Journal of KIIT. Vol. 20, No. 11, pp. 65-72, Nov. 30, 2022. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2022.20.11.65

Q-학습을 이용한 소코반 풀이에서 하이퍼-파라미터 분석

장지영*, 권기현**

Analysis of Hyper-parameters in Solving Sokoban using Q-learning

Jiyoung Chang*, Gihwon Kwon**

이 논문은 과학기술정보통신부 및 정보통신기술진흥센터의
2022년 SW공학기술 역량강화 지원사업의 연구결과로 수행되었음(S0436-22-1007)

요 약

본 논문에서는 Q-학습을 이용하여 소코반을 풀이한다. Q-학습은 환경의 현재 상태에 따라 무작위의 행동을 선택하여 탐험을 진행하거나 학습의 활용을 통하여 최대의 보상을 얻을 수 있는 행동을 선택한다. 최적의 Q-테이블을 만들기 위한 과정에서 학습에 영향을 미치는 학습율, 할인율 그리고 감소율 3가지의 하이퍼-파라미터 값을 모든 조합으로 변경하며 학습 성공률을 비교 분석하였다. 이를 이용해 하이퍼-파라미터 값을 각각 학습율은 0.3, 할인율은 0.9 그리고 감소율은 0.9로 고정하고 하나의 값만을 0에서부터 1까지 0.01씩 변경하며 소코반 풀이에 성공과 실패를 결정하는 Q-테이블에 대한 평가를 진행하였다. 그 결과, 소코반 학습과 풀이 평가에 가장 좋은 학습 결과를 제공할 수 있는 하이퍼-파라미터 값을 도출하였다. 감소율의 값이 0.2 부근 일 때 학습 성공률이 가장 컸으며, 감소율을 0.96으로 지정했을 때 최적의 풀이에 가까운 학습에 성공하였다.

Abstract

In this paper, we solve Sokoban using Q-learning algorithm. Q-learning selects random actions according to the current state of the environment to proceed with exploration or select actions that can obtain maximum rewards through the use of learning. We analyzed and compared the learning success rate by changing the values of learning rate, discount factor and decay rate to all combinations which are three hyper-parameters that affect on learning in the process of creating an optimal Q-table. By using these results, the hyper-parameter values were fixed, learning rate to 0.3, discount factor to 0.9 and the decay rate to 0.9. And changed one of them by 0.01 from 0 to 1. Then, we evaluated the Q-table to determine success and failure in solving Sokoban using these values. As a result, hyper-parameter values that can provide the best solution was derived. When the value of the decay rate was around 0.2, the learning success rate was the largest, and when decay rate was 0.96, the learning result was close to the optimal solution for Sokoban.

Keywords
reinforcement learning, Q-learning, learning rate, discount factor, decay rate

* 경기대학교 컴퓨터공학부 학사과정
- ORCID: https://orcid.org/0000-0002-8605-8806
** 경기대학교 컴퓨터공학부 교수(교신저자)
- ORCID: https://orcid.org/0000-0002-8221-4939

· Received: Nov. 01, 2022, Revised: Nov. 21, 2022, Accepted: Nov. 24, 2022
· Corresponding Author: Gihwon Kwon
Dept. of Computer Engineering, Kyonggi University, 154-42, Gwanggyosan-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, Korea
Tel.: +82-31-249-9666, Email: khkwon@kgu.ac.kr

Ⅰ. 서 론

소코반(Sokoban) 퍼즐은 탐색 알고리즘[1], 모델 체킹[2], 심층 신경망[3] 등 다양한 기법으로 풀이할 수 있다. 모델-기반의(Model-based) 이들 기존 기법과는 다르게, 본 논문에서는 소코반을 모델-자유로운(Model-free) 강화학습인 Q-학습 문제로 간주하여 풀이한다.

Q-학습에서는 세 가지 하이퍼-파라미터인 학습율, 할인율 그리고 감소율이 학습 성능에 큰 영향을 준다. 학습율은 현재 Q-값과 미래 Q-값 중에서 새로운 값을 학습에 얼마나 이용할 것인지를 정의한다. 할인율은 미래 보상의 중요 정도이며, 감소율은 경험해보지 않은 행동을 선택하는 탐험(Exploration)과 기존 학습 결과 중에서 가장 좋은 행동을 취하는 활용(Exploitation)과의 균형을 지정한다. 이들 세 가지 하이퍼-파라미터 값이 학습 성능을 좌우하기 때문에 적절한 값 추정 및 이들 간의 관계 분석이 Q-학습에서 중요하다.

