KIPS Trans. Comp. and Comm. Sys.
Vol.10, No.5 pp.155~162
ISSN: 2287-5891 (Print), ISSN 2734-049X (Online)

https://doi.org/10.3745/KTCCS.2021.10.5.155

Max-Mean N-step Temporal-Difference Learning Using Multi-Step Return

Gyu-Young Hwang† · Ju-Bong Kim†† · Joo-Seong Heo†† · Youn-Hee Han†††

ABSTRACT

n-step TD learning is a combination of Monte Carlo method and one-step TD learning. If appropriate n is selected, n-step TD learning is known as an algorithm that performs better than Monte Carlo method and 1-step TD learning, but it is difficult to select the best values of n. In order to solve the difficulty of selecting the values of n in n-step TD learning, in this paper, using the characteristic that overestimation of Q can improve the performance of initial learning and that all n-step returns have similar values for $Q \approx Q^*$, we propose a new learning target, which is composed of the maximum and the mean of all k-step returns for $1 \le k \le n$. Finally, in OpenAI Gym's Atari game environment, we compare the proposed algorithm with n-step TD learning and proved that the proposed algorithm is superior to n-step TD learning algorithm.

Keywords : Reinforcement Learning, Q-learning, DQN, n-step Temporal-Difference Learning

멀티-스텝 누적 보상을 활용한 Max-Mean N-Step 시간차 학습

황 규 영† · 김 주 봉†† · 허 주 성†† · 한 연 희†††

요 약

n-스텝 시간차 학습은 몬테카를로 방법과 1-스텝 시간차 학습을 결합한 것으로, 적절한 n을 선택할 경우 몬테카를로 방법과 1-스텝 시간차 학습보다 성능이 좋은 알고리즘으로 알려져 있지만 최적의 n을 선택하는 것에 어려움이 있다. n-스텝 시간차 학습에서 n값 선택의 어려움을 해소하기 위해, 본 논문에서는 Q의 과대평가가 초기 학습의 성능을 높일 수 있다는 특징과 $Q \approx Q^*$ 경우, 모든 n-스텝 누적 보상이 비슷한 값을 가진다는 성질을 이용하여 $1 \le k \le n$에 대한 모든 k-스텝 누적 보상의 최댓값과 평균으로 구성된 새로운 학습 타겟인 $\Omega-return$을 제안한다. 마지막으로 OpenAI Gym의 Atari 게임 환경에서 n-스텝 시간차 학습과의 성능 비교 평가를 진행하여 본 논문에서 제안하는 알고리즘이 n-스텝 시간차 학습 알고리즘보다 성능이 우수하다는 것을 입증한다.

키워드 : 강화 학습, Q-learning, DQN, n-스텝 시간차 학습

1. 서 론

n-스텝 시간차 학습은 몬테카를로 방법과 1-스텝 시간차 학습을 결합한 것으로, n-스텝까지 관찰된 누적 보상과 n번째 스텝에서 기대되는 행동 가치가 결합된 값을 학습의 업데이트 타겟으로 활용한다. n의 값을 적절하게 선택할 경우 n-스텝 시간차 학습은 강화 학습의 성능을 높일 수 있어 1-스텝 시간차 학습보다 좋을 수 있다[1,14]. 그러나 n-스텝 시간차 학습은 n의 값이 커질수록 행동 가치의 분산이 높아져 환경에 맞는 적절한 n의 값을 찾기가 어렵다[22].

최근 연구에서는 n-스텝 시간차 학습의 변형을 제안했다[3-5]. 하지만 학습의 가속화를 위한 최적의 n값을 찾는 방법에 대한 논의는 없었다. 또 다른 연구에서는 TD(λ)를 제안하였는데, 이는 n-스텝 시간차 학습에서 적절한 n의 값을 찾기가 어렵기 때문에 모든 n-스텝 누적 보상을 활용하여 학습의 업데이트 타겟을 계산하는 알고리즘이다[2,6,7]. 이 알고리즘은 λ라는 하이퍼-파라미터를 사용하여 모든 n-스텝 누적 보상에 가중치를 다르게 주어 평균을 계산한다. 하지만 적절한 λ값을 정해야 하는 문제와 몬테카를로 방법과 같이 에피소드가 종료되어야 누적 보상을 계산할 수 있다는 문제가 있다.

본 논문에서는 n-스텝 시간차 학습에서 최적의 n을 선택해야 하는 문제를 해결하기 위해, $\Omega-return$이라는 새로운 n-스텝 업데이트 타겟을 제안한다. $\Omega-return$은 모든 n-스텝 누적 보상의 최댓값과 평균으로 구성된다. 또한, $\Omega-return$을 적용한 n-스텝 Q-learning 알고리즘과 DQN 알고리즘을 제안하고 다양한 환경에서 실험을 진행하여 제안하는 방식의 성능을 검증한다.

본 논문의 2장에서는 강화 학습의 기본 개념과 실험에서 사

※ 이 논문은 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2018R1A6A1A03025526 및 No. NRF-2020R1I1A3065610).
† 준 회 원 : 한국기술교육대학교 컴퓨터공학과 미래융합공학전공 석사
†† 준 회 원 : 한국기술교육대학교 컴퓨터공학과 미래융합공학전공 박사과정
††† 종신회원 : 한국기술교육대학교 컴퓨터공학과 교수
Manuscript Received : December 10, 2020
Accepted : February 20, 2021
* Corresponding Author : Youn-Hee Han(yhhan@koreatech.ac.kr)

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/ licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

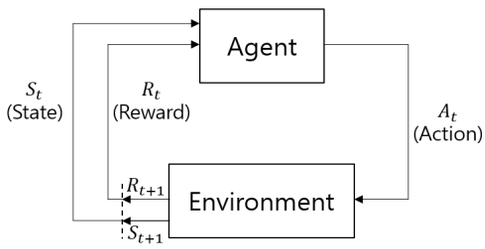

Fig. 1. Environment and Agent Interaction

용된 강화 학습의 여러 알고리즘을 설명하고, 간단한 그리드 월드 환경을 구성하여 실험을 통해 기존 n-스텝 시간차 학습 알고리즘의 문제점을 살펴보며 연구 동기를 밝힌다. 3장에서는 기존 n-스텝 시간차 학습 알고리즘의 문제점을 개선한 알고리즘을 제안한다. 4장은 먼저 실험 환경을 소개하고 기존 n-스텝 시간차 학습 알고리즘과 본 논문에서 제안하는 알고리즘을 비교하는 실험을 진행한다. 마지막으로 5장에서 결론을 제시한다.

