SmartConnected.World 2018
AI EXPO KOREA 2018

#Session 5.

Double, Dueling DQN

김진호

경희대학교 빅데이터연구센터

경희대학교 소셜네트워크과학과

Problem of DQN

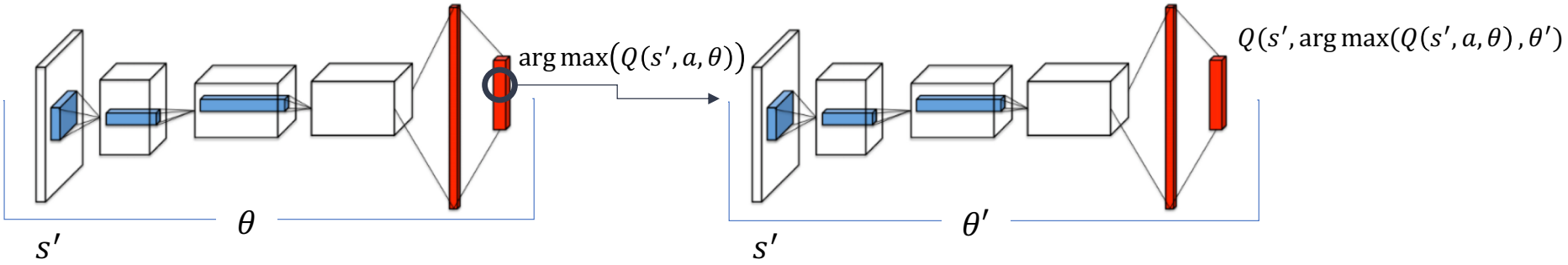
- DQN의 문제점은,
  - 각 상태에서 잠재적으로 Action의 Q값을 과대평가한다는 점이다.
  - 최적화되지 못한 액션이 최적화된 액션보다, 주기적으로 높은 Q를 갖는다면?
  - 혹은, 아래 표와 같다면

Real Q_{target}	-1	-1	+1	+1
Q_{target}	+1	+1	-1	-1
$Q_{current}$	+1	-1	+1	-1
Square	0	4	4	0
Loss	0	4	4	0

  - 잘못된 학습이 강화되는 현상이 발생한다.

Double DQN

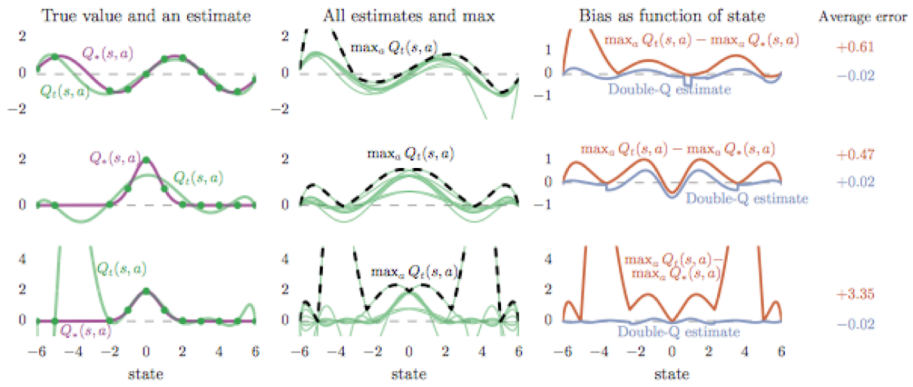
- 그렇다면, 자기자신에게 복사하지 말고 다른 걸 하나 만들면 되지 않을까?
  - 새로운 네트워크를 하나 만들어서, Q_{target}으로 쓰자 (AAAI 2, p. 5, (2016))
  - Q_{target}을 한번에 계산하지 말고, Action을 제 1 네트워크에서 구하고, 해당 액션에 대한 Q_{target}을 구하자.



