Deep Deterministic Policy Gradient 알고리즘을 응용한 자전거의 자율 주행 제어★

최 승 윤*, Le Pham Tuyen**, 정 태 충***

요 약

DDPG(Deep Deterministic Policy Gradient)알고리즘은 인공신경망과 강화학습을 사용하여 학습하는 알고리즘이다. 최근 많은 연구가 이루어지고 있는 강화학습과 관련된 연구 중에서도 DDPG 알고리즘은 오프폴리시로 학습하기 때문에 잘못된 행동이 누적되어 학습에 영향을 미치는 경우를 방지하는 장점이 있다. 본 연구에서는 DDPG 알고리즘을 응용하여 자전거를 자율주행 하도록 제어하는 실험을 진행하였다. 다양한 환경을 설정하여 시뮬레이션을 진행하였고 실험을 통해서 사용된 방법이 시뮬레이션 상에서 안정적으로 동작함을 보였다.

Autonomous control of bicycle using Deep Deterministic Policy Gradient Algorithm

Choi Seung Yoon*, Le Pham Tuyen**, Chung Tae Choong***

ABSTRACT

The Deep Deterministic Policy Gradient (DDPG) algorithm is an algorithm that learns by using artificial neural networks and reinforcement learning. Among the studies related to reinforcement learning, which has been recently studied, the DDPG algorithm has an advantage of preventing the cases where the wrong actions are accumulated and affecting the learning because it is learned by the off-policy. In this study, we experimented to control the bicycle autonomously by applying the DDPG algorithm. Simulation was carried out by setting various environments and it was shown that the method used in the experiment works stably on the simulation.

Key words : Balancing Control, DDPG, Reinforcement Learning, Autonomous Driving Control

접수일(2018년 3월 9일), 게재확정일(2018년 9월 13일)

★ 본 논문은 한국과학재단 기초연구비 지원에 의하여 연구되었음. (NRF-2017R1D1A1B04036354)

* 경희대학교/컴퓨터공학과
** 경희대학교/컴퓨터공학과
*** 경희대학교/컴퓨터공학과(교신저자)

1. 서 론

기계학습 분야에서 강화학습에 관련된 연구가 최근 많이 이루어져 왔다. 강화학습은 다른 기계학습의 방법과는 특징이 다른데 보상을 통해서 학습을 하게 된다. 강화학습의 장점은 환경에 대한 사전 지식이 없어도 학습이 가능하다는 것이다. 강화학습은 순차적으로 결정을 내려야 하는 문제에 적용할 수 있는데 최근 여러 연구를 통하여 인공신경망을 결합하는 형태를 통하여 거대한 규모의 문제를 해결하는 성능을 보여 주목을 받고 있다.[1][2][3][21] 고전적 강화학습은 한정된 양의 상태를 가진 환경에 대해서만 학습이 가능한데 인공신경망을 사용하게 되면 방대한 양의 상태를 가진 환경에 대해서도 학습하는 것이 가능하기 때문이다. 인간만이 할수 있다고 여겨지는 영역들이 강화학습 기반의 알고리즘을 통해 에이전트가 해결하도록 하고 있는데 대표적 영역으로 바둑 등을 들 수가 있다.[4][5][6][7] 기존의 연구 중 에이전트를 학습시켜 자전거를 운행하도록 하는 연구가 있었는데 학습은 가능하지만 최적 정책을 찾지 못하는 경우가 발생할 수 있는 단점이 있었다. 부정적인 방향으로 학습이 진행되는 것은 사용자의 의도에 부합하지 않기 때문에 이러한 단점을 극복하려는 많은 연구가 있었다. 또한, 이러한 연구들은 자전거를 제어에 적용하였을 때 에이전트를 통하여 균형 제어를 포함하여 자율적 주행이 가능하도록 할 수 있을 것이다.[8][9][10][11]