2. 관련 연구

2.1 강화 학습

강화 학습(Reinforcement learning)은 머신 러닝의 한 분야로, 에이전트가 환경과 상호작용하며 시행착오를 거쳐 원하는 목표를 달성하기 위해 행동하는 법을 배우는 것이다. 강화 학습의 구성 요소로는 환경(Environment), 에이전트(Agent), 그리고 환경과 에이전트의 상호작용으로 얻는 상태(State), 행동(Action), 보상(Reward) 등 5가지가 있다.

Fig. 1은 환경과 에이전트의 상호작용 흐름을 보여준다. 타임 스텝 t에서, 에이전트는 상태 S_t에서 정책(Policy) π에 의해 행동 A_t를 선택하고 다음 상태 S_{t+1}로 전환되며 보상 R_{t+1}을 받는다. 여기서 정책 π는 각 상태에서 어떤 행동을 취할 것인가를 나타내는 확률밀도함수로 정의된다. 다음 상태 S_{t+1}와 보상 R_{t+1}은 에이전트에게 전달되어 이 과정을 다시 반복한다. 이 순환 과정은 에피소드가 종료될 때까지 계속되며, 강화 학습의 최종 목표는 환경으로부터 받는 누적 보상을 최대화하는 최적 정책(Optimal Policy) π^*를 찾는 것이다.

타임 스텝 t 이후부터 받은 누적 보상 G_t는 Equation (1)과 같으며 완전 누적 보상(Complete Return)이라고도 한다. $\gamma \in [0,1]$는 감가율(Discount Factor)이고 T는 에피소드 종료 시점의 타임 스텝을 나타낸다. 감가율 γ는 최근에 받은 보상에 더 큰 가중치를 주고, $T \to \infty$일 때 누적 보상 G_t의 값이 수렴하도록 돕는 역할을 한다.

$$\begin{aligned} G_t &= R_{t+1} + \gamma R_{t+2} + \gamma R_{t+3} + \cdots + \gamma^{T-t-1} R_T \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned} \tag{1}$$

에이전트가 t시점에 어떤 상태 s에서 시작해 정책 π를 따라 행동하며 받는 누적 보상 G_t의 기댓값을 상태 가치 함수(State-Value Function) $V_\pi(s)$라고 하며 다음과 같이 정의한다.

$$V_\pi(s) = E_\pi[G_t | S_t = s] \tag{2}$$

그리고, 에이전트가 t시점에 어떤 상태 s에서 행동 a를 선택하고 그 후 정책 π를 따라 행동하며 받는 누적 보상 G_t의 기댓값을 행동 가치 함수(Action-Value Function) $Q_\pi(s,a)$라고 하며 다음과 같이 정의한다.

$$Q_\pi(s,a) = E_\pi[G_t | S_t = s, A_t = a] \tag{3}$$

강화 학습 문제를 해결하는 방법으로는 크게 가치 기반(Value-based) 강화 학습과 정책 기반(Policy-based) 강화 학습이 있다. 본 논문에서는 가치 기반 강화 학습을 다룬다. 가치 기반 강화 학습에서 최적 정책 π^*를 구하는 방법은 최적의 행동 가치 함수 Q^*를 추정한 후, 각 상태 s에서 행동 가치 값이 최대인 행동을 선택하는 것이다. 이는 다음과 같이 표현된다.

$$\pi^*(s) = \arg\max_a Q^*(s,a) \tag{4}$$

에이전트가 학습을 진행할 때 탐욕적(Greedy) 행동 즉, $\arg\max_a Q(s,a)$로만 행동을 선택한다면 에이전트가 잘못된 방향으로 학습하여 국소 최적해(Local Optimum)에 빠질 가능성이 크다. 이 문제를 해결하는 방법으로 ε-그리디(ε-greedy) 정책이 있다. 에이전트가 ε의 확률로 랜덤하게 행동하거나 $1-\varepsilon$의 확률로 $\arg\max_a Q(s,a)$인 행동을 하는 것이다. ε-그리디 정책은 에이전트가 지속적으로 탐험하여 최적 정책을 찾도록 돕는다. ε값은 에이전트의 학습이 진행될수록 점차 낮아져 탐험 빈도를 떨어뜨리는 것이 일반적이다.

2.2 n-스텝 시간차 학습

최적의 행동 가치 함수 Q^*를 추정하는 방법으로는 몬테카를로 방법(Monte Carlo Method)과 시간차 학습(Temporal-Difference Learning)이 있다[1]. 몬테카를로 방법과 1-스텝 시간차 학습은 행동 가치 함수 Q를 업데이트할 때 타겟 값을 설정하는 측면에서 차이가 있다.

몬테카를로 방법은 타임 스텝 t부터 에피소드의 종료 시점 T까지의 누적 보상 G_t를 Q의 업데이트 타겟 값으로 사용한다. 이는 Equation (1)과 같으며 에피소드의 종료 시점까지를 나타내는 의미로 완전 누적 보상이라고도 한다.

반면, 1-스텝 시간차 학습에서 Q의 업데이트 타겟은 다음과 같은 수식을 이용해 계산하고 $G_{t:t+1}$로 표기한다. 이는 1-스텝 누적 보상(1-step return)이라 한다.

$$G_{t:t+1} = R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) \tag{5}$$

1-스텝 누적 보상은 에이전트가 타임 스텝 t에서 행동을 수행하고 다음 상태 S_{t+1}로 전환되며 받은 보상 R_{t+1}과 그

이후에 받을 보상에 대해서는 행동 가치 함수 Q로 예측한 값을 사용해 계산한다. 따라서 1-스텝 시간차 학습은 타임 스텝마다 Q를 업데이트할 수 있지만, 몬테카를로 방법은 타겟을 구하기 위해 에피소드의 종료 시점까지 가야 한다는 차이점이 있다. 이러한 이유로 에피소드가 종료되지 않는 환경이나 에피소드의 길이가 긴 경우에는 몬테카를로 방법이 적합하지 않다.

