Tobig' Signal

강화학습을 이용한 교차로 신호등 제어

임채빈 한재연 강민성 윤기오 김태한

목차

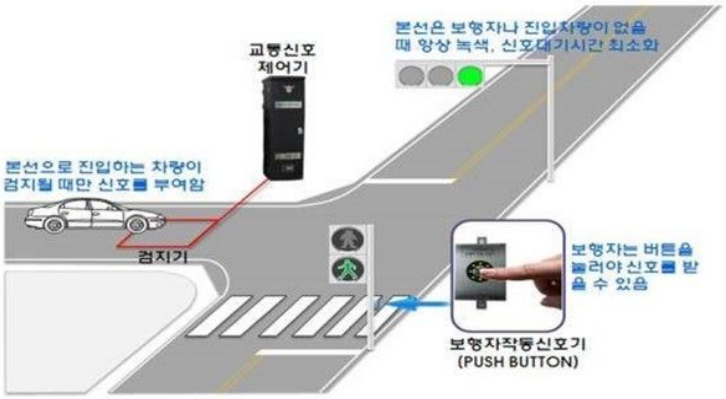

실시간 교통상황을 교통신호에 반영하여
교통흐름을 원활히 하기 위한 프로젝트

1 INTRODUCTION
문제 진단

‘지능형 교통신호시스템’이란?

교차로에서 주도로와 부도로의 차량 흐름을 감지해 부도로에 대기 차량이 없는 경우,
항상 주도로에 직진 신호**를 부여함**으로써 원활한 차량흐름을 유도하는 시스템입니다.

즉, 도로에 설치된 감지기를 통해 신호등 스스로 제어하는 것!
차량이 감지되면 신호가 바뀝니다.



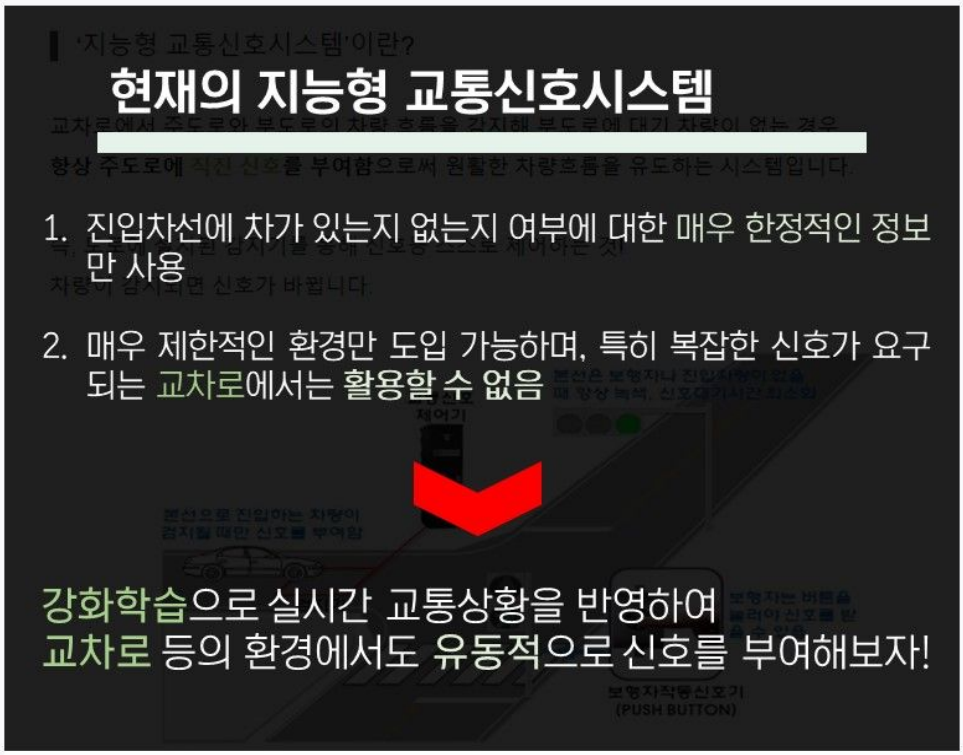
1 INTRODUCTION
문제 진단

'지능형 교통신호시스템'이란?

현재의 지능형 교통신호시스템

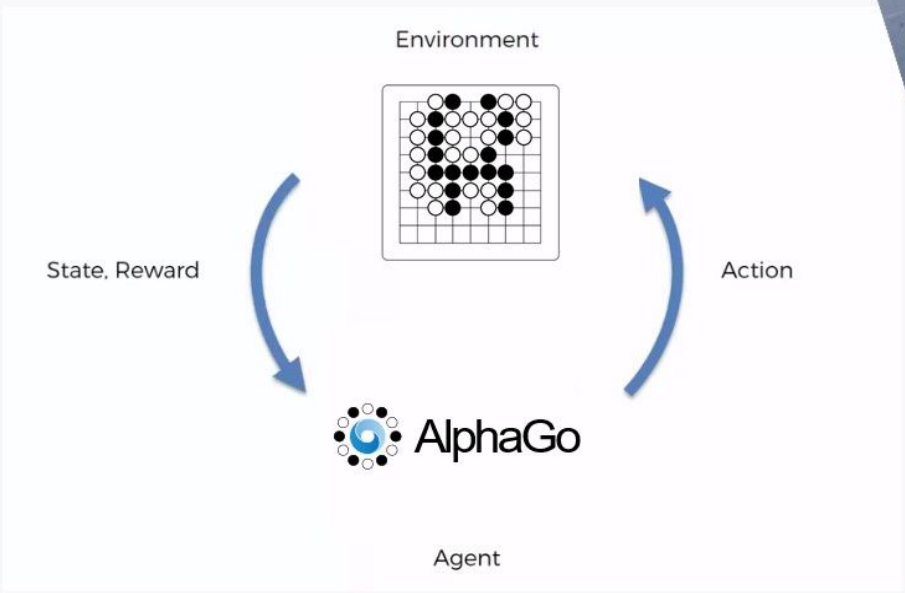
1. 진입차선에 차가 있는지 없는지 여부에 대한 매우 한정적인 정보만 사용
2. 매우 제한적인 환경만 도입 가능하며, 특히 복잡한 신호가 요구되는 교차로에서는 활용할 수 없음

