

## SARSA와 Q-learning의 Decaying 입실론

Episode가 충분히 많이 실행되어 큐함수가 많이 업데이트 되고 agent가 목표를 잘 찾아가게 되고, 탐험이 많이 이루어져 더 이상 가보지 않은 곳이 없게 되면, 더 이상 e-greedy에 의해 Random한 Action을 하는 것이 비효율적이게 된다. 따라서 iteration이 계속될수록 e 값을 계속해서 작게 만들어보았다.

```
class QLearningAgent:
    def __init__(self, actions):
        self.actions = actions
        self.step_size = 0.01
        self.discount_factor = 0.9
        self.epsilon = 0.1
        self.iteration = 1
        self.q_table = defaultdict(lambda: [0.0, 0.0, 0.0, 0.0])

class SARSAgent:
    def __init__(self, actions):
        self.actions = actions
        self.step_size = 0.01
        self.discount_factor = 0.9
        self.epsilon = 0.1
        self.iteration = 1 #시간이 지남을 알려주는 변수 생성
        # 0을 초기값으로 가지는 큐함수 테이블 생성
        self.q_table = defaultdict(lambda: [0.0, 0.0, 0.0, 0.0])
```

SARSA와 Qlearning 모두에 iteration이라는 value를 만들어준다

```
# 입실론 탐욕 정책에 따라서 행동을 반환
def get_action(self, state):
    new_epsilon = self.epsilon / self.iteration
    if np.random.rand() < new_epsilon:
        # 무작위 행동 반환
        action = np.random.choice(self.actions)
    else:
        # 아니면 큐함수에 따른 행동 반환
        state = str(state)
        q_list = self.q_table[state]
        action = arg_max(q_list)
    self.iteration += 1
    return action
```

두가지 모두 get\_action함수는 동일하고 iteration값으로 epsilon값을 나눠준다

매번 iteration값에 1씩 더해주어 점점 epsilon값이 작게 만들어준다.

또한 episode마다 거쳐온 state의 개수와 terminal state가 원이였는지 세모였는지를 구분해주기위해 final reward도 출력해보았다.

이제 실행결과를 비교해보자.

### 1-1. SARSA with no decaying epsilon

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help gridworld1 [C:\Users\Wokke\PycharmProjects\untitled6] - sarsaAgent.py - PyCharm
untitled6\sarsa\agent.py
gridworld1 C:\Users\Wokke\PycharmProjects\untitled6\sarsa\agent.py
agent.py
agent,decayingE.py
agent,v2.py
environment.py
qlearning\agent,decayingEpy
sarsa\agent.py
sarsa\agent,decayingEpy
sarsa\agent,v2.py
sarsa\environment.py
External Libraries
Scratches and Consoles
agent (1) agent
action = agent.get_action(state)
# reward 값을 담는 list 생성
reward_table = []
# return 값을 담는 list 생성
return_table = []
while True:
    env.render()
    # 행동을 위한 후 다음상태 보상 에피소드의 종료 여부를 받아옴
    next_state, reward, done = env.step(action)
    # 다른 상태에서의 다음 행동 선택
    next_action = agent.get_action(next_state)
    # <s,a,r,s',a'>로 큐함수를 업데이트
    agent.learn(state, action, reward, next_state, next_action)
    # reward_table에 reward들을 차근차근
    reward_table.append(reward)
    state = next_state
    action = next_action
if __name__ == "__main__":
    for episode in range(1,100):
        30 th episode's return vale, number of state: 6 final reward = 100
        31 th episode's return vale, number of state: 8 final reward = 100
        32 th episode's return vale, number of state: 7 final reward = 100
        33 th episode's return vale, number of state: 6 final reward = 100
        34 th episode's return vale, number of state: 8 final reward = 100
        35 th episode's return vale, number of state: 6 final reward = 100
        36 th episode's return vale, number of state: 8 final reward = 100
        37 th episode's return vale, number of state: 6 final reward = 100
        38 th episode's return vale, number of state: 7 final reward = 100
        39 th episode's return vale, number of state: 6 final reward = 100
        40 th episode's return vale, number of state: 6 final reward = 100
        41 th episode's return vale, number of state: 6 final reward = 100
        42 th episode's return vale, number of state: 6 final reward = 100
        43 th episode's return vale, number of state: 5 final reward = -100
        44 th episode's return vale, number of state: 79 final reward = 100
        45 th episode's return vale, number of state: 73 final reward = 100
        46 th episode's return vale, number of state: 131 final reward = -100
Run: agent (1) agent
Event Log
Connection to Python debugger failed: Interrupted function call: accept failed (today 오후 4:25)
85:107 LF UTF-8 4 spaces Python 3.8 오전 6:23 2020-10-10
```

한 40번째쯤 episode에서부터 잘 찾아가는가 싶더니

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help gridworld1 [C:\Users\Wokke\PycharmProjects\untitled6] - sarsaAgent.py - PyCharm
untitled6\sarsa\agent.py
gridworld1 C:\Users\Wokke\PycharmProjects\untitled6\sarsa\agent.py
agent.py
agent,decayingE.py
agent,v2.py
environment.py
qlearning\agent,decayingEpy
sarsa\agent.py
sarsa\agent,decayingEpy
sarsa\agent,v2.py
sarsa\environment.py
External Libraries
Scratches and Consoles
agent (1) agent
Run: agent (1) agent
Event Log
Connection to Python debugger failed: Interrupted function call: accept failed (today 오후 4:25)
85:107 LF UTF-8 4 spaces Python 3.8 오전 6:24 2020-10-10
```

