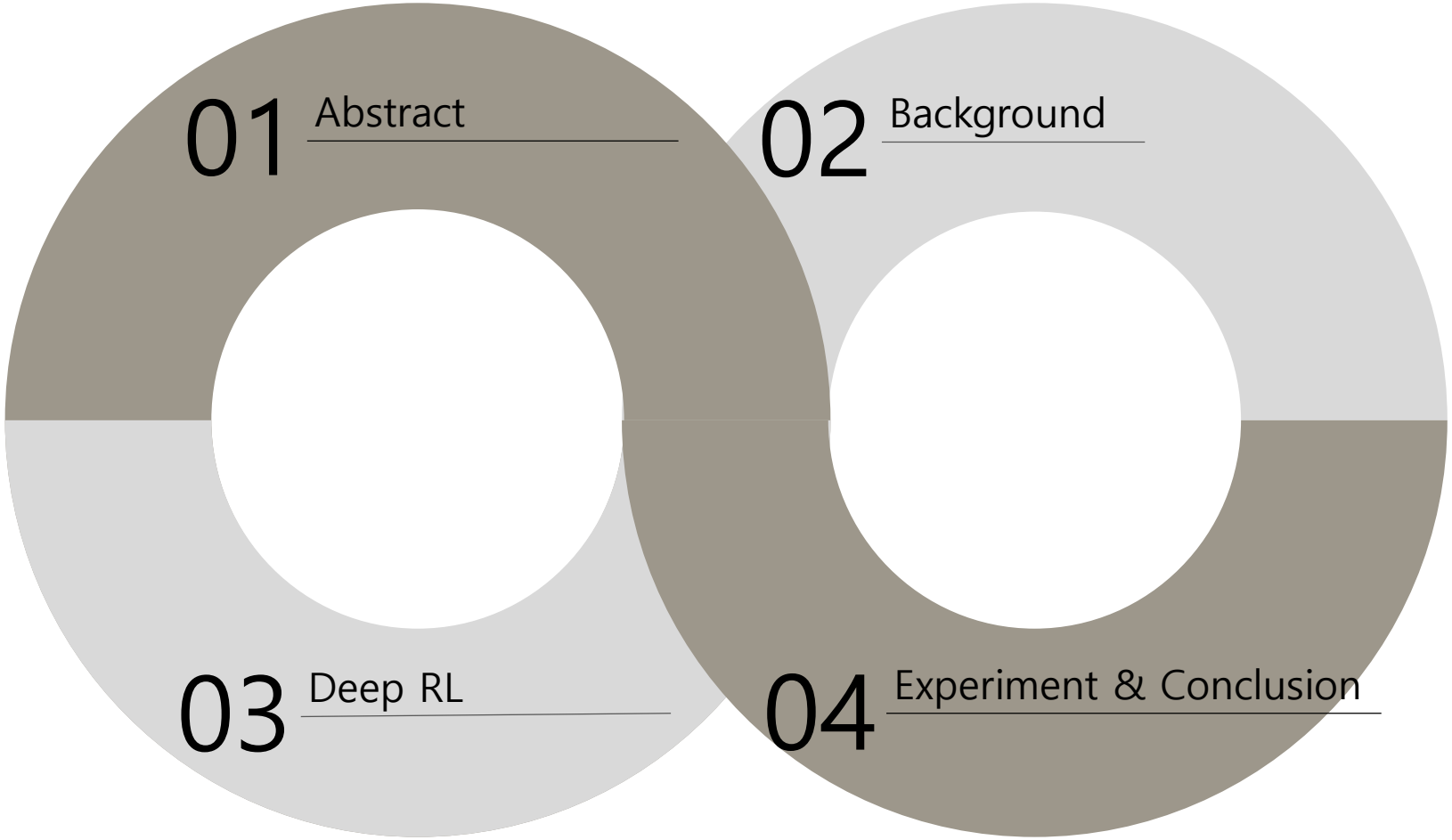
차세대 데이터베이스

- Deep Reinforcement Learning with Double Q-learning-

2019/05/14 Tuesday

컴퓨터공학과
201972220 조민규

Contents


01 Abstract
02 Background
03 Deep RL
04 Experiment & Conclusion

Part
01

Abstract

[논문 요약]

