모방적 강화학습: 시연과 보상으로부터의 학습

곽동현[O1] 이상우[2] 이성태[2] 장병탁[1,2]
협동과정 뇌과학[1], 컴퓨터공학부[2],서울대학교
{dhkwak, cwlee, stlee, btzhang}@bi.snu.ac.kr

Imitational Reinforcement Learning: Learning from Demonstration and Reward Signal

Donghyun Kwak[O1] Sang-Woo Lee[2] Sungtae Lee[2] Byoung-Tak Zhang[1,2]
Brain Science Program[1], School of Computer Science & Engineering[2], Seoul National University

요 약

최근 강화학습은 딥러닝을 함수 근사에 사용하면서, 경이로운 성과를 보이며 빠른 발전이 이루어지고 있는 연구 분야이다. 그런데 일반적으로 강화학습은 문제의 난이도가 높아 빠른 exploration이 어려워 reward를 자주 얻기 힘든 문제에서, 학습에 실패하는 경우가 빈번하다. 반면 Learning from Demonstration은 명시적인 reward가 없는 문제에서 오직 전문가가 보여준 시연으로부터 reward에 대한 함수를 학습하고, 이렇게 구한 reward를 사용해 다시 강화학습을 하여 최종적으로 문제를 해결하는 연구 분야이다. 그런데 실 세계의 많은 문제에서는 reward 함수를 정의하는 것이 매우 어려워 아주 희소한 reward를 얻는 것만이 가능하다. 그러나 이러한 문제에서 적당한 길이의 전문가의 시연을 수집하는 것은, reward 함수를 정교하게 정의하는 것에 비해 훨씬 더 수월한 경우가 많다. 본 논문에서는 이 두 가지를 접근 방법을 모두 활용한 문제 해결하는 방법을 제시하고, 각각의 알고리즘이 갖는 성능 차이를 비교 분석하는 연구를 수행하였다.

1. 서 론

강화학습은 최근 딥러닝을 이용해 Q-value function 혹은 policy 등의 함수를 근사하면서 Deep Reinforcement Learning(이하 Deep RL)이라고 불리우고 있다. 가장 기본적인 형태의 Deep RL은 강화학습의 Q-value function을 딥러닝을 통해 근사한 Deep Q Networks 이다[1]. 현재 Deep RL 분야는 굉장히 빠른 속도의 발전을 하고 있으며, 매우 다양한 형태의 Network 구조들이 생겨나고, 또한 알고리즘 도 더 강건하고, 빠른 학습이 가능한 연구가 진행되고 있다[2][3][4].

그러나 이러한 연구들에서도 모두 공통적으로 겪는 어려움 중 한가지는 바로 exploration이 어려워서 충분한 수의 Reward를 받지 못하는 deep exploration이 필요한 문제 케이스이다[5]. 이러한 경우는 근본적으로 강화학습에 필요한 reward signal을 받는 것이 너무나 어려워 학습에 실패해버리는 경우가 많다. 그래서 이런 exploration이 어려운 문제를 해결하기 위한 연구들은 최근 Deep RL 연구분야에서의 큰 이슈 중 하나이다[6][7].

반면 Learning from Demonstration이라는 연구에서는 환경으로부터 받는 reward signal이 없는 대신, 전문가의 시연으로부터 reward function을 먼저 학습하는 역강화학습을 수행한 뒤, 이를 reward로 사용해 다시 강화학습을 하여 문제를 해결하는 indirect한 방법을 사용한다. 이러한 문제 해결 방식은 reward function을 정의하기 어려운 대부분의 현실 문제들에서 전문가의 행동을 모방하는 policy를 학습하기 위해 만들어진 방법이다[8][9].

그러나 Learning from Demonstration을 하기 위해 먼저 필요한 역강화학습에서는 reward function을 학습하는 것이 state의 feature들에 대한 linear combination라는 가정을 하고 있어 실제로 reward 함수가 비선형인 경우 학습이 제대로 되지 않는다.