1 INTRODUCTION
문제 진단



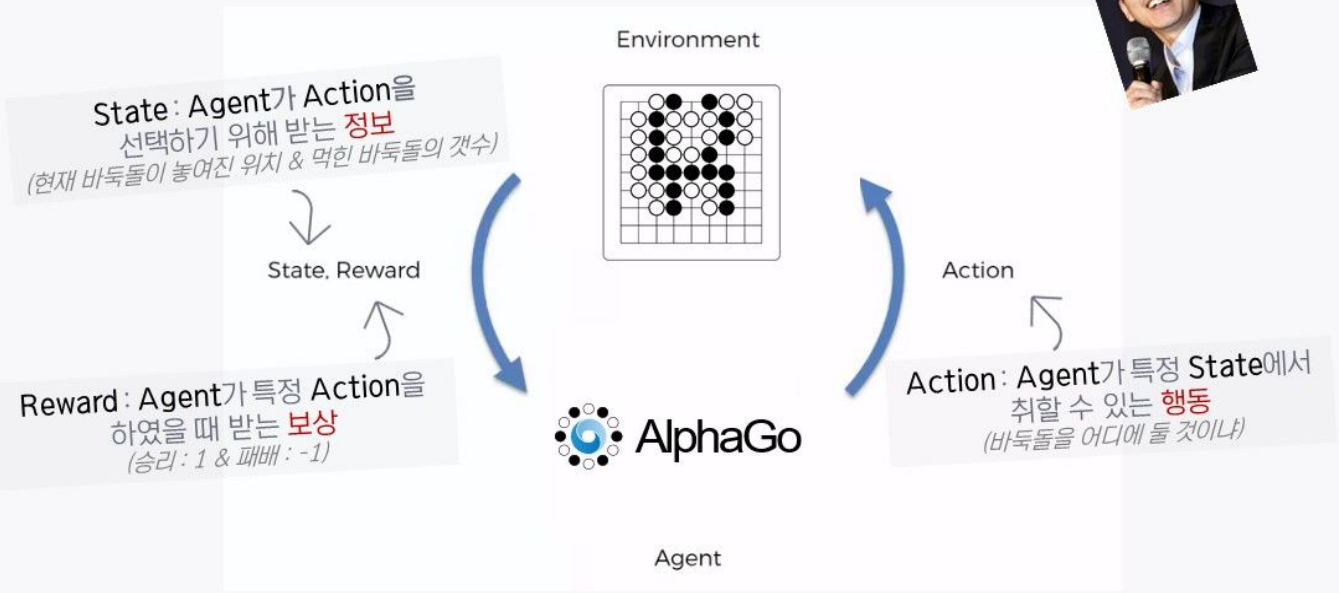
2 Reinforcement Learning
강화학습

2 Reinforcement Learning
강화학습



에이전트가 특정 환경과 상호작용하여, 선택 가능한 행동들 중 보상을 최대화하는 행동 또는 행동 순서를 학습

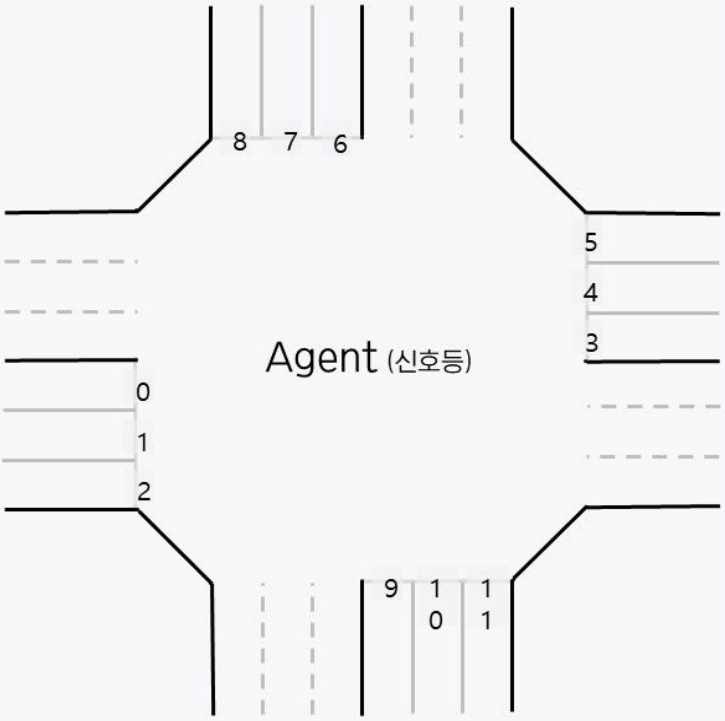
2 Reinforcement Learning
강화학습



에이전트가 특정 환경과 상호작용하여, 선택 가능한 행동들 중 보상을 최대화하는 행동 또는 행동 순서를 학습

3 Environment Setting & RL Algorithm
환경 구성 & 강화학습 알고리즘

3 Environment Setting
환경 구성 - Environment



Environment

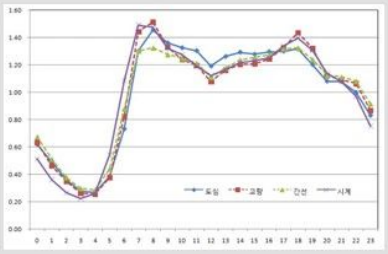
각 좌회전, 직진, 우회전을 담당하는 3차선 교차로 환경 구성

종료 조건

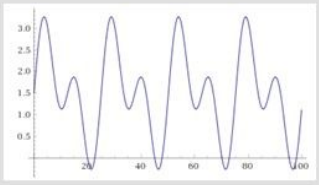
1. 특정 스텝에 도달할 경우 (200 steps)
2. 특정 차선의 차가 정의한 한계를 넘어갈 경우

