Evaluating Correctness of Reinforcement Learning based on Actor-Critic Algorithm

Youngjae Kim, Manzoor Hussain, Jae-Won Suh and Jang-Eui Hong
College of Electrical & Computer Engineering, Chungbuk National University
Cheongju, South Korea
youngjae@cbnu.ac.kr, hussain@selab.cbnu.ac.kr, sjwon@cbnu.ac.kr, jehong@cbnu.ac.kr

***Abstract*— Deep learning is used for decision making and functional control in various fields, such as autonomous systems. However, rather than being developed by logical design, deep learning models are trained by itself through learning data. Moreover, only reward values are used to evaluate its performance, which does not provide enough information that the model learned properly. This paper proposes a new method to assess the correctness of reinforcement learning, considering other properties of the learning algorithm. The proposed method is applied for the evaluation of Actor-Critic Algorithms, and correctness-related insights of the algorithm are confirmed through experiments.**

Keywords—reinforcement learning, actor-critic algorithm, safety-critical system, quality evaluation, correctness

I. INTRODUCTION

The rapid development of artificial intelligence (AI) technology is making a huge impact on human life. For example, autonomous robots and cars can drive autonomously without human intervention, the AI-based speech recognition understands and executes human commands based on its excellent voice recognition ability. Also, AlphaGo of Google Deep Mind [1] beat the top-ranked human player of game Go. Classical AI before deep learning was designed and developed using deductive and logical approaches while modern AI based on deep learning uses a huge amount of data that learns automatically hence the development process is inductive. However, the problem with inductive methods is the evaluation process and it is hard to deduce reason regarding the results driven by inductive approaches. The modern deep learning has black-box characteristics from the human point of view [2] as black-box models are not explainable by themselves. Hence we need several techniques to extract the explanations from the inner logic or outputs of the models. Although research on explainable AI [3-5] is being conducted to understand and track the decision-making process of deep learning. However, there are many research gaps and challenges that needed to be solved in this area. In particular, the application of such AI algorithms used in safety-critical areas may pose serious threats to safety.

Reinforcement learning (RL) is a branch of AI that uses deep neural networks (DNNs) that learn the optimal policy from the experience of the agent through the reward against the actions performed in a given state [6]. The application of the RL is growing exponentially in various fields such as games, robotic controls, and autonomous vehicles. The traditional approach uses reward values to evaluate the performance of RL algorithms. Although such evaluation techniques evaluate the problem-solving ability and learning speed of the algorithms. However, there are different other aspects needed to be considered while evaluating these algorithms. For example, the conventional evaluation does not consider the safety, and robustness attributes while evaluating the algorithms. Using the conventional techniques, it is hard to determine the correctness, safety, and robustness attributes specifically when it is being used in safety-critical scenarios.

Therefore, this paper proposes a novel technique to evaluate the correctness of RL algorithms. The proposed method was used to evaluate the different types of actor-critic algorithms [7]. The algorithm used in our experiment has an actor network that learns policy and a critic network that learns value. Our method defines and evaluates the correctness of the actor network and critic network respectively. By comparing the ideal value derived and the value generated by each network, we evaluate how closely the actor network learned toward the ideal behavior and how closely the critic network learned toward the ideal value. We compare and analyze the correctness for the three types of algorithms using two types of activation functions. From the experimental results, we confirmed that there is a new attribute, which cannot be found in the reward-value-based evaluation methods. The proposed approach also provides many insights to identify the weaknesses of the RL algorithms. The contribution of this paper can be summarized as follows:

- A method for evaluating the correctness through the learned policy network and value network of the actor-critics algorithms
- Evaluation of three mostly used actor-critic algorithms such as DDPG and SAC, and TD3 used to evaluate their correctness using the proposed technique.
- Exploring the insights of the changes in the policy network that were not confirmed by the existing reward-based methods. Furthermore identification of the weaknesses of the RL algorithm.

The rest of the paper is structured as follows. Section 2 analyzes related work on AI quaility attributes and RL, Section 3 proposes a correctness evaluation method. Section 4 presents the experimental design and its results for the evaluation, and Section 5 concludes our work and explains future work.

II. Research Background

A. AI Software Quality Attributes

The research on deep learning-based AI applications are rapidly growing especially in safety-critical systems like autonomous vehicles, and medical diagnostics. The need for reliable AI models has also grown. Meanwhile, the quality assurance of AI-based software applications has emerged as a new problem.