그런데 앞서 제안된 서로 다른 이 두 가지 방법론을 동시에 사용할 경우, 각각의 단점을 상호보완 할 수 있는 효과적인 학습이 가능하다. 본 논문에서는 강화학습에서 사용 가능한 reward와 더불어 전문가의 시연으로부터 학습한 reward function을 동시에 사용하는 모방적 강화학습 방법을 제안한다. 그리고 이러한 방법을 쓸 경우 정보량이 부족한 각각의 방법에 비해 훨씬 더 빠르고 안정적으로 최적 policy를 학습하는 것이 가능함을 일련의 실험을 통해 보인다.

2. 알고리즘

2.1 Reinforcement Learning

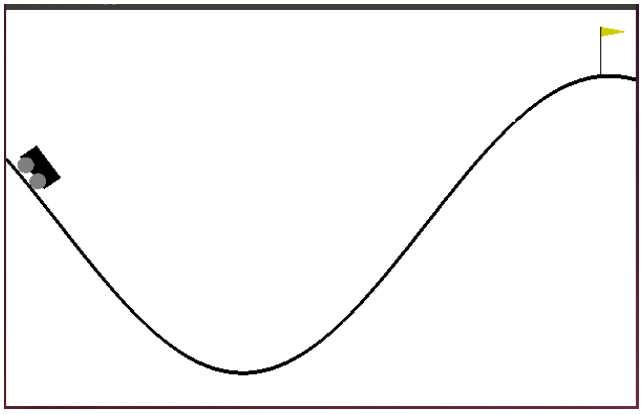

그림 1. Mountain-Car 문제

Mountain-Car 문제의 state는 연속된 2차원 실수 공간으로, 차의 x축 위치와 속도로 정의된다. Action은 [좌측 가속, 우측 가속, 가속 없음]의 3가지 이산 공간으로 정의된다. Reward는 가장 우측의 깃발에 도달하면 +1을, 그렇지 않으면 0을 받는다. 또한 본 논문에서는 Exploration의 난이도를 낮추기 위해 8-frame skipping[12]을 사용하여 전체적인 time scale을 작게 설정하였다.

강화학습은 사용하는 방법론에 따라 크게 Value-Based와 Policy-Based의 두 가지 방법으로 나뉜다. Value-Based 방법의 경우, Q-value 함수 혹은 state-value 함수를 학습한 다음, 이에 따라서 가장 큰 value를 선택하도록 자동 결정되는 deterministic policy를 사용한다. 반면에 Policy-Based 방법의 경우 value에 대한 학습 없이, 바로 policy 함수를 미분하여 discounted sum of future reward를 최대화하기 위한 방향으로 gradient descent method를 사용해 가장 많은 reward를 받을 수 있는 최적의 policy를 학습한다[10].

이러한 강화학습에서는 환경으로부터 받는 reward를 그대로 사용하여, 각 state와 action의 pair에 대한 credit assignment 문제를 푸는 것으로 결국 최적의 policy를 학습한다.

2.2 Apprenticeship Learning

견습학습은 강화학습과 달리 환경으로 받는 명시적인 reward가 존재하지 않고, 대신 이 reward 함수를 전문가의 시연으로부터 학습하여 사용한다. 그 후 미리 찾아낸 reward 함수를 사용한 강화학습을 하여 최적의 policy를 학습하게 된다. 따라서 이러한 견습학습의 목적은 강화학습과 달리, 전문가의 시연을 일반화해서 따라 할 수 있는 policy를 학습하는 것이 된다[8][9].