차량 분포

실제 차량 분포를 모방한 분포를 따라 무작위로 생성

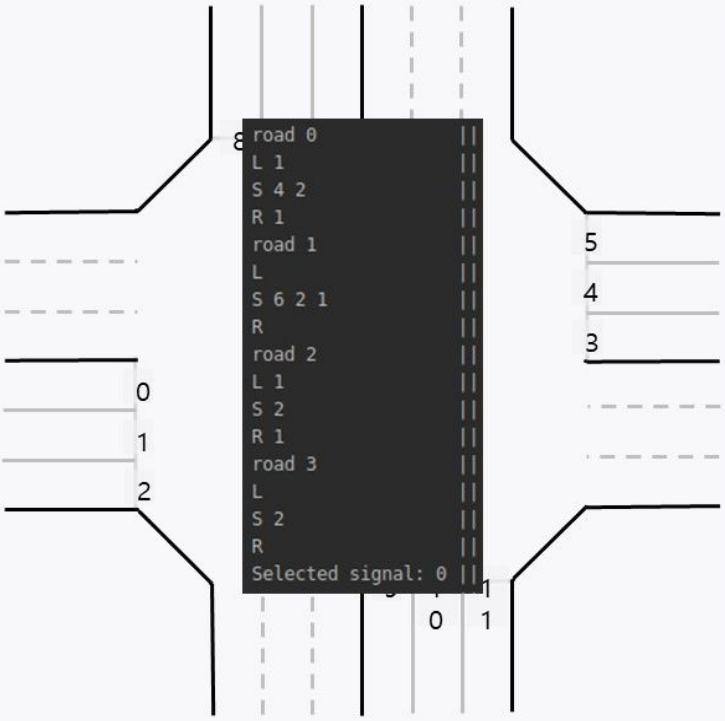


실제 교통량



모방한 분포

3 Environment Setting
환경 구성 - Environment



Environment

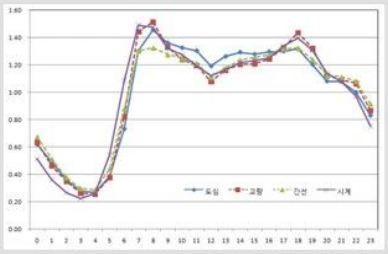
각 좌회전, 직진, 우회전을 담당하는 3차선 교차로 환경 구성

종료 조건

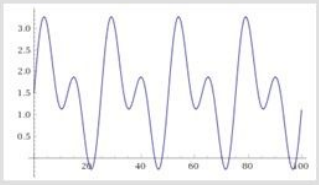
1. 특정 스텝에 도달할 경우 (200 steps)
2. 특정 차선의 차가 정의한 한계를 넘어갈 경우