소코반에 강화학습을 적용한 이전 논문들은 스테이지 크기에 따른 관계를 분석하거나[4], 상자 수 간의 관계를 분석하였다[5]. 본 논문에서는 상태공간과 행동공간을 갖는 소코반 환경에서 Q-학습을 적용시키고 Q-학습에 사용되는 하이퍼-파라미터 값의 관계를 중점으로 분석한다. 그리고 소코반 상태공간과 행동공간에 대해 풀이에 성공하거나 실패한 Q-테이블도 분석하여 그 원인을 살펴본다.

논문 구성은 다음과 같다. 2장에서 배경 지식을 설명하고, 3장에서는 소코반에 Q-학습을 적용하는 방법과 환경에 대해서 설명하고, 4장에서는 분석 결과를 기술한다. 이어서 5장에서는 결론 및 향후 연구를 기술한다.

Ⅱ. 배경 지식

2.1 Q-학습

창고지기라 불리는 소코반은 일종의 운송 퍼즐이다. 창고에서 창고지기의 역할을 하는 에이전트가 상자를 밀어 목표 위치로 상자를 옮겨 퍼즐을 해결하고자 한다[6].

에이전트의 목표는 제한 스텝 수 안에 모든 상자를 목표 위치들로 옮기는 것이다. 이 과정에서 에이전트가 준수해야 하는 조건들은 다음과 같다. 첫째, 상자가 교착 상태에 빠지면 상자는 더 이상 움직일 수 없다. 둘째, 두 개 이상의 상자를 동시에 움직일 수 없다. 셋째, 상자 뒤에 다음 공간이 비어 있는 경우에 한해서만 인접한 상자 밀기를 시도할 수 있다.

본 논문에서는 소코반에 강화학습 알고리즘인 Q-학습을 적용한다. 강화학습은 행동 심리학에서 영감을 받은 기계학습의 한 분야이다[7]. 지도학습이 정답이 주어진 데이터를 이용하여 규칙을 학습한다면, 강화학습은 입력값과 쌍이 되는 출력값을 준비할 필요 없이 행동에 대한 보상을 이용하여 학습한다. 에이전트 행동이 환경에 영향을 주며, 행동은 환경의 상태를 변화시켜 그에 따른 보상을 받는다.

Q-학습은 에이전트가 환경이 주는 보상을 사용하여 주어진 상태에서 행동할 수 있는 최상의 행동을 학습한다. 현재 상태에서 행동에 대해 보상 받은 후 Q-값을 업데이트 하여 해당 행동이 유익한 지 여부를 기억한다. 또한, Q-학습은 오프-정책(Off-policy)을 사용하여 학습 과정과 행동 평가를 분리한다.

Q-테이블에 저장되는 값을 Q-값이라 하며(상태, 행동) 쌍에 매핑된다. 상태는 에이전트의 관측 가능한 상태 s(State)들의 집합을 가진다. 행동은 상태 s에서 가능한 행동 a(Action)들의 집합이다. 모든 상태에서 에이전트들이 가질 수 있는 행동들은 동일하다. Q-값은 0으로 초기화되어 있으며, 에이전트가 여러 환경에 노출되고 각기 다른 행동을 실행함으로써 다양한 보상을 받게 된다. 초기화되어 있던 Q-값들은 식 (1)을 이용하여 연속적으로 업데이트된다[8].

$$Q(s,a) \leftarrow (1-\alpha)\,Q(s,a) + \alpha[R + \gamma \max_{a'} Q(s',a')] \quad (1)$$

현재 상태 s에 대한 행동 a의 Q-값 $Q(s,a)$는 기존 $Q(s,a)$에 가중치 $(1-\alpha)$를 준 값에 할인된 미래 최대 보상을 결정하는 식 (2)와 합으로 갱신된다. 이 과정을 반복하여 Q-테이블이 특정 값으로 수렴되도록 한다. 그러므로 한 상태에서 더 큰 Q-값을 가진 행동이 더 큰 보상을 받을 수 있다.

$$\alpha[R+\gamma\max_{a'}Q(s',a')] \quad (2)$$

- α: 학습율 $(0 \le \alpha \le 1)$
- γ: 할인율 $(0 \le \gamma \le 1)$
- R: 보상

Q-학습은 단순하고 단일 에이전트 환경에서 우수한 학습 능력을 나타낸다. 본 논문에서는 소코반 풀이를 위한 Q-학습과정에서 소코반 학습 성공률에 대한 영향을 조사하기 위해 학습율과 할인율의 값의 변경을 수행한다.

