실제 주행 데이터를 이용한 오프라인 강화학습기반 자율주행 에이전트 학습

이성하○, 김경중*
광주과학기술원 융합기술원 융합기술학제학부
shlee0414@gm.gist.ac.kr, kjkim@gist.ac.kr

Training Self-Driving Vehicles using Offline Reinforcement Learning with Real Driving Data

Sung-Ha Lee○, Kyung-Joong Kim*
School of Integrated Technology, Gwangju Institute of Science and Technology

요 약

최근, 강화학습을 이용하여 자율주행차의 제어기를 학습하려는 시도가 다양하게 이루어지고 있으나, 대부분 시뮬레이터를 활용한 연구들이다. 강화학습의 경우 시행착오를 통한 직접 경험을 통해 데이터를 수집하고, 이를 통해 학습을 진행하기 때문에, 기존에 쌓여 있는 빅 데이터를 잘 활용하는 교사학습과 달리 확장성에 어려움이 있다. 본 논문에서는 강화학습 중 별도의 탐색(Exploration) 과정 없이 주어진 데이터만으로 학습을 진행하는 오프라인 강화학습을 이용해 기존에 수집한 실제 주행 데이터로부터 자율주행 에이전트를 학습하는 방법을 제안한다. 모형 자율주행차를 사용한 실험 결과 오프라인 강화학습 방법 중 배치 제한 심층 Q 학습(Batch-constrained Deep Q Learning) 방법이 좋은 성능을 보임을 확인하였다.

1. 서 론

최근 모방학습(Imitation Learning)과 강화학습을 적용하여 자율주행 자동차를 구현하려는 연구가 활발히 이루어지고 있다[1]. 그러나 이러한 연구들에도 불구하고 강화학습을 적용하여 자율주행을 구현한 사례는 모방학습에 비해 찾아보기 힘들다. 이 현상에 대한 원인 중 하나는 일반적으로 강화학습은 학습 도중 실제 환경과의 상호작용이 필요하기 때문이다. 이 때문에 강화학습은 자율주행과 같이 실패하였을 때 위험부담이 큰 임무에 대해서는 적용하기 힘들다. 시뮬레이터를 만들어 그 안에서 에이전트를 학습시키는 시도들도 있지만[1)2)], 이러한 시뮬레이터는 만들기도 어렵고, 이후에 시뮬레이터 안에서 학습한 에이전트를 실제 환경으로 옮길 때 문제가 발생할 수 있다.

본 연구에서는 이러한 문제점을 해결하기 위해 별도로 환경과의 상호작용 없이도 이미 수집해놓은 데이터만을 이용하여 학습하는 방법을 사용한다. 첫째로 실제 환경에서 모형 자동차 동키카(Donkey Car)[2]를 주행하면서 수집한 데이터를 사용하였다. 특히 주행에 성공한 데이터뿐만이 아니라 실패한 데이터를 함께 포함하는 데이터를 구축하였다. 둘째로 환경과 상호작용 없이 수집한 데이터만으로 학습을 진행하는 오프라인 강화학습(Offline Reinforcement Learning)[3]을 도입하였다. 본 연구에서는 대표적인 오프라인 강화학습 방법인 배치 제한 심층 Q 학습(Batch-Constrained Deep Q Learning, BCQ)[4]과 랜덤 앙상블 조합(Random Ensemble Mixture, REM)[5]을 사용한다.

기존 온라인 강화학습을 이용하여 자율주행차를 학습하는 연구는 정교한 시뮬레이터를 활용한 연구들이었지만, 본 연구에서는 실제 환경에서 수집한 데이터를 기반으로 오프라인 강화학습을 적용한 연구로 1) 기존 주행 관련 빅 데이터를 활용할 수 있는 확장성과 2) 실제 차로 전이가 쉽다는 기여점을 가지고 있다.

2. 관련 연구

2.1 오프라인 강화학습 (Offline RL)

대부분의 강화학습 알고리즘은 에이전트가 지속적으로 온라인 환경과 상호작용하면서 보상의 합을 최대화하는 정책(policy)을 찾는다. 기존의 온라인 강화학습은 기본적으로 환경과의 상호작용에 기반하여 학습을 진행하는 반면 오프라인 강화학습은 탐색 과정 없이 주어진 데이터만을 이용하여 학습을 진행한다.