차량 분포

실제 차량 분포를 모방한 분포를 따라 무작위로 생성



실제 교통량

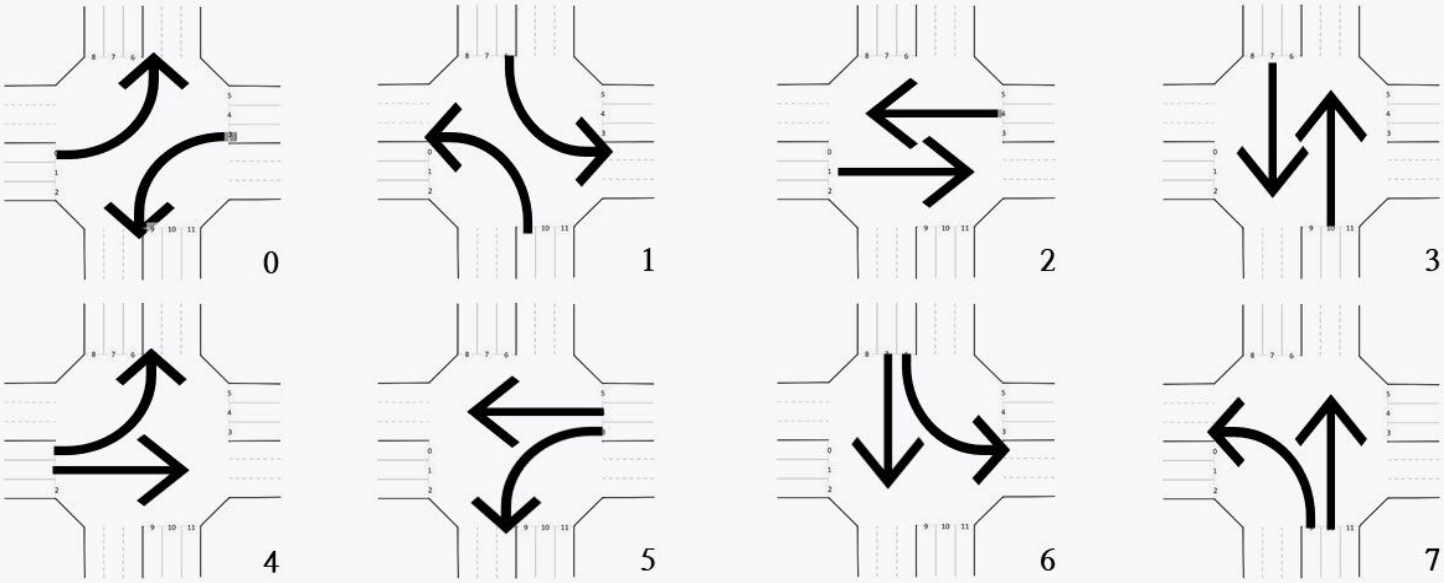


모방한 분포

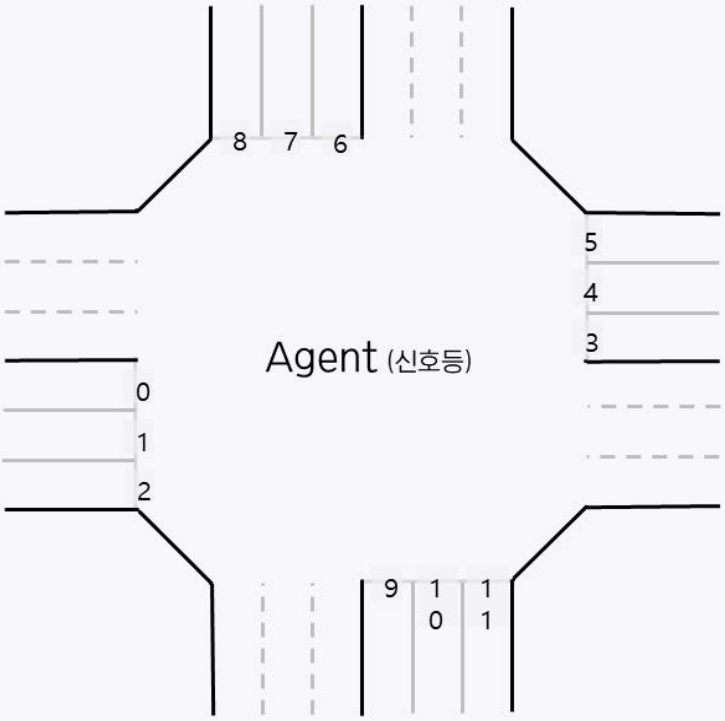
3 Environment Setting
환경 구성 - Action

Action

교차로를 통제하는 **8가지 신호**
신호를 비현실적으로 자주 바꾸는 것을 방지하기 위해 신호를 변경할 때마다 딜레이 부여



3 Environment Setting
환경 구성 - State



State
현재 도로의 상태

우회전 차선을 제외한 각 차선마다 최대 10개씩, 모든 차들의 각 대기시간

State size = 80

우회전 차선을 제외한 각 차선마다
(대기 중인 차의 개수, 가장 오래 기다린 차의 대기시간,
대기 중인 차들의 평균 대기 시간)

State size = 24

3 Environment Setting
환경 구성 - Reward

Reward

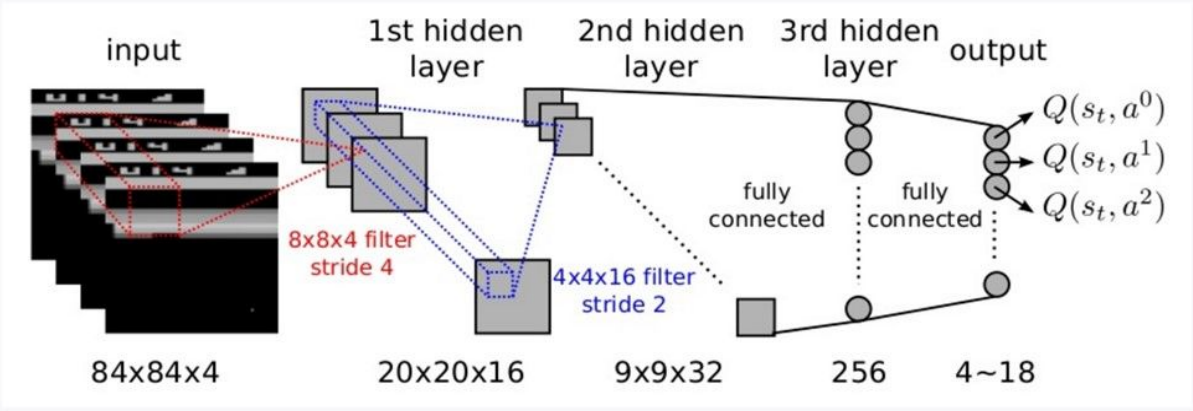
설정 목표

1. 전체적인 차들의 평균 대기시간 최소화
2. 가장 오래 기다린 차의 대기시간 최소화

Reward

1. 도로에 차가 한계치를 초과하지 않고 버틴 Step마다 +1
2. - (이번 스텝의 차 대기시간의 총 합)
3. (이번 Step에 내보낸 차의 대기시간) ^ n
   (n = 0.9 ~ 3.0)
4. (이번 Step에 내보낸 차의 대기시간) ^ 1
   - (가장 오래 기다린 차의 대기 시간) * a
   (a = 0.3 ~ 1.0)
5. 특정 차선의 대기 차량 수가 한계치를 넘을 경우
   (-1000)
6. ……

3 Deep Q-Networks
DQN



DQN

= Q-Learning + Neural Network

Google DeepMind 에서 Atari Game에 강화학습을 적용시킬 때 사용한 모델

3 Deep Q-Networks
DQN

Q-Learning

State-Action 쌍의 가치 함수를 반복 시행을 통해 업데이트

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

Experience Replay

Agent의 경험을 Step 단위로 데이터셋에 저장한다.

이후 데이터셋에서 랜덤 샘플링을 통해 미니배치를 구성하여 학습한다.

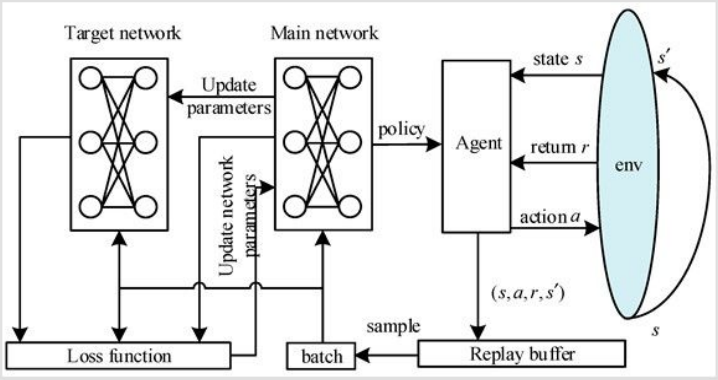
지역적 정보에 집중하여 overfitting되는 문제를 해결

Target Network

목표 함수가 계속 변해서 학습이 힘들어지는 문제를 해결하기 위해

Network를 2개 구성해서 하나는 현재의 값을 계산하고 다른 하나는 예측된 값을 계산한다.

Target Network는 몇 천번의 학습에 한번씩 교체된다.



