2021-2 한양대학교 HAI

강화학습 부트캠프

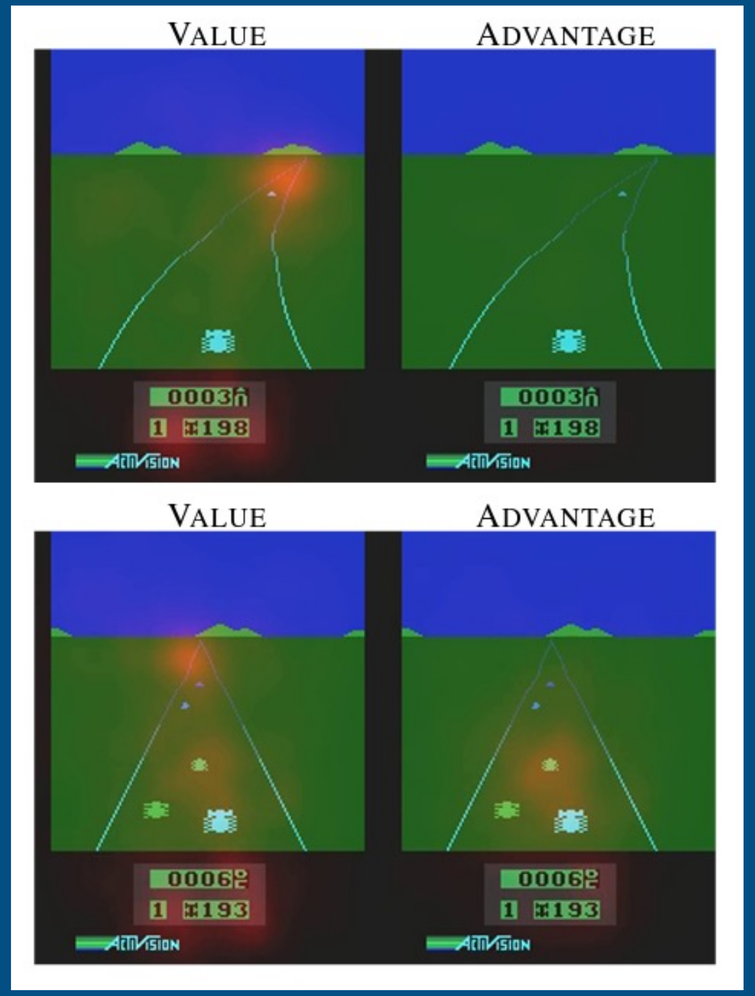
Chris Ohk
utilForever@gmail.com

오늘 배울 내용

- DQN Extensions #2
  - Dueling DQN
  - Categorical DQN
  - Rainbow DQN

Dueling DQN

- Dueling Network Architectures for Deep Reinforcement Learning (Wang, 2015)
- 위 그림 (플레이어 차와 다른 차들이 먼 상황)
  - Value Stream : 앞으로의 보상을 최대화하기 위해
    1) 점수, 2) 멀리 있는 차, 3) 가야할 길에 집중한다.
  - Advantage Stream : 크게 신경쓰지 않는다.
    (당장에 행동을 하지 않아도 어차피 차는 나아가고 점수는 쌓이고 있기 때문)
- 아래 그림 (플레이어 차와 다른 차들이 매우 가까운 상황)
  - Value Stream : **1) 점수, 2) 앞에 있는 차, 3) 가야할 길**에 집중한다.
  - Advantage Stream : 앞에 있는 차에 집중한다.
    (당장에 행동을 취하지 않으면 보상을 얻는 데 영향이 있기 때문)



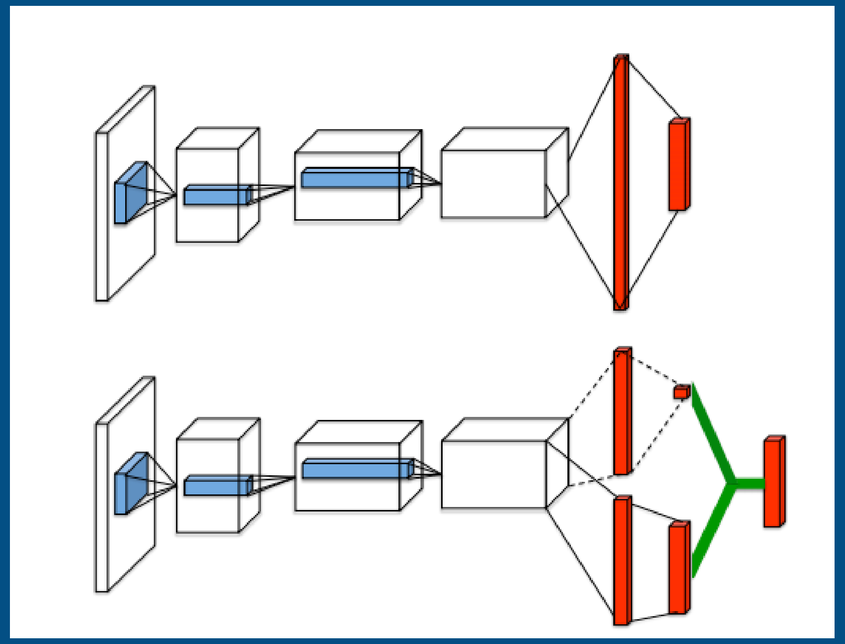
Dueling DQN

- Dueling DQN은 Q 값을 두 값으로 나눈다.

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

  - $V(s)$: 상태 s로부터 얼마나 많은 보상을 받을지 알려주는 함수 (Value)
  - $A(s,a)$: 해당 행동이 다른 행동에 비해 얼마나 더 좋은지를 나타내는 함수 (Advantage)

→ 행동에 대해서 정확한 값을 알 필요 없이, 상태 가치 함수를 배우는 것만으로도 충분하다.



Dueling DQN

- 하지만, Q 값에 V와 A가 얼마나 영향을 줬는지 알 방법이 없다는 문제가 있다.
  예를 들어 $Q = 20$이면 $V + A = 20$인데, 이때 경우의 수는 거의 무한에 가깝다.
- 이 문제를 해결하기 위해, 가장 높은 Q 값($Q(s, a^*)$)을 $V(s)$와 같게 한다.
  즉, Advantage 함수의 최댓값을 0으로 만들고 다른 모든 값을 음수로 만든다.
  이렇게 하면 V 값을 정확하게 알 수 있으므로 문제를 해결할 수 있다.

