# 심층 강화학습 인 액션

## Ch5: 행위자-비평자 모형

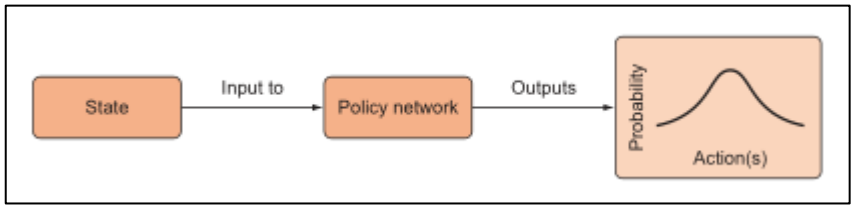
최우진

2021/08/13

# 0. 목차

# 1. 분산 이익 행위자-비평자(distributed advantage actor-criti

- actor-critic 알고리즘: REINFORCE의 장점 + 심층 Q 신경망의 장점
- 정책망(policy network)




- 수익(R): 에피소드 보상들의 가중합

$$R = \sum_t \gamma_t \cdot r_t \quad (r_t\text{: 보상}, \gamma_t\text{: 감가율})$$

- 손실함수:

$$loss = -\log\big(P(a|S)\big) \cdot R$$

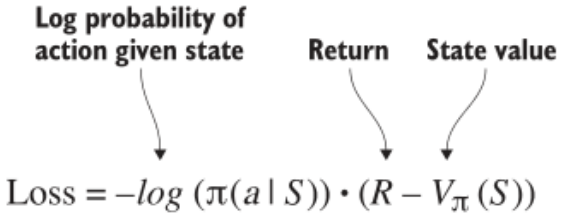
# 1. 분산 이익 행위자-비평자(distributed advantage actor-criti

- 심층 Q 신경망은 에피소드마다 매개 변수를 갱신하는 것이 아니라, 일정한 간격마다 매개 변수를 갱신하는 '온라인' 방식을 사용했다.
- 단, 학습의 효율과 안정성을 위해 '경험 재현 기법'을 사용했다.

- 분산 이익 행위자-비평자
  - 1) 심층 Q 신경망처럼 온라인 학습
  - 2) 경험 재현이 필요하지 않다.
  - 3) 동작들에 관한 확률분포에서 직접 동작을 선택하는 정책 기울기 방법(따로 정책을 정할 필요가 없음)

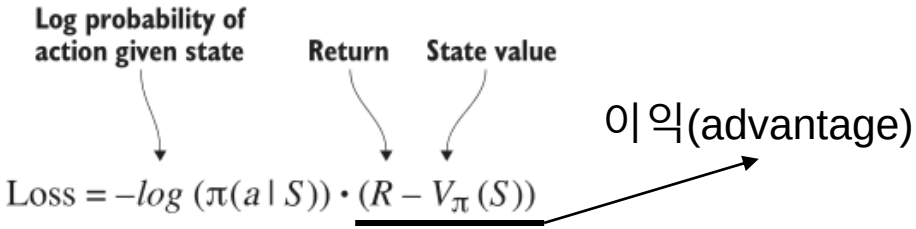
# 2. 가치 함수와 정책 함수의 결합

- 가치-정책 학습 알고리즘
- 정책망 학습 모형의 안정성을 위해 아래 두가지 문제를 해결해야한다.
  - 1) 갱신 빈도를 높여서 표본 효율성을 개선해야 한다.
  - 2) 모형 매개변수 갱신에 사용할 보상들의 분산을 줄여야 한다.
- 수집한 표본의 크기가 커질수록 보상의 분산이 작아지는 경향이 있기 때문에 위에 두 문제는 연관되어 있다.
- 결합된 가치-정책 알고리즘의 핵심은 가치 학습 모형을 이용해서 정책망의 훈련에 사용되는 보상들의 분산을 줄이는 것이다.
- 보상들의 분산을 줄인 손실함수



