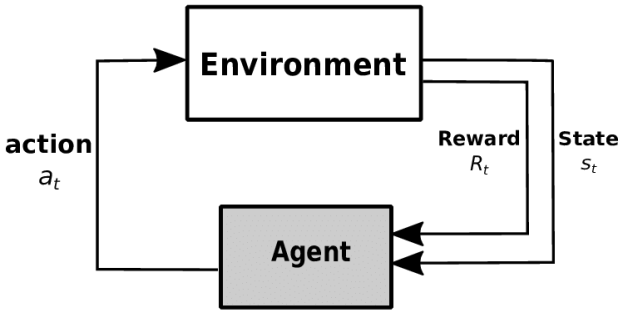
강화학습 맛보기

0. 목표

강화학습(Reinforcement Learning)이라는 키워드를 검색하면 다음과 같은 그림을 많이 보게 됩니다.



위 그림을 이해하고 이번주 내로 간단한 예시를 만들어 봅시다.

1. 강화학습이란?

강화학습이란 어떤 환경(environment)을 탐색하는 에이전트(Agent)가 현재의 상태(sate)에서 향후 기대되는 누적 보상값(reward)가 최대가 되도록 행동(Action)을 선택하는 정책을 찾는 것입니다. 저희가 고양이 마리오를 하는것과 비슷합니다. 사실 이렇게만 보면 이해가 어려우니 더 공부해 봅시다.

2. 강화학습 문제 정의

강화학습으로 문제를 풀기 위해서는 문제를 수학적으로 정의해야 합니다. 그리고 정의된 문제는 다음과 같은 구성 요소를 가집니다.

1. 상태 (State)
   현재 에이전트의 정보 (정적 + 동적)
2. 행동 (Action)
   에이전트가 어떤한 생태에서 취할 수 있는 행동
3. 보상 (Reward)
   에이전트가 학습 할 수 있는 유일한 정보, 자신이 했던 행동을 평가할 수 있는 지표
4. 정책 (Policy)
   상태에 대해 에이전트가 어떤 행동을 해야 하는지 정해놓는 것

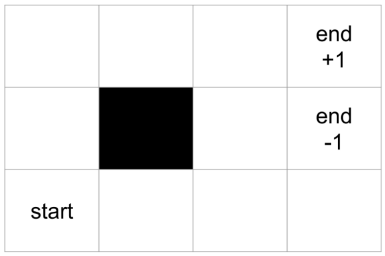
3. 마르코프 결정과정(Markov Decision Process, MDP)

순차적으로 행동을 계속 결정해야 하는 문제를 수학적으로 표현한 것이 마르코프 결정과정(Markov Decision Process, MDP)라고 합니다. 강화학습 문제를 풀기 위해 사용합니다.

- MDP의 구성 요소
  - 상태
  - 행동
  - 보상 함수
  - 상태 전이 확률

- 감가율

다음과 같은 grid가 있고(좌측 하단이 영점), 플레이어는 start에서 출발하여 한번에 한칸만 움직여서 +1 블럭까지 가는것이 목표라고 합시다. (검은색 블럭은 지나갈 수 없습니다.) 또한, 빛이 없기 때문에 한치앞도 보이지 않는다고 상상합시다.



여기서 MDP의 구성 요소를 정의해봅시다.
상태 집합 $S = \{s_1, s_2, \ldots, s_N\}$ 는 각 좌표가 되고, 행동 집합 $A = \{a_1, a_2, \ldots, a_N\}$는 {상,하,좌,우}가 됩니다.

- 상태
  $S = \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3)(3,1),(3,2),(3,3),(4,1),(4,2),(4,3)\}$
- 행동
  $A = \{east, west, south, north\}$
- 상태 전이 확률
  상태 전이 확률은 행동을 취했을 때 상태가 변할 확률입니다. 지금은 넘어갑시다.
- 보상 함수
  $R(\{4,3\}) = 1, R(\{4,1\}) = -1,\ R(\{else\}) = 0$
- 감가율
  $\gamma \in [1,0]$는 얻게 되는 보상이 미래에 얻게 될 보상보다 얼마나 더 중요하지 나타내는 값입니다. 자세한 설명은 뒤에서 하겠습니다.

