Dueling Network Architectures for Deep Reinforcement Learning

- **Hado van Hasselt , Arthur Guez,** and **David Silver**

김민철
mclearning2@gmail.com

Summary

- Model-free RL을 위한 neural architecture를 제안
- 두 개의 Estimator
  - Estimator for state value function
  - Estimator for state-dependent action advantage function
- 장점
  - 별다른 수정 없이 (가치 기반, Model-free) 강화학습에 쉽게 적용할 수 있다.

Motivatation

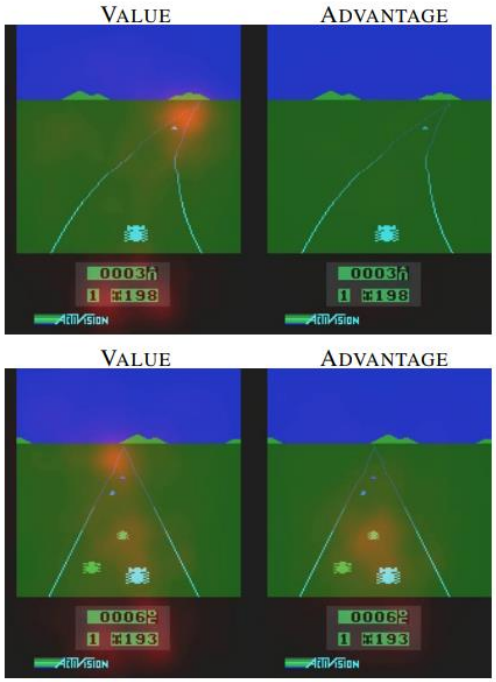
- 기존 방법들은
  - contro이나 RL 자체의 문제를 개선하려 한다.
  - RL과 neural network 를 결합하는 데에만 집중한다.

- model-free RL을 위해 neural network 자체를 개선한다.
- Policy gradient 에서 variance 를 줄이기위해 사용되었던 Advantage 아이디어 활용
- 꼭 모든 상태에서 각 행동의 가치를 판단할 필요가 없다.

  (중요하지 않는 순간들도 많기에)

Motivatation

- 기존 방법들은
  - contro이나 RL 자체의 문제를 개선하려 한다.
  - RL과 neural network 를 결합하는 데에만 집중한다.



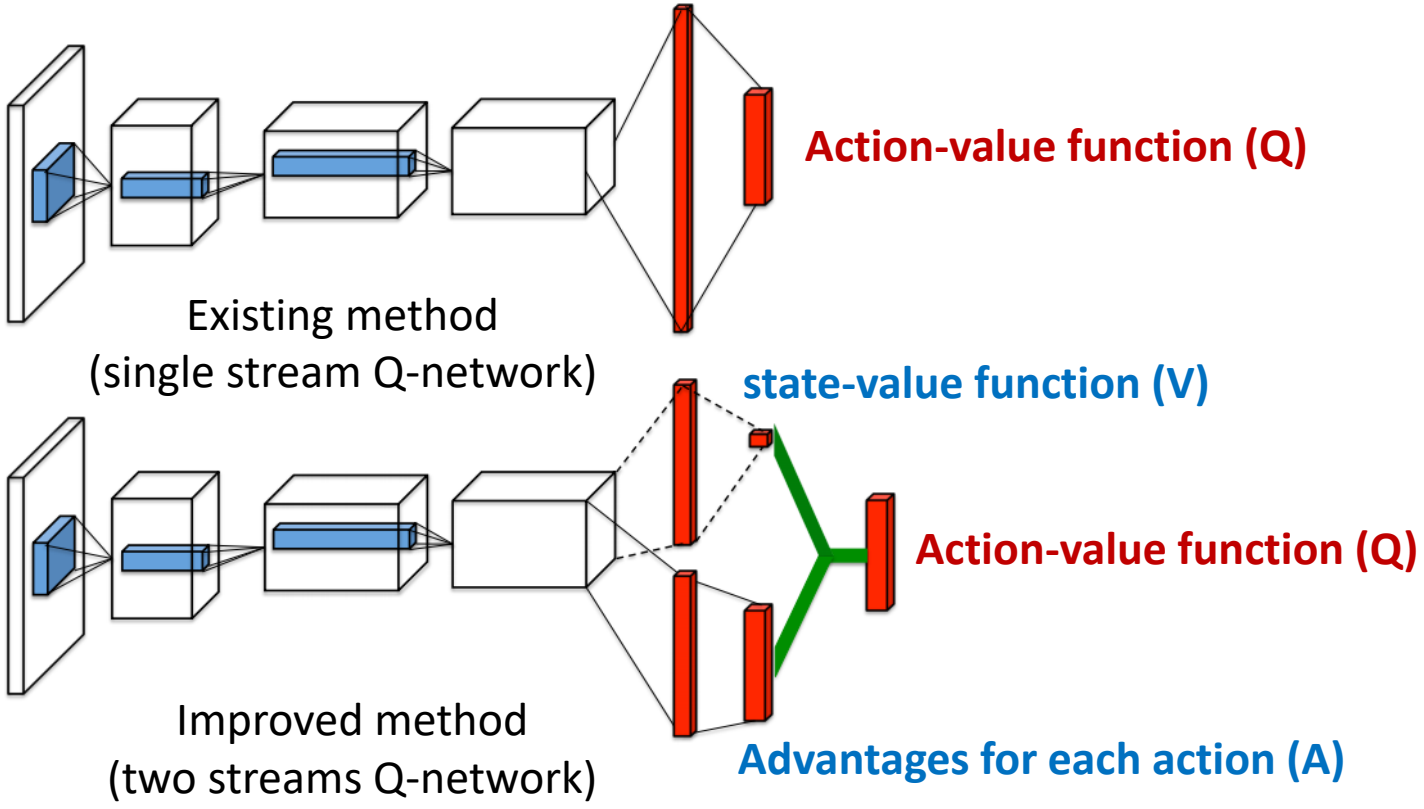
- model-free RL을 위해 neural network 자체를 개선한다.
- Policy gradient 에서 variance 를 줄이기위해