1-스텝 누적 보상을 타겟으로 하는 Q의 업데이트 수식은 다음과 같으며, α는 학습률(Learning Rate)을 나타낸다.

$$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha[G_{t:t+1} - Q_t(S_t, A_t)] \tag{6}$$

또한, 2-스텝 시간차 학습에서 Q의 업데이트 타겟은 다음과 같다.

$$G_{t:t+2} = R_{t+1} + \gamma R_{t+2} + \gamma^2 Q_{t+1}(S_{t+2}, A_{t+2}) \tag{7}$$

n-스텝 시간차 학습은 몬테카를로 방법과 1-스텝 시간차 학습을 결합한 것으로 n의 값을 적절하게 선택할 경우 강화 학습의 성능을 높일 수 있다[14]. $n \geq 1, 0 \leq t < T-n$에 대하여 n-스텝 누적 보상(n-step return)은 다음과 같이 정의된다.

$$\begin{aligned} G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} \\ + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}) \end{aligned} \tag{8}$$

n-스텝 누적 보상은 n-스텝 이후의 누적 보상에 대해서는 행동 가치 함수 Q를 이용하여 값을 추정하기 때문에 완전 누적 보상의 근삿값으로 볼 수 있다. $t+n > T$ 일 경우, T를 초과하는 모든 항은 0으로 간주하며, n-스텝 누적 보상은 완전 누적 보상과 같다.

n-스텝 시간차 학습에서 Q의 업데이트 수식은 다음과 같다.

$$Q_{t+n}(S_t, A_t) = Q_{t+n-1}(S_t, A_t) + \alpha[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)] \tag{9}$$

한편, TD(n) 알고리즘은 하이퍼-파라미터 n를 기반으로 모든 n-스텝 누적 보상에 가중치를 주어 평균을 계산하고 학습의 업데이트 타겟으로 사용한다[1]. 이를 λ-누적 보상(n-return)이라 하고 다음과 같이 정의한다.

$$G_t^{\lambda} = (1-\lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_{t:t+n} + \lambda^{T-t-1} G_t \tag{10}$$

λ-누적 보상은 에피소드의 종료 시점인 T에 도달해야 계산할 수 있다. 만약 $\lambda = 0$이면, λ-누적 보상은 1-스텝 누적 보상인 $G_{t:t+1}$와 같다. 그리고 $\lambda = 1$이면, λ-누적 보상은 몬테카를로 방법과 같은 완전 누적 보상인 G_t와 같다. $0 < \lambda < 1$일 때, 1-스텝 누적 보상의 가중치는 $1-\lambda$이고, 2-스텝 누적 보상의 가중치는 $(1-\lambda)\lambda$가 된다. 다른 누적 보상의 가중치는 동일한 방법으로 계산되며 모든 가중치의 합은 1이 된다.

n-스텝 시간차 학습과 마찬가지로 TD(λ) 알고리즘 또한 적절한 λ값을 정해야 하는 문제가 있다[15].

2.3 Q-learning

강화 학습에서 정책을 업데이트하는 방법으로는 온-폴리시(on-policy) 알고리즘과 오프-폴리시(off-policy) 알고리즘이 있다. 온-폴리시 알고리즘은 에이전트가 행동할 때 사용하는 정책(행동 정책)과 학습하기 위해 타겟을 계산할 때 사용하는 정책(타겟 정책)이 같은 것을 말하고, 오프-폴리시 알고리즘은 행동 정책과 타겟 정책이 다른 것을 말한다. 오프-폴리시 알고리즘의 장점은 행동 정책과 타겟 정책을 분리했기 때문에 에이전트의 탐험을 위한 행동 정책으로는 ε-그리디 정책을 사용하면서, 학습할 때는 그리디 정책을 사용하여 최적 정책을 배울 수 있다.

Q-learning은 오프-폴리시 강화 학습 알고리즘으로, Q-learning의 1-스텝 누적 보상 수식은 다음과 같다[8,17].

$$G_{t:t+1} = R_{t+1} + \gamma \max_a Q_t(S_{t+1}, a) \tag{11}$$

n-스텝 시간차 학습을 Q-learning에 적용하면, $n \geq 1, 0 \leq t < T-n$에 대하여 n-스텝 누적 보상은 다음과 같이 정의된다[21,23].

$$\begin{aligned} G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} \\ + \gamma^n \max_a Q_{t+n-1}(S_{t+n}, a) \end{aligned} \tag{12}$$

2.4 n-스텝 시간차 학습의 파라미터 민감성

이번 장에서는 몇 가지 실험 결과를 통해 연구 동기를 밝힌다. 실험은 Fig. 2에 나타낸 네 가지의 5×5 그리드월드 환경에서 진행하였다. 각각의 환경은 트랩이나 구덩이의 개수, 보상의 크기가 다르다.

에이전트는 동, 서, 남, 북 4가지의 행동을 취할 수 있고 벽과 맞닿아있는 상태에서 벽 쪽으로 가는 행동을 취할 때는 상태변화가 없다. 매 타임 스텝마다 -0.1의 보상을 받고, 'S'에서 출발해 목표 'G'에 도달하면 1.0의 보상을 받으며 에피소드는 종료된다. Fig. 2의 (b) 환경은 트랩 'T'에 도달하면 -3.0, (c) 환경은 트랩 'T'에 도달하면 -6.0의 보상을 받는다. (d) 환경에서는 구덩이 'H'에 도달하면 -3.0의 보상을 받으며 에피소드가 종료된다.

Fig. 3은 네 가지의 5×5 그리드월드 환경(Fig. 2)에서의 다양한 n과 학습률 α에 대한 n-스텝 Q-learning의 실험 결과이다. Fig. 3 (a), (b), (c), (d)의 실험 환경은 Fig. 2의 (a), (b), (c), (d) 각각에 대응된다. 에이전트의 행동을 추론할 때는 ε-그리디 정책을 사용하였고 ε값은 매 타임 스텝마다 점진적으로 감소시켰다. 200 에피소드가 끝나면 학습을 종료하였고, 이 과정을 100번 진행하였다. 그래프의 Return은 마지막 에피소드에 받은 누적 보상을 나타내며, RMS Error는 다이나믹 프로그래밍(Dynamic Programming)으로 구한 행동 가치 Q의 true value와 강화 학습 n-스텝 Q-learning 알고리즘으로 구한 행동 가치 Q의 평균 제곱근 편차(RMSE)를 나타낸다. α는 학습률을 의미한다. 그래프의 결과는 100번 진행하며 얻은 값들의 평균과 표준편차를 표시한 것이다.