# 2. 가치 함수와 정책 함수의 결합

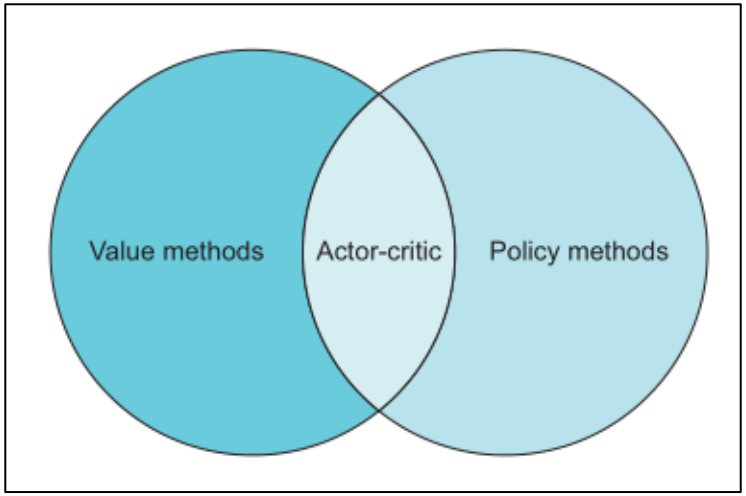
- 보상들의 분산을 줄인 손실함수




- 이익: 동작의 가치가 기대했던 것보다 얼마나 더 좋은지를 나타냄
- 예1) 상태 S1에서 동작 1을 선택해서 +10의 보상을 얻었고, S1의 가치는 +5인 경우
  => 이익(R - V) = 10 - 5 = 5이므로, 해당 보상이 기대 보상보다 좋다는 것을 알 수 있다.
- 예2) 상태 S2에서 동작 1을 선택해서 +10의 보상을 얻었고, S2의 가치는 +15인 경우
  => 이익(R - V) = 10 - 15 = -5, 위와 같은 보상을 얻었지만 상대적으로 나쁜 보상이다.

# 2. 가치 함수와 정책 함수의 결합

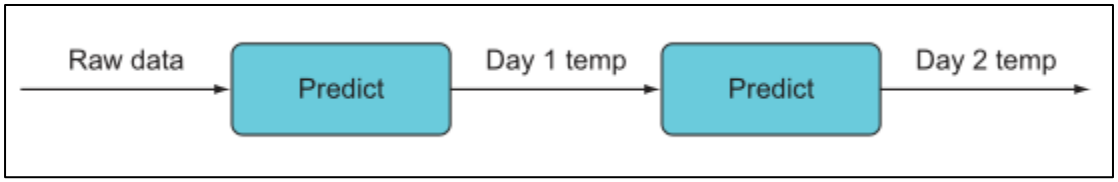
- 이런 종류의 알고리즘을 '행위자-비평자' 방법이라 부른다.
- 정책(동작 선택): '행위자', 가치함수: '비평자'
- Q 학습 알고리즘: 1) 동작 가치를 배워서 동작을 선택하는 가치 방법과 2) 동작을 선택하는 정책 자체를 학습하는 정책 방법(REINFORCE 등)
- 행위자-비평자: 가치 방법과 정책 방법의 교집합인 방법



# 2. 가치 함수와 정책 함수의 결합

- 정책망의 손실함수는 에피소드 끝에서 수집된 보상들에 의존한다.
- Gridworld에서 기본 동작은 -1의 보상을 받고 에피소드 마지막에 $\pm 10$을 받기 때문에 단순한 정책 기울기 방법은 동작을 충분히 강화하지 못한다.
- 하지만, Q 신경망은 '부트스트래핑(bootstrapping)' 덕분에 보상이 희박해도 적절한 Q 가치들을 학습할 수 있다.

