# DQN 실험 결과 및 개선



#### q\_map이란?

=> 각 방향으로 갔을 때 얻을 수 있는 보상의 기댓값을 나타낸 것.. 가치함수를 바탕으로 학습하여 agent가 어느방향으로 갈 지 판단하기 위해 필요한 것

## 1. Goal buffer를 사용하지 않았을 때

#### ▼ image

```
episode: 6035 & 0.100 steps: 7 state_cur: [46] time: 17:14 score: 43 memory:10000 reward: -10.0 episode: 6036 & 0.100 steps: 9 state_cur: [28] time: 17:14 score: 44 memory:10000 reward: +10.0 episode: 6037 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 45 memory:10000 reward: +10.0 episode: 6038 & 0.100 steps: 13 state_cur: [28] time: 17:14 score: 46 memory:10000 reward: +10.0 episode: 6039 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 46 memory:10000 reward: +10.0 episode: 6040 & 0.100 steps: 9 state_cur: [28] time: 17:14 score: 47 memory:10000 reward: +10.0 episode: 6041 & 0.100 steps: 9 state_cur: [28] time: 17:14 score: 48 memory:10000 reward: +10.0 episode: 6042 & 0.100 steps: 9 state_cur: [55] time: 17:14 score: 48 memory:10000 reward: -10.0 episode: 6043 & 0.100 steps: 9 state_cur: [28] time: 17:14 score: 49 memory:10000 reward: +10.0 episode: 6044 & 0.100 steps: 9 state_cur: [28] time: 17:14 score: 49 memory:10000 reward: +10.0 episode: 6045 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 50 memory:10000 reward: +10.0 episode: 6046 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 51 memory:10000 reward: +10.0 episode: 6046 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 51 memory:10000 reward: +10.0 episode: 6046 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 51 memory:10000 reward: +10.0 episode: 6046 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 51 memory:10000 reward: +10.0 episode: 6046 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 51 memory:10000 reward: +10.0 episode: 6046 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 51 memory:10000 reward: +10.0 episode: 6046 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 51 memory:10000 reward: +10.0 episode: 6046 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 51 memory:10000 reward: +10.0 episode: 6046 & 0.100 steps: 11 state_cur: [28] time: 17:14 score: 51 memory:10000 reward: +10.0 episode: 6046 & 0.100 steps: 11 state_cur: 6040 steps: 11 state_cur: 6040 steps: 6040 steps: 6040 steps: 6040 ste
```

```
| Application | Control |
```

```
ppisode: 3810 c. 0.100 steps: 13 state_cur: [28] time: 10:59 score: 10 semony:10000 remand: +10.0 episode: 3811 c: 0.100 steps: 15 state_cur: [28] time: 10:59 score: 11 semony:10000 remand: +10.0 episode: 3812 c: 0.100 steps: 13 state_cur: [28] time: 10:59 score: 12 semony:10000 remand: +10.0 episode: 3813 c: 0.100 steps: 13 state_cur: [28] time: 10:59 score: 13 semony:100000 remand: +10.0 episode: 3815 c: 0.100 steps: 13 state_cur: [28] time: 10:59 score: 15 semony:100000 remand: +10.0 episode: 3815 c: 0.100 steps: 13 state_cur: [28] time: 10:59 score: 15 semony:100000 remand: +10.0 episode: 3815 c: 0.100 steps: 13 state_cur: [28] time: 10:59 score: 15 semony:100000 remand: +10.0 episode: 3815 c: 0.100 steps: 13 state_cur: [28] time: 10:59 score: 17 semony:100000 remand: +10.0 episode: 3817 c: 0.100 steps: 13 state_cur: [28] time: 10:59 score: 17 semony:100000 remand: +10.0 progress = 38 % --> 3818/100000 Early Stopping

Process finished with exit code 0
```