- ✓ Q-Learning Algorithm의 Overestimation 문제에 대해 소개
- ✓ Double Q-Learning 알고리즘에 대해 소개
- ✓ Double Q-Learning 알고리즘을 Atari Game에 적용한 결과 확인

Part
01

Abstract

[논문 요약]

- ✓ Q-Learning Algorithm에는 Maximization Step이 존재 → Overestimation을 야기시킴
- ✓ Overestimation이 Learning Policy의 Quality를 저하시킴
- ✓ 학습을 분산시키는 Double Q-Learning을 통해 Overestimation문제 해결

Part
02

Background

[Q-Learning]

- ✓ Sequential Decision 문제를 해결하기 위해 Action-Value Function Q(s, a)를 사용
- ✓ Q-Value는 어떤 시간 t에서 전략 π를 따라 행동 a를 할 때 미래의 보상들의 총합의 기대값을 의미
- ✓ Optimal Policy는 각 state에서 maximum value action을 선택함으로써 학습을 진행
- ✓ θ를 Weight로 갖는 Non-Linear Network Function Approximator를 통해 Approximate $Q(s,a;\theta) \simeq Q^*(s,a)$.

$$Q_\pi(s,a) = E_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... | S_t = s, A_t = a]$$

Part
02

Background

[Deep Q Networks]

- ✓ DQN은 target network와 Experience Replay를 사용하여 성능을 높임
- ✓ Target network를 통해 Target Value를 향해 Approximate 시킴
- ✓ Experience Replay를 통해 transition을 저장하고 이를 통해 학습 진행

Part
01

Abstract

[논문 요약]

- ✓ 근사값 $\hat{q}(s', a', W)$는 action-value function $q_\pi(s', a')$에 Noise(Y)를 더한 값이다.
  → $\hat{q}(s', a', W) = q_\pi(s', a') + Y_{s'}^{a'}$
- ✓ $Q^{approx}(s, a) = R_{t+1} + \gamma max_a Q^{approx}(s', a)$
- ✓ $Q^{target}(s, a) = R_{t+1} + \gamma max_a Q^{target}(s', a)$
- ✓ 두개의 차이를 구한 값은 $Z_s = \gamma(max_a Q^{approx}(s', a) - max_a Q^{target}(s', a))$ 해당한다.
- ✓ 여기서 Noise(Y)는 평균이 0인 정규분포를 따르지만, Max연산에 의해 선택된 Approximation Value는 Max operation에 의해 평균이 0인 특성을 따르지 않게 된다.
- ✓ 이러한 Max operation에 의해 Overestimation문제가 발생하고 propagate된다.

Part
02

Background

[Double Q-Learning]

✓ Q-Learning은 Action Select와 Action Evaluate을 위해 같은 θ를 사용 → Overestimation

$$Y_t^Q = R_{t+1} + \gamma Q(S_{t+1}, argmax_a Q(s_{t+1}, a; \theta_t); \theta_t)$$

✓ Selection과 Evaluation을 Decouple하여 아래와 같이 구분

$$Y_t^Q = R_{t+1} + \gamma Q(S_{t+1}, argmax_a Q(s_{t+1}, a; \theta_t); \theta_t')$$

