자율주행자동차의 안전 주행을 위한 강화학습 알고리즘의 연속적인 행동 공간 이산화

주호택°, 박현철, 김경중*
세종대학교 컴퓨터공학부
ureca87@gmail.com, p890413@gmail.com, kimkj@sejong.ac.kr

Discretization of Continuous Action Space of Reinforcement Algorithm for Safe Autonomous Vehicle Driving

HoTaek Joo°, HyunCheol Park, Kyung-Joong Kim*
School of Computer Science and Engineering, Sejong University

요 약

자율주행 자동차 분야에서 강화학습(Reinforcement Learning)은 딥러닝(Deep Learning)과 함께 가장 주목받는 기술이다. 강화학습이란 기계 학습의 방법 중 하나로 현재 상태(state)에서 어떤 행동(action)을 선택했을 때 받는 미래 보상(reward)의 합을 최대화하는 방향으로 행동하여 학습하는 방법이다. 본 논문의 선행 연구에서 자율주행 자동차와 같은 연속적인 행동 공간(Continuous Action Space)에서 기존의 강화학습을 적용하면 직선코스에서의 핸들 떨림 문제를 확인하였다. 본 논문에서는 이 문제를 해결하기 위하여 연속적인 행동 공간을 이산화(Discretization)하는 방법을 제시하였으며, 이를 구현하기 위해 경주용 차량 시뮬레이터인 TORCS(The Open Racing Car Simulator)에서 강화학습 알고리즘 중 하나인 PPO(Proximal Policy Optimization)를 접목하여 실험을 진행하였다.

1. 서 론

최근 자율주행 자동차는 규칙기반 주행 제어를 벗어나 딥러닝을 중심으로 연구가 활발하다. 자율주행 자동차 연구를 위한 딥러닝 기반 학습법으로는 대표적으로 심층 강화학습과 모방학습이 있다[1][2]. 자율주행 자동차는 핸들의 각도, 엑셀 및 브레이크 값과 같은 연속적인 행동 공간을 갖는데, 이를 제어하기 위한 강화학습 알고리즘으로는 대표적으로 DDPG(Deep Deterministic Policy Gradient)[3], Continuous A3C(Asynchronous Methods for Deep Reinforcement)[4], Continuous PPO(Proximal Policy Optimization)[5] 등이 있다.

이러한 알고리즘들의 가장 큰 문제는 출력 층 노드의 표현 범위에 있다. DDPG 알고리즘은 Actor 출력 층 노드에서 행동 범위 전체를 표현하는 구조이며, Continuous A3C, PPO는 평균값과 표준편차를 구한 후 가우시안 분포(Gaussian Distribution)를 이용하여 행동 범위의 값을 출력하는 구조이다. 즉, 이 세 가지의 알고리즘 모두 출력 층의 단 하나의 노드에서 모든 범위의 액션 값을 출력한다.

이러한 모델은 딥러닝 신경망의 전방 전파(forward propagation)에 있어 표현 범위의 과부하를 야기할 수 있으며, 자율주행 자동차의 관점에서 보면 안전하고 편안한 주행을 하는데 큰 문제로 다가올 수 있다. 본 논문의 선행연구로 세 가지 알고리즘을 실험해본 결과 직선 코스를 주행할 때 핸들을 좌우로 급격히 흔들면서 주행하는 문제를 확인할 수 있었다.

* 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (2017R1A2B4002164) *교신저자

본 논문에서는 앞서 언급한 연속적인 행동 공간에서 대부분의 강화학습 알고리즘이 갖는 문제를 해결하기 위해 연속적인 행동 공간 이산화(Discretization)를 제시한다. 앞서 언급한 문제점을 해결하기 위해 강화학습 알고리즘인 PPO와 실험 환경으로 경주용 차량 시뮬레이터인 TORCS(The Open Racing Car Simulator)[6]를 사용하였다.

2. 관련 연구

2.1 Policy Gradient

강화학습 알고리즘 분류의 기준은 가치기반 강화학습(Value-based RL)과 정책기반 강화학습(Policy-based RL)로 나뉘는데, 자율주행자동차와 같은 연속적인 공간을 다루는 강화학습은 정책기반 강화학습을 주로 사용한다. 정책이란 어떤 상태에서 취할 수 있는 전체 행동에 대한 확률 분포를 나타내는 용어로, 정책기반 강화학습은 현재 상태를 입력으로 넣어, 선택할 수 있는 각 행동의 확률분포를 출력 층의 노드에서 얻어내는 방식이다. 정책 모델은 목적 함수(objective function)를 최적화(optimize)하는 방식을 통해 업데이트 하는 방식인데 목적함수 $J(\theta)$는 현재 상태(state)의 특정 행동(action)이 행해질 때의 미래 보상(reward)의 총합을 통해 계산한다.

$$J(\theta) = E[\sum_{t=0}^{T-1} r_{t+1} | \pi_\theta]$$

$$\nabla_\theta J(\theta) \simeq E[\nabla_\theta \log \pi_\theta(a_t|s_t) A_t]$$

$$\theta \leftarrow \theta + \alpha \nabla J(\theta)$$

목적 함수의 미분 값은 다음과 같으며 손실함수(Loss Function)로 사용하여 경사 상승을 통해 모델을 업데이트 한다.