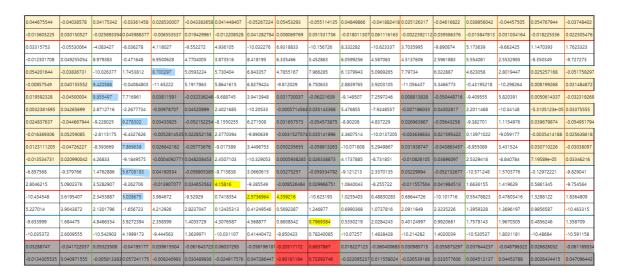
### ▼ 실패 개수(2개)

```
episode: 9990 ε: 0.100 steps: 8 state_cur: [79] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9991 ε: 0.100 steps: 76 state_cur: [79] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9992 ε: 0.100 steps: 1 state_cur: [76] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9993 ε: 0.100 steps: 30 state_cur: [79] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9994 ε: 0.100 steps: 44 state_cur: [79] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9995 ε: 0.100 steps: 6 state_cur: [79] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9996 ε: 0.100 steps: 49 state_cur: [78] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9997 ε: 0.100 steps: 30 state_cur: [79] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9998 ε: 0.100 steps: 30 state_cur: [79] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9999 ε: 0.100 steps: 13 state_cur: [78] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9999 ε: 0.100 steps: 13 state_cur: [78] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9999 ε: 0.100 steps: 13 state_cur: [78] time: 29:08 score: 0 memory:10000 reward: -10.0 episode: 9999 ε: 0.100 steps: 13 state_cur: [78] time: 29:08 score: 0 memory:10000 reward: -10.0
```

goal에 도착해서 reward를 받았음에도 goal 방향으로 안가고 자꾸 오른쪽으로 감

#### ▼ 실패 q-map 이상(2개)

```
episode: 7419 & 0.100 steps: 9 state_cur: [28] time: 21:12 score: 53 memory:10000 reward: +10.0 episode: 7420 & 0.100 steps: 9 state_cur: [28] time: 21:12 score: 53 memory:10000 reward: +10.0 episode: 7421 & 0.100 steps: 9 state_cur: [28] time: 21:12 score: 54 memory:10000 reward: +10.0 episode: 7422 & 0.100 steps: 9 state_cur: [28] time: 21:12 score: 54 memory:10000 reward: +10.0 episode: 7423 & 0.100 steps: 11 state_cur: [28] time: 21:12 score: 54 memory:10000 reward: +10.0 episode: 7424 & 0.100 steps: 9 state_cur: [28] time: 21:12 score: 54 memory:10000 reward: +10.0 episode: 7425 & 0.100 steps: 9 state_cur: [28] time: 21:12 score: 54 memory:10000 reward: +10.0 episode: 7426 & 0.100 steps: 9 state_cur: [28] time: 21:12 score: 54 memory:10000 reward: +10.0 episode: 7426 & 0.100 steps: 9 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 11 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 11 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 12 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 12 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 12 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 12 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 12 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 12 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 12 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 12 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 12 state_cur: [28] time: 21:12 score: 55 memory:10000 reward: +10.0 episode: 7427 & 0.100 steps: 12 state_cur: [28] time: 21:12 score: 54 memory:10000 reward: +10.0 episode: 7426 & 0
```



기존에 사용하던 방법으로 학습을 진행했을 때 goal에 70%로 도착해 조기종료가 되었음에도 g-map이 제대로 안나오는 경우가 10번 중 2번 있었다.

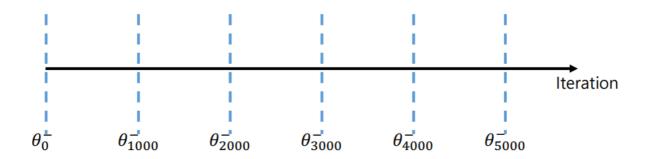
최단 경로가 9 임에도 11로 종료되어 마지막에 나온 q-map이 최단거리가 아닌 경우도 10번 중 4번 있었다.