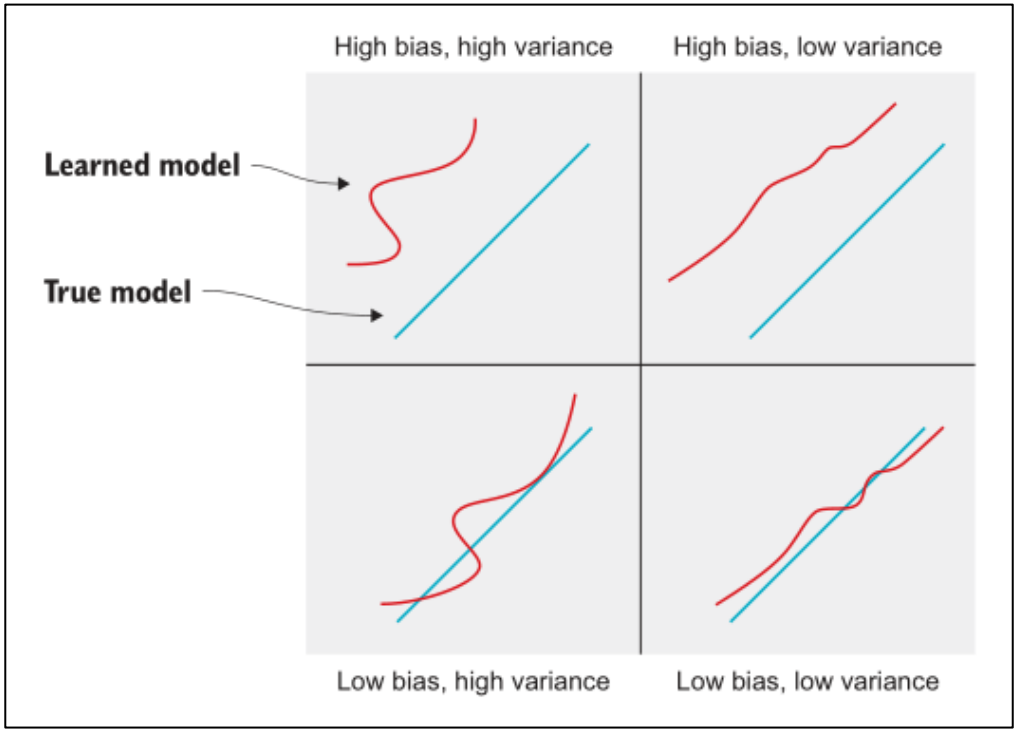
- 부트스트래핑: 예측 결과로부터 또 다른 결과를 예측하는 것




- 예) 2일 후의 온도를 예측하는 방법: 내일 기온을 예측하고 그 값으로 모레의 기온을 예측

# 2. 가치 함수와 정책 함수의 결합

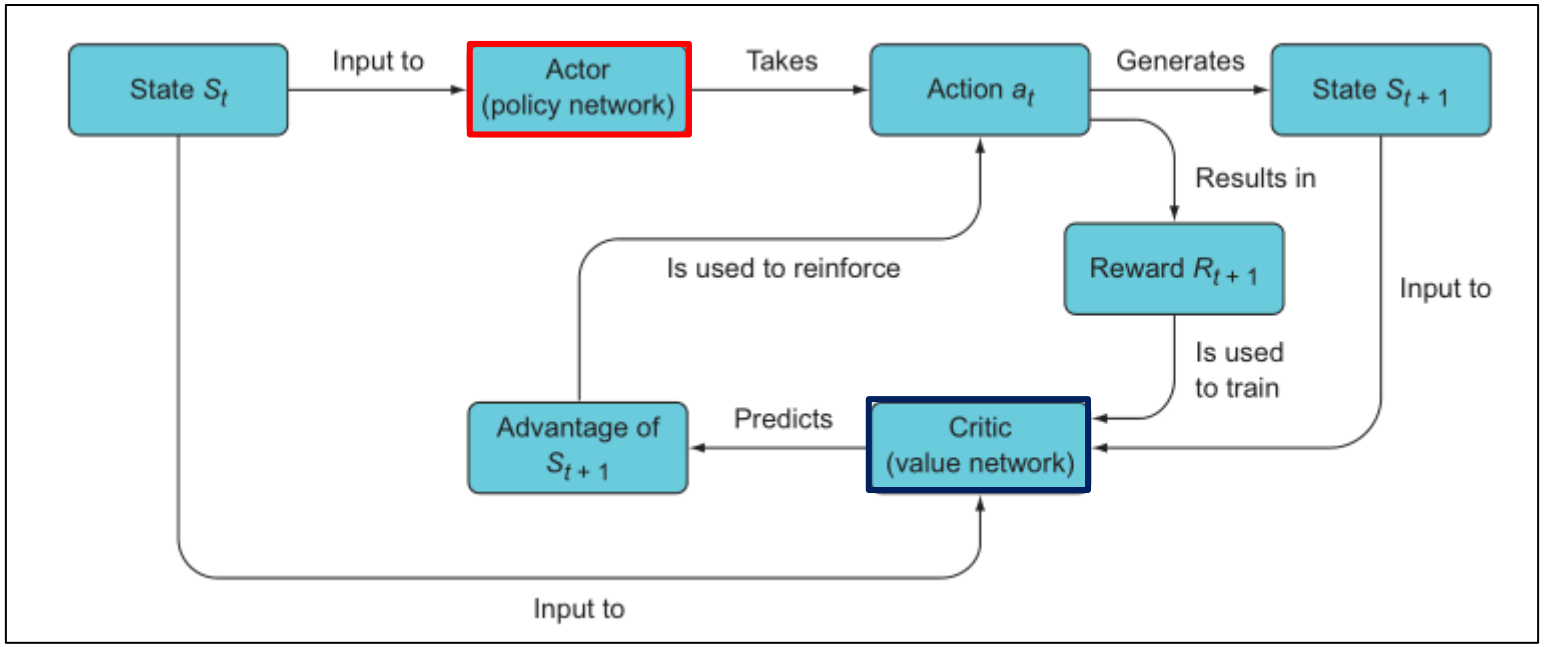
- 편향(bias): 어떤 대상에 대한 예측이 그 참값과 얼마나 벗어났는지에 대한 오차
- 분산(variance): 값들이 얼마나 '퍼져 있는지'(혹은 '다양한지')를 나타냄