2.2 ϵ - 그리디 알고리즘

ϵ-그리디 알고리즘은 감소율에 따라 탐험에 의한 행동을 선택하거나 또는 학습 결과를 활용하여 행동을 선택한다. 학습 초기에는 탐험에 의해 행동을 선택하다가 시간이 지남에 따라 지금까지 학습한 결과를 활용하여 행동을 선택하도록 한다. 감소율은 이러한 탐험과 활용 비율을 조정한다.

- ϵ: 감소율 $(0 \le \epsilon \le 1)$

감소율이 1이면 완전히 무작위 탐험하며 0이면 현재 학습 결과를 그대로 활용한다. 본 논문에서는 감소율의 영향을 조사하기 위해 감소율 값을 변경해 가면서 소코반 풀이를 학습한다.

III. 소코반 Q-학습 및 평가

3.1 소코반 환경

본 논문에서는 강화학습을 위해 고안된 Gym-Sokoban의[9] 환경을 이용한다. 소코반 퍼즐 맵을 나타내는 스테이지 요소로는 벽, 바닥, 상자, 목표 위치 및 에이전트가 있다. 추가로 상자가 목표 위치 위에 있는 경우와 없는 경우, 에이전트가 목표 위치 위에 있는 경우와 없는 경우에 따라 7가지 상태를 가진다. 그림 1은 소코반 스테이지의 예시이다:

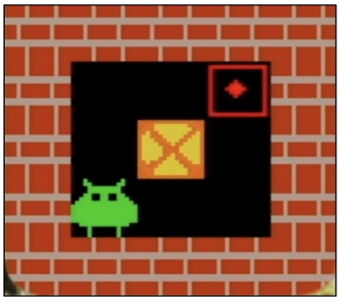

그림 1. 소코반 스테이지
Fig. 1. Sokoban stage

소코반 스테이지로는 직관적인 5X5 사이즈(80X80 픽셀)의 가장 작은 스테이지를 이용한다. 하나의 상자, 하나의 에이전트 그리고 하나의 상자 목표 위치가 존재한다. 해당 스테이지의 상태공간은 5000으로 정의한다. Q-테이블의 행의 크기가 결정되었다. 또한 실제로 에이전트가 움직이는 공간은 벽을 제외한 3X3으로 소코반 풀이 시 너무 많은 스텝을 제공할 필요가 없기 때문에 최대 스텝은 150으로 정하였다.

3.1.1 에이전트 행동

본 실험에서 사용된 에이전트 행동은 9가지이다. 0번 행동은 환경에서 에이전트가 아무것도 변경하지 않는다. 1번 행동부터 4번 행동은 상자를 미는 행동으로, 상자 뒤 다음 공간이 비어있는 경우 해당 방향으로의 인접한 상자를 밀기 위한 시도를 한다. 다만, 인접한 상자가 없는 경우 5~8번의 이동 동작처럼 해당 방향으로 에이전트가 이동한다. 따라서 소코반의 행동공간은 9로 정의된다. Q-테이블의 열의 크기가 결정되었다.

표 1. 소코반 에이전트의 행동
Table 1. Set of actions for Sokoban

ID	Action
0	No operation
1	Push up
2	Push down
3	Push left
4	Push right
5	Move up
6	Move down
7	Move left
8	Move right

3.1.2 보상 값 배정

사용된 보상 구조는 다음과 같다:

- <Perform_Step>: 하나의 스텝마다 보상 값 -0.1을 배정한다.
- <Push_Box_on_Target>: 목표 위치에 상자를 밀어 넣으면 보상 값 1을 배정한다.
- <Push_Box_off_Target>: 목표 위치에서 상자를 밀어 꺼내면 보상 값 -1을 배정한다.
- <Push_All_Boxes_on_Target>: 모든 목표 위치에 상자를 밀어 넣으면 보상 값 10을 배정한다.
- <Game_Done_with_Max_Step>: 최대 스텝으로 게임이 종료되면 보상 값 -10을 배정한다.

<Perform_Step>,<Push_Box_off_Target>와 <Game_Done_with_Max_Step>는 보상 구조에서 패널티에 속한다. <Perform_Step>는 시간의 흐름에 따라 에이전트의 행동 수가 늘어나게 되기 때문에 해당 행동과 결과적인 풀이에 불이익을 준다.

3.2 소코반 Q-학습

본 논문에서는 소코반 스테이지를 풀이하기 위해 에이전트가 수행해야 할 행동의 연속을 알고자 그림 2 알고리즘을 따라 Q-학습한다. 학습된 Q-값은 현재 행동을 수행했을 때의 보상 및 다음 상태부터 할인된 최대 보상을 결합한다.