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

Dueling DQN

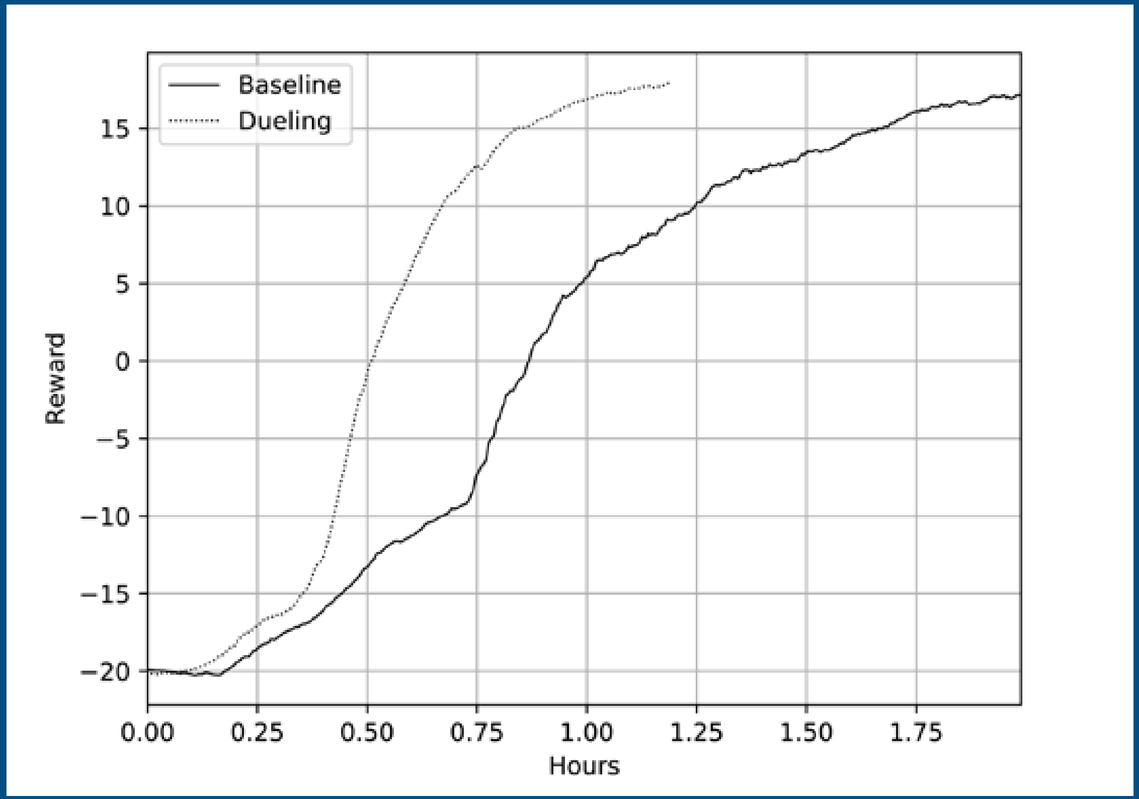
- 논문에서는 max 연산자를 평균으로 대체해서 사용한다.

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

- 원래 수식을 사용하면 V와 A의 의미를 갖게 되지만 평균으로 대체하면 의미를 잃게 된다. 하지만 상수로 고정해 놓은 목표가 아니기 때문에, 최적화의 안정성이 증가하게 된다.
- 실제로 두 수식을 같이 실험하면 유사한 결과가 나온다.

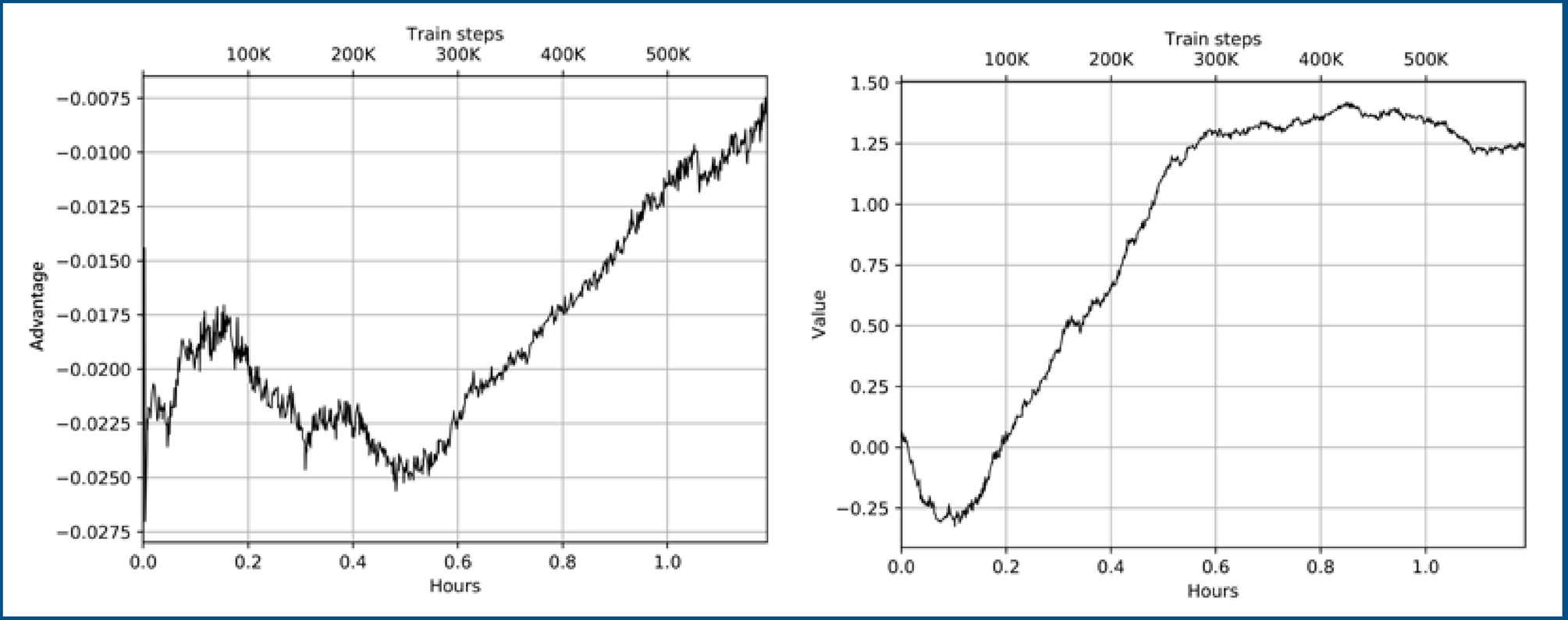
Dueling DQN

- Basic DQN vs Dueling DQN (Reward)



Dueling DQN

- Dueling DQN (Advantage, Value)