# 2. 가치 함수와 정책 함수의 결합

- 편향이 크고 분산이 작은 '가치 예측'과 편향이 작고 분산이 큰 '정책 예측'을 결합하여 편향과 분산이 적당한 행위자-비평자 모형을 만들 수 있다.

- 행위자-비평자 모형



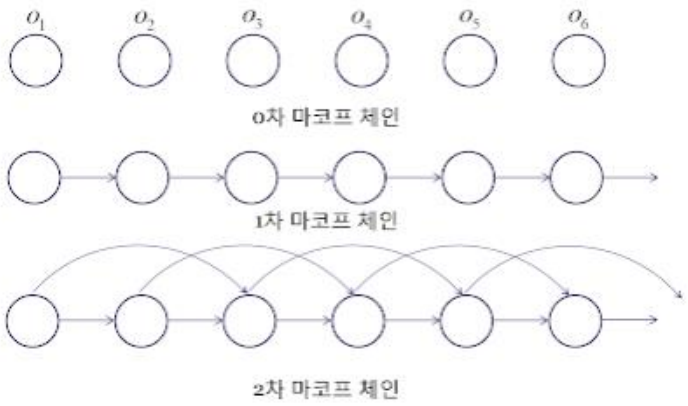
# 3. 분산 훈련

- 배치 훈련: 전체 훈련 데이터셋에서 무작위로 부분집합을 추출하고 해당 부분집합 전체에 대한 손실함수를 계산해서 역전파와 경사 하강법을 모델에 실행하는 것
- 경험 재현 기법은 강화학습 문제의 환경과 에이전트가 '마르코프 성질'을 만족하는 경우에만 사용할 수 있다.
- 마르코프 성질