To accommodate these needs, the QA4AI (Quality Assurance for AI-based products and services) consortium was formed in 2018 and announced the quality assurance guideline for AI products in 2020 [2]. This guideline introduces five viewpoints for the quality of deep learning software applications. For each perspective, they defined checkpoints that need to be considered for quality assurance. In addition, in order to improve the quality of deep learning-based software, it was recommended that the quality of each point of view should be well balanced.

Several researchers worked on AI-based software quality. The study related to the quality assurance of deep learning conducted by J. Siebert et al. [8] classifies the quality perspective of deep learning systems into five categories: Model, Data, System, Infrastructure, and Environment, and the quality attributes of each classification were defined properly. Since this study was conducted mainly considering supervised learning. Quality evaluation of other machine learning algorithms more specifically RL algorithms was not discussed in this study.

B. Reinforcement Learning

RL is about an agent that interacts with the environment and learns an optimal policy by trial and error. When a positive reward is received from the environment against an action, the action is evaluated as good and reinforced, whereas when a negative reward is given, the action is evaluated as bad action and weakened. For a long time, RL was used to learn in discrete action space, but it has been successfully applied to environments with continuous action space such as robotic control [9].

Currently, among the RL algorithm, the actor-critic algorithm [10] is the most widely used algorithm. Several actor-critic algorithms have been proposed by different researchers such as the SAC (Soft Actor-Critic) [11], the TD3 (Twin Delayed DDPG) [12] and the TQC(Truncated Quantile Critics) [13]. These are the widely used and representative actor-critic algorithms in the environment having continuous action space.

The Gym library [14] is popularly for evaluating the RL algorithms. It provides various environments and makes it easy to compare the performance of different RL algorithms.

III. Evaluation Method for Correctness

One of the most important quality attributes of actor-critic RL earning is the correctness, which is the degree of how accurately the learned network outputs meet the original purpose of each network.

The goal of RL is to maximize rewards. The policy network (Actor) has a sub-objective to produce output that maximizes the sum of future rewards, and the value network (Critic) has a sub-objective to learn the value of the current state and behavior. In this paper, we evaluate how much of these sub-goals have been achieved.

A. Research Motivation

A commonly used indicator to evaluate the performance of RL is to monitor increasing or decreasing trends in reward value. At the beginning of learning, the agent acquires a low reward value from the environment, but as the learning progresses, the reward value increases, indicating that the agent solves the assigned task in the environment and the agent learning target is well achieved.

Recent studies [13, 14] on the RL generally have shown that their algorithms can solve the assigned task in the environment by increasing the reward value. However, it is difficult to know whether the task was solved by correct action value with enough learning. In particular, when RL is applied to a safety-critical system where dangerous behavior can cause loss of life and property. It is very critical to carefully evaluate whether the learned agent does not cause dangerous behavior. It is hard to ensure that the developed system based on RL has a reliable model in it. Therefore, the evaluation method of the correctness of RL becomes an important research issue.

B. Environment for Providing Ideal Values

The reason for using deep learning for AI is to entrust the complex judgment process of human logical thinking to machine learning. A deep learning agent normally trained using data can solve problems on its own without human intervention. In some areas, AI agents have achieved performance that surpasses humans. However, in an environment that provides rewards values to an agent when an action is performed. Only reward value-based evaluation does not provide enough information regarding other attributes such as correctness. In this paper, we define a Provisioning Ideal Values (PIV) environment that can measure ideal values in an environment. The derived ideal value is used to evaluate the correctness of networks after learning is finished. The important fact is that, during the learning, the RL agent does not know the ideal value in a given state at all. Therefore, if an RL agent solves various tasks of PIV environments well and discovers a common characteristic, we can generalize that it will exhibit the same characteristics even in an environment where the ideal values is unknown.

To achieve this goal, the PIV environment requires a function to find the ideal behavior by obtaining the maximum sum of future rewards in a given state. Let S be the set of states obtained by the agent observing the environment, and let A be the set of actions of the agent. The function ${}^{i}f: S \rightarrow A$ satisfies ${}^{i}a \in {}^{i}f(s)$ as a function to find the ideal behavior ${}^{i}a \in A$ in a given state $s \in S$ then the function ${}^{i}f$ can be used to evaluate the actor correctness.