오프라인 강화학습은 사전에 수집한 데이터만으로 학습을 진행한다는 점에서 모방학습과 유사한 면이 있지만 두 방법 사이에는 큰 차이점이 있다. 대부분의 모방학습은 주어진 데이터가 최적이라고 가정하고 데이터에 나타난 전문가의 정책을 있는 그대로 모방하려고 한다. 결과적으로 데이터에 내포된 정책을 그대로 따라 하며, 그 이상을 목표 하지는 않는다. 반면 오프라인 강화학습은 보상의 합을 최대화하도록 학습을 수행하기 때문에, 데이터를 생성해낸 정책의 성능을 넘어서는 행동을 학습할 수 있다. 다만, 오프라인 학습은 기존의 off-policy 강화학습에 비해 학습이 더 어려운 것으로 알려져 있다.

버클리 대학의 오프라인 강화학습 벤치마크 중 하나인 D4RL[6]은 시뮬레이터를 이용하여 오프라인으로 CARLA 시뮬레이터 기반 자율주행 에이전트를 학습시키려 시도

1) https://carla.org/
2) https://microsoft.github.io/AirSim/

를 하였다. 본 연구는 모형 자율자동차를 실제 환경에서 수집한 데이터로 오프라인 강화학습을 수행하는 점에서 차이가 있다.

3. 오프라인 강화학습 기반 자율주행

본 연구에서는 전문가가 모형 자동차를 실내 환경에서 조작하여 주행 데이터를 수집하였다. 수집한 데이터에 오프라인 강화학습을 적용한 후, 학습한 모델을 새로운 데이터에 적용하는 형태로 작업을 진행하였다. 전문가의 행동, 즉 성공한 에피소드의 데이터만을 이용하여 학습시키는 모방학습과는 달리 Q값을 보다 정확히 알아내기 위하여 실패한 에피소드와 성공한 에피소드 모두를 학습 데이터에 포함하였다.

입력 상태(state)는 RGB 이미지와 속력으로, 행동은 방향을 바꾸는 스로틀(throttle)과 속력을 바꾸는 스티어링(steering)으로 나뉘며 각각 -1에서 1까지의 값을 가진다. 보상 함수(reward function)는 임무 중간에 보상이 자주 있다면 목적을 이루지 않고 중간 보상만을 얻기만을 위해 행동할 수도 있으므로 간단하게 설계하였는데, 목표에 도달하였을 때 +1, 충돌이 일어났을 때 -1을 주었다. 또한 너무 저속력으로 주행하지 않게 하기 위하여 매 스텝마다 -0.005의 보상을 주었다.

3.1 배치 제한 심층 Q 학습(BCQ)

강화학습의 행동 가치를 의미하는 Q값을 최대화하는 행동을 선택할 때, 일반적인 Q-Learning과 달리 데이터 배치(batch)에서 나타날 법한 행동을 특별히 고려하는 알고리즘이다. 따라서 손실함수 L은 일반적인 Q-Learning과 달리 Q값을 최대화하는 행동을 선택할 때 행동이 제한된다는 조건이 붙어 다음과 같은 식으로 변형한다.

$$L(\theta) = \ell_k(r + \gamma(\max_{a', \frac{G_w(a'|s')}{\max_{\hat{a}} G_w(\hat{a}|s') > \tau}} Q_{\theta'}(s', a')) - Q_\theta(s, a))$$

이때 G_w는 배치 내에서의 행동과 비슷한 행동을 생성하는 생성기이다.

3.2 랜덤 앙상블 조합(REM)

REM은 Q값 여러 개를 추정한 후 각 학습 단계에서 그 Q값들을 랜덤하게 조합한 것을 새로운 Q값으로 사용한다. 따라서 손실함수 L은 다음과 같은 식으로 변형되어 나타내진다.

$$L(\theta) = E(\ell_\lambda(\sum_k \alpha_k Q_\theta^k(s,a) - r - \gamma \max_{a'} \sum_k \alpha_k Q_{\theta'}^k(s', a')))$$

이때 ℓ_λ는 후버 손실(Huber loss)이며 α는 최적의 Q값을 근사하기 위한 카테고리 분포(Categorical distribution)이다.

4. 실험 및 평가

RC카에 라즈베리 파이와 카메라 등이 장착된 형태인 동키카(그림 1)로 실내 환경에서 직진 후 우회전을 하는 26 에피소드(총 2400스텝)의 데이터를 수집한 후, 이 데이터를 사용하여 에이전트를 학습하였다. 이때 데이터에는 주행에 실패한 에피소드와 성공한 에피소드 모두가 포함되어 있다. 이후 대표적인 오프라인 강화학습 방법인 BCQ와 REM 두 가지 알고리즘을 이용하여 수집한 데이터들로 총 100,000스텝 동안 자율주행 에이전트를 학습시켰다.

