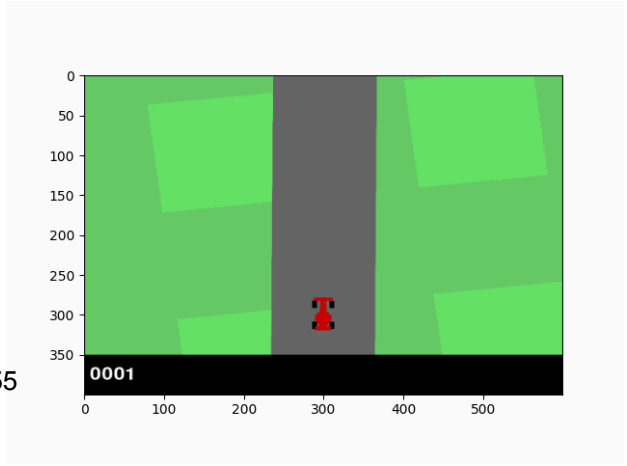
강화학습 과제 2 (Gym CarRacing V2)

action

discrete

5 action

1. do nothing
2. steer left
3. steer right
4. gas
5. brake



observation Shape (96, 96,3): RGB값으로 range 0 ~ 255

termination

1. 모든 타일을 방문했을 경우
2. 차가 밖으로 나갈경우 -100 reward를 받고 종료

reward

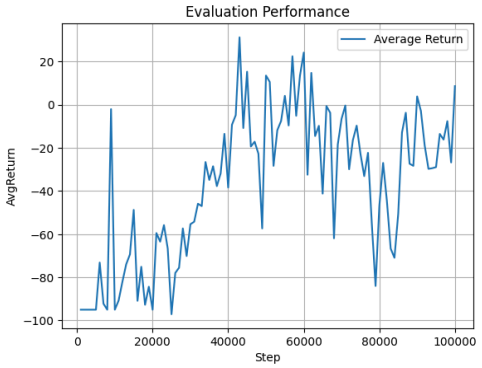
1. frame이 증가할 때마 -0.1씩, N개의 트랙 상의 타일 하나를 방문할 때마다 + 1000/N 점이 보상
   즉 모든 tile을 방문했을 시에 총 보상은은 1000 -0.1*frame을 받는다
2. 차가 track 밖으로 나갔을 경우 -100 reward 받고 종료

실험 내용

2.1 기본 DQN 실행 및 결과 분석 (10^5)

설정한 파라미터

lr: 0.00025
epsilon: 1.0
epsilon_min: 0.1
gamma: 0.99
batch_size: 32
warmup_steps: 5000
buffer_size: 20,000
target_update_interval: 10,000
epsilon_decay: (epsilon - epsilon_min) / 100,000 = 0.000009



결과: episode 20개

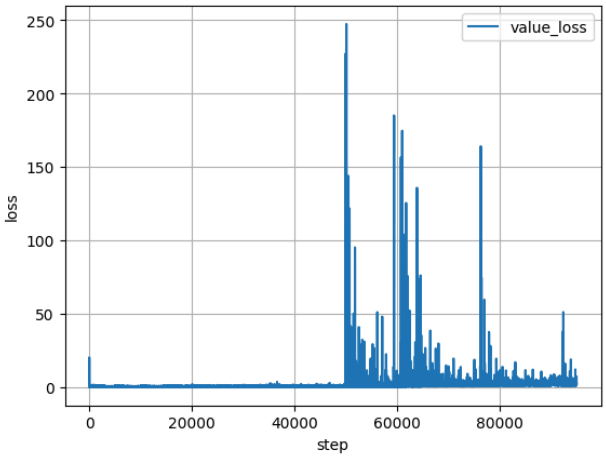
Mean: -29.13, Std: 51.33, Max: 88.10, Min: -115.92