DQN의 전체적인 흐름

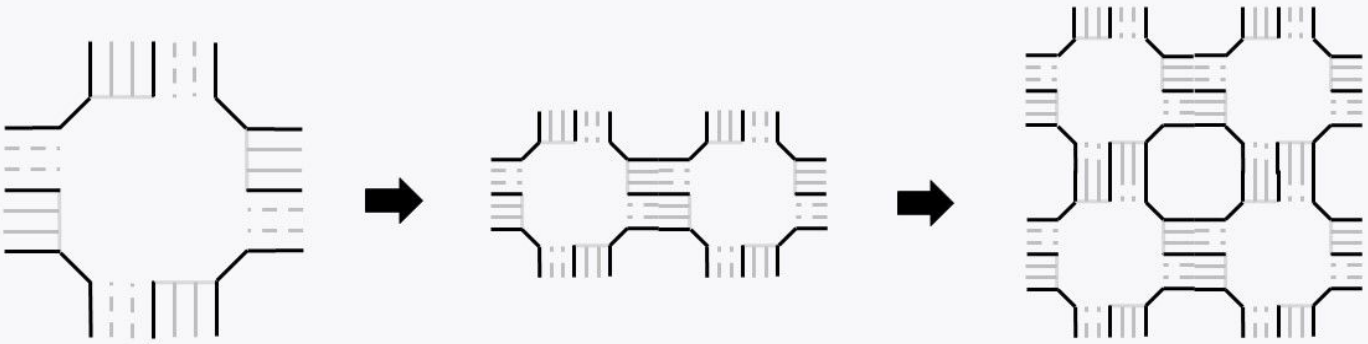
4 Process of Training
학습 진행 과정

4 Process of Training
학습 진행 과정

목표

1. 차들의 대기시간 최소화
2. 특정 차량이 너무 오래 기다리지 않도록

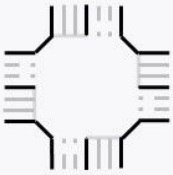
Discount Factor	Learning Rate	ε decaying rate	ε min	Loss	Optimizer
0.95	0.000005	0.999	0.05	MSE	Adam



단일 교차로

다중 교차로

4 Single Intersection
단일 교차로



단일 교차로

State

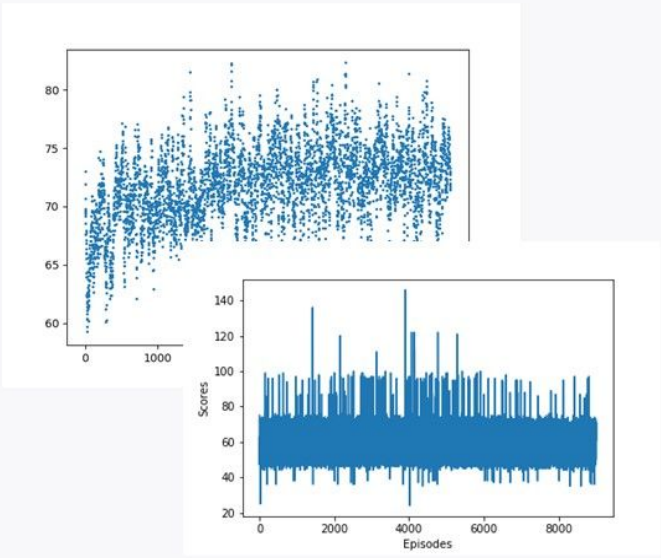
우회전 차선을 제외한 각 차선마다 최대 10개씩, 모든 차들의 각 대기시간
State size = 80

State가 너무 Sparse 했기 때문에 의사결정에 필요한 정보로 가공할 필요를 느낌

Reward

오래 기다린 차를 우선적으로 내보내기 위해
(이번 스탭에 내보낸 차의 대기 시간) ^ n 을 사용

모델이 많은 Reward를 얻기 위해 일부로 차를 내보내지 않고 도로에 차를 쌓아두는 현상이 발생



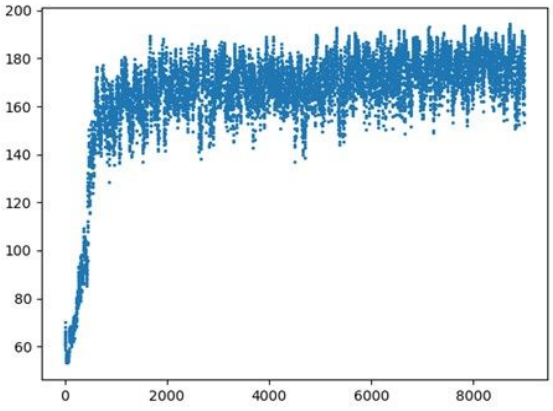
학습에 실패한 DQN 학습 그래프

4 Single Intersection
단일 교차로

State
우회전 차선을 제외한 각 차선마다
(대기 중인 차의 개수, 가장 오래 기다린 차의 대기시간,
대기 중인 차들의 평균 대기 시간)

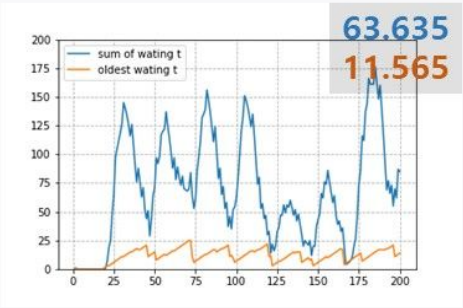
State size = 24

Reward
(이번 스탭에 내보낸 차의 대기 시간) ^ 1
- (해당 교차로에서 가장 오래 기다린 차의 대기 시간) * 0.5

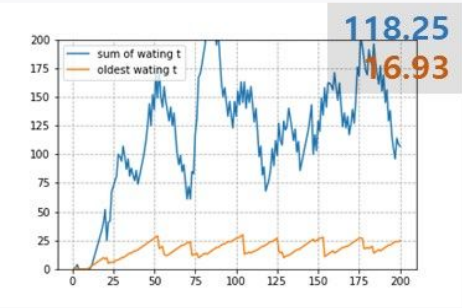


DQN 학습 그래프

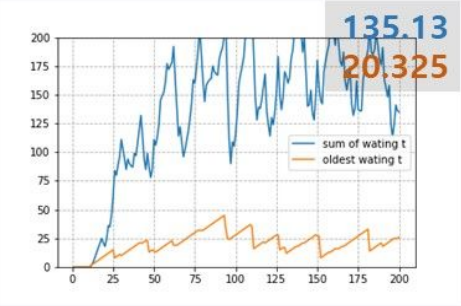
4 Single Intersection
단일 교차로



DQN 성능 그래프



Greedy 알고리즘 성능 그래프



Sequential 알고리즘 성능 그래프

차량 대기 시간의 총합을 기준으로 작동하는 Greedy 알고리즘 및 일정 순서에 따라 신호를 주는 현실에 가까운 Sequential 알고리즘에 비해 더 좋은 성능을 보여준다!

