2021-2 한양대학교 HAI

강화학습 부트캠프

Chris Ohk
utilForever@gmail.com

오늘 배울 내용

- Actor-Critic Extensions
  - Basic Actor-Critic
  - Advantage Actor-Critic (A2C)
  - Asynchronous Advantage Actor-Critic (A3C)
    - Data Parallelism
    - Gradient Parallelism

Basic Actor-Critic

- REINFORCE의 장점과 DQN의 장점을 조합해 행동을 취하는 동작과 정책을 비판하는 동작을 교대로 수행하는 알고리즘
- 액터(Actor) : 직접적인 보상에 기반해 정책을 결정한다.
- 크리틱(Critic) : 환경 상태의 가치 추정에 비해 우리의 정책이 얼마나 좋은지 알려준다.
- 정책 반복의 구조를 사용해 학습한다.

$$\pi_0 \rightarrow v_{\pi_0} \rightarrow \pi_1 \rightarrow v_{\pi_1} \rightarrow \pi_2 \rightarrow \cdots \rightarrow \pi_* \rightarrow v_*$$

  - 정책 평가 : 가치 신경망을 이용해 정책을 평가
  - 정책 발전 : 정책 신경망의 업데이트
- 오류 함수

$$L_{\text{actor}} = \big(R + \gamma v(s') - v(s)\big) \log \pi_\theta(a|s)$$

$$L_{\text{critic}} = \big(R + \gamma v(s') - v(s)\big)^2$$

Advantage AC (A2C)

- Asynchronous Methods for Deep Reinforcement Learning (Mnih, 2016)
- Dueling DQN에서 다뤘던 Advantage 개념을 Actor-Critic에 사용한다.

$$Q(s_t, a_t) = V(s_t) + A(a_t, s_t)$$

- 기존 Policy Gradient 식

$$\nabla_\theta J(\theta) \approx E_{\pi_\theta}[Q(s_t, a_t)\nabla_\theta \log \pi(a_t|s_t;\theta)]$$

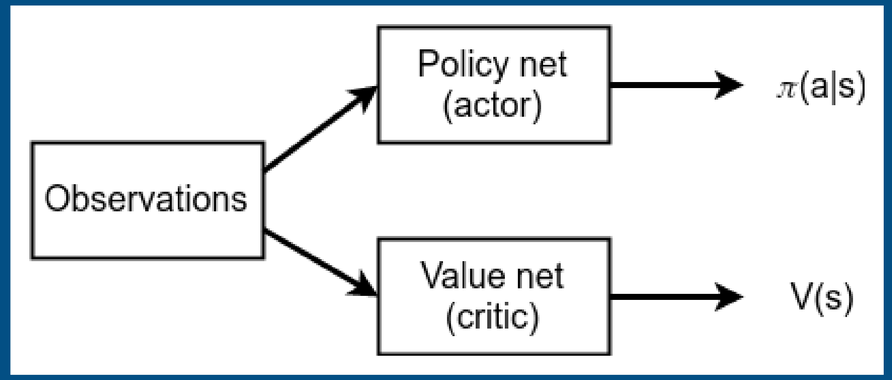
- A2C의 Policy Gradient 식

$$\nabla_\theta J(\theta) \approx E_{\pi_\theta}\big[\big(Q(s_t, a_t) - V(s_t)\big)\nabla_\theta \log \pi(a_t|s_t;\theta)\big]$$

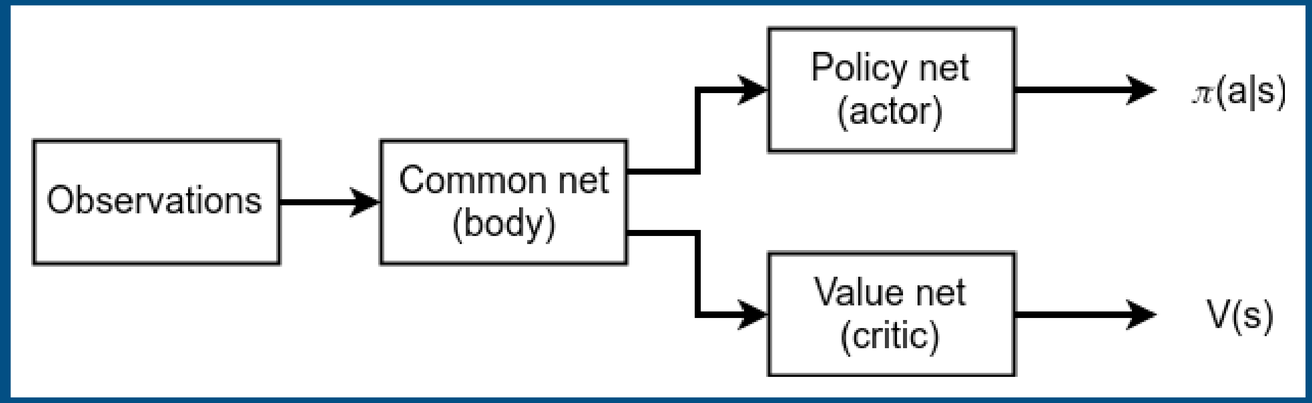
$$\approx E_{\pi_\theta}[A(s_t, a_t)\nabla_\theta \log \pi(a_t|s_t;\theta)]$$

Advantage AC (A2C)

- A2C 구조



- 실제로는 정책 신경망과 가치 신경망의 구조가 일부 겹치기 때문에 결합할 수 있다.

