자율주행 환경에서 Unity-RL 을 이용한 심층강화학습 시각화 및 시뮬레이션에 관한 연구

이구상, 윤원준, 정소이, 김중헌
고려대학교 전기전자공학부

rntkd0917@korea.ac.kr, ywjoon95@korea.ac.kr, jungsoyi@korea.ac.kr, joongheon@korea.ac.kr

Deep Reinforcement Learning Visualization and Simulations using Unity-RL in an Autonomous Driving Environment

Gu Sang Lee, Won Joon Yun, Soyi Jung, Joongheon Kim
School of Electrical Engineering, Korea University, Seoul, Korea

요 약

본 논문은 자율주행 환경에서 Unity-RL 을 이용하여 강화학습의 시각화 및 시뮬레이션에 대해 설명한다. 강화학습에 사용된 정책을 최적화하기 위한 알고리즘은 Deep Q Network (DQN)이다. 학습방법은 Unity 에서 제공하는 ML-Agent 의 강화학습 환경을 이용하여 배치(Batch) 학습하였다. 4 개의 바퀴에 속도와 각속도가 Action 으로 적용되며 도로의 과 충돌하는 것을 방지하고 도로를 따라 이동하는 시뮬레이션을 구현을 통해 강화학습의 시각화를 보여준다.

Ⅰ. 서 론

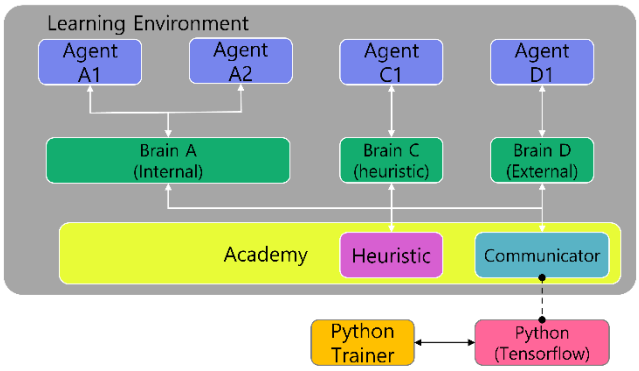
AI 와 6G 기술은 금융, 의료, 법률 서비스, 게임, 로봇 등 다양한 산업 분야에 이미 적용되어 널리 활용되고 있다. 그 중 자율주행은 인공지능에서 가장 빠르게 성장하고 있는 분야로 떠오르고 있다. [1] 이에 대한 연구로는 장애물 및 표지판 인식과 최적경로 이동 등이 있다 이러한 목표를 위하여 심층강화학습을 최적화 방법으로 적용하여, 해결하려는 노력이 존재한다.

그러나 자율주행 시뮬레이션을 구현하는데 정교한 환경이 주어지지 않아 이러한 강화학습을 이용하여 해결하는 것이 많은 노력이 소요된다. 하지만 오픈소스로 공개된 Unity-RL 을 이용한다면 손쉽게 자율주행 환경을 사용할 수 있다. 본 논문은 Unity-RL 을 이용하여 자율주행 환경을 구성하고, 충돌을 방지하여 주어진 길을 따라 이동하는 목적을 지닌 자율주행의 인공신경망을 구성하여 최적화하는 방법에 대해 다룬다. 본 논문이 기여하는 바는 다음과 같다.