## 2. target network 초기화

train 네트워크(=update network)를 학습할 때마다 target network를 train 네트워크로 초기화 했을 때 target 값이 자꾸 바뀌어 제대로된 학습을 진행하지 못했습니다.

```
pisode: 9989 c: 0.100 steps: 28 state_cur: [75] time: 29:41 score: 0 memory:10000 reward: -10.0 episode: 9990 c: 0.100 steps: 05 state_cur: [72] time: 29:44 score: 0 memory:10000 reward: -10.0 episode: 9992 c: 0.100 steps: 20 state_cur: [75] time: 29:44 score: 0 memory:10000 reward: -10.0 episode: 9992 c: 0.100 steps: 46 state_cur: [75] time: 29:44 score: 0 memory:10000 reward: -10.0 episode: 9992 c: 0.100 steps: 26 state_cur: [75] time: 29:44 score: 0 memory:10000 reward: -10.0 episode: 9995 c: 0.100 steps: 74 state_cur: [75] time: 29:44 score: 0 memory:10000 reward: -10.0 episode: 9996 c: 0.100 steps: 30 state_cur: [75] time: 29:44 score: 0 memory:10000 reward: -10.0 episode: 9996 c: 0.100 steps: 10 state_cur: [73] time: 29:44 score: 0 memory:10000 reward: -10.0 episode: 9997 c: 0.100 steps: 16 state_cur: [73] time: 29:44 score: 0 memory:10000 reward: -10.0 episode: 9999 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:10000 reward: -10.0 episode: 9999 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:10000 reward: -10.0 episode: 9999 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:10000 reward: -10.0 episode: 9999 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:100000 reward: -10.0 episode: 9999 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:100000 reward: -10.0 episode: 9999 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:100000 reward: -10.0 episode: 9999 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:100000 reward: -10.0 episode: 9990 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:100000 reward: -10.0 episode: 9990 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:100000 reward: -10.0 episode: 9990 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:100000 reward: -10.0 episode: 9990 c: 0.100 steps: 18 state_cur: [75] time: 29:41 score: 0 memory:100000 reward: -10.0 episode: 9990 c: 0.10000 reward: -10.0 episode: 9990 c: 0.10000 reward: -10.0 episode: 9990 c: 0.10
```

target이 계속 바뀌어서 step을 멀리 밟지 못한다.



target 네트워크를 일정 episode마다 초기화해주는 방법을 통해 이 문제를 해결했고 100~900까지 테스트해본 결과 500 episode마다 업데이트 해주는 것이 제일 성능이 좋았습니다 .

## 3. goal 버퍼 사용

기존의 랜덤하게 샘플링 방법은 goal 로 들어가는 데이터가 샘플링이 되지않아 학습속도가 느리고 의미없는 데이터를 많이 학습하여 학습이 제대로 이루어지지않는 경우가 있었습니 다.

```
# ksy 20%반영 -> 25%

goal_sampling = int(sampling / 4)

if len(self.__goal_buffer) > 0:
    if len(self.__goal_buffer) > goal_sampling:
        goalbatch = random.sample(self.__goal_buffer, goal_sampling)
    else:
        goalbatch = list(self.__goal_buffer)

if len(goalbatch) > 0:
    minibatch = random.sample(self.__replay_buffer, sampling - len(goalbatch))
    minibatch = minibatch + goalbatch

else:
    minibatch = random.sample(self.__replay_buffer, sampling)
```

goal 버퍼를 따로 만들어 학습할 때 (goal로 들어가는 데이터 + 샘플링 데이터)로 학습한 결과 10번 테스트한 것중 1번을 제외하고 평균 episode 7000 정도에 조기종료했고 최종 q-map도 잘 나왔다.

goal 데이터로만 학습하는 것보다 샘플링 데이터를 섞어주는 것이 더 좋은 성능을 보였다. goal 데이터만 사용하면 기존의 q-network가 가진 **Correlation** 문제를 해결하지 못하는 것으로 보임.