네트워크 모델은 배우-비평가(Actor-Critic) 모델에 행동 분포 값들을 안정시켜주기 위해 배치 정규화(Batch Normalization) 층을 각 층 사이에 추가하여 사용하였으며 컨볼루션(Convolution) 층을 통하여 들어오는 이미지를 전처리 해 주었다. BCQ와 REM 모두 배치 수는 100이었으며 감쇄율(discount rate)은 0.99, 타겟 업데이트 비율은 0.005였다. REM의 경우 랜덤한 Q값 수는 50이었다.

그림 1. 실험에 사용한 Donkey 자동차

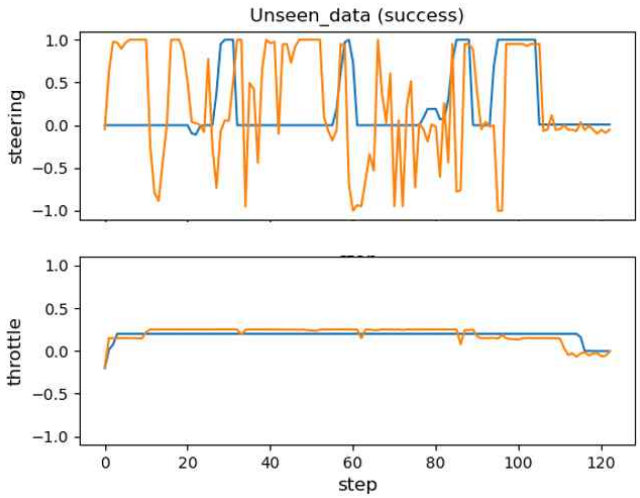

그림 2. Unseen 데이터 (주행 성공한 케이스)에 대한 에이전트(BCQ)의 행동 (푸른색: Test 데이터, 주황: BCQ)

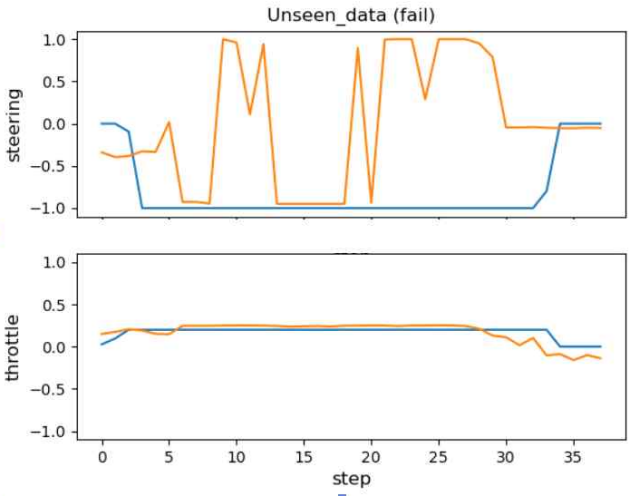

그림 3. Unseen 데이터 (주행 실패한 케이스)에 대한 에이전트(BCQ)의 행동

평가는 오프라인 환경에서 이루어졌으며 에이전트를 학습할 때 사용하지 않은 데이터를 이용하였다. 평가 데이터는 성공적으로 우측으로 핸들을 꺾어 주행에 성공한 에피소드와 좌측의 장애물에 충돌해 실패한 에피소드의 데이터를 각각 사용하였다.

그림 4. 성공한 상황 중 우회전을 해야 할 때 BCQ 에이전트(푸른색)와 데이터(붉은색)에서 나타난 행동(좌측), 실패한 상황 중 좌측 장애물에 충돌하기 직전에 나타난 행동(우측)

BCQ 알고리즘의 경우 성공한 데이터를 이용하여 평가한 그림 2를 보면 비록 스티어링 값이 불안정하기는 하나 90스텝 부근에서 우회전을 할 때 에이전트 역시 우회전을 하는 판단을 내리는 것을 볼 수 있다. 실패한 데이터를 이용하여 평가한 그림 3을 보면 에이전트의 행동(주황색)이 계속 좌회전을 하는 데이터의 행동(푸른색)에 비해 우회전을 하여 충돌하지 않으려 하는 등 실패한 데이터와 달리 적합한 행동을 취하는 것을 관찰할 수 있었다. 만일 성공한 데이터만으로 학습시켰다면 이러한 상황에서 어떠한 판단을 해야 할지 알 수 없었을 것이나, 실패한 데이터를 포함함을 통하여 에이전트가 그러한 행동을 하지 않는, 데이터에 나타난 정책보다 좋은 정책을 취하게 할 수 있었다.

