강화학습 $\epsilon - greedy$ 정책에서 Neurotransmitter 규칙 적용을 통한 생물체의 의식적 행동 반영 이론적 접근

Lim Jung Yeoul (2022-3-18)

Abstract

강화학습 이론에서 model-free 한 경우 epsilon-greedy 한 정책을 사용하게 되는데 랜덤 분포로 탐색적 선택을 하도록 설정을 하도록 되어있습니다. 하지만 '사람은 탐험적 선택을 랜덤으로 하는가'에서 의문점을 가지고 랜덤 함수와 cellular automata 를 확인한 결과 생명체는 일반 랜덤함수가 아닌 cellular automata 와 유사한 neurotransmitter 규칙이 적용됨을 주장하고자 합니다.

Index

1.Reinforcement Learning $\epsilon - greedy$ policy

강화학습에서는 현재 가치가 높아 그리디한 선택을 하는 것보다 미래에 더 나은 가치가 있는 상태가 있을 것에 대비해 다른 선택에 대한 여지를 남겨둬야 합니다. 이때 엡실론 그리디를 사용합니다. ϵ확률로 탐함적 선택을 합니다.

$$\epsilon - greedy\text{정책} : \pi^*(a|s) = \begin{cases} \frac{\epsilon}{|\mathcal{A}|} + 1 - \epsilon & ,\text{if a} = \underset{a\in\mathcal{A}}{\text{argmax}}\ Q_\pi(s,a) \\ \frac{\epsilon}{|\mathcal{A}|} & ,\text{otherwise} \end{cases}$$

$|\mathcal{A}| =$ 가능한 액션의 개수

그리디한 선택 반대인 탐험적 선택을 할 경우 임의적[1] 선택인 랜덤 함수를 사용한 것에 의문점을 가집니다

2-1. Cellular automata and Random Generator

순차적 랜덤 숫자들이란 규칙을 찾을 수 없고 예측할 수 없는 연속적인 모음입니다.[6] 순차적 랜덤 숫자들은 Uniformity 와 Independence 속성을 가집니다. [7,8] Uniformity 는 어디서나 균등하게 발생할 수 있다는 뜻이고, Independence 는 다른 값들과 어느 하나 관련이 없어야 한다는 뜻입니다.

강화학습 이론으로 파이썬 프로그램을 만들 때 쓰이는 random 함수는 실제 랜덤함수가 아닌 의사 랜덤 함수입니다. 버전 3.10.2 python 은 메르센 트위스터(Mersenne Twister)를 핵심 생성기로 사용합니다. 53 비트 정밀도의 float 를 생성하며, 주기는 $2^{19937-1}$입니다.[9] 이와 같은 의사 랜덤 함수는 4 가지 특성을 목표로 만들어집니다.[8]

- 순차적으로 출력된 값은 좋은 랜덤성.
- 빠른 생성속도와 효율성.
- 생산성
- 주기성

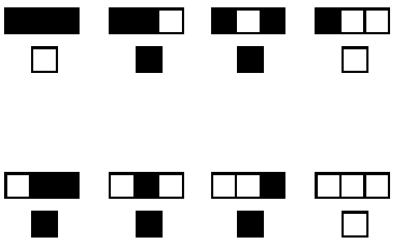
[1] 임의적 결정은 랜덤 결정과 반드시 동일하지는 않습니다. 예를 들어 1973년 석유 파동 동안 미국인들은 번호판이 홀수일 경우 홀수일에만, 번호판이 짝수일 경우 짝수일에만 휘발유를 구입할 수 있었습니다. 시스템은 잘 정의되었으며 제한 사항이 무작위가 아닙니다. (e.g,[5])

Cellular automata 는 인접한 셀의 상태를 기반으로 한 일련의 규칙에 따라 여러 개별 시간 단계를 통해 진화하는 셀의 모음입니다.[10] 콘웨이의 Game of Life 의 규칙의 경우 아래와 같습니다.

