Deep Reinforcement Learning with Double Q-learning

2020.05.25

김 도 훈

Abstract

대중적인 Q-learning 알고리즘 → 특정 조건에서 action value를 overestimate(과대평가)하는 것으로 알려짐

그런데 이게 실제로 **일반적인지 / 성능에 해를 끼치는지 / 일반적으로 예방할 수 있는지**는 이전에는 알 수 없었음.

→ 이 논문에서 긍정적으로 답한다!

최근의 DQN 알고리즘(Q-learning + DNN) → overestimation으로 인해 몇몇 게임에서 좋은 성능을 보이지 못함

우리가 제안하는 Double Q-learning은 large-scale function approximation로 작동하도록 일반화 될 수 있음

DQN 알고리즘에 대해 구체적인 적용을 제안, 그 결과 overestimation을 감소시킬 뿐만 아니라, 여러 게임에서 훨씬 더 나은 성과를 이끌어 낸다는 것을 보여준다.

Introduction

- 강화학습의 목표 : 누적되는 미래의 보상 신호를 최적화하여 순차적 의사결정 문제에 대한 좋은 정책을 배우는 것
- Q-learning : 때때로 비현실적인 high action value를 학습 – underestimated value보다 overestimated value를 더 선호하는 경향이 있는 estimated action values에 대한 maximization step이 존재

- 이전 논문: overestimations이 불충분한 flexible function approximation과 노이즈에 기반했음
- 이번 논문 : 이러한 관점을 통합 & 근사 오차 값에 관계없이 action value가 부정확할 때 overestimation 발생 가능함을 보임

Introduction

- Overestimation이 그럼 실제 작동에 부정적인 영향을 미치는가 아닌가 → 열린 질문
- Overoptimistic value 추정은 그 자체로 반드시 문제가 되지는 않음 // 때때로는 더 좋기도 함!
- 그러나 overestimation이 균일하지 않고 우리가 더 학습하고자 하는 상태에 집중 안 해? → 그럼 부정적

- Overestimation이 practice 와 scale에서 발생하는지를 테스트하기 위해 최신 DQN 알고리즘의 성능 조사
- DQN = Q-learning + flexible deep neural network

  → Atari 2600 게임들에 대해 인간수준의 능력까지 올라옴 // 그리고 이 설정은 Q-learning의 최적사례
- 근데 음 놀랍게도 이러한 비교적 유리한 설정에서도 때때로 DQN은 action values를 상당히 overestimate한다.

Introduction

- 표 형식(tabular setting)으로 처음 제안된 Double Q-learning algorithm 이면에 있는 아이디어가 deep neural network를 포함한 임의 함수 근사치로 작동하도록 일반화될 수 있음을 보여줌

  → 우리는 이것들 Double DQN이라는 새로운 알고리즘이라고 할거임!

- 이건 더 정확한 value estimates를 도출할 뿐만 아니라, 여러 게임에서 훨씬 높은 점수로 이어진다는 것을 보여줌!
- 이것은 DQN의 everestimation이 실제로 더 나쁜 정책으로 이어졌고 이를 줄이는 것이 낫다는 것을 보여줌

Summary

- Q-learning algorithm에는 maximization step이 존재 → **Overestimation**을 발생시킨다!
- Overestimation이 learning policy의 **quality를 저하**시킨다
- 학습을 분산시키는 Double Q-learning을 통해 overestimation 문제 해결!

+) Overestimation이란?

- 어떤 대상을 근사(apporoximate) 한다는 것 → 어느 정도의 오차를 감안하고 대략적으로 측정하는 것!
- Optimal policy π를 따르는 action-value function $Q_\pi(s', a')$에 Noise(Y)를 더한 값은 근사값 $\hat{Q}(s', a', W)$와 같고, 이를 정리하면 $\hat{Q}(s', a', W) = Q_\pi(s', a') + Y_{s'}^{a'}$와 같다.

Background

순차적 의사결정 문제(sequential decision problem)을 해결하기 위해 각각의 action에 대한 optimal value에 대한 추측값을 학습할 수 있고, 이러한 action을 취할 때 그리고 이후에 최적의 정책을 따를 때 예상되는 미래 보상들의 합으로 정의된다.

주어진 policy를 π, state s에 있는 action의 true value를 a라고 하면,

$$Q_\pi(s,a) \equiv \mathbb{E}\left[R_1 + \gamma R_2 + \ldots \mid S_0 = s, A_0 = a, \pi\right]$$

이고 $\gamma \in [0,1]$인 즉각적인 보상과 나중의 보상의 중요성을 상쇄하는 discount factor

Optimal policy - 여러 action value 값 중에서 max 값을 선택하는 것

$$Q_*(s,a) = \max_\pi Q_\pi(s,a)$$

Background

시간차이 학습의 형태인 Q-learning 을 사용해서 최적의 action value를 추정하는 것을 학습할 수 있다.
대부분 흥미로운 문제들은 너무 커서 모든 상황에 개별적으로 action value들을 학습할 수 없었다.