Additionally, the function $g: S \times A \rightarrow S \times R \times D$, where R is the set of reward values, and D is the set of done signals of the Boolean type, is provided to evaluate critical networks. This

function is similar to the 'step' function in the gym library, however, the difference is that it can obtain a transition from any state to the next state.

C. Correctness Evaluation of Actor-Critic Algorithm

Let S be the set of states of the trained agent and A be the set of actions performed in given states. The input of the policy network π_θ for the parameter θ is a state $s \in S$, and the output is a set of actions $a \in A$ such that $a = \pi_\theta(s)$. In this case, the a is a tuple whose elements are real values $a_{(1)}, a_{(2)}, a_{(3)}, \cdots, a_{(n)}$ generated by n neurons that constitute the output layer of the policy network π_θ, and $n = \dim(A)$. Therefore, we can define a vector $\vec{a}$ of an n-dimensional vector space using each element of tuple a such that:

$$\vec{a} = \left(a_{(1)}, a_{(2)}, a_{(3)}, \cdots, a_{(n)}\right) \quad (1)$$

Also, from two actions a_p ($\in A$) and a_q ($\in A$) having similar actions value and two vectors $\overrightarrow{a_p}$, $\overrightarrow{a_q}$ having similar lengths and directions. Various methods exist to measure the similarity of two vectors, we use the normalized L2-norm.

· Normalized L2-norm: L2-norm, also called Euclidean distance, is a method of measuring similarity by using the property that the distance between the endpoints of the two vectors becomes shorter as the lengths of two vectors are similar or the angle between the two vectors is smaller. However, the L2-norm is a difficult method to accurately measure the similarity of vectors with a large difference in length. Therefore, we used normalized L2-Norm in which a vector $\overrightarrow{M_A}$ consisting of the maximum value of each element of action set A, that is,

$$\overrightarrow{M_A} = \left(\max\left(a_{(1)}\right), \max\left(a_{(2)}\right), \max\left(a_{(3)}\right), \cdots, \max\left(a_{(n)}\right)\right).$$

The normalized L2-norm $dist$ we defined for two vectors, $\vec{p} = \left(p_{(1)}, p_{(2)}, p_{(3)}, \cdots, p_{(n)}\right)$ and $\vec{q} = \left(q_{(1)}, q_{(2)}, q_{(3)}, \cdots, q_{(n)}\right)$ in an n-dimensional vector space is given by the following equation (Equ. 2).

$$dist = \sqrt{\sum_{k=1}^{n} \left(\frac{p_{(k)}}{M_{A(k)}} - \frac{q_{(k)}}{M_{A(k)}}\right)^2} \Big/ \sqrt{\dim(A)} \quad (2)$$

where $dist$ has a range from $[0, 2]$. The value of $dist$ approaches 0 when the two vectors are similar.

Now in the following, we explain how the correctness of the critic network outputs value for the actions taken in the current state. The critical network of RL is learned to satisfy the following equation called Bellman equation [6].

$$q(s_t, a_t) = r_t + \gamma \cdot (1 - d_t) \cdot q(s_{t+1}, a_{t+1}) \quad (3)$$

where $q: S \times A \to \mathbb{R}$ is a value function, $r \in R$ is a reward, γ is a discount rate, and d means a done signal, for any step t. This difference between the two sides is called an error, and the objective of RL is to minimize this error. Hence, The error function $h: S \to \mathbb{R}$ for state s is defined as follows.

$$h(s_t) = \left\| r_t + \gamma \cdot (1 - d_t) \cdot q(s_{t+1}, a_{t+1}) - q(s_t, a_t) \right\| \quad (4)$$

where $a_t = \pi_\theta(s_t)$, $a_{t+1} = \pi_\theta(s_{t+1})$ and the (s_{t+1}, r_t, d_t) can get from function $g(s_t, a_t)$ in PIV environments.

Based on the above-mentioned methods, the procedure for evaluating the correctness of actors and critics in RL is as follows;

(1) Obtain m number of states $s_1, s_2, s_3, \cdots, s_m$ ($s_k \in S$, $k = 1, 2, 3, \cdots, m$) from the set of states S using Continuous Uniform Distribution.

(2) Find the actions $a_1 = \pi_\theta(s_1)$, $a_2 = \pi_\theta(s_2)$, $a_3 = \pi_\theta(s_3)$, $\cdots$, $a_m = \pi_\theta(s_m)$ for the obtained m states $s_1, s_2, s_3, \cdots, s_m$, respectively using the policy network π_θ.