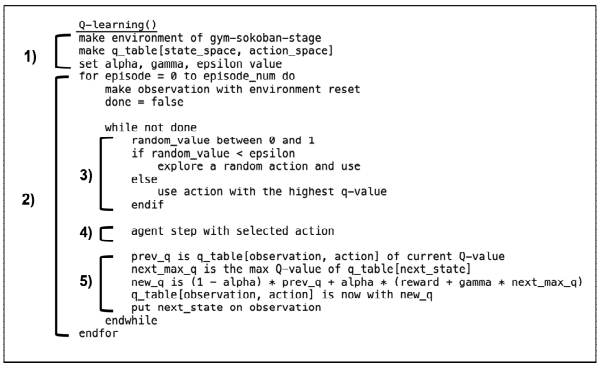

그림 2. Q-학습 알고리즘
Fig. 2. Algorithm for Q-learning

1) 가장 처음에 Q-학습을 진행하고자 하는 소코반 스테이지 환경을 불러온다. 그리고 0으로 초기화된 상태공간X행동공간의 Q-테이블을 생성한다. 이어서 세 가지 하이퍼-파라미터의 값을 설정한다.

2) 학습시킬 에피소드 수를 설정한다. 환경을 초기상태로 재설정하고 시작한다. 에이전트는 상자가 교착상태에 빠져 최대 스텝을 달성하거나, 상자를 목표에 넣는 걸 성공할 때마다 해당 에피소드는 끝나고 새로운 에피소드가 시작된다.

3) ϵ-그리디를 적용하기 위해 0과 1사이의 임의의 값을 이용해 ϵ값보다 작으면 탐험을 통해 새로운 무작위 행동을 선택하고, ϵ값보다 크다면 현재 상태에서 가장 큰 Q-값을 가지는 행동을 선택해 활용한다.

4) 에이전트가 선택된 행동을 바탕으로 스텝을 진행한다. 해당 스텝을 통한 상태 변화에 따라 보상 값과 다음 상태 값을 반환한다.

5) 현재 상태에 대한 Q-값을 prev_q에 저장한다. 그리고 다음 상태에 대해 이미 학습됐던 Q-값 중에서 가장 큰 값을 가져와 next_max_q에 저장한다. 이제 Q-값에 식 (1)에 맞게 업데이트를 수행한다. 다음 상태 값을 상태로 넣어주고 3), 4), 5)를 에피소드가 끝나기 전까지 반복한다.

소코반에 Q-학습을 진행하기 위해서는 세 가지 하이퍼-파라미터 α, γ 그리고 ϵ에 대한 값을 설정한다. α는 지도학습 설정과 같이 Q-값들이 매 반복마다 업데이트 되는 것의 기대 정도이다. 현재 Q-값과 미래 Q-값 중에 새로운 값을 얼마나 학습에 받아들일 것인지를 나타낸다. γ은 찾은 행동에 대한 Q-값에 곱해지는 값으로 0에 가까울수록 현재의 Q-값을 중시하고 1에 가까울수록 미래의 Q-값을 중시한다. ϵ는 경험해보지 않은 행동에 따른 경로를 찾기 위한 탐험 그리고 활용의 균형 정도를 설정한다. 이렇게 서로의 역할이 다르기 때문에, 각자 의미에 따라 적절한 값을 지정해주어야 해당 스테이지에 대해 더 좋은 Q-학습이 진행된다.

Q-학습을 통해 만들어진 Q-테이블의 Q-값이 소코반 풀이 과정을 나타낸다. 본 논문에서는 소코반 풀이를 평가하기 위해 한 번의 에피소드를 초기 상태에서부터 가장 큰 Q-값으로 행동을 선택해 한 스텝씩 풀이한다. 평가 결과로 학습된 Q-테이블이 소코반 풀이에 성공한다면, Q-테이블 평가 성공 보상 값과 Q-테이블 평가 성공 스텝 수를 반환한다. 상자가 교착 상태에 빠졌거나, 최대 스텝 이내로 상자를 목표 위치에 옮기지 못한다면 좋지 않은 학습 과정에서 Q-값이 업데이트되어 결국 소코반 풀이에 실패한다.

