CartPole 예제도 마찬가지로 Qnet 적용해보자.

random\_uniform 쓰던거, 이렇게 바꿔줬음 – xavior initializing

        initializer = tf.initializers.GlorotUniform()

        self.W = tf.Variable(initializer(

            shape = (input\_size, output\_size)))#weight initializing?

my\_model에서 float 64를 float32로 캐스팅하려 한다는 warning이 뜬다.

학습 시도를 해보는데 딱히 학습이 잘되지는 않는다. 강의에서도 평균 몇십 step정도 reward를 받던데, 성능 자체는 정상적이다. 문제는, 속도가 너무 안나온다는 것이다! 강의에서 학습되는 속도보다 한 20배는 느린 것 같다… 해결법이 없을까? 코드를 이곳 저곳에서 누더기처럼 기워 댔기에 어쩔 수 없는 것일까, 아니면 tensorflow 2.0의 속도가 어쩔 수 없이 이정도 느린걸까.

----속도 문제 관련 구글링 하다가 찾게된 잡설 -------

<https://data-newbie.tistory.com/425> 포스트에 따르면, 요즘 대세는 Pytorch라고 한다. 새로운 논문들이 전부 pytorch 기반으로 나온다고 한다. tf진영 창시자인 Deepmind조차 pytorch로 넘어가고 있는 추세라고. 이게 사실이면…

Tf의 장점은 기업에서 쓰고있고 배포에서 장점이 있다는 것인데, 문제는 TF2는 그런 장점이 하나도 없다는 것이다. 게다가 3월기준 TF2는 성능 메모리 버그 문제들로 고통받고있다는데 어…

어찌되었건, 왜 학습이 잘 안되는가?

network 변수가 4개밖에 없다, 너무 shallow함. +

샘플간 correlation, 타겟의 불안정(non-stationary)이 원인.

DQN으로 들어가보자.

<https://dryjelly.tistory.com/140?category=785786> 참고할만한 tistory post

Nature 표지를 장식한 Deepmind 팀의 DQN 논문.

Correlation between sample의 의미는 무엇인가?

action 1회를 취했을 때 state는 극적으로 변화하지 않는다. 즉, 학습에 필요한 sample(state) 간에유사성이 너무 크다. 샘플이 넓게 퍼져 있으면 학습하기 쉽지만, 비슷한 위치에 몰려 있는 샘플로는 제대로 된 모델이 학습되는 것을 기대하기가 힘들다.

non-stationary targets