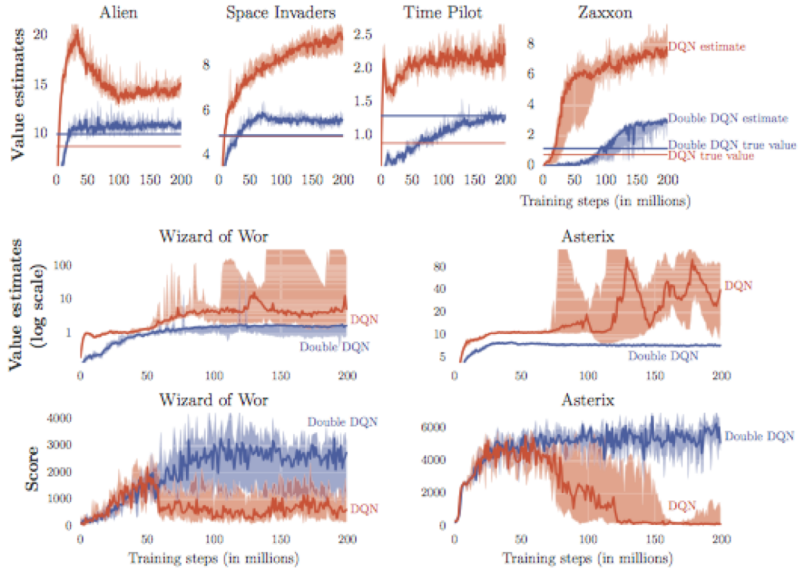
$$Q_{\text{target}} = r + \gamma Q(s', \arg\max(Q(s', a, \theta), \theta')$$

$$L = \sum (Q_{target} - Q_{current})^2$$

Double DQN



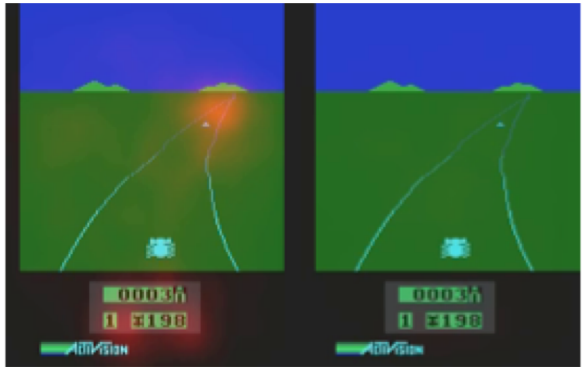
DQN과 DDQN의 Q-Value비교



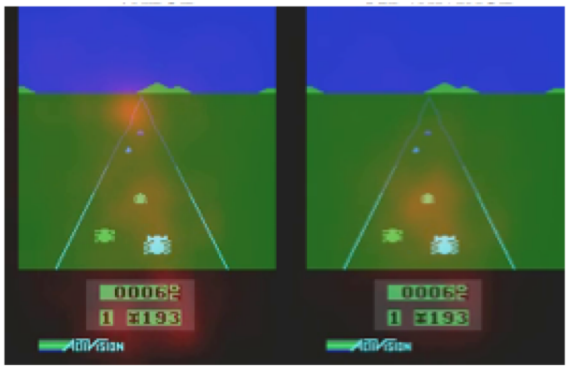
Atari에 대한 Value Estimation (위), Score(아래)

Dueling DQN

- Q는 무엇일까?
  - 특정 State에서 취해진 특정 Action이 얼마나 좋은지를 나타낸다.
  - 특정 Action이 좋다는 것은, 사실 하나의 Output 사이의 비교이다.



게임의 목표는 멀리 멀리 가는 것이다.