2.3 Imitational Reinforcement Learning

본 논문에서 제안하는 모방적 강화학습은 앞서 설명한 강화학습과 견습학습의 장점을 동시에 사용하는

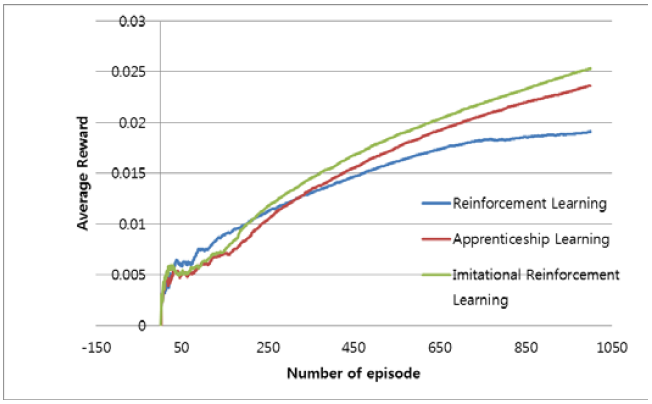

그림 2. 각 알고리즘의 학습에 따른 평균 reward 증가 그래프

이 그래프는 세가지 알고리즘의 학습 속도를 Mountain Car 문제에서 평균 reward로 측정한 것이다. X축은 학습하는 데 사용한 에피소드의 개수이고, Y축은 학습하면서 얻은 모든 보상을 총 time step으로 나눈 평균 reward 값이다. 실험 결과 강화학습이 가장 느리고, 견습학습은 그 다음으로 빨랐으며, 모든 정보를 다 활용한 모방적 강화학습이 가장 빨랐다.

방법이다. 강화학습에서의 deep exploration 문제의 경우 reward 값이 희소하여 학습이 매우 어려운 문제를 견습학습에서 시연으로부터 학습한 reward 함수를 사용하여 문제를 해결하는 것이다. 본 논문에서는 견습학습 단계의 역강화학습으로는 Feature Matching 알고리즘을 사용하였고, 선형으로 된 reward 함수를 가정하여 학습하였다[9]. 즉 모방적 강화학습에서는 최종 reward 값을 환경으로부터 얻은 reward와 견습학습을 통해 얻은 reward function의 값을 더하여 사용하였다. 또한 최종적으로 policy를 학습하는데 필요한 강화학습 알고리즘은 기존에 널리 쓰이는 Q-learning을 사용하여 학습하였다[11].

3. 실험

실험에서 비교로 사용한 알고리즘은 Q-learning만을 이용한 일반적인 강화학습, 전문가로부터 얻은 최적의 시연 데이터 3개를 사용해 학습한 견습학습, 그리고 마지막으로 앞의 두 가지 알고리즘에서 사용한 reward 정보를 모두 활용한 모방적 강화학습, 이 세가지를 비교하였다. 실험은 Mountain Car [그림 1]라는 오래된 강화학습의 문제를 선택하였다. 이 문제를 선택한 이유는 전문가의 시연 데이터를 수집하기가 매우 용이한 환경이기 때문이다.

실험 결과 [그림 2]와 같이 시연 정보와 환경으로부터 얻은 reward를 모두 사용한 모방적 강화학습이 가장 빠른 속도로 최적의 policy를 학습하는 것을 볼 수 있다. 만약 이 문제가 무작위

표 1. 실험에서 사용한 상세 파라미터 설정

실험 변수	변수 값
각 에피소드 최대 시간 길이	100
Epsilon-greedy가 줄어드는 최대 시간 길이	50000 시간 스텝
Epsilon-greedy annealing method	Linear annealing
Training Network 학습 주기	매 에피소드마다
Target Network 업데이트 주기	10 에피소드마다
Adaptive Learning Rate 알고리즘	RMS prop
Discount Factor	0.99
Feature Matching 학습 알고리즘	Linear projection

exploration으로는 결코 reward를 얻을 확률이 희박한 어려운 문제일수록 알고리즘간의 성능차이는 훨씬 더 명확히 나타났을 것이다.

실험에서 강화학습이 가장 학습 속도가 느렸는데, 이는 Mountain-Car 문제에서 무작위 exploration으로 정상에 도착하는 것이 상대적으로 어렵기 때문이다. 따라서 무작위 exploration보다 훨씬 수월히 초반 exploration을 할 수 있도록, 전문가 시연으로부터 reward 함수를 학습한 견습학습이 강화학습보다 더 빨리 학습하였다.

