

Lab 02: Deep Reinforcement Learning

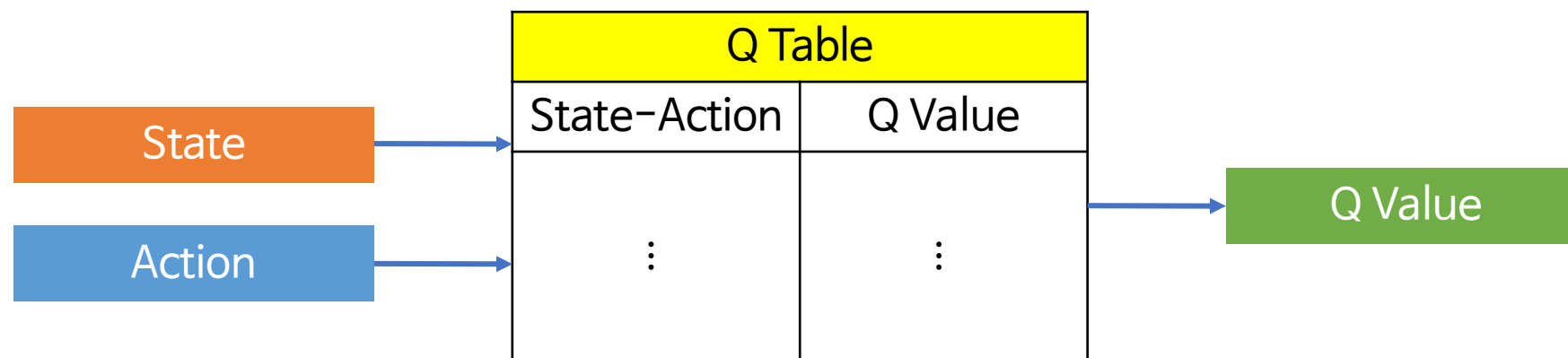
Junyeong Park

The limits of Q-Learning

Q-Learning은 테이블 형식의 강화학습이다.

- FrozenLake에서 전체 상태의 개수는 16개였다.
- 에이전트가 선택할 수 있는 행동은 4개였다.

→ 총 $16 \times 4 = 64$ 칸의 테이블로 큐함수를 표현할 수 있다.



The limits of Q-Learning

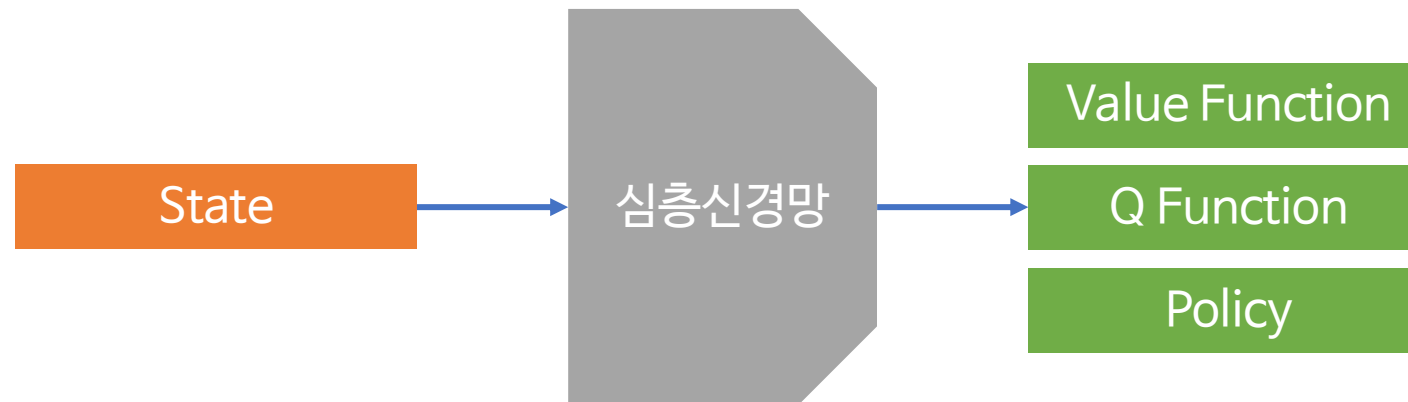
하지만 우리가 풀고자 하는 문제들은 단순하지 않다.