대신에 우리는 파라미터화 된 value function을 학습할 수 있다

$$Q(s, a; \boldsymbol{\theta}_t)$$

Standard Q-learning은 state S_t안의 action A_t 후에 파라미터들을 업데이트 해주고, immediate reward R_{t+1}와 resulting state S_{t+1}를 관측하면

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha(Y_t^{\mathrm{Q}} - Q(S_t, A_t; \boldsymbol{\theta}_t))\nabla_{\boldsymbol{\theta}_t} Q(S_t, A_t; \boldsymbol{\theta}_t)\,.$$

α가 scalar step size이고 타겟 Y_t^Q가 다음과 같이 정의된다

$$Y_t^{\mathrm{Q}} \equiv R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \boldsymbol{\theta}_t)\,.$$

이 업데이트는 확률적 그래디언트 감소와 비슷하고, 현재 값 $Q(S_t, A_t; \boldsymbol{\theta}_t)$ 를 타겟 값 Y_t^{Q}. 를 목표로 업데이트

Background

[Deep Q Networks]

DQN – multi-layered neural network
주어진 상태 s에 대해 action value의 vector $Q(s,\cdot\,;\theta)$ 출력, θ는 네트워크의 파라미터

$$Y_t^{\mathrm{DQN}} \equiv R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \boldsymbol{\theta}_t^-)$$

Background

[Double Q-learning]

Max operator in standard Q-learning & DQN – use same values both to select and to evaluate an action

$$Y_t^{\mathrm{Q}} \equiv R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \boldsymbol{\theta}_t)\,. \qquad Y_t^{\mathrm{DQN}} \equiv R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \boldsymbol{\theta}_t^-)$$

→ Overestimated 값을 더욱 선택하게 하며, overoptimistic 한 결과를 불러 일으킴!

그래서! Q-learning에서 selection과 evaluation을 분리 – 보다 명확한 비교를 위해 → 이게 Double Q-learning

$$Y_t^{\mathrm{Q}} = R_{t+1} + \gamma Q(S_{t+1}, \operatorname*{argmax}_a Q(S_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}_t)$$

Background

기존의 Q 함수

$$Y_t^{\mathrm{Q}} \equiv R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \boldsymbol{\theta}_t) .$$

위 식을 select와 evaluation을 명확히 하기 위해 아래와 같이 변형

$$Y_t^{\mathrm{Q}} = R_{t+1} + \gamma Q(S_{t+1}, \operatorname*{argmax}_a Q(S_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}_t)$$

위 식을 select와 evaluation를 분리하면 → Double Q-learning

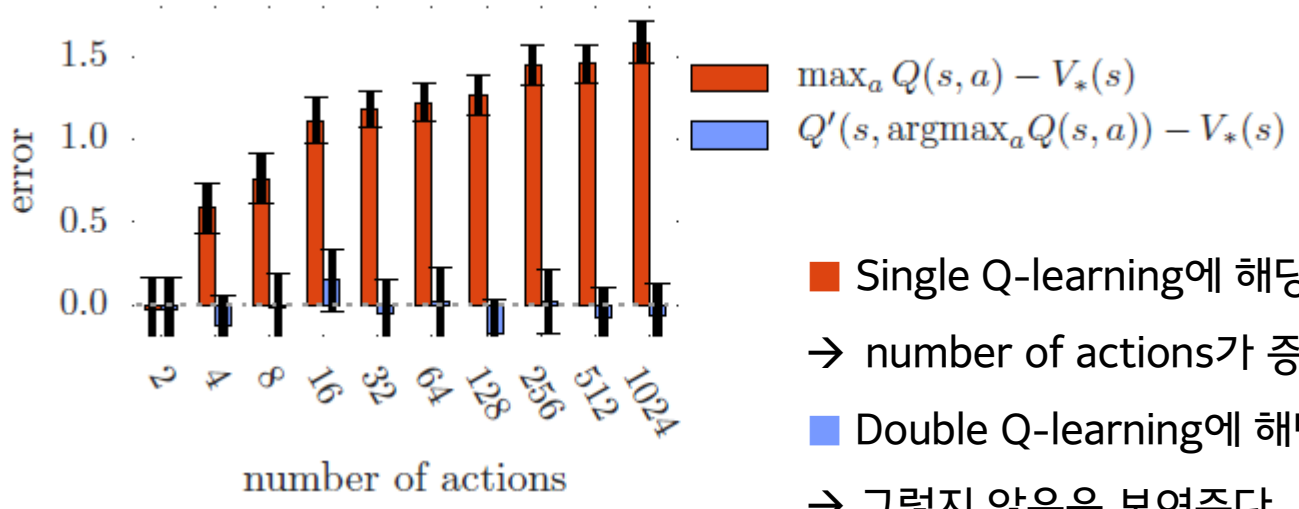
$$Y_t^{\mathrm{DoubleQ}} \equiv R_{t+1} + \gamma Q(S_{t+1}, \operatorname*{argmax}_a Q(S_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}'_t)$$

θ_t는 argmax에서 action을 선택, θ_t'는 policy 를 evaluate → 2개의 weights는 서로 switching하며 update

Overoptimism due to estimation errors

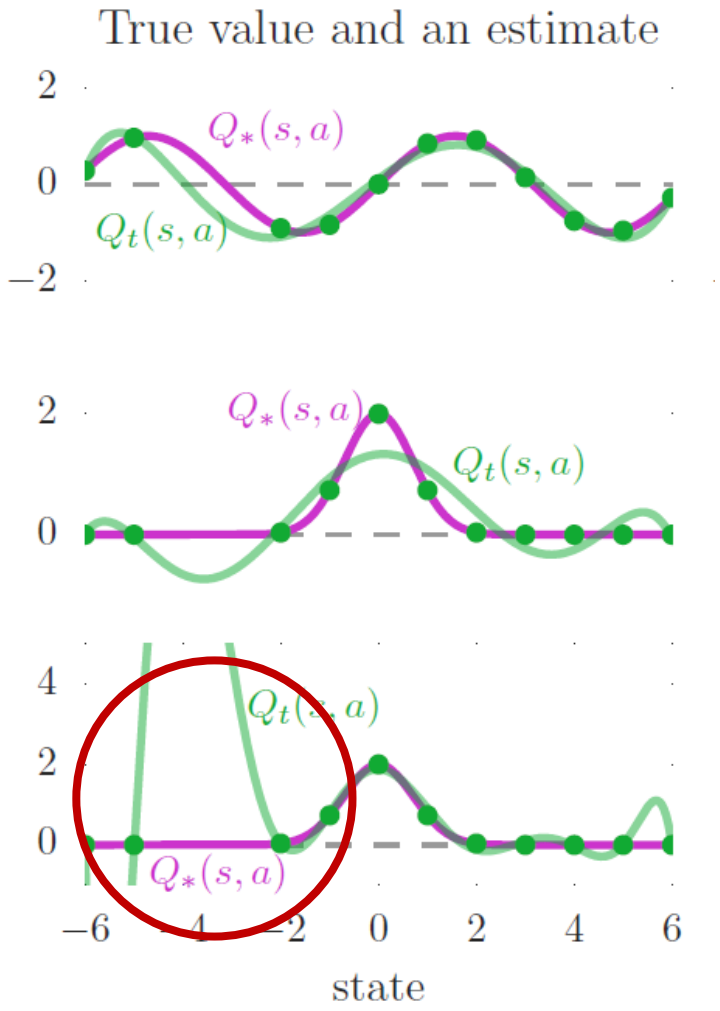
우리는 overestimation 의 lower bound 와 upper bound를 구할 수 있다.
→ Theorem 1.이 lower bound 증명, upper bound는 93년도 Thrun & Schwartz 논문에서 제시