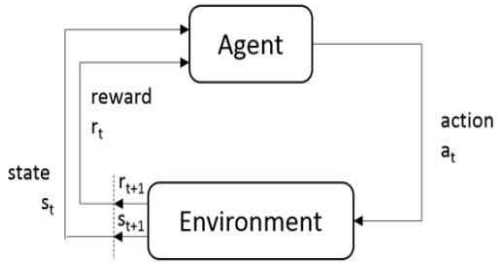
본 논문에서는 신경망을 사용한 강화학습 방법 중 DDPG(Deep Deterministic Policy Gradient)를 사용하여 자전거의 자율 주행에 대해 시뮬레이션 하고자 한다. 이를 통해서 현실세계의 특징을 반영하여 에이전트가 연속되는 순차적 결정 문제를 해결할 수 있는지에 대해 알아보고 문제 해결에 대한 접근법에 대해 알아보고자 한다.

논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 설명하고 3장에서는 시뮬레이션 환경에 대해 설명한다. 4장에서는 실험의 결과를 설명하고 5장에서는 결론을 설명한다.

2. 관련 연구

2.1 강화학습

강화란 좋은 행동을 점점 더 많이 하는 것을 말하며 이런 강화의 개념을 컴퓨터의 학습에 접목시킨 것이 강화학습이라고 할 수 있다. 강화학습에서는 에이전트를 통해서 환경과 상호작용하며 학습하는데 이때 에이전트는 환경이 주는 정보인 보상을 통해서 어떤 행동을 더 해야 할지 알 수 있고 사전지식이 없어도 학습할 수 있다.



(그림 1)강화학습의 구성

그림 1을 보면 환경과 에이전트가 어떠한 관계를 가지고 있는지 알 수 있다. 에이전트는 상태와 행동을 가지며 환경이 주는 보상 값을 통하여 행동에 대한 가치를 평가하게 된다. 이를 통해서 행동을 개선시켜 나갈 수 있다.[12][13][14]

2.2 MDP(Markov Decision Problem)

강화학습을 통하여 풀고자 하는 것은 순차적으로 행동을 결정해야 하는 문제이다. 순차적 행동 결정 문제는 MDP(Markov Decision Problem)을 통하여 정의할 수 있다. MDP는 상태, 행동, 보상함수, 상태변환 확률, 감가율, 정책으로 구성된다. 순차적 행동 결정 문제를 푼다는 것은 결국 더 좋은 정책을 찾는 것을 의미하는데 에이전트는 어떤 정책이 더 좋은 정책인지를 판단할 때 가치함수를 사용한다. 가치함수의 정의는 다음과 같다.

$$(s) = \ _{\pi}[R_{t+1} + \gamma v_{\pi}(S_{t+1}) \ \ S_t = s] \quad (1)$$

위 수식을 통하여 구하고자 하는 것은 현재 상태로부터 정책을 따라갔을 때 받을 것이라고 예상되는 보상의 합이다. 에이전트는 정책을 업데이트 할 때 가치함수를 사용하는데 보통 가치함수보다 에이전트가 선택할 각 행동의 가치를 직접적으로 나타내는 큐함수를 사용한다. 큐함수의 정의는 다음과 같다.

$$(s,a) = \ _{\pi}[R_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) \ \ S_t = s, A_t = a] \quad (2)$$

더 좋은 정책을 찾는 과정을 반복하면 결국 최적의 정책을 찾을 수가 있다. 최적 정책은 최적 가치함수를 받게 하는 정책이며 이때 가치함수 사이의 관계식은 다음과 같이 정의할 수 있다.[15][16][17]