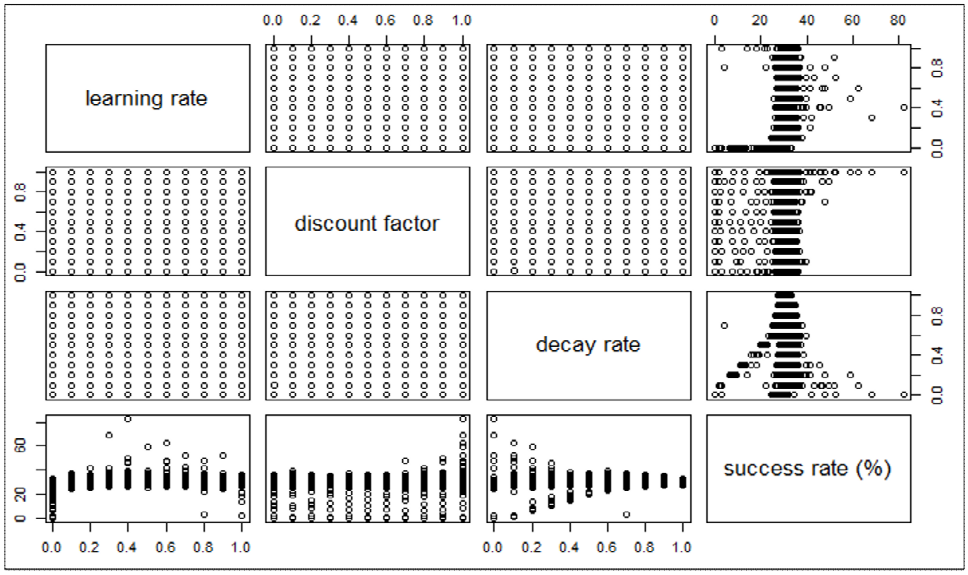

그림 3. 하이퍼-파라미터 값에 따른 학습 성공률
Fig. 3. Learning success rate by Hyper-parameters

IV. 분 석

4.1 Q-학습 성공률

본 논문에서는 최적의 Q-테이블을 만들기 위한 과정에서 학습에 영향을 미치는 세 가지 하이퍼-파라미터인 α, γ 그리고 ϵ에 대한 값을 변경하며 두 단계에 걸쳐 학습 성공률을 비교한다.

첫 번째 단계에서는 각 하이퍼-파라미터들의 값을 모두 0에서부터 1까지 0.1 단위로 지정하며 얻은 모든 조합의 학습 성공률 값 분포를 구한다. 그림 3에 나타난 분포를 살펴보면 학습율의 값은 0.3 그리고 할인율의 값은 0.9가 안정적이며 감소율의 값은 0.9에서 높은 학습 성공률을 가짐을 확인하였다. 이를 이용하여 두 번째 단계로, α, γ 그리고 ϵ에 대한 값을 변경하는 과정에서 변경할 값을 제외하고 나머지 값은 표 2의 고정 값을 주고 0.01 단위로 세분화시켜 각 1000번의 에피소드를 통해 Q-테이블을 학습시킨다:

표 2. 고정 하이퍼-파라미터 값
Table 2. Fixed value of Hyper-parameters

Episode	α	γ	ϵ
1000	0.3	0.9	0.9

그림 4 그래프의 x축은 각 0에서부터 1까지 α,γ와 ϵ의 값이며 y축은 학습 성공률을 나타낸다.

