Interpretable Saliency Map for Deep Reinforcement Learning

Hong Zheng¹, Yinglong Dai²³, Fumin Yu³, Yuezhen Hu³

¹School of Physics and Electronics, Hunan Normal University, Changsha, 410081, China
²College of Liberal Arts and Sciences, National University of Defense Technology, Changsha, 410073, China
³Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha, 410081, China

* daiyl@hunnu.edu.cn

Abstract. Deep reinforcement learning (deep RL) achieved big successes with the advantage of deep learning techniques, while it also introduces the disadvantage of the model interpretability. Bad interpretability is a great obstacle for deep RL to be applied in real situations or human-machine interaction situations. Borrowed from the deep learning field, the techniques of saliency maps recently become popular to improve the interpretability of deep RL. However, the saliency maps still cannot provide specific and clear enough model interpretations for the behavior of deep RL agents. In this paper, we propose to use hierarchical conceptual embedding techniques to introduce prior-knowledge in the deep neural network (DNN) based models of deep RL agents and then generate the saliency maps for all the embedded factors. As a result, we can track and discover the important factors that influence the decisions of deep RL agents.

Keywords: Intelligent Agent, Deep Reinforcement Learning, Deep Neural Networks, Interpretability.

1. Introduction

By incorporating the advanced deep learning techniques, deep reinforcement learning (deep RL) methods demonstrate excellent performance in many complex decision-making tasks. After Mnih et al. [1] proposed the deep Q-network (DQN) of Q-learning method, which has gained human-level performance in a majority of Atari games, researchers proposed a good many deep RL methods by combining deep learning and reinforcement learning methods, such as deep deterministic policy gradient (DDPG), asynchronous advantage actor-critic (A3C), proximal policy optimization (PPO), and soft actor-critic (SAC). Deep RL methods have achieved a lot of successes in various fields, e.g. the game of Go, robotics, and natural language processing. When employed in StarCraft II, the miraculous deep RL method even evolved to the grandmaster level in a very complicated multi-agent environment with enormous action space [2].

The behavior of artificial intelligence (AI) agents trained by deep RL methods can often be effective but not be interpretable for humans. Human beings cannot feel relieved when AI agents execute
important tasks without reason, particularly in the fields like automatic pilot and health care. On the other hand, the end-to-end training deep RL methods lacking interpretability are very difficult to debug and optimize. Deep RL agents with good interpretability will be a significant move before being deployed to real-world scenes. Accordingly, the interpretability of deep RL methods is becoming increasingly important. Despite its importance, the interpretable aspect of deep RL methods has not received enough attention yet. The difficulties are mainly from the poor interpretability of deep neural networks (DNNs) that applied to the deep RL models. And the interpretability of a model is difficult to measure. Recently, Greydanus et al. [3] applied saliency maps for deep RL to visualize and interpret the agents. Through using perturbation techniques to generate saliency maps, they demonstrated the effectiveness of the methods in explaining the agents’ attention, reasons for the decision, and the evolving learning process. Puri et al. [4] proposed a specific and relevant feature attribution (SARFA) to improve Greydanus’s work. The saliency maps generated by referring to the proposed attribution can focus more on the action-related features, so they can provide more accurate and focused interpretations. Although the perturbation-based methods can provide some related information between the observation data and the decision-making processes, the related information is not so clear none the less. Specifically, they still do not have explicit interpretability of the relationship between the action factors and input data.

In this paper, we propose a hierarchical conceptual embedding method to enhance the interpretability of saliency maps. In order to provide the explicit interpretability for the saliency maps and the selected actions, we propose to seek out the salient features that directly relate to certain actions. For instance, we need to locate the factors that affect the agents’ decisions. Consequently, we can directly track the decision-driven factors for the agents’ different behaviors.

The remainder of this paper is organized as follows. An overall review of the related work is presented in Section 2. Then, we introduce our methods to improve the interpretability of deep RL agents in Section 3, including the hierarchical conceptual embedding techniques and the corresponding saliency map generation methods. After that, some experiments are demonstrated in Section 4. Finally, we conclude this paper with some discussion in Section 5.