✓ 각 Experience들을 통해 random하게 하나의 weight만을 학습

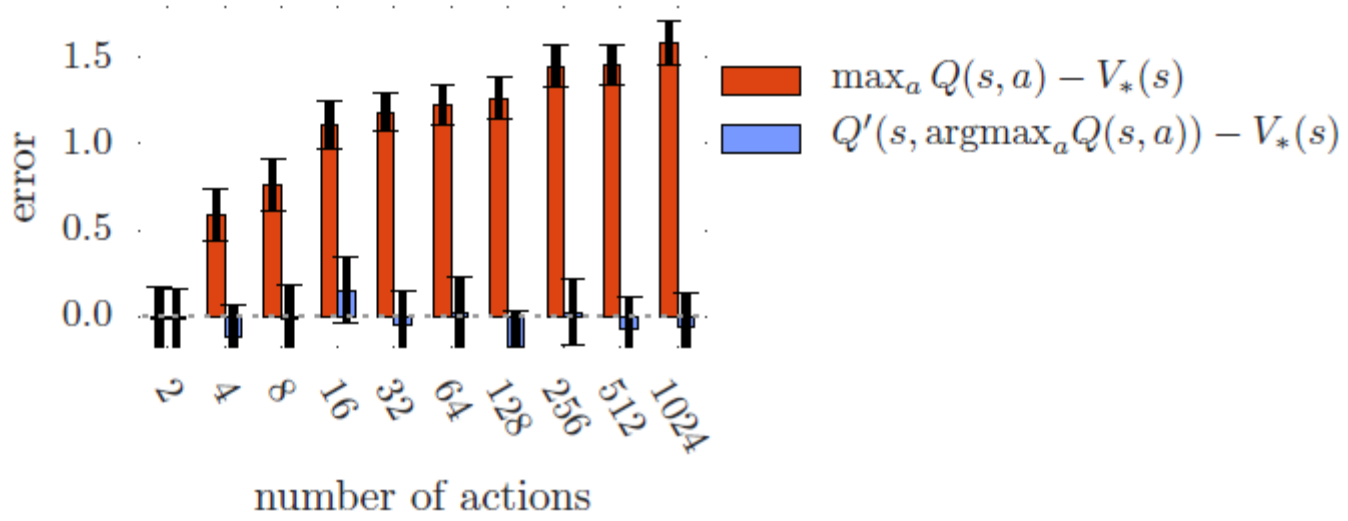
✓ 2개의 weights를 switching 함으로써 대칭적으로 weight update를 진행

Part
02

Overoptimism

[Due to Estimation Error]

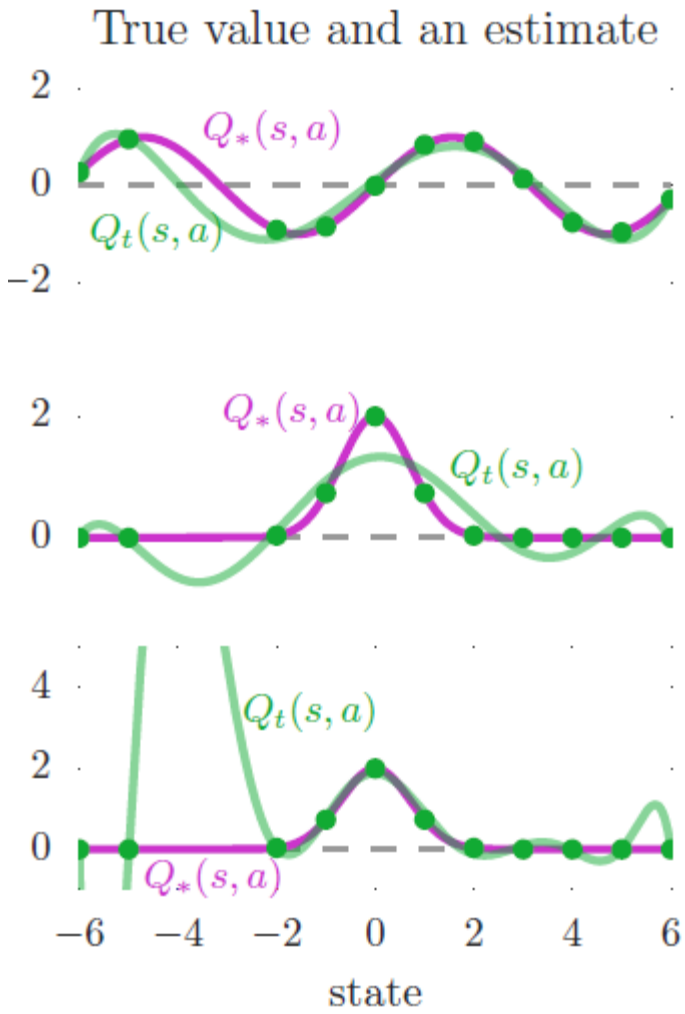
✓ Q-Learning은 Action이 증가함에 따라 overestimation도 커지는 현상을 보여줌



Part
02

Background

[Double Q-Learning]