$$v_*(s) = \max_a E[R_{t+1}\gamma v_*(S_{t+1} \ \ S_t = s, A_t = a] \quad (3)$$

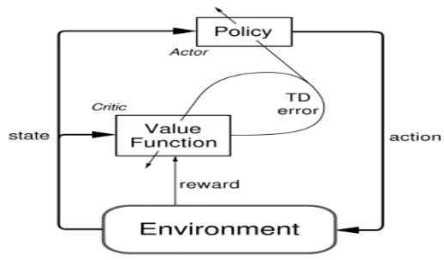
2.3 액터 크리틱

학습을 에피소드 마다만 학습하는 경우 에피소드가 길어지면 상태 $s,a)$에 대한 반환값의 변화가 커지기 때문에 분산이 커지게 된다. 이 경우 학습이 느려지는 단점이 있다. 액터 크리틱 알고리즘은 인공신경망을 사용해 큐 함수를 근사하는 경우 매 타임스텝마다 학습할 수 있는 장점이 있다. 행동을 선택하는 인공신경망이 액터가 되며 큐 함수를 근사하며 행동에 대한 판단을 하는 인공신경망은 크리틱이 된다. 액터와 크리틱의 업데이트 식은 다음과 같이 표현된다.[18][19]

$$\theta_{+1} \quad \theta_t + \alpha[\quad _{\theta}\log\pi_{\theta}(a \ s)\delta \ ] \quad (4)$$

$$SE = (R_{t+1} + \gamma V_v(S_{t+1}) - V_v(S_t))^2 \quad (5)$$

액터 크리틱 알고리즘은 두 개의 신경망을 가지며 각각 정책 신경망과 가치 신경망에 해당한다. 액터 크리틱의 모델 구성은 다음과 같다.[20]



(그림 2)액터 크리틱 알고리즘 구성

3. DDPG를 사용한 자전거의 자율주행 제어

3.1 DDPG(Deep Deterministic Policy Gradient)

본 논문에서는 DDPG 알고리즘[21]을 사용하여 자전거를 에이전트가 제어하도록 하고 자율주행이 가능한지에 대하여 시뮬레이션한다.. 자전거를 제어하기 위한 DDPG 알고리즘의 구성은 다음과 같다.

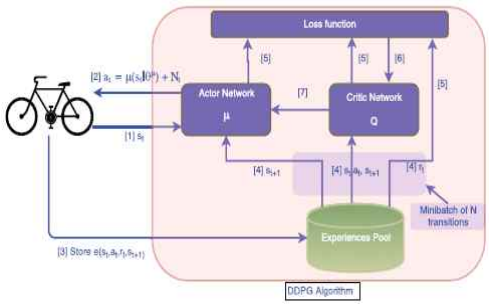
[1] 자전거는 상태 를 관찰한 후 액터 네트워크로 넘겨준다.
[2] 액터 네트워크는 입력값으로 s_t를 받아서 액션을 출력으로 얻는다.
[3] 자전거는 보상 r_t과 다음 상태 s_t+1을 관찰한다. 이 후 $s_{t,}a_{t,}r_{t,}s_{t+1}$의 튜플을 저장한다.
[4] 저장한 튜플 데이터에서 N개의 튜플을 무작위로 선택하여 정책을 학습하는데 사용한다.
[5] 다음의 식에 따라 손실 함수를 계산한다.

$$= \frac{1}{N}_{\ i} \quad (y_i - Q(s_i, a_i|\theta \ ))^2$$

[6] 손실 L을 최소화하도록 크리틱 네트워크를 업데이트 한다.
[7] Deterministic Policy Gradient를 사용하여 액터 네트워크를 업데이트 한다.[11]