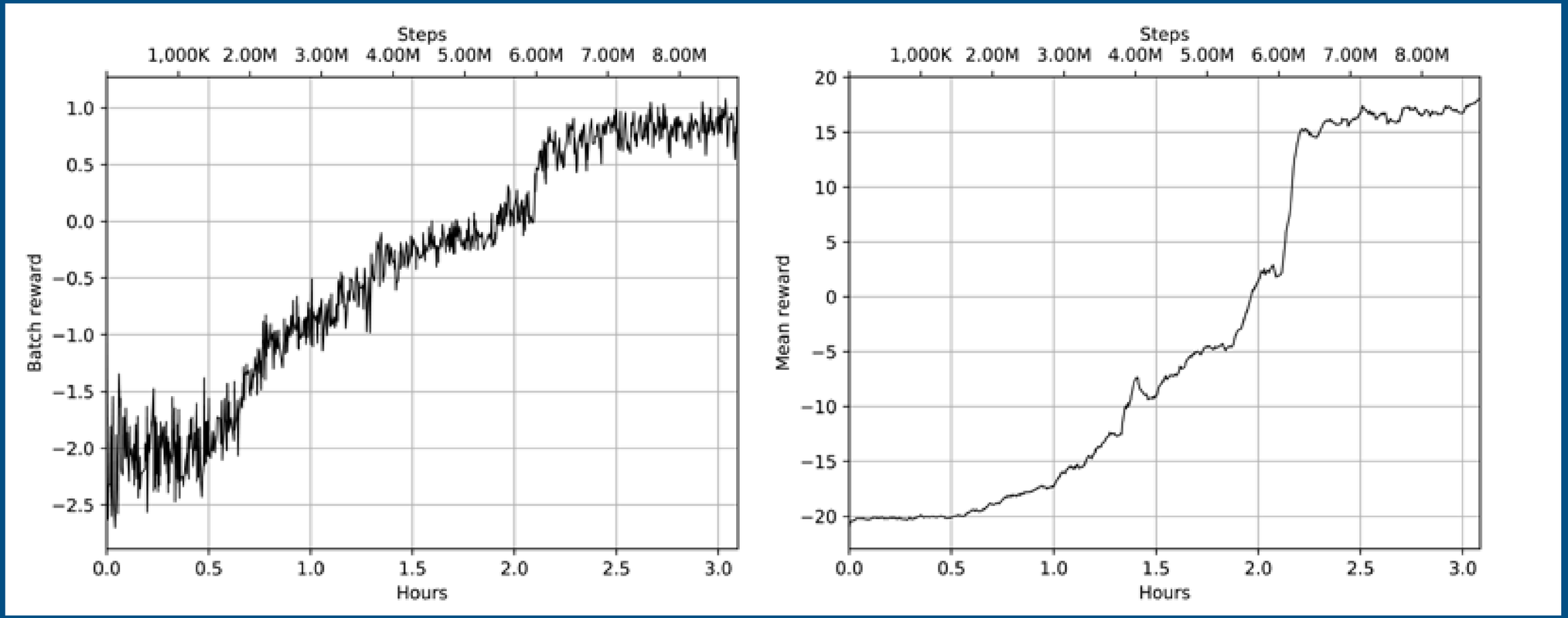


Advantage AC (A2C)

- A2C의 훈련 과정
  1. 신경망 매개 변수 θ를 임의의 값으로 초기화한다.
  2. 현재 정책 π_θ를 사용해 환경에서 N 스텝 동안 진행한다. 이때 상태 s_t, 행동 a_t, 보상 r_t를 저장한다.
  3. 만약 에피소드 끝에 도달했다면 $R = 0$이고, 아니면 $R = V_\theta(s_t)$이다.
  4. 반복 과정을 수행한다. $(i = t - 1 \dots t_{start})$
     1. 값을 갱신한다. $(R \leftarrow r_i + \gamma R)$
     2. 정책 그레디언트 값을 누적시킨다. $(\partial\theta_\pi \leftarrow \partial\theta_\pi + \nabla_\theta \log \pi_\theta(a_i|s_i)\left(R - V_\theta(s_i)\right))$
     3. 가치 그레디언트 값을 누적시킨다. $(\partial\theta_v \leftarrow \partial\theta_v + \frac{\partial\left(R - V_\theta(s_i)\right)^2}{\partial\theta_v})$
  5. 누적된 그레디언트 값들을 사용해 신경망 매개 변수들을 갱신한다.
  6. 수렴할 때까지 2~5를 반복한다.

