

환경 복잡도에 따른 시계열 데이터 처리 시 LSTM과 프레임 스택킹 방법의 성능 비교

A Comparative Study on the Performance of LSTM and Frame Stacking Methods
in Time Series Data Processing Based on Environmental Complexity

박영주 (발표자)

한국항공대학교 항공우주공학과 4학년, andy1andy@kau.kr

박도희

한국항공대학교 항공우주공학과 2학년, lahee0803@kau.kr

서보승

한국항공대학교 소프트웨어학과 3학년, sbs7696@kau.kr



한국항공대학교
Korea Aerospace University

DEEPERENT

Contents

서론

실험 환경 및 모델

실험 결과

결론



Contents

서론

실험 환경 및 모델

실험 결과

결론



한국항공대학교
Korea Aerospace University



I. 서론

- 강화학습에서 시계열 데이터는 학습에 자주 사용됨.
 - Ex) 로봇틱스 제어, 자율주행, 게임 AI 등
- 시계열 데이터를 처리하는 대표적인 두가지 방법 : Frame Stacking, RNN
 - Frame Stacking → 이전 상태 데이터를 누적해 시간 의존성을 반영
 - RNN → 순환 신경망 구조를 활용해 시간적 의존성을 학습
- 문제의 특성 및 환경적 요인에 따라 두 방법 중 어떤 것이 더 효과적일지 결정됨
 - ➔ 문제의 특성에 따라 어떤 방법을 선택할지에 관한 기준 필요



- 관측 환경의 난도에 따라서 Frame Stacking과 LSTM의 성능이 달라진다고 가정
 - OpenAI Gym의 Pendulum 환경을 3가지 형태로 변형하여 난도별 환경 구성
 - 각 환경에 대하여 Frame Stacking과 LSTM 기법을 적용하여 성능을 비교, 적합성을 평가
- 기존 연구와의 차이점
 - Frame Stacking와 LSTM의 차이점 연구한 기존 연구 → 벡터 환경 한정
 - 기존의 벡터 환경 학습 연구를 보완하고자, 본 연구에서는 이미지 프레임 환경으로 새롭게 정의된 학습을 통해 모델 성능을 평가



Contents

서론

실험 환경 및 모델

실험 결과

결론



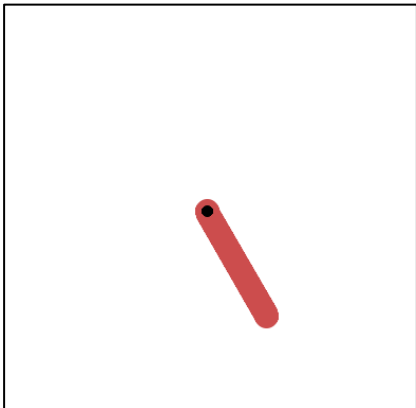
한국항공대학교
Korea Aerospace University



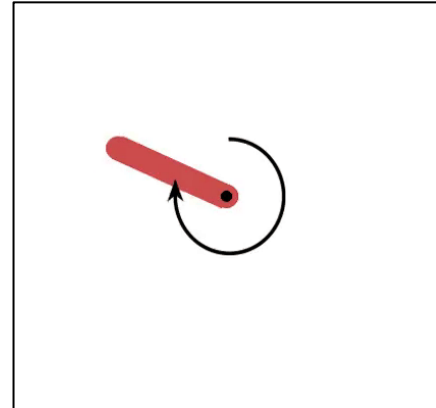
II. 실험 환경 및 모델 정의

2.1 실험 환경

- 실험 환경 : Pendulum 환경 (from OpenAI Gym)
 - 운동방정식 : $I\ddot{\theta} + b\dot{\theta} + mgl \sin \theta = T$
- 이 환경의 목표
 - 관측값을 입력으로 받아 제어 입력인 토크 T 를 조절하여 진자를 위로 세운 후 이를 유지하는 것



