Q-learning

강화학습의 기초가 되는 초기 모델로써 원리 또한 단순

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0 R: 1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

위 그림을 토대로 원리를 알아보자.

GOAL : 0(R:1) 즉, Reward를 부여하는 지점에 도달하는 것

State : 각각의 타일을 의미

Q-Value : 각 State에 표기되는 값

기본 학습(Greedy)

1.모든 State의 Q-value가 아직 0인 상태이기 때문에 처음에는 Agent가 무작위 이동

2.이동 중 우연하게 GOAL State에 도착(바로 이전 State에 Reword를 부여)

3.2번 과정 반복

문제점 : MaxQ_value를 따르기 때문에 한번 경로가 짜여질 경우 더 이상 학습이 불가

ε-Greedy

1.모든 State의 Q-value가 아직 0인 상태이기 때문에 처음에는 Agent가 무작위 이동

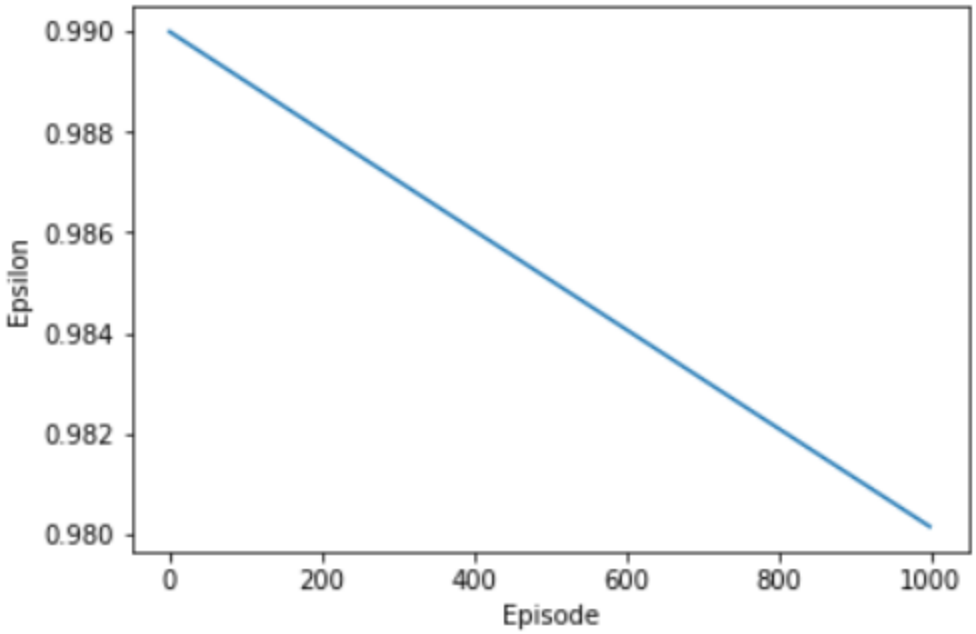
2.이동 중 우연하게 GOAL State에 도착(바로 이전 State에 Reword를 부여)

3.2번 과정 반복

4.ε값(0~1)에 따라서 일정 횟수는 탐험(무작위) 경로 이동

***Decaying ε-Greedy**

Episode가 진행할수록 ε값을 줄여나가며 학습하는 형태



Thomson Sampling (탐험 50% / 착취 50%)

탐험과 착취를 적절하게 섞어 학습하는 형태

(깊게 들어가면 의미가 다르긴 하지만 이정도로만 이해하고 넘어가자)

(유독 MAB 모델에서 강점을 띄는 알고리즘이다)

Hyper parameter

1.**ε - ε-Greedy의 경우 해당**

2.Discount Factor(Gamma) (0~1)

모든 알고리즘의 공통적인 문제점으로 Gamma 값을 설정하지 않을 경우 일정 Episode를 진행했을 경우 모든 State의 값이 동일한 상태가 되어 더 이상 Agent가 경로 설정을 하지 못하는 문제점이 존재함.

이러한 문제점을 해결하기 위해 State를 업데이트 시 Reward * Gamma를 해주어 각 State Q값의 차별을 두어 효율적인 경로를 찾게 해줄 수 있다.

3.Alpha - 새로운 미래를 얼마나 받아들일 것인지 설정하는 값 (0~1)

Gamma값과 함께 Reward에 곱해지는 값으로 값이 클수록 미래를 더 많이 수용함.