Q-learning은 Action이 증가함에 따라 overestimation도 커지는 현상을 보여준다.



- Single Q-learning에 해당하는 빨간 그래프
  → number of actions가 증가함에 따라 overestimation이 증가
- Double Q-learning에 해당하는 파란 그래프
  → 그렇지 않음을 보여준다.

Overoptimism due to estimation errors



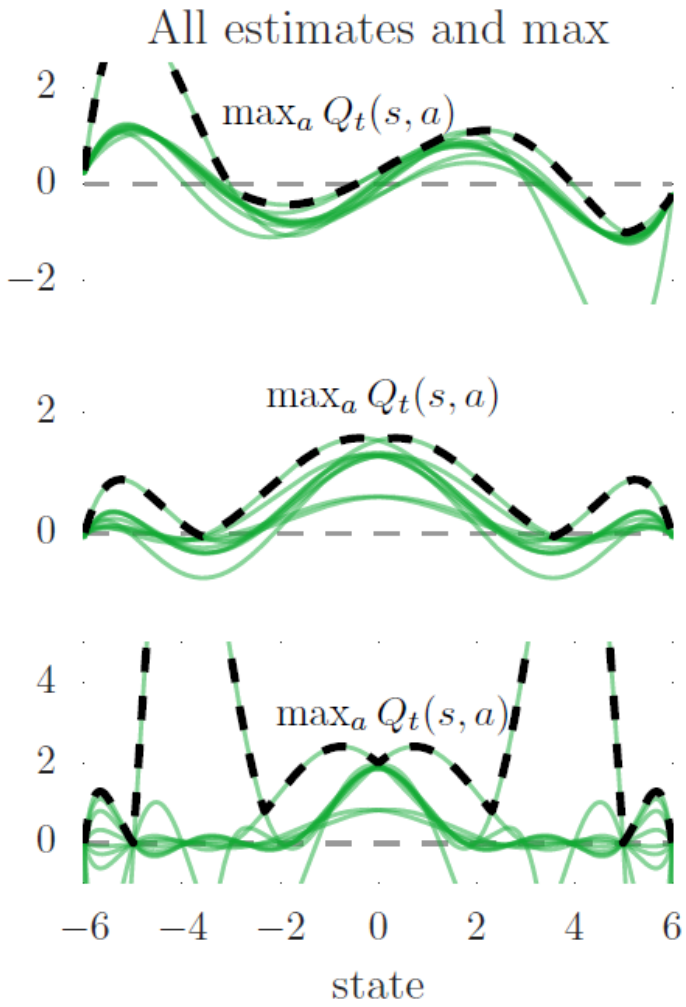
각 State는 10개의 action을 가짐.
우리가 고려하는 상황의 true optimal action values는 state에만 의존
→ 각 state에서 모든 action은 같은 true value 가짐

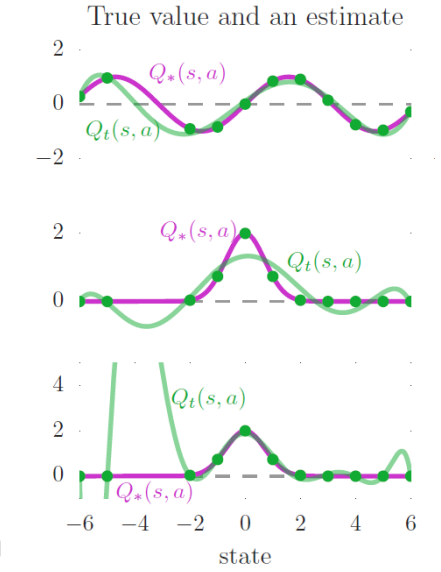
■ true value graph / ■ single action approximation

Estimation Function은 sampled states(초록 점들)에서의 true value를 알고 있는 상황에서 구현됨

Sampled states가 거리가 벌어진 경우, 더 큰 estimation error가 발생하였고, 제한적인 데이터만을 지니는 실제 환경과 유사