4 Single Intersection
단일 교차로

DQN 성능 그래프

Greedy 알고리즘 성능 그래프

Sequential 알고리즘 성능 그래프

하지만!

실제 도로는 많은 교차로가 유기적으로 작용한다.

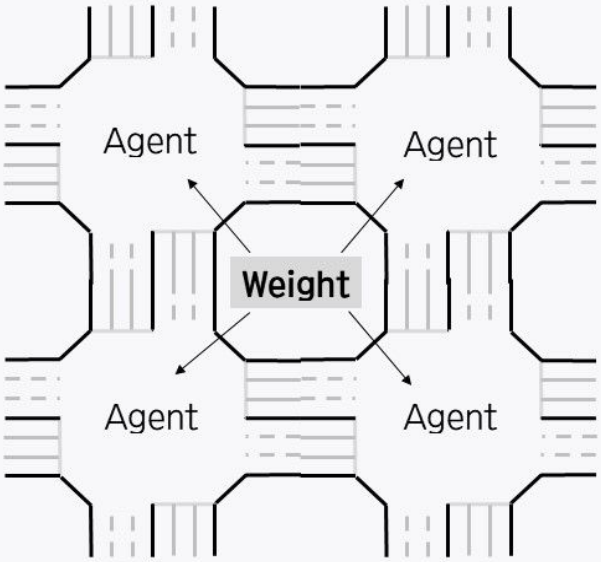
이러한 복잡한 상황에서도 강화학습이 효과적으로 작용할 수 있을까?

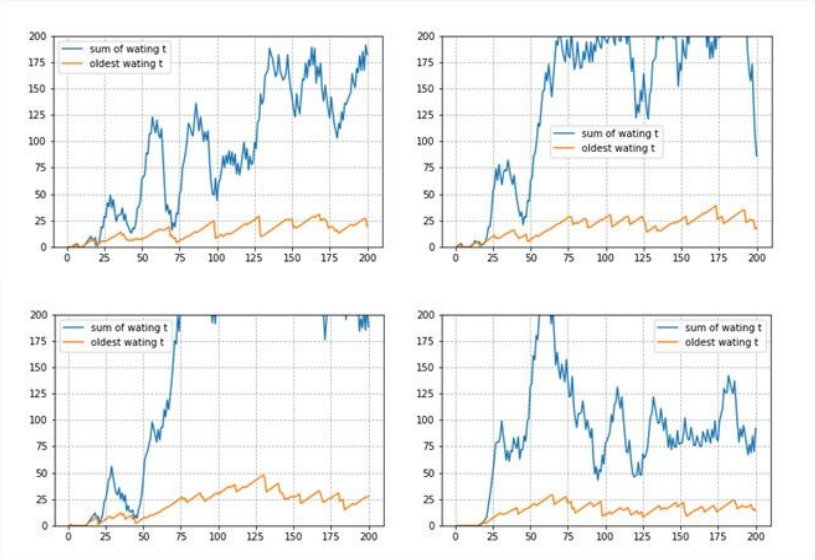
차량 대기 시간의 총합을 기준으로 작동하는 Greedy 알고리즘
보다 유기적으로 작용하는 다중 교차로에 적용해보자!

Sequential 알고리즘에 비해 더 좋은 성능을 보여준다!

4 Multi Intersection
다중 교차로 – 2x2

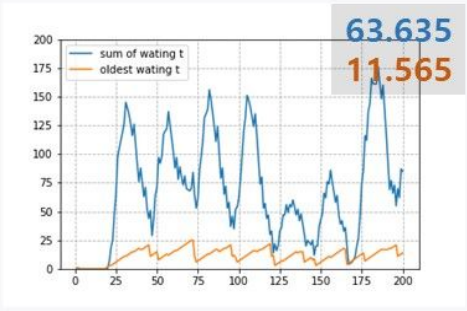
단일 교차로에서 학습시킨 Weight를 차량 분포 등이 동일한 2x2 교차로에 그대로 적용



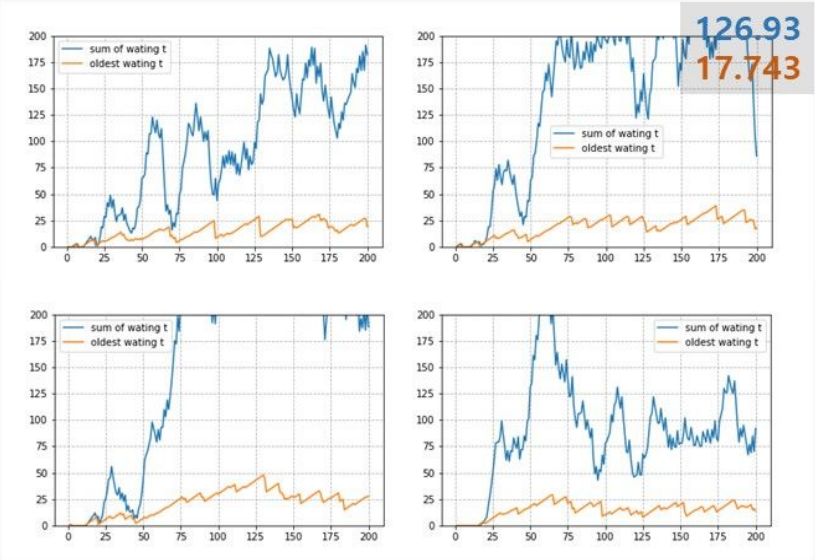


4 Multi Intersection
다중 교차로 – 2x2

같은 Weight를 단일 교차로에 적용시켰을 때에 비해 다중 교차로에 적용시켰을 때 **더 안 좋은 성능**을 보여주고 있다.



단일 교차로에 적용시켰을 때



같은 Weight를 각 교차로에 적용시켰을 때

4 Multi Intersection
다중 교차로 – 2x2

같은 Weight를 단일 교차로에 적용시켰을 때에 비해 다중 교차로에 적용시켰을 때 **더 안 좋은 성능**을 보여주고 있다.