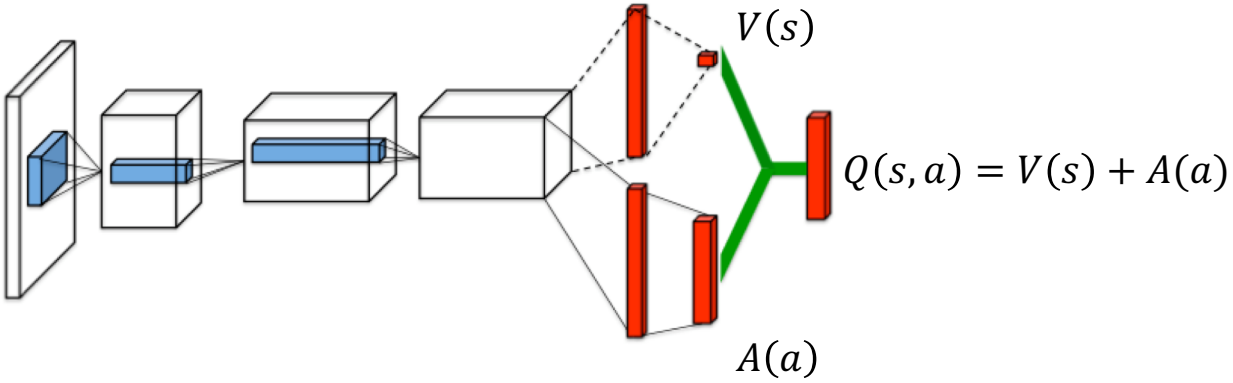
과연, 장애물이 신경쓰일까?

Dueling DQN

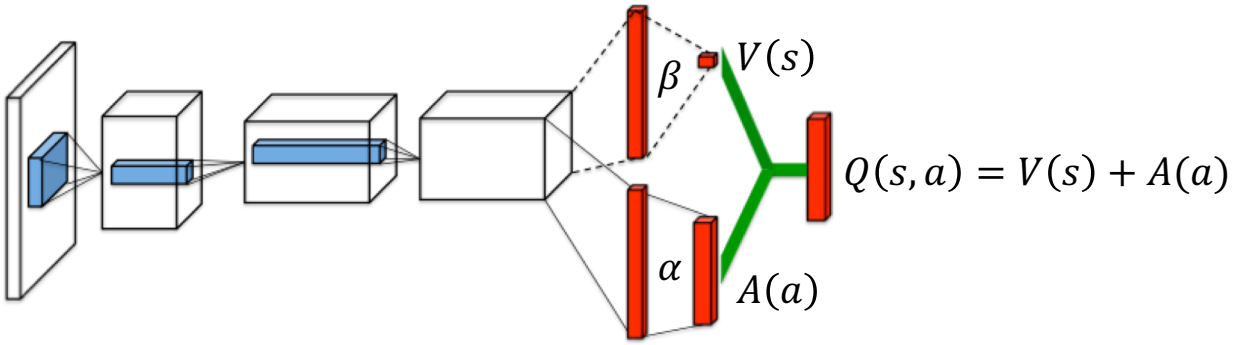
- 주어진 State에서의 Action은 두 가지 개념으로 분리가 가능하다.
  - Value Function $V(s)$, 단순히 어떤 상태가 얼마나 좋은지를 수치화 한 것
  - Advantage $A(a)$, 다른 액션에 비해 특정 액션을 취하는 것이 얼마나 좋은지를 수치화 한 것