알고리즘 1. 자전거 제어를 위한 DDPG 알고리즘

제안된 알고리즘은 표1과 같이 Actor 모듈과 Critic 모듈을 사용한다. Actor 모듈은 신경망을 통해서 행동에 대한 결과 값을 얻을 수 있고 Critic 모듈은 Actor 모듈을 통해 얻은 행동의 결과 값에 대한 행동 가치를 평가한 후 행동 가치가 높은 행동은 좋은 행동으로 판단하여 동일한 행동을 반복할 수 있도록 Actor 모듈의 정책을 업데이트 한다. 또한 이 과정에서 가치가 낮은 행동은 반복하지 않도록 정책이 업데이트 된다. DDPG 알고리즘 최대한 샘플을 많이 확보하는 것이 학습에 유리하기 때문에 샘플을 저장하기 위해 리플레이 메모리를 사용한다.[15][16][17][18] DDPG 알고리즘을 사용하여 자전거를 제어하기 위한 구성은 그림 3과 같다. 구성을 보면 학습을 위해서 두 개의 네트워크로 구성한 것을 확인할 수 있다. 각각 액터 네트워크와 크리틱 네트워크이며 에이전트는 두 개의 신경망을 제어하고 결과 값을 통해서 자전거를 제어하게 된다.



(그림 3) DDPG를 사용한 자전거 제어를 위한 구성

4. 실험 및 결과

본 논문에서는 자전거의 주행제어를 실험하기 위하여 두 가지 측면을 고려하였다. 첫 번째는 에이전트가 넘어지지 않고 목표지점에 도달했는지이고 두 번째는 균형 제어를 기반으로 목표지점을 무작위로 주는 경우에도 쓰러지지 않고 자율적으로 주행을 하는가 이다. 이를 위해서 첫 번째로 자전거의 속도를 각각 3km, 10km로 지정하고 목표를 고정하는 경우에 대해 실험하였다. 두 번째로 속도와 목표지점을 무작위로 지정한 후 상황에 따라 에이전트가 속도를 조절하며 주행하는지도 실험을 진행하였다.

실험에서는 여러 보상 함수를 통하여 학습 과정을 평가한다. 보상 함수마다 자전거의 수렴 속도와 궤적이 다르게 측정된다. 먼저 다음의 보상 알고리즘으로 학습 알고리즘을 평가한다.

$$(s,a) = \begin{cases} 0 & |\omega| > \frac{\pi}{6} \\ 1 & |\omega| < \frac{\pi}{6} \end{cases} \quad (6)$$

이 알고리즘은 자전거가 균형 상태라면 1을 쓰러진 경우 0을 반환한다. 따라서 자전거가 오랜 시간동안 균형을 잡을 수 있다면 더 많은 보상을 받을 수 있다. 두 번째로 평가된 보상 함수는 다음과 같다.

$$r(s,a) = \begin{cases} 0 & |\omega| > \frac{\pi}{6} \\ -(\omega + 0.1\dot{\omega}^2 + 0.01\ddot{\omega}^2) & |\omega| < \frac{\pi}{6} \end{cases} \quad (7)$$

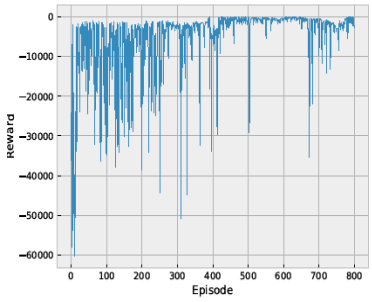
$\dot{\omega}, \ddot{\omega}$의 값이 작으면 보상이 높을 것이다. 이 경우 자전거가 장시간 바로선 자세를 유지할 수 있다면 높은 보상을 받게 된다. 세 번째 평가된 보상 함수에서 크기에 가장 영향을 미치는 매개변수는 $\ddot{\omega}$이다.

$$r(s,a) = \begin{cases} 0 & |\omega| > \frac{\pi}{6} \\ -(0.01\omega^2 + 0.1\dot{\omega}^2 + \ddot{\omega}^2) & |\omega| < \frac{\pi}{6} \end{cases} \quad (8)$$

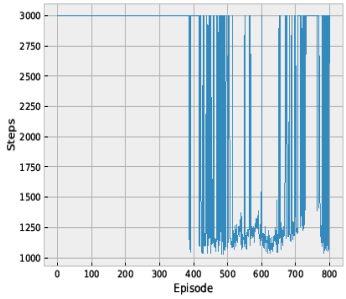
세 번째 보상함수에서 자전거가 갑자기 가속도를 변경하지 않는 경우 높은 보상을 받게 된다.