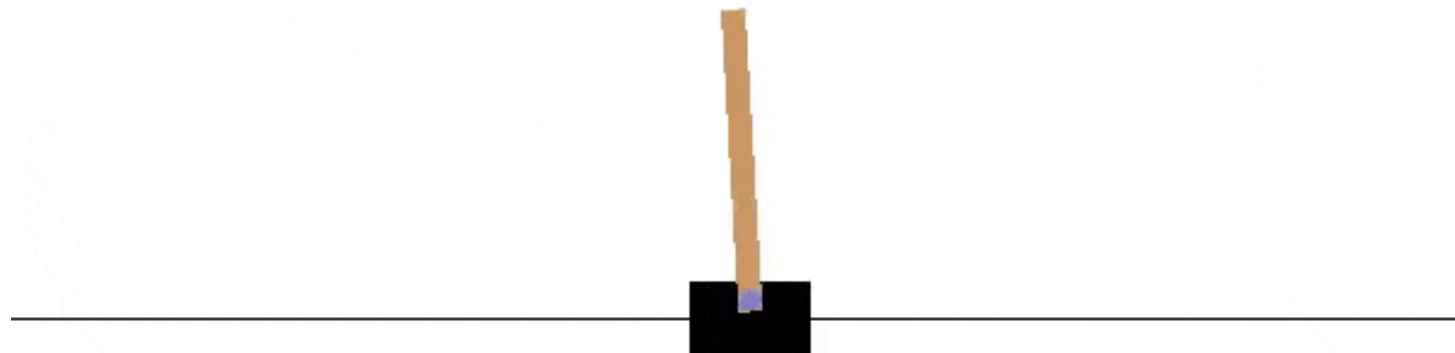
→ 테이블 형식으로 풀기는 힘들다.

Approximation Function

- 강화학습도 결국 함수를 만드는 과정이다.
 - 표현하기 복잡한 함수를 근사할 수 있다.
- 신경망을 통해 함수를 근사할 수 있다.