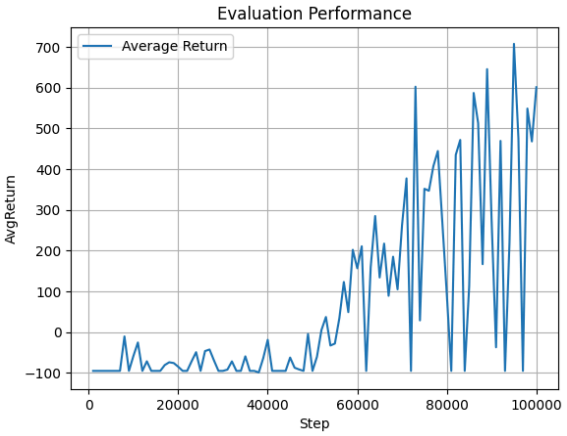
3.2 파라미터 변화 실험

3.2.1 더 복잡한 CNN 구조 + target_interval과 batchsize의 변화

설정한 파라미터

batch_size: 32 -> 64
target_update_interval: 10000 -> 5000
CNN 구조
1. Conv2D : 16 filters, 3×3, stride 2, padding 1 + BatchNorm + ReLU → N × 16 × 42 × 42
2. AvgPool: 2×2, stride 2
3. Conv2D : 32 filters, 3×3, stride 1, padding 1 + BatchNorm + ReLU → N × 32 × 21 × 21
4. AvgPool: 2×2, stride 2
5. Conv2D : 64 filters, 3×3, stride 1, padding 1 + BatchNorm + ReLU → N × 64 × 10 × 10
6. AvgPool: 2×2, stride 2



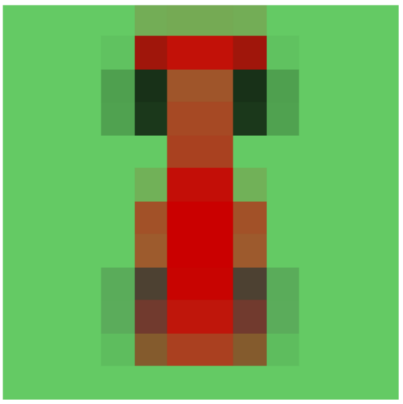


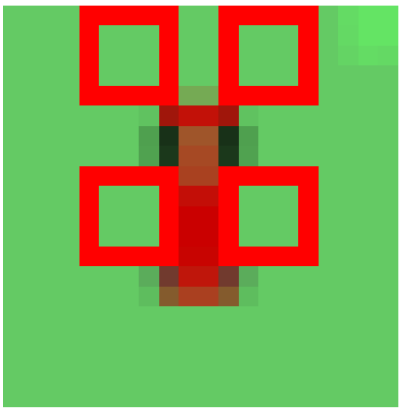
*50000번째에 갑작스런 loss 증가에 대해 5000번 마다 target이 일단 update 되고, 새로운 경로를 탐색한 것이 쌓여 replay buffer에서 꺼내져 큰 loss가 발생할 가능성이 있다. 보통 loss가 증가하는 부분은 주기가 5000번임

결과: episode(20개)
Mean: 182.93, Std:288.38, Max:738.33 Min: -95.00

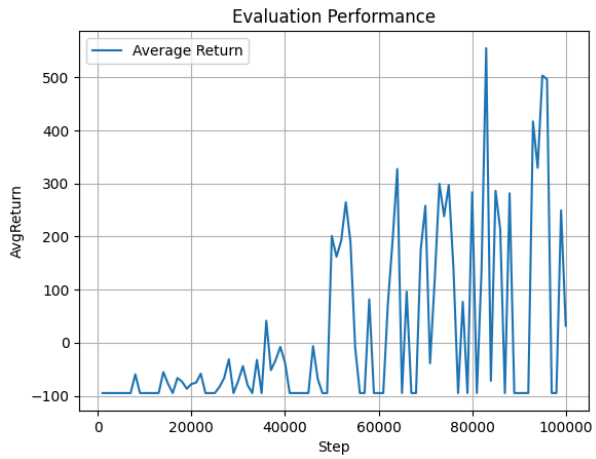
3.2.2 reward 방식 고려

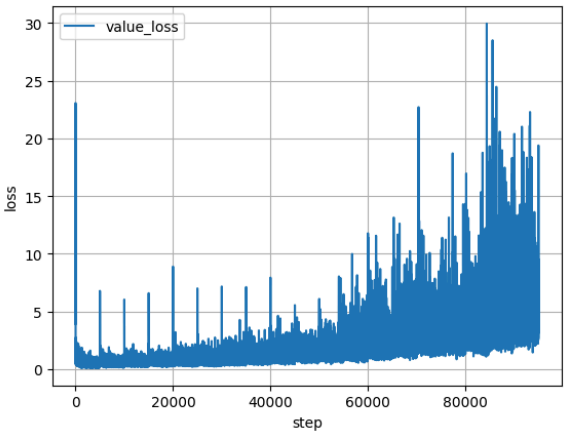
CNN구조로 도로와 잔디를 구분할 가능성이 매우 높지만, 잔디에 빠질때에 페널티나 도로에 있으면 보상이 없는 경우가 reward엔 반영이 되지 않아 train시 reward를 반영하도록 변경, DQN구조는 같음음





