JPEE 실천공학교육논문지
JOURNAL OF PRACTICAL ENGINEERING EDUCATION

실천공학교육 관련
J. Pract. Eng. Educ. 14(1), 219-224, 2022

에이전트 학습 속도 향상을 위한 Q-Learning 정책 설계

Q-Learning Policy Design to Speed Up Agent Training

용성중*, 박효경, 유연휘, 문일영

한국기술교육대학교 컴퓨터공학과

Sung-jung Yong*, Hyo-gyeong Park, Yeon-hwi You, Il-young Moon

Department of Computer Science and Engineering, Korea University of Technology and Education, Cheonan 31253, Korea

[요 약]

강화학습의 기본적인 알고리즘으로 많이 사용되고 있는 Q-Learning은 현재 상태에서 취할 수 있는 행동의 보상 중 가장 큰 값을 선택하는 Greedy action을 통해 보상을 최대화하는 방향으로 에이전트를 학습시키는 기법이다. 본 논문에서는 Frozen Lake 8*8 그리드 환경에서 Q-Learning을 사용하여 에이전트의 학습 속도를 높일 수 있는 정책에 관하여 연구하였다. 또한, Q-learning의 기존 알고리즘과 에이전트의 행동에 '방향성'이라는 속성을 부여한 알고리즘의 학습 결과 비교를 진행하였다. 결과적으로, 본 논문에서 제안한 Q-Learning 정책이 통상적인 알고리즘보다 정확도와 학습 속도 모두 크게 높일 수 있는 것을 분석되었다.

[Abstract]

Q-Learning is a technique widely used as a basic algorithm for reinforcement learning. Q-Learning trains the agent in the direction of maximizing the reward through the greedy action that selects the largest value among the rewards of the actions that can be taken in the current state. In this paper, we studied a policy that can speed up agent training using Q-Learning in Frozen Lake 8×8 grid environment. In addition, the training results of the existing algorithm of Q-learning and the algorithm that gave the attribute 'direction' to agent movement were compared. As a result, it was analyzed that the Q-Learning policy proposed in this paper can significantly increase both the accuracy and training speed compared to the general algorithm.

Key Words: OpenAI Gym, Q-Learning, Reinforcement Learning, Reward Policy, Training

http://dx.doi.org/10.14702/JPEE.2022.219

Received 23 March 2022; **Revised** 11 April 2022
Accepted 11 April 2022

***Corresponding Author**
E-mail: zeros952@koreatech.ac.kr

I. 서 론

인공지능 분야에서 에이전트를 학습시키는 것은 매우 중요하다. 학습이 어떻게 이루어지는지에 따라 인공지능의 성능이 달라지고, 학습에는 많은 시간을 필요로 한다. 강화학습 기법 중 Q-Learning은 현재 상태에 대해서 최상의 보상 값을 향해 이동하는 기법으로, 행동에 의한 보상과 처벌을 통해 최적의 행동을 구분한다. 본 논문에서는 Frozen Lake 환경에서 Q-Learning 정책을 변경하는 것을 통해서 에이전트의 학습 속도를 높여 시간을 단축하고자 한다. 마지막에는 기존 정책과 학습 속도 및 정확도에 대해 비교 및 평가하여 제시된 방법의 효용성을 확인하고자 한다.

II. 관련 기술 및 연구

A. 강화학습

강화학습(Reinforcement Learning)은 에이전트가 환경으로부터 더 많은 보상을 방식으로 학습하는 과정을 의미한다. 강화학습의 일반적인 모델 구조의 구성 요소는 그림 1과 같으며 에이전트(Agent), 환경(Environment), 행동(Action), 보상(Reward), 상태(State), 정책(Policy)이 포함된다. 에이전트는 강화학습에서 학습하는 주체로, 주어진 문제 상황에서 행동을 수행한다. 환경은 에이전트가 직접 상호 작용하는 대상으로서, 에이전트의 행동을 입력 받아 처리하며 보상과 다음 단계의 정보를 반환한다. 행동은 에이전트가 환경에 전달하는 입력 정보이며, 행동을 수행하여 환경과 상호작용한다. 상태는 에이전트 스스로 관리하는 환경의 상태 정보를 의미한다. 보상은 에이전트가 수행한 행동에 대하여 환경이 에이전트에게 전달하는 값이며, 에이전트의 정책에 영향을 미치는 정보이다. 상태는 에이전트 스스로 관리하는 환경의 상태 정보이다. 정책이란, 에이전트가 주어진 상태에서 어떤 행동을 수행해야 하는지 결정하는 방식을 의미한다.