  사용되었던 Advantage 아이디어 활용
- 꼭 모든 상태에서 각 행동의 가치를 판단할 필요가 없다.

  (오른쪽 그림을 보면 Advantage는 장애물이 없다면 굳이 아무런 가치를 따지지

  않는다.)

Main method



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

θ : shared parameters
α : advantages parameters
β : state-value parameters

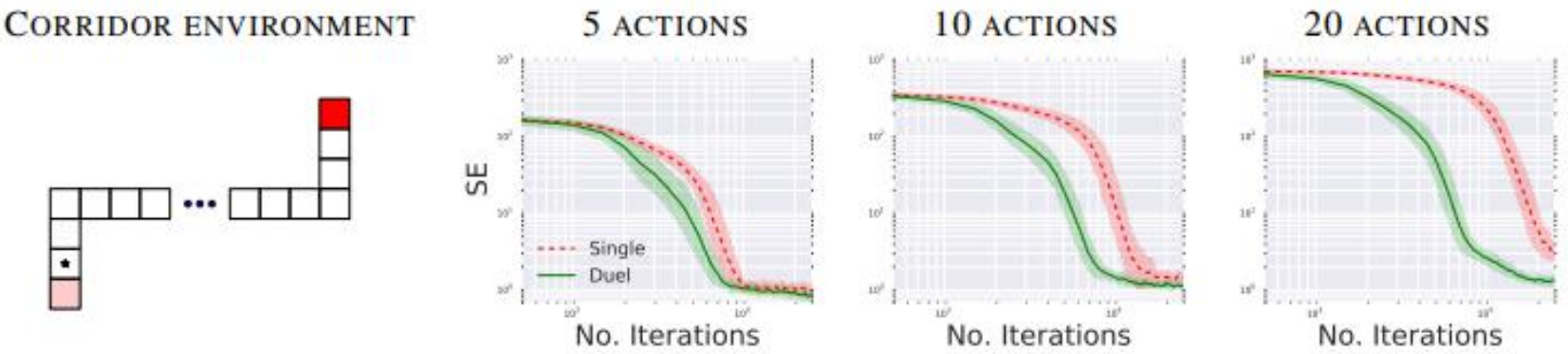
Main method

- V와 A를 더해 Q를 만드는 것이기 때문에 V나 A가 잘못 estimat해도 더했을 때 좋은 결과가 나오면 잘 estimate 했다고 판단할 수도 있다.(The lack of identifiability)
- Advantage function estimator가 선택한 행동에 대해서는 0이 나오도록 하는데 집중
  - Trial 1 $$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \max_{a' \in |\mathcal{A}|} A(s,a';\theta,\alpha) \right)$$
  - Trial 2 $$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s,a';\theta,\alpha) \right)$$ this paper uses this
  - Trial 3 Trial 1의 max를 softmax로(하지만 성능이 비슷)