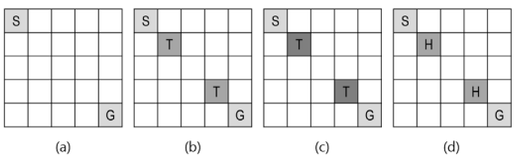

Fig. 2. Four Types of 5×5 Gridworld

Table 1. The Best and Worst n for four Types of Gridworld

	$\alpha = 0.1$		$\alpha = 0.3$	
Experiment	best n	worst n	best n	worst n
(a)	3	1	3	6
(b)	6	3	1	3
(c)	1	3	1	6
(d)	6	3	1	3

Fig. 3에 나타난 결과와 같이, RMSE는 n과 α에 비례하여 증가하는 반면, Return은 n과 α가 변경됨에 따라 결과가 크게 달라진다. n과 α가 작을 때는 에이전트의 정책이 느리게 수렴하면서 많은 상태를 방문하며 대부분의 행동 가치 Q값을 업데이트하지만, n과 α가 클 때는 최적 정책이 아닌 잘못된 정책으로 빠르게 수렴하면서 일부의 Q값만 업데이트되어 RMSE가 커지는 것으로 생각된다.

Table 1은 $\alpha = 0.1, 0.2, \cdots, 1.0$ 중 2개의 값 $\alpha = 0.1$과 $\alpha = 0.3$에 대한 실험 결과이며, Return이 가장 높을 때의 n과 가장 낮을 때의 n을 나타낸다. Table 1에 나타난 결과와 같이 n은 실험 환경과 학습률 α의 변화에 민감하여 최적과 최악의 n을 찾는 것이 어렵다. 따라서 n-스텝 시간차 학습의 성능은 파라미터에 민감하다는 것을 알 수 있다[16].

2.5 n-스텝 시간차 학습의 편향-분산 트레이드오프

Equation (12)의 Q-learning n-스텝 누적 보상은 다음의 2가지로 분리하여 볼 수 있다.

$$\sum_{k=1}^{n} \gamma^{k-1} R_{t+k} \tag{13}$$

$$\gamma^{n} \max_{a} Q_{t+n-1}(S_{t+n}, a) \tag{14}$$

Equation (13)은 에이전트가 환경과 상호작용하며 받은 실제 보상을 의미하며, Equation (14)는 에이전트가 받은 보상이 아닌 행동 가치 함수 Q를 사용하여 누적 보상을 추정한 값이다.

n의 값이 커지면 n-스텝 누적 보상에서 Equation (13)의 비중이 높아지며 Equation (14)의 비중은 낮아진다. 에이전트는 환경을 탐험하기 위해 확률적으로 행동하기 때문에 같은 상태에 존재하더라도 다른 행동을 취하며 다른 보상을 받을 수 있다. 따라서 n이 커지면 확률적으로 행동하며 받은 보상의 비중이 커지므로 n-스텝 누적 보상의 분산이 커지게 된다.

이와 반대로 n의 값이 작아지면 n-스텝 누적 보상에서 Equation (13)의 비중은 낮아지며 Equation (14)의 비중은 높아진다. n의 값이 1인 경우를 예로 들면, 1-스텝 누적 보상을 계산할 때 에이전트가 받은 실제 보상은 단 하나만 사용되며 이후의 누적 보상에 대해서는 S_{t+1}에서의 행동 가치 함수 Q로 계산하기 때문에

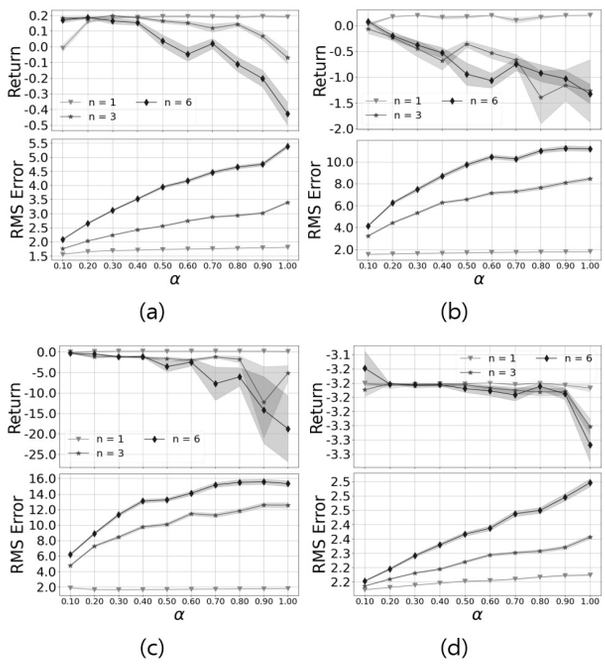

Fig. 3. The Results of n-step Q-learning

Equation (14)의 비중이 높아지는 것이다. 행동 가치 함수 Q는 학습 중인 에이전트가 받을 누적 보상을 추정한 값이기 때문에 실제 값과는 차이가 있고 이로 인해 편향이 발생하게 된다.

따라서 n-스텝 시간차 학습 알고리즘은 n이 커지면 n-스텝 누적 보상의 분산이 커지고 n이 작아지면 편향이 발생하는 편향-분산 트레이드오프 문제가 있다[21,22].

3. Max-Mean n-스텝 시간차 학습

2장에서 n-스텝 시간차 학습이 환경과 하이퍼-파라미터에 민감하여 적절한 n을 선택하는 것이 어렵다는 문제를 밝혔다. 본 장에서는 기존 n-스텝 시간차 학습 알고리즘에서 적절한 n을 선택해야 하는 문제를 해결하기 위해, $1 \le k \le n$에 대한 모든 k-스텝 누적 보상의 최댓값과 평균으로 구성된 $\Omega - return$이라는 새로운 n-스텝 업데이트 타겟을 제안한다.