4.1 목표지점을 고정하는 경우의 결과

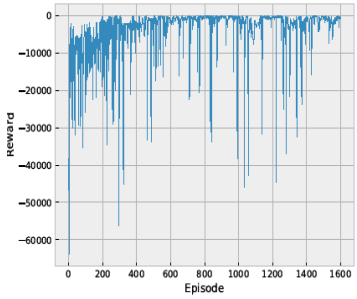
먼저 목표지점을 고정한 후 속도를 각 각 3km, 10km로 다르게 설정한 후에 실험을 진행하였다.



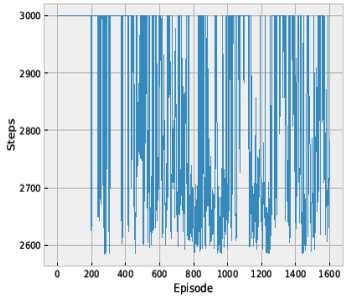
(그림 5) 시속 3km에서 보상 값 측정



(그림 6) 시속 3km에서 에피소드 측정 결과



(그림 7) 시속 10km에서 보상 값 측정

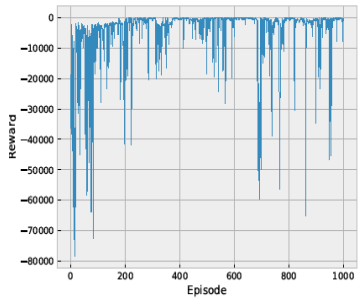


(그림 8) 시속 10km에서 에피소드 측정 결과

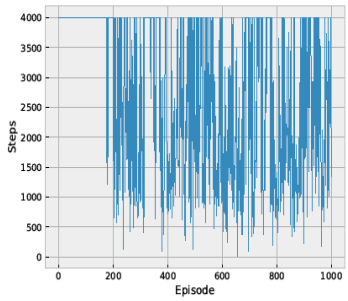
실험 결과를 보면 제어를 위한 학습이 시속을 3 km로 설정한 경우 800회 에피소드에 보상값이 충분히 증가하는 것을 확인할 수 있다. 10km의 경우에는 1600회 진행되었을 때 보상값이 충분해지는 것을 확인할 수 있다.

4.2 속도와 목표지점을 무작위로 설정한 경우의 결과

시작 지점만 지정한 후 속도와 목표지점을 무작위로 설정한 경우의 결과는 다음과 같다.



(그림 9) 속도와 목표지점이 무작위인 경우 보상값 측정 결과



(그림 10) 속도와 목표지점이 무작위인 경우 에피소드 측정 결과

속도를 무작위로 하는 경우 에이전트가 상황에 따라 균형을 잡아야 하므로 속도를 조절하게 된다. 주행 결과는 속도를 고정하였을 경우와 비슷

하지만 에피소드가 진행됨에 따라 쓰러짐 없이 주행한 것을 확인할 수 있다.

5. 결론

본 논문에서는 강화학습을 통해 에이전트가 자전거를 자율주행 하도록 제어하도록 실험을 통해 확인하였다. 알고리즘으로는 DDPG(Deep Deterministic Policy Gradient) 알고리즘을 사용하였다. DDPG 알고리즘은 오프폴리시로 학습하기 때문에 신경망을 통해서 학습하는 데이터간의 연관성을 분리시킬 수가 있고 따라서 최적 정책을 학습시킬 확률이 높아지게 된다. 실험을 통해서 제안한 알고리즘을 사용하여 에피소드가 진행됨에 따라 보상 함수가 안정적으로 동작하고 이를 바탕으로 학습이 잘 이루어져 균형 및 주어진 환경을 제어하는 것을 확인할 수 있었다.

참고문헌

[1] Herlihy, David V. Bicycle: the history. Yale University Press, 2004.