(3) Find the ideal behaviors ${}^{i}a_1 = {}^{i}f(s_1)$, ${}^{i}a_2 = {}^{i}f(s_2)$, ${}^{i}a_3 = {}^{i}f(s_3)$, $\cdots$, ${}^{i}a_m = {}^{i}f(s_m)$ for the obtained m states $s_1, s_2, s_3, \cdots, s_m$, respectively using the function ${}^{i}f$ that calculates the ideal behavior.

(4) For $k = 1, 2, 3, \cdots, m$, where a_k and ${}^{i}a_k$ are transformed into vectors $\overrightarrow{a_k}$ and $\overrightarrow{{}^{i}a_k}$, respectively, and then calculate the normalized L2-norm for actor correctness.

(5) Evaluate the actor correctness of policy network by mean squared error of m normalized L2-norms

(6) Get the next state, reward, done signal for (s_1, a_1), (s_2, a_2), (s_3, a_3), $\cdots$, (s_m, a_m) respectively using the function g.

(7) Evaluate the critic correctness of value network by mean squared error of m error values.

This method of evaluating correctness attributes by directly examining the learned network has the following advantages over the existing reward values-based evaluation.

- Existing evaluation method builds a test environment and uses the reward received from the environment. However, our method for evaluating the correctness can directly evaluate the network itself, so that it can be a more fundamental evaluation of the learning process. Additionally, as the interactions in the test environment are not required, the computing resources required to build the test environment can be reduced.
- In conventional RL evaluation methods, the transitions in the test environment can occur only for some states in response to the learned results, therefore, it is difficult to evaluate the states in which no transitions have occurred. On the other hand, our method solves such a problem because the entire state set is uniformly investigated.

IV. Experimental Design and Its Results

A. Design of Evaluation Method

The correctness of the three most widely used RL algorithms such as DDPG, TD3, and SAC, was evaluated in a single environment. Each algorithm was trained using two activation functions such as the ReLU function [26] and hyperbolic tangent function (Tanh). After learning was completed, the correctness at each epoch was evaluated, and the change in the network was properly investigated.

To validate the proposed method we consider an autonomous robot as a running example in our experiment. We assumed that an autonomous robot learns the direction of

movement according to its location. The PIV environment for correctness evaluation was designed, as follows.

As shown in Fig 1(a), there are 7 concentric circles with radius 0.2, 0.5, 0.8, 1.1, 1.4, 1.7, and 2.0 respectively in two-dimensional space. The observation space is the entire interior of the largest circle, the agents are located in the observation space which is represented by the points within the space. This agent has a state $s = (x, y)$ which consists of x coordinates and y coordinates that indicate its location. The actions of this agent are expressed as $a = (a_{(1)})$ which is a real number $a_{(1)}$. These real numbers have the ranges $[-\pi, +\pi]$ that determine the direction of movement. The agent that started the episode at any point in observation space, moves inside the circle according to the output action value, and the episode ends when it touches one of the seven circles.

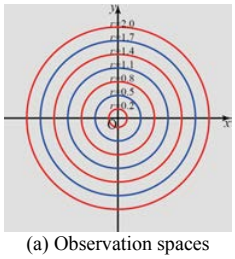
In Fig 1(a), when the agent touches the circle represented in red color, it receives the reward value $+100$ in positive, and when the agent touches the blue-colored circle, it receives the negative reward value of -100. Thus, the agent learns to move from its current position toward the red circle. In addition, the agent will receive negative rewards if it does not move in parallel diriections to the radius. So, in this environment, the agent will learn how to get to the red circle as quickly as possible.

In this environment, Fig 1 (b) shows the ideal behavior in each state s. In the figure, the horizontal axis means the x value of the state, the vertical axis means the y value of the state, and the ideal behavior value in each state is converted and represented into color. Therefore, red indicates the action value of $+\pi$, and blue indicates the action value of $-\pi$. The RL agent has to learn to yield the target value of the policy network as shown in Fig 1(b).

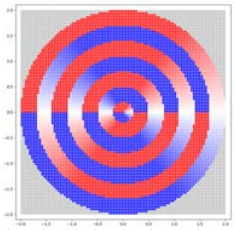
B. *Evaluation Results of Correctness*

Fig 2(a) shows the visual representation of whether the behavioral output values of the learned policy network are approaching to the ideal values. Each action value is converted into colors as illustrated in the above-mentioned figure.