← 학습이 되지 않은
Pendulum



← 학습이 완료된
Pendulum



2.2 관측값(Observation) 정의

- Pendulum 환경의 관측값을 변형하여 3가지로 구성

- 환경 1 : 완전 관측 환경 (Fully Observable Environment)

- 관측값 = [Pendulum x좌표, y좌표, 각 속도] = $[\cos(\theta), \sin(\theta), \omega]$

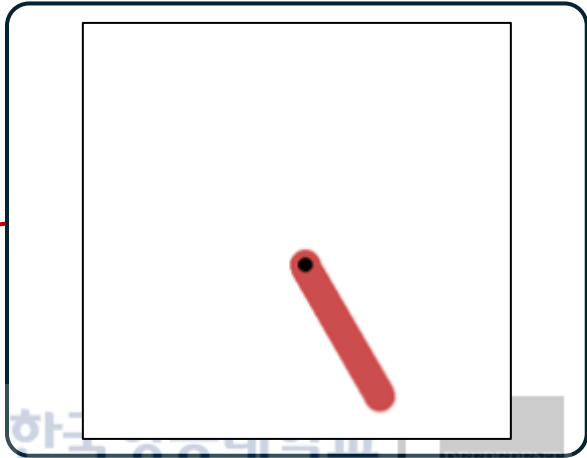
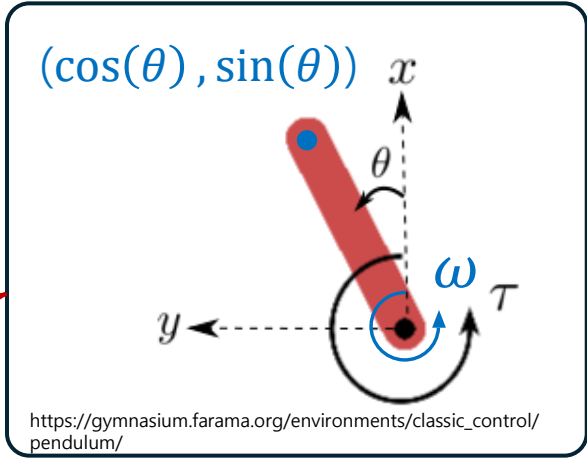
- 환경 2 : 부분 관측 환경 (Partially Observable Environment)

- 관측값 = [Pendulum x좌표, y좌표] = $[\cos(\theta), \sin(\theta)]$

- 환경 3 : 이미지 입력 환경

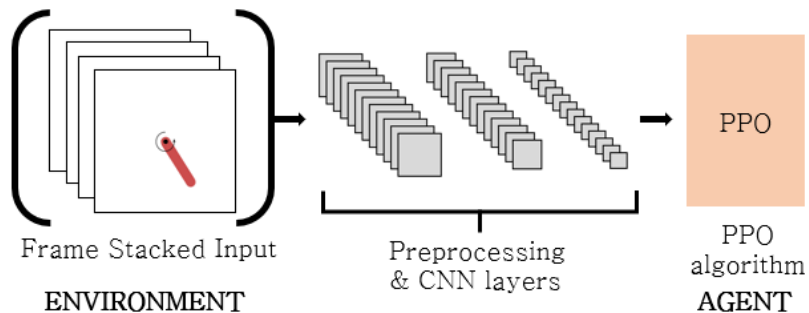
- 관측값 = 500 x 500 Pixel, RGB Image Input