2. Related Work

After the success of DQN, Zahavy et al. [5] proposed a method that aggregated the state representations and used a t-Distributed Stochastic Neighbor Embedding (t-SNE) to analyze DQN in Atari games. Although the aggregated states on the t-SNE map can provide some insights for the agents’ behaviors, they cannot provide direct explanations of the agents’ policy. Recently, due to the advantage of directly visualizing the relationships between input data and decisions, interpretable methods based on saliency maps have become a popular approach to improve the interpretability of deep RL agents. The different saliency map methods can be classified into two main categories, i.e. gradient-based methods, and perturbation-based methods. Gradient-based methods compute the gradients to estimate the output variances with respect to the input variances, so as to discover the salient input features that will largely impact the results of the output. Perturbation-based methods search for the important features by replacing certain features in a domain-aware manner and judging the output variances. This approach is more intuitive and can provide straightforward explanations.

The interpretable methods can be borrowed from the DNN field because the bad interpretability of deep RL models mainly comes from the DNN parts. Researchers have a lot of interest in the perturbation-based methods for interpretable deep RL agents. Greydanus et al. fell back on the visualization techniques of DNNs, and they adopted the perturbation-based approach to generate saliency maps for the policy DNN model and the value DNN model of deep RL agents based on the A3C algorithm. Iyer et al. [6] presented object saliency maps to interpret DQN agents’ decisions. Rather than perturbing in pixel-level, they masked the objects to judge the impacts on Q-value. In order to provide more focused saliency maps related to the actions, Puri et al. proposed a specific and relevant feature attribution (SARFA) to measure the saliency features only relating to specific actions. However, they still cannot
provide direct interpretable factors that result in certain actions. In this work, we are going further to build direct relationships between the interpretable factors and the specific actions.

3. Methods
The interpretability of deep RL agents maybe not a clear statement, and there could be many different definitions on the basis of different aspects. However, it will not be wrong to say that a model has good interpretability if we can track and detail the decision process of the model. In this work, we aim to track the main decision process of deep RL agents and discover the salient features and interpretable factors that directly impact certain actions of the deep RL agents. At first, we propose to use hierarchical conceptual embedding techniques to build the DNN-based architectures in deep RL agents, so we can embed prior knowledge into the DNN architectures and constrain the representation spaces of deep RL agents. After that, we propose to generate saliency maps for the conceptual embedding activations in the DNNs, so we can discover the important factors and track the decision process of deep RL agents.

As the increasingly high-dimensional state space and action space, deep RL agents must explore exponentially growth spaces that face the problem of curse of dimensionality. Hierarchical learning architecture is a way to solve the problem by dividing a large problem into small subproblems and conquering each subproblem respectively so as to solve the high-level problem [7]. Hierarchical learning architecture partially splits the complicated problem while detailing the multi-level learning process from easy level to difficult level, so it can make the model easy to be interpreted. In order to explicitly demonstrate the interpretable factors aligned with human concepts, the conceptual embedding technique [8] can be utilized to embed prior knowledge in DNN models. We propose to combine hierarchical learning architecture and conceptual embedding techniques to improve the interpretability for deep RL agents, as illustrated in Fig.1.

The DNN-based policy model of a Deep RL agent produces actions according to the observations from the environment. Generally, it is a black-box for humans, and we care about only its input and output. The representation spaces of the hidden layers are often meaningless and too complicated to be understood by our concepts. Interpreting all the variables of the hidden neurons are far beyond human cognitive ability. Nonetheless, we can catch the important factors and reduce them into human-interpretable representation spaces.

The DNN-based model illustrated in Fig.1 can be regarded as a complex function that maps an observation $o$ to action through multi-layer non-linear transformation,
When we track back the DNN architecture from the output action layer by layer, the computational result of $a$ is from a previous sub-function that takes the previous computational result $(x^f_1, x^c_1)$ as input,

$$a = f(o).$$  \hfill (1)

Where $x^f_1$ and $x^c_1$ denotes the activation values of FNs and CNs respectively in macro-policy presentation layer 1, and $f_1(\cdot)$ is a non-linear transformation function. The activation values of CNs $x^c_1$ are meaningful representations related to human concepts. Similarly, we can continue to track back to the next layer,

$$(x^f_1, x^c_1) = f_2(x^f_2, x^c_2).$$  \hfill (2)

Where $x^f_2$ and $x^c_2$ denotes the activation values of FNs and CNs respectively in macro-policy presentation layer 2, and $f_2(\cdot)$ is another non-linear transformation function. We can also continue to track back to the input layer in a similar way. However, the shallow layers near the input layer are often used to extract shallow features, such as the edge information in images. We mainly care about the deep layers near the output layer that have abstracted high-level representations, such as macro strategy targets.