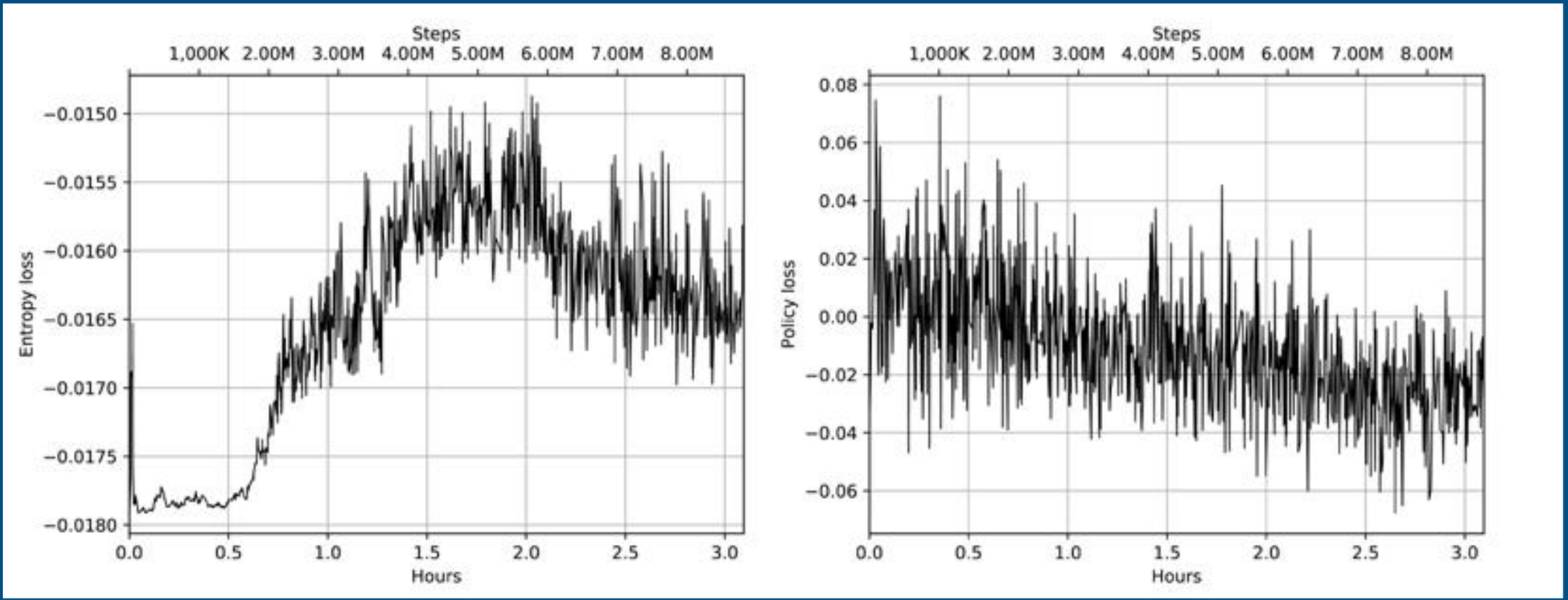
Advantage AC (A2C)

- Mean Batch Value and Average Training Reward



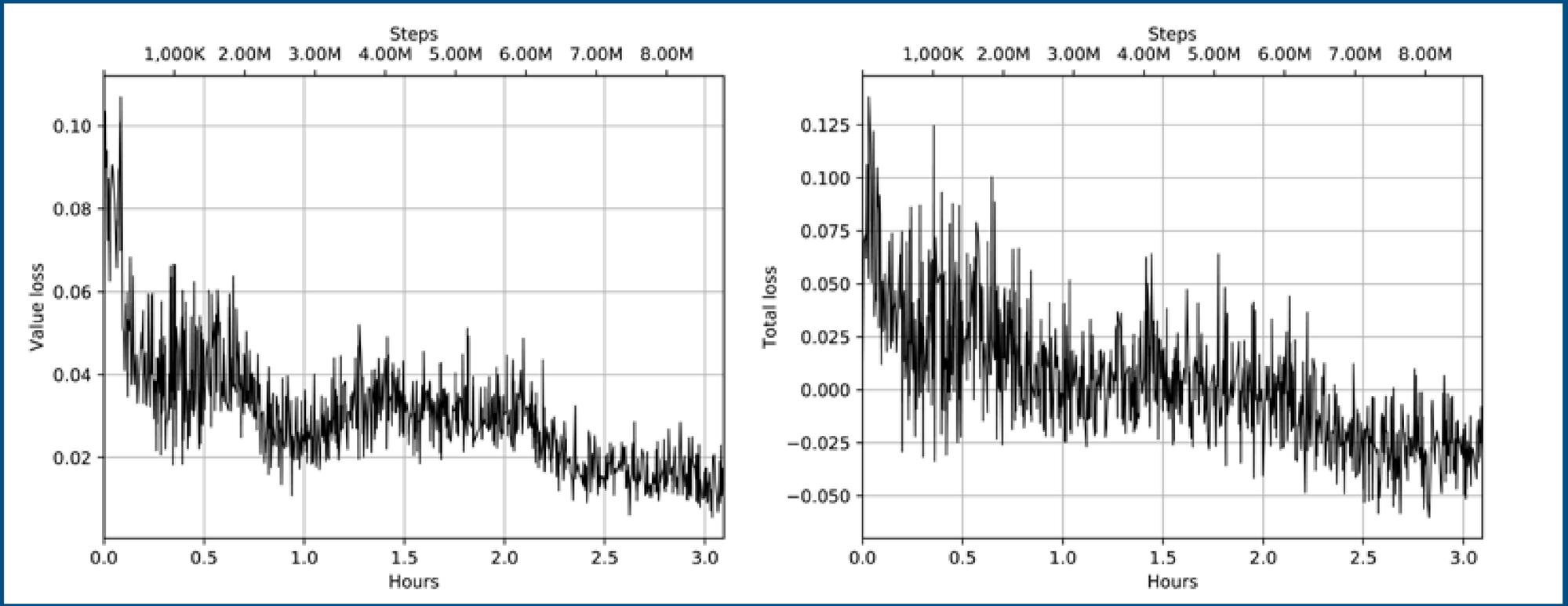
Advantage AC (A2C)

