안정적인 학습을 위한 멀티-스텝 DQN

황규영*, 김주봉*, 한연희*†
*한국기술교육대학교 컴퓨터공학과
{to6289, rlawnqhd, yhhan}@koreatech.ac.kr

Multi-Step DQN for Stable Learning

Gyu-Young Hwang*, Ju-Bong Kim*, Youn-Hee Han*
*Dept. of Computer Science and Engineering, Korea University of Technology and Education.

요 약

n-스텝 temporal-difference(TD)학습은 몬테카를로와 1-스텝 TD 학습을 통합한 것이다. n-스텝 TD 학습에서 최적의 n값은 환경과 하이퍼-파라미터에 따라 달라지기 때문에 적절한 n값을 정하는 것은 어렵다. 본 논문에서는 여러 n-스텝 누적 보상의 평균과 최대값으로 구성된 Ω-return 이라는 새로운 타겟을 제안하며, Ω-return 을 n-스텝 DQN 에 적용하여 OpenAI Gym 의 'MountainCar-v0' 환경에서 기존의 n-스텝 DQN 과의 성능 비교 실험을 진행한다.

Ⅰ. 서 론

n -스텝 temporal-difference(TD)학습 [1]은 n -스텝까지 관찰된 누적 보상(reward)과 n번째 스텝에서 부트스트래핑 값을 합하여 업데이트 타겟으로 사용한다. 적절한 n값을 선택하였다면 n-스텝 TD 학습은 1-스텝 TD 학습보다 좋은 성능을 낼 수 있다. 그러나 n-스텝 TD 학습은 n의 값이 커질수록 state-action value 의 분산이 높아져 환경에 맞는 적절한 n 값을 찾는 것이 어렵다 [2].

최근 연구 [3, 4]에서는 n-스텝 TD 학습의 변형을 제안했다. 하지만 학습의 가속화를 위한 최적의 n 값을 찾는 방법에 대한 논의는 없었다. 본 논문에서는 n-스텝 TD 학습을 하이퍼-파라미터 n과 환경에 대해 강건하게 만들기 위해, Ω-return 이라는 새로운 n-스텝 업데이트 타겟을 제안한다. 또한, Ω -return 을 적용한 n -스텝 DQN 알고리즘으로 실험을 진행하여 제안하는 방식의 성능을 검증한다.

Ⅱ. 배경

몬테카를로와 1-스텝 TD 학습은 state-action value 함수 Q를 업데이트할 때 타겟 값을 설정하는 측면에서 차이가 있다. 몬테카를로에서는 누적 보상 G_t를 타겟 값으로 사용한다. 이는 타임 스텝 t에서 종료 시점까지의 감가된 누적 보상을 나타내므로 완전 누적 보상이라고 한다.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1} \quad (1)$$

반면, 1-스텝 TD 학습의 업데이트 타겟은 1-스텝 누적 보상 $G_{t:t+1}$이다.

$$G_{t:t+1} = R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) \quad (2)$$

따라서, 1-스텝 TD 학습에서 Q의 업데이트 수식은 다음과 같다.

$$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha[G_{t:t+1} - Q_t(S_t, A_t)], \quad 0 \le t < T \quad (3)$$

α는 학습률을 나타낸다.

$n \ge 1$, $0 \le t < T - n$ 에 대하여 n-스텝 누적 보상은 다음과 같이 정의된다.

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}) \quad (4)$$

n-스텝 TD 학습에서 Q 업데이트 수식은 다음과 같다.

$$Q_{t+n}(S_t, A_t) = Q_{t+n-1}(S_t, A_t) + \alpha[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)], \quad 0 \le t < T \quad (5)$$

n -스텝 누적 보상은 n -스텝 이후의 누적 보상에 대해서는 state-action value 함수를 사용해 값을 추정하기 때문에 완전 누적 보상의 근사값으로 볼 수 있다. $t + n > T$ 일 경우, T를 초과하는 모든 항은 0 으로 간주되며, n-스텝 누적 보상은 완전 누적 보상과 같다. n -스텝 TD 학습은 몬테카를로와 1-스텝 TD 학습을 결합한 방법이다. 하지만 n-스텝 TD 학습에서는 n값을 휴리스틱하게 결정해야 하는 문제가 있다.

Ⅲ. 본론

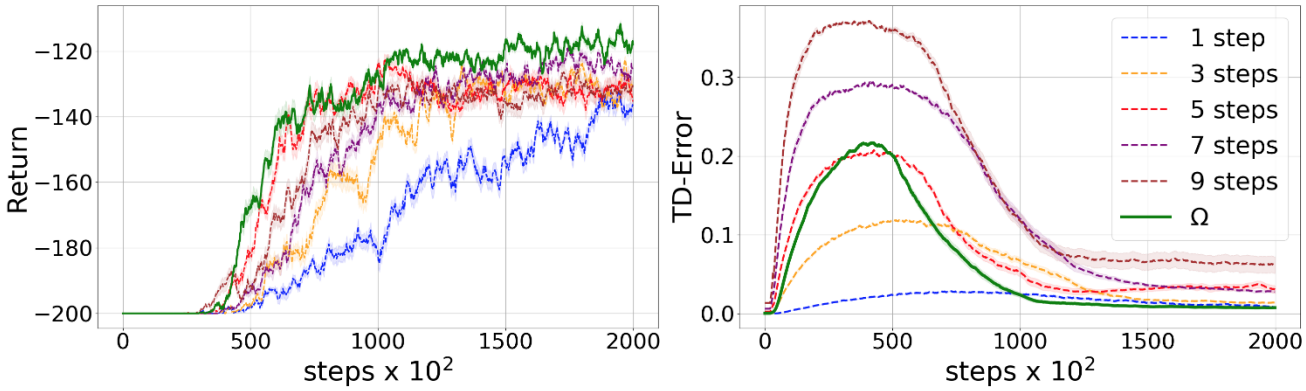
n-스텝 TD 학습이 하이퍼-파라미터와 환경의 변화에 대해 강건성을 가질 수 있도록, $1 \le k \le n$에 대한 모든 k -스텝 누적 보상의 평균과 최대값으로 구성된 Ω -return 이라는 새로운 n-스텝 업데이트 타겟을 제안한다.

1. 누적 보상($G_{t:t+k}$)의 평균과 최대값

제안하는 방법은 n -스텝 누적 보상에 기초한 가장 좋은 업데이트 타겟을 결정하는 것에서 시작된다. 만약 모든 상태에 대한 최적의 행동 가치를 구했다면, $1 \le k \le n$에 대한 모든 $G_{t:t+k}$는 같은 값을 가질 것이다. 상태 S_t에서 행동 A_t가 최적 정책을 따른다면 다음과 같이 표기할 수 있다.

$$\frac{1}{n}\left[\sum_{k=1}^{n} G_{t:t+k}\right] - Q(S_t, A_t) \approx 0 \quad (6)$$

† 교신 저자 한연희



(그림 1) $\boldsymbol{\Omega}$-return 을 적용한 $\boldsymbol{n}$-스텝 DQN 의 성능 비교

수식 6 으로부터 $1 \le k \le n$에 대한 모든 $G_{t:t+k}$ 값들의 평균이 업데이트 타겟으로 좋다는 것을 알 수 있다. 따라서 우리는 이 평균을 G_{avg}라 명명하고 다음과 같이 정의한다.

$$G_{avg} = \frac{1}{n}\left[\sum_{k=1}^{n} G_{t:t+k}\right] \tag{7}$$

한편, 최적의 행동 가치 함수 q^*는 누적 보상의 최대 기댓값 $\max_{\pi} \mathbb{E}[G_t|S_t, A_t]$으로 정의된다. 그러므로 n-스텝 DQN 에서, $1 \le k \le n$에 대한 모든 $G_{t:t+k}$값들의 최대값을 업데이트 타겟으로 사용하면 에이전트는 state-action value 함수 Q를 빠르게 학습할 수 있다. 따라서 우리는 이 최대값을 G_{max}라 명명하고 다음과 같이 정의한다.