Overoptimism due to estimation errors





true value graph
single action approximation

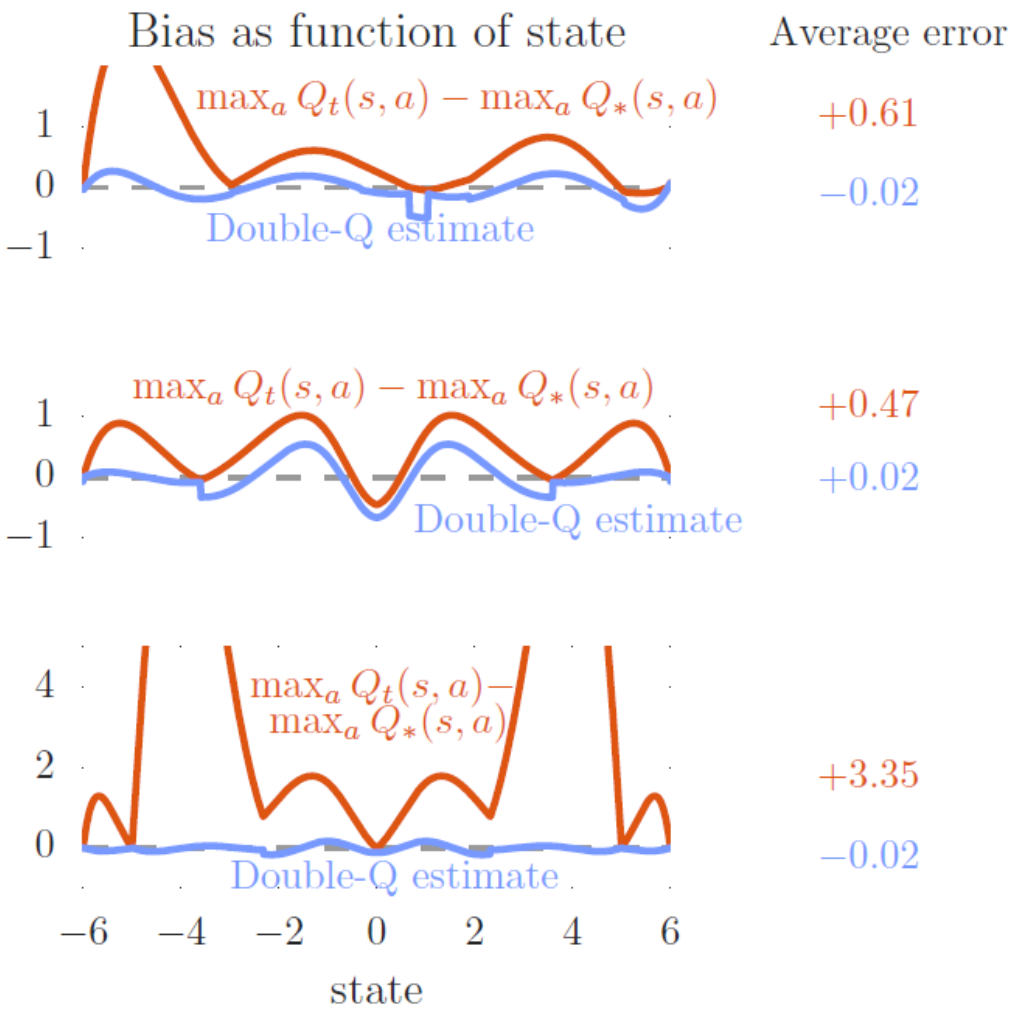
각 state에서 10개의 action에 대한 estimation function(초록 선들)

검은색 점선: maximum action value in each state

Estimate에 대한 함수는 d-degree 방정식으로 앞에서 sample한 state에서 true value와 같은 값을 갖는다.