SARSA의 고질적인 문제점인 간힘현상이 일어났다. 더 이상 움직이지 않는 지경에 이르렀다.

## 1-2. SARSA with decaying epsilon

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** untitled6 [gridworld1]
- Code Editor:** agent\_decayingE.py (highlighted)
- Terminal:** Shows training logs from episode 10 to 24, indicating final rewards of 100 for most episodes.
- SARSA Visualization:** A grid-based heatmap showing Q-table values. The grid has 5 columns and 5 rows. Colored cells represent non-zero values: red for -1.0, green for 0.0, blue for 1.99, and yellow for 26.77.
- Status Bar:** PEP 8: no newline at end of file, 90:107 LF, UTF-8, 4 spaces, Python 3.8, 2020-10-10, 6:28 오전

약 15번째 episode부터 최적의 경로로 원에 도착하는 것을 확인 할 수 있다.

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** untitled6 [gridworld1]
- Code Editor:** agent\_decayingE.py (highlighted)
- Terminal:** Shows training logs from episode 87 to 100, with all final rewards reaching 100.
- SARSA Visualization:** A grid-based heatmap showing Q-table values. The grid has 5 columns and 5 rows. Colored cells represent non-zero values: red for -1.0, green for 0.0, blue for 1.99, and yellow for 60.729.
- Status Bar:** PEP 8: no newline at end of file, 90:107 LF, UTF-8, 4 spaces, Python 3.8, 2020-10-10, 6:29 오전

위에  $\epsilon$ 을 안 줄여줬을 때와 달리 episode가 계속 진행됨에도 간힘 현상 또한 일어나지 않았다. 이유는 간힘 현상이란 agent가 특정상황에서 random한 행동에 의해서 - reward를 가지게 되었을 경우 이전 state의 큐함수가 -값을 가지는 경우에 잘못 판단하게 되어 이런 현상이 일어나게 되는데  $\epsilon$ 값을 iteration 할때마다 줄여주는 경우 episode가 진행됨에 따라 agent가 random하게 행동하지 않고 점점 더 greedy하게 행동하게 됨으로 아주 운이 나쁜 경우를 제외하고 계속 최적의 경로로 도착하게 된다.

## 2-1. Q-learning with no decaying epsilon

The screenshot shows the PyCharm IDE interface with a project named 'gridworld1'. The code editor displays 'agent\_decayingE.py' with the following content:

```

import numpy as np
import random
from qlearning.environment import Env #이 부분을 교수님 파일에 맞게 수정해주세요.
from collections import defaultdict

class QLearningAgent:
    def __init__(self, actions):
        self.actions = actions
        self.step_size = 0.01
        self.discount_factor = 0.9
        self.epsilon = 0.1
        self.q_table = defaultdict(lambda: [0.0, 0.0, 0.0, 0.0])

    def learn(self, state, action, reward, next_state):
        state, next_state = str(state), str(next_state)
        q_1 = self.q_table[state][action]
        # 뱀만 최적 방정식을 사용한 큐함수의 업데이트
        if reward >= 0:
            q_2 = self.q_table[next_state][action]
            self.q_table[state][action] = q_1 + self.step_size * (reward + self.discount_factor * q_2 - q_1)
        else:
            self.q_table[state][action] = q_1 - self.step_size * (reward - q_1)

    if __name__ == "__main__":
        for episode in range(1,1000):
            for i in range(5):
                ...

```

The bottom right of the screen shows the date and time: 2020-10-10 오후 6:40.

약 19번째 episode부터 최적의 경로를 슬슬 찾아가기 시작한다.

The screenshot shows the PyCharm IDE interface. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and a tab for the current file: gridworld1 [C:\Users\fokke\PycharmProjects\untitled6] - ...#qlearning\agent.py - PyCharm.

The left sidebar displays the project structure under 'untitled6 [gridworld1]'. It contains several files: img, pi, qlearning (with agent.py, agent\_decayingE.py, agent\_v2.py, environment.py), sarsa (with agent.py, agent\_decayingE.py, agent\_v2.py, environment.py), and vi (with pi.zip and vi.zip).

The main editor window shows the code for 'agent\_decayingE.py'. The code defines a QLearningAgent class with methods \_\_init\_\_ and learn. The learn method takes state, action, reward, and next\_state as arguments, updates the q-table, and prints the episode's return value and final reward.