이때, 시간 t에서의 상태를 S_t, 행동을 A_t 라고 표현합니다.

4. Q-Learning

Q-Learning은 Model없이 학습하는 강화학습 알고리즘입니다. Q-learning의 목표는 유한한 마르코프 결정과정에서 Agent가 특정 상황에서 특정 행동을 하라는 최적의 Policy를 배우는 것으로, 현재 상태부터 시작해서 모든 연속적인 단계들을 거쳤을 때, 전체 보상의 예측값을 극대화시킵니다. 'Q'라는 단어는 현재 상태에서 취한 행동의 보상에 대한 quality를 상징합니다. 이때 기본적으로 최적의 Policy는 현재 State에서 Q-value가 가장 높은 action을 취하는 것입니다. (저는 구현 할때 E-Greedy Algorithm(입실론 그리디 알고리즘)이라는 것을 사용하였는데, 이것도 공부해보시길 바랍니다)

Q-value

Q-Learning에서는 어떤 State에서 어떤 Action를 했을 때, 그 행동이 가지는 Value를 계산하는 Q-Value를 사용하는데, 이를 행동-가치 함수라고도 부릅니다. 이러한 행동 가치 함수는 Discounted Factor를 사용하여 특정 Action을 취했을 때, Episode가 종료되기까지 보상의 총합의 예측값을 계산합니다. 현재 상태로부터 Δt 시간이 흐른 후에 얻는 보상 $reward$은 $\gamma^{\Delta t}$만큼 할인되어 $reward \cdot \gamma^{\Delta t}$로 계산됩니다. 여기서 는 0~1사이의 값을 갖는 Discount Factor로 현재 얻는 보상이 미래에 얻는 보상보다 얼마나 더 중요한지 나타내는 수치입니다. 알고리즘은 각 상태-행동 쌍에 대해 $Q : S \times A \to \mathbb{R}$ 같은 $Q - Function$을 갖고 상세 식은 아래와 같습니다.

$$Q_\pi(s,a) = E_\pi[R_{t+1} + \gamma R_{t+2} + \gamma R_{t+2} + \cdots \mid S_t = s, A_t = a]$$

알고리즘

알고리즘은 매우 간단합니다.

매 $time\ step(t)$ 마다 Agent는 S_t에서 행동 A_t를 선택하게 되고, $reward$를 받으며 새로운 상태 S_t로 갱신됩니다.

$$Q(s_t,a_t) \leftarrow (1-\alpha)\cdot \underbrace{Q(s_t,a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1},a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} \right)$$

4. 예제

사실 RL을 깊게 이해하기 위해서는 더 많은 지식이 필요하지만 맛보기인 만큼 여기까지만 소개하겠습니다. (궁금하신 부분은 검색하시거나 질문하시길 바랍니다)
또 예제를 어떻게 마무리할까 고민하다가, 한번 풀어보시는게 좋을거 같아서 문제를 내어 드리겠습니다. (제공해드린 코드로 정답을 확인해보시길 바랍니다!)

> 다음과 같은 environment에서 Q-Learning을 수행한다고 하자. 각 gird position (x, y)가 state이고, action은 Up, Left, Right, Down 네가지이다. 그림에 있는 (+1), (-1)은 해당 state에 도달했을 때에 reward이며, 이외의 state에서 reward는 0이다. Start state(1,1)에서 시작하여 아래와 같은 action들이 차례로 수행된 후 학습된 모든 Q(s,a)를 그림에 표현하여라. (learning_rate = 0.8, discount_factor = 0.9 이며, 초기 Q(s,a)는 모두 0 이다.)
>
> 1) R-R-U-U-R
> 2) U-U-R-R-R
> 3) R-R-U-R
> 4) U-U-R-R-D-R
> 5) R-R-U-U-R
> 6) R-R-U-U-R