$$G_{max} = \max_{k} G_{t:t+k}, \quad 1 \le k \le n \tag{8}$$

2. Ω-return

G_{max}는 초기 단계에 학습을 가속화 시킬 수 있지만 부트스트래핑 값을 과대평가하는 경향이 있어 학습하는 동안 문제가 될 수 있다 [5, 6]. 반면, G_{avg}는 학습 후반에도 업데이트의 타겟으로써 좋다. 그러므로 G_{avg}와 G_{max}를 결합한 업데이트 타겟인 Ω-return 을 제안한다.

$$\boldsymbol{\Omega} = (1-\beta)G_{avg} + \beta G_{max}, \quad 0 \le \beta \le 1 \tag{9}$$

β 는 G_{avg} 와 G_{max} 의 비율을 결정하는 하이퍼-파라미터이다. 우리는 하이퍼-파라미터 β 를 다음의 수식으로 대체 하였다.

$$\beta = \frac{G_{max} - G_{avg}}{G_{max} - G_{min}} \tag{10}$$

위 수식에서 $G_{min} = \min_{k} G_{t:t+k}$, $1 \le k \le n$ 이다. 학습이 진행되면서 β 값은 줄어들어 G_{avg} 의 비율이 높아진다. $G_{max} = G_{avg} = G_{min}$ 일 때, 분모가 0 이 되는 것을 방지하기 위해 분모에 매우 작은 값을 더해 주었다.

Ⅳ. 실험

제안하는 Ω-return 의 성능을 평가하기 위해 Ω-return 을 n-스텝 DQN 에 적용하여 기존의 n-스텝 DQN 과 비교한다.

실험 환경은 OpenAI Gym 의 'MountianCar-v0'이다. 에이전트는 스텝당 -1 의 보상을 받으며 목표지점에 도달하거나 200 스텝이 지나면 에피소드는 종료된다. 에이전트의 행동을 추론할 때는 ϵ-greedy 정책을 사용하였고 ϵ값은 40000 스텝동안 1.0 에서 0.02 까지 선형적으로 감소시켰다. 학습률 $\alpha = 0.0005$, 감가율 $\gamma = 0.99$, 에이전트의 경험이 저장된 메모리 크기는 50000, state-action value Q 네트워크를 업데이트할 때 사용되는 경험의 배치 크기는 32 로 설정하였다.

200000 스텝이 지나면 학습을 종료하였고, 이 과정을 10 번 반복하였다. (그림 1) 그래프는 10 번 반복한 것의 이동 평균을 나타낸다. 실험 결과 그래프는 기존의 n-스텝 DQN 보다 제안하는 Ω-return 을 적용한 n-스텝 DQN 의 성능이 우수하다는 것을 보여준다.

Ⅴ. 결론

n-스텝 TD 학습은 1-스텝 TD 학습보다 좋은 알고리즘으로 알려져 있지만 n, α 등 하이퍼-파라미터와 주어진 환경의 변화에 민감하기 때문에 최적의 n 값을 선택하는 것은 어렵다. 본 논문에서는 $1 \le k \le n$에 대한 모든 k-step 누적 보상의 평균과 최대값으로 구성된 Ω-return 이라는 새로운 타겟을 제안하고 n-스텝 DQN 알고리즘에 적용하여 실험을 진행한다. 실험 결과는 우리가 제안하는 방법이 기존의 n-스텝 DQN 보다 성능이 더 좋다는 것을 입증한다. 추후 연구 계획은 Ω-return 을 double DQN, dueling DQN, PER 등 여러 확장된 DQN 알고리즘에 적용하여 다양한 환경에서 성능을 평가하는 것이다.

ACKNOWLEDGMENT

이 논문은 2018 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2018R1A6A1A03025526 및 No. NRF-2020R1I1A3065610).

참 고 문 헌

[1] Sutton, R. S. Learning to Predict by the Methods of Temporal Differences. Machine Learning, 3(1):9-44, 1988.

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

[3] Asis, K. D., J. Hernandez-Garcia, G. Holland, et al. Multi-Step Reinforcement Learning: A Unifying Algorithm. AAAI, 2018.

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

[5] Thrun, S., A. Schwartz. Issues in using function approximation for reinforcement learning. In In Proceedings of the Fourth Connectionist Models Summer School. Erlbaum, 1993.

[6] van Hasselt, H., A. Guez, D. Silver. Deep Reinforcement Learning with Double Q-learning. In AAAI. 2016.