게임에서 자동차의 좌표는 고정되어 있으므로 그 주변 지형에 대해 보상을 결정할 수 있을 거라 판단
총 4가지 구역에 빨간색 부분에 모두 도로이면 (색의 차이가 없으면) 칸 수 x (-penalty), 4칸이 모두 잔디일 경우 8*(penalty)으로 reward를 바꿈, 평가는 이러한 reward 환경이 아닌 것으로 평가

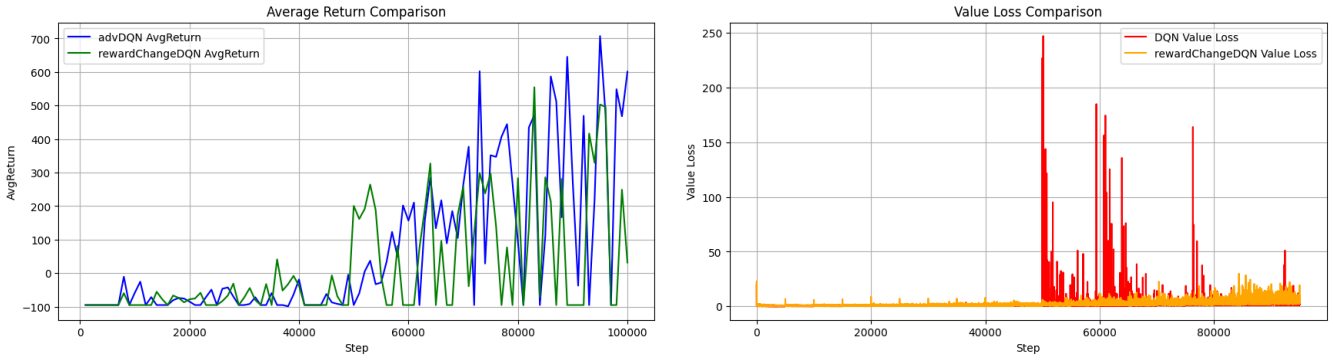




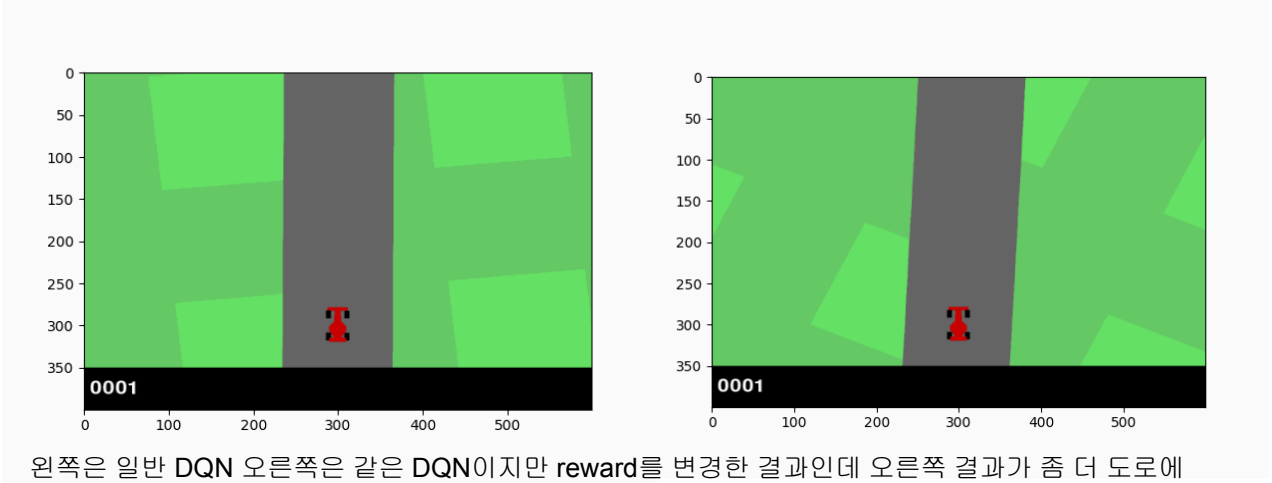
실험 결과 episode(20)

