**Homework 3**

2018324133 김태우

이 문제의 해결을 위해 ε-greedy Sarsa를 사용하여 매 episode에서의 Q value의 Temporal Diffrence를 구하는 방법을 사용하였다.

Python과 numpy 라이브러리를 사용하였다.

|  |
| --- |
| # Set Constant  ALPHA = 0.5  EPSILON = 0.1  # Set Wind  wind = np.zeros(10, dtype=np.int32)  wind[3] = 1  wind[4] = 1  wind[5] = 1  wind[6] = 2  wind[7] = 2  wind[8] = 1  # Set Initial Q value (x, y, action)  Q = np.zeros((10, 7, 4)) |

우선 상수들과 windy grid, Q value의 초기값을 정의하였다.

Q value는 모든 state와 action에 대하여 0으로 초기화하였다.

|  |
| --- |
| def NextPosition(x, y, action) :  nx = x  ny = y - wind[x]    if action == 0:  ny -= 1  elif action == 1:  ny += 1  elif action == 2:  nx += 1  elif action == 3:  nx -= 1    if nx < 0:  nx = 0  elif nx > 9:  nx = 9    if ny < 0:  ny = 0  elif ny > 6:  ny = 6    return nx, ny |

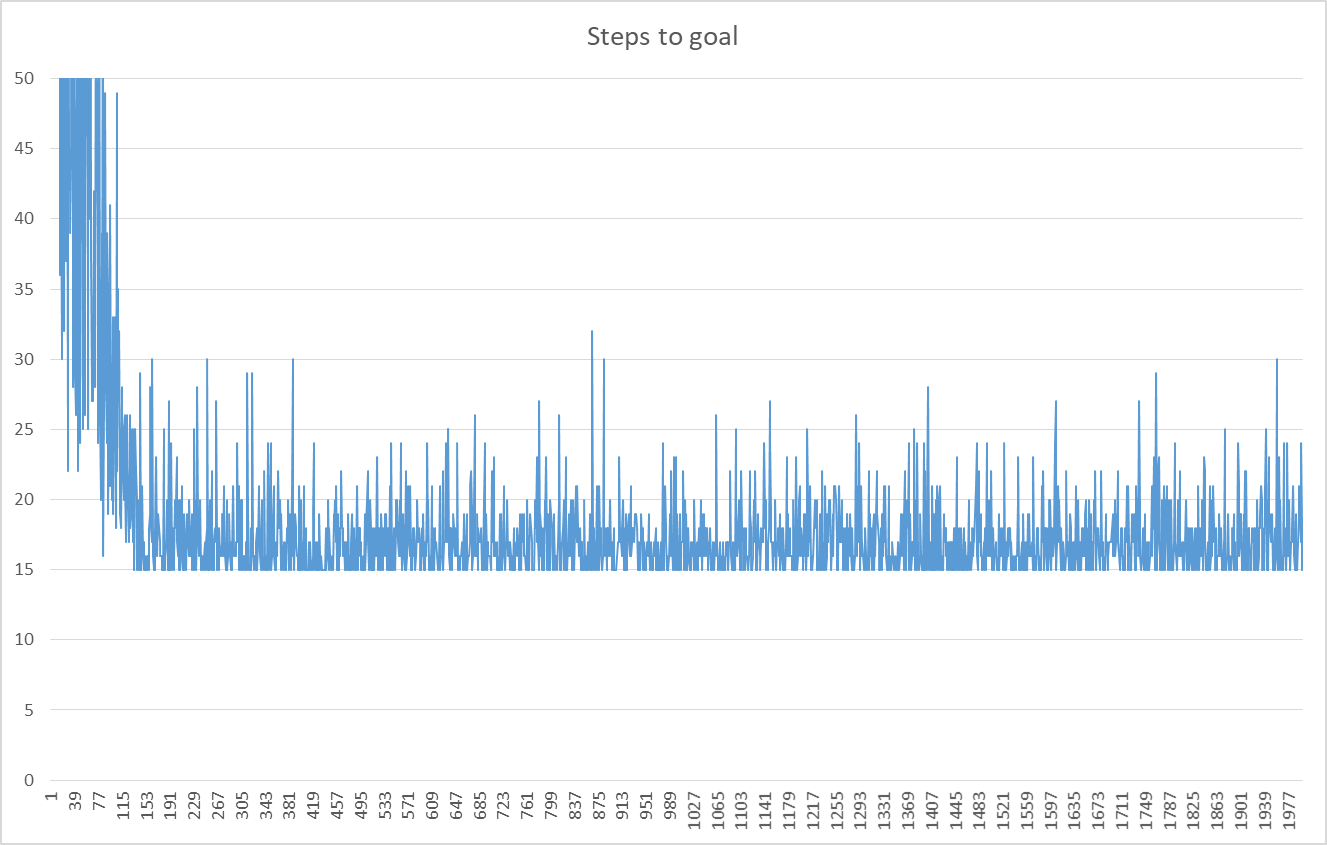
트레이닝에 앞서 다음 state의 x, y를 쉽게 구하기 위한 NextPosition 함수를 정의하였다.