- 환경 1 < 환경 2 < 환경 3 순으로 난도를 구성함

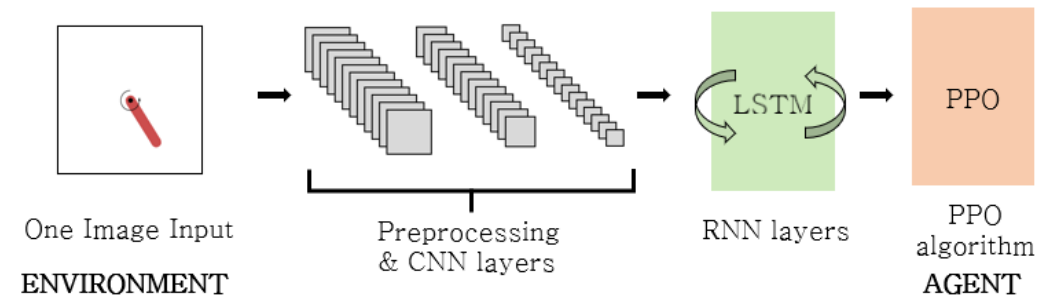


2.3 모델 정의

- PPO(Proximal Policy Optimization) 알고리즘 사용
- PPO + Frame Stacking
 - PPO 알고리즘에 Frame Stacking 을 추가한 모델
 - 각 스텝마다 최근 4 개의 관측값(벡터 또는 이미지)을 쌓아 하나의 새로운 관측값을 구성
- PPO + LSTM
 - PPO 알고리즘에 LSTM층을 추가한 모델
 - 각 스텝에서 단일 관측값(벡터 또는 이미지)을 활용해 학습이 이루어짐



[환경 3에 적용된 PPO + Frame Stacking 모델]



[환경 3에 적용된 PPO + LSTM 모델]

Contents

서론

실험 환경 및 모델

실험 결과

결론



한국항공대학교
Korea Aerospace University



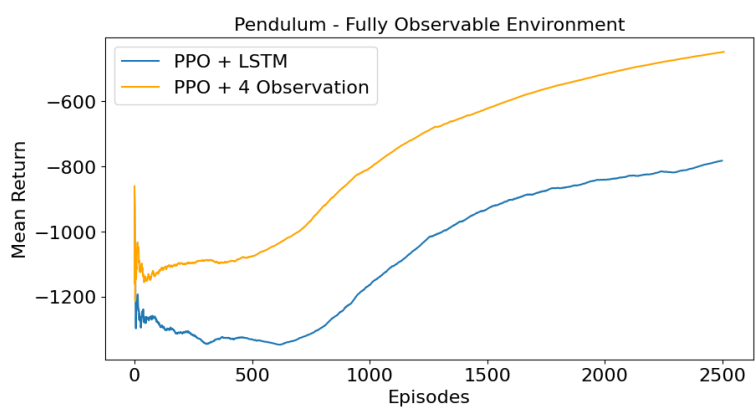
III. 실험 결과

환경 난도 상승

PPO + LSTM

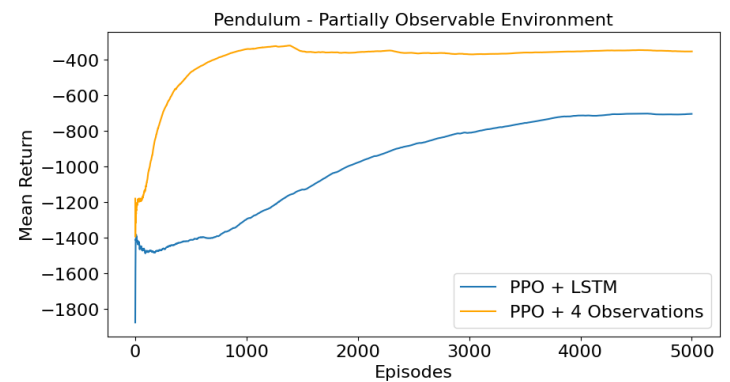
PPO + Frame Stacking

환경 1 (완전 관측 환경)