Similarly, the Fig 2(b) is a visualization of changes in actor correctness and Fig 2(c) is a visualization of changes in critic correctness. The average normalized L2-norm has a value closer to 0 (i.e., it is closer to the ideal value). Therefore the highest actor correctness is achieved when the entire state space of the figure is filled with dark blue. The 0 epoch means the initial neural network before learning.



(a) Observation spaces



(b) Ideal behavior in each state

Fig. 1. Evaluation Environment

From Fig 2(b) and Fig 2(c), we can visually confirm that the actor correctness, and critic correctness are close to the ideal value as the learning proceeds. From the experimental result, it can also be confirmed that the experiment using the ReLU activation function generates more ideal behavior values than the Tanh activation function.

Fig 3(a) depicts the change in rewards received from the environment during learning while the graph in Fig 3(b) presents the variations in critic correctness as the learning process progresses. From Fig 3(b), it can be seen that, as the learning progresses, the critic correctness also gradually increases. The decreasing trends in the graph represents that the correctness is increasing. This shows that the value network is gradually being learned to output values close to the ideal value. Fig 3(c) shows the actor correctness when the movements of agent are same to the ideal action direction. While Fig 3(d) shows the actor correctness values when the movements of agent are opposite to the ideal action direction. In both cases, the decreasing trends on the graphs indicates that the actor correctness is increasing.

C. *Finding Insight from Experiment Result*

In contrast to the reward value-based RL evaluation methods, our proposed evaluation method provides various observation results. These observations can be used to analyze the causes of gains and losses in RL outputs and provides guidelines for the application of RL in systems that are operated in the safety-critical domains. The insights obtained from the novel evaluation method can be used as a verification technique to confirm the correctness of the RL models. The deduced insights can help to develop more reliable, robust, and safe RL-based safety-critical systems. In the following we report the insights obtained from our experiments as follows:

- In all experiments, the behavioral correctness was significantly reduced in those areas where the ideal action values were not continuous.
- In all experiments, in the case of having a small state value (about $(-0.2, +0.2)$) fails to achieve an appropriate action value. This is due to some specific problem with transition distribution or neural network structure. During the experiment, we confirmed that the value network was not learned in those areas.
- In Fig 3(d), the DDPG algorithm with Tanh activation functions generates a significantly incorrect value even though the model has high accuracy in the normal direction. It results in a low reward, meaning that there is a high possibility of uncertain behavior of the RL agent.
- In contrast, the SAC algorithm achieves high rewards by minimizing inaccurate behavior.
- From Fig 3(a) and Fig 3(b), it can be observed that the critic correctness tends to be proportional to the rewards.

Generally in RL, when the average reward value increases to a certain point and then converges. This indicates that the agent has solved the environmental task and does not need to learn anymore. Therefore, continuous learning causes an overfitting problem. However, even after the reward converges, we observed that a continuous change in values occurs, this