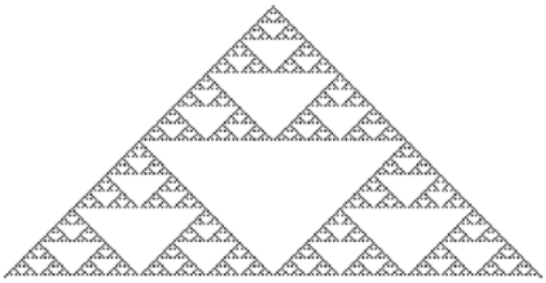
- 죽어 있는 칸(하얀색)과 인접한 8 칸 중 3 칸에 세포가 살아 있다면 그 세포는 다음 세대에 살아납니다.
- 살아 있는 칸(검은색)과 인접한 8 칸 중 2 칸 혹은 3 칸의 세포가 살아 있다면 해당 칸의 세포는 살아 있는 상태를 유지합니다.
- 그 이외의 경우 해당 칸의 세포는 다음 세대에 고립돼 죽거나 혹은 주위가 너무 복잡 해져서 죽거나 죽은 상태를 유지합니다.

이런 식으로 모든 세포가 죽어버리는 패턴이 존재하는가 하면 세포가 전멸하지 않고 영원히 살아남거나 생사가 무한히 반복되는 패턴이 존재합니다.[11]

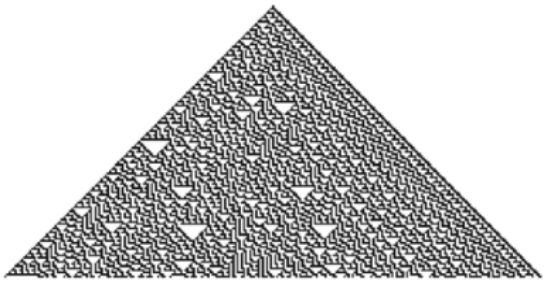
1-dimentional cellular automata 에서는 검은색은 1 하얀색은 0 인 3 개의 칸을 가지고 다음 세대의 칸에서는 어떻게 칸이 생성될지 규칙을 아래와 같이 만들었습니다.



이 규칙들은 총 2^8개의 다양성을 가집니다.



Rule 18



Rule 86

규칙의 번호를 선택하면 위와 같은 형태가 보여집니다. 그리고 이런 순차적 숫자를 제공하는 함수들 중 너무 복잡해서 랜덤하게 보이는 경우가 있습니다. **Cellular automata 도 자연의 랜덤의 모델로 사용할 수 있습니다.** [6,11]

2-2. Cellular automata and real brain neural network

Peter B. Lloyd 는 Modelling Consciousness within Mental Monism: An Automata-Theoretic Approach 에서 의식의 해결법을 3 가지로 제시합니다. 사람의 정신은 물리적 세계와 정신적 세계가 모두 실제하고 마음이 순수한 물리적 시스템으로 환원될 수 없다고 주장하는 이원론, 그리고 궁극적으로 현실은 의식적인 마음으로만 구성되어 있다고 주장하는 정신적 일원론이 있습니다. 그리고 오직 물체만이 존재한다는 물리적 일원론이 있습니다.[15] 저는 정신적 일원론을 주장합니다. 최소 단위의 지적인 생각을 하는 물리적 생물체가 세상을 센서를 통해 받아들이고 현실에 대해 상대적인 plus, minus 를 평가를 하고 이 사고들이 모여 고등 사고 체계가 되었다고 생각합니다.[2]

Cellular automata 는 뇌의 뉴런의 dendritic 성장 형태 에도 기여를 합니다.[16] 현재의 뉴런 모델은 상미분 방정식을 사용하여 뉴런의 전압 패턴을 모방할 수 있습니다. 이를 해결하기 위해 뇌의 많은 수의 뉴런생성을 시뮬레이션 하기 위해 cellular automata 를 사용합니다. [17]