3.1 멀티-스텝 누적 보상의 최댓값과 평균

본 논문에서 제안하는 방법은 n-스텝 누적 보상에 기초하여 가장 좋은 업데이트 타겟을 결정하는 것에서 시작된다. Q-learning 알고리즘의 학습 타겟인 1-스텝 누적 보상의 Equation (11)을 보면 $\max_{a} Q_t(S_{t+1}, a)$가 사용된다. Q의 max를 취하는 것은 실제 Q값보다 과대평가되어 편향이 발생해 문제를 야기할 수 있지만, Qingfeng Lan은 Q의 과대평가가 항상 해로운 것은 아니며 학습 성능을 향상시킬 수도 있다고 하였다[24]. 강화 학습 환경에서는 에이전트가 탐험을 통해 의미 있는 경험을 빠르게 배우는 것이 중요하기 때문에, 만약 높은 보상을 받을 수 있는 상태-행동(State-Action)에 대한 Q값이 과대평가되어 있다면 에이전트가 해당 영역을 탐

험하도록 장려하여 학습에 이점을 줄 수 있기 때문이다.

이러한 이유로, $1 \le k \le n$에 대한 모든 k-스텝 누적 보상 중에서 최댓값은 편향이 크지 않은 학습 초기에 좋은 업데이트 타겟으로 활용될 수 있다. 이 최댓값을 G_{max}라 하고 다음과 같이 정의한다.

$$G_{max} = \max_k E[G_{t:t+k}],\ 1 \le k \le n \tag{15}$$

만약 모든 상태와 행동에 대한 최적의 행동 가치 함수 Q^*를 찾았다면, $1 \le k \le n$에 대한 모든 k-스텝 누적 보상 $G_{t:t+k}$는 같은 값을 가진다.

Theorem 1.
Q-learning에서 만약 $Q \approx Q^*$이면,
$E[G_{t:t+1}] \approx E[G_{t:t+2}] \approx \cdots \approx E[G_{t:t+n}]$.

proof. 벨만 최적 방정식에 의해, 최적의 행동 가치 함수 Q^*는 다음과 같이 정의할 수 있다[20].

$$\begin{aligned}
&Q^*(S_t, A_t) = E[R_{t+1} + \gamma \max_a Q^*(S_{t+1}, a)] \\
&= E[R_{t+1} + \gamma R_{t+2} + \gamma^2 \max_a Q^*(S_{t+2}, a)] \\
&= \cdots \\
&= E[R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \max_a Q^*(S_{t+n}, a)]
\end{aligned}$$

그리고 Equation (12)에 따라서, $G_{t:t+1}, G_{t:t+2}, \cdots, G_{t:t+n}$은 다음과 같이 쓸 수 있다.

$$\begin{aligned}
&E[G_{t:t+1}] = E[R_{t+1} + \gamma \max_a Q(S_{t+1}, a)] \\
&E[G_{t:t+2}] = E[R_{t+1} + \gamma R_{t+2} + \gamma^2 \max_a Q(S_{t+2}, a)] \\
&\cdots \\
&E[G_{t:t+n}] = E[R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} \\
&\qquad + \gamma^n \max_a Q(S_{t+n}, a)]
\end{aligned}$$

만약 $Q \approx Q^*$이면, 모든 S_t에 대해 $\max_a Q(S_t, a) \approx \max_a Q^*(S_t, a)$가 성립하며 행동 a는 최적 정책에 의해 $\arg\max_a Q^*(S_t, a)$로 정해지므로 다음과 같이 쓸 수 있다.

$$\begin{aligned}
&E[G_{t:t+1}] \approx E[R_{t+1} + \gamma \max_a Q^*(S_{t+1}, a)] \\
&E[G_{t:t+2}] \approx E[R_{t+1} + \gamma R_{t+2} + \gamma^2 \max_a Q^*(S_{t+2}, a)] \\
&\cdots \\
&E[G_{t:t+n}] \approx E[R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} \\
&\qquad + \gamma^n \max_a Q^*(S_{t+n}, a)]
\end{aligned}$$

∴벨만 최적 방정식에 의해,
$E[G_{t:t+1}] \approx E[G_{t:t+2}] \approx \cdots \approx E[G_{t:t+n}]$.
■

Corollary 1.1.
Q-learning에서 만약 $Q \approx Q^*$이면,

$$E[G_{t:t+1}] \approx E[G_{t:t+2}] \approx \cdots \approx E[G_{t:t+n}] \approx Q(S_t, A_t).$$

Corollary 1.1에 따라, 아래와 같은 수식을 얻을 수 있다.

$$\frac{1}{n}\left[\sum_{k=1}^{n} E[G_{t:t+k}]\right] - Q(S_t, A_t) \approx 0 \tag{16}$$

Equation (16)으로부터, 학습이 충분히 이루어졌을 때 $1 \le k \le n$에 대한 모든 k-스텝 누적 보상의 평균이 Q-learning의 좋은 학습 타겟으로 활용될 수 있음을 알 수 있다. 이 평균을 G_{mean}이라 하고 다음과 같이 정의한다.

$$G_{mean} = \frac{1}{n}\left[\sum_{k=1}^{n} E[G_{t:t+k}]\right],\ 1 \le k \le n \tag{17}$$

3.2 $\Omega-return$

3.1에서 살펴본 바와 같이 G_{max}는 학습 초기에 좋은 업데이트 타겟으로 활용될 수 있지만, 학습이 진행되며 과대평가 편향 문제가 발생할 위험이 있다[11]. 반면, G_{mean}은 학습이 충분히 이루어졌을 때 좋은 업데이트 타겟으로 쓰일 수 있다. 따라서, G_{max}와 G_{mean}을 결합한 새로운 학습 타겟인 $\Omega-return$을 제안하며, 다음과 같이 정의한다.

$$\Omega-return = \beta G_{max} + (1-\beta) G_{mean},\ 0 \le \beta \le 1 \tag{18}$$

Equation (18)에서, β는 G_{max}와 G_{mean}의 비율을 결정하는 파라미터이다. 본 논문은 파라미터 β를 Equation (19)로 대체하였다.

$$\beta = \frac{\max_k |G_{t:t+k}| - \min_k |G_{t:t+k}|}{\max_k |G_{t:t+k}|},\ 1 \le k \le n \tag{19}$$

학습 초기에는 $\max_k |G_{t:t+k}|$와 $\min_k |G_{t:t+k}|$의 값이 차이가 나기 때문에 $\Omega-return$에서 G_{max}의 비율이 높지만, 학습이 진행되며 $\max_k |G_{t:t+k}|$와 $\min_k |G_{t:t+k}|$의 차이가 줄어들면 G_{max}의 비율은 낮아지고 G_{mean}의 비율이 높아지게 된다.