1. Unity-RL 을 이용한 자율주행 환경 구성 및 작동 방법 소개
2. 경로 최적화 및 장애물 회피를 위한 심층강화학습을 위한 상태, 행동, 보상 소개

Ⅱ. 시스템 모델 설명

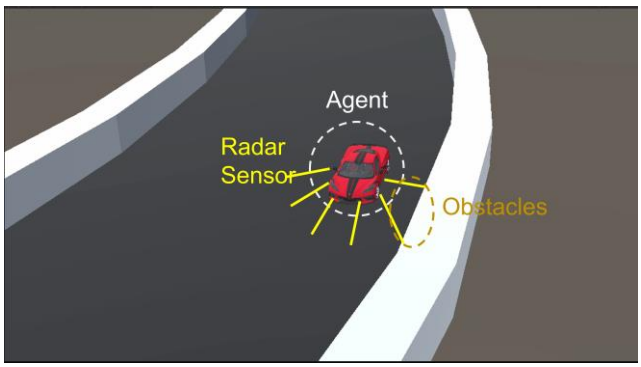
Unity-RL 과 심층강화학습 모델은 [그림 1]과 같이 나타낼 수 있다. [그림 1]의 A1 및 A2 는 Unity-RL 의 내부 학습 모델을 사용하여 학습 신경망과 연동을 하여 학습시키는 모듈이다. 이렇게 학습된 모델을 불러와서 행동을 추론할 수 있는 모듈은 C1 이다. 마지막으로 외부의 Python Trainer 를 이용하여 학습하는 모듈은 D1 에 해당한다. 자율주행 환경은 3D 디자인을 통해 얻을 수 있다.



[그림 1] 시스템 모델

Ⅲ. 자율주행 환경 설명

본 연구에서는 자율주행 환경은 Unity 내에서 길을 만들고 3D 디자인한 자동차모델을 이용하였다. [그림 2]는 자율주행 시작 환경이다. 자동차에 레이더 센서를 앞, 좌, 우 부분에 따로 Unity 내의 센서를 이용하여 충돌을 감지할 수 있게 구성하였다. 자동차는 에이전트이고, 매 회 에피소드의 초기 단계 (t=0)일 때 시작 지점에서 출발하여 도착 지점에 도달하였을 때, 방해물과 충돌하였을 때, 혹은 t=1000 일 때, 에피소드가 종료된다. 에이전트가 움직이는 것은 네 가지 모터의 속도와 각속도에 의해 방향과 속도가 결정된다. 또한 자동차의 초기 위치는 매 회 에피소드 마다 임의의 위치에 생성되도록 환경을 추가적으로 구성하였다. 에이전트의 상태는 현재의 위치, 현재의 속도, 현재의 각속도, 레이더 센서의 충돌감지 등을 받아서 심층강화학습 정책에 반영한다.



[그림 2] 자율주행 환경 예시

에이전트가 취할 수 있는 행동은 네 가지 바퀴에 대한 연속적인 속도와 각속도를 조절하는 것이다. 이를 위한 보상은 다음과 같이 구성된다.

1. 에피소드가 종료되기 전까지 충돌하지 않고 이동한 거리 (양의 보상)
2. 에피소드가 종료되기 전까지 충돌하면 보상 -1 이 주어짐 (음의 보상)
3. 에피소드가 종료되기 전까지 움직이지 않는다면 -1 이 주어짐 (음의 보상)

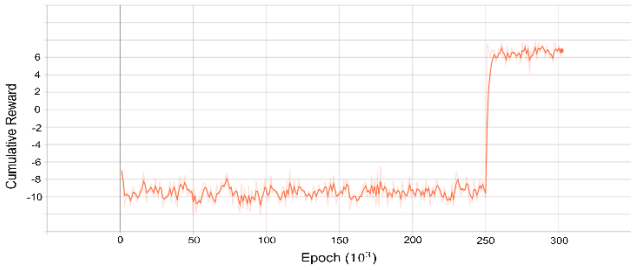
Ⅲ. 실험 및 실험 결과

i. 실험 환경 구성: DQN 을 이용하여 강화학습의 정책을 최적화하였고, 강화학습의 정책은 Dense 레이어 2 장과 레이어를 구성하는 노드의 개수는 32 개이다 [2]. 정책 탐사를 위해 epsilon-greedy 를 사용하였고, 초기 epsilon 은 1.0 로 구성하고 학습률 (learning rate)는 10^{-5} 로 구성하였다. 효율적인 학습을 위해 Multi Agents Parallel Processing 방법을 이용하여 배치(Batch) 학습하였다. 학습 실험은 Unity 20.1.17f1 버전과 mlagents 0.8 버전, 파이썬 3.7, tensorflow 2.0 버전으로 진행하였다 [3]. 총 학습은 30 만 회를 수행하였고 i7-9700k 3.60GHz @ 2, 64G RAM, GTX 1660 super 의 하드웨어 환경에서 실험을 진행하였다.