3-1. Neurotransmitter rule and cellular automata rule

뉴런에서 뉴런으로 선택과 전달이 뇌의 활동의 근본 원리입니다. 신경 전달물질은 뉴런의 axon terminal 에서 받은 값을 전달할지 말지 규칙을 줍니다. 신경 전달 물질 또한 cellular automata 처럼 규칙이 있습니다.[20]

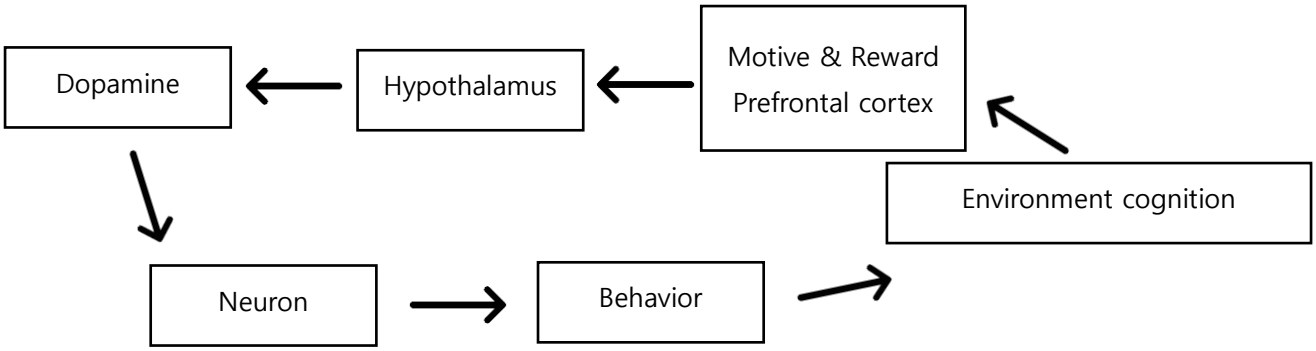
- 흥분성 신경전달물질
- 억제성 신경전달물질
- 조절 신경전달물질

신경 전달 물질의 유일한 직접 작용은 수용체를 활성화하는 것입니다. 따라서 신경전달물질 시스템의 효과는 전달자를 사용하는 뉴런의 연결과 수용체의 화학적 특성에 따라 달라집니다.[21]

[2] 공리주의에서 옳다고 여기는 것은 개인 행복 총합의 최대화이다.[19], Neurotransmitter : ex) glutamate (Excitatory neurotransmitters), GABA(Inhibitory neurotransmitters)[21] 신경전달 물질도 억제성 신경전달물질과 흥분성 신경전달물질로 나뉘어 있습니다.

3-2 Neurotransmitter and Circulation of environment and agent behavior

뇌(의식)가 발달하기 위해서는 뇌자체의 발달에서도 중요하지만 주위 환경에 대한 인지에 의한 뇌 발달도 중요합니다. 예를 들어 뇌의 시신경 발달을 위해선 유아기(18개월 이상)에 주위 환경인 빛의 의한 자극이 필수적입니다. [22] 도파민에 의한 뇌의 행동 원리도 비슷한 것 같습니다. 도파민은 만족감에 영향을 주고 만족을 예상하기만해도 도파민 수치가 높아집니다. 예를 들어 쿠키 구은 냄새와 요리가 완성된 것을 보고 도파민 수치 값이 올라 갈 수 있습니다. 도파민은 이외에도 혈류량, 소화, 집중력과 기억 등 여러 몸의 기능에 영향을 미칩니다.[23]



위에처럼 뇌에 의한 자극이 먼져나, 자의식의 형성이 먼저냐 할 수 없을 만큼 얽혀 있는 것 같습니다.