실험에 사용한 자세한 알고리즘 파라미터와 실험 환경은 [표 1]에 정리되어 있다. 또한 본 실험 알고리즘의 구현은 TensorFlow를 사용하였고[13], 문제 환경은 OpenAI Gym을 사용하였다[14].

4. 논의 및 결론

본 논문에서는 아주 간단한 방법으로 강화학습과 견습학습에서 구한 reward를 결합하여 학습을 진행하였다. 역강화학습에서는 선형으로 된 reward 함수를 가정하고 이를 closed form solution으로 구했기 때문에 시간계산량 상의 손실은 거의 무시할 수 있는 수준이었다. 대신 전문가의 시연 데이터를 수집해야 한다는 단점이 있지만, 실험에 사용한 문제보다 훨씬 복잡한 환경에서는 무작위 exploration으로는 거의 학습이 불가능한 경우가 많기 때문에 이는 충분히 감수할 만한 비용이다.

이와 반대로 전문가의 시연 데이터만 있는 문제의 경우에도, 아주 간단한 reward 함수를 직접 정의해서 사용하는 것은 대부분 쉽게 가능하다. 따라서 이 두 정보를 모두 사용해서 모방적 강화학습은 어떠한 문제 상황에서도 큰 단점을 갖지 않는 아주 일반적인 방법이다.

후속 연구에서는 reward 함수를 보다 잘 결합할 수 있는 방법과 전문가의 시연으로부터 환경의 다이나믹스를 학습한 model을 사용해 더욱 빠른 학습 알고리즘을 연구하고자 한다.

감사의 글

이 논문은 2016년도 정부(미래창조과학부, 국방부)의 재원으로 정보통신기술진흥센터(R0126-16-1072-SW스타랩), 한국산업기술평가관리원(10044009-HRI.MESSI, 10060086-RISF), 국방과학연구소(UD130070ID-BMRR)의 지원을 받았음.

참고문헌

[1] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529-533.
[2] Lever, Guy. "Deterministic policy gradient algorithms." (2014)..
[3] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." arXiv preprint arXiv:1602.01783 (2016).
[4] Wang, Ziyu, Nando de Freitas, and Marc Lanctot. "Dueling network architectures for deep reinforcement learning." arXiv preprint arXiv:1511.06581 (2015).
[5] Osband, Ian, et al. "Deep Exploration via Bootstrapped DQN." arXiv preprint arXiv:1602.04621 (2016).
[6] Mnih, Volodymyr, et al. "Strategic Attentive Writer for Learning Macro-Actions." arXiv preprint arXiv:1606.04695 (2016).
[7] Bellemare, Marc G., et al. "Unifying Count-Based Exploration and Intrinsic Motivation." arXiv preprint arXiv:1606.01868 (2016).
[8] Ng, Andrew Y., and Stuart J. Russell. "Algorithms for inverse reinforcement learning." Icml. 2000.
[9] Abbeel, Pieter, and Andrew Y. Ng. "Apprenticeship learning via inverse reinforcement learning." Proceedings of the twenty-first international conference on Machine learning. ACM, 2004.
[10] Sutton, Richard S., and Andrew G. Barto. Introduction to reinforcement learning. Vol. 135. Cambridge: MIT Press, 1998.
[11] Watkins, Christopher JCH, and Peter Dayan. "Q-learning." Machine learning 8.3-4 (1992): 279-292.
[12] Bellemare, Marc G., Joel Veness, and Michael Bowling. "Investigating Contingency Awareness Using Atari 2600 Games." AAAI. 2012.
[13] Abadi, Martın, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467 (2016).
[14] Brockman, Greg, et al. "OpenAI Gym." arXiv preprint arXiv:1606.01540 (2016).