$r=0$이면, 0차 마코프 체인 $P(o_t \mid o_{t-1}o_{t-1}...o_1) = P(o_t)$

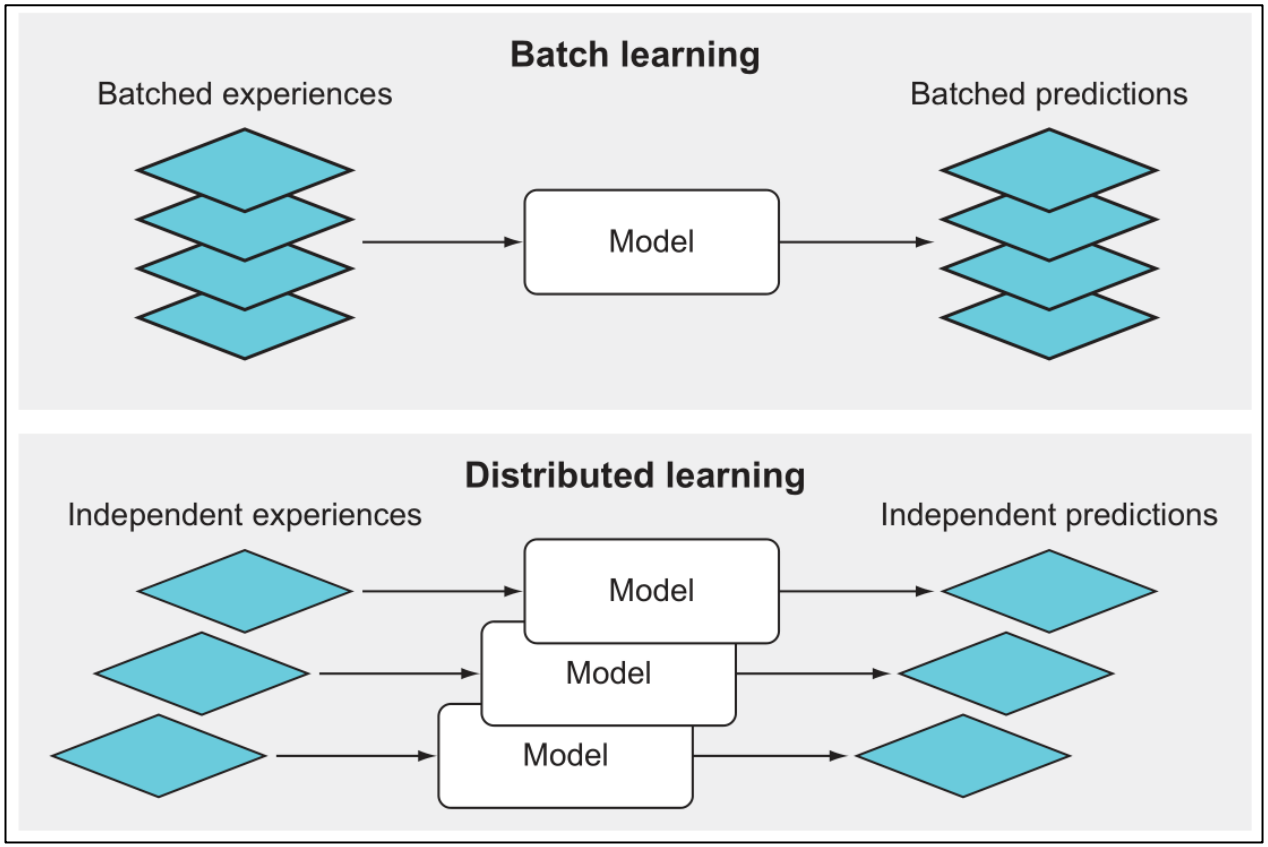
$r=1$이면, 1차 마코프체인 $P(o_t \mid o_{t-1}o_{t-1}...o_1) = P(o_t \mid o_{t-1})$

$r=2$이면, 2차 마코프 체인 $P(o_t \mid o_{t-1}o_{t-1}...o_1) = P(o_t \mid o_{t-1}, o_{t-2})$



# 3. 분산 훈련

- 배치 훈련, 분산 훈련



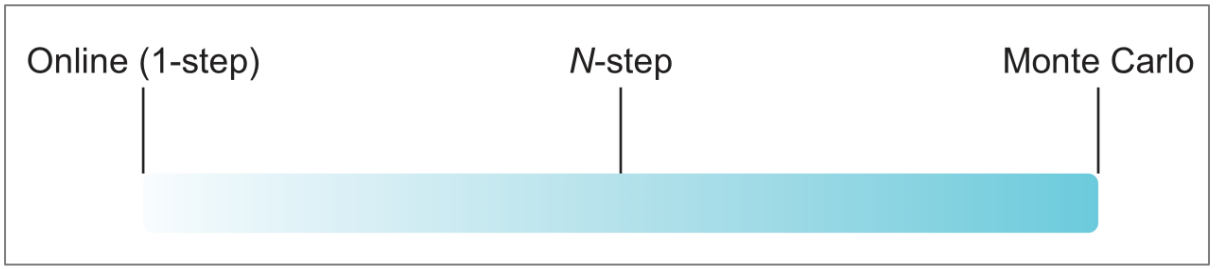
# 4. 이익 행위자-비평자

- 온라인 이익 행위자-비평자의 의사코드

```
gamma = 0.9
for i in epochs:
    state = environment.get_state()
    value = critic(state)
    policy = actor(state)
    action = policy.sample()
    next_state, reward = environment.take_action(action)
    value_next = critic(next_state)
    advantage = reward + (gamma * value_next – value)
    loss = -1 * policy.logprob(action) * advantage
    minimize(loss)
```

