<https://github.com/seungeunrho/minimalRL/blob/master/ddpg.py> minimalRL을 바탕으로

<https://github.com/yanpanlau/DDPG-Keras-Torcs> Keras 구현을 참고하여 DDPG for Keras를 만들어 본다.

의문점이 있다. Keras예시에서 Critic은 마지막 Q value를 action dimension만큼 뱉는다. 이건 원래 DQN에서 non-continuous한 action space를 다룰 때나 나왔던 방식 아닌가? frozen lake에서의 상하좌우, Cartpole에서의 좌우 등. 지금 우리가 하는 DDPG방식은 action과 state를 한꺼번에 집어넣어 Q value를 단 하나 구하는 방식으로 이해했는데, action dimension만큼 output dimension을 잡은 것은 실수인가 의도인가? 나는 일단 1개출력이 나오는 방식으로 해 보기로 하였다.

actornetwork 모듈과 criticnetwork모듈, replaybuffer모듈을 만든 뒤 gym\_torcs와 연동이 잘 되는지 간단한 테스트를 해보았고 잘 되었다.64by64해상도로 설정되어 있어서 메뉴 글자가 전혀 보이지 않아 화면을 키울 수가 없었는데, 해상도 설정을 조작하는 config 파일을 찾으려 고생하다 결국 유튜브로 메뉴 순서를 본 뒤 gui상에서 해결하였다.

여전히 OU가 문제. 두 예시에서 OU를 다루는 방법이 달라 보인다.

minimalRL의 OU가 실제 정의에 부합하는 것 같다.

여기서 는 noise를 의미. 는 예전 noise에 더해져 새로운 noise를 만든다.

<https://github.com/openai/baselines/blob/master/baselines/ddpg/noise.py>

baseline을 보다가 깨달은 사실. noise를 더해주면 action이 가끔 범위를 벗어날 수도 있다. 어떻게 해야 하는가? torcs는 action을 받을 때 범위 아래 값도 받는가? noise를 더해준 뒤 범위를 벗어나면 clip으로 처리해야한다. np.clip(a,min,max)

일정 시간 특정 속도 이하일 때 env.end()시키는 방법..? 은 나중에 알아보자. 지금은 급하지 않다.

step function은 1차원 array를 값으로 받는다.

np.zeros()는 1차원 array를 출력한다.

model의 입출력은 tensor이다.(수정)1차원도 상관없는 것 같지만, cat은 같은 모양을 더해줘야 하므로 현재 우리는 2차원을 사용하여야 한다.

actor의 backpropagation은 어떻게 이뤄지는 것? 기억이 또 안나네.

loss = -q(s,mu(s)).

q(s,mu(s))를 최대화 해야 된다는 것은 이해함.

…

critic model에 action\_output\_original은 들어가지만, 이를 noise에 더하기 위해 numpy()로 numpy화 했던 action\_output은 들어가지 않음. tensor는 들어가지만 그렇지 않으면 들어가지 않는다?

minimal DDPG에서 저장된 a는 tensor가 아닌 것 같은데. 검증이 필요하다.

tensor였다.

OU에 대해 감 잡은 것 같다. 평균을 향해 가는 성질이 있는 noise. 단, theta가 크면 클수록 빨리 가고 적을수록 느리게 감. sigma가 클수록 변동이 크고 작을수록 변동이 작음. theta는 1이상으로 두지 말 것. 발산한다.