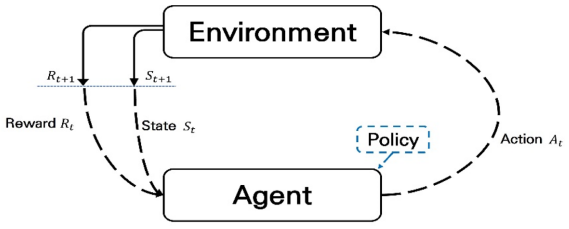

그림 1. 강화학습 프레임워크

Fig. 1. Reinforcement Learning Framework.

보상 함수와 정책을 통해 학습 문제의 목표를 정의하고, 행동하게 된다. 에이전트는 타임 스텝 t마다 에이전트는 환경으로부터 보상(R_t)와 상태(S_t) 정보를 받는다. 에이전트가 지닌 모델에 상태(S_t) 정보를 입력 값으로 넣어 행동(A_t) 출력 값을 받는다. 이를 환경에 보내고 해당 행동을 수행하게 된다. 행동 A_t가 수행된 이후 환경은 다시 새로운 보상 R_{t+1}를 에이전트에게 전달한다. 에이전트는 곧바로 새로운 상태 S_{t+1}를 구성하고 이를 활용하여 동일한 동작 과정을 반복한다[1,2].

B. Q-Learning

Q-Learning은 대표적인 강화학습 알고리즘으로, 특정 상태에서 어떤 결정을 내리는 것이 미래의 보상을 가장 높일 수 있을 것인지에 대한 정책 데이터를 지속해서 업데이트한다. 업데이트한 값을 통해 어떤 상태로부터 어떤 행동을 취할지 결정하는 것이다.

해당 알고리즘은 마르코프 결정 과정(MDP) 이론을 기반으로 한다[3]. 또한, 수식 (1)과 같이 입실론 탐욕적 정책(ε-Greedy Policy)을 사용하여 특정 상태에서의 확률로 무작위 행동을 취하고, (1–ε)의 확률로 제일 높은 Q 값을 가지는 탐욕적인 행동(greedy action)을 취한다. 각 타임 스탬프 t에서 에이전트는 상태(S_t)에서 행동(a_t)을 수행한다. 수행 후, 새로운 상태(S_{t+1})로 전이되며, 이를 통해 보상(R_t)을 획득한다. 이 과정을 이전의 값과 새 정보의 가중 합을 이용해 지속한다. 이때, 학습된 행동 가치 함수 Q는 자신이 따르는 정책에 상관없이 최적 행동 가치 함수 q^*를 직접적으로 근사한다.

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \quad (1)$$

where $Q : S \times A \rightarrow \mathbb{R}$, 각 상태-행동 쌍에 대한 함수

S_t: 현재 상태

a_t: 현재 수행하는 행동

α: 학습 속도율

R_{t+1}:다음 단계의 보상

γ: 감가율

C. DQN

DQN은 Deep Q-Network의 약자로, 강화학습과 딥 러닝의 인공신경망(CNN)을 합친 것이라고 정의할 수 있다. 기존 강화학습은 이미지나 자연어와 같은 고차원 데이터(High-dimensional data)에 적용하기 매우 어려웠다. 2013년, Google의 Deep Mind 사에서 Atari 게임을 통해 실험한 결과와 DQN

기술을 공개함으로써, 고차원 데이터가 인가되었을 때, 에이전트의 통제 정책을 효과적으로 학습시킬 수 있게 되었다[4].