## 문제점

▼ image

```
episode: 4413 ɛ: 0.100 steps: 15 state_cur: [28] all_steps: 83349 time: 12:07 score: 38 memory:10000 reward: +150.0 episode: 4414 ɛ: 0.100 steps: 11 state_cur: [28] all_steps: 83360 time: 12:08 score: 39 memory:10000 reward: +150.0 episode: 4415 ɛ: 0.100 steps: 15 state_cur: [28] all_steps: 83375 time: 12:08 score: 40 memory:10000 reward: +150.0 episode: 4416 ɛ: 0.100 steps: 13 state_cur: [28] all_steps: 83388 time: 12:08 score: 41 memory:10000 reward: +150.0 episode: 4417 ɛ: 0.100 steps: 15 state_cur: [28] all_steps: 83403 time: 12:08 score: 42 memory:10000 reward: +150.0 episode: 4418 ɛ: 0.100 steps: 11 state_cur: [28] all_steps: 83414 time: 12:08 score: 43 memory:10000 reward: +150.0 episode: 4419 ɛ: 0.100 steps: 11 state_cur: [28] all_steps: 83425 time: 12:08 score: 43 memory:10000 reward: +150.0 episode: 4420 ɛ: 0.100 steps: 11 state_cur: [28] all_steps: 83436 time: 12:08 score: 44 memory:10000 reward: +150.0 progress = 44 % --> 4420/10000 Early Stopping
```

0.2473889	0.5878572	-0.1718118	0.5972686	-0.25143892	0.645792	-0.35927343	0.7055502	-0.3386294	0.701625	-0.32093525	0.6088719	-0.32700735	0.5686799	-0.285765	0.60056925	-0.20059359	0.64830357
0.124564506	-0.2028528	-0.09606858	-0.23641115	-0.13938227	-0.28765938	-0.09586877	-0.3221788	-0.1784575	-0.24008924	-0.09230142	-0.2224371	-0.11925353	-0.31353584	-0.1399423	-0.23191783	-0.16539149	-0.2418842
13.9755335	-95.79291	21.461718	-103.88494	-0.22313043	-62.958084	12.01349	-101.27118	16.325247	-101.759476	7.263352	-93.72817	8.696232	-82.575485	10.905174	-84.605286	10.810156	-100.70357
18.07641	24.385368	15.918595	29.28204	6.2556505	8.5586	16.504421	29.278183	9.525335	27.806738	26.562532	28.954842	56.144127	14.638397	48.975643	24.027843	2.3609865	12.827578
0.2721488	0.6147004	-115.601746	35.210068	12.2816305	5.974453	-19.156551	-6.007237	22.210808	19.047987	37.9417	-6.0270963	47.36741	16.728518	68.75802	32.556923	-0.32959306	0.5693771
0.16231923	-0.16105503	-4.7466497	-45.29772	-90.87451	20.347275	3.2183366	41.3367	-69.50841	37.454586	10.096941	45.02338	-72.98039	28.555363	-3.2526991	-75.03459	-0.19797796	-0.24046418
0.2881182	0.5485849	147.30394	-50.575527	-0.23267783	0.6327717	-104.88033	21.099552	-0.30611566	0.6221279	-47.092075	60.55494	-0.22385393	0.4967631	-97.507416	58.216446	-0.32118458	0.5708907
0.1469117	-0.19162232	8.607803	64.52641	-0.10571553	-0.29072413	1.1234684	-70.66753	-0.11255999	-0.33187434	13.162766	-83.81691	-0.16099744	-0.20262712	8.839823	-94.938194	-0.051856276	-0.21502595
0.29016632	0.64379907	-28.973242	128.8085	-0.33753324	0.5738443	-88.77383	11.884001	-0.18123326	0.613845	-102.79353	27.16368	-0.34295157	0.6326435	-120.89048	25.219528	-0.24382097	0.6572108
0.1132907	-0.31496358	-25.135576	-39.763664	-0.15602034	-0.2759093	-2.675142	-84.84964	-0.07875834	-0.21397111	5.790535	-76.28093	-0.15375619	-0.30381745	2.8600354	-37.70178	-0.11183336	-0.18813764
0.28752568	0.6373791	-106.20159	38.071445	-0.3297759	0.65830535	-98.28887	15.0936775	-0.3167851	0.6395739	-82.9954	19.795115	-0.21520156	0.58722186	-87.133446	9.378679	-0.082473114	0.5858929
0.111509904	-0.29793707	7.871302	-91.16181	-0.1919755	-0.22607268	-1.3562152	-71.37271	-0.17756078	-0.15678982	5.1068573	-90.652794	-0.12268769	-0.17792922	-1.66988	-71.7897	-0.08940843	-0.29080436
76.25648	-58.970818	9.3019285	19.670109	-0.24824017	0.65508616	-79.70718	12.039303	-0.251911	0.66437507	-95.574295	21.248825	-0.3073617	0.68721724	-109.021706	0.77279925	-0.69100684	-79.386215
2.858119	16.310036	8.173183	-94.51579	-0.16116638	-0.19680752	3.3600075	-84.366844	-0.07162302	-0.29112723	8.306572	-92.57197	-0.1596622	-0.3590595	7.8855195	6.839245	32.99828	-17.808983
102.52744	6.142514	5.522835	5.3911805	-2.3931632	-89.07437	4.236037	2.5601764	8.961399	-106.4994	4.8415527	-7.5729437	24.908094	-103.941444	-5.72899	40.182957	15.325282	-0.22740233
3.8298514	-0.502198	3.542073	4.04315	3.0397155	16.114279	1.0926585	3.0822866	3.9628997	27.061127	2.478307	22.899431	7.3886914	36.916935	-7.591771	10.402594	9.3316	-89.70458
64.09393	3.1309857	20.889694	5.312881	23.202827	4.749989	18.223444	3.6686845	17.40622	2.605297	4.00788	15.7105	16.701681	18.764019	36.441395	-2.482955	25.168362	23.494535
76.37362	-0.028770685	-95.84708	8.714478	-99.15136	7.346711	-101.390976	11.623903	-101.3766	10.858876	-102.78494	7.981905	-100.01811	19.79296	-93.71337	4.2945185	-53.926212	-72.58015
0.15604773	0.54851335	-0.31968704	0.675596	-0.22216882	0.682511	-0.20208773	0.57392377	3.0752583	2.6022365	-0.30745086	0.6397861	-0.3411168	0.6392216	-0.24701779	0.6627938	-0.27336085	0.5504874
0.14339979	-0.23758732	-0.17455323	-0.23451398	-0.19147426	-0.16703323	-0.19471805	-0.2637072	0.65726006	-0.40027934	-0.07964119	-0.29105717	-0.15405828	-0.28129062	-0.0958648	-0.23485062	-0.16778584	-0.21588399