3.3 $\Omega-return$을 적용한 n-스텝 Q-learning 알고리즘

본 논문에서 제안하는 새로운 학습 타겟인 $\Omega-return$을 적용한 n-스텝 Q-learning 알고리즘의 의사코드(Pseudo-code)는 Algorithm 1과 같다. ψ는 분모가 0이 되는 것을 방지하기 위한 매우 작은 양의 상수를 나타낸다. 12번째 줄부터 18번째 줄까지는 에피소드가 종료될 때까지 에이전트가 상태 S_t에서 행동 A_t를 선택하고 보상 R_{t+1}을 받으며 다음 상태 S_{t+1}로 넘어가는 것을 나타낸다. 22번째 줄부터는 Q를 업데이트하는 과정을 보여준다. $1 \le k \le n$에 대한 모든 k-스텝 누적 보상을 계산한 뒤, 이의 최댓값과 평균을 이용하여 학습 타겟인 $\Omega-return$을 계산한다. 학습 타겟과 현재 Q와의 차이를 이용하여 학습률 α만큼 Q를 업데이트한다.

Algorithm 1: The proposed n-step Q-learning with Ω-return

Algorithm parameters: a positive integer n, step size $\alpha \in (0, 1]$,
small $\epsilon > 0$, small $\psi > 0$
Initialize $Q(s, a)$ arbitrarily, for all $s \in S, a \in A$
Initialize π to be ϵ-greedy w.r.t. Q

foreach *episode* **do**
  Initialize and store $S_0 \neq$ terminal
  Select and store an action $A_0 \sim \pi(\cdot|S_0)$
  $T \leftarrow \infty$
  loop for *step* $t = 0, 1, 2, \cdots$ **do**
    if $t < T$ **then**
      Take action A_t
      Observe and store the next reward as R_{t+1}
      and the next state as S_{t+1}
      if S_{t+1} is terminal, **then**
        $T \leftarrow t + 1$
      else
        select and store an action $A_{t+1} \sim \pi(\cdot|S_{t+1})$
    $\tau \leftarrow t - n + 1$
    (τ is the time whose estimate is being updated)
    if $\tau \geq 0$ **then**
      $\hat{G} \leftarrow [\,]$
      for $i = \tau + 1, \cdots, min(\tau + n, T)$ **do**
        $G \leftarrow \sum_{j=\tau+1}^{i} \gamma^{j-\tau-1} R_j$
        if $i < T$, **then**
          $G \leftarrow G + \gamma^{i-\tau} max_a Q(S_i, a)$
        Append G to $\hat{G}$
      $\beta = \frac{max(|\hat{G}|) - min(|\hat{G}|)}{max(|\hat{G}|) + \psi}$
      $\delta_\tau = \beta max(\hat{G}) + (1 - \beta) mean(\hat{G}) - Q(S_\tau, A_\tau)$
      $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha\delta_\tau$
      Update $\pi(\cdot|S_\tau)$ to be ϵ-greedy w.r.t. Q
  until $\tau = T - 1$

4. 실 험

4.1 학습 정도에 따른 β값 변화 실험

Fig. 4는 두 가지의 5×5 그리드월드 환경(Fig. 2)에서 학습 정도에 따른 $\Omega-return$에서 G_{max}와 G_{mean}의 비율을 결정하는 파라미터 β값의 변화에 대한 실험 결과이다. Fig. 4 (a), (b)의 실험 환경은 Fig. 2의 (a), (b) 각각에 대응된다. 200 에피소드가 끝나면 학습을 종료하였고, 이 과정을 100번 진행하였다. 그래프에 나타낸 β값은 해당 에피소드에서 매 타임 스텝마다 계산한 n값들의 평균을 의미한다. 실험 결과를 통해 학습 초반에 높았던 n값이 학습이 진행되면서 점점 낮아지는 것을 볼 수 있다.

4.2 $\Omega-return$의 성능 평가

본 논문의 실험 목적은 n-스텝 시간차 학습에서 적절한 n을 선택해야 하는 문제를 해결하기 위해 제안하는 새로운 학습 타겟인 $\Omega-return$의 성능을 평가하는 것이다. 따라서 실험은 기존 n-스텝 시간차 학습 알고리즘과의 비교 평가를 진행한다.

본 논문의 실험은 Nature DQN[9, 10]과 이의 확장 알고리즘인 Double DQN[12,13], Dueling DQN[18], Prioritized Experience Replay[19] 알고리즘을 통합한 기법 위에 기존 n-스텝 시간차 학습과 제안하는 방법을 구현하여 비교 실험을 진행하였다[16].

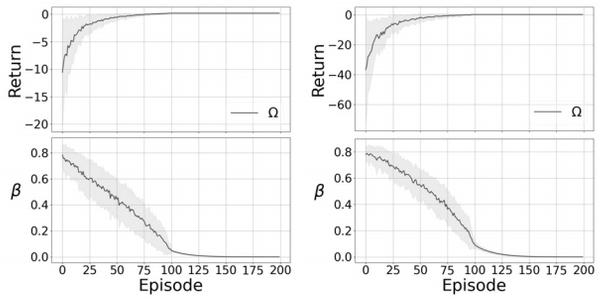

Fig. 4. Variation of β According to Learning Degree

기존 n-스텝 시간차 학습 알고리즘에서 ={1, 3, 6} 3가지에 대해서 실험을 진행하였다. 본 논문에서 제안하는 $\Omega-return$의 $1 \leq k \leq n$ 에서의 n은 기존 n-스텝 시간차 학습 알고리즘에서 6까지 실험하였으므로 동일하게 6으로 설정하였다.