이 DQN은 Experience Replay를 통해 각 타임 스탬프 별로 얻은 샘플을 수식 (2)과 같이 시계열대 순으로 유한한 크기의 데이터 세트에 저장한다. 그 후, 학습에 쓰일 샘플을 무작위로 추출하여 미니 배치를 구성하고, 파라미터를 학습시키게 된다. 이를 통해, 데이터 샘플을 일회성으로 업데이트한 후 제거하는 방법과 달리, 재활용함으로써 데이터를 효율적으로 사용할 수 있다.

$$e_t(s_t, a_t, r_t, s_{t+1}),\ D = e_1, e_2, e_3, \ldots, e_N \tag{2}$$

where e_t : 경험 정보

(s_t, a_t, r_t, s_{t+1}) : 환경과 상호작용하여 얻은 샘플

D: 한정된 크기의 데이터셋

벨만 방정식에 따라, DQN의 가중치를 갱신하기 위해 설정한 손실 함수 $L_i(\theta_i)$은 아래의 수식 (3)과 같다[5]. 이때, i는 반복 횟수, $p(s, a)$는 확률 분포, y_i는 목표(target)를 의미한다.

$$L_i(\theta_i) = \mathop{\mathbb{E}}_{s,a\sim p(\cdot)}\left[\left(y_i - Q(s,a;\ \theta_i)\right)^2\right] \tag{3}$$

where $y_i = \gamma \max_{a'} Q(s', a';\ \theta_{i-1})$, 정답값, 목표

$L_i(\theta_i)$: 손실함수

i : 반복횟수

$p(s, a)$: 확률분포

$Q(s, a; \theta_{i-1})$: 예측값

θ : 가중치

목표값을 계산하기 위한 Target Network는 DQN과 똑같은 NN(Neural Network)를 하나 더 만들어, 가중치 값이 업데이트되는 방식을 설계한다. 이를 통해, 학습 도중 변경된 가중치 값으로 목표 Q 값의 근사치(Approximator)가 어떤 값으로 수렴해야 하는지 파악할 수 있는 특징이 있다.

III. 연구방법

A. 연구환경

본 논문은 강화학습 라이브러리 Open AI Gym을 이용하여 Q-Learning 보상에 대한 설계를 제안하고 시뮬레이션을 진행하였다. Open AI Gym이란 강화학습 알고리즘 학습을 위해 Open AI에서 만든 라이브러리 패키지이다. Open AI Gym 라이브러리는 강화학습 알고리즘을 개발하고, 훈련을 수행할 수 있는 에이전트와 환경을 제공한다[6]. Open AI Gym에 있는 다양한 환경 중 Frozen Lake 환경은 에이전트가 출발 지점부터 도착 지점까지 함정에 빠지지 않고 이동하도록 구성된 환경이다. 강화학습을 이용한 효율적인 경로 선택 시뮬레이션으로 많은 연구자가 사용하는 환경으로 본 논문에서도 제안된 알고리즘을 평가하기 위해 선택하게 되었다.

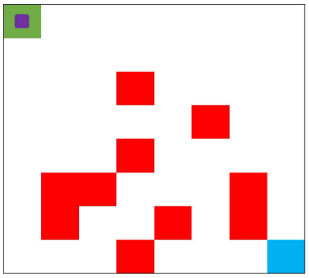

그림 2. Frozen Lake 시뮬레이션 환경

Fig. 2. Frozen lake Simulation Environment.

그림 2와 같이 8×8 Frozen Lake 환경은 8×8 그리드 환경에서 출발지점, 도착지점 그리고 총 10개의 함정으로 구성되어 있다. 초기에 모든 그리드에서 행동에 대한 보상 값이 0으로 설정되어 있어, 에이전트는 모든 행동을 랜덤으로 취하게 된다. 우리는 학습을 통해 행동에 대한 보상 값을 정해줌으로써 에이전트가 현재 상태에서 취할 수 있는 최적의 방향을 제시한다.