4. summary

Cellular automata 는 사람의 뇌의 정신적 분석, 생물학적 뇌의 뉴런 생성 시뮬레이션, 랜덤함수에서 쓰이기 때문에 강화학습의 $\epsilon - greedy$ 정책에서 랜덤함수 대신에 cellular automata 으로도 강화학습을 구현할 수 있습니다. 인간의 행동을 추적하려면 절대적인 랜덤함수가 아닌 뇌의 발달과 주위환경에 영향을 받는 랜덤함수임을 주장합니다. 생명체 의식처럼 강화학습에 적용하기 위해선 탐색적 랜덤함수를 신경전달물질에 의한 가치 형성으로 지정을 합니다. 그 다음 반복적으로 똑같은 상황을 시뮬레이션을 시켜 그 생명체의 가치 판단으로 인한 앞으로의 행동을 생각해 볼 수 있을 것 같습니다.

5. Reference

[1] Reinforcement Learning: An Introduction Second edition, in progress Richard S. Sutton and Andrew G. Barto

[2] Decision Theory Principles and Approaches Giovanni Parmigiani Johns Hopkins University, Baltimore, USA

[3] "Markov Decision Process(MDP)" UCL Course on RL lecture2,

last modified Jan 11. 2021, accessed Feb 26.2022 ,

https://www.davidsilver.uk/wp-content/uploads/2020/03/MDP.pdf

[4] "Markov Decision Process" 위키피디아

last modified Feb 06. 2022, accessed Feb 26.2022 ,

https://en.wikipedia.org/wiki/Markov_decision_process

[5] "Arbitrariness" 위키피디아

Last modified Nov 01.2021, accessed Feb 26.2022 ,

https://en.wikipedia.org/wiki/Arbitrariness

[6] Random Sequence Generation by Cellular Automata STEPHEN WOLFRAM

[7] " Properties of Random Numbers " Bucknell university

Last modified Oct 18.2002, accessed Feb 26.2022,

https://www.eg.bucknell.edu/~xmeng/Course/CS6337/Note/master/node37.html#:~:text=must%20have%20two%20important%20properties,relation%20with%20the%20previous%20values

[8] STATISTICAL PROPERTIES OF PSEUDORANDOM SEQUENCES By Ting Gu

[9] "random – random generator" 파이썬 3.10.2 document

Last modified Feb 27. 2022 , accessed Feb 27. 2022

https://docs.python.org/ko/3/library/random.html

[10] "Cellular Automaton" wolfram mathworld

Last modified Feb 21 2022, accessed Feb 27. 2022

https://mathworld.wolfram.com/CellularAutomaton.html#:~:text=A%20cellular%20automaton%20is%20a,many%20time%20steps%20as%20desired.

[11] "Conway Game of Life" 나무위키

Last modified Jan 07 2021, accessed Feb 27. 2022
https://namu.wiki/w/%EC%BD%98%EC%9B%A8%EC%9D%B4%EC%9D%98%20%EC%83%9D%EB%AA%85%20%EA%B2%8C%EC%9E%84
[12] When are cellular automata random? J. B. Coe13, S. E. Ahnert4 , and T. M. A. Fink123
[13] Cellular Automata-Based Parallel Random Number Generators Using FPGAs
[14] CONTINUOUS VALUED CELLULAR AUTOMATA AND DECISION PROCESSES OF AGENTS FOR URBAN DYNAMICS
[15] Modelling Consciousness within Mental Monism: An Automata-Theoretic Approach Peter B. Lloyd
[16] Cellular Automata and Artificial Brain Dynamics Alberto Fraile 1,2,* , Emmanouil Panagiotakis 1 , Nicholas Christakis 1 and Luis Acedo 3,*
[17] Totalistic cellular automata model of a neuronal network on a spherical surface Reinier Xander A. Ramos* and Johnrob Y. Bantang
[18] Why Brain Criticality Is Clinically Relevant: A Scoping Review -Vincent Zimmern
[19] "공리주의" 나무위키
Last modified Mar 6 2022, accessed Mar 10. 2022
https://namu.wiki/w/%EA%B3%B5%EB%A6%AC%EC%A3%BC%EC%9D%98
[20] "The Role of Neurotransmitters" VeryWellMind
Last modified Jul 8 2021, accessed Mar 18. 2022
https://www.verywellmind.com/what-is-a-neurotransmitter-2795394#:~:text=A%20neurotransmitter%20is%20a%20chemical,%2C%20muscles%2C%20or%20other%20neurons.
[21] "neurotransmitter" Wikipedia
Last modified Jan 6 2022, accessed Mar 18. 2022
https://en.wikipedia.org/wiki/Neurotransmitter#Neurotransmitter_actions