Maximum값은 true value보다 높게 나타나곤 함

Overoptimism due to estimation errors



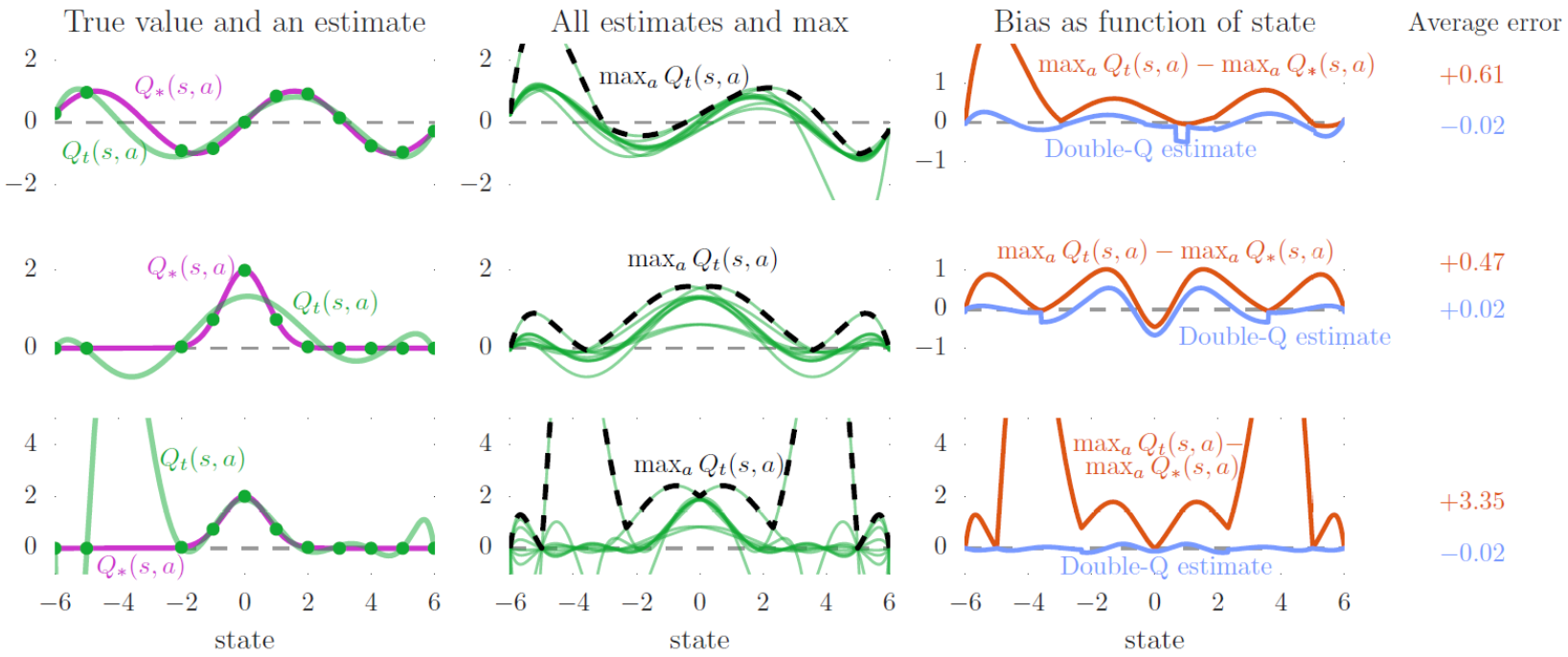
Maximum은 때때로 ground truth보다 높음

■ 주황색 그래프 → Maximum Estimation과 True Value의 차이 upward bias를 지니기에 항상 positive한 값

■ 파란색 그래프는 Double Q-Learning을 활용한 Estimate 평균적으로 거의 0에 근접

Double Q-learning이 성공적으로 Q-Learning의 Overestimation을 줄였음을 보여준다.

Overoptimism due to estimation errors



Row1, row2는 true value function만을 다르게 한 것 → Overestimation이 특정 구조에서만 발생하지 않음

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 악화된다.

Overoptimism due to estimation errors

overestimation은 어느 상황에서나 발생할 수 있음

OverEstimation은 정확한 action-value 값을 갖는 상태에서도 발생

그러나 uniformly overestimating된 값들은 문제가 되지 않는다. 왜냐면 다른 상태와 다른 행동에 대해 estimation error가 다르기 때문

overestimation은 exploration bonus가 특정 state나 action에 대해 uncertain한 값으로 주어지는 optimism과 혼동되면 안된다. 여기서 overestimation은 update를 한 후에 발생하는 것으로, certainty하게 overoptimism을 유발한다. Optimism과 달리 uncertainty에도 불구하고, 이러한 overestimation은 최적의 policy를 학습하는데 방해가 될 수 있다. 그리고 우리는 policy quality가 받는 negative effect를 실험에서 확인할 수 있다.

Double DQN

Double Q-learning의 기본 idea
→ action selection과 action evaluation으로 max operation을 분리함으로써 overestimation을 줄이는 것!

비록 완전히 분리되지는 않지만, network를 추가하지 않고 DQN 구조의 target network는 후보군의 value function을 제안한다.

$$Y_t^{\text{DoubleDQN}} \equiv R_{t+1} + \gamma Q(S_{t+1}, \underset{a}{\text{argmax}}\, Q(S_{t+1}, a; \boldsymbol{\theta}_t), \boldsymbol{\theta}_t^-)$$

Double Q-learning과 달리 second network θ_t' 현재 greedy policy의 evaluation을 위해 target network θ_t^- 로 replaced 되었다.

$$Y_t^{\text{DoubleQ}} \equiv R_{t+1} + \gamma Q(S_{t+1}, \underset{a}{\text{argmax}}\, Q(S_{t+1}, a; \boldsymbol{\theta}_t); \boldsymbol{\theta}_t')$$

Empirical results

DQN의 overestimations에 대해서 분석
Double DQN이 value accuracy와 policy quality의 관점에서 성능이 좋아짐을 증명함

이러한 접근법의 견고함을 테스트 → random start로 알고리즘을 평가

목표: screen의 pixel값 만을 입력으로 사용, 독립적인 여러 개의 게임을 하나의 알고리즘을 통해 고정된 parameter로 학습하는 것

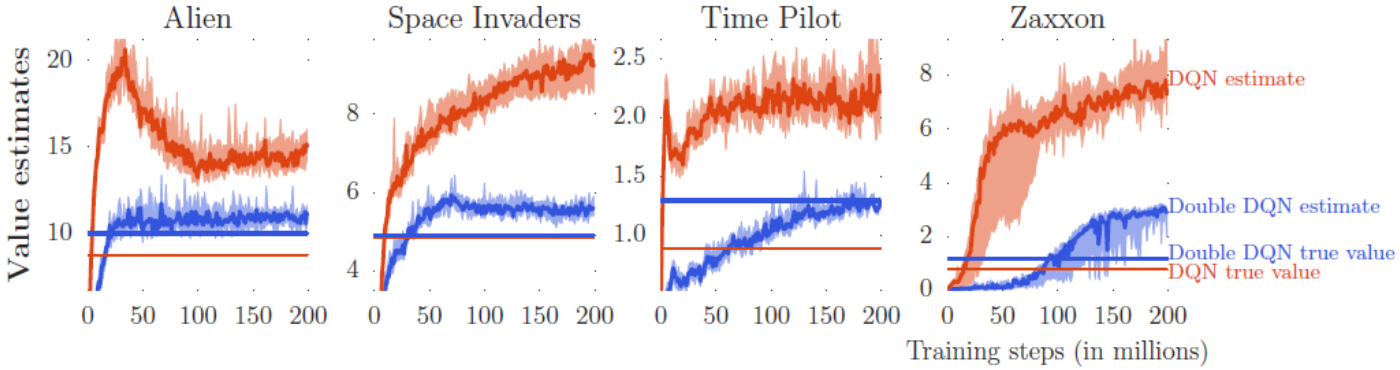
입력: high-dimensional, 다양한 종류의 게임

좋은 해결책은 learning algorithm을 통해 학습하는 것, tuning하여 특정 domain에 대해 끼워 맞추는 것은 실현가능함

Network Architecture → 3개의 convolution layer & fully connected된 hidden layer를 지나는 CNN
입력으로 마지막 4개의 frame 사용, 각 행동에 대한 action value를 output으로 보냄
각 게임에서 network는 200M frames에 대해 하나의 GPU로 학습, 약 1주일 걸림.