# 4. 이익 행위자-비평자

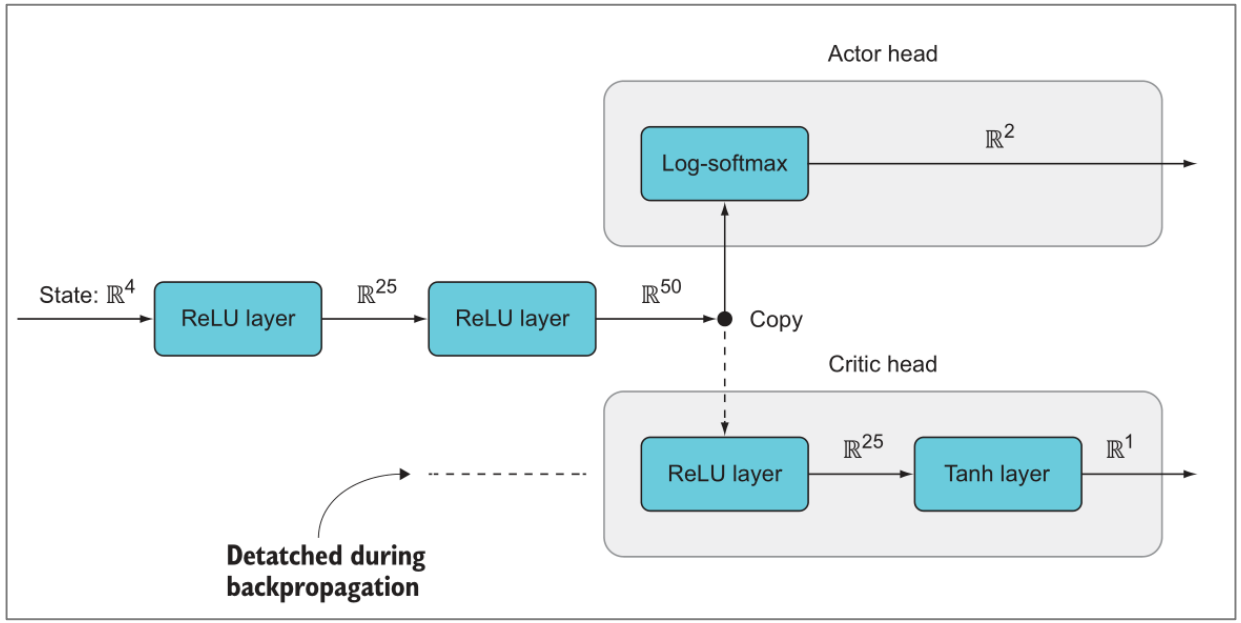
- N-단계 학습




- Online(1-step)은 1-step마다 학습을 진행
- N-step은 N-step마다 학습을 진행
- Monte Carlo는 모든 에피소드가 끝나야 학습 진행

# 4. 이익 행위자-비평자

- CartPole 행위자-비평자 모형(이중 출력 신경망)