The 'Run' tab at the bottom indicates the script is running ('agent\_decayingE'). The output window shows the results of 92 episodes:

```

79 th episode's return vale, number of state: 6 final reward = 100
80 th episode's return vale, number of state: 6 final reward = 100
81 th episode's return vale, number of state: 8 final reward = 100
82 th episode's return vale, number of state: 11 final reward = 100
83 th episode's return vale, number of state: 6 final reward = 100
84 th episode's return vale, number of state: 6 final reward = 100
85 th episode's return vale, number of state: 6 final reward = 100
86 th episode's return vale, number of state: 6 final reward = 100
87 th episode's return vale, number of state: 6 final reward = 100
88 th episode's return vale, number of state: 7 final reward = 100
89 th episode's return vale, number of state: 6 final reward = 100
90 th episode's return vale, number of state: 6 final reward = 100
91 th episode's return vale, number of state: 8 final reward = 100
92 th episode's return vale, number of state: 6 final reward = 100

```

The status bar at the bottom right shows the date (2020-10-10) and time (9:41).

하지만 계속해서 높은  $\epsilon$ 값 때문에 탐험을 하여 가끔씩 오래 걸리거나 – reward를 가지는 곳으로 가기도 하는 것을 관찰할 수 있다.

## 2-2. Q-learning with decaying epsilon

This screenshot shows the same PyCharm setup as the previous one, but with a different script: 'agent\_decayingE.py'.

The code for 'agent\_decayingE.py' is identical to the one in the previous screenshot, except for the addition of a self.iteration variable in the \_\_init\_\_ method. The learn method also includes a print statement for each iteration.

The output window shows the results of 15 episodes:

```

1 th episode's return vale, number of state: 25 final reward = -100
2 th episode's return vale, number of state: 3 final reward = -100
3 th episode's return vale, number of state: 12 final reward = -100
4 th episode's return vale, number of state: 10 final reward = -100
5 th episode's return vale, number of state: 19 final reward = -100
6 th episode's return vale, number of state: 51 final reward = -100
7 th episode's return vale, number of state: 243 final reward = 100
8 th episode's return vale, number of state: 9 final reward = 100
9 th episode's return vale, number of state: 69 final reward = 100
10 th episode's return vale, number of state: 11 final reward = 100
11 th episode's return vale, number of state: 6 final reward = 100
12 th episode's return vale, number of state: 7 final reward = 100
13 th episode's return vale, number of state: 8 final reward = 100
14 th episode's return vale, number of state: 6 final reward = 100
15 th episode's return vale, number of state: 6 final reward = 100

```

The status bar at the bottom right shows the date (2020-10-10) and time (7:27).

Decaying epsilon을 한 경우에도 최적의 경로를 약 15번째쯤부터 찾아가기 시작한다

The screenshot shows the PyCharm IDE interface with the following details:

- Project:** untitled6 [gridworld1] C:\Users\fokke\PycharmProjects\untitled6\ - qlearning\agent\_decayingEpy - PyCharm
- Code Editor:** agent\_decayingEpy (1) showing the following Python code:

```
import numpy as np
import random
from qlearning.environment import Env
from collections import defaultdict

class QLearningAgent:
    def __init__(self, actions):
        self.actions = actions
        self.step_size = 0.01
        self.discount_factor = 0.9
        self.epsilon = 0.1
        self.iteration = 1
        self.q_table = defaultdict(lambda: [0.0, 0.0, 0.0, 0.0])

    # <s, a, r, s'> 샘플로부터 큐함수 업데이트
    def learn(self, state, action, reward, next_state):
        state, next_state = str(state), str(next_state)
        q_1 = self.q_table[state][action]
        q_2 = max(self.q_table[next_state].values())
        if __name__ == '__main__':
            for episode in range(1, 1000):
                for i in range(100):
```

- Run:** agent\_decayingEpy (1)
- Output:** Shows the terminal output of the script running for 100 episodes, with final rewards consistently at 100.
- Bottom Bar:** Includes icons for Run, Debug, Terminal, Python Console, and Event Log. Status bar shows "77:27 LF UTF-8 4 spaces Python 3.8" and "오늘 6:45 2020-10-10".

하지만 위와 다르게 100번째의 iteration시에 완전히  $\epsilon$  값이 0에 가까운 값이 됨으로 계속해서 최적의 경로를 택해 episode를 수행하게 된다.

이와 같이  $\epsilon$ 을 계속해서 줄여줌으로서 많은 iteration 이후에는 agent가 최적의 경로에 따라 행동한다는 것을 알 수 있다. 하지만 만약  $\epsilon$ 를 너무 빠르게 줄이게 된다면 explore를 너무 안하여 map 전체를 탐색하지 않는 경우가 생기기 때문에 적절하게 줄여주는 것이 좋은 방향이라고 생각한다.