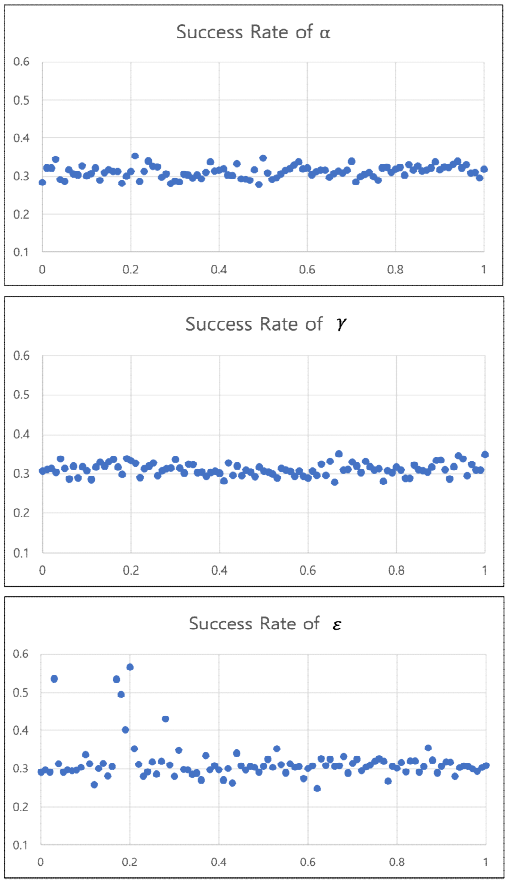

그림 4. 하이퍼-파라미터 변경에 따른 학습 성공률
Fig. 4. Success rate by changing Hyper-parameters

3가지 모두 대부분 30%의 학습 성공률을 가지고 있다. 1000개의 에피소드에서 기본적으로 300개 정도의 에피소드가 학습 과정에서 소코반 풀이에 성공한 것이다. 특히 ϵ의 경우, 0.2 부근의 값으로 지정되었을 때, 50% 이상의 학습 성공률을 보이는 모습을 살펴볼 수 있다. 0.2라면 1보다는 0에 가까운 수치이기 때문에, 무작위 탐험으로 9가지의 행동을 살펴보며 스텝 수를 낭비하기보다 지금까지 학습한 해당 상태에 대한 Q-값을 더 가치를 두는 경우이다. 이렇게 값을 비교해보니 3가지 하이퍼-파라미터는 모두 시간이 지남에 따라 감소하는 형식으로 제공됨이 좋다는 것을 알 수 있다. 에이전트가 계속 학습함에 따라 정보가 축적되기 때문이다. 만약 에피소드 수를 늘리게 된다면 시간에 따라 값들을 감소시키는 방향이 Q-학습을 이용한 소코반 풀이에 유용하다.

4.2 소코반 풀이 결과

이번에는 α, γ 그리고 ϵ의 값을 바꿔가며 학습시킨 이후 생성된 Q-테이블을 평가한다. Q-테이블의 값에 따라서 소코반 풀이는 성공하거나 실패한다. 그 중에서도 성공한 경우의 결과 값을 분석한다. α는 100개의 Q-테이블에서 14개가 풀이에 성공하였다. γ는 100개 중에 12개, ϵ는 100개 중에 20개로 가장 많이 성공하였다(표 3-5).

표 3. α값에 따른 학습 성공률과 평가 성공 결과 값
Table 3. Comparative results by α value

No	α	Success rate(%)	Q-Table evaluation	
			Reward	Step
28	0.28	30.7	-0.5	125
37	0.37	31.1	3.6	84
40	0.4	31.6	0.7	113
41	0.41	31.9	2.7	93
54	0.54	30.7	5.8	62
55	0.55	31.5	10.6	14
65	0.65	29.7	5.7	63
67	0.67	31.4	9.5	25
75	0.75	29.9	1.2	108
77	0.77	32.2	5.8	62
79	0.79	31.1	8	39
86	0.86	31.4	5.8	62
98	0.98	31.1	6.1	59
99	0.99	29.6	8.3	37

표 4. γ값에 따른 학습 성공률과 평가 성공 결과 값
Table 4. Comparative results by γ value

No	γ	Success rate(%)	Q-Table evaluation	
			Reward	Step
2	0.02	31.3	11.3	7
4	0.04	33.8	11.3	7
17	0.17	31.6	2.5	95
20	0.2	33.2	4	80
28	0.28	31.2	3	90
49	0.49	31.6	6.8	52
52	0.52	29.9	8.4	36
55	0.55	30.9	11.1	9
70	0.7	32.9	3.4	86
75	0.75	30.9	2.1	99
79	0.79	30.3	0.9	111
91	0.91	31.0	11	10

표 5. ϵ값에 따른 학습 성공률과 평가 성공 결과 값
Table 5. Comparative results by ϵ Value

No	ϵ	Success rate(%)	Q-Table evaluation	
			Reward	Step
3	0.03	53.5	10.7	13
9	0.09	30.3	0.8	112
12	0.12	25.9	10.7	13
18	0.18	49.5	9.8	22
20	0.2	56.6	10.7	13
26	0.26	28.5	7.7	43
28	0.28	43.1	10.9	11
30	0.3	28.0	0.7	113
31	0.31	34.8	1.4	106
35	0.35	28.8	3	90
38	0.38	29.7	7	50
40	0.4	29.7	3.6	84
47	0.47	30.7	7	50
48	0.48	30.3	-1.1	131
51	0.51	32.5	-0.1	121
65	0.65	32.5	1.8	102
66	0.66	30.7	4.4	76
90	0.9	30.7	10.6	14
91	0.91	31.8	-1.7	137
96	0.96	30.6	11.4	6

그리고 50% 이상의 학습 성공률을 갖는 대부분은 Q-테이블 평가에서 성공한다. 이 점에서 하이퍼-파라미터 값을 적절히 조정하여 학습 과정에서 더 나은 Q-값을 만드는 작업의 중요성이 드러난다. 가능한 한 빨리 최대 보상을 얻을 수 있는 풀이를 해내고자 하기 때문에, 최고의 보상을, 시간을 나타내

