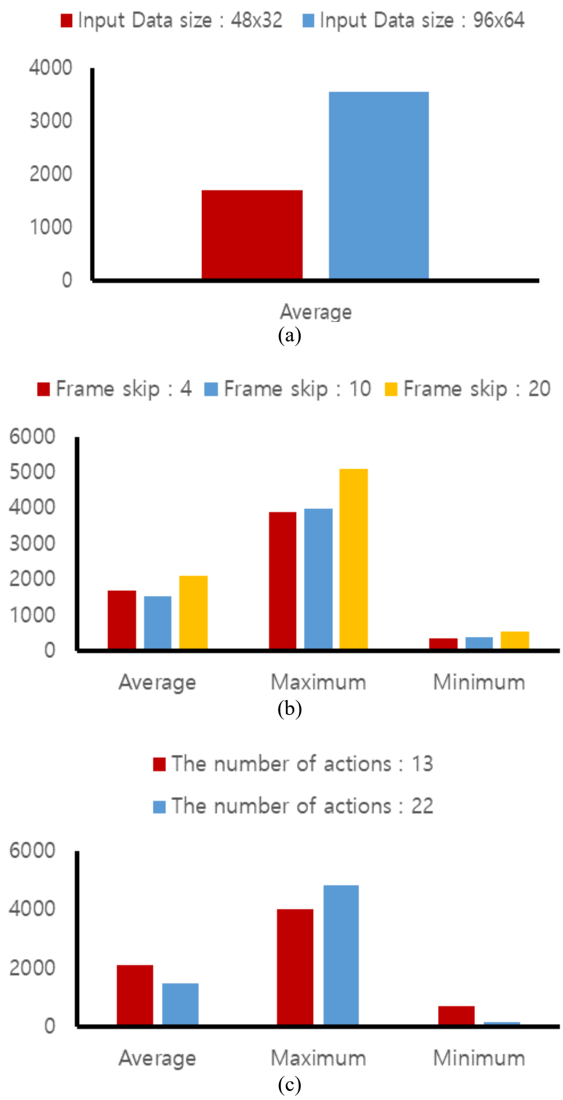

Fig. 4. Setting of the others. (DQN agent vs None agent) (a): Input Data Size (b): Frame Skip (c): The number of actions

C. *Experimental Result*

- Figure 3 shows the average score over 141 testing epochs (One testing epoch per 10 training epochs).
- Figure 4 shows the sensitivity of performance on the parameter change. It's the average of multiple runs given the parameter sets. As expected, it's desirable to use high-resolution input than the low-resolution one. In frame skip, it results in higher performance with large frame skip parameter value. The number of actions shows that the small-sized actions help to increase the performance. In addition to the eleven output actions, two additional actions are added with simple rule execution. In the case of the 22 actions, AIR actions are added into the basic set of actions.

III. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed the visual Fighting Game AI (DQN). In Figure 4, the results show that the combination keys are directly related to performance. Also, in Figure 4, the parameters are important in Fighting Game because it can control performance sensitively. We tested with the 6 combination keys and 5 single keys in this paper.

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).

REFERENCES

[1] M. Ishihara, T. Miyazaki, C.Y. Chu, T. Harada and R. Thawonmas, "Applying and improving Monte-Carlo Tree Search in a fighting game AI," ACM Int. Conf. Advances in Computer Entertainment Technology, 2016.

[2] Michal Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek & Wojciech Jaskowski, Poznan University of Technology, Poznan, Poland, "ViZDoom: A Doom-based AI Research Platform for Visual Reinforcement Learning."

[3] C. J. C. H. Watkins and P. Dayan. "Q-learning. Machine Learning," vol. 8, pp. 279–292, 1992.