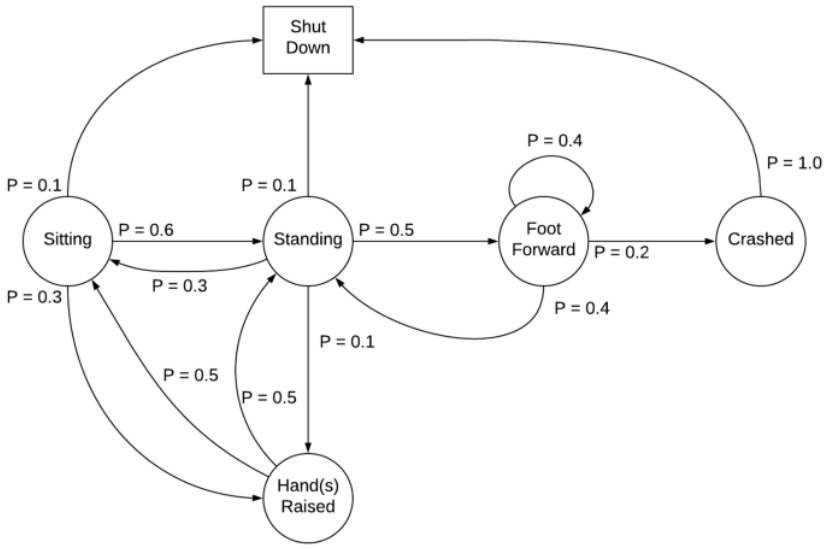
[22] Light Exposure and Eye Growth in Childhood -Scott A. Read; Michael J. Collins; Stephen J. Vincent

[23] "How Does Dopamine Affect the Body?" Healthline
Last modified Nov 5 2019, accessed Mar 18. 2022

https://www.healthline.com/health/dopamine-effects#definition

6.Reinforcement Learning 기본 정리

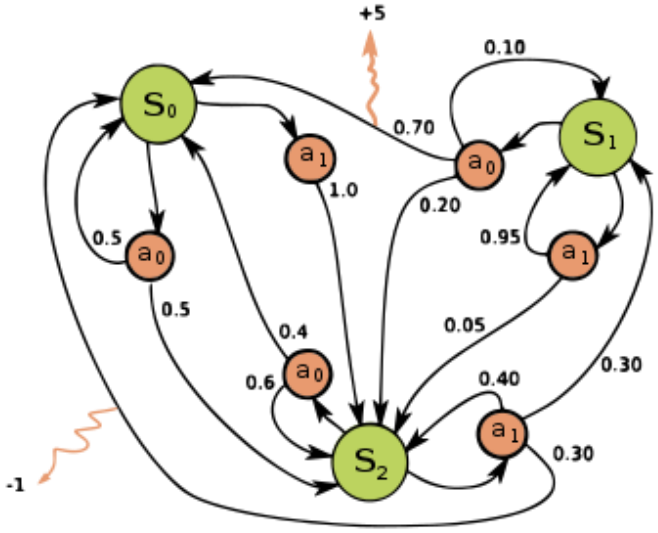
강화학습이란 순차적 의사결정 sequential decision making[1,2] 문제를 풀고자 하는 학습 이론입니다. 강화학습을 시작하기 위해선 우선 MP(Markov decision process)에 대해서 간단하게 설명 드리겠습니다.