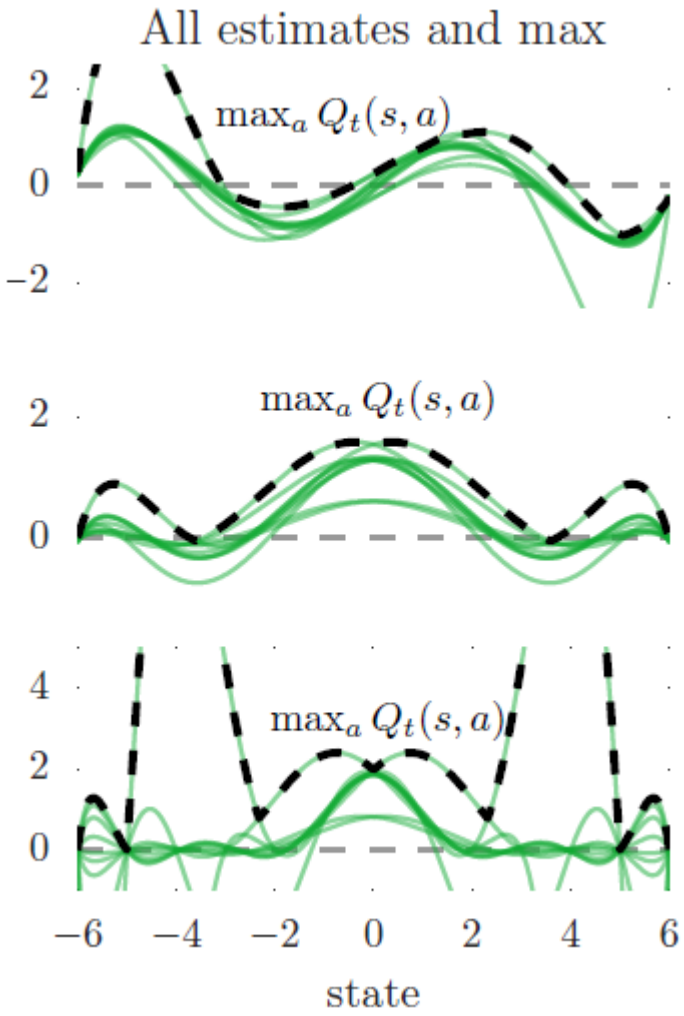


- ✓ 각 State는 10개의 Action을 가지며, True Value그래프와 1개의 Action에 대한 Estimation Value를 보여줌
- ✓ Estimation Function은 Sampled States에서의 True Value를 알고 있는 상황에서 구현됨
- ✓ Sampled States가 거리가 벌어진 경우, 더 큰 Estimation Error가 발생하였고, 제한적인 데이터만을 지니는 실제 환경과 유사

Part
02

Background

[Double Q-Learning]



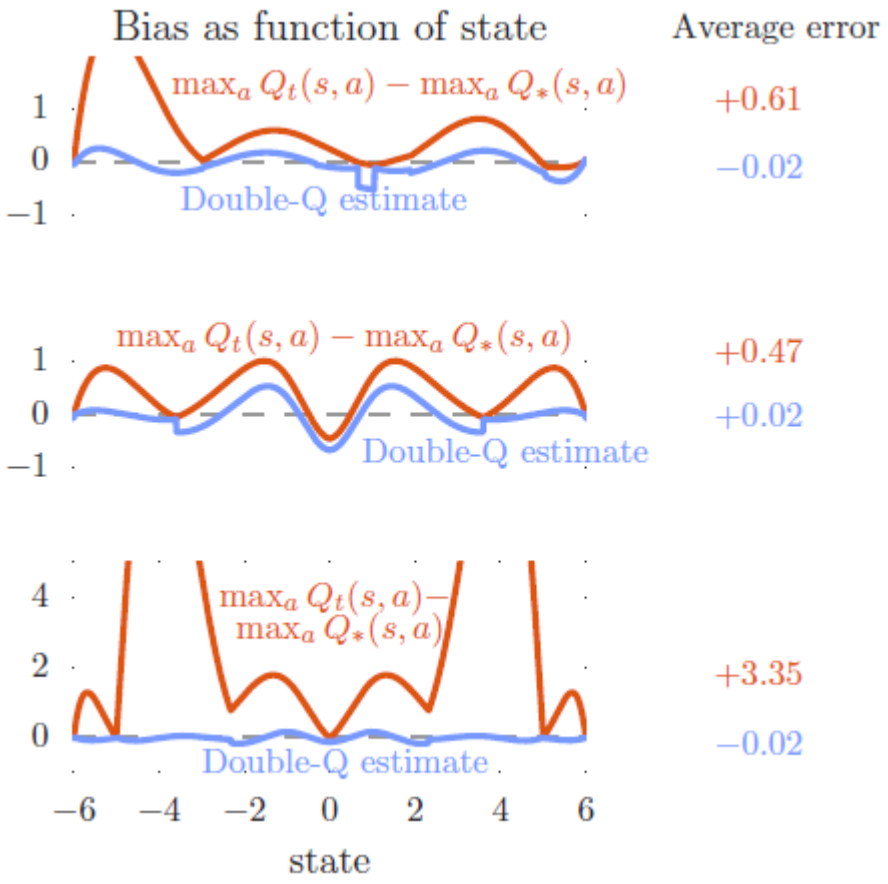
✓ 각 State에서 10개의 Action에 대한 Estimation Function