B. Q-Learning 정책 제안

1) 제안된 정책 : 발생 확률 감소

Frozen Lake 환경에서 에이전트를 학습하는 시뮬레이션을 살펴보면, 다음 상태에서 위치 값이 바뀌지 않는 경우가 많다. 예를 들어, 시작 지점(S)에서 좌(←) 또는 상(↑)의 행동 값이 나오게 되면, 해당 위치에 얼은 면(F)이 없기 때문에 상태가 변하지 않고 그대로 유지된다. 이러한 경우는 에이전트가 최상단, 좌측, 우측, 하단(벽면)에 위치했을 때, 확률적으로 발생하게 된다. 이러한 경우가 연속적으로 발생했을 때, 시도 횟수를 소모하면서 해당 회차가 엉뚱한 위치에서 종료(실패)하게 된다. 본 논문에서는 이전 상태와 이후 상태가 동일한 위치 값이 발생할 확률을 줄이기 위해 벽면에 부딪히게 되면, 해당 행동에 대해 처벌 값을 부여하는 방법을 수식 (4)와 같이 제안한다.

$$if\,(s_t = s_{t+1}),\ Q(s_t, a_t) = -1 \quad (4)$$

where s_t : t시점에서의 상태

s_{t+1} : t+1시점에서의 상태

a_t : t 시점에서의 행동

상태 s_t와 행동을 취한 결과의 상태(s_{t+1})가 같다면, 해당 행동(a_t)에 대해 처벌 값을 영구적으로 부여하여 다음 에피소드 진행 시 같은 실수를 반복하지 않게 되어, 학습 속도가 올라가게 된다.

2) 제안된 정책 : 방향성 부여

본 논문에서는 Frozen Lake 환경에 '방향성'이라는 개념을 추가하여 학습 성공률을 높이고자 한다. Frozen Lake 시뮬레이션은 모든 행동을 무작위로 설정하기 때문에 행동에 대한 방향성이 존재하지 않는다. 아래와 같이, 에이전트의 행동에 방향성을 부여하기 위해 이전에 왔던 길을 되돌아갈 수 없게, 일시적으로 처벌 값을 부여하는 방안을 제안한다. 제안된 알고리즘의 에이전트 이동 순서는 다음과 같다.

• 제안된 알고리즘의 에이전트 이동 순서

가) $Q(s_t, x_t)$ = -1

■ where x_t : t 시점에서 s_{t-1}로 돌아가기 위한 행동

나) Q-Learning 정책에 따른 에이전트 이동

다) $Q(s_t, x_t)$= 0

현재 상태 s_t에서 행동 a_t를 진행할 때, 현재 상태(s_t)에서 이전 상태(s_{t-1})로 가는 행동(x_t)에 대해 처벌 값을 부여하여 이전 상태로 돌아가지 못하게 만든 후 에이전트가 이동한다. 이동이 완료된 후, 이전에 주었던 $Q(s_t, x_t)$에 대한 처벌 값을 0으로 복구한다. 이러한 방법을 통해 일시적으로 처벌 값을 부여함으로써 에이전트의 이동에 방향성을 부여할 수 있다.

IV. 연구 결과

A. 기존 Q-Learning 알고리즘 시뮬레이션 결과

에이전트가 장애물에 도달 시 해당 행동(action)에 처벌 값을 주는 알고리즘에 대해 학습을 진행하였다. 그림 3과 같이 8×8 그리드 환경에서 100에피소드의 학습을 진행하였다. 학습이 일정 이상 진행이 되면, 에이전트가 정해진 루트를 통해 단순히 이동하므로, 에피소드에 100회의 제한을 두었다. 그림 4와 같이 기존 알고리즘의 학습 결과 정확도 52%, 학습 시간 01:46.0836s가 소모되었다.

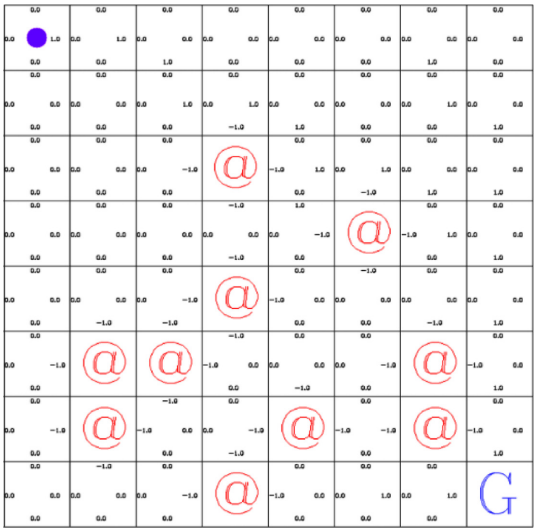

그림 3. Q-Learning 장애물 보상 시뮬레이션 결과

Fig. 3. Q-Learning Obstacle Compensation Simulation Result.