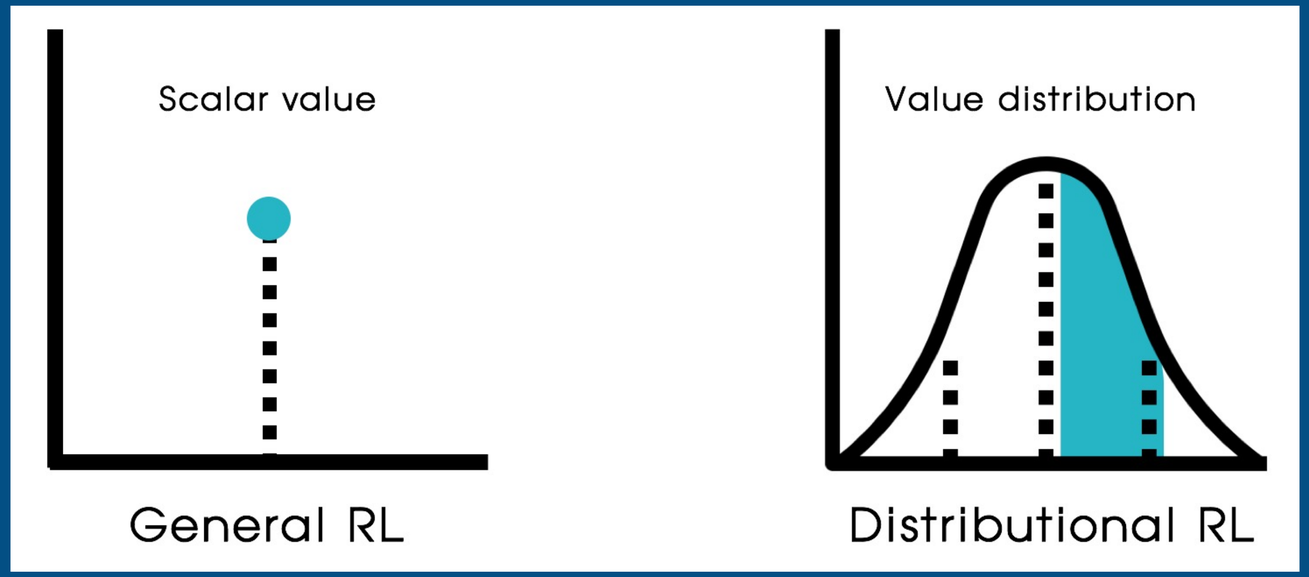
Categorical DQN

- A Distributional Perspective on Reinforcement Learning (Bellemare, 2017)
- General RL : 가치를 하나의 스칼라 값으로 예측