✓ 검정색 그래프는 Max값들을 바탕으로 그린 Graph

Part
02

Background

[Double Q-Learning]



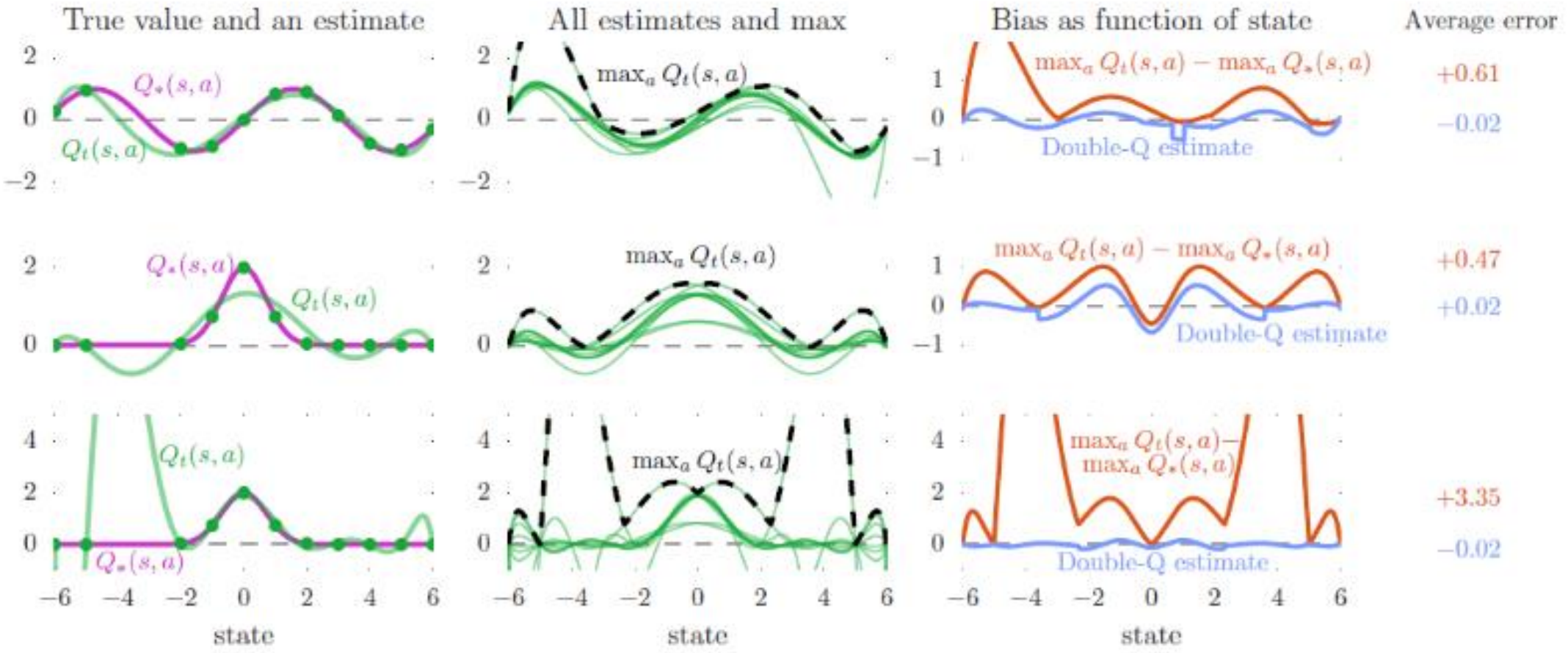
✓ 주황색 그래프는 Maximum Estimation Function – True Function

