

# Deep Q Network

Q\_learning + DNN

DQN은 Q 값을 DNN을 이용해서 업데이트를 하는 Q\_learning 방법

딥러닝을 이용하기 때문에 보다 복잡한 함수로 Q 값을 표현이 가능해짐.

## Q\_learning의 한계점 개선

1.State가 다양해짐에 따라 복잡한 정확한 Q 값을 도출하기에는 한계가 존재함.

2.이를 해결하기 위해 비슷한 State끼리 묶은 후

3.하나의 State의 Q 값을 도출한 후 비슷한 State에 대입하는 형식

4.새로운 sample값이 들어와도 lr을 이용해 Q 값을 미세하게 업데이트 해나감

ex) 10이라는 샘플을 열 번 얻었다고 가정.

11번째 값에 12가 들어왔을 때 평균 값은 아무리 커도 12 이하일 것이다(10.xxxx)

## DQN 특징

가치 기반 Agent로 운용되는 강화학습 모델 -  $Q(st, at)$ 가 아닌 State값만 사용

\*그럼 Action을 어떻게 뽑는가?

DNN 모델에 State를 input으로 넣고 output으로 Action 값을 출력

ex) output -  $Q(st, \text{왼쪽})$  &  $Q(st, \text{오른쪽})$  - 둘 중  $\text{Max}(Q)$ 의 Action을 선택

선택된 Action에 대한 Q 값만 업데이트를 진행

기존의 DL은 모든 결과값을 업데이트 한다는 점과 차이점이 존재

## 핵심 포인트

1.CNN(DNN) 사용 - 인간처럼 행동하도록 구현

2.Experience Replay - 예전에 했던 experience를 다시 참고하고 업데이트 하는 것

3.State Corr을 확연히 줄어들었음. (연속형 → 이산형)

State Corr이 너무 높으면 Action의 진행상황을 연속적으로 보게 되는 경우가 발생해서 학습시간이 오래걸릴 뿐만아니라 이산적으로 보는 것과 다를 바가 없다는 문제가 있음.

## DQN 2015 차별성

Target Net과 Main Net으로 구분을 지음

Target Net - Sample을 뽑아서  $w$ (가중치)를 업데이트 진행

Main Net - Target Net에서 충분히 업데이트 된  $w$ 값을 Main Net에 적용

- 1.Target Net  $w \leftarrow$  Main Net  $w$ (복사)
- 2.Target Net  $w$  Update 진행 (C 스텝 마다) \*C는 사용자 정의
- 3.충분히 업데이트 한 후 Target Net  $w \rightarrow$  Main Net  $w$ 로 복사
- 4.반복