NextPosition 함수는 현재 위치와 선택한 action을 인자로 받아, action과 바람 세기, 그리고 벽의 유무에 따른 다음 위치를 계산하여 반환한다.

|  |
| --- |
| for iteration in range(10000) :  # Reset Player  player\_x = 0  player\_y = 3  step = 0  while True:  r = random.random()  if r < EPSILON :  action = random.randrange(0, 4)  else :  action = np.argmax(Q[player\_x][player\_y]);    # Get Next Position  player\_next\_x, player\_next\_y = NextPosition(player\_x, player\_y, action)    # If reached goal, then reward is zeros  # else reward is -1  if (player\_next\_x == 7 and player\_next\_y == 3):  reward = 0  else :  reward = -1    # Update Q Value  Q[player\_x][player\_y][action] += ALPHA \* \  (reward + np.max(Q[player\_next\_x][player\_next\_y]) - Q[player\_x][player\_y][action])    # If reached goal or maximum step (100) reached, exit the loop  if (player\_next\_x == 7 and player\_next\_y == 3) :  break  elif step == 100 :  step = -1  break  else :  player\_x = player\_next\_x  player\_y = player\_next\_y  step += 1    if step >= 0:  print("It " + str(iteration) + " : Goal at " + str(step))  f\_res.write("It " + str(iteration) + " : Goal at " + str(step) + "\n")  else :  print("It " + str(iteration) + " : Terminated")  f\_res.write("It " + str(iteration) + " : Terminated\n") |

이제 10000회의 게임을 통해 reward를 구하고 Q value를 update한다.

Epsilon과 alpha는 문제에서 제공되었고 한 episode의 maximum step은 100회로, episode 개수는 10000개로 제한하였다.



2000회의 episode 동안 goal 에 도달하기까지의 step 수이다.