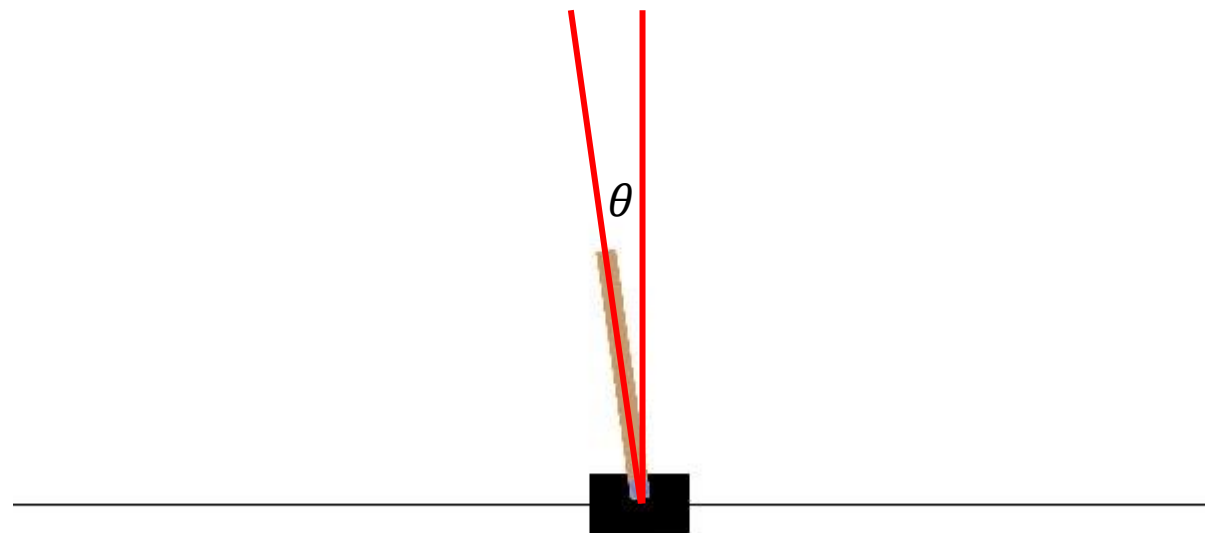
CartPole-v0



- 검은색 카트를 움직여 막대가 떨어지지 않도록 하는 것이 목표.
- 카트가 움직이지 않으면 중력에 의해 막대가 아래로 늘어뜨려진다.

CartPole-v0

- Observation : $[x, \theta, dx/dt, d\theta/dt]$
 - x : 트랙 상에서 카트의 위치
 - θ : 막대와 수직선이 이루는 각도
 - dx/dt : 카트의 속도
 - $d\theta/dt$: 막대의 각속도



CartPole-v0

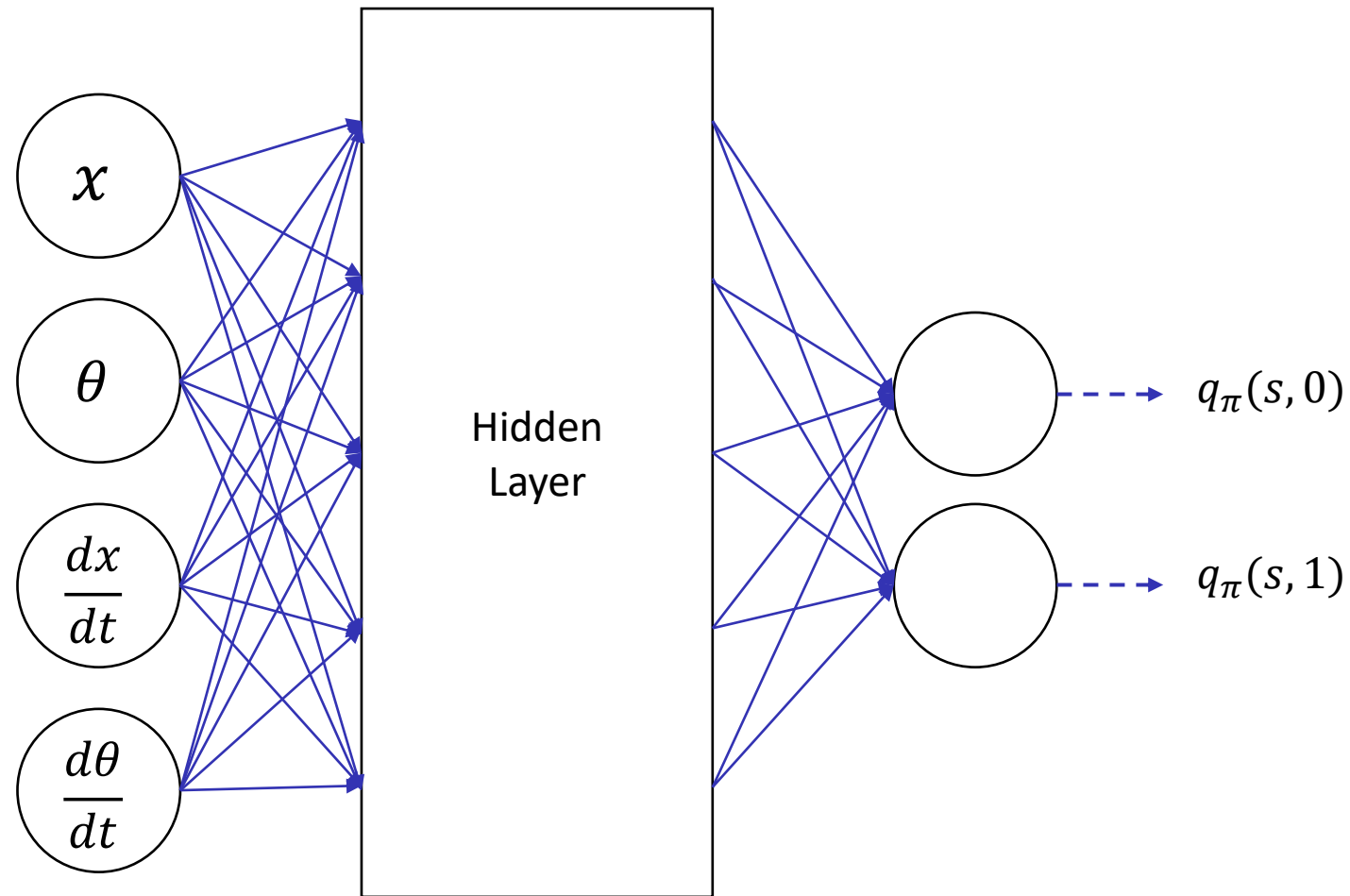
- 종료 조건
 - θ 가 15° 이상
 - 원점으로부터의 거리가 2.4 units 이상
- Action : 좌우로 이동 (0 or 1)
- Reward : 에피소드가 유지된 시간