4 Multi Intersection
다중 교차로 – 2x2

State 개선

State

우회전 차선을 제외한 각 차선마다
(대기 중인 차의 개수, 가장 오래 기다린 차의 대기시간,
대기 중인 차들의 평균 대기 시간)

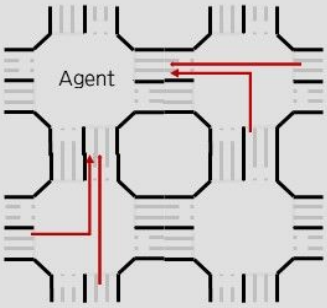
State size = 24

State

우회전 차선을 제외한 각 차선마다
(대기 중인 차의 개수, 가장 오래 기다린 차의 대기시간,
대기 중인 차들의 평균 대기 시간)

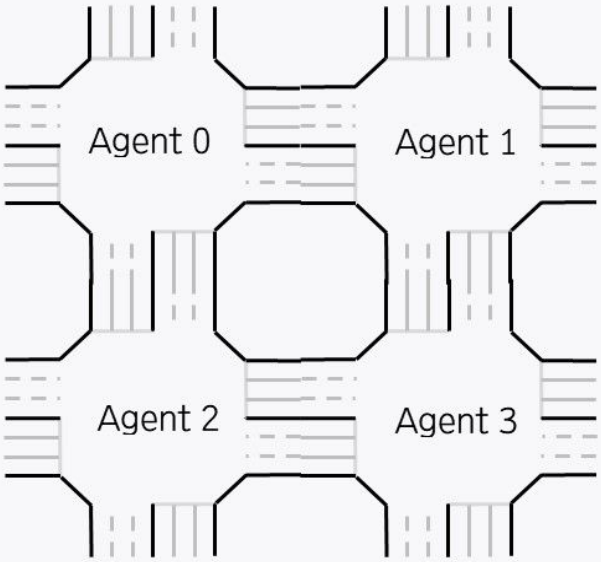
+ 인접 교차로에서 해당 교차로로부터 차를 받는 차선의 차의 개수

State size = 28



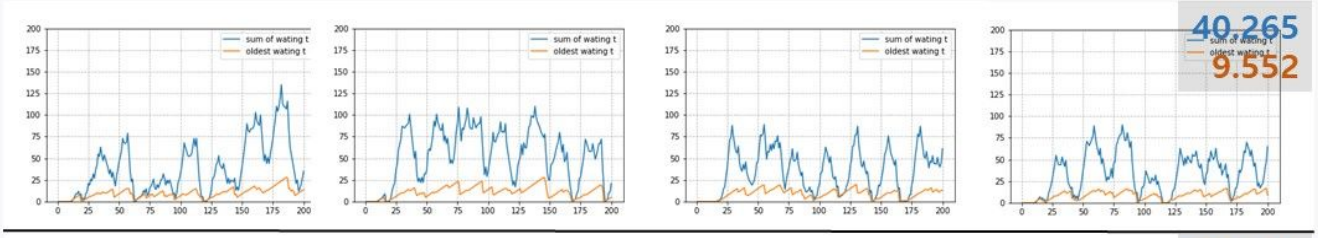
4 Multi Intersection
다중 교차로 – 2x2

각 교차로에 모델을 따로 학습시켜서 교차로마다 고유의 Weight를 갖도록 함
즉 4개 Agent가 서로 동시에 상호작용하며 학습이 진행됨



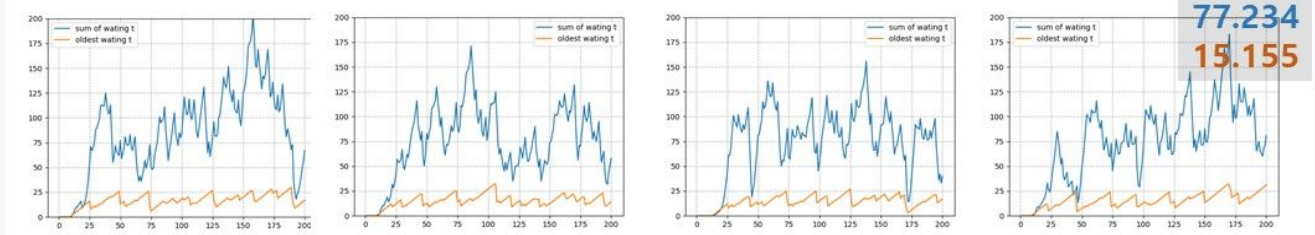
4 Multi Intersection
최종 성능 비교

DQN



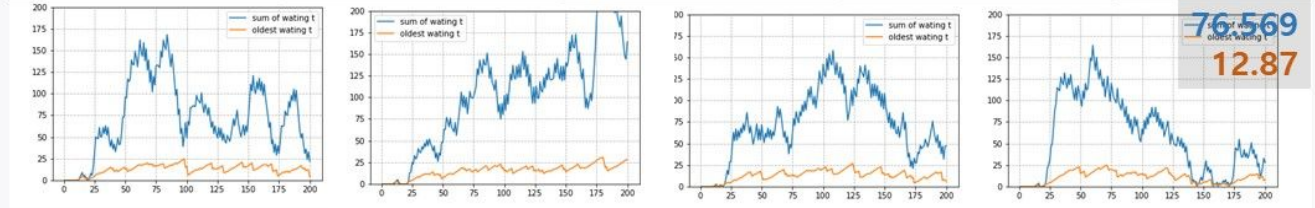
40.265
9.552

Sequential



77.234
15.155

Greedy

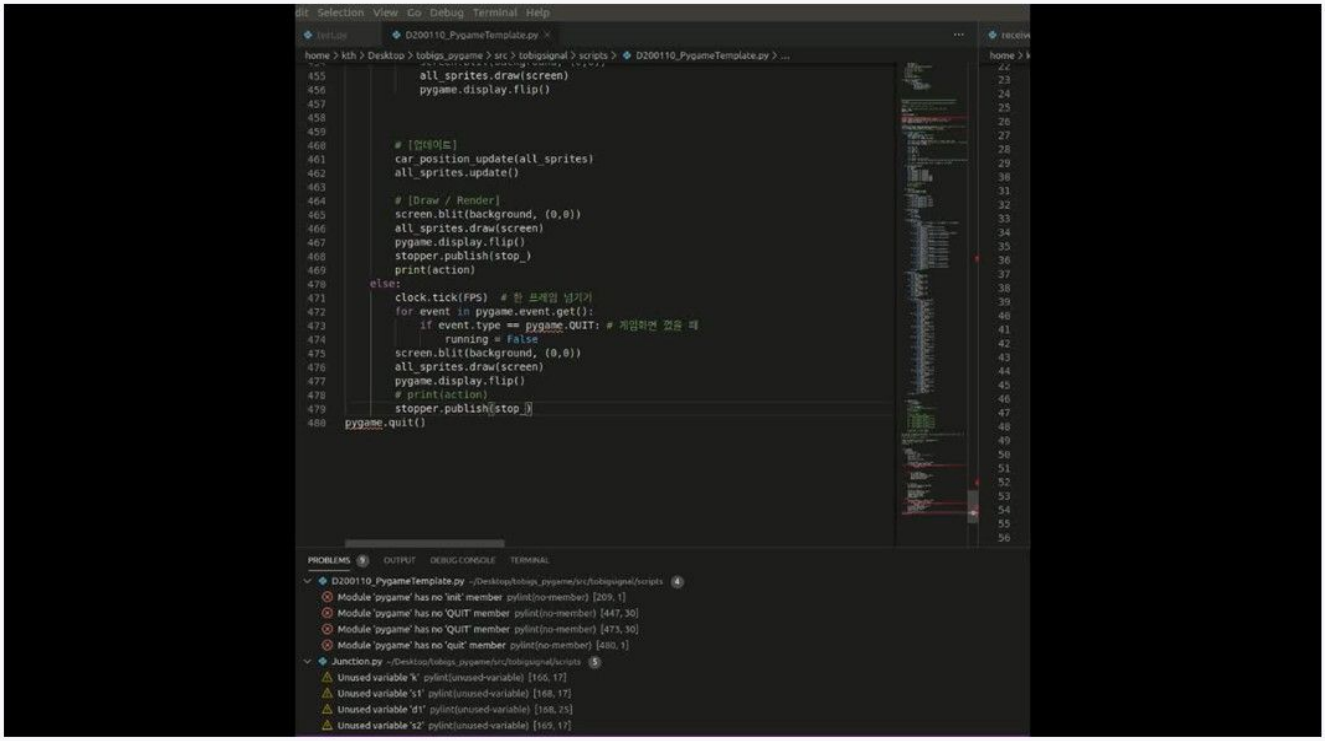


76.569
12.87

4 Multi Intersection
최종 성능 비교

		1	2	3	4	5	6	7	8	9	10	Overall
Sequential	Sum	77.234	72.623	88.511	89.376	91.98	80.901	72.048	76.282	74.182	73.826	79.6963 (0%)
	Old	15.155	14.71	17.211	16.124	16.665	16.108	14.786	14.548	15.011	14.73	15.5048 (0%)