[2] Schwab, A. L., J. P. Meijaard, and J. D.G. Kooijman. "Some recent developments in bicycle dynamics." Proceedings of the 12th World Congress in Mechanism and Machine Science. 2007.

[3] Meijaard, Jaap P., et al. "Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review." Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. Vol. 463. No. 2084. The Royal Society, 2007.

[4] http://ai2001.ifdef.jp/primer_V2/primer_V2.html

[5] Basso, Michele, and Giacomo Innocenti. "Lego bike: A challenging robotic lab project to illustrate rapid prototyping in the mindstorms/simulink integrated platform." Computer Applications in Engineering Education 23.6 (2015): 947-958.

[6] Basso, Michele, Giacomo Innocenti, and Alberto Rosa. "Simulink meets lego: Rapid controller prototyping of a stabilized bicycle model." 52nd IEEE Conference on Decisionand Control. IEEE, 2013.

[7] Randløv, Jette, and Preben Alstrøm. "Learning to Drive a Bicycle Using Reinforcement Learning and Shaping." ICML. Vol. 98. 1998.

[8] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. Vol.1, No.1, Cambridge: MIT press, 1998.

[9] Lagoudakis, Michail G., and Ronald Parr. "Model-free least-squares policy iteration." NIPS, Vol.14, 2001.

[10] Lever, Guy. "Deterministic policy gradient algorithms.", 2014.

[11] Phyo Htet Kyaw, Dyna-Q based Univector Field Obstacle Avoidance for Fast Mobile Robots, Master, KyungHee University, Korea, Seoul, 2011.

[12] Kaelbling, Leslie Pack, Michael L. Littman, and Andrew W. Moore. "Reinforcement learning: A survey." Journal of artificial intelligence research 4 (1996): 237-285.

[13] Irodova, Marina, and Robert H. Sloan. "Reinforcement Learning and Function Approximation." FLAIRS Conference. 2005.

[14] Watkins, Christopher JCH, and Peter Dayan. "Q-learning." Machine learning 8.3-4 (1992): 279-292.

[15] G.A. Rummery and M. Niranjan, On-Line Q-Learning Using Connectionist Systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department, 1994.

[16] Grondman, Ivo, et al. "A survey of actor-critic reinforcement learning: Standard and natural policy gradients." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42.6 (2012): 1291-1307.

[17] Sutton, Richard S., et al. "Policy Gradient Methods for Reinforcement Learning with Function Approximation." NIPS. Vol. 99. 1999.

[18] Peters, Jan, and Stefan Schaal. "Policy gradient methods for robotics." 2006 IEEE/RSJ International Conference on

Intelligent Robots and Systems. IEEE, 2006.
[19] Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning." arXiv preprint arXiv:1509.02971 (2015).
[20] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529-533.
[21] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
[22] Kim Tae Hee, Kang Seung Ho, "An Intrusion Detection System based on the Artificial Neural Network for Real Time Detection." Journal of Information and Security. 2018.

〔저자소개〕

최 승 윤 (Seung-yoon Choi)
2010년 8월 나사렛대학교 정보통신 학과(이학사)
2012년 8월 경희대학교 컴퓨터공학과 (공학석사)
2012월 9월~현재 경희대학교 컴퓨터공학과 박사과정
관심분야: 기계학습, 강화학습, 로보틱스
email : sychoi84@khu.ac.kr

레 팜 투옌 (Tuyen P. Le)
2013년 2월 HCMC 대학교 컴퓨터과학과(학사)
2014년 3월~현재 경희대학교 컴퓨터공학과 석박통합과정
관심분야: 기계학습, 강화학습, 로보틱스
email : tuyenple@khu.ac.kr

정 태 충 (Tae Choong Chung)
1987년 2월 KAIST 전산학과 (공학박사)
1987년 KIST 시스템공학센터 선임 연구원
1988년~현재 경희대학교 컴퓨터공학과 교수
관심분야: Machine Learning, Meta Search and Robotics
email : tcchung@khu.ac.kr