Empirical results

[Results on overoptimism]



6개의 Atrai 게임에서 DQN의 overestimation에 대한 예제를 보여줌

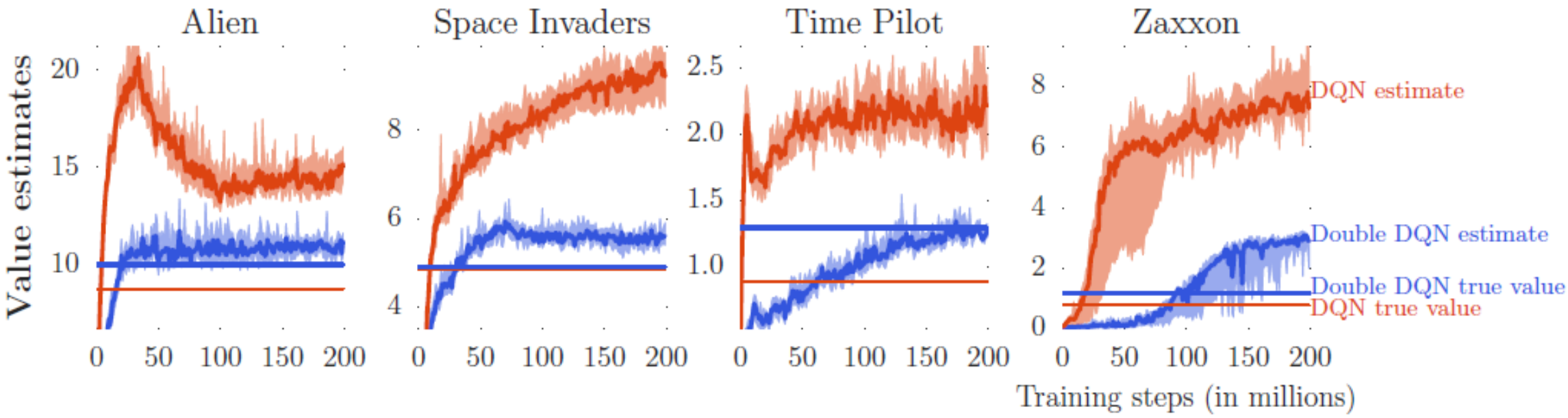
- 주황색 직선과 학습 곡선 → DQN이 끊임없이, 때로는 과하게 current greedy policy에 대해 overiptimistic하다

이 그래프에서 y값은 best learned policy에 대한 actual discounted value임

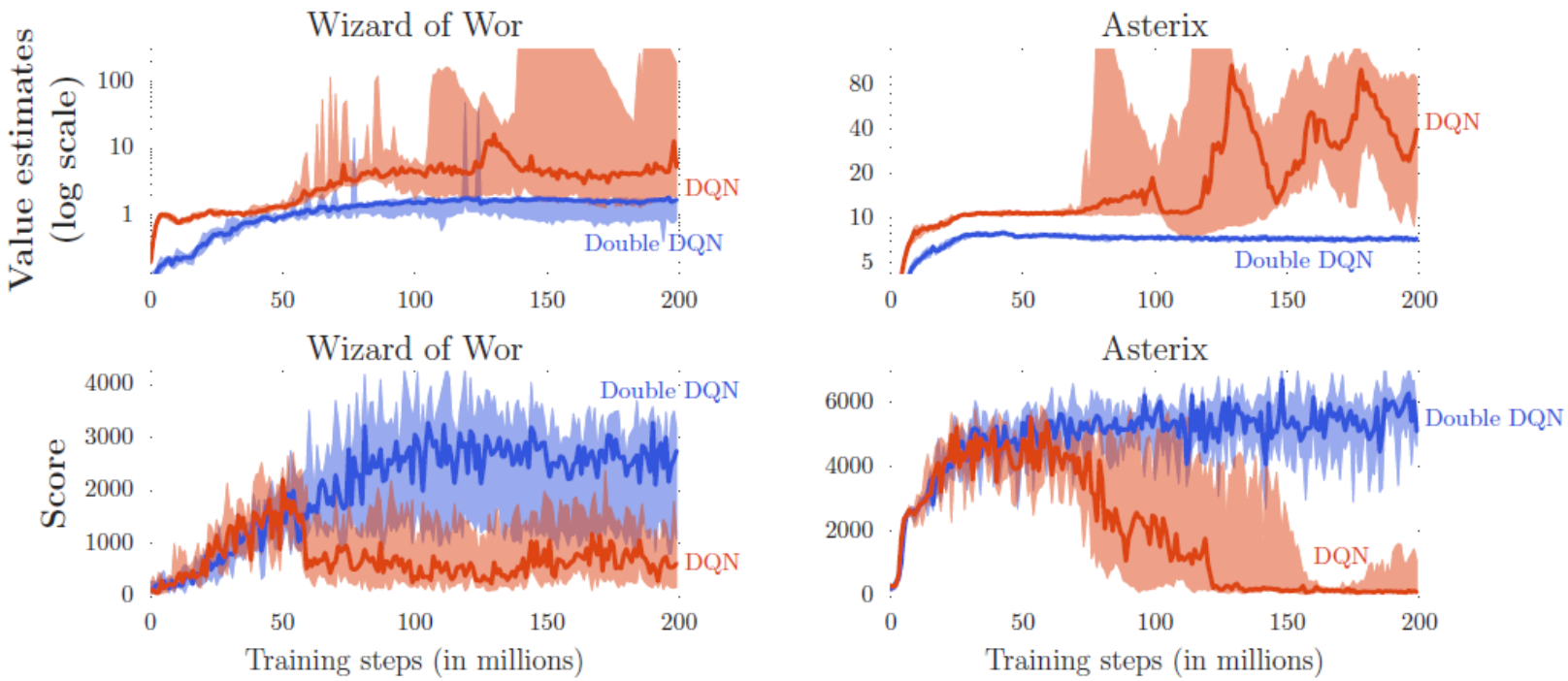
각 (averaged) value estimates는 다음과 같이 계산

$$\frac{1}{T}\sum_{t=1}^{T}\operatorname*{argmax}_{a} Q(S_t, a; \boldsymbol{\theta})\,.$$

Empirical results



Empirical results



Overestimation이 발생 → score가 떨어지는 모습을 볼 수 있다.

