보고서

과 목 명 : 딥 강화학습

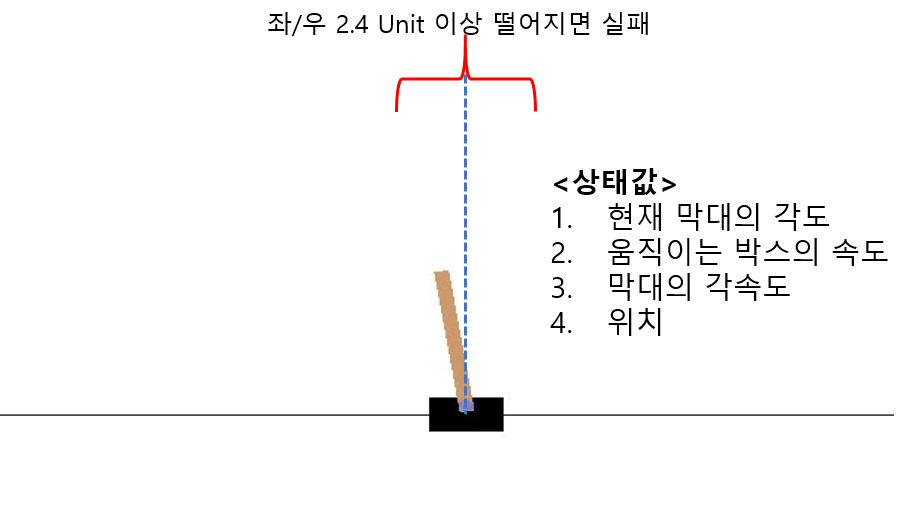
과제 제목 : CartPole, Hopper –Actor Critic



|  |  |
| --- | --- |
| **과목명** | **딥 강화학습** |
| **교수명** | **최영식 교수님** |
| **학과** | **컴퓨터공학과** |
| **학번** | **2021307006** |
| **이름** | **이태강** |
| **제출일** | **2022년 06월 26일** |

**1. 설명**

**1) Gym Library 중 CartPole-v1 이란 게임(그림.1)**



<그림.1 Gym-CartPole-v.1 의 환경과 수행 조건>

**2) CartPole-v.1 을 학습 시키는 알고리즘은 a3c로 구성**

TD Actor-Critic

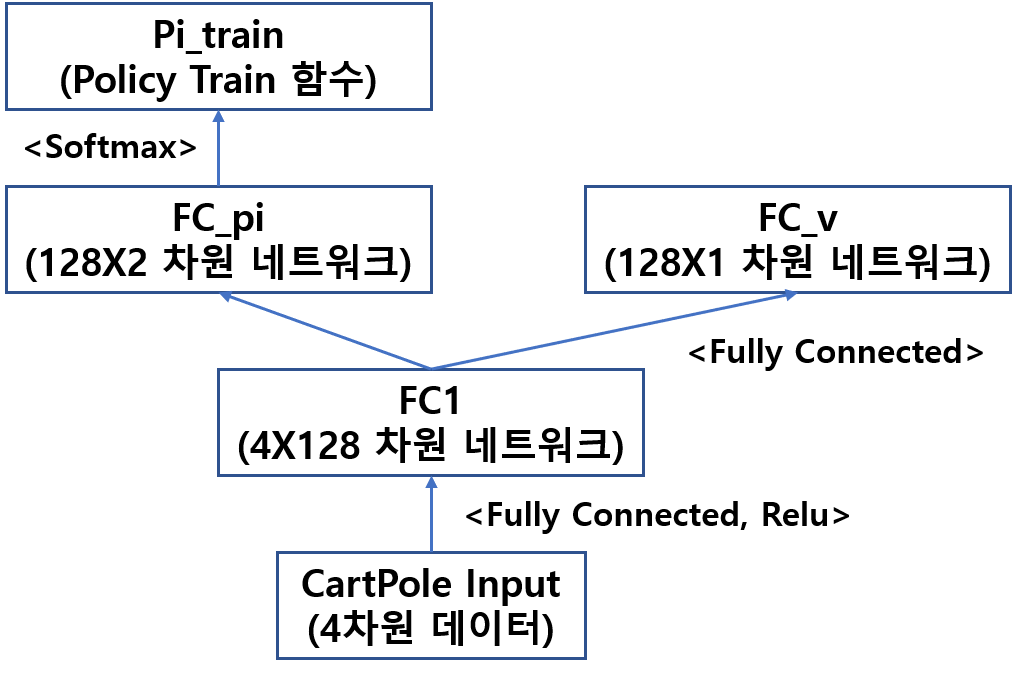
🡪네트워크 설계 시 Policy와 Value 부분을 나누어지게 설계하여 Reinforce에서 하지못했던 Episode 진행 시 Gradient를 업데이트 하여 학습 진행

**2. 알고리즘 분석**

**TD Actor-Critic**

🡪 TD Error를 이용한 Actor Critic 알고리즘을 CartPole 게임에 적용시켜 본다.

🡪 Actor-Critic 이기 때문에 기존의 Reinforce알고리즘과 다르게 모델이 Policy 와 Value(actor 와 critic) 두개를 주게 된다.



<그림.14 Actor-Critic의 모델을 만드는 형태>

🡪fc1까지는 기존의 Reinforce와 같은 형태로 들어가고 이 후 fc\_pi (policy) 와 fc\_v (value) 두갈래 길로 나누어 지게 된다

🡪fc\_pi 의 경우 policy는 확률 분포이기 때문에 softmax를 취하지만 value의 경우 어떤 값이든 나올 수 있기 때문에 128\*1차원으로 그냥 value값이 나오게 설계 한다.

🡪모델의 인풋으로 observation을 넣으면 output으로 확률이랑 value가 둘다 나와야 한다. 다음 스테이트인 s\_prime을 이용해 td target을 구하고, value가 나올경우 델타를 계산 할 수 있다. (이전 Reinforce 에서는 log(pi)\*Reward 였지만 Actor-Critic에서는 log(pi)\*delta이다.)

🡪전체 로스를 계산하기 위해 그림.15의 초록색부분 (폴리시의 로스함수) 를 먼저 계산하고, 파란색 부분(밸류의 로스함수 = delta의 제곱)을 계산하여 TD에러를 미니마이즈 한다.

🡪 이를 매 스텝마다 업데이트 하여 Actor-Critic 의 장점을 살린다.

🡪 로스를 백프로파 게이션(backward)을 통해 그라디언트를 계산 한 후 Optimizer 를 실행 하면 그라디언트가 업데이트 된다.

🡪모델에 관찰한 observation을 넣으면 output으로 확률과 value를 둘다 나오게 된다. Policy의 확률 분포에서 카테고리컬로 바꾸고 샘플 하여 action을 뽑는다.

🡪action을 환경에 넘겨주어 다음 state(s\_prime)와 reward를 받고 모델에 데이터를 입력한다.

🡪x축은 episode, y축은 score로 1800번 정도 episode에서 450점을 넘는 것을 처음 발견했지만 다시 효율이 덜어지고 4000번 이상 episode를 진행 했을 때에 450 score 이상이 많이 보인다. 하지만 안정적이진 못하고 점수가 떨어지기도 한다.

텍스트이(가) 표시된 사진

자동 생성된 설명