로봇 예제 샘플[3]

위 그림은 로봇이 행동할 수 있는 상황을 마르코프 프로세스로 모델링한 그림입니다. 총 6 개의 상태가 있고 1 분이 지나면 다음 상태로 상태 전이를 합니다. 예를 들어 맨처음이 standing 이라면 0.5 확률로 footforward 상태로 움직이게 됩니다. 또 standing 에서는 0.1 확률로 shutdown 상태로 움직일 수 있고 shutdown 의 사각형 상태는 종료 상태입니다. 여기에 도달하는 순간 마르코프 프로세스는 끝이 납니다. 상태전이들의 확률들의 모음을 행렬로 표현하면 전이 확률 행렬 P 로 P_{ss}, 로 표현할 수 있습니다.

MRP(Markov Reward Process)는 MP 에서 보상을 추가한 것으로 상태에 도착하게 되면 보상을 받게 됩니다. 감쇠인자와 return 값이 여기서 또 추가가 되는데 감쇠인자(γ)는 현재와 미래의 보상의 중요성을 표현한 파라미터이고 감쇠인자와 보상의 곱들의 합을 Return 이라고 합니다.



그림은 MDP 의 예로 초록색 원은 상태를 나타내고 액션은 주황색 원으로 그리고 보상은 주황색 화살표로 표현되어 있습니다.[4]

MDP(Markov Decision Process)는 의사결정을 MRP 에서 추가를 하였습니다. 의사결정에는 나(Agent)와 나의 의지인 action(A)[3]이 들어가 있습니다. 할 수 있는 action 과 보상, 상태 전이 확률 등이 포함이 되어서 복잡하게 변해서 정책함수를 추가하여 표현합니다. 여기서 정책함수는 Agent 안에 존재하고 MDP 는 환경으로 외부에 존재합니다.

$$\pi(a|s) = P(A_t = a|S_t = s)$$

$P^a{}_{ss'}$: action 이 추가된 전이 확률 행렬

상태가치함수는 여기서 s 의 상태를 평가하고 싶은 의도에서 출발하였습니다. 이는 주어진 상태로부터 미래에 얻을 리턴의 기대값을 상태의 value 로 정의하였습니다.

$$v_\pi(s) = E_\pi[r_{t+1} + \gamma r_{t+2} \dots |S_t = s] = E_\pi[G_t|S_t = s]$$

$$G_t = r_{t+1} + \gamma r_{t+2} \dots$$

상태액션가치함수는 s 상태일 때 a 의 가치를 평가하기 위해서 생겼습니다. 또 정책에 따라 이후 상태가 달라지므로 정책을 고정시켜 놓고 평가합니다.

$$q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a]$$

---

[3] 행동이라 하기엔 MP에서 기계적인 결과들과 다름을 구분하기 위해서 의지라고 표현했습니다. e.g 의지적으로 선택 하였지만 몸이 실제로 그러한 행동을 하지 않을 수 있습니다.

actions(allowing choice)[4]

이는 s 에서 a 를 선택하였고 그 이후에는 정책π를 따라서 움직일 때 얻는 리턴의 기대값을 표현한 것입니다.

Model-free 는 실제로 액션을 해 보기 전까지는 보상을 얼마를 받을지도 모르고 어떤 상태로 이동하게 될 지 확률 분포도 전혀 모르는 상황입니다.[1] 그래서 테이블을 가지고 경험을 통해 업데이트를 하는 방법을 사용합니다. 큰수의 법칙으로 값을 점점 정확하게 만들 수 있는 방법이 있는데 이 방법은 몬테카를로 방법입니다.