Empirical results

[Quality of the learned policies]

Overestimation이 항상 learned policy의 quality에 악영향? → 꼭 그런건 아니다! (ex. Pong game)

하지만, overestimation을 줄이는 것 → 안정된 학습을 할 수 있게 도와준다!

	DQN	Double DQN
Median	93.5%	114.7%
Mean	241.1%	330.3%

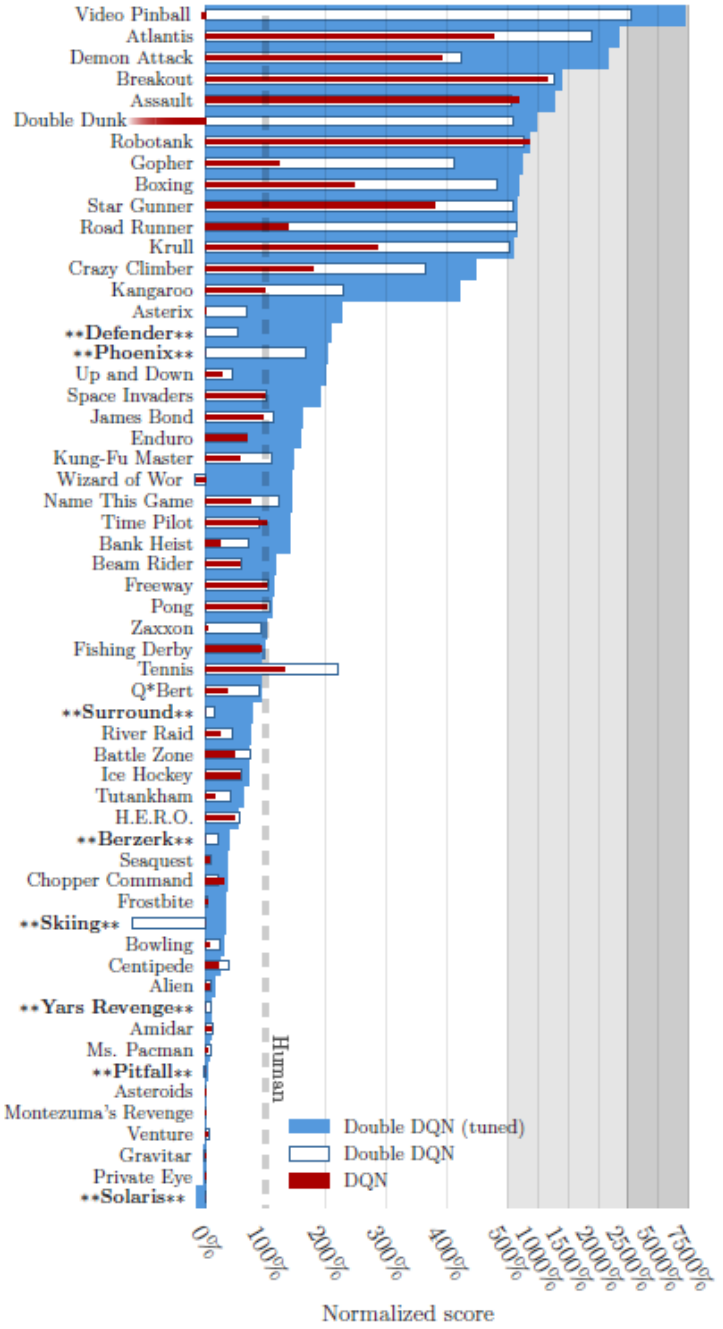
$$\text{score}_{\text{normalized}} = \frac{\text{score}_{\text{agent}} - \text{score}_{\text{random}}}{\text{score}_{\text{human}} - \text{score}_{\text{random}}}$$

- Double DQN과 DQN은 같은 Hyper-parameter 사용
→ 학습된 policy들은 ϵ=0.05 를 지니는 greedy policy와 함께 5분 (18000 frames) 동안 평가
- 둘의 차이점은 target network!
- Hyper-parameter가 DQN의 것에 맞춰져 있어 적합하지 않을 수도 있음에도 Double DQN이 성능 더 좋음!
- Tuning → ϵ=0.01 & frame 수 증가

Empirical results

[Robustness to Human starts]

Double DQN이 더 나은 결과를 도출해낸다!



Discussion

이 논문 → 5가지 기여

① Q-learning이 대규모 문제에서 왜 지나치게 overoptimistic 할 수 있는지 보임

② Atari game에 대한 value estimates를 분석함으로써, overestimation이 더 흔하고 심각하다는 것을 보여줌

③ Double Q-learning이 overestimation를 성공적으로 감소시켜 보다 안정적이고 신뢰할 수 있는 학습으로 이어질 수 있다는 것을 보여줌

④ DQN 알고리즘의 기존 구조와 DNN을 추가 네트워크나 파라미터를 요구하지 않고 활용하는 Double DQN이라는 구체적인 구현을 제안함

⑤ Double DQN이 atari 2600 domain에서 새로운 state-of-the-art 결과를 얻으면서 더 나은 policy를 찾는다는 것을 보여줌

Reference

[1] Deep Reinforcement Learning with Double Q-learning (https://arxiv.org/pdf/1509.06461.pdf)