$$Q(x,a) = \mathbb{E}R(x,a) + \gamma \mathbb{E}Q(X',A')$$

- Distributional RL : 가치를 분포로 예측

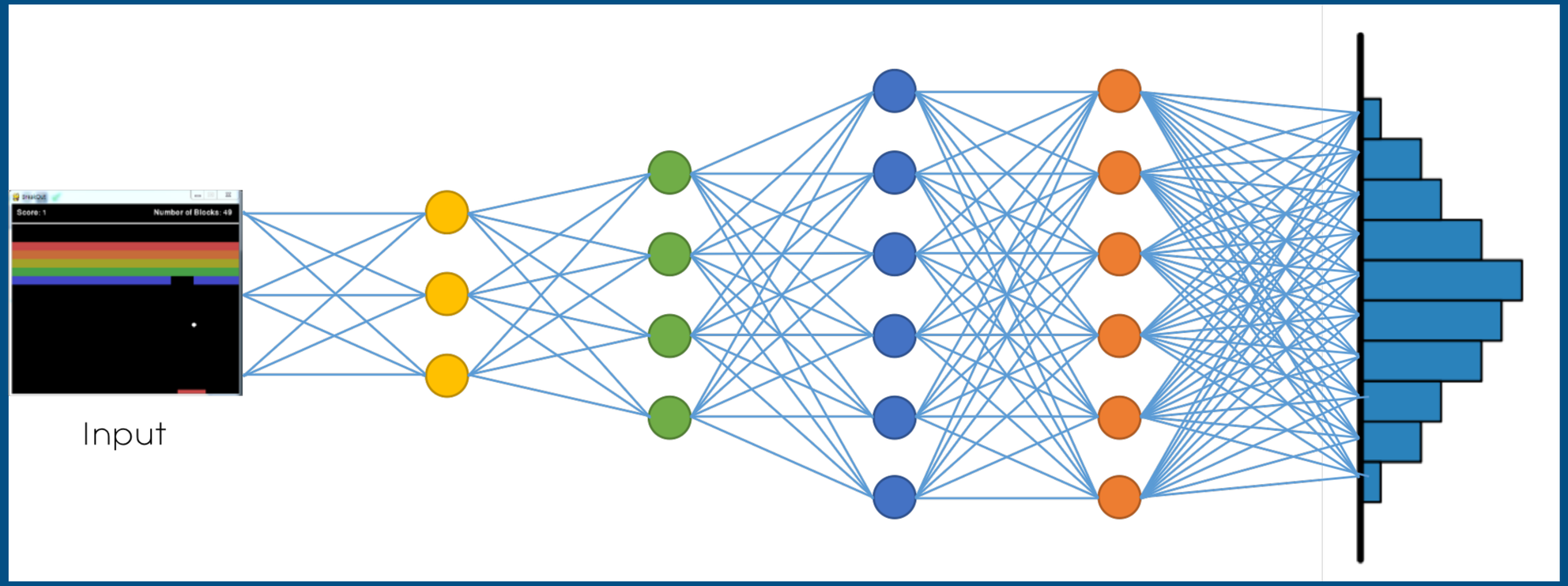
$$Z(x,a) = R(x,a) + \gamma Z(X',A')$$



https://reinforcement-learning-kr.github.io/2018/10/02/C51/

Categorical DQN

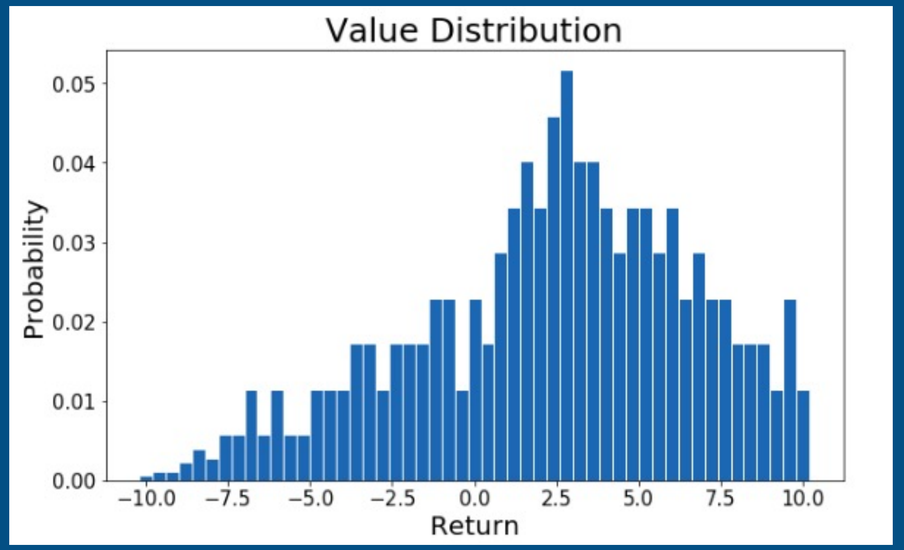
- DQN에서 네트워크의 출력 : 각 행동에 대한 Q 값
- Distributional RL에서 네트워크의 출력 : 각 행동에 대한 가치 분포



https://reinforcement-learning-kr.github.io/2018/10/02/C51/

Categorical DQN

- 각 행동에 대한 가치 분포 = 이산 확률 분포 (Discrete Probability Distribution)
  - 가로축 : Support 또는 Atom (가치 값), 세로축 : 확률 → 각각의 가치와 그 가치를 받을 확률을 나타내는 분포
- 이 분포를 결정하기 위해서 몇 가지 매개 변수가 필요하다.
  - Support의 개수, Support의 최댓값, Support의 최솟값
  - Support 값은 최솟값부터 최댓값까지 Support의 개수에 맞게 일정한 간격으로 나누게 된다.
    즉, 미리 결정된 매개 변수들에 의해 그 값이 정해지게 된다. 신경망은 바로 이 Support들에 대한 확률을 구한다.
    각 행동에 대해서 하나의 분포가 필요하기 때문에 신경망의 출력 크기는 [Support의 개수 * 행동의 개수]가 된다.



Categorical DQN

- Categorical DQN의 알고리즘은 DQN과 유사하다. 차이점은 다음과 같다.
- Q 값 계산 : 이산 확률 분포의 기댓값

$$Q(x_{t+1}, a) = \sum_i z_i p_i(x_{t+1}, a)$$

- Loss 계산 : 타겟 분포와 추정 분포 간의 차이를 줄이는 방향으로 학습
  → 크로스 엔트로피 (Cross Entropy)

$$Loss = -\sum_i m_i \log p_i(x_t, a_t)$$

Categorical DQN

- 타겟 분포 : 우선 Support 값들을 통해 타겟 값을 구한다.

$$\hat{\mathcal{T}} z_j \leftarrow \left[r_t + \gamma_t z_j \right]_{V_{\mathrm{MIN}}}^{V_{\mathrm{MAX}}}$$

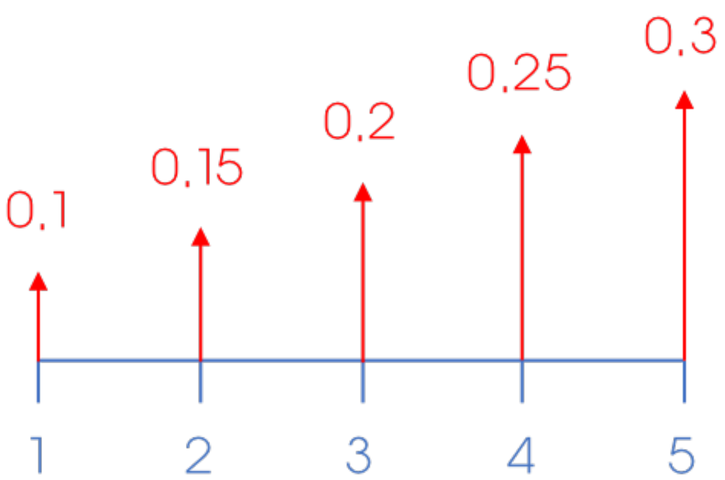
- 각 Support에 감가율을 곱하고 보상을 더해준다. (에피소드가 끝난 경우에는 모든 Support 값들을 보상 값으로 사용)
  그리고 최댓값보다 큰 경우 최댓값과 같도록, 최솟값보다 작은 경우 최솟값과 같도록 설정한다.
- 그런데 이 경우 문제가 생길 수 있다.
  - 예를 들어, Support들이 $[1, 2, 3, 4, 5]$이고 $r_t = 0.1$, $\gamma_t = 0.9$라고 가정하자.
    위 식에 따라 연산을 하면 Support들이 $[1, 1.9, 2.8, 3.7, 4.6]$이 된다.
  - Loss인 크로스 엔트로피 연산을 하기 위해서는 두 분포의 Support들이 일치해야 하는데, 현재는 서로 다르다.
    → 타겟 분포의 Support들을 원래의 Support들과 같이 분배해주는 Projection 과정이 추가로 필요하다.

$$m_l \leftarrow m_l + p_j(x_{t+1}, a^*)\left(u - b_j\right)$$

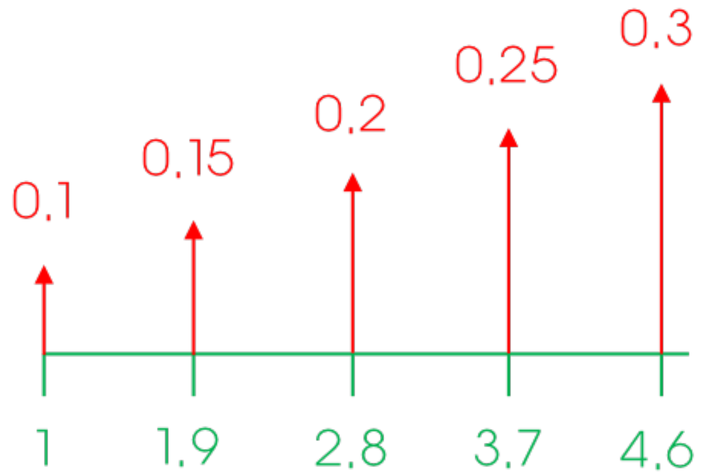
$$m_u \leftarrow m_u + p_j(x_{t+1}, a^*)\left(b_j - l\right)$$

Categorical DQN

- Support: [1, 2, 3, 4, 5]
- Probability: [0.1, 0.15, 0.2, 0.25, 0.3]
- Reward: 0.1
- Discount factor: 0.9