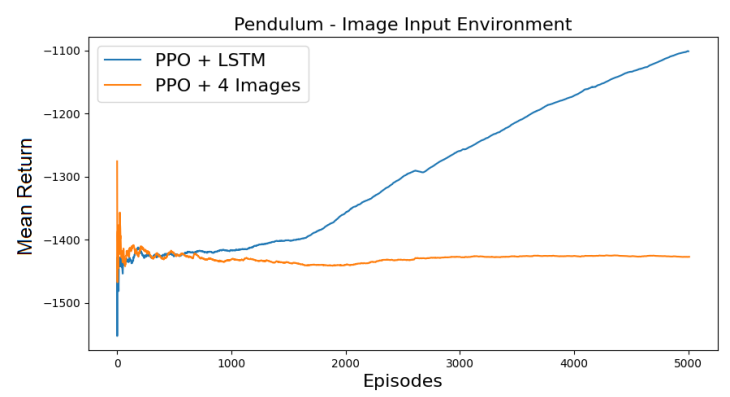
* Total timestep = 500k

환경 2 (부분관측 환경)



* Total timestep = 1M

환경 3 (이미지 입력 환경)




* Total timestep = 1M

- 환경 1, 2에서는 PPO + Frame Stacking(■)의 성능이 PPO + LSTM(■)보다 월등히 뛰어남
- 환경 3에서는 PPO + Frame Stacking(■)은 학습이 전혀 진행되지 않지만, PPO + LSTM(■)은 시간이 지나며 학습이 진행되어 PPO + Frame Stacking(■)의 성능을 현저히 뛰어 넘음

- 학습이 완료된 두 모델을 100번의 에피소드에 대하여 테스트한 결과

소
문
내
고
정
화



	P-Frame Stacking	P-LSTM
환경 1	-175.92 ± 114.44	-592.84 ± 79.61
환경 2	-354.71 ± 234.05	-584.38 ± 75.91
환경 3	-1386.18 ± 224.57	-727.30 ± 139.20

- PPO + Frame Stacking 모델은 환경 1에서 가장 좋은 성능을 보였고 환경 난도가 증가할수록 성능이 낮아짐. PPO + LSTM 모델은 환경 1, 2에서 비슷한 성능을 보임
- 환경 3에서는 환경난도가 크게 증가하여 두 모델 모두 환경 1, 2보다 낮은 성능을 보임
 - PPO + Frame Stacking 모델은 학습이 전혀 진행되지 않았지만, PPO + LSTM 모델은 학습이 진행되어 PPO + Frame Stacking 모델의 성능을 뛰어넘음

Contents

서론

실험 환경 및 모델

실험 결과

결론



한국항공대학교
Korea Aerospace University



IV. 결론

- 관측 환경의 난도가 증가할수록 PPO + LSTM의 적합도가 증가함
- Pendulum 이라는 단일 환경에서만 실험을 진행한 한계가 존재함
 - 더욱 다양한 환경에서의 실험을 통해 두 모델 간의 성능 차이를 일반화할 필요 있음
- 본 연구는 기존 연구들과 달리 이미지 프레임을 도입했다는 차별점이 존재
 - 환경 3 에서 PPO-LSTM 이 PPO-Frame Stacking 에 비하여 뛰어난 성능을 보인 것에 주목할 만함
- 이미지 프레임 환경과 CNN층이 포함된 순환신경망 모델을 활용한다면, Gymnasium 과 같은 가상 환경의 제어 뿐만 아니라 카메라와 같은 장비를 활용하여 현실 세계에서보다 범용적인 적용 가능성을 기대할 수 있을 것

제5회 한국 인공지능 학술대회

The 5th Korea Artificial Intelligence Conference

Q&A

환경 복잡도에 따른 시계열 데이터 처리 시 LSTM과 프레임 스택킹 방법의 성능 비교

A Comparative Study on the Performance of LSTM and Frame Stacking Methods in Time Series Data Processing Based on Environmental Complexity

박영주 (발표자)

한국항공대학교 항공우주공학과 4학년, andy1andy@kau.kr

박도희

한국항공대학교 항공우주공학과 2학년, lahee0803@kau.kr

서보승

한국항공대학교 소프트웨어학과 3학년, sbs7696@kau.kr



한국항공대학교
Korea Aerospace University