Mean: 494.50, Std: 287.80 Max: 860.56, Min: -95.00

비교



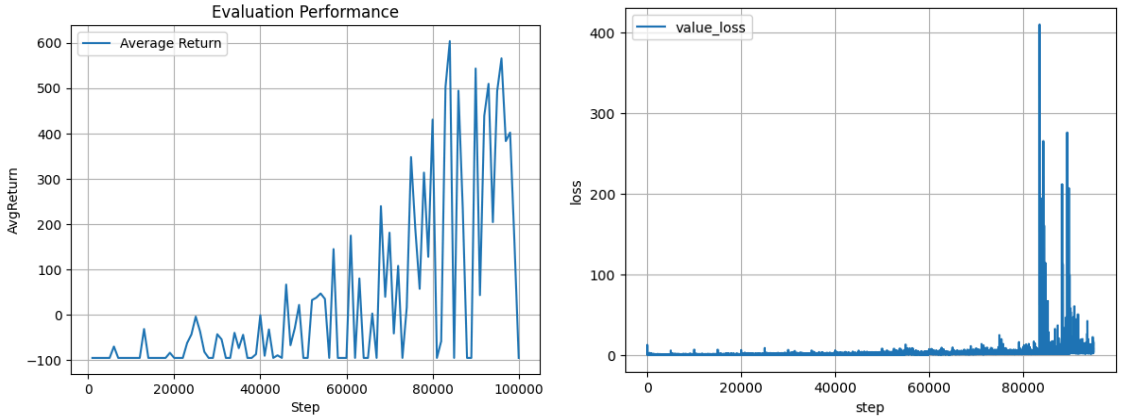
둘을 비교하였을 때 1000번의 episode(3)개의 평가에서는 리워드를 변경하지 않은 것이 더 return이 높아보이며 Loss는 리워드를 변경한 것이 훨씬 더 적어 안정적인 수렴이 가능해보인다. 학습의 경우 평균은 300이 더 높지만 max는 더 낮다. 그러나 min값은 같다.



왼쪽은 일반 DQN 오른쪽은 같은 DQN이지만 reward를 변경한 결과인데 오른쪽 결과가 좀 더 도로에 있으려고 하는 경향이 있다. 다만 급격한 방향 변경을 한다.

3.2.2 DDQN(double DQN) + 수정된 Reward

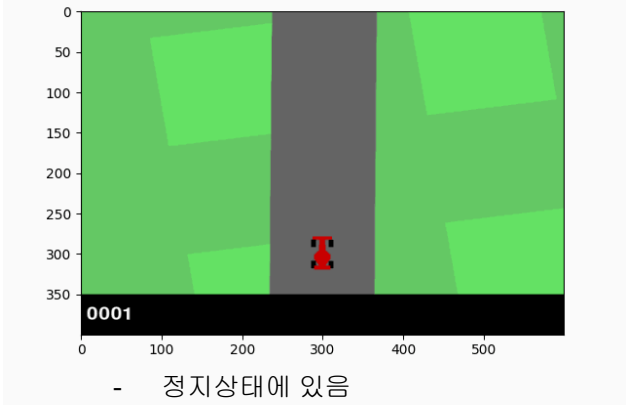
DDQN 방식을 도입하여 얻은 max Action을 target에 넣은 값으로 error를 설정함



학습 마지막에서 확 꺽이며, loss도 폭발적으로 증가한다.그래프를 보면 600을 찍었을 때 갑자기 loss가 증가한 것으로 보아 다른 경로를 찾은 데이터 들이 replay에서 꺼내져서 그랬을 가능성이 있다.