는 스텝 수로 나눈 비율을 구하여 얻을 수 있는 가장 좋은 하이퍼-파라미터의 값들을 선택하도록 하는 것이 바람직하다. 그리고 성공 스텝 수가 길어질수록 패널티를 받기 때문에 성공 보상 값이 작아진다. γ과 ϵ는 20 스텝 이내로 성공한 경우가 다수 존재한다. 이는 해당 스테이지에서 에이전트가 행동 가능한 가장 짧은 스텝인 5 스텝에 가까운 수준으로 학습에 성공하였다는 것이다.

4.3 Q-테이블 분석

앞의 성공한 경우 중 가장 잘 학습된 Q-테이블을 분석한다. ϵ의 No.96 인 0.96으로 값을 지정하고 학습을 시킨 경우의 성공 스텝 과정은 다음과 같다: 초기 상태에는 1의 보상을 갖고 시작한다. 에이전트는 위로 이동하고 우측으로 상자를 민다. 이어서 우측으로 이동을 시도한 뒤, 아래로 이동하고 우측으로 이동한다. 마지막으로 위로 상자를 밀어 상자를 목표에 위치시켰다. 이렇게 총 11.4의 보상으로 6 스텝만에 종료되었다. 이용된 상태의 Q-값들은 모두 0 이상의 좋은 양수의 값을 가지고 있다.

그리고 모든 경우의 Q-테이블에서, 가장 첫 번째 상태에서의 행동은 0~249(상태)에서 Q-값이 0인 경우를 제외하고 모두 한 가지의 행동을 가리킨다. 무작위 행동을 탐험하는 과정에서 선택되는 첫 행동이 중요함을 알 수 있다. 상대적으로 더 많은 스텝으로 풀이한 Q-테이블의 값은 표 6의 수치보다 작은 값의 Q-값을 갖고 있음을 확인할 수 있다. 이는 대부분의 스텝이 효율적으로 사용되지 못하고, 패널티의 보상을 많이 받은 것을 의미한다.

표 6. ϵ=0.96일 때 평가 결과
Table 6. Evaluation result for ϵ=0.96

No	State	Q-value	Reward	Action	Box
1	176	4.298	-0.1	Push up	-
2	1726	3.793	-0.1	Push right	Moved right
3	1753	3.522	-0.1	Move right	-
4	1773	2.915	-0.1	Move down	-
5	1800	2.125	-0.1	Move right	-
6	1819	3.074	+10.9	Push up	On target

V. 결론 및 향후 연구

본 논문에서는 소코반 풀이를 위해 에이전트의 행동에 따라 환경의 상태를 변화시키고 보상 값을 주는 Q-학습을 적용하는 기법을 선택하였다. Q-학습은 매 스텝마다 학습하기 때문에 횟수가 빈번하고 더 정확하다. 첫 행동을 선택할 때는 ϵ-그리디하게 선택하며 정책이 없어 소코반 풀이에 적용하기 용이하다. 또한, 다음 상태에서 최대의 보상을 받을 수 있는 행동을 선택하기 때문에 직관적이다.

본 논문에서는 소코반의 에이전트가 Q-값을 구하는 과정에서 영향을 주는 Q-학습의 하이퍼-파라미터 값을 두 단계에 걸쳐 변화를 주며 학습 성공률이 달라지는 모습을 확인하였다. 첫 번째 단계에서 세 가지 하이퍼-파라미터 값을 모든 조합으로 학습을 진행해본 결과, α를 0.3, γ를 0.9 그리고 ϵ를 0.9로 지정하였을 때, 안정적인 학습 성공률을 가진다는 것을 파악하였다. 두 번째 단계에서는 두 가지 하이퍼-파라미터를 앞 단계에서 도출한 값으로 고정한 뒤, 변화를 주고자 하는 값을 세분화하여 0.01의 단위로 변경해보며 학습과 Q-테이블 평가를 진행하였다. 그 결과, ϵ가 0.96일 때, 최적의 풀이에 가깝게 성공하였다. 그리고 풀이에 성공한 Q-테이블은 실패한 Q-테이블에 비해 큰 Q-값을 가지고 있음을 확인하였다. 따라서, 이번 연구를 통하여 적절한 하이퍼-파라미터의 선택이 Q-학습의 성능에 영향을 주며 더 나은 최적의 경로를 찾는 Q-테이블을 만들 수 있다는 사실을 확인하였다.

