

ML6 Reinforcement Learning

October 27, 2023

```
[1]: import numpy as np

maze = np.array([
    [0, 0, 0, 0, 0],
    [0, 1, 0, 1, 0],
    [0, 0, 0, 0, 0],
    [0, 1, 1, 1, 0],
    [0, 0, 0, 0, 2] # 2 is the goal
])

learning_rate = 0.1
discount_factor = 0.9
epsilon = 0.1
num_episodes = 1000

num_states, num_actions = maze.size, 4
Q = np.zeros((num_states, num_actions))

for _ in range(num_episodes):
    state = 0 # Starting position

    while True:
        action = np.random.choice(num_actions) if np.random.uniform(0, 1) < epsilon else np.argmax(Q[state, :])
        new_state = state + [0,1,2,3][action] # Up, Down, Left, Right
        reward = [-1, 1, 0][maze.flat[new_state]]
        if reward: break
        state = new_state

current_state = 0
while current_state != 16: # Goal state
    action = np.argmax(Q[current_state, :])
    current_state = current_state + (action + 1)
    print("Agent moved to state:", current_state)
```

```
Agent moved to state: 1
Agent moved to state: 2
Agent moved to state: 3
```

Agent moved to state: 4
Agent moved to state: 5
Agent moved to state: 6
Agent moved to state: 7
Agent moved to state: 8
Agent moved to state: 9
Agent moved to state: 10
Agent moved to state: 11
Agent moved to state: 12
Agent moved to state: 13
Agent moved to state: 14
Agent moved to state: 15
Agent moved to state: 16

[]: