MCTS 와 결합하여 향상된 Deep Q-Network

Improved Deep Q-Network in combination with Monte Carlo Tree Search

저자
(Authors)

정영빈, 김기범, 김병희
YeongBin Jeong, Kibeom Kim, Byoung – Hee Kim

출처
(Source)

한국정보과학회 학술발표논문집 , 2020.7, 877-879 (3 pages)

발행처
(Publisher)

한국정보과학회
The Korean Institute of Information Scientists and Engineers

URL

http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09874611

APA Style

정영빈, 김기범, 김병희 (2020). MCTS 와 결합하여 향상된 Deep Q-Network. 한국정보과학회 학술발표논문집, 877-879.

MCTS 와 결합하여 향상된 Deep Q-Network

정영빈[1], 김기범[2], 김병희[1]
[1]써로마인드, [2]서울대학교
ybjeong@surromind.ai, kbkim@bi.snu.ac.kr, bhkim@surromind.ai

Improved Deep Q-Network in combination with Monte Carlo Tree Search

YeongBin Jeong[1], Kibeom Kim[2], Byoung – Hee Kim[1]
[1]Surromind. Inc., [2]Seoul National University

요 약

최근 로봇 제어 문제에서 심층 인공 신경망을 사용하는 방법들이 좋은 성능을 보이고 있다. 그 중에서도 Deep Q-Network 는 이산화 된 행동 공간을 학습하는데 있어 매우 높은 성능을 보이고 있다. 하지만 Deep Q-Network 는 학습이 되었을 때, 좋은 성능을 보이지만 로봇이 작업을 학습하기까지 많은 시간이 소모된다는 문제가 있다. 이러한 문제가 발생하는 이유 중 하나는 희박한 보상으로 인해 로봇이 높은 보상을 얻는 행동을 찾아내기까지 걸리는 시간이 길기때문이다. 본 논문에서는 위의 문제를 해결하기 위한 방법 중 하나로 시뮬레이션 기반 트리 탐색을 같이 사용하는 강화학습 기법을 제안한다. 시뮬레이션 기반 트리 탐색을 통해 보상이 높은 행동을 찾아내는 것으로 보다 빠른 학습이 가능하게 되며, 학습이 되었을 때는 트리 탐색을 제외하는 것으로 빠른 행동 계획이 가능하도록 한다. 로봇 팔이 선반 위에 올려진 물체에 도달하는 작업을 통하여 기존의 강화학습 알고리즘과 비교한 결과, 트리 탐색 기법인 MCTS 를 결합하여 보다 적은 시도로 자주 성공함을 확인하였다.

1. 서 론

최근에 심층학습과 강화학습 등 인공지능 기술의 발전으로 이를 활용한 로봇 제어 연구가 활발하다. 그 중에서도 로봇 팔은 산업현장이나 식당과 같은 반복작업을 위한 일에 주로 사용되고 있고, 이러한 작업을 잘 하도록 로봇 팔을 학습시키는 것은 중요한 문제 중 하나이다. 로봇 팔을 자동으로 학습시키는 데는 주로 강화학습 기법이 많이 사용된다. 강화학습은 로봇이 환경과 상호 작용하여 어떠한 행동을 취했을 때 보상을 주고, 얻은 보상을 바탕으로 많은 보상을 받도록 학습되는 기법이다. 그러한 강화학습 기법 중 하나인 Deep Q-Network (DQN)[1]-[2]는 이산화 된 행동 공간에 대해 높은 성능을 보이는 것으로 알려져 있다. 그런데 로봇이 취할 수 있는 행동에 비해 보상을 얻을 수 있는 행동이 상당히 적은 경우가 생길 수 있다. 이러한 것을 보상이 희박하다고 하는데, 이 경우 일반적인 강화학습 기법을 사용한 학습이 매우 느리게 진행된다.

본 논문에서는 이 문제점을 보완하기 위한 방법 중 하나로 몬테 카를로 트리 탐색 (Monte Carlo Tree Search) [3]-[5]을 결합한 강화학습 기법을 제안한다. MCTS 는 탐색하고자 하는 트리가 매우 클 경우 적은 계산으로도 신속한 탐색이 가능하게 해주는 알고리즘으로, 딥 러닝과 결합하여 보상이 희박한 환경에서 희박한 보상을 탐색하고 찾아낸 보상을 바탕으로 효과적으로 학습되는 것을 보여준다.

본 논문에서는 선반 위에 올려진 물체에 로봇 팔이 도달하는 것을 목적으로 하여 시뮬레이션 환경을 구성하고 실험을 진행하였다. 실험 환경에서 로봇 팔의 각 관절은 정해진 각도만큼 움직일 수 있으며 물체에 접촉하면 보상을 얻게 된다. 또한 제안하는 강화학습 알고리즘의 성능을 평가하기 위하여, DQN 알고리즘을 선택하였다. 시뮬레이션 환경에서의 실험을 통하여 최종적으로 기존의 DQN 알고리즘과 본 논문에서 제안하는 알고리즘의 성능 비교하고 MCTS 를 결합한 강화학습 기법이 보다 적은 시도로 자주 성공함을 확인하였다.

2. 배 경

2.1. Monte Carlo Tree Search (MCTS) 알고리즘

MCTS 는 최적의 노드를 찾기 위한 트리 탐색 알고리즘 중 하나이다. 모든 노드를 탐색하는 방법은 탐색해야 하는 트리가 너무 클 경우 탐색에 많은 시간이 소모되기 때문에 탐색의 범위를 줄일 필요가 있다. MCTS 는 시뮬레이션을 사용하여 현재까지 탐색된 노드들의 가치를 추정한 뒤 가치가 높은 노드를 우선 탐색하는 것으로, 탐색해야하는 트리가 클 경우 비교적 적은 시간을 소모하여 탐색을 하는 것이 가능하다. Markov Decision Process (MDP)[6] 를 적용하는 문제에 대해서 일반적으로 MCTS 의 노드는 상태, 엣지는 행동으로 설계한다.

MCTS 의 탐색과정에서 어떤 노드를 탐색할지 결정하는 것은 중요한 매개변수 중 하나인데, 본 논문에서는 Upper Confidence bound for Trees (UCT) 함수[5]를 사용하여 가치가 높을 것으로 예상되는 상태와 시뮬레이션 횟수가 부족한 상태의 균형을 고려한 트리 탐색을 진행한다.

UCT 함수의 수식은 다음과 같다.

$$v_i + C\sqrt{\frac{\ln \ln N}{n_i}} \quad (1)$$

여기서 i는 선택하고자 하는 노드, v_i는 노드 i의 평균 가치, C 는 exploration-exploitation 의 정도를 조정하는 상수이며 N은 부모 노드의 방문 횟수, n_i는 노드 i의 방문 횟수를 의미한다.

2.2. Deep Q-Network (DQN)

DQN 은 MDP 를 기반으로 하는 강화학습 알고리즘 중 하나인 Q-Learning[1] 알고리즘을 심층 인공 신경망을 사용하여 향상시킨 알고리즘이다. Q-Learning 은 현재 상태가 주어졌을 때, 에이전트가 취할 수 있는 각각의 행동에 Q-Value 라는 가치를 매기게 되고, 매겨진 가치를 바탕으로 행동을 선택하게 된다. Q-Value 에 대한 학습이 진행됨에 따라 높은 보상을 받을 것으로 기대되는 행동에 높은 가치가 매겨지게 되고 최종적으로 최적의 행동을 취하도록 학습이 된다. Q-Value 의 학습 수식은 다음과 같다.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left(R_{t+1} + \gamma Q(S_{t+1}, a') - Q(S_t, A_t)\right)' \quad (2)$$

$x_a\ row$ 여기서 S_t 는 t 시점에서의 상태, A_t 는 t 시점에서 가능한 행동, R_{t+1}는 t시점에서의 보상 그리고 $Q(S_t, A_t)$는 상태 S_t에서 행동 A_t의 Q-Value 를 의미한다.

3. 관 련 연 구

MCTS 와 같은 탐색 알고리즘을 활용하여 신경망 기반의 강화학습 알고리즘을 향상시키는 다른 연구도 있었다.

[7]에서는 기존의 Q-Learning[1]이 Q-table 이 완성되지 않았을 때, Q-Table 을 완성시키기 위한 탐색이 랜덤하게 진행되므로 비효율적인 Q-Table 작성이 발생한다는 문제점을 MCTS 를 활용하여 해결하였다. 하지만 신경망을 사용한 Deep Q-Network 의 경우 심층 신경망이 Q-Value 를 추정하여 주기 때문에 Q-Table 작성 단계에서의 탐색의 필요성이 떨어진다. 또한 해당 논문에서는 Q-Table 의 갱신을 위해 기존의 랜덤 탐색을 하는 반면 본 논문에서는 Q-Table 갱신을 위해 MCTS 활용하기 때문에 두 연구에 차이가 있다.

또 다른 연구로 [8]에서는 Asynchronous Advantage Actor-Critic (A3C)[9]와 MCTS 를 결합한 결과를 보여주었다. 위의 논문에서는 A3C 알고리즘을 학습함에 있어서 본 논문에서 말하는 것과 같은 희박한 보상이 학습을 느리게 만든다는 문제가 있어서, 신경망 학습 과정에서 신경망이 출력한 결과가 아닌 여러 개의 MCTS 에이전트를 사용하여 얻은 결과를 바탕으로 신경망을 학습하여 좋은 성능을 보여줬다. 본 논문에서는 단일 MCTS 에이전트를 사용하며, epsilon-greedy[10] 기법을 사용하여 MCTS 를 사용하여 얻은 결과와 신경망을 통해 얻은 결과를 학습 정도를 고려하여 조정하는 것으로 위의 논문과 차이가 있다.

4. 시뮬레이터

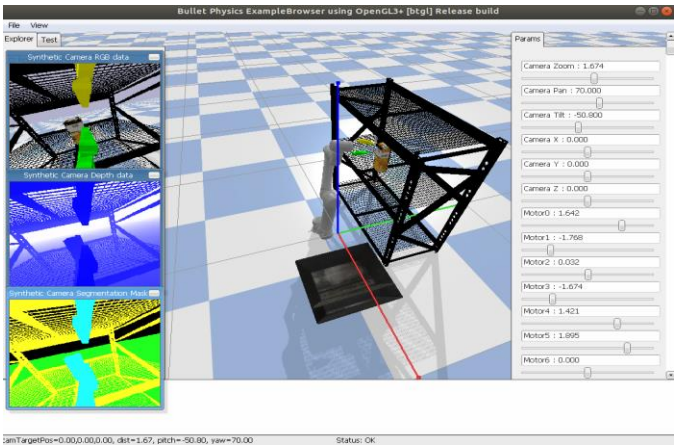

그림 1 실험에 사용된 시뮬레이션 환경

본 논문에서는 실험을 진행하기 위하여 오픈소스 기반의 물리 엔진인 Pybullet 을 사용하여 시뮬레이션 환경을 구성하였다. 시뮬레이션 환경은 메인 에이전트로 UR5 로봇 팔을 사용하며, 2 층으로 이루어진 선반 1 개와, 2 층에 위치한 물체 1 개로 구성하였다.

UR5 로봇 팔은 총 6 개의 관절과 집게 2 개, 총 8 의 자유도를 가진다. 각각의 팔은 0 도부터 360 까지 60 도 단위의 7 가지 움직임을 가지며, 집게의 조이는 정도 또한 7 가지로 나누어 실험하였다. 추가로 로봇 팔은 한 에피소드에서 각 관절 당 5 번씩의 움직임이 가능하도록 설계하였다. 시뮬레이터에서 제공하는 상태에 대한 정보로 로봇 팔의 움직임 정도, 물체의 위치 좌표, 로봇 팔의 집게가 바라보는 영상이 주어지지만 본 실험에서는 강화학습 알고리즘의 빠른 수렴을 목적으로 하였으므로, 고정된 물체에 접촉하는 작업을 위해 비교적 적은 정보인 로봇 팔의 움직임 정보만을 사용하여 상태를 구성하였다.

5. 실 험

표 1 DQN 과 DQN 에 MCTS 를 추가한 알고리즘의 성공한 에피소드 상위 4 개의 에피소드 순번을 보여주는 표.

	첫번째 성공	두번째 성공	세번째 성공	네번째 성공
DQN	689	696	850	851
DQN+MCTS	530	539	558	579

DQN 은 Q-Value 가 높은 행동을 선택하도록 학습이 되는데, 이 때 학습 중 큰 Q-Value 로 인한 편향이 발생 할 수 있기 때문에 epsilon-greedy[10] 기법을 사용하여 정해진 주기마다 Q-Value 와 무관한 랜덤 탐색을 진행하도록 한다. 본 실험에서 DQN+MCTS 는 랜덤 탐색을 MCTS 로 대체한다.

실험에 사용한 매개변수 값은 다음과 같다. 먼저 DQN 의 리플레이 메모리는 100 만으로 설정하였으며, 신경망의 learning rate 는 0.0001, optimizer 로는 Adam[11]을 사용하였다. 그리고 Policy Netowork 는 매 에피소드마다 학습이 되도록 하였으며, Target Network 는 10 에피소드마다 업데이트되도록 구성하였다. 다음으로

MCTS 의 매개변수 설정의 경우 exploration-exploitation 의 정도를 결정하는 상수 C 를 2 로 구성하였다.

표 1 은 실험의 결과를 나타내는 표이다. 본 실험은 강화학습 알고리즘이 보상이 희박한 환경에서 보다 빠르게 수렴하도록 하는 것을 목표로 하였으므로, 학습된 신경망이 어느 시점에서 목표한 작업을 수행하는지를 성능 측정의 기준으로 삼았다. DQN 만 사용한 경우, 전체 작업을 완벽하게 수행하게 되기까지 689 에피소드를 진행하였고, 추가 학습 과정에서 두번째에서 세번째 작업 성공 에피소드의 차이가 큰 것으로 보아, 희박한 보상을 찾아내지 못하는 행동들로 인한 편향이 생기는 것으로 보인다. DQN 과 MCTS 를 결합한 경우, 최초로 작업을 성공하기까지 530 에피소드를 진행하였는데, 기존의 DQN 만 사용한 경우에 비해 에피소드 수가 약 23%정도 줄어든 결과를 확인 할 수 있다. 또한 신경망이 추가로 학습되더라도 다음 작업 수행까지 에피소드 수의 차이가 적은 결과를 보여주는데 이것은 MCTS 를 사용하여 얻은 학습 데이터가 희박한 보상을 찾아내어 보상이 적은 데이터에 대한 편향이 줄어든 것으로 볼 수 있다.

6. 결 론

보상이 희박한 환경에서는 기존의 강화학습 알고리즘을 사용할 경우 학습이 잘 안된다는 문제가 있다. 본 논문에서는 이러한 문제를 해결하기 위해 학습 과정에서 시뮬레이션 기반의 트리 탐색 알고리즘을 결합하였고 시뮬레이션 환경에서 기존의 강화학습 알고리즘에 비해 더 빠른 학습이 가능하다는 것을 보여줬다. 이후 연구에서는 더 복잡한 환경과 복잡한 강화학습 알고리즘에 대해 이러한 기법을 적용해 볼 수 있을 것이다.

감사의 글

이 논문은 2020 년도 정부(산업자원통상부)의 재원으로 한국산업기술진흥원(P0006720-ILIAS),
정부(과학기술정보통신부)의 재원으로 정보통신기획평가원(2017-0-00162-HumanCare, 2019-0-01339-AssembleAI)의 지원을 받았음.

참 고 문 헌

[1] Watkins, Christopher JCH, and Peter Dayan, "Q-learning," *Machine Learning*, 8(3-4), pp. 279-292, 1992.

[2] Mnih, Volodymyr, et al., "Human-level control through deep reinforcement learning," *Nature,* 518(7540), pp. 529-533, 2015.

[3] Tsitsiklis, John N, "Asynchronous stochastic approximation and Q-learning," *Machine Learning*, 16(3), pp. 185-202, 1994.

[4] Coulom, Rémi, "Efficient selectivity and backup operators in Monte-Carlo tree search," *International conference on computers and games*, Springer, Berlin, Heidelberg, 2006.

[5] Browne, Cameron B., et al., "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, 4.1, pp. 1-43, 2012.

[6] Kocsis, Levente, and Csaba Szepesvári, "Bandit based monte-carlo planning," *European conference on machine learning*, Springer, Berlin, Heidelberg, 2006.

[7] Wang, Hui, Michael Emmerich, and Aske Plaat, "Monte Carlo Q-learning for General Game Playing," *arXiv preprint arXiv:1802.05944,* 2018.

[8] Kartal, Bilal, Pablo Hernandez-Leal, and Matthew E. Taylor, "Action Guidance with MCTS for Deep Reinforcement Learning," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 15, No. 1, 2019.

[9] Mnih, Volodymyr, et al., "Asynchronous methods for deep reinforcement learning," *International conference on machine learning*, 2016.

[10] Watkins, Christopher John Cornish Hellaby, "Learning from delayed rewards," 1989.

[11] Kingma, Diederik P., and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.