- Entropy Loss and Policy Loss during The Training



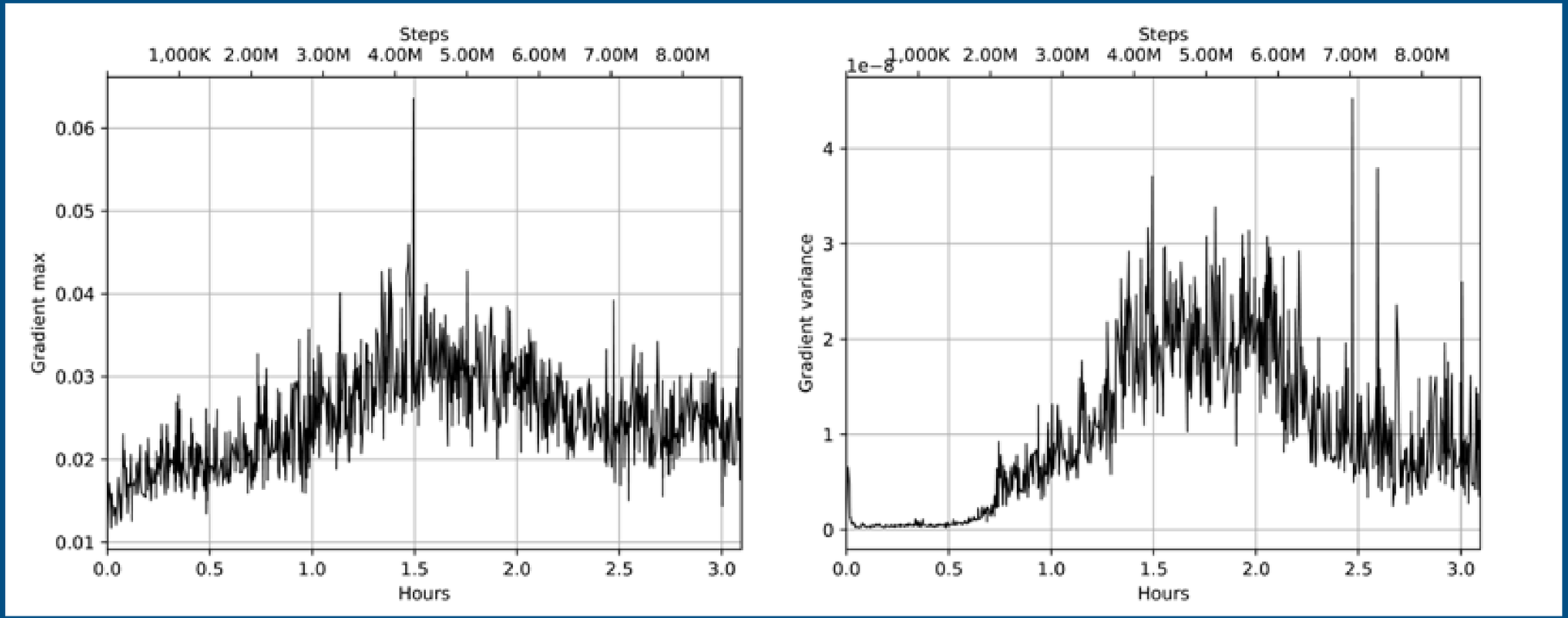
Advantage AC (A2C)

- Value Loss and Total Loss



Advantage AC (A2C)