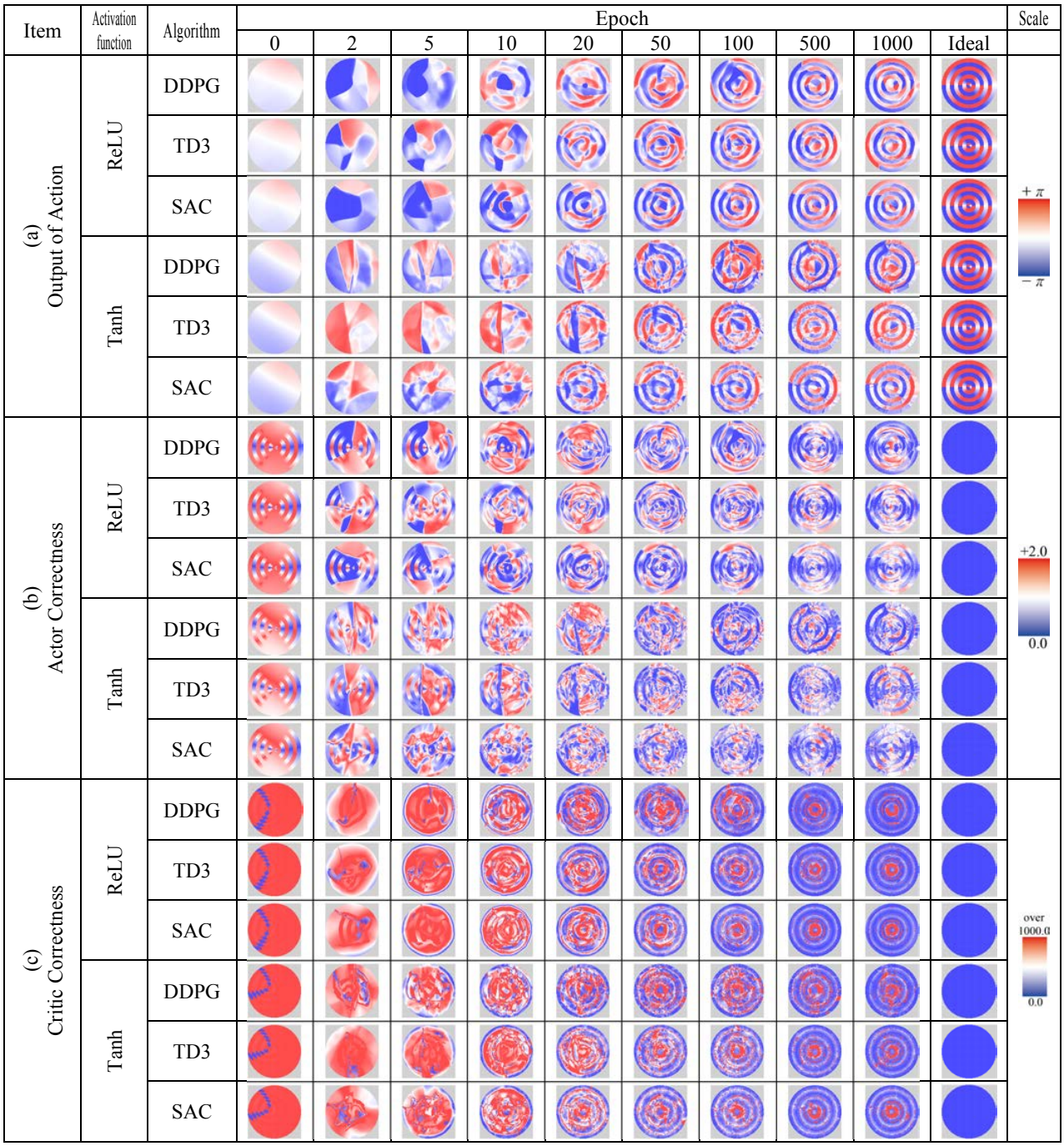


Fig. 2. Visualization of variations in learning progresses

phenomenon was observed inside the network. This means that while developing an RL-based system that requires high correctness, it is very important to decide when to stop the learning process. However, the conventional evaluation method does not exploit these insights.

V. Conclusion and Future Works

This paper explained the actor network and critic network correctness, which is one of the quality attributes of the actor-critic RL algorithm, and proposed a method to evaluate the correctness. In addition, in order to measure how close is the output of the RL algorithm to the ideal values, we evaluated the