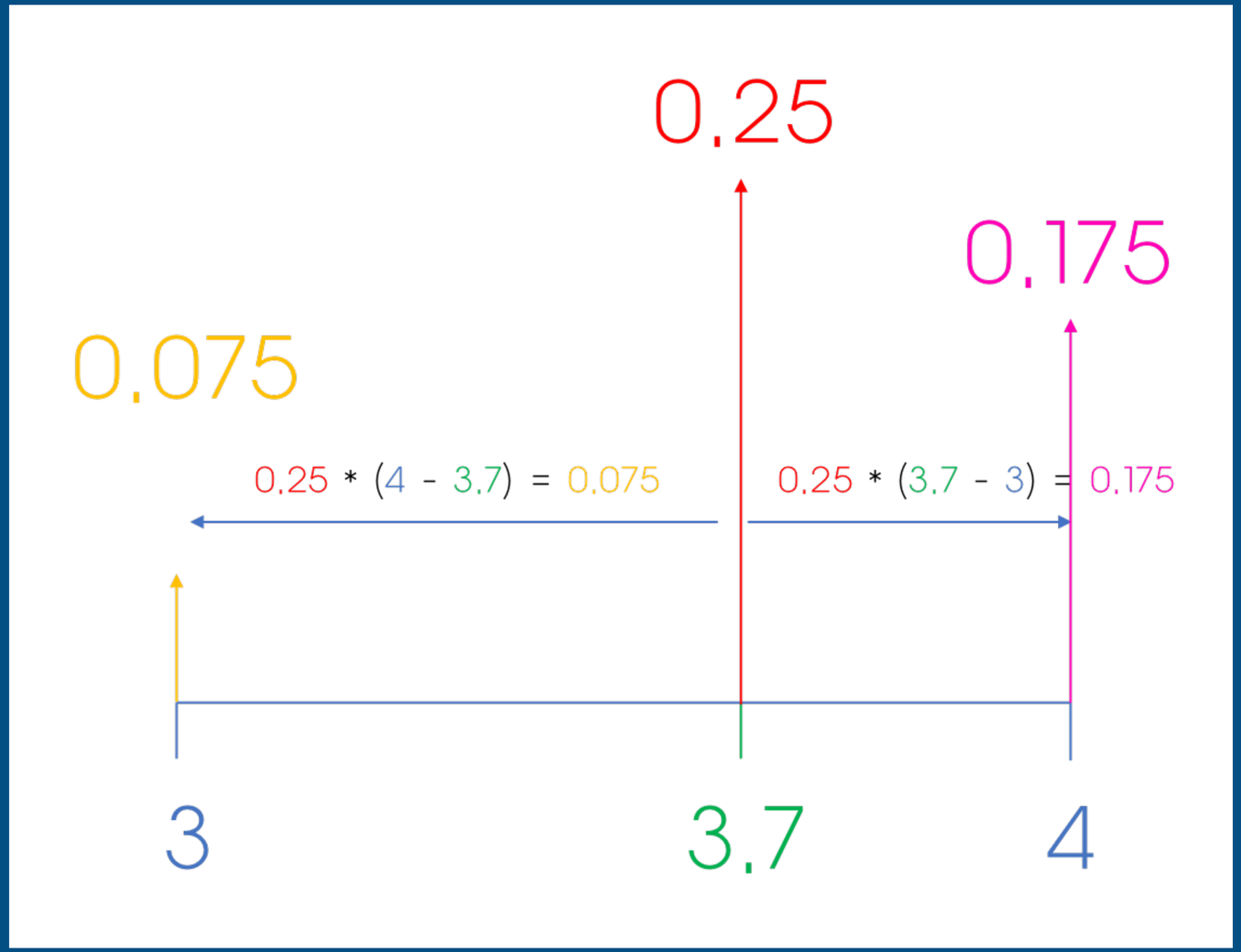


$$\hat{\mathcal{T}} z_i \leftarrow [r_t + \gamma z_i]_{V_{min}}^{V_{max}}$$



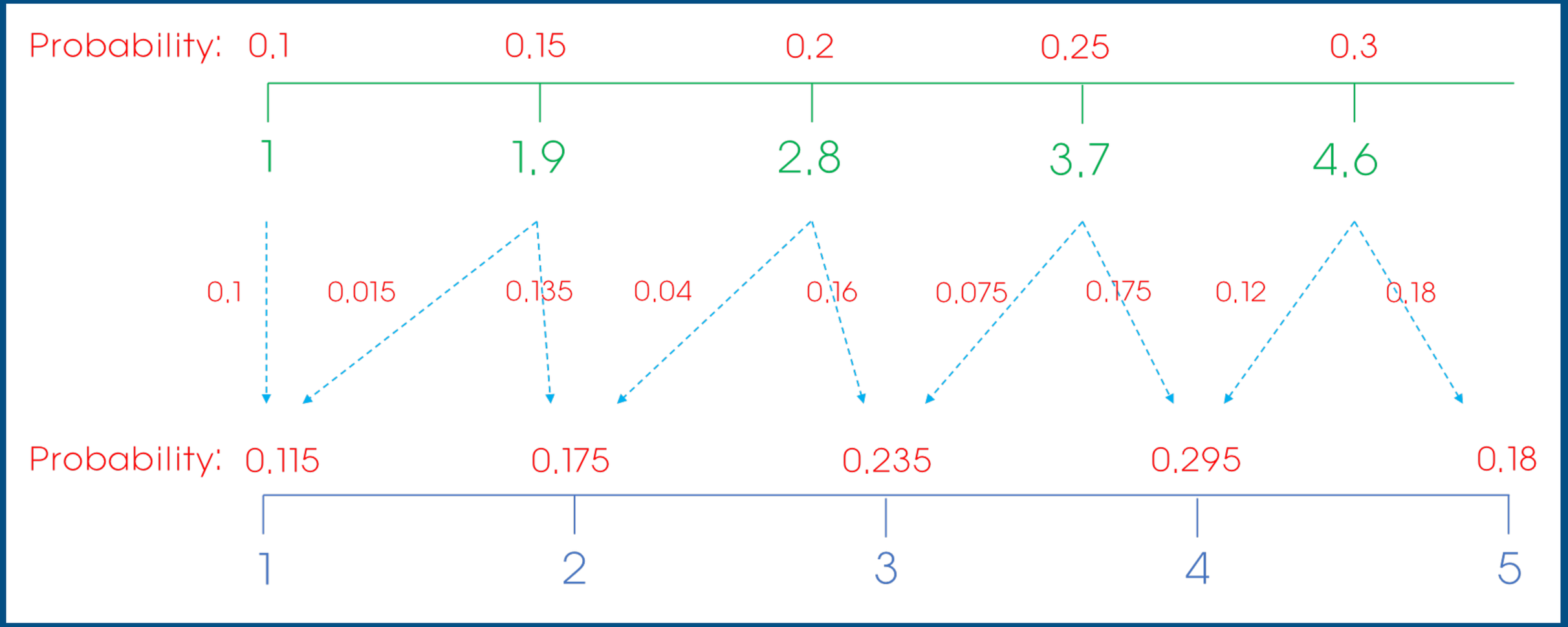
https://reinforcement-learning-kr.github.io/2018/10/02/C51/

