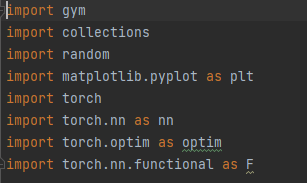
**컴퓨터네트워크1 HW (2021.11.22)**

학번 : 120210200

이름 : 전장우

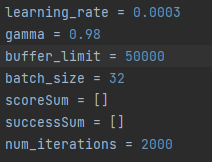
1. A description of the algorithm I used.

-Library



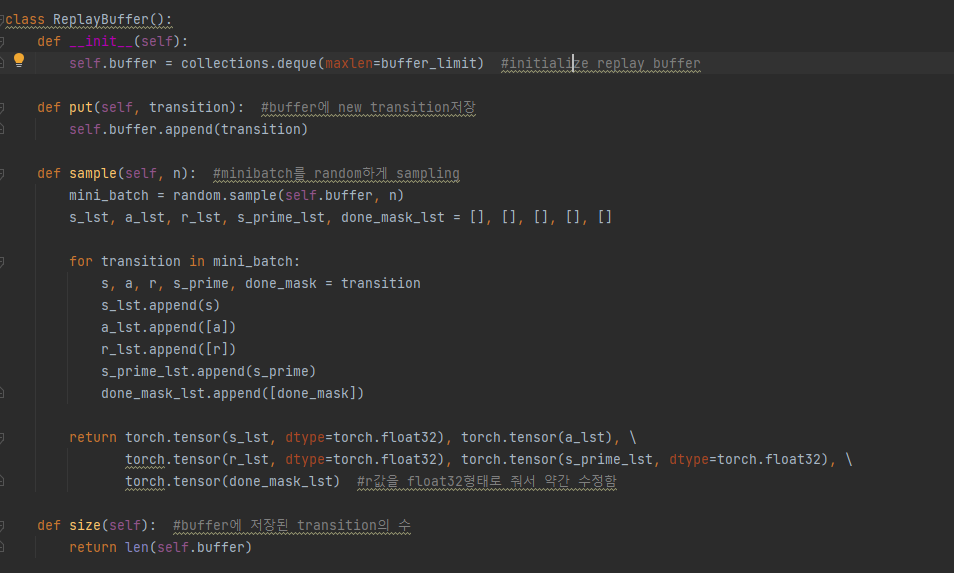
Library는 DQN을 사용하기 위해 pytorch를 사용하였다.

-Hyperparameter



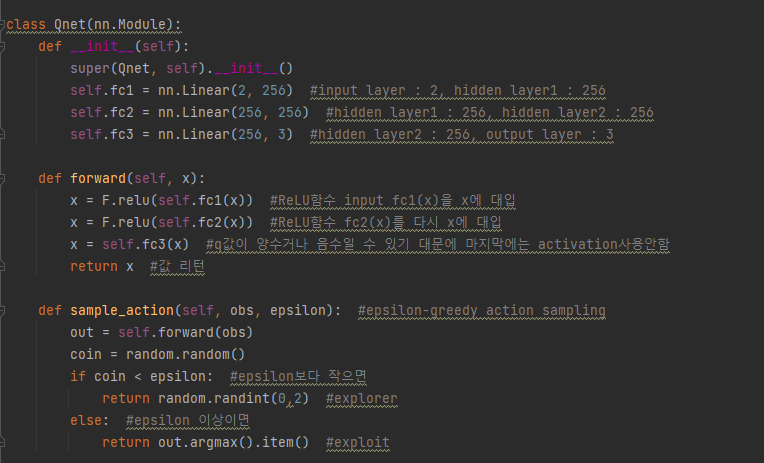
Learning rate는 0.0003, 감쇄율 은 0.98, replay buffer의 사이즈는 50000, batch size는 32, 그리고 iteration은 2000번을 하도록 설정하였다.

-Replay buffer



Replay buffer의 코드는 수업시간에 했던 코드를 참고했다. 단 나중에 main함수에서 쓸 r(reward)를 실수 형태로 줄 것이기 때문에 dtype을 float형식으로 변환하였다.