실험은 OpenAI Gym의 Atari 게임 중 'Pong', 'Enduro', 'Seaquest', 'Freeway' 4개 환경에서 진행하였다[25]. 실험 결과는 Fig. 5와 같으며 데이터는 100개에 대한 이동평균(Moving Average)을 나타낸다. 실험은 1-스텝, 3-스텝, 6-스텝, Ω 모두 각각 3번씩 측정하였으며, 3번 측정한 것의 평균과 표준편차를 그래프에 나타내었다. 그래프의 가로축은 강화 학습 환경에서의 타임 스텝 t를 나타내며, 세로축은 에이전트가 한 에피소드의 시작부터 끝날 때까지 받은 누적 보상을 의미한다. Return은 매 1000타임 스텝마다 측정하였으며 직전 100개 에피소드에서 받은 누적 보상의 평균을 의미한다. 따라서 그래프가 가로축 대비 세로축이 빠르게 올라가는 것은 에이전트가 학습을 빠른 시간 내에 배우고 있다는 것을 의미하기 때문에 좋은 성능을 낸 것이라고 볼 수 있다.

Fig. 5의 실험 결과로부터 본 논문에서 제안하는 알고리즘이 기존 n-스텝 시간차 학습 알고리즘보다 전반적인 학습 성능이 우수하다는 것을 알 수 있다. 학습 초반 모든 실험 환경에서 제안하는 알고리즘의 성능이 우수한 것을 보아 학습 초기에 G_{max}가 좋은 타겟이라는 것을 알 수 있고, 학습이 진행되며 성능이 떨어지지 않는 것을 보아 G_{mean}이 과대평가 문제를 방지하였다고 볼 수 있다. 또한 Fig. 5-2나 Fig. 5-4의 결과와 같이 기존 n-스텝 시간차 학습 알고리즘의 성능이 1-스텝 시간차 학습보다 떨어지는 경우에도 본 논문에서 제안하는 알고리즘의 성능이 좋다는 것을 알 수 있다. 따라서 $\Omega-return$이 n-스텝 시간차 학습에서 n을 선택해야 하는 문제를 해결하는 새로운 학습 타겟으로 적절하다고 생각한다.

5. 결 론

n-스텝 시간차 학습은 적절한 n을 선택할 경우 1-스텝 시간차 학습보다 성능이 좋은 알고리즘으로 알려져 있다. 하지만, n-스텝 시간차 학습은 강화 학습 환경의 변화에 따라 최적의 n이 바뀌는 파라미터 민감성의 문제가 있다. 또한, n-스텝 시간차 학습의 타겟 값인 n-스텝 누적 보상은 n의 값이 커지면 분산이 높아지고, n의 값이 작아지면 편향이 커지는 편향-분산

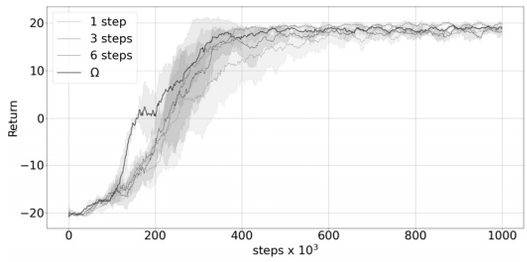

Fig. 5-1. OpenAI Gym Atari-Pong

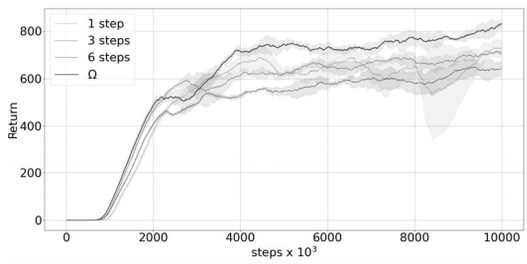

Fig. 5-2. OpenAI Gym Atari-Enduro

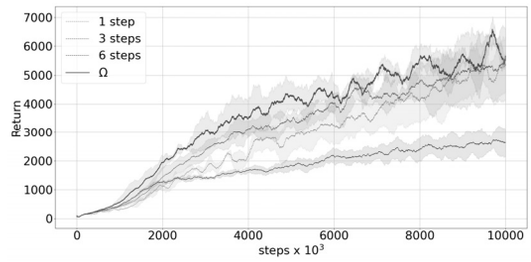

Fig. 5-3. OpenAI Gym Atari-Seaquest Environment

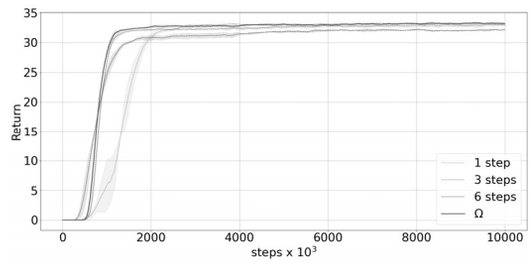

Fig. 5-4. OpenAI Gym Atari-Freeway Environment

트레이드오프 문제가 있다. 따라서 n-스텝 시간차 학습 알고리즘은 하이퍼-파라미터 n을 선택하기 어려운 문제가 있다.

본 논문에서는 $1 \le k \le n$에 대한 모든 k-스텝 누적 보상의 최댓값과 평균으로 구성된 $\Omega-return$이라는 새로운 n-스텝 업데이트 타겟을 제안하고, $\Omega-return$을 적용한 n-스텝 Q-learning 알고리즘을 제안한다.

실험은 Nature DQN과 이의 확장 알고리즘인 Double DQN, Dueling DQN, Prioritized replay buffer 알고리즘들을 통합한 기법 위에서 기존 n-스텝 시간차 학습과 제안하는 알고리즘의 성능 비교 평가를 진행하였다. OpenAI Gym의 Atari 게임 환경에서의 실험 결과 본 논문에서 제안하는 알고리즘이 기존 n-스텝 시간차 학습 알고리즘보다 성능이 우수하다는 것을 입증하였다.

본 논문에서 제안하는 $\Omega-return$은 $1 \le k \le n$에 대한 모든 k-스텝 누적 보상의 최댓값과 평균으로 구성되기 때문에 범위를 지정하는 n이 하이퍼-파라미터라고 볼 수 있다. 하지만, 기존 n-스텝 시간차 학습에서도 최적의 n을 찾기 위해서 1-스텝부터 몇 스텝까지 실험을 할지 결정해야 하는 문제는 동일하며, 지정된 범위 내의 모든 값을 전부 실험하여 적절한 n값을 찾아야 하는 문제가 있다. 본 논문에서 제안하는 알고리즘은 기존 n-스텝 시간차 학습 알고리즘에서 n을 선택하는 문제를 해결하였고 n-스텝 시간차 학습이 1-스텝 시간차 학습보다 낮은 성능을 보이는 환경에서도 제안하는 알고리즘의 성능이 우수한 것을 보였다.