### 4. Experiments

To demonstrate the interpretable methods, we do experiments in Atari game environments with Open AI Gym API. The observation space of Atari environments is a video screen, which displays frames of color images of 160-pixel width and 210-pixel height. The frame rate is 60 frames per second (FPS) when running in real-time, and it can reach up to 6000 FPS at full emulation speed. The action space is comprised of 18 discrete numbers corresponding to the buttons of the joystick controller, and the different games just use a different minimal set of numbers. The reward can generally be defined by the game scores.

- **Jacobian**
- **Greydanus et al.**
- **Puri et al.**
- **Ours**

![Images](images.png)

(a) SpaceInvaders-v0.
We tested on SpaceInvaders-v0 and Breakout-v0. In order to compare the saliency maps generated by different methods, we deployed the same pre-trained model in the same game. The experimental comparisons of the saliency maps of related gradient-based and perturbation-based methods are illustrated in Fig.2. Different from the previous work, we relate the saliency maps with specific strategies and actions. The saliency maps can highlight the observation spots that have impact on the specific action decision. For example, the green spots motivate the model to choose the left action, and the blue spots motivate the model to choose the right action.

5. Conclusion
Interpretability is a key aspect of deep RL agents. To improve the interpretability of deep RL models, the difficulties are mainly from the DNN-based parts. Some researchers borrowed the techniques of saliency maps from the DNN research area to interpret the behaviors of deep RL agents, and they gained some insights into the relationships between the observations and actions. Based on the perturbation method, we proposed to use a hierarchical conceptual embedding method that introduces prior knowledge to constrain the representation spaces of DNN models. As demonstrated in the experiments, our method can produce clear saliency maps and indicate the action-driven factors explicitly.

We believe that it is indispensable to introduce prior knowledge to increase the interpretability of deep RL agents. Just as we need to teach our children systematic knowledge, so society can form the same common sense. In order to improve the interpretability of deep RL agents, we believe that prior knowledge embedding methods will be a potential approach.

Acknowledgment
This work is supported in part by Hunan Provincial Natural Science Foundation under Grant Number 2020JJ5367, Hunan Provincial Science and Technology Project Foundation under Grant Numbers 2018TP1018 and 2018RS3065.

References
[1] V Mnih, K Kavukcuoglu, D Silver, A A Rusu, J Veness, M G Bellemare, A Graves, M Riedmiller, A K Fidjeland and G Ostrovska 2015 Human-level control through deep reinforcement learning J. Nature 518 pp529–533
[2] O Vinyals, I Babuschkin, W M Czarnecki, M Mathieu and D Silver 2019 Grandmaster level in StarCraft II using multi-agent reinforcement learning J. Nature 575 pp350–354
[3] S Greydanus, A Koul, J Dodge and A Fern 2018 Visualizing and understanding Atari agents ARTIF Proc. of the 35th Int. Conf. on Machine Learning pp1792–1801
[4] N Puri, S Verma, P Gupta, D Kayastha, S Deshmukh, B Krishnamurthy and S Singh 2020 Explain your move: Understanding agent actions using specific and relevant feature attribution
[5] T Zahavy, N Ben-Zrihem and S Mannor 2016 Graying the black box: Understanding DQNs Proc. of The 33rd Int. Conf. on Machine Learning pp1899–1908

[6] R Iyer, Y Li, H Li, M Lewis, R Sundar and K Sycara 2018 Transparency and explanation in deep reinforcement learning neural networks AAAI/ACM Conf. on Artificial Intelligence Ethics and Society pp144-150

[7] Z J Pang, R Z Liu, Z Y Meng, Y Zhang, Y Yu and T Lu 2019 On reinforcement learning for full-length game of Starcraft Proc. of the AAAI Conf. on Artificial Intelligence pp4691–4698

[8] Y Dai, G Wang and K C Li 2020 Conceptual alignment deep neural networks J. Journal of Intelligent & Fuzzy Systems 34 pp1631–1642