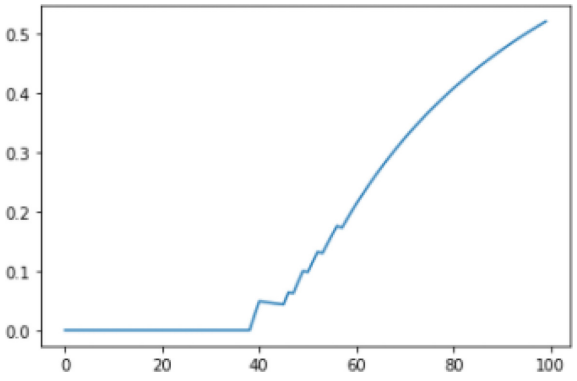

그림 4. Q-Learning 장애물 보상 시뮬레이션 학습 성공률

Fig. 4. Q-Learning Obstacle Compensation Simulation Learning Success Rate.

B. 제안된 Q-Learning 알고리즘 시뮬레이션 결과

본 논문에서 제안한 Q-Learning 정책은 Frozen Lake환경의 벽면에 해당하는 부분에 대한 처벌 값 부여 방법과 '방향성'이라는 개념을 추가하여 에이전트가 일정한 방향으로 움직이도록 유도하는 방안이 있다. 그림 5와 같이, 발생 확률 감소 정책과 방향성 부여 정책, 두 가지를 모두 적용해 시뮬레이션을 진행한 결과 정확도 71%, 학습 시간 00:44.3086s가 소모되었다. 그림 6과 같이, 기본적인 Q-Learning 알고리즘보다 정확도는 21% 상승하였고, 학습 시간은 기존 알고리즘에 비해 41.76%의 시간이 소모되어, 학습 시간에 대해 약 2.4배의 효율을 보여준다.

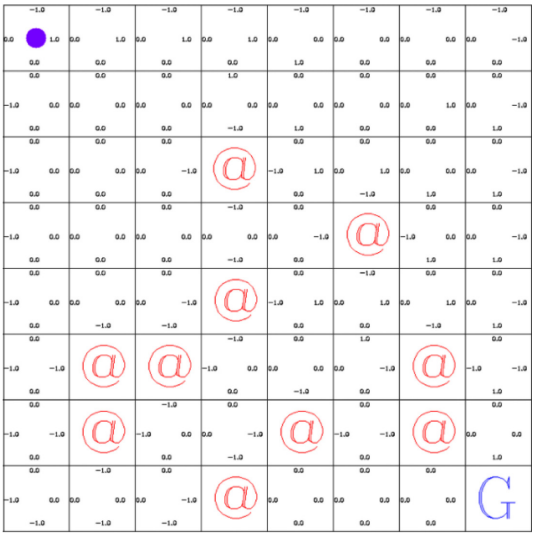

그림 5. 처벌강화 Q-Learning 시뮬레이션 결과

Fig. 5. Strengthening Punishment Q-Learning Simulation Result.

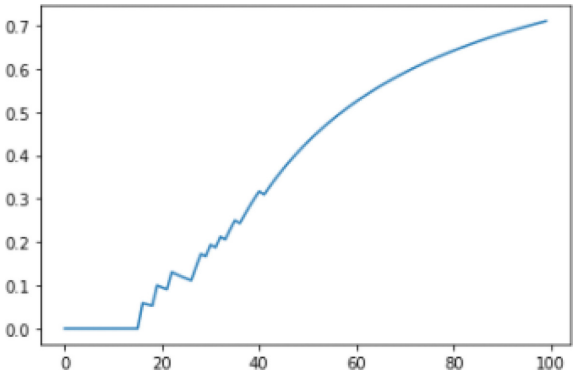

그림 6. 처벌강화 Q-Learning 시뮬레이션 학습 성공률

Fig. 6. Strengthening punishment Q-Learning Simulation Learning Success Rate.