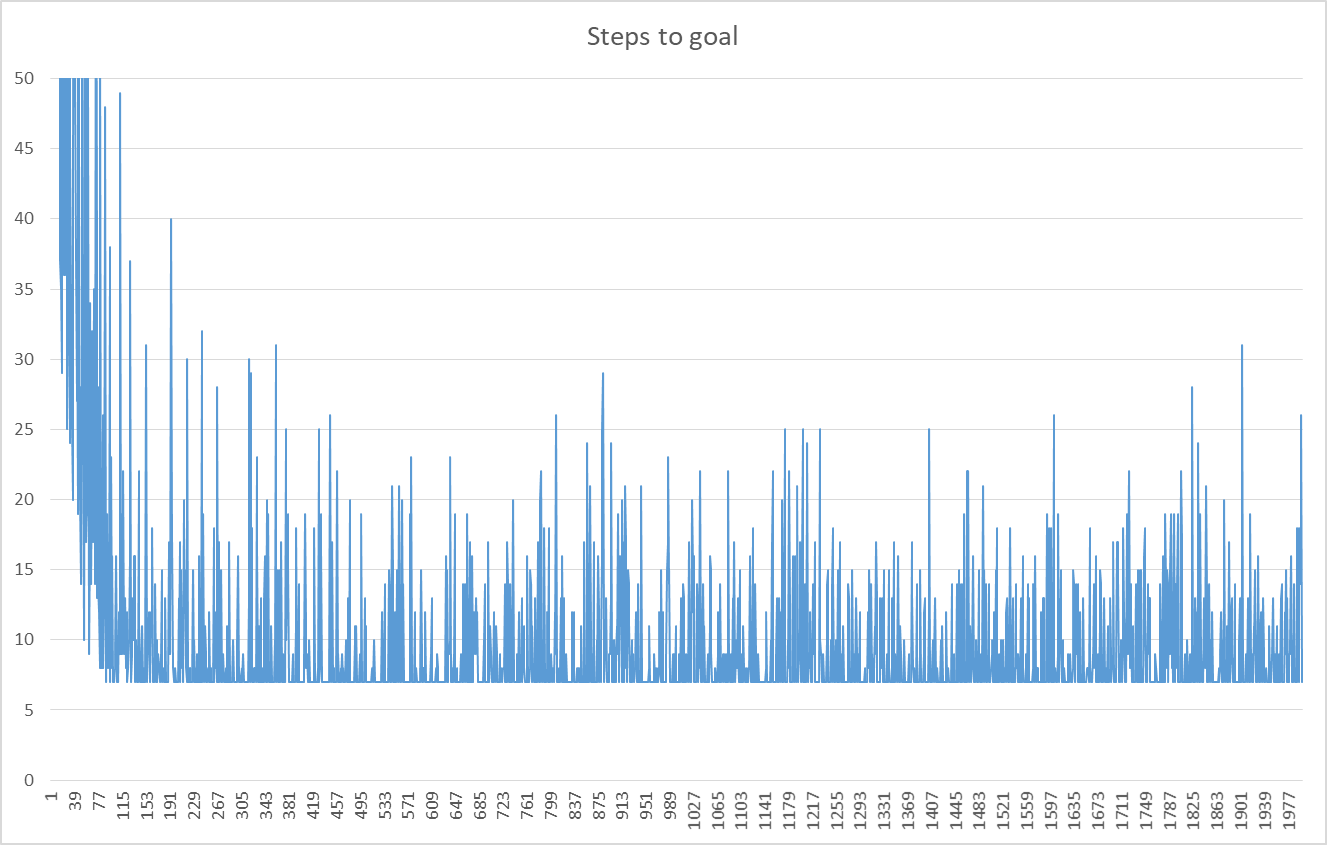
빠르게 수렴하기 시작하여 약 200회부터 이미 최저 step 수인 15회에 가깝게 수렴했음을 알 수 있다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ▼ | ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▼ |
| ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▼ |
| ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▼ | ◀ | ▼ |
| ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▶ |  | ◀ | ▼ |
| ▶ | ▶ | ▶ | ▶ | ▶ | ▶ | ▲ | ▼ | ◀ | ◀ |
| ▶ | ▶ | ▶ | ▶ | ▶ | ▲ | ▲ | ▼ | ◀ | ▲ |
| ▶ | ▶ | ▶ | ▶ | ▲ | ▲ | ▲ | ▲ | ▲ | ◀ |
| 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 1 | 0 |

문제의 예시와 같은 policy로 수렴했음을 확인할 수 있다.

|  |
| --- |
| def NextPosition(x, y, action) :  ...  if action == 0:  ny -= 1  elif action == 1:  ny += 1  elif action == 2:  nx += 1  elif action == 3:  nx -= 1  elif action == 4:  nx += 1  ny -= 1  elif action == 5:  nx += 1  ny += 1  elif action == 6:  nx -= 1  ny -= 1  elif action == 7:  nx -= 1  ny += 1  ... |

다음으로 대각선 방향까지 고려하여 8개의 action을 가진 player로 learning을 돌려 보았다.