$$Q(s,a) = V(s) + A(a)$$

  - 두 값을 합쳐서 Q를 표현하는 것, Dueling Network



Dueling DQN

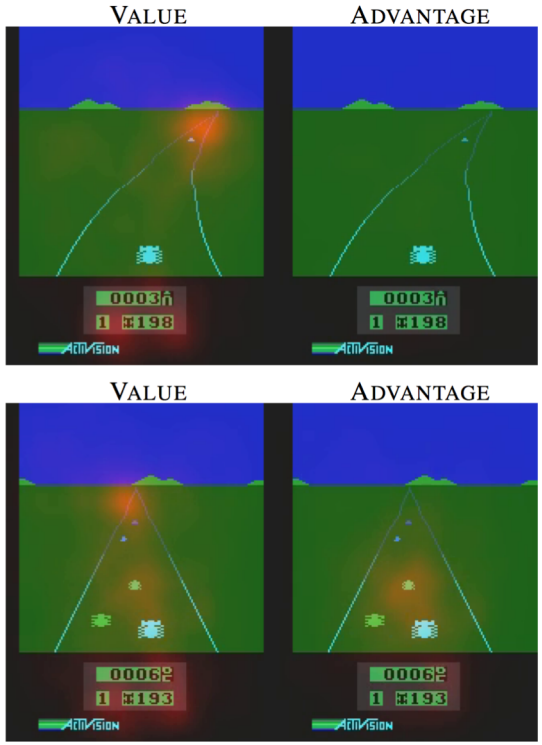


$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + A(s,a;\theta,\alpha)$$

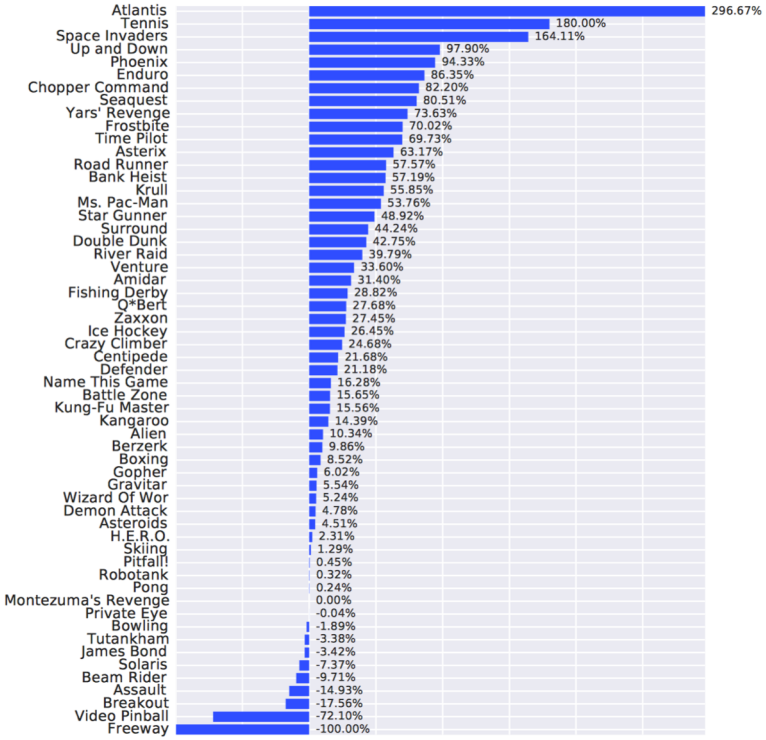
- α, β are parameters of the two streans of fully-connected layers.

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a';\theta,\alpha) \right)$$

Dueling DQN



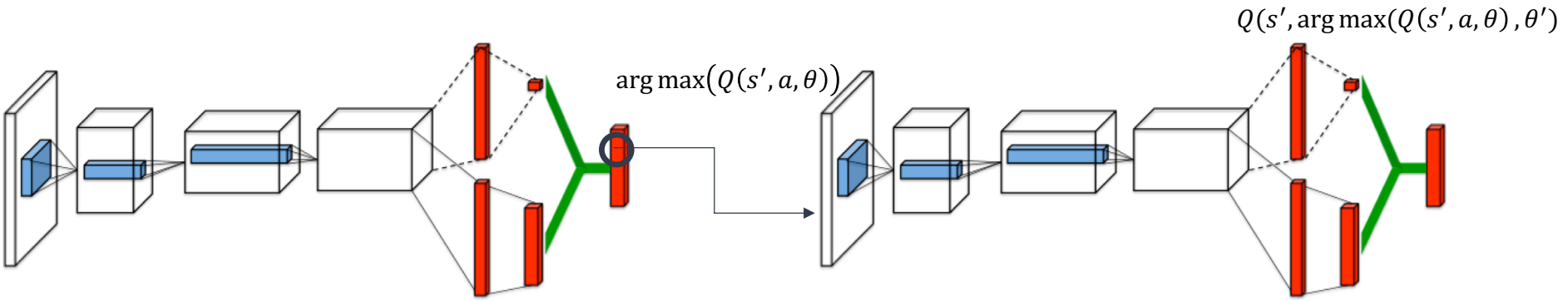
Value와 Advantage



Atari 성능비교, vs Nature, 518, p.529.

Dueling-DDQN

- Double DQN (DDQN)과 Dueling DQN은 각각의 장점을 가지고 있다.
  - 굳이, 하나를 쓸 필요가 있을까?



Where,

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left(A(s, a; \theta, \alpha) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s, a'; \theta, \alpha) \right)$$

Python Code