- Maximum of Gradients and Variance of Gradients

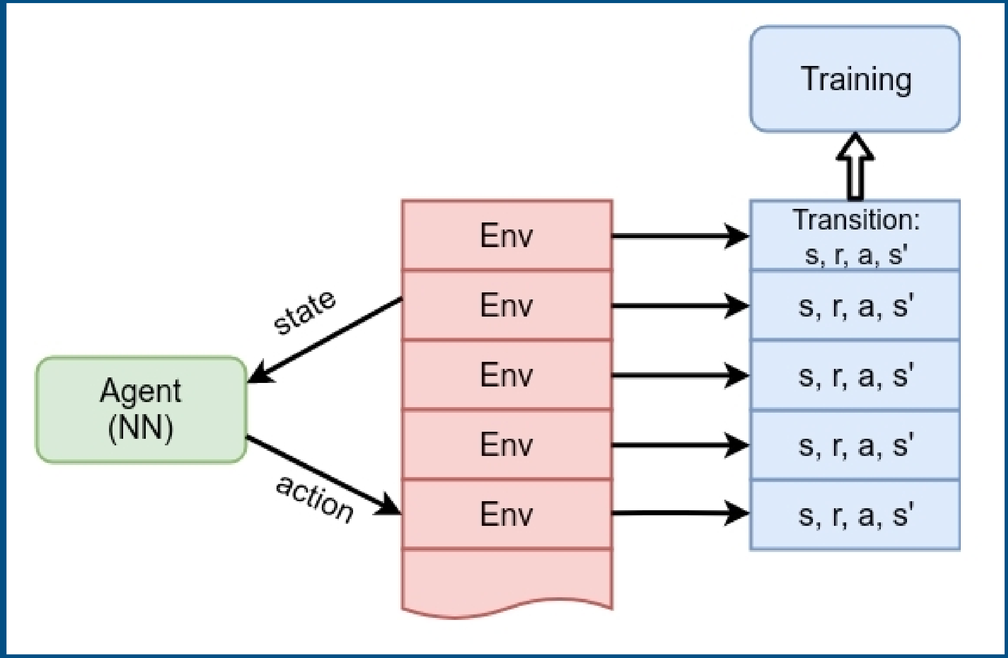


Asynchronous Advantage AC (A3C)

- Asynchronous Methods for Deep Reinforcement Learning (Mnih, 2016)
- DQN이 이룬 점
  - 게임 화면을 상태 입력으로 받아서 학습 한다. (CNN)
  - 사람보다 게임 플레이를 더 잘하는 에이전트를 만들 수 있다.
  - 샘플들 사이의 강한 상관관계를 리플레이 메모리로 해결한다.
- DQN이 부족한 점
  - 많은 메모리를 사용한다.
  - 학습 속도가 느리다.
  - 학습 과정이 불안정하다. (가치 함수에 대해 그리디 정책을 따르기 때문)

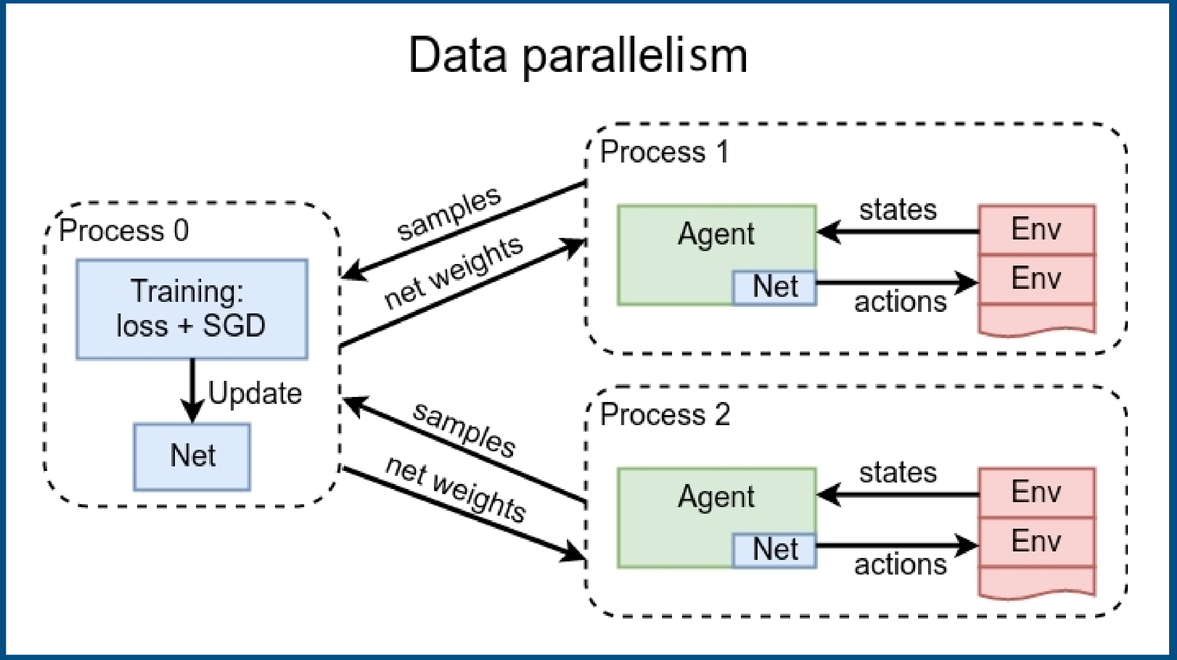
Asynchronous Advantage AC (A3C)

- A3C = A2C + Asynchronous (비동기)
  - 샘플 사이의 상관관계를 비동기 업데이트로 해결한다.
  - 리플레이 메모리를 사용하지 않는다.
  - 폴리시 그레디언트 알고리즘을 사용할 수 있다. (Actor-Critic)
  - 상대적으로 빠른 학습 속도를 갖는다. (여러 에이전트가 환경과 상호작용하기 때문)