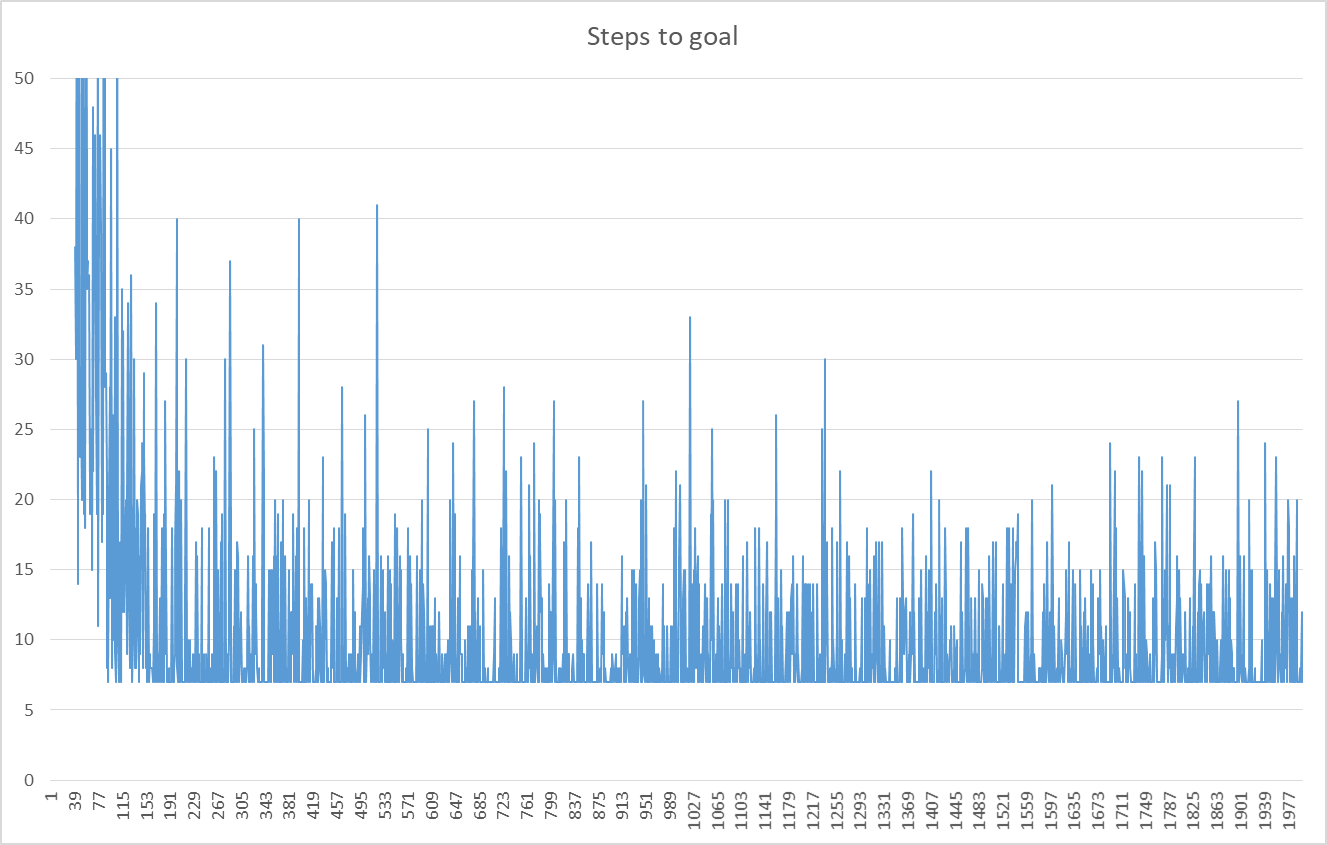


Goal까지의 최소 action 수가 7회로 줄었다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ↙ | ↑ | ↙ | ↙ | ↘ | ↘ | → | → | → | ↓ |
| ↘ | ↘ | ↓ | ↙ | ↘ | → | → | → | ↘ | ↓ |
| → | ↘ | ↓ | ↙ | ↙ | → | → | ↘ | ↘ | ↙ |
| → | → | ↘ | ↙ | ↙ | ↘ | ↘ |  | ↙ | ↙ |
| → | → | → | ↘ | ↘ | ↘ | ↘ | ↑ | ← | ↖ |
| ↘ | → | → | → | → | → | → | ↓ | ← | ↘ |
| → | ↗ | → | → | ↘ | ↗ | ↗ | ↑ | ↑ | ↓ |
| 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 1 | 0 |

Policy를 보면 돌지 않고도 바로 goal로 향할 수 있는 길이 있음을 확인할 수 있다.

마지막으로 멈춰있는 action까지 포함하여 9개의 action으로 learning을 돌려 보았다.



최소 action 수는 같다. 유의미하게 변화된 부분은 보이지 않았다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ← | ↙ | ↓ | ↖ | → | → | → | → | → | ↓ |
| ↘ | ↘ | ↓ | ↙ | ↘ | ↘ | → | → | ↘ | ↓ |
| → | ↘ | ↓ | ↙ | ↙ | → | → | ↘ | ↘ | ↙ |
| → | → | ↘ | ↙ | ↙ | ↘ | ↘ |  | ↙ | ← |
| → | → | → | ↘ | ↘ | ↘ | ↘ | ↓ | ← | ← |
| ↘ | → | → | → | → | → | → | ↓ | ↗ | ↙ |
| ↗ | ↗ | → | → | → | ↘ | ↗ | ↑ | ↑ | ↗ |
| 0 | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 1 | 0 |

Policy 역시 방문을 잘 하지 않는 구석자리를 제외하고는 큰 변화를 보이지 않았다.