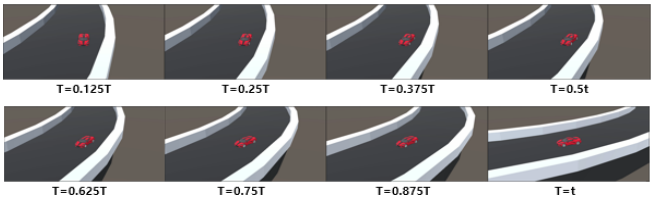
ii. 실험 결과: 학습 결과는 [그림 3]과 [그림 4]에서 확인할 수 있다. [그림 3]에서 보상 수렴성이 확인되었음을 확인할 수 있다. 보상은 학습 횟수 25 만 회부터 수렴성이 나타났고, 수렴성이 나타난 뒤로는 누적 보상에 대한 분산도 적게 나타났다. 더불어 [그림 4]에서는 한 에피소드 내, 자동차가 주어진 길을 따라 이동하는 모습을 시간에 대해 나타낸 것이다. t=0 은 에피소드 시작의 모습을 나타낸 것이다. 그리고 t=T 는 자동차가 완벽하게 코너의 벽을 인식하고 회피하여 길을 따라 움직이는 것을 볼 수 있다. $t \in [0,0.625T]$ 에서의 에이전트는 코너의 벽을 인식하여 에이전트의 각속도를 조절하는 행동을 내리고, $t \in [0.625T, T]$에서는 에이전트가 코너의 벽과 충돌을 피해 이동하는 행동을 취해 주어진 길을 따라 이동하는 모습을 확인할 수 있다.

Ⅳ. 결론 및 향후 연구 방향

Unity-RL 을 이용하여 환경을 불러오고, 또한 상태, 행동, 보상을 정의하여 학습하는 방법에 대해 소개하였다. 심층강화학습 정책은 DQN 를 사용함으로써 누적 보상의



[그림 3] 에피소드에 따른 학습 결과



[그림 4] 시간에 따른 학습 결과

분산을 줄이고 학습의 효율을 늘렸다. 그 결과 학습 25 만회에 보상이 수렴함을 알 수 있었고, 또한 에피소드 내 시간에 따른 학습 결과도 최적화가 진행되었음을 확인할 수 있었다.

향후 연구 방향은 다음과 같이 정리할 수 있다.

1. 현재 자동차 1 대만 나타나지만 여러 대의 자동차가 존재하고 서로를 인식하여 회피하거나 각자의 목적지에 맞는 차선 변경을 하는 연구를 할 수 있을 것이다.
2. CNN 을 기반으로 한 자율주행 자동차를 이용하여 실제 환경에서 장애물을 인식하고 목표 지점에 도달하는 연구를 할 수 있을 것이다.

ACKNOWLEDGMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2017-0-01637) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation and also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2019R1A2C4070663). J. Kim and J.-H. Kim are the corresponding authors of this paper

참 고 문 헌

[1Yu, April, Raphael Palefsky-Smith, and Rishi Bedi. "Deep reinforcement learning for simulated autonomous vehicle control." Course Project Reports: Winter 2016 (2016).

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing Atari with deep reinforcement learning," CoRR, vol. abs/1312.5602, July 2013

[3] Y.J. Mo, J. Kim, J.-K. Kim, A. Mohaisen, and W. Lee, "Performance of Deep Learning Computation with TensorFlow Software Library in GPU-Capable Multi-Core Computing Platforms," in *Proc. IEEE International Conference on Ubiquitous and Future Networks (ICUFN)*, Milan, Italy, July 2017.