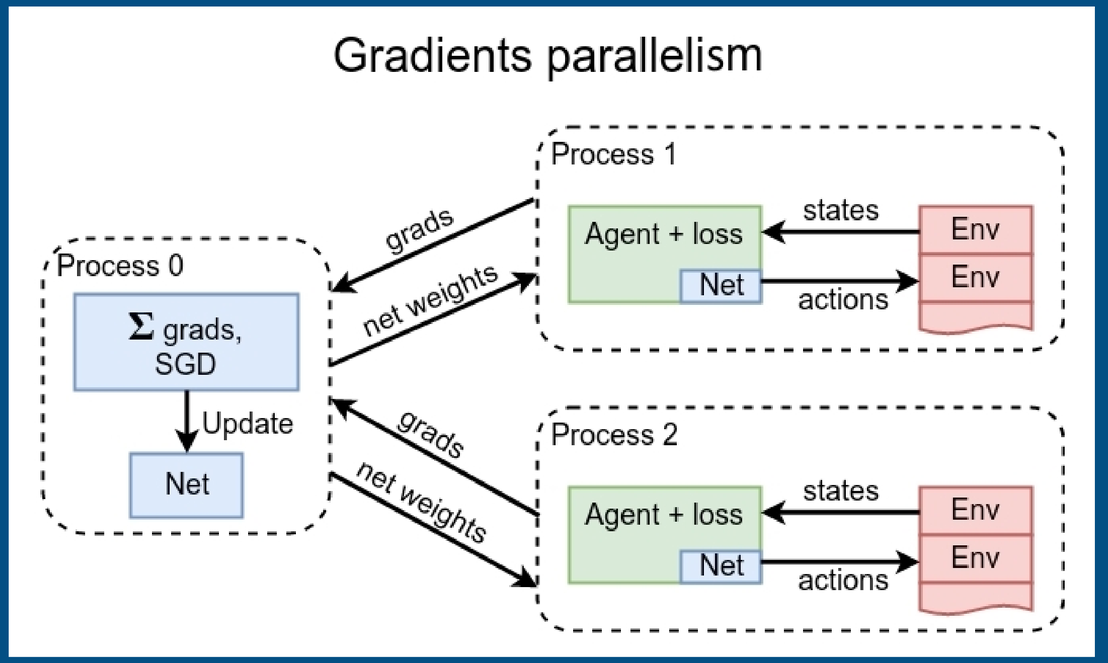
Asynchronous Advantage AC (A3C)

- 두 가지 접근 방식
  - 데이터 병렬화 : 각 에이전트에서 샘플(s, r, a, s')을 전달한다. 그러면 글로벌 신경망에 모든 샘플들이 모이게 되는데, 여기서 글로벌 신경망에서 Loss와 SGD 갱신을 수행한다. 그 뒤 각 에이전트에서 샘플을 보낼 때 갱신된 가중치를 전달한다.



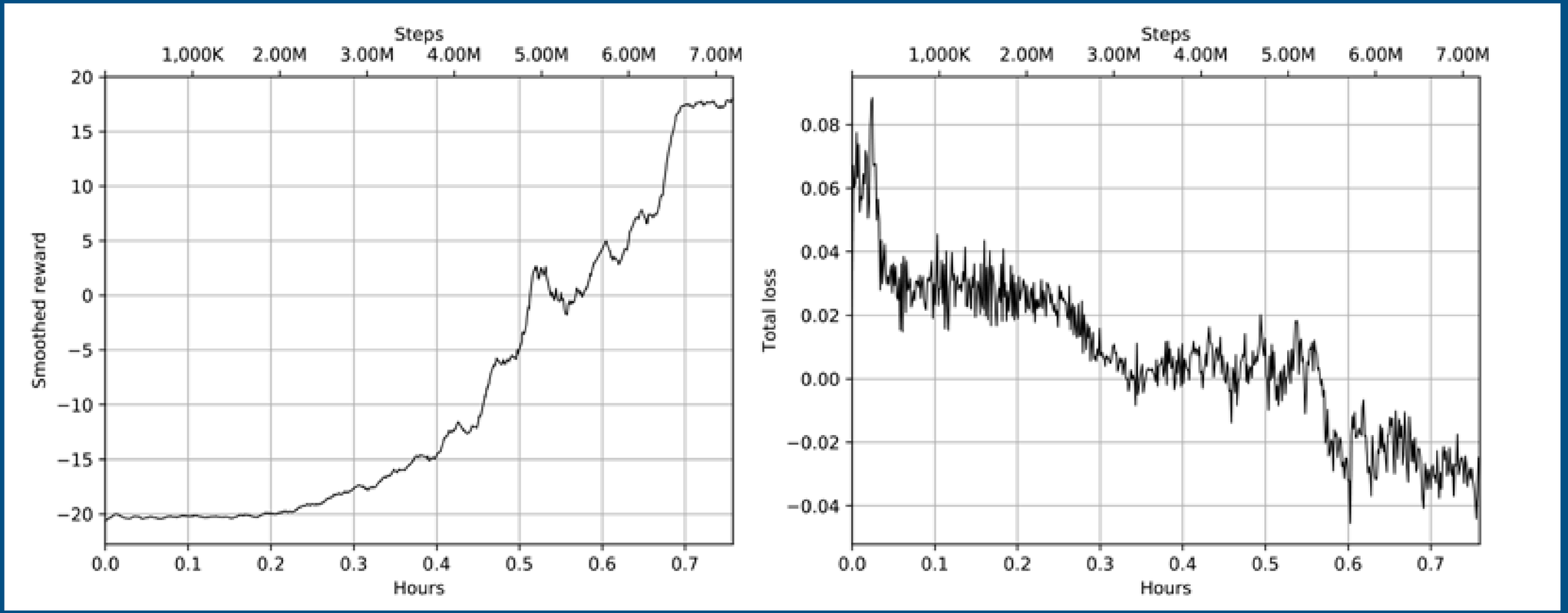
Asynchronous Advantage AC (A3C)