Greedy *매우 느림*	Sum	76.569	97.746	85.049	66.216	96.118	72.818	61.547	69.81	97.466	89.672	81.3001 (+2.01%)
	Old	12.87	13.986	12.775	11.574	13.719	11.916	11.661	12.156	13.781	13.328	12.7766 (-17.59%)
DQN	Sum	40.265	47.867	61.393	59.636	49.672	46.15	51.868	53.18	59.365	54.342	52.3738 (-34.28%)
	Old	9.552	9.86	11.138	11.313	9.954	10.004	10.368	10.743	11.294	11.018	10.5244 (-32.12%)

4 Multi Intersection
Demo



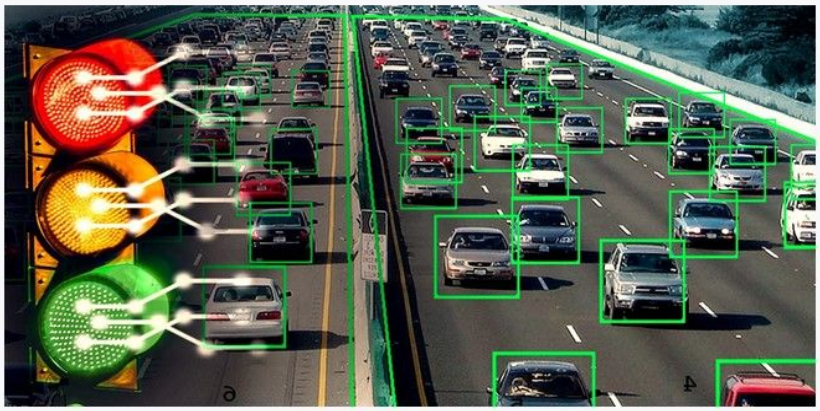
5 Conclusion
결론

5 Conclusion
의의

교통량 통제를 위한 강화학습 적용의 가능성

실제 교차로 환경과 유사한 환경을 구현해서 강화학습을 적용시켜 교통량을 효과적으로 통제할 수 있음을 보였다.

5 Conclusion
발전방향



발전 방향 더 현실에 가까운 환경 구성

Image Detection을 통해 도로상황을 받아 Environment로 구현한다면 실제 상황에 더욱 밀접한 학습 및 실제적인 적용이 가능할 것으로 기대된다.

5 Conclusion
한계

1 단순화된 Environment 및 State

차량의 속도 및 크기, 횡단보도 등의 변수를 고려하지 않은 단순화된 환경에서 학습을 진행했다.

특히 실제와 같은 복잡한 도로상황이 아니라 분포를 따라 무작위로 차량이 생성되어 실제와 구성된 환경 간에 큰 차이가 있다.

2 임의의 신호 순서로 인한 운전자 혼동

기존의 신호는 운전자의 혼동을 막기 위해 정해진 순서를 따른다. 하지만 본 모델은 매번 최적의 신호를 선택하기 때문에 운전자의 혼동이 있을 수 있다.

Q & A

Member