Visualization of Deep Reinforcement Learning using Grad-CAM: How AI Plays Atari Games?

Ho-Taek Joo
Institute of Integrated Technology
GIST
Gwangju, South Korea
ureca87@gmail.com

Kyung-Joong Kim
Institute of Integrated Technology
GIST
Gwangju, South Korea
kjkim@gist.ac.kr

***Abstract* — Deep Reinforcement Learning (DRL) allows agents to learn strategies to solve complex tasks. It has been applied to solve various problems such as natural language processing, games, etc. However, it is still difficult to apply DRL to certain real-world problems because each action is not predictable, and we cannot know why the results are coming out. For this reason, a technology called eXplainable Artificial Intelligence (XAI) has been recently developed. As this technology shows a visualization of the AI process, people can easily understand the results of AI. In this paper, we proposed to use Grad-CAM, one of the XAI techniques, when we visualize the behaviors of AI players trained by DRL. Our experimental results show which part of the input state is focused on when one well-trained agent takes action.**

Keywords — Explainable AI, Grad-CAM, Deep Reinforcement Learning

I. INTRODUCTION

It is well known that Deep Reinforcement Learning (DRL) has played a significant role to solve the various tasks, but it has been still difficult to apply to certain AI systems which deal with personal information and assets such as financial, insurance, etc. Explainable Artificial Intelligence (XAI) [1] techniques aim to be trusted and easily understood by humans. By cracking open the black box of AI, XAI provides the evidence for the results, and validity of the process.

Gradient-weighted Class Activation Mapping (Grad-CAM) [2], which is one of the XAI techniques, shows the visualization result through a certain Convolutional Neural Network (CNN) layer in the reflection of the predicted class.

In this paper, we developed a method for applying Grad-CAM to DRL. For this method, we redesigned the state of the art DRL model called DeepMind's A3C [3], and then trained agents in various Atari Games. Our experimental results allow human to deeply understand how the trained agent observes the game screen and determines the action.

II. BACKGROUND

A. *Asynchronous Advantage Actor-Critic (A3C)*

After DQN [4] was developed, several DRL algorithms have been developed because of several limitations, such as memory complexity, computational complexity, and unstable learning. A3C is developed by DeepMind, and one of the best performing algorithm in the DRL. A3C is the policy gradient method and uses actor-critic networks. The actor learns a policy, and the critic evaluates the selected action by the policy. A3C's biggest novelty is asynchronous structure to improve learning speed and performance by using a global network and multiple parallel agents. Each agent learns policy through interaction in its environment, calculating the gradient and asynchronously updates to the global network. Each agent can make different explorations because the experience of each agent is independent. It is faster, more robust, and can score better than DQN.

B. *Grad-CAM*

Since AI outperformed human performance in many areas, visualization methodologies in the DL have been developed, such as deconvolution[5], gradient, and so on. Although deconvolution considers only the backward pass, Grad-CAM, one of the gradient methodologies, considers the backward pass and forward pass. In other words, Grad-CAM can obtain better visualization information that shows the local influence of the input image that affects the predicted class.

III. VISUALIZATION OF DRL AGENTS USING GRAD-CAM

A. *Extensions to A3C (E-A3C)*

In order to apply Grad-CAM to DRL, there are two requirements. The first is a well-trained agent, and the second is a deep neural network structure enables to clearly show visual attention through Grad-CAM.

In order to meet the two conditions mentioned above, we redesigned the state-of-the-art A3C algorithm. For example, the extended network structure includes max pooling layer after each CNN layer. Adding the max pooling layer is not a general method because it can cause loss of spatial information. However, our various experiments showed that adding max pooling layer made it easier to represent more implicit information in visualization such as object extraction. To complement this, we also expanded output channels each CNN layer. In our experiments, we selected six Atari Games and trained model agents for each game. Table I shows E-A3C model scores, and it is higher scores than normal A3C.

Table I.
Extension to A3C (E-A3C) scores

Tables	A3C-LSTM	E-A3C Scores	
		Best 100 episode average	Best Score
DemonAttack	115201.9	36277 ± 821.71	125307
SpaceInvaders	164766.0	79060.1 ± 5826.59	168337.0
BeamRider	24622.2	8441.22 ± 221.24	52117.0
Phoenix	74786.7	65150 ± 1442.07	75127.2
Pong	10.7	21.00 ± 0.00	21.0
Qbert	21307.5	21182.25 ± 276.12	27800.5

978-1-7281-1884-0/19 /$31.00 ©2019 IEEE

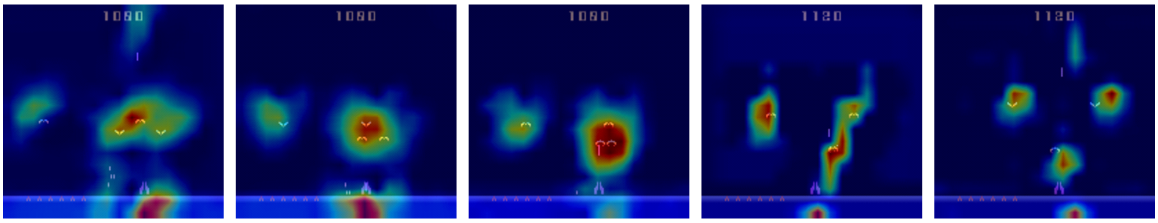

Fig. 2. Visualization of AI Plays on Demon Attack Atari Games.