## ▼ 안좋은 goal 데이터

```
idx_epi : 999
time,action,p_cur,p_new,state_cur,state_new,reward,done,result_step
220530_161441,U,"(9, 4)","(8, 4)",85,76,-0.01,False,ID_GENERAL_MOVE
220530_161441,R,"(9, 4)","(8, 5)",76,77,-0.01,False,ID_GENERAL_MOVE
220530_161441,R,"(9, 4)","(8, 6)",77,78,-0.01,False,ID_GENERAL_MOVE
220530_161441,U,"(9, 4)","(7, 6)",78,69,-0.01,False,ID_GENERAL_MOVE
220530_161441,L,"(9, 4)","(7, 5)",69,68,-0.01,False,ID_GENERAL_MOVE
220530_161441,D,"(9, 4)","(8, 5)",68,77,-0.01,False,ID_GENERAL_MOVE
220530_161441,L,"(9, 4)","(8, 4)",77,76,-0.01,False,ID_GENERAL_MOVE
220530_161441,L,"(9, 4)","(8, 3)",76,75,-0.01,False,ID_GENERAL_MOVE
220530_161441,L,"(9, 4)","(8, 2)",75,74,-0.01,False,ID_GENERAL_MOVE
220530_161441,L,"(9, 4)","(8, 1)",74,73,-0.01,False,ID_GENERAL_MOVE
220530_161441,L,"(9, 4)","(8, 0)",73,72,-0.01,False,ID_GENERAL_MOVE
220530_161441,U,"(9, 4)","(7, 0)",72,63,-0.01,False,ID_GENERAL_MOVE
220530_161441,U,"(9, 4)","(6, 0)",63,54,-0.01,False,ID_GENERAL_MOVE
220530_161441,D,"(9, 4)","(7, 0)",54,63,-0.01,False,ID_GENERAL_MOVE
220530_161441,D,"(9, 4)","(8, 0)",63,72,-0.01,False,ID_GENERAL_MOVE
220530_161441,U,"(9, 4)","(7, 0)",72,63,-0.01,False,ID_GENERAL_MOVE
220530_161441,U,"(9, 4)","(6, 0)",63,54,-0.01,False,ID_GENERAL_MOVE
220530_161441,R,"(9, 4)","(6, 1)",54,55,-0.01,False,ID_GENERAL_MOVE
220530_161441,U,"(9, 4)","(5, 1)",55,46,-0.01,False,ID_GENERAL_MOVE
220530_161441,U,"(9, 4)","(4, 1)",46,37,-0.01,False,ID_GENERAL_MOVE
220530_161441,U,"(9, 4)","(3, 1)",37,28,-0.01,False,ID_GENERAL_MOVE
220530_161441,D,"(9, 4)","(4, 1)",28,37,-0.01,False,ID_GENERAL_MOVE
220530_161441,U,"(9, 4)","(3, 1)",37,28,-0.01,False,ID_GENERAL_MOVE
220530_161441,L,"(9, 4)","(3, 0)",28,27,150,True,ID_GOAL
```