✓ 파란색 그래프는 Double Q Estimation Function – True Function

Part
02

Background

[Double Q-Learning]



Part
02

Background

[Double Q-Learning]

✓ row1, row2는 True Value Function만을 다르게 한 것 → Overestimation이 특정 구조에서만 발생 X

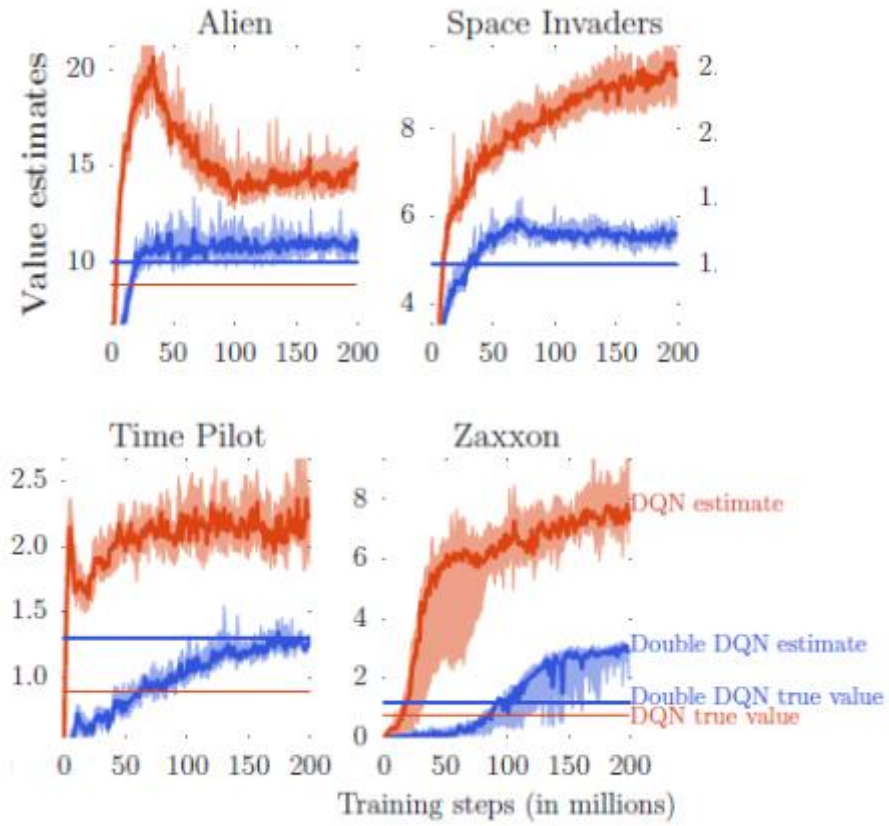
✓ row2, row3는 Function approximation의 flexibility가 다름
→ row2는 flexibility가 낮아서 True Value에서도 정확한 값을 갖지 않음
→ row3는 flexibility가 높지만 주어진 True Value거리가 먼 경우에서 정확한 값을 갖지 않음

✓ 이렇게 시작되는 Overestimation은 계속해서 propagate되고, 상황은 계속해서 악화됨

Part
02

Background

[Double DQN]