V. 결 론

강화학습은 정해진 환경 속에서 에이전트가 현재 상태에 대해서 선택 가능한 행동 중 미래의 보상을 최대화하는 행동을 선택하는 방법이다. Q-Learning은 특정 상황에서 최적의 행동을 하기 위해 정책을 학습하는 것으로, 미래의 보상 기대값을 최대로 만드는 정책을 학습한다. Q-Learning은 초기 상태에 대해서 모든 행동을 무작위로 취하기 때문에 시뮬레이션의 반복 회차마다 다른 결과가 나온다는 한계점이 있다. 본 논문에서는 Q-Learning의 정책을 경로 탐색에 유리한 방향으로 변경하여 Frozen Lake 8×8 그리드 환경에서 시뮬레이션을 통해 효용성을 판별하였다. 결과적으로 본 논문에서 제안한 Q-Learning 정책은 100에피소드를 기준으로 기존 정책보다 21% 높은 정확도와 약 2.4배 빠른 학습 시간을 보여주었다. 추후 연구를 통해 한정된 시뮬레이션 환경이 아닌 다양한 시뮬레이션 환경을 구성하여 제안된 알고리즘을 학습해보고, 실제 환경에서 장애물을 회피하는 방법들을 연구한다면 효율적인 경로를 생성할 수 있는 개선된 알고리즘을 기대해볼 수 있을 것이다. 또한 공학교육에서 고전적인 방법론 이외에 본 논문에서 제시한 다양한 방법론 및 알고리즘을 적용하여 교육을 진행한다면 응용 학습에 유용하게 활용할 수 있을 것이다.

감사의 글

본 연구는 2021년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력 기반 지역혁신 사업의 결과입니다(2021RIS-004).

참고문헌

[1] X. Wang, L. Jin, and H. Wei, “The shortest path planning based on reinforcement learning,” *Journal of Physics: Conference Series*, vol. 1584, 012006, 2020.

[2] R. S. Sutton and A. G. Barto, “Reinforcement learning: an introduction,” MIT Press Cambridge, vol. 135, 1998.

[3] C. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, pp. 279-292, May 1992.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *Proceeding of the 2013 Conference on Neural Information Processing Systems Deep Learning Workshop*, California: USA, 2013.

[5] J. Clifton and E. Laber, “Q-learning: theory and applications”, *Annual Review of Statistics and Its Application*, vol. 7, pp. 279-301, 2020.

[6] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “OpenAI Gym,” Jun. 2016, arXiv [Online]. Available: https://arxiv.org/abs/1606.01540v1.

용 성 중 (Sung-jung Yong)_정회원

2007년 2월 : 한국기술교육대학교 공학사
2020년 8월 : 한국기술교육대학교 대학원 컴퓨터공학과 공학석사
2021년 8월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 박사과정
〈관심분야〉 AI, 빅데이터, 추천시스템, 웹

박 효 경 (Hyo-gyeong Park)_정회원

2021년 8월 : 한국기술교육대학교 컴퓨터공학부 졸업 (공학사)
2021년 8월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 석사과정
〈관심분야〉 AI, 웹서비스, 빅데이터, 추천 시스템

유 연 휘 (Yeon-hwi You)_정회원

2022년 2월 : 한국기술교육대학교 컴퓨터공학부 졸업 (공학사)
2022년 3월 ~ 현재 : 한국기술교육대학교 대학원 컴퓨터공학과 석사과정
〈관심분야〉 AI, 빅데이터, 추천 시스템

문 일 영 (Il-young Moon)_종신회원

2005년 2월 : 한국항공대학교 항공통신정보공학과 공학박사
2005년 3월 ~ 현재 : 한국기술교육대학교 컴퓨터공학부 정교수
〈관심분야〉 AI, 무선인터넷 응용, 무선 인터넷, 모바일 IP