Categorical DQN



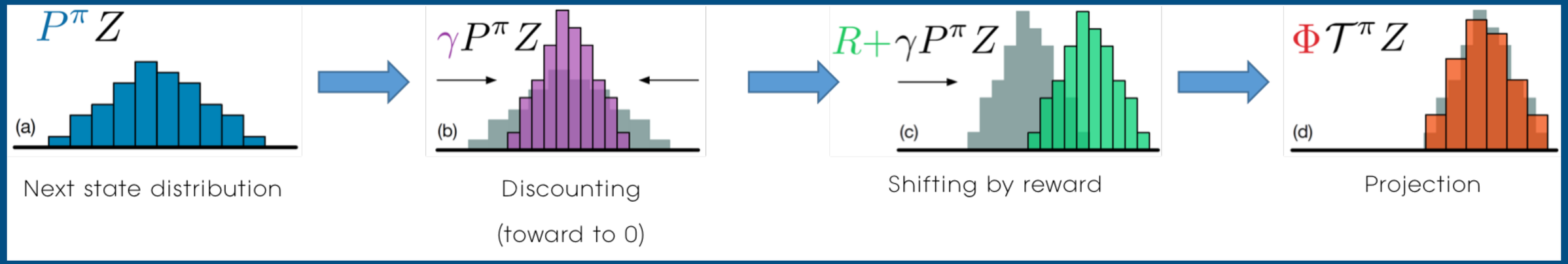
https://reinforcement-learning-kr.github.io/2018/10/02/C51/

Categorical DQN



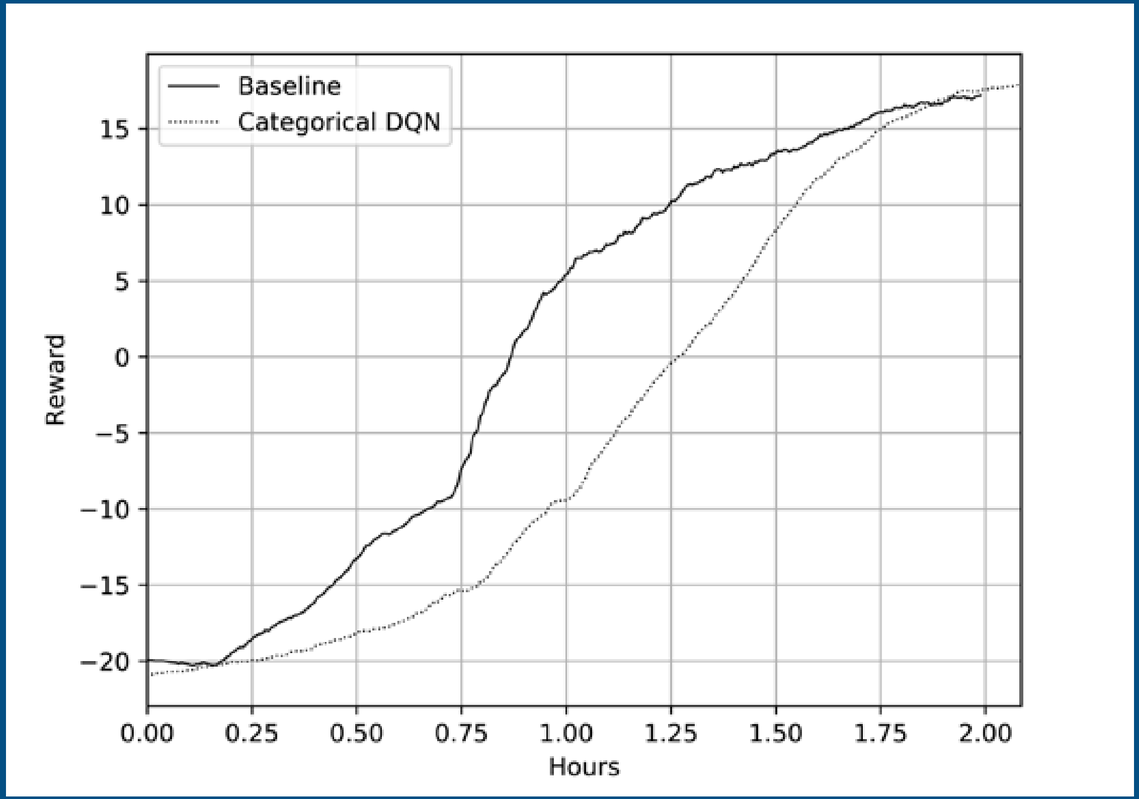
https://reinforcement-learning-kr.github.io/2018/10/02/C51/