-Qnet

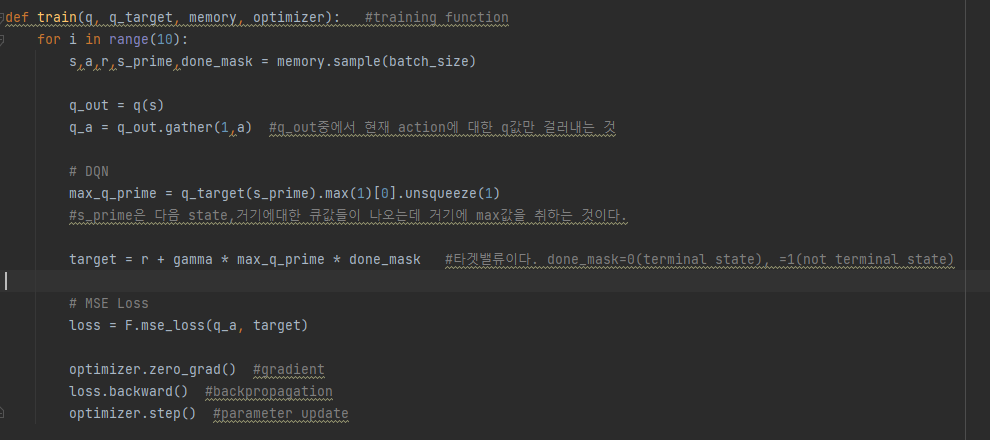


Neural network는 Input layer는 2개, First hidden layer는 256개, Second hidden layer 256개, Output layer는 3개로 설정하였다.

Forward propagation은 ReLU 함수를 사용하였다.

Sample action은 를 사용하여 보다 작으면 explorer, 보다 크면 exploit하게 동작하도록 설정하였다.

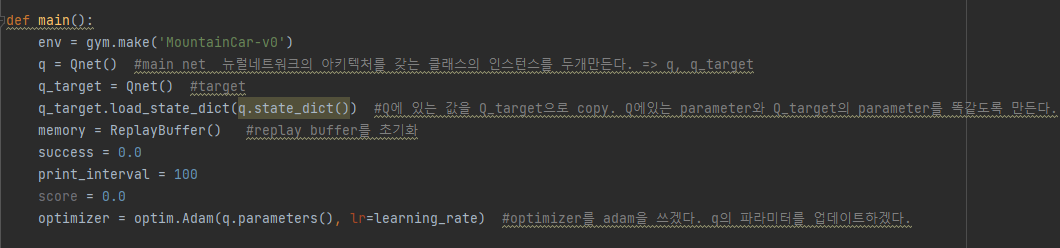
-Training function



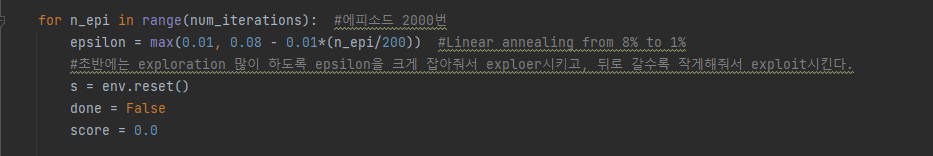
S = state, a = action, r = reward, s\_prime = next state, done\_mask = 0(terminal state), 1(terminal state)를 나타낸다.

Deep Q Learning을 사용하였고, target value는 R+에다가 done mask라는 변수를 사용하여 freeze했다.

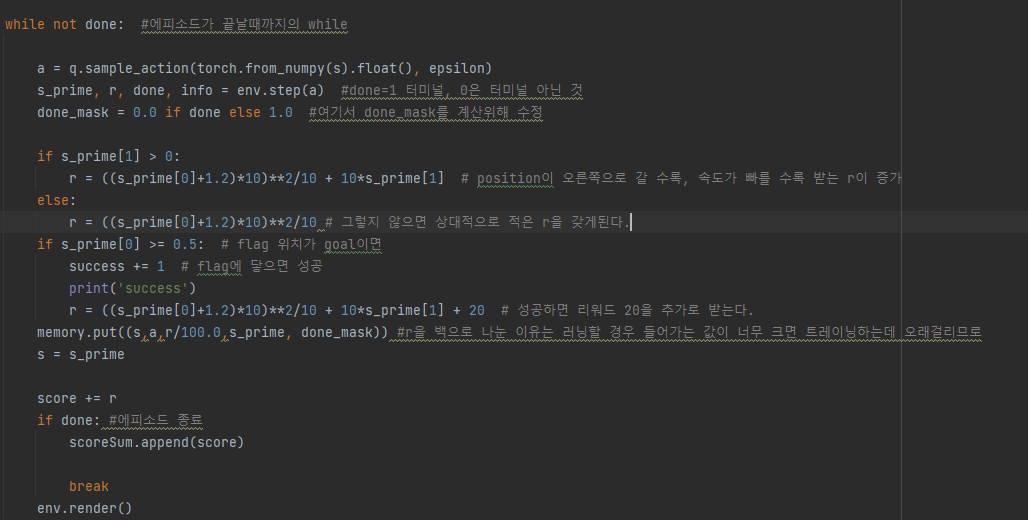
-Main



Qnet 함수에서 만든 neural network의 아키텍처를 갖는 클래스의 q와 q\_target이라는 2개의 인스턴스를 만든다.