References

[1] R. S. Sutton, "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, Vol.3, No.1, pp.9-44, 1988.

[2] H. Seijen and R. Sutton, "True Online TD(λ)," In *Proceedings of the 31st International Conference on Machine Learning*, vol. 32 of Proceedings of Machine Learning Research, pp.692-700. PMLR, Bejing, China, 2014.

[3] K. D. Asis, J. Hernandez-Garcia, G. Holland, et al., "Multi-Step Reinforcement Learning: A Unifying Algorithm," In *Association for the Advancement of Artificial Intelligence*, 2018.

[4] S. L. Chen, H. Z. Wu, X. L. Han, and L. Xiao, "Multi-Step Truncated Q Learning Algorithm," *In 2005 International Conference on Machine Learning and Cybernetics*, Vol.1, pp.194-198. 2005.

[5] K. De Asis and R. Sutton, "Per-decision Multi-step Temporal Difference Learning with Control Variates," arXiv:1807.01830, 2018.

[6] A. R. Mahmood, H. Yu, and R. Sutton, "Multi-step Off-policy Learning without Importance Sampling Ratios," arXiv: 1702.03006, 2017.

[7] L. Yang, M. Shi, Q. Zheng, W. Meng, and G. Pan, "A Unified Approach for Multi-step Temporal-Difference Learning with Eligibility Traces in Reinforcement Learning," In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, IJCAI-18, pp.2984-2990, 2018.

[8] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, Vol.8, No.3, pp.279-292, 1992.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," Cite arXiv:1312.5602Comment: NIPS Deep Learning Workshop 2013.

[10] V. Mnih, et al., "Human-level Control through Deep Reinforcement Learning," *Nature*, Vol.518, pp.529-533, 2015.
[11] S. Thrun A. and Schwartz, "Issues in using function approximation for reinforcement learning," In *Proceedings of the Fourth Connectionist Models Summer School*, Erlbaum, 1993.
[12] H. van Hasselt, "Double Q-learning," *In Advances in Neural Information Processing Systems*, Vol.23, pp.2613-2621, 2010.
[13] H. van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-learning," In *Association for the Advancement of Artificial Intelligence*, 2016.
[14] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," The MIT Press, Second Edn., 2018.
[15] J. Peng and R. J. Williams, "Incremental Multi-Step Q-Learning," *Machine Learning*, Vol.22, No.1, pp.283-290, 1996.
[16] M. Hessel, et al., "Rainbow: Combining Improvements in Deep Reinforcement Learning," In *Association for the Advancement of Artificial Intelligence*, pp.3215-3222, AAAI Press, 2018.
[17] C. J. C. H. Watkins, "Learning from delayed rewards," (Doctoral dissertation, Cambridge University).
[18] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," In *Proceedings of The 33rd International Conference on Machine Learning*, in PMLR 48:1995-2003, 2016.
[19] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," 2015. Cite arXiv:1511.05952Comment: Published at ICLR 2016.
[20] S. J. Bradtke and M. O. Duff, "Reinforcement learning methods for continuous-time Markov decision problems," In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, MIT Press, Cambridge, MA, USA, pp.393-400, 1994.
[21] J. Hernandez-Garcia and R. Sutton, "Understanding Multi-Step Deep Reinforcement Learning: A Systematic Study of the DQN Target," 2019. Cite arXiv:1901.07510Comment: NIPS Deep Learning Workshop 2018.
[22] M. J. Kearns and S. P. Singh, "Bias-variance error bounds for temporal difference updates," *In Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pp.142-147. San Francisco, CA, USA, 2000.
[23] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. Hasselt, and D. Silver, "Distributed prioritized experience replay," 2018. Cite arXiv:1803. 00933Comment: Published at ICLR 2018.
[24] Q. Lan, Y. Pan, A. Fyshe, M. White, "Maxmin Q-learning: Controlling the Estimation Bias of Q-learning," 2020. Cite arXiv:2002.06487Comment: Published at ICLR 2020.
[25] OpenAI. OpenAI Gym Docs [Internet], https://gym.openai.com/docs/. Accessed: 2020-11-20.

황 규 영
https://orcid.org/0000-0002-5014-8826
e-mail : to6289@koreatech.ac.kr
2019년 한국기술교육대학교 컴퓨터공학부 (학사)
2021년 한국기술교육대학교 컴퓨터공학과 미래융합공학전공(석사)
2021년~현 재 (주)인투와이즈 연구원
관심분야 : 강화 학습, 딥러닝

김 주 봉
https://orcid.org/0000-0001-6406-3092
e-mail : rlawnqhd@koreatech.ac.kr
2017년 한국기술교육대학교 컴퓨터공학부 (학사)
2019년 한국기술교육대학교 컴퓨터공학과 (석사)
2019년 ~ 현 재 한국기술교육대학교 컴퓨터공학과 미래융합공학전공 박사과정
관심분야 : 강화 학습, 다중 에이전트 강화 학습, 딥러닝

허 주 성
https://orcid.org/0000-0002-2486-9515
e-mail : chill207@koreatech.ac.kr
2016년 한국기술교육대학교 컴퓨터공학부 (학사)
2019년 한국기술교육대학교 컴퓨터공학과 (석사)
2019년 ~ 현 재 한국기술교육대학교 컴퓨터공학과 미래융합공학전공 박사과정
관심분야 : 강화 학습, 딥러닝, 금융데이터 분석

한 연 희
https://orcid.org/0000-0002-5835-7972
e-mail : yhhan@koreatech.ac.kr
1998년 고려대학교 컴퓨터학과(석사)
2002년 고려대학교 컴퓨터학과(박사)
2002년 ~ 2006년 삼성종합기술원 전문연구원
2013년 ~ 2014년 SUNY at Albany, Department of Computer Science 방문교수
2006년 ~ 현 재 한국기술교육대학교 컴퓨터공학부 교수
관심분야 : 사물인터넷, 기계 학습, 강화 학습