B. Grad-CAM procedure for DRL

In order to apply the Grad-CAM to DRL, it goes through several steps. We first prepared each well-trained agents in selected Atari Games using E-A3C and tested. When testing, we collected the 160x160 RGB input images. The following is a sequence for applying Grad-CAM to DRL.

- We selected the last CNN layer to activate.
- We input collected images, did forward-propagation to the selected layer and saved the feature map (A^k) (2) and then proceeded again.
- By back-propagation from the predicted action (y^c) to the selected CNN layer, we can get the weighted-gradient for each node of the selected CNN layer. And then we calculated a global average pooling. (1)

$$w_k^c = \alpha_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \tag{1}$$

- We multiplied by the feature map (A^k) (2) and results of (1) and applied activation function; ReLU (2)

$$L_{\text{Grad-CAM}}^c = ReLU\underbrace{\left(\sum_k w_k^c A^k\right)}_{\text{linear combination}} \tag{2}$$

- The Formulation (3) describes the whole process for applying the Grad-CAM.

$$S^c = L_{\text{Grad-CAM}}^c = ReLU\left(\sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}}\right) \tag{3}$$

Fig. 1 shows the overall architecture of the sequences described above.

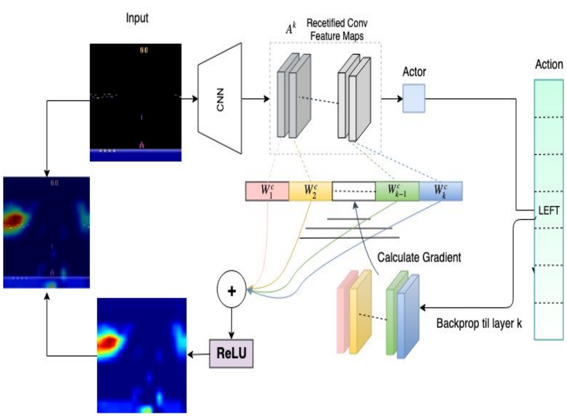

Fig. 1. Architecture for applying Grad-CAM to DRL

IV. RESULTS AND DISCUSSIONS

Fig. 2 shows the Grad-CAM results on consecutive frames when the trained agent plays Demon Attack. The results visualize information that passed through the last CNN layer considering input states and selected output action. Based on the results, we can see the role of the CNN layer that is to detect objects which are enemies, bullets, and the agent itself. Considering that the input states are continuous frames, we can assume that the trained agent is observing the movement of the objects found. We can also see that the role of the FC layer following the CNN layer chooses an action based on the detected objects. In summary, model-free DRL such as A3C detects objects in the CNN layer without distinction of objects and selects actions using objects detected in the FC layer. Our argument is supported in Fig. 3 by the Grad-CAM results of various Atari Games.

V. CONCLUSIONS

Our main contribution is to show directly the role of the CNN layer and indirectly the FC layer when the trained agent plays Atari Games by applying Grad-CAM to DRL. Our results also demonstrate how AI plays Atari Games and provide an opportunity for researchers to develop a deeper understanding of DRL and its learning process.

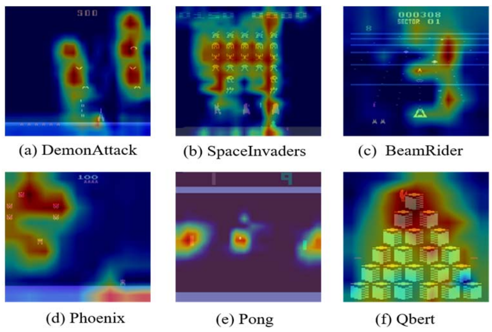

Fig. 3. Grad-CAM results for various Atari games

ACKNOWLEDGMENTS

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (grant no. 2017R1A2B4002164).

REFERENCES

[1] Gunning, David. "Explainable artificial intelligence (xai)." Defense Advanced Research Projects Agency (DARPA), nd Web (2017).
[2] Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." Proceedings of the IEEE International Conference on Computer Vision. 2017.
[3] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." International conference on machine learning. 2016.
[4] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.
[5] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in Computer vision–ECCV 2014. Springer, 2014, pp. 818–833.