[2] https://mangkyu.tistory.com/66

[3] https://parkgeonyeong.github.io/Double-DQN의-이론적-원리/

Q&A

발표 때 나온 질문들이고,

하나씩 답변을 달아보겠습니당

- What do you mean by "large-scale function estimation approximation
- 그래프에서 에러 수치의 기준이 무엇인가요? 13쪽 y축 에러
- 혹시 현재 Double Q-Learning에 대해서 overestimation의 Upper/Lower bound에 대해 보여준 Article이 존재하나요?
- dqn과 double dqn의 true value가 다른 이유가 잘 이해가지 않습니다.
- Experimental Result 설명하는 곳에서 graph에서 보이는 상하진동(?)의 의미랑 Truth Line(?)에서의 offset의 의미를 놓쳤습니다. 다시 설명해주실수있나요?
- 추가 네트워크를 않았다는 것은 Estimate Function과 Max? Function의 네트워크 구조는 동일하나 학습에 사용하는 input sample이 다르다는 건가요?
- 왜 double로 쓰면 overestimate가 줄어드는거에요?
- 듀얼이랑 비교해주실수 있나요 어떤점이 다른거에용
- 최종적인 best를 찾기위해 현 시점 max가 아닌 다른 값을 넣는다는 개념으로 생각해도 되나요?
- Double DQN에서는 두 네트워크를 서로 다른 샘플로 학습시켜서 overestimate의 가능성을 줄이는 건가요?
- 듀얼로 했을때도 overestimate가 줄어드나용?
- 제 질문의 요지는 double dqn에서 overestimation이 덜 일어나는 이유에용 network를 두개로 분리해서 얻는 이점이 정확히 뭔지 조금 이해가 덜 되어서요

Q&A

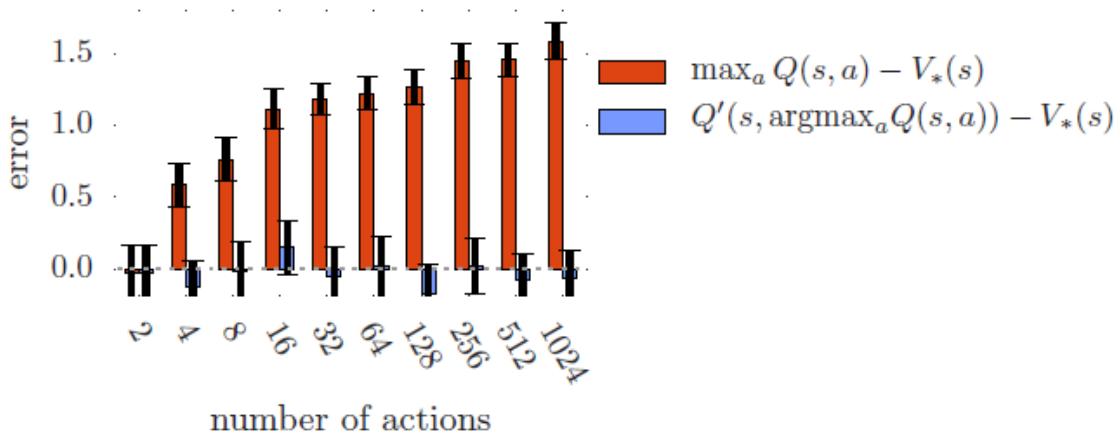
Q1. What do you mean by "large-scale function estimation approximation”?

A1.

문제에서 주어진 상황이 large-scale이기 때문에 large-scale function estimation approximation이라고 하는 것 같습니다. 찾아본 결과 받아들여 지는걸로 봐선 그냥 large-scale의 function estimation이 approximate 하는 것으로 이해하면 될 것 같습니다.

Q&A

Q2. 그래프에서 에러 수치의 기준이 무엇인가요?



A2.

논문에서 가져오면 다음과 같습니다.

while Double Q-learning is unbiased. As another example, if for all actions $Q_*(s,a) = V_*(s)$ and the estimation errors $Q_t(s,a) - V_*(s)$ are uniformly random in $[-1, 1]$, then the overoptimism is $\frac{m-1}{m+1}$. (Proof in appendix.)

여기서 말하는 error는 estimation errors로, $Q_t(s,a) - V_*(s)$를 의미하는 것 같습니다.

Q-value – True value인듯합니다.

Q&A

Q3. 혹시 현재 Double Q-Learning에 대해서 overestimation의 Upper/Lower bound에 대해 보여준 Article이 존재하나요?

A3.

네, 논문에 명시되어 있습니다.

Q&A

Q4. dqn과 double dqn의 true value가 다른 이유가 잘 이해가지 않습니다.

A4.

Orange straight line → represent the actual discounted value of the best learned policy
More precisely, the (averaged) value estimates are computed regularly during training with full evaluation phases of length T=125000 steps

The ground truth averaged values are obtained by running the best learned policies for several episodes and computing the actual cumulative rewards

The straight horizontal orange (for DQN) and blue (for Double DQN) lines in the top row are computed by running the corresponding agents after learning concluded, and averaging the actual discounted return obtained from each visited state. These straight lines would match the learning curves at the right side of the plots if there is no bias.

Q&A

(추가질문)
So far, I think that they basically have two Q-functions that have different experience sets and alternate at every step when bootstrapping

(참고 : https://www.reddit.com/r/MachineLearning/comments/57ec9z/discussion_is_my_understanding_of_double/)

Yes, you are essentially correct. In the original double Q-learning algorithm there are two action-value functions, and we update one of these for each sampled transition.

More precisely, we update one value function, say Q1, towards the sum of the immediate reward and the value of the next state. To determine the value of the next state, we first find the best action according to Q1, but then we use the second value function, Q2, to determine the value of this action.
Similarly, and symmetrically, when we update Q2 we use Q2 to determine the best action in the next state but we use Q1 to estimate the value of this action.
The goal is to decorrelate the selection of the best action from the evaluation of this action. You don't need two symmetrically updated value functions to do this. In our follow-up work on Double DQN
(https://arxiv.org/abs/1509.06461) we instead used a slow moving copy to evaluate the best action according to the main Q network. This turns out to decorrelate the estimates sufficiently as well.