goal로 들어간 데이터가 최적화된 경로가 아닐 경우 위와 같은 문제가 발생했다.

## 해결

위 문제를 해결하기위해 매번 goal 버퍼에 저장할 때 step을 비교해 최소 step을 갱신하고 최소 step인 데이터만 goal 버퍼에 저장해서 최적화된 경로를 학습한 결과 평균 종료 episode가 3000~4000 사이로 감소했고 q-map도 최적 경로로 잘 출력되었다.

```
while len(self.l_temp_buffer) > 0 and cnt_step <= self.__goal_min_step:
    self.__goal_buffer.clear()
    self.__goal_buffer.append(self.__temp_buffer.pop())
    self.__goal_min_step = cnt_step</pre>
```

```
| March | Marc
```

#### ▼ 조기종료

• 기존 방법

```
def __check_early_stopping(self, early_stopping, idx_epi, buf_result):
    if early_stopping:
        if idx_epi == 0:
            early_stopping.clear()
        if not self.dqn_update.loss == 0:
            flg = idx_epi
            if buf_result == ID_GOAL:
                flg = True
            # if early_stopping.check_stopping(round(float(self.dqn_update.loss), 5)):
            if early_stopping.check_stopping(flg):
                return False
            return True
```



마지막에 들어오는 값과 리스트 안에 들어 있는 값을 비교한다. 성공은 TRUE로 실패는 episode를 저장한다. Ratio 비율을 넘으면 종료한다

기존 방법은 최적경로로 가지 않았음에도 종료가 되어서 최종으로 나온 q-map이 최적 경로가 아닌 경우가 있었다.

• 수정된 방법

FALSE	FALSE	FALSE	11	9	9	9	FALSE	27	9

비교하는 방법은 이전과 동일하나 수정된 코드에서는 성공했을 때 TRUE가 아닌 step으로 저장한다. 즉 최소 스텝으로 goal로 도착한 비율이 ratio를 넘길 때 종료된다.

## 4. 학습된 가중치를 사용해 학습한 결과

▼ S

 $\bigvee A \rightarrow S$ 

**▼** TEST

• 1차(epi: 2522)



• 2차(epi: 1947)



• 3차(epi: 2022)



#### **▼** B → S

#### **▼** TEST

• 1차(epi: 3110)



• 2차(epi: 2516)



• 3차(epi: 2361)



• 4차(epi: 2810)



• 5차(epi: 2867)



### ▼ 결론

기존에 학습했던 A  $\rightarrow$  S로 가는 q-map이 2차 학습까지 남아있었으나 3차 학습에서 사라졌다.