2.2 PPO

전통적인 Policy Gradient는 단일 정책 신경망의 손실함수를 계산하여 업데이트를 하였다면, PPO 알고리즘은 이전 정책(old policy)과 업데이트가 진행 중인 정책(new policy)의 비교를 통해 모델을 개선한다.

$$maximize\, E_t[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}A_t] = E_t[r_t(\theta)A_t]$$

기존의 Policy Gradient와 다른 점은 $\log \pi_\theta(a_t|s_t)$라는 수식이 $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$로 변화했다는 점이다. 변화한 수식은 이전정책에서 선택한 행동을 기준으로, 업데이트 하는 정책을 이전 정책으로 나눔으로써 비율(ratio)을 생성하여 정책을 비교한다. 이 때 (A_t(Advantage) > 0) 때는 선택한 샘플 내에서는 좋은 편에 속한다는 것을 의미하고, 이와 동시에 앞서 정한 비율을 곱하여 현재 정책의 방향성 및 크기를 정한다. 단, 좋은 방향성을 가지고 있다고 하더라도, 손실함수의 크기는 모델의 전 구역의 파라미터(weight, bias)에 영향을 미치므로 이 전의 정책을 기반으로 신뢰구간(Trust Region)을 생성하여 업데이트의 비율을 제어한다.

$$E_t[min(r_t(\theta)A_t, clip(r_t(\theta),1-\epsilon,1+\epsilon)A_t)]$$

3. 연구 내용 및 방법

3.1 TORCS 시뮬레이터 및 실험 환경 구축

TORCS는 오픈 소스 기반 경주용 차량 시뮬레이터로 실제 자율주행 자동차를 실험하기 전 초기 단계의 테스트 용도로 자주 사용하며, 본 논문에서 사용한 TORCS의 스펙은 다음과 같다.

표 1 상태정보

센서	설명
Angle	차량과 트랙의 각도
Track	차량 중심과 트랙 가장자리의 거리
TrackPos	트랙의 중심과 차량의 중심의 거리
SpeedX	차량의 정방향(x축) 속도
SpeedY	차량의 y축 속도
SpeedZ	차량의 z축 속도

표 2 행동정보

행동	범위	설명
Steer	[-1, 1]	조향 각도
Accel	[0, 1]	가속 페달
Brake	[0, 1]	브레이크 페달

표 1에 해당하는 상태 정보를 모두 포함시켰고, 출력은 표 2의 행동정보에서 Steer로 Accel은 0.8로, Brake는 0로 고정시킨 뒤 학습하였다.

3.2 Discretized PPO

그림 1 연속적인 행동 모델

기존의 연속적인 행동을 출력하는 강화학습 알고리즘은 그림 1과 같이 하나의 출력 노드로 액션을 표현하고 있으며, 기존의 TORCS 연구도 이와 마찬가지로 Steer의 범위, 즉 -1(-90도) ~ 1(90도)사이의 실수 값을 출력하도록 연구를 진행하였다. 이러한 모델은 직선구간에서 큰 문제를 보였다.

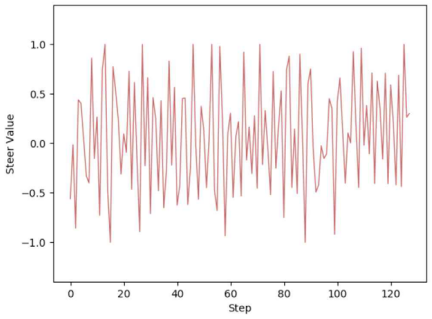

그림 2 기존 연속 행동 모델의 steer 값 변화

기존의 TORCS 연구들을 실험해본 결과 위 그림 2와 같이 전 구간(x축-step)에서 핸들을 좌우로 흔들면서(y축-steer 값) 주행을 하고 있다는 사실을 알 수 있었다.

본 논문에서는 이러한 연속적인 액션 범위를 이산화 하여 노드의 표현력을 높이고자 하였다. 이산화(Discretize)하는 액션 값의 산정은 기존의 Continuous PPO 알고리즘에서 많이 사용하는 액션 값을 산출하였다. 산출한 결과인 그림 3을 확인해보면, -0.25와 0.25 사이에서의 값이 최빈값 이었고, 이 값을 이용하여 그림 4와 같은 모델을 설계하였다.