CartPole-v0

- FrozenLake와는 다르게 observation이 연속적인 값이다.
- State-Action 쌍의 개수가 무한히 많다.
- 테이블 방식으로 풀기는 어렵다.

→ 신경망으로 큐 함수를 근사하자!

Network Architecture



Optimization

지난 시간 살펴본 Q-Learning 수식은 다음과 같다.

$$Q(s, a) \leftarrow Q(s, a) + \alpha [\underbrace{R_{t+1} + \gamma \max_{a'} Q(s', a')}_{\text{학습의 목표}} - Q(s, a)]$$

학습의 목표

- MSE (Mean Square Error)를 통해 신경망을 학습시킬 수 있다.

$$\rightarrow L = \left(R_{t+1} + \gamma \max_{a'} Q(s', a') - Q(s, a) \right)^2$$

Practice



Google Colab 링크

<http://bitly.kr/fsDHmBQrlo>

Policy-based Reinforcement Learning

지금까지의 강화학습 알고리즘은 가치 함수를 바탕으로 동작

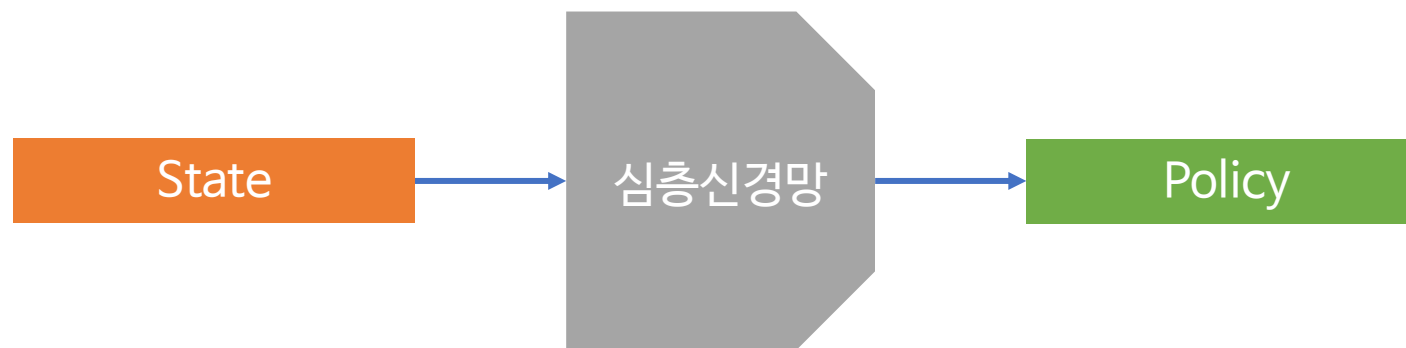
→ 가치 기반 강화학습(Value-based Reinforcement Learning)

정책을 기반으로 한 강화학습 알고리즘도 생각해볼 수 있다.