#### **▼** A

### **▼** S → A

#### **▼** TEST

• 1차( epi: 2052)

											_						
0.05807693	-0.0018005731	0.05062159	-0.005216345	0.02688203	0.011590421	0.046649568	0.007321138	0.048895117	-0.004683429	0.03220471	0.008266726	0.045371473	0.0040016766	0.054119583	-0.012343239	0.04944288	0.023826394
0.013371028	-0.040792346	0.024137301	-0.027443495	0.003614995	-0.021454643	-0.009991177	-0.037127126	0.0045996425	-0.026282907	0.007153435	-0.02919371	0.013735766	-0.026866136	0.010746002	-0.036808234	0.0005076914	-0.03798352
-4.335371	-9.366942	-4.136647	0.5837093	0.040425584	0.0010306358	-6.6448927	-3.356661	-1.7077775	1.0842205	7.917059	-1.4348501	4.8330374	-10.061203	4.4572225	-8.120426	0.044470623	0.011462334
-1.1547534	-2.8219764	0.08136659	2.6367958	0.014670823	-0.038792413	-0.31758112	-2.465414	-1.1376712	1.109452	-1.3696288	-0.6818811	-0.37319344	-0.9564227	-1.3753155	-2.8963795	-0.018789722	-0.04529345
0.055794153	-0.005229513	-10.002027	-1.526043	3.8677783	0.041055664	-11.633376	1.6124252	3.6983917	-6.4774494	1.2810271	-2.8251703	1.2307805	3.0468028	-8.129963	1.5643526	0.038340293	0.00463592
-0.011946514	-0.02173438	0.2945191	0.11917636	-1.0299813	1.0159545	-0.94346356	6.131336	-0.7075354	5.6808167	-0.3448562	4.4552855	-10.368867	-1.4838614	-1.4421313	1.8778133	-0.00072655914	-0.026829377
0.045369085	-0.00288274	0.053421427	0.0044827479	0.028057283	0.012769553	-11.012691	3.6542625	0.041942727	0.0082472265	-6.863311	3.7718258	0.03607912	-0.009584432	-8.7526	1.0541797	0.034834884	0.005534431
0.0015913565	-0.018495785	0.0028045974	-0.037083417	0.016179275	-0.02190382	2.935171	1.9180566	-0.0028745318	-0.026147552	0.76109856	-3.463946	0.007781923	-0.026609883	4.949219	-8.737877	-0.0048239883	-0.02704375
0.03879848	0.019147174	1.405904	-1.0485235	0.03855523	0.006371443	-8.563506	0.26573128	0.04409509	0.0039343964	-5.244801	2.5605423	0.035402875	0.011294606	-7.5150094	0.19679135	0.06336076	0.007875506
0.0141997	-0.033033147	6.0370994	-2.3348897	-0.007075087	-0.024026148	1.150068	-5.275425	-0.013032355	-0.021582335	-0.12761778	-8.894237	-0.00829375	-0.023631107	-1.3438882	-8.973109	-0.027031181	-0.029227167
0.021347392	-0.0017763075	9.930548	2.0256004	0.030494237	0.004526842	-8.7997465	2.111753	0.037503406	0.002068821	-9.672575	0.9539288	0.04526798	-0.0015471079	-9.667547	6.3300824	0.057702415	0.009417773
0.008634145	-0.016724158	0.5646818	-9.712404	0.006259796	-0.024496585	-0.38371933	-6.7706265	0.011664889	-0.012337618	0.7234888	-3.923676	-0.011172224	-0.016872678	3.5903258	-4.333091	0.001931627	-0.037466645
-8.606993	8.920186	8.899704	5.245031	0.0496143	0.017937424	-6.372694	0.90695935	0.033321604	0.005234294	-8.143991	-0.124334574	0.05624076	0.010100159	-9.862979	2.1884587	4.638806	-8.36465
6.3578677	6.6239014	-0.43672 <mark>4</mark> 75	-11.284216	-0.008375078	-0.038081817	0.99048007	-8.872091	0.01720698	-0.028831055	0.11682275	-7.2756047	0.012386021	-0.03278055	1.2132627	2.3521953	4.128715	-9.827279
-11.426656	8.263478	5.092229	7.5497093	1.0196182	-9.650094	2.2517996	-0.25035644	1.3870549	-9.339776	1.092217	1.0107067	0.8935725	-9.254855	4.812119	4.3197656	2.029335	2.3042774
0.78558505	2.1950216	1.3080223	2.924825	0.8369875	0.68892795	0.47092202	0.84082884	0.6992683	0.4916435	0.95655805	4.468657	0.8143224	1.7058021	5.256752	3.085553	1.4137837	-10.010117
-10.089447	5.4231124	1.2936754	1.0131072	2.3127415	2.1088736	0.69976604	1.1874475	0.52588934	0.9214995	0.18583679	0.7157853	0.9734179	4.605439	0.50033945	1.7780033	4.870592	3.8633165
-10.750721	2.476605	-10.309705	2.4878356	-10.136413	0.33209294	-10.56317	0.5572728	-10.542552	0.85152245	-10.571666	1.2171423	-10.294524	5.186062	-10.157032	1.7404807	-8.209067	-9.302083
0.047136154	-0.002604878	0.053112607	0.0009177525	0.056316104	0.01043384	0.0611598	0.002629270	-4.7274766	0.6249134	0.062796876	0.0047356687	0.06082682	0.0074325344	0.048341062	0.022668246	0.05056151	-0.004366295
-0.007392925	-0.028497688	-0.006917572	-0.050098486	-0.012711301	-0.040175684	0.011468669	-0.033234436	-0.98263395	1.4321493	-0.008091897	-0.03169921	-0.012809563	-0.038979147	0.01993895	-0.027652914	-0.010780538	-0.030567735