Result 1

- Corridor Environment
  - * 에서 시작해서 빨간 지점까지 도달해야 한다.
  - SARSA로 실험 $y_i = r + \gamma \mathbb{E}_{a' \sim \pi(s')} \left[Q(s', a'; \theta_i) \right]$
  - 5 actions (right, left, up, down, no-op)
  - 10 actions (right, left, up, down, no-op, no-op, ... , no-op)
  - 20 actions (right, left, up, down, no-op, no-op, ... , no-op)

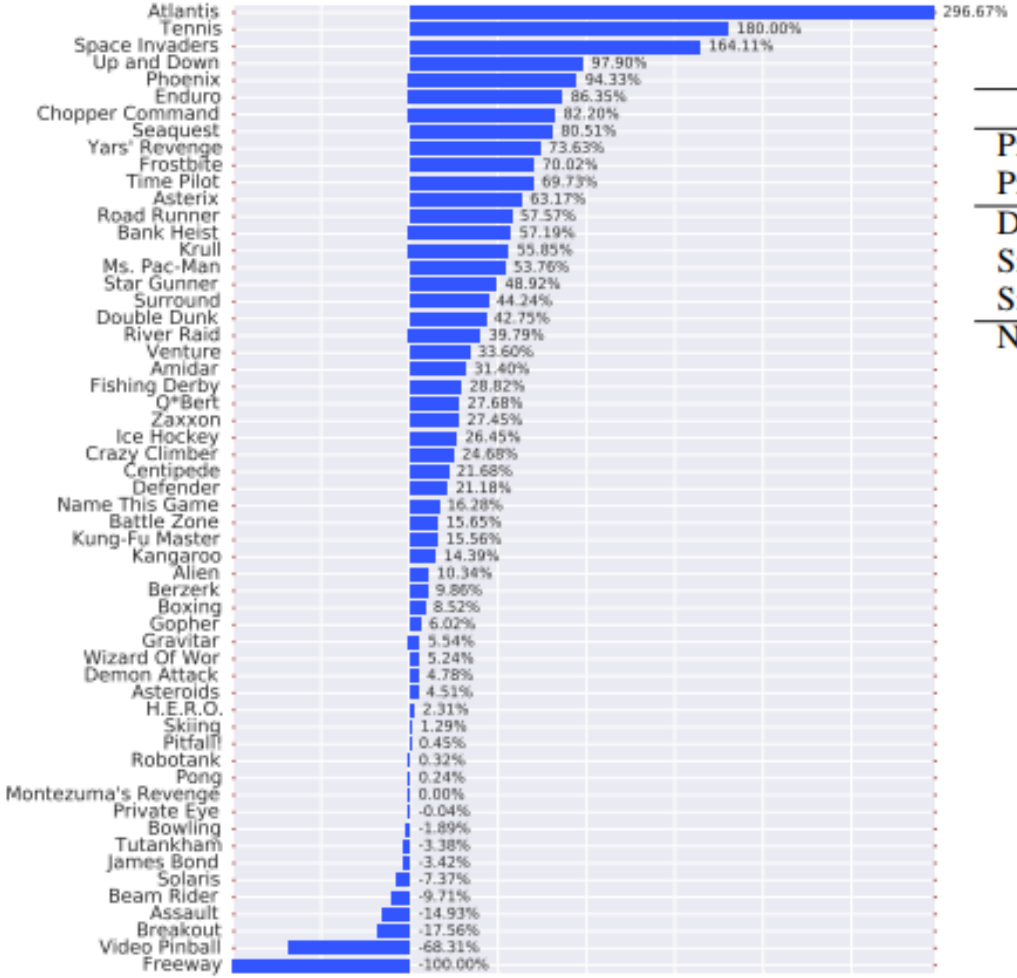
- MSE로 비교환 결과 Dueling이 효과적임을 알 수 있다



Result 2

- Atari Environment
  - DQN과 거의 똑같지만 learning rate를 좀 더 낮게
  - gradient norm clipping (10 이하로)
  - DQN은 fully conntected 1024, Dueling DQN에서는 각각 512, 512 fully connected layer