Categorical DQN



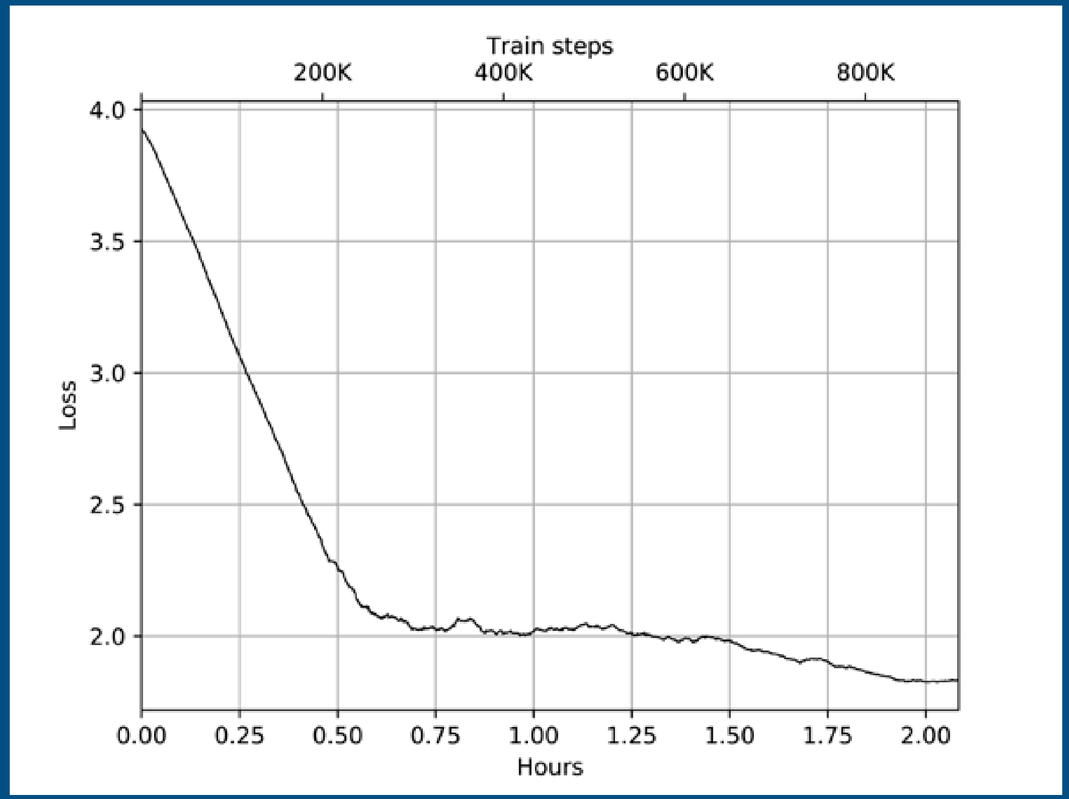
Categorical DQN

- Basic DQN vs Categorical DQN (Reward)



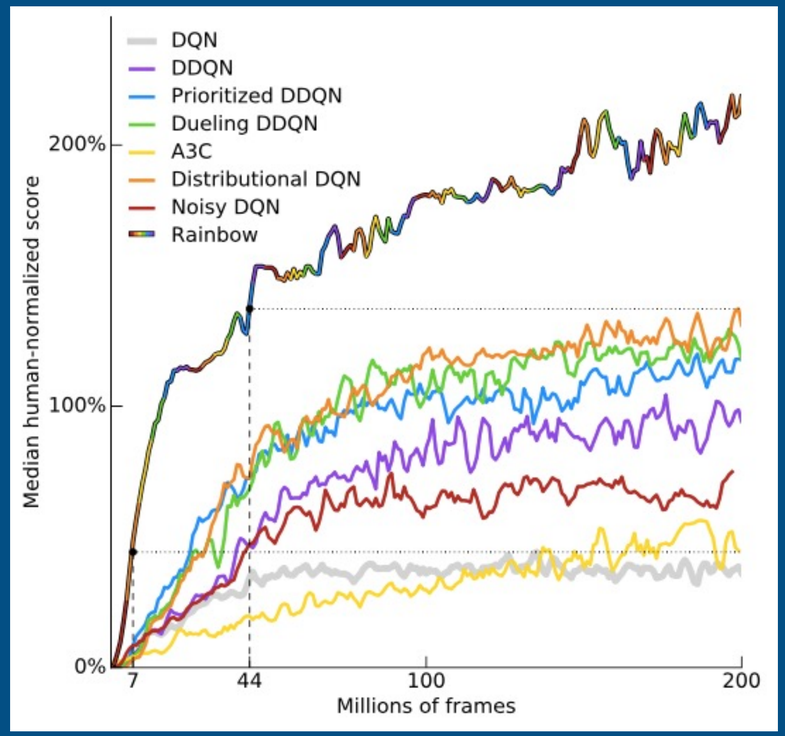
Categorical DQN

- Categorical DQN (Loss)



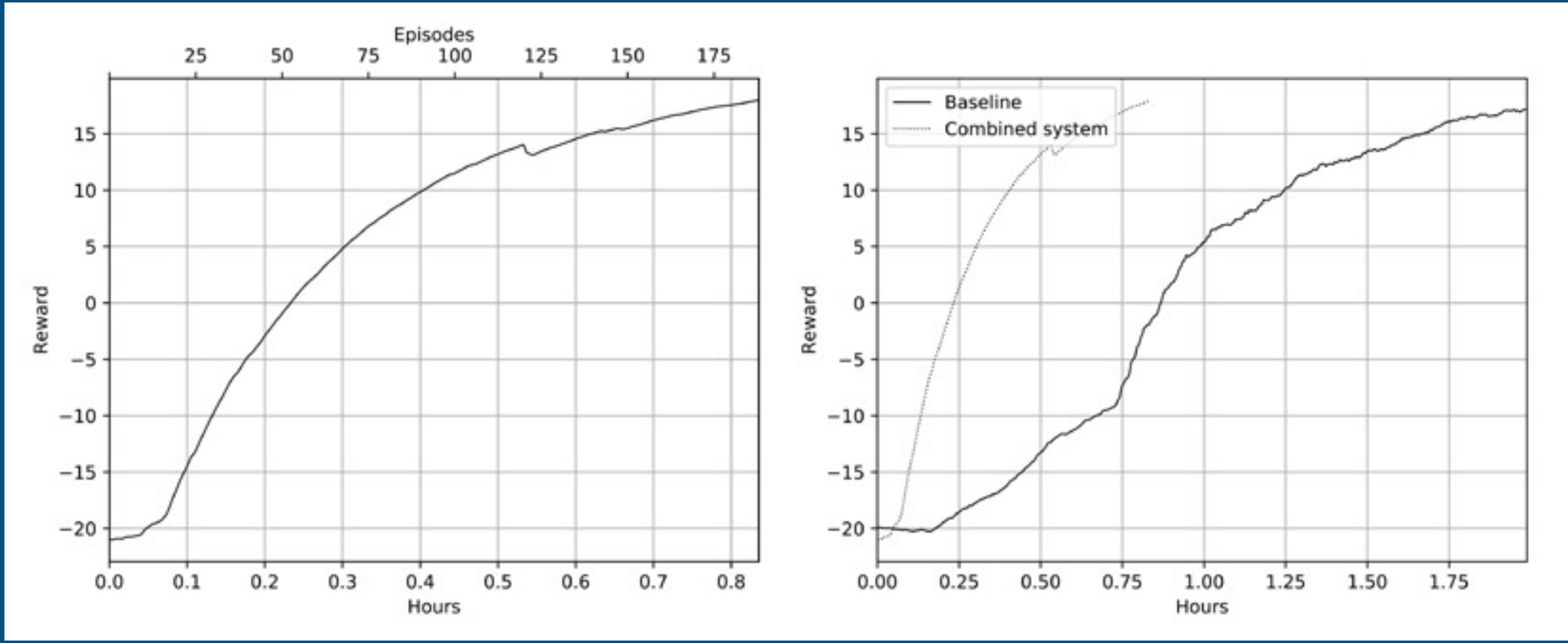
Rainbow DQN

- Rainbow: Combining Improvements in Deep Reinforcement Learning
- Rainbow = DQN + Multi-step DQN + Double DQN (DDQN) + Prioritized Experience Replay (PER) + Dueling DQN + Categorical DQN