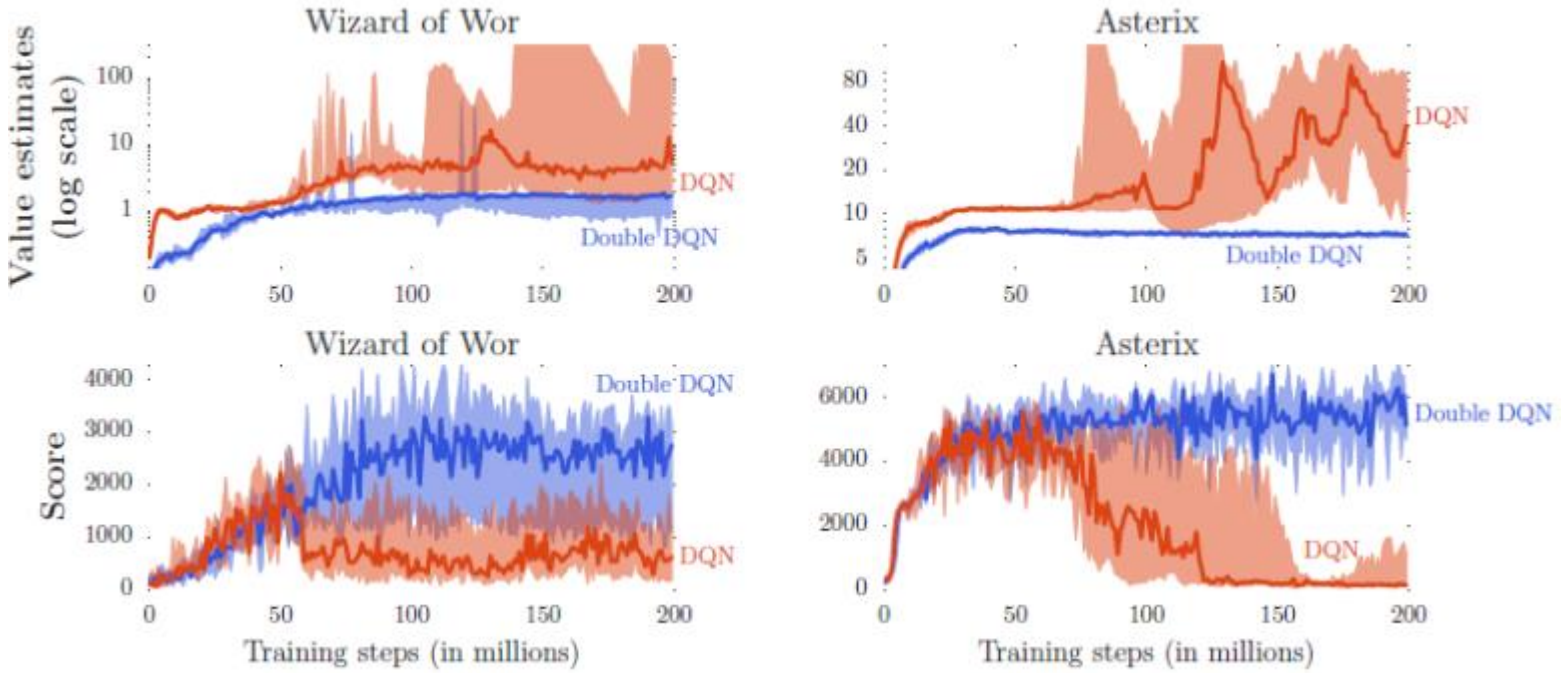


- ✓ 주황색 그래프는 DQN, 파란색 그래프는 Double DQN
- ✓ 직선은 True Value, 곡선은 Estimation Value
- ✓ Y값은 Actual Discounted Value
- ✓ DQN은 Overestimation이 Propagate되면서 성능이 안 좋음

Part
02

Background

[Double DQN]



✓ Overestimation이 발생하면서 Score가 하락함

Part
03

Quality of the Learned policy

[Double Q-Learning]

	DQN	Double DQN	Double DQN (tuned)
Median	47.5%	88.4%	116.7%
Mean	122.0%	273.1%	475.2%

$$score_{normalized} = \frac{score_{agent} - score_{random}}{score_{human} - score_{random}}$$

- ✓ Double DQN과 DQN은 같은 Hyper-parameter를 사용
  → $\epsilon = 0.05$ and 5분(18000 frames)로 Evaluation
- ✓ Double DQN과 DQN의 차이는 Only Target Network
- ✓ 해당 Hyper-parameter가 DQN에 optimized 되어있음에도 불구하고 Double DQN이 성능을 압도
- ✓ Tuning → $\epsilon = 0.01$ and Frame 수 증가

Part
04

Discussion

[논문 요약]

1. 왜 Q-Learning에서 Overestimation이 발생하는지를 설명
2. Atari Game을 Value Estimation 함으로써, Overestimation이 Common하게 발생함을 보여줌
3. Double Q-Learning을 통해 Overestimation을 줄임으로써, stable하고 reliable한 학습을 진행
4. 추가적인 Network나 Hyper Parameter의 변화 없이 기존 DQN의 값이나 구조를 사용
5. Double DQN은 더 좋은 Policy를 얻고, Atari Game에서 State-of-art 결과를 얻음

THANK YOU

For your attention