Result 2

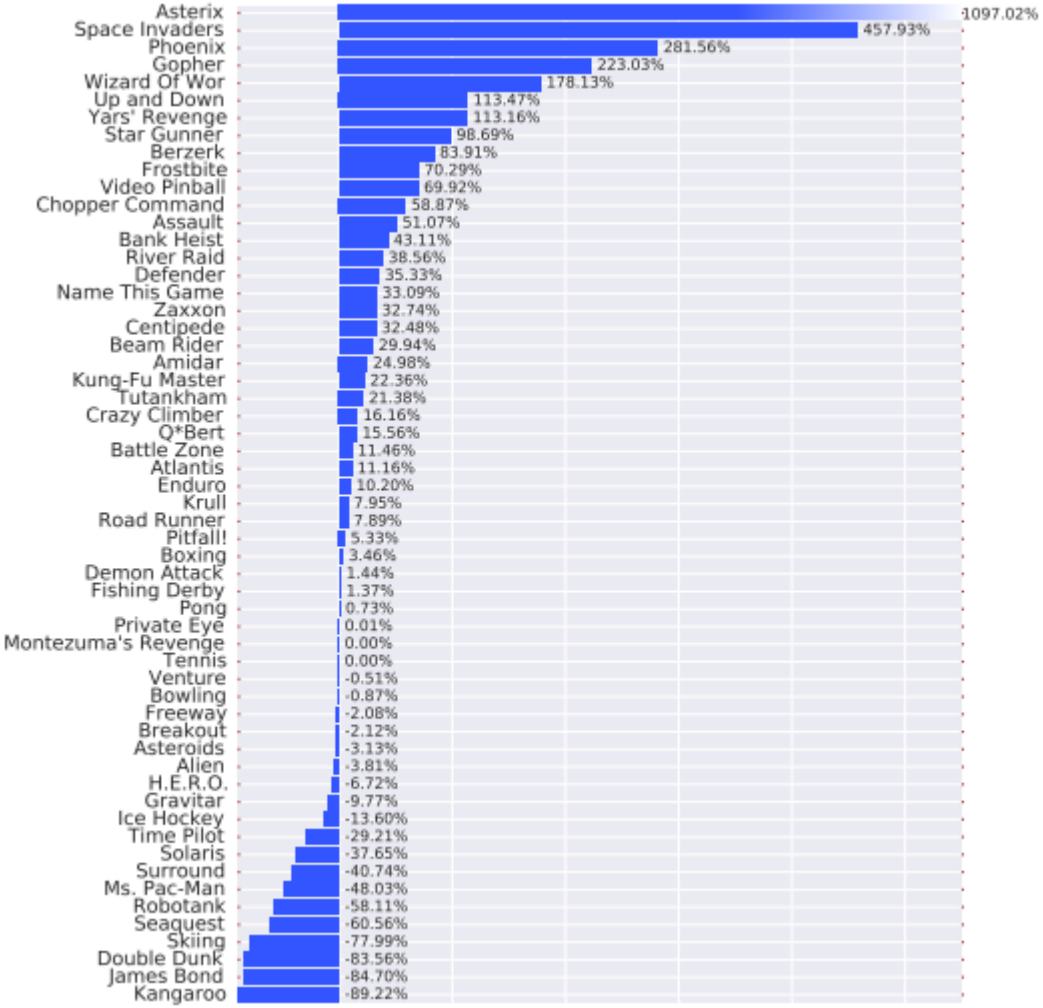


	30 no-ops		Human Starts	
	Mean	**Median**	**Mean**	**Median**
Prior. Duel Clip	**591.9%**	**172.1%**	**567.0%**	**115.3%**
Prior. Single	434.6%	123.7%	386.7%	112.9%
Duel Clip	**373.1%**	**151.5%**	**343.8%**	**117.1%**
Single Clip	341.2%	132.6%	302.8%	114.1%
Single	307.3%	117.8%	332.9%	110.9%
Nature DQN	227.9%	79.1%	219.6%	68.5%

- Dueling Network 적용 결과

$$\frac{\text{Score}_{\text{Agent}} - \text{Score}_{\text{Baseline}}}{\max\{\text{Score}_{\text{Human}}, \text{Score}_{\text{Baseline}}\} - \text{Score}_{\text{Random}}}$$

Result 2



- Prioritized Replay Memory

  적용을 같이 한 결과