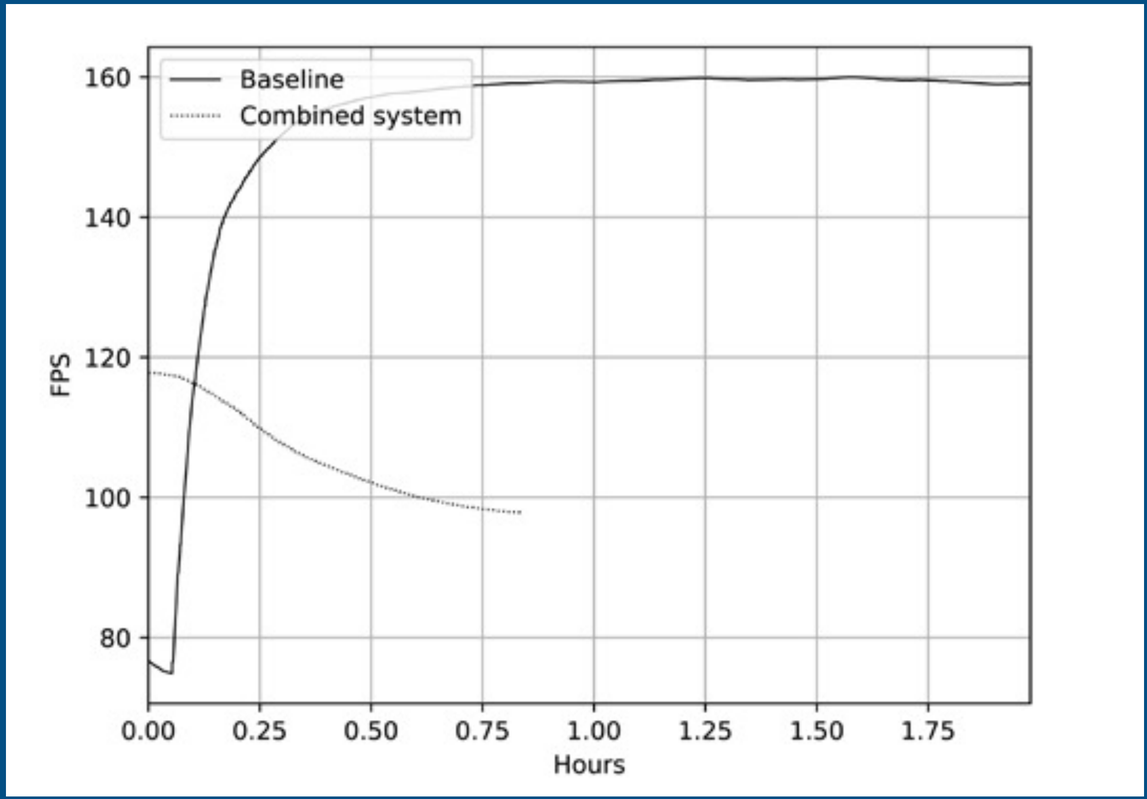
Rainbow DQN

- Basic DQN vs Rainbow DQN (Reward)



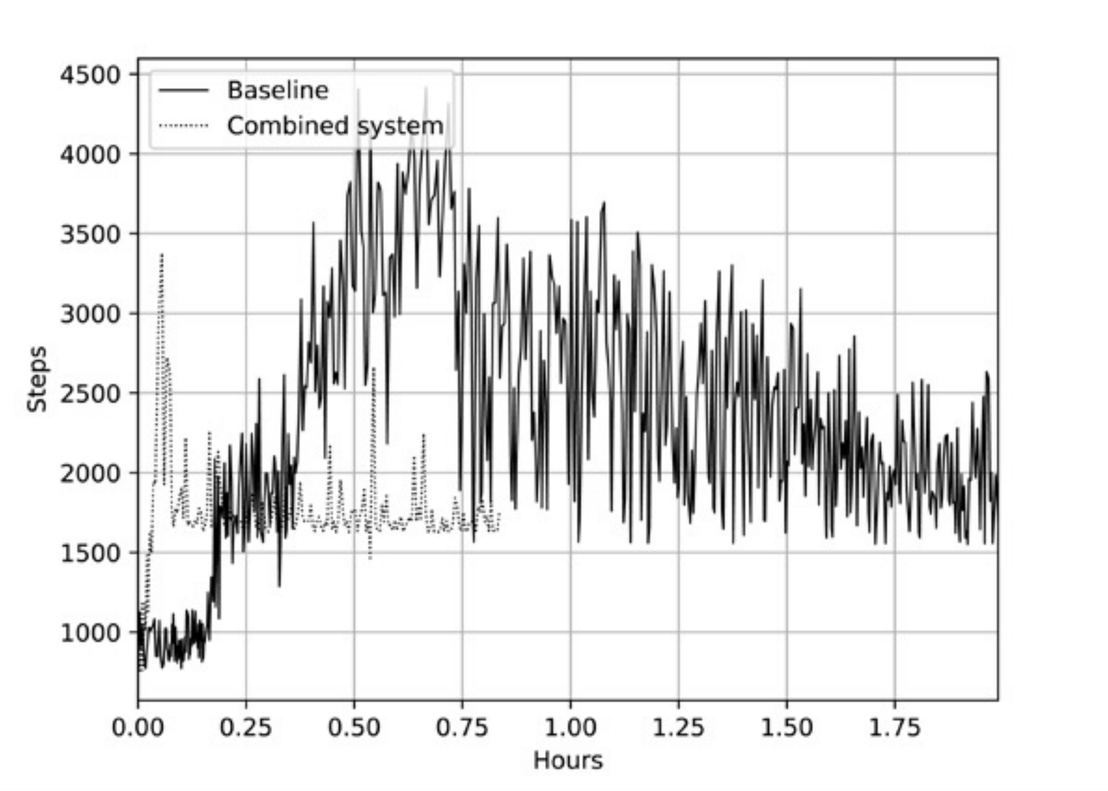
Rainbow DQN

- Basic DQN vs Rainbow DQN (FPS)



Rainbow DQN

- Basic DQN vs Rainbow DQN (Steps)



감사합니다!

스터디 듣느라 고생 많았습니다.