• 2차( epi: 1910)



• 3차(epi: 1799)



• 4차(epi: 1584)



• 5차(epi: 1545)



#### **▼** B

**▼** S → B

▼ TEST

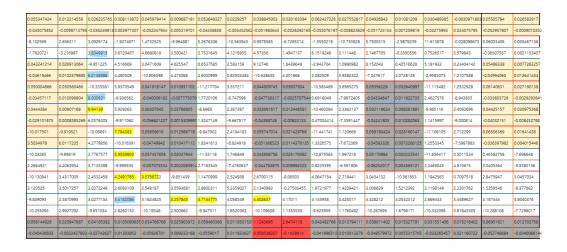
• 1차(epi: 2266)



• 2차(epi: 3219)



• 3차(epi: 1591)



• 4차(epi: 2210)



**▼** A → B

**▼** TEST

• 1차(epi: 1254)



A → B를 갈 때 위로 바로 가지 않도록 예외처리

```
# 예외처리 : 시작지점과 Goal Item이 붙어있는 경우 장애물로 처리
if reward == self.env.REWARD.GOAL and cnt_step == 0:
reward = self.env.REWARD.OBSTACLE
```

결과적으로 바로 위로 가지않고 goal에 도착한다.

• 2차(epi: 1610)



 $A \rightarrow B$ 를 학습하며 기존에 가지고 있던  $S \rightarrow B$  데이터가 일부 지워짐

▼ 가설

기존의 가중치를 사용해서 학습을 하면 어느 지점에서 출발하더라도 같은 경로로 이동하는 q-map이 만들어질 것이다.

#### ▼ 결론

- 1. 학습된 가중치를 사용해도 또 다른 경로를 찾는 경우가 존재한다.
  - a. 탐험으로 인해 noise가 생기지만 전체적인 q-map이 다양한 경로를 통해 goal로 가는 것을 학습한다
- 2. 기존에 학습된 가중치 값을 일부 가지고 있어서 새로운 경로 + 기존 경로가 q-map에 나타난다.
- 3. 학습된 가중치를 사용해 학습하는 것이 훈련 시간을 단축 시키는데 유의미한 영향을 준다.
- 4. 학습된 가중치를 사용해 학습을 할수록 goal로 향하는 q-value값이 reward를 잘 받아온다. 다양한 경로를 통해 goal로 도착하는 q-map이 생성된다.
- 5. 학습을 반복할수록 noise가 발생한다. 이로 인해 기존에 가지고 있던 q-map이 일부 소실된다.
- 6. 여러번 반복 학습하는 것이 오히려 더 않좋은 성능을 보인다.