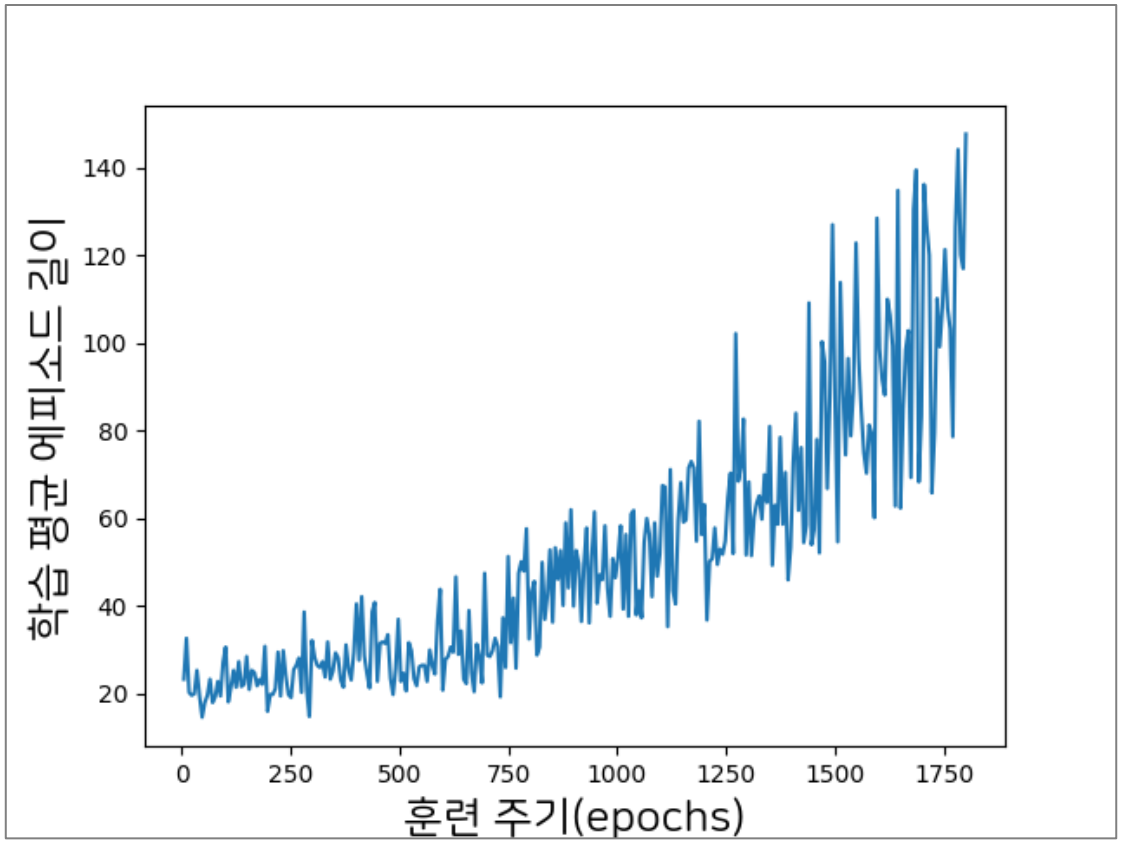

- 입력: CartPole 상태(4차원 실수 벡터)
- 출력: (행위자) 두 가지 동작에 관한 이산 확률분포인 2차원 벡터
  (비평자) 상태 가치인 하나의 값

# 4. 이익 행위자-비평자

- 행위자-비평자 전체적인 과정
- 1. 행위자-비평자 모형 생성
- 2. 에피소드 실행
  - 초매개변수 $\gamma$ 정의 및 새 에피소드 시작
  - 현재 상태 $s_t$의 가치 $v(s_t)$를 계산해서 목록에 저장
  - 확률분포 $\pi(s_t)$를 계산해서 목록에 저장하고, 동작 $a_t$를 추출해서 실행
  - 새 상태 $s_{t+1}$과 보상 $r_{t+1}$을 얻고 보상을 목록에 저장
- 3. 훈련
  - 수익을 R = 0으로 초기화, 보상 목록을 거꾸로 훑으면서 수익 $R = r_i + \gamma R$을 계산
  - 행위자 손실 $-1 \cdot \gamma_t(R - v(s_t)) \cdot \pi(a|s)$를 최소화
  - 비평자 손실 $(R - v)^2$을 최소화
- 4. 전체 에피소드 횟수를 넘지 않았으면 3번부터 다시 반복

# 4. 이익 행위자-비평자

- CartPole 훈련 평가



# 5. N-단계 행위자-비평자

- 앞 예제는 몬테카를로 방식이지만, N-단계 학습은 N-단계마다 매개변수를 갱신한다.
- 1-단계 학습은 부트스트래핑을 매번 적용하기 때문에 편향이 커질 수 있다.

# Ch5: 행위자-비평자 모형

감사합니다