- 두 가지 접근 방식
  - 그레디언트 병렬화 : 각 에이전트에서 만들어진 샘플들을 사용해 그레디언트를 계산한다. 그리고 각 에이전트에서 계산한 그레디언트들을 글로벌 신경망에 전달해 SGD 갱신을 수행한다. 그 뒤 각 에이전트에서 샘플을 보낼 때 갱신된 가중치를 전달한다.



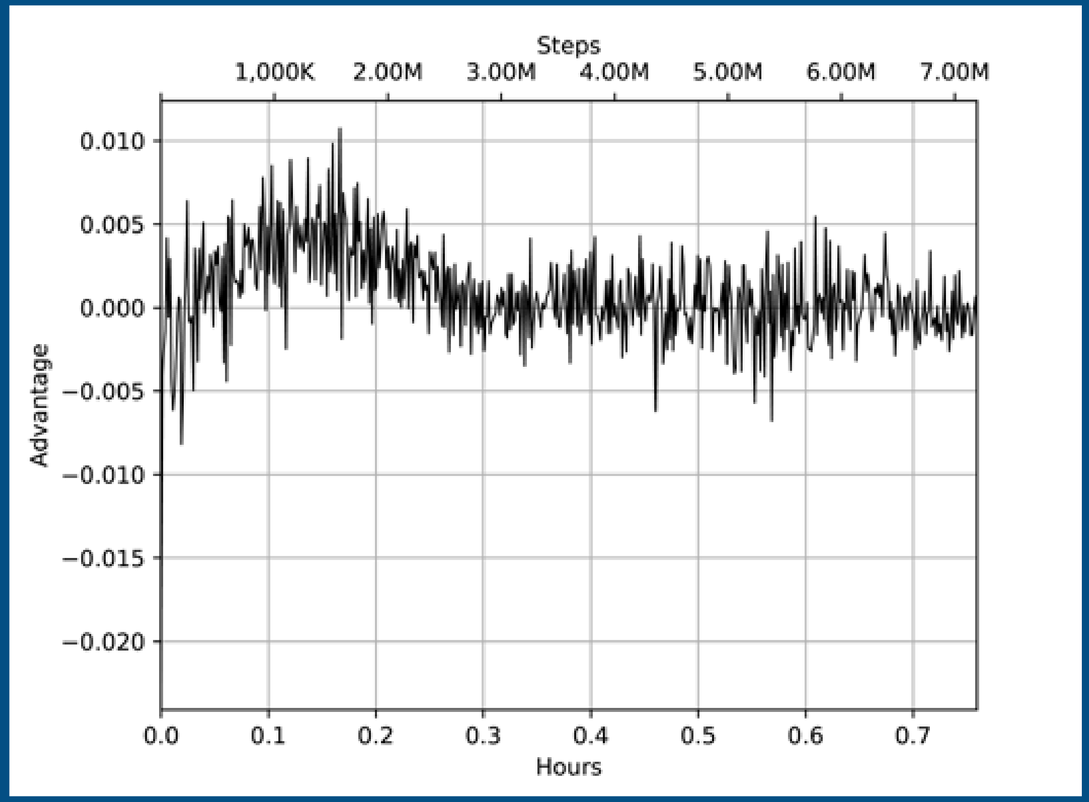
Asynchronous Advantage AC (A3C)