→ 정책 기반 강화학습(Policy-based Reinforcement Learning)

Policy-based Reinforcement Learning

- 상태에 따라 바로 행동을 선택한다
→ 가치함수를 토대로 행동을 선택하지 않는다.
- 정책을 직접적으로 근사한다.



- 신경망으로 정책을 근사하고, 신경망의 출력은 정책이 된다.

Advantages and Disadvantages

- 장점
 - value-based 방식보다 수렴이 더 잘 된다.
 - 가능한 action이 여러 개이거나 action이 연속적인 경우에도 효과적이다.
 - 확률적인 정책을 배울 수 있다. (ex 가위바위보)
- 단점
 - local optimum에 빠질 수 있다.
 - policy를 평가하는 게 비효율적이다. (반환값을 계산 해야한다)
 - variance가 높다.

Policy-based Reinforcement Learning

신경망으로 근사된 정책은 다음과 같이 표현할 수 있다.

$$\pi_{\theta}(a|s)$$

- θ 는 정책 신경망의 가중치다.
- 목표함수는 $J(\theta)$ 로 표현할 수 있다.
- 정책을 근사하는 신경망의 출력층은 Softmax를 사용한다.
→ 가장 최적의 행동을 선택하는 분류 문제로 생각할 수 있다.

Policy Gradient

강화학습의 목표는 누적 보상을 최대로 하는 최적 정책을 찾는 것이다.

따라서 정책 기반 강화학습의 목표를 수식으로 표현하면 다음과 같다.

$$\text{Maximize } J(\theta)$$

목표함수 $J(\theta)$ 의 최대화는 미분을 통해 미분한 값에 따라 업데이트 하면 된다.

→ 일반적인 경사하강법의 반대로 "경사상승법"이라고 한다.

Policy Gradient

미분을 통해 정책 신경망을 업데이트 해보자.

어느 시간 t 에서 신경망의 가중치 θ_t 에서 다음과 같이 θ_{t+1} 을 구할 수 있다.

$$\theta_{t+1} = \theta_t + \alpha \nabla_{\theta} J(\theta)$$

목표함수는 $J(\theta) = v_{\pi}(s)$ 이므로 목표함수의 미분은 다음과 같다.

$$\nabla_{\theta} J(\theta) = \nabla_{\theta} v_{\pi}(s)$$

Policy Gradient

가치함수의 정의를 이용하면 최종적으로 가중치의 업데이트 식은 다음과 같다.

$$\theta_{t+1} = \theta_t + \alpha [\nabla_{\theta} \log \pi_{\theta}(a|s) q_{\pi}(s, a)]$$

하지만 에이전트에 가치함수나 큐 함수가 없기 때문에 $q_{\pi}(s, a)$ 를 구할 수 없다.

REINFORCE

가치함수는 반환값(Return)의 기댓값이다.

→ 큐 함수를 반환값 G_t 로 대체할 수 있다. 이를 REINFORCE 알고리즘이라 한다.

REINFORCE 알고리즘의 업데이트 식은 다음과 같다.

$$\theta_{t+1} = \theta_t + \alpha [\nabla_{\theta} \log \pi_{\theta}(a|s) G_t]$$

- $\log \pi_{\theta}(a|s)$ 는 실제로 한 행동을 정답으로 둔 것이다.
- 하지만 잘못된 선택을 할 수 있어 반환값을 곱해준다.
→ 부정적인 보상을 받게 된다면 그 행동을 선택할 확률을 낮춘다.

Practice



Google Colab 링크

<http://bitly.kr/etMGCTcKsi>