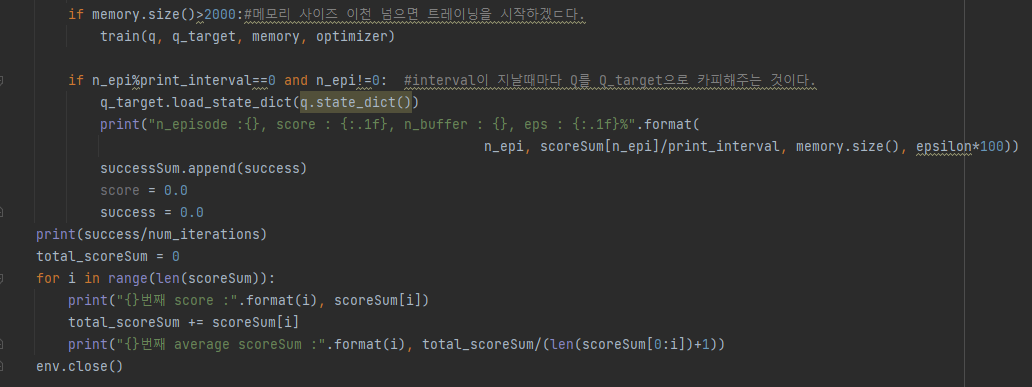


를 사용하는데 처음에는 의 크기를 크게 해서 explorer를 많이하도록 한 다음, iteration을 많이 할수록 의 크기를 줄여나가서, exploit하게 동작하도록 한다.



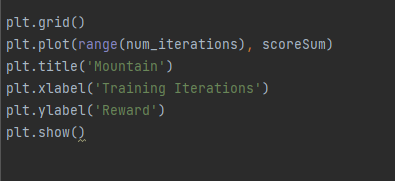
Velocity가 0보다 클 때, Position이 오른쪽으로 갈수록, Velocity가 빠를수록 받는 Reward를 크게 잡았다. 그렇지 않으면 상대적으로 적은 Reward를 가지게 설정하였다.

그리고 agent가 goal(position=0.5)에 도달하면 추가적으로 20의 Reward를 받도록 설정하였다.

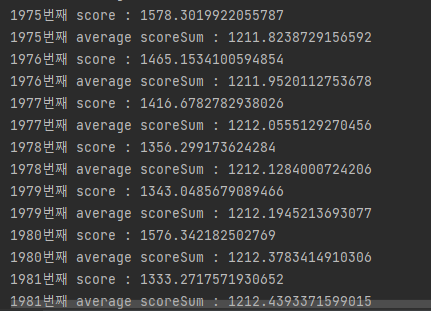
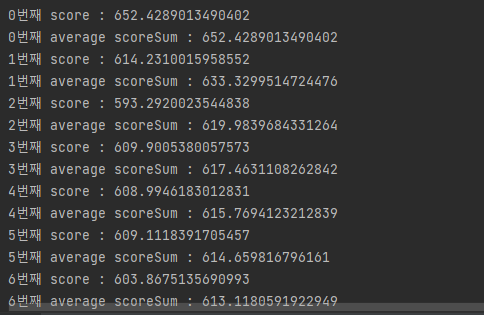
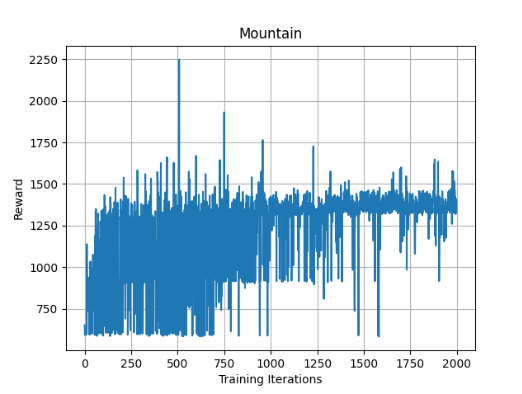


Iteration이 Interval(100회)가 될 때마다, 현재 몇 번째 episode(iteration)인지와 Score(sum of reward / Interval), 메모리 사이즈, 하고 goal에 도착하면 success를 출력하도록 설정한다. 그 뒤, 각 iteration에서의 score와 score의 평균을 출력하도록 설정하였다.

-Graph code



X축은 Iteration, Y축은 scoreSum(Reward)를 출력하는 그래프를 나타내도록 설정하였다.



처음에는 학습률이 저조하고 이 크기 때문에 explorer를 많이 해서, Reward의 범위도 넓고, 평균적으로 낮은 Reward를 갖게 되지만, 점차 학습률이 상승함에 따라 이 작아지고, explorer를 줄이고 exploit하게 행동하기 때문에 평균적으로 높은 Reward로 수렴하게 행동하는 것을 확인할 수 있다.