그림 2와 3을 보면 스로틀의 경우 값이 약 0.2 정도로 비교적 일정하게 나왔음을 알 수 있다. 이는 데이터에서 스로틀 값의 대부분이 0.2로 고정되어 있었기 때문에 행동의 범위를 배치 데이터의 행동으로 제한하는 BCQ 알고리즘의 경우 에이전트의 행동 역시 0.2 가량의 제한적인 값을 가졌다. 그러나 수집한 데이터의 다양성이 부족하였기 때문에 0.2 외의 다른 값들은 거의 나타나지 않았다.

REM 알고리즘은 BCQ 알고리즘에 비해 안정적이지 못했으며 스티어링과 스로틀 모두 대부분 -1 혹은 1의 극단적인 값만을 출력하였다. 이는 BCQ 알고리즘과 달리 행동을 제한해주는 부분이 없었기에 실제 데이터 상에서 자주 나타난 행동과는 크게 다른 값을 네트워크가 출력할 수 있기 때문이다. 이러한 현상은 BCQ에서처럼 행동을 제한 해주는 것이 필요함을 보여준다.

전체적으로 성능에 악영향을 준 원인은 크게 두 가지로 나누어지는데, 먼저 보유하고 있었던 데이터의 개수가 비교적 적었으며, 오프라인 강화학습에서는 다양한 정책을 따라 움직이는 데이터들이 있어야 유리하나 데이터의 다양성 역시 작았다. 또 다른 문제점으로는 보상함수를 들 수 있다. 본 연구에서는 에이전트가 중간 보상만을 쫓는 상황을 피하기 위해 보상 함수를 간단하게 설정하였으나 주행과 같은 복잡한 상황에서는 적합하지 않았을 수 있다.

그림 5. 성공한 상황 중 우회전을 해야 할 때 REM 에이전트(푸른색)와 데이터(붉은색)에서 나타난 행동(좌측), 실패한 상황 중 좌측 장애물에 충돌하기 직전에 나타난 행동(우측)

5. 향후 연구

오프라인 강화학습은 에이전트가 뛰어난 성능을 가지도록 학습시키는 데 있어 상당한 양의 데이터를 필요로 한다. 때문에 적은 에피소드로 학습 속도를 더 빠르게 하기 위해 에이전트를 모방학습으로 먼저 학습시키고 그 에이전트를 이용하여 파라미터를 초기화(Initialize) 시켜보다 학습이 효율적으로 진행될 수 있도록 한다. 또한 데이터의 경우 다양성이 부족하였다. 따라서 차후 에이전트를 학습시킬 때 데이터 내의 행동의 다양성과 데이터의 양을 늘려 학습을 진행하도록 한다.

본 논문에서는 충돌 등 위험이 따라오는 실제 환경과 일체의 상호작용 없이 자율주행 에이전트를 구현하기 위하여 순수한 오프라인 환경에서만 학습과 평가를 진행하였다. 때문에 실제 환경에서 에이전트가 주행을 하였을 때 에이전트의 행동이 오프라인에서 평가한 것과 약간의 차이를 보일 수 있다. 따라서 에이전트를 동키카와 같은 실차에 올려 실제로 테스트 해보는 것이 필요하다.

감사의 글

이 논문은 2020년도 광주과학기술원의 재원으로 글로벌 선도대학 육성 사업의 지원을 받아 수행된 연구임
본 연구는 UD180026RD 위탁연구의 일환으로 방위사업청과 국방과학 연구소의 지원으로 수행되었음

참고문헌

[1] M. Bojarski *et al.*, “End to end learning for self-driving cars,” arXiv preprint arXiv:1604.07316 (2016).

[2] DonkeyCar, https://www.donkeycar.com/

[3] Levine, Sergey, *et al.* “Offline reinforcement learning: Tutorial, review, and perspectives on open problems,” arXiv preprint arXiv:2005.01643 (2020).

[4] Scoot Fujimoto, David Meger, and Doina Precup, “Off-policy reinforcement learning without exploration,” *International Conference on Machine Learning*, 2019”

[5] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi, “An optimistic perspective on offline reinforcement learning,” *International Conference on Machine Learning*, 2020.

[6] Datasets for Deep Data-Driven Reinforcement Learning, https://github.com/rail-berkeley/d4rl