s_0	s_1	s_2	s_3
s_4	s_5	s_6	s_7
s_8	s_9	s_{10}	s_{11}
s_{12}	s_{13}	s_{14}	종료

(0,0)	(0,0)	(0,0)	(0,0)
(1,-5)	(1,-4)	(1,-3)	(0,0)
(0,0)	(0,0)	(1,-2)	(0,0)
(0,0)	(0,0)	(1,-1)	종료

$s_0 \rightarrow s_4 \rightarrow s_5 \rightarrow s_6 \rightarrow s_7 \rightarrow s_{10} \rightarrow s_{14}$ →종료
오른쪽 표의 내용 $N(S_t)$: 방문 횟수, $V(S_t)$: 보상의 합

$v_\pi(s)$ 인 s 의 가치평가는 정책 π 하에 s 상태를 지나서 경험으로 얻을 수 있습니다. 처음 지나지 않고 여러 번 지날 수 있습니다. 처음방문만 한 것을 적용할 때는 최초 방문 몬테 카를로 정책 수정, 여러번 방문한 것을 적용할 때는 모든 방문 몬테 카를로 정책 수정을 사용합니다. 저는 최초 몬테 카를로 정책 수정을 사용하겠습니다.

$$v_\pi(s_t) \cong \frac{V(s_t)}{N(s_t)}$$

이 가치함수는 여러 개의 sequential data 에 대비에 배치 산술평균을 온라인 평균 기법[4]으로 변형해 incremental 한 모습으로 바꿉니다.

$$V(s) \leftarrow V(s) + \frac{1}{N(s)}(G_t - V(s))$$

---

[4] $\mu_k = \frac{1}{k}\sum_{j=1}^{k} x_j = \frac{1}{k}(x_k + \sum_{j=1}^{k-1} x_j) = \frac{1}{k}(x_k + (k-1)\mu_{k-1}) = \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})$

=기존의 알던 지식 + $\frac{1}{k}$*새롭게 관측된 지식

$V(s) \leftarrow V(s) + \alpha(G_t - V(s))$: $N(s)$를 정확히 세기 어려워 바꿉니다.

$v_\pi(s_t)$ 상태가치 함수는 원래 있는 정책 즉 action 들의 가치를 평가를 한 값입니다. 그리디 Greedy 한 정책은 탐욕적으로 행동한다는 뜻인데 상태가치 함수는 가치를 평가를 했을 뿐 앞으로 어떤 action 을 통한 행동을 해야 하는지에 대한 내용이 없습니다. 상태행동가치 함수는 action 을 상태가치함수에서 추가적으로 기록한 것으로 만들고 최고의 $q_\pi(s,a)$를 만들어 주는 action 을 정책 함수에 기록해 정책을 개선합니다.

하지만 현재 가치가 높아 그리디한 선택을 하는 것보다 미래에 더 나은 가치가 있는 상태가 있을 것에 대비해 다른 선택에 대한 여지를 남겨둬야 합니다. 이때 저는 엡실론 그리디를 사용합니다.

$$\epsilon - greedy\text{정책} : \pi^*(a|s) = \begin{cases} \frac{\epsilon}{|\mathcal{A}|} + 1 - \epsilon & \text{, if a } = \text{ argmax }_{a\in\mathcal{A}}\ Q_\pi\ (s,a) \\ \frac{\epsilon}{|\mathcal{A}|} & \text{, otherwise} \end{cases}$$

$|\mathcal{A}|$ = 가능한 액션의 개수

그리디한 선택 반대인 탐험적 선택을 할 경우 임의적[5] 선택인 랜덤 함수를 사용한 것에 의문점을 가집니다.

[5] 임의적 결정은 랜덤 결정과 반드시 동일하지는 않습니다. 예를 들어 1973년 석유 파동 동안 미국인들은 번호판이 홀수일 경우 홀수일에만, 번호판이 짝수일 경우 짝수일에만 휘발유를 구입할 수 있었습니다. 시스템은 잘 정의되었으며 제한 사항이 무작위가 아닙니다.(e.g,[5])