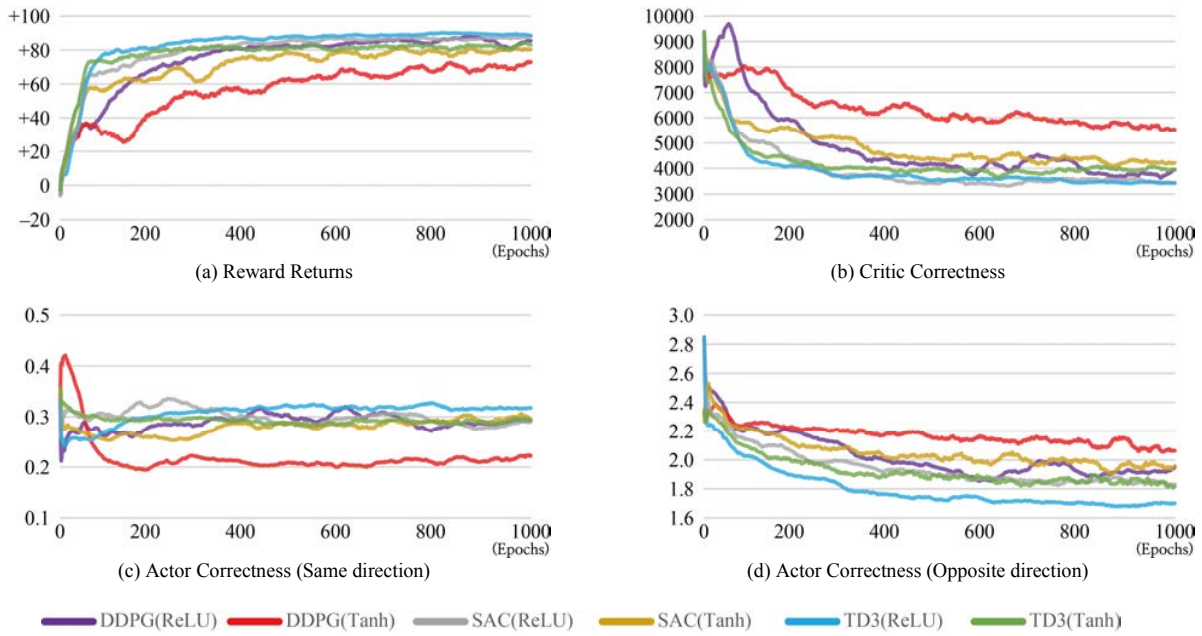


Fig. 3. Evaluation Result of reward return and critic correctness

correctness after learning the three most widely used algorithms such as DDPG, SAC, and TD3, through various activation functions.

From these evaluation results, new attributes of RL that could not be exploited by existing evaluation methods were confirmed. Therefore, we argue that the proposed evaluation method will be useful in evaluating the quality of RL algorithms.

We are aimed to diversify the application of our proposed method so that it can be used to improve the performance of RL algorithms in the future. Fisrtly, we are planning to develop diversified PIV environments, so that they can be used to evaluate and analyzed the characteristics of RL algorithms, activation functions, and the size of neural networks. Secondly, correctness can be used to evaluate other quality attributes of RL, such as robustness, stability, and obliviousness. We will continue our research to analyze various properties.

ACKNOWLEDGMENT

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT. (NRF-2020R1A2C1007571).

REFERENCES

[1] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, ... & D. Hassabis, "Mastering the game of go without human knowledge," nature, vol.550, no.7676, pp.354-359, 2017.

[2] G. Fujii, K. Hamada, F. Ishikawa, S. Masuda, M. Matsuya, T. Myojin, ... & Y. Ujita, "Guidelines for quality assurance of machine learning-based artificial intelligence," International journal of software engineering and knowledge engineering, vol.30, no.11n12, pp.1589-1606, 2020.

[3] A. Holzinger, "From machine learning to explainable AI," In proceedings of the World symposium on digital intelligence for systems and machines (DISA), IEEE, pp.55-66, 2018.

[4] R. R. Hoffman, G. Klein & S. T. Mueller, "Explaining explanation for "Explainable AI"," In Proceedings of the human factors and ergonomics society annual meeting, SAGE Pubilications, pp.197-201, 2018.

[5] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen & K. R. Müller, "Explainable AI: interpreting, explaining and visualizing deep learning," Springer Nature. 2019.

[6] R. S. Sutton, A. G. Barto, "Reinforcement learning: An introduction," MIT press, 2018.

[7] V. Konda, J. Tsitsiklis, "Actor-critic algorithms - Advances in neural information processing systems," MIT press, 1999.

[8] J. Siebert, L. Joeckel, J. Heidrich, K. Nakamichi, K. Ohashi, I. Namba, ... & M. Aoyama, "Towards guidelines for assessing qualities of machine learning systems," In proceedings of the International conference on the quality of information and communications technology, Springer, pp.17-31, 2020.

[9] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, ... & D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.

[10] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, ... & K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," In proceedings of the International conference on machine learning, PMLR, pp.1928-1937, 2016.

[11] T. Haarnoja, A. Zhou, P. Abbeel & S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," In proceedings of the International conference on machine learning, PMLR, pp.1861-1870, 2018.

[12] S. Fujimoto, H. Hoof & D. Meger, "Addressing function approximation error in actor-critic methods," In proceedings of the International conference on machine learning, PMLR, pp.1587-1596), 2018.

[13] A. Kuznetsov, P. Shvechikov, A. Grishin & D. Vetrov, "Controlling overestimation bias with truncated mixture of continuous distributional quantile critics," In the proceedings of the International conference on machine learning, PMLR, pp.5556-5566, 2020.

[14] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang & W. Zaremba, "Openai gym," arXiv preprint arXiv:1606.01540, 2016.