- Reward and Total Loss of Data-Parallel Version



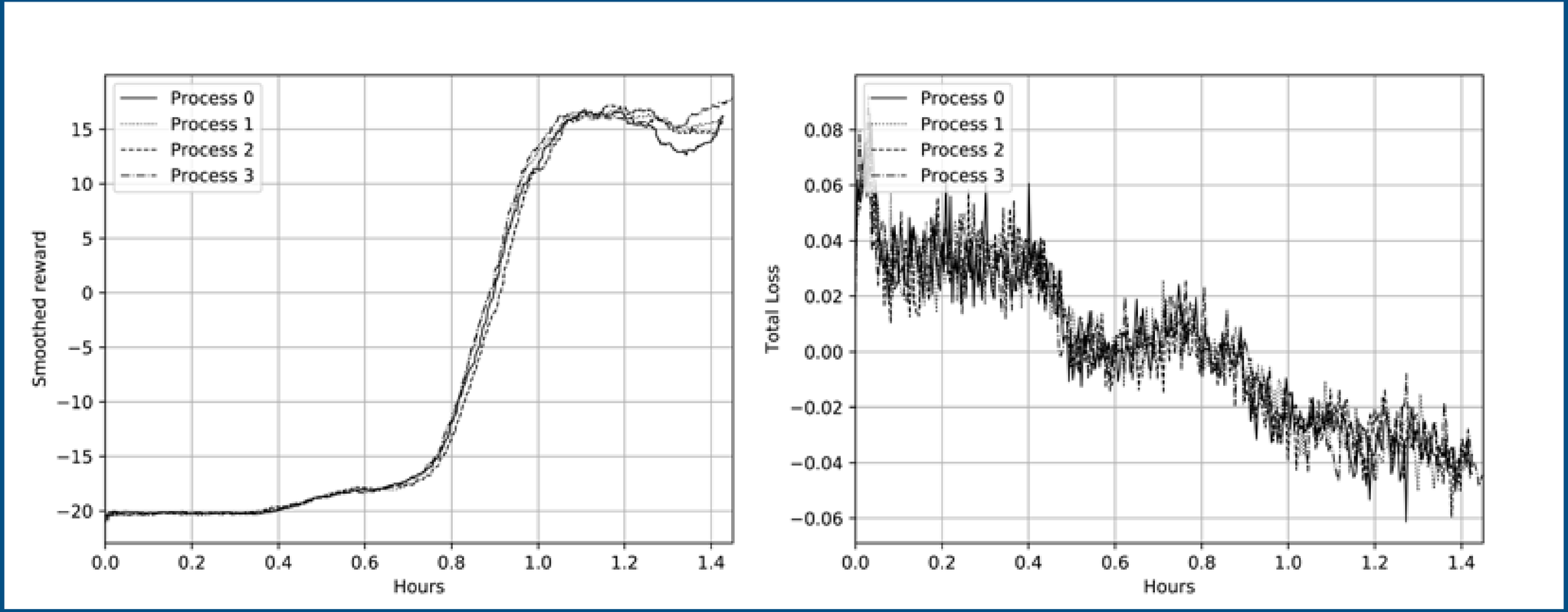
Asynchronous Advantage AC (A3C)

- Advantage of Data-Parallel Version during The Training



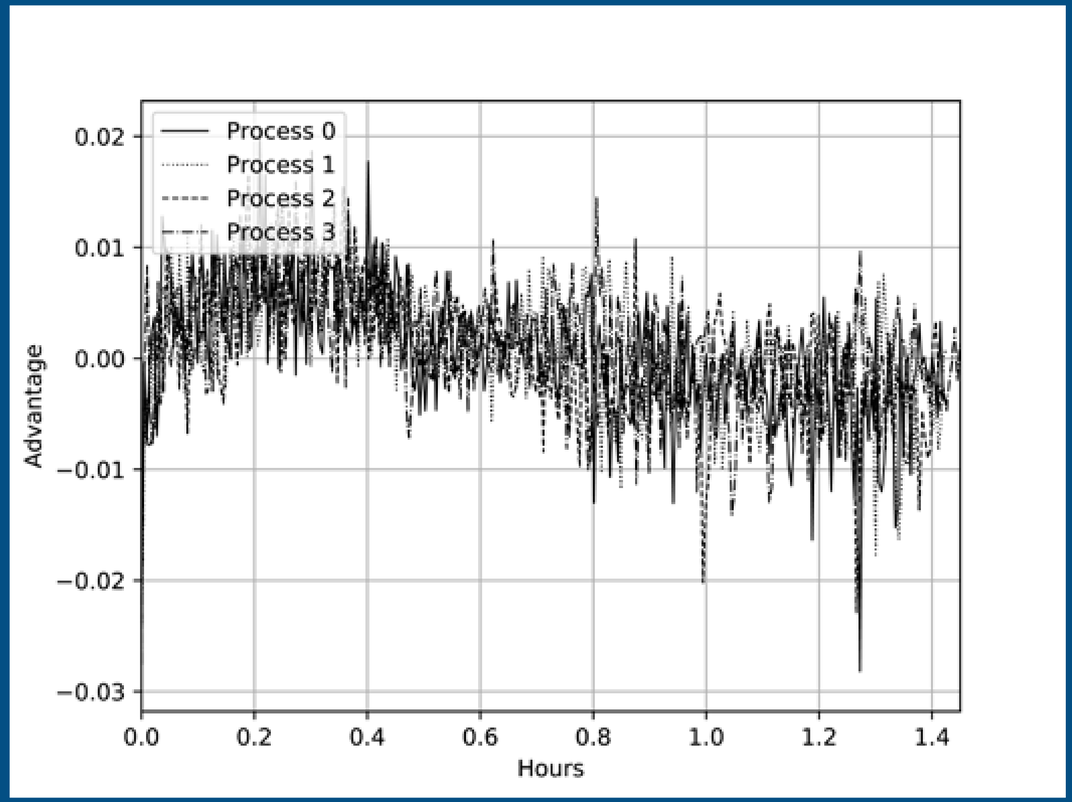
Asynchronous Advantage AC (A3C)

- Mean Reward and Total Loss of Gradient-Parallel Version



Asynchronous Advantage AC (A3C)

- Advantage of Data-Parallel Version during The Training



감사합니다!

스터디 듣느라 고생 많았습니다.