실험 결과

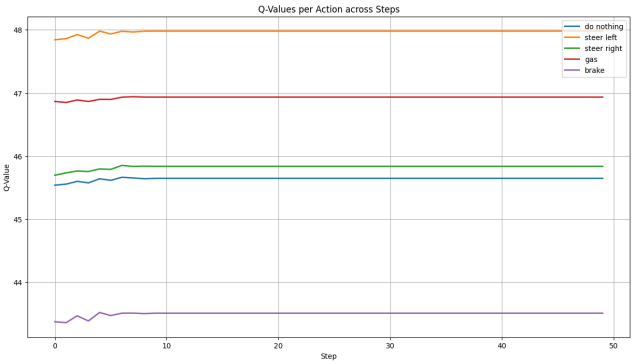
Mean:-22.0, Std: 234.77, Max: 901.40, Min: -95.00
max값은 제일 큰 값이지만 Min값(95)이 제일 많았음



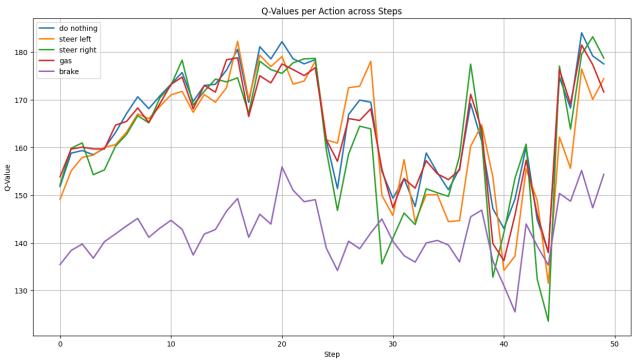
- 정지상태에 있음

Q(s,a)값의 분포

- DDQN의 0에 수렴한 값



- DQN의 일반적인 값



처음 시작에 Q(s,a)값이 가장 높은 action이 turn left로 오른쪽의 경우 gas이다, 즉 action을 하더라도 처음에 gas로 시작하지 않다면, 화면이 고정되어 state가 잘 변하지 않게 된다. 20개 중 2개의 값만 900, 270을 기록한 것으로 보아 입실론의 확률로 이 처음 state를 gas로 탈출하지 않았을까 생각되어진다.

Algorithm	Mean Return	Std	Max Return	Min Return
advDDQN	-22.00	234.77	901.40	-95.00
advDQN (Reward Change)	494.50	287.80	860.56	-95.00
advDQN	182.93	288.38	738.33	-95.00
DQN(Normal)	-29.13	51.33	88.10	-115.92

*각 episode 20번

4. 결론

실험 결과를 통해 Q값과 return 등 여러 관찰을 할 수 있었다. 먼저, 수렴성 판단은 매우 어려웠다. 학습 스텝 수의 제한으로 인해 충분한 학습이 이루어지지 않았을 가능성이 있으며, 특히 표준편차가 매우 크게 나타난 점은 에이전트가 안정적인 정책을 학습하지 못했을 가능성이 있으며. 이는 모델이 수렴하는 과정에 있지 않았을 가능성이 높음. 초기 모델은 표준편차가 50으로 수렴했다고 볼 수 있지만 이후 실험에서 관측된 return 값의 불안정성과 높은 변동성은 여전히 학습이 진행 중임을 나타내며, 정책이 환경 전반에 대해 일반화되지 않았음을 보여줌.

평가 과정에서도 문제가 확인되었다. 멀티스레드가 아닌 방식으로 평가를 수행함에 따라, step 수 대비 평가 과정에서 소요되는 시간이 과도하게 길어졌으며, 이는 전체 학습 프로세스의 효율성을 저하시킬 수 있는 요소로 작용함. 따라서 평가 프로세스의 최적화가 필요해 보임.

DDQN의 경우 성능이 특정 시점 이후 급격히 저하되는 현상이 발생하였다.. 이는 Replay Buffer 내에 초기 학습 단계에서 수집된 초기 transition들이 과도하게 저장되고 학습에 사용되었기 때문으로 해석된다. 이후 환경의 다양성이 증가하며 새로운 데이터가 들어갔고, 이는 기존에 편향된 모델이 적절히 학습하지 못함으로써 성능 저하로 이어졌을 가능성이 크다.

이러한 문제점을 개선하기 위한 방안으로 Prioritized Experience Replay 기법이 필요해 보인다. 이를 통해 중요도가 높은 transition들을 우선적으로 학습에 반영함으로써, 에이전트가 더욱 효율적이고 안정적으로 정책을 개선할 수 있을 것으로 보인다.

https://github.com/kosy990000/RL/tree/main/CarRacingV2