향후 연구로써 에피소드의 경과에 맞춰 점차 하이퍼-파라미터의 값들을 축소시키는 방향의 연구와 유전자 알고리즘을 통한 하이퍼-파라미터의 값을 추정하여 비교하는 연구로 지속하고자 한다. 또한 안전성 분야에 기계학습을 적용하려 한다. 기존의 정리 증명(Theorem-proving)[10], 모델 검증, 합성(Synthesis)[11]과 같은 기법들을 실세계에 적용하려는 시도들이 많이 있었지만 그 범위와 효용성은 제한적이었다. 하지만 기계학습의 발전과 최근 많은 분야에 적용하려는 노력과 결과를 확인해보면 기계학습을 통해 안전성 검증 및 확인, 그리고 안전한 시스템 개발에 기여를 할 수 있을 것이라 확신한다.

References

[1] R. Setiani and B. Indriyono, "Implementation of A * Algorithm for Solving Sokoban Logic Games", J ournal of Applied Intelligent System, Vol. 4, No. 2, pp. 104-111, Dec. 2019. https://doi.org/10.33633 /jais.v4i2.3409.

[2] T. Lee and G. Kwon, "Relay Model Checking Based on Partitioning Properties", Transactions on Programming Languages, Vol. 17, No. 3, pp. 1-7, Nov. 2003.

[3] F. Pardo, V. Levdik, and P. Kormushev, "Scaling All-Goals Updates in Reinforcement Learning Using Convolutional Neural Networks", Proc. of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 4, pp. 5355-5362, Apr. 2020. https://doi.org/10.1609/aaai.v34i04.5983.

[4] Z. Hong, T. Chen, and Y. Lin, "Topological Experience Replay", arXiv:2203.15845, May 2022. https://doi.org/10.48550/arXiv.2203.15845.

[5] Z. Yang, M. Preuss, and A. Plaat, "Transfer Learning and Curriculum Learning in Sokoban", Communications in Computer and Information Science, Vol. 1530, Springer, Cham, Nov. 2022. https://doi.org/10.1007/978-3-030-93842-0_11.

[6] COMP 3211 Final Project – Group 6, "A Sokoban Solver Using Multiple Search Algorithms and Q-learning", HongKong University of Science and Technology, 2017. [Accessed: Nov. 10, 2022]

[7] S. Mun, K. Hwang, and T. Kim, "A Comparative Study on Reinforcement learning Using Automatic Driving", Proc. of Symposium of the Korean Institute of communications and Information Sciences, pp. 481-482, Nov. 2020.

[8] B. Jang, M. Kim, and G. Harerimana, "Q-Learning Algorithms: A Comprehensive Classification and A pplications," in IEEE Access, Vol. 7, pp. 133653-133667, Sep. 2019. https://doi.org/10.1109/ACCES S.2019.2941229.

[9] Schrader, M.P.B.: Gym-Sokoban (2018), https://git hub.com/mpSchrader/gym-sokoban. [Accessed: Nov. 12, 2022]

[10] D. Loveland, "Automated Theorem Proving: A Logical Basis", Fundamental Studies in Computer Science, Elsevier, ISBN 9781483296777, Vol. 6, 2016.

[11] R. Bloem, B. Jobstmann, N. Piterman, and A. Pn ueli, "Synthesis of reactive (1) designs", Journal o f Computer and System Sciences, Vol. 78, No. 3, pp. 911-938, 2012. https://doi.org/10.1016/j.jcss.201 1.08.007.

저자소개

장 지 영 (Jiyoung Chang)

2018년 3월 ~ 현재 : 경기대학교 컴퓨터공학부(학사과정)
관심분야 : 소프트웨어 공학, 소프트웨어 안전, 기계학습

권 기 현 (Gihwon Kwon)

1985년 2월 : 경기대학교 전자계산학과(이학사)
1987년 8월 : 중앙대학교 전자계산학과(이학석사)
1991년 2월 : 중앙대학교 전자계산학과(공학박사)
1991년 2월 ~ 현재 : 경기대학교 컴퓨터공학부 교수
1999년 ~ 2000년 : 미국 카네기멜론대학 전산학과 방문교수
2006년 ~ 2007년 : 미국 카네기멜론대학 전산학과 방문교수
2014년 ~ 2016년 : 한국정보과학회 소프트웨어공학 소사이어티 회장
2021년 ~ 현재 : 경기대학교 SW중심대학 사업단장
2022년 ~ 현재 : 경기대학교 소프트웨어경영대학 학장
관심분야 : 소프트웨어 공학, 소프트웨어 안정성, 정형 검증 및 정형 합성