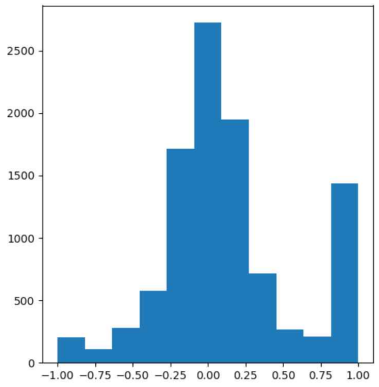

그림 3 기존 연속 행동 모델의 steer 범위 히스토그램

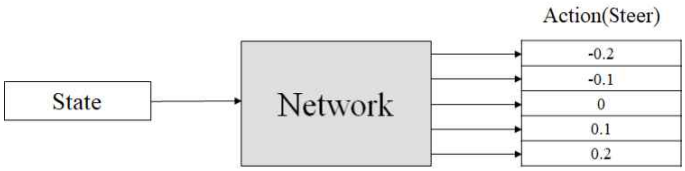

그림 4 Discretized PPO 모델

4. 실험 결과 및 향후 과제

그림 5는 기존의 강화학습 모델 중 하나인 Continuos PPO의 직선 구간 주행에서 Steer-핸들을 보여주는 그래프이다. 이 모델은 0도를 중심으로 좌우로 급격하게 꺾어가며 직선 주행을 하는 문제점을 확인할 수 있다.

그림 6은 본 논문에서 제시한 이산화한 PPO 모델의 직선 구간에서의 Steer-핸들 값을 보여준다. 위 그래프를 분석해보면 직선 주행을 할 때 사람의 운전과 비슷한 양상을 보인다. 사람의 운전은 핸들 각을 0도를 중심으로 주행을 하다가 차가 차도의 중심에서 조금 멀어졌을 때 핸들의 방향을 조절한다. 이와 마찬가지로 본 실험 결과는 0도를 기본으로 주행을 하다가 -0.2(-18도)~0.2(18도) 정도에서 움직여 주행 하는 결과를 보인다.

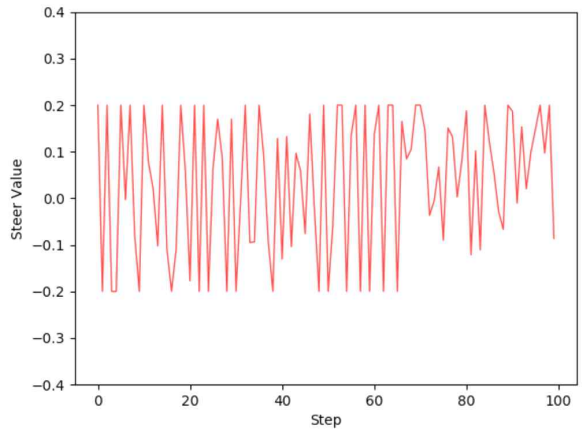

그림 5 Continuos PPO 모델의 steer 값 변화

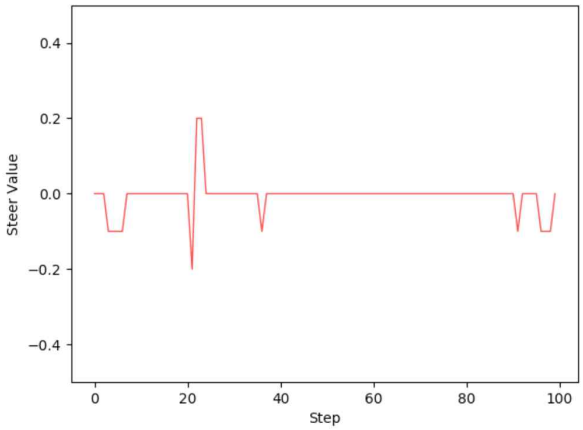

그림 6 Discretized PPO 모델의 steer 값 변화

본 연구에서는 연속적인 행동을 제어하는 기존 강화학습 알고리즘이 가지고 있는 문제점인 행동 표현 과부하에 대한 문제를 정의하고 해결책을 제시하였다.

제시한 연구 결과인 그림 6의 20~25 step을 확인해보면, -0.2(-18도)에서 0.2(18도)로 갑자기 핸들을 꺾는 결과를 확인할 수 있다. 이와 같은 문제는 기존 연구인 그림 5에서도 항상 발생하는 문제이다. 이러한 주행 컨트롤은 실제 주행 상황에서는 물리적으로 일어날 수 없는 일이며, 따라서 -0.2→-0.1→0→0.1→0.2와 같은 연속적인 주행 시퀀스가 필요하다. 본 연구의 후속 연구에서는 이러한 문제점을 개선하는 연구를 진행할 것이며, 이러한 연구는 실제 자율주행자동차를 안전하게 주행시키기 위한 필수적인 연구이다.

참고문헌

[1] Chen, Chenyi, et al. “Deepdriving: Learning affordance for direct perception in autonomous driving.“ 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, 2015.

[2] Shalev-Shwartz, Shai, Shaked Shammah, and Amnon Shashua. “Safe, multi-agent, reinforcement learning for autonomous driving.“ arXiv preprint arXiv:1610.03295 (2016).

[3] Lillicrap, Timothy P., et al. “Continuous control with deep reinforcement learning.“ arXiv preprint arXiv:1509.02971 (2015).

[4] Mnih, Volodymyr, et al. “Asynchronous methods for deep reinforcement learning.“ International conference on machine learning. 2016.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017

[6] Wymann, Bernhard, et al. “TORCS, the open racing car simulator.“ Software available at http://torcs. sourceforge. net 4 (2000): 6.