

Week 9: Introduction to Reinforcement Learning

INFO7001 Advanced Machine Learning

Implementing a small environment for RL

1. Implement a (deterministic) transition function

```
def transition(state, action):
    # return if state is terminal
    if state == 9 or state == 14:
        return state

    # actions: 0 left, 1 right, 2 up, 3 down
    # left
    if action == 0:
        if state == 0 or state == 5 or state == 10 or state == 8:
            newstate = state      # do nothing
        else:
            newstate = state - 1
    # right
    elif action == 1:
        if state == 4 or state == 5:
            newstate = state      # do nothing
        else:
            newstate = state + 1
    # up
    elif action == 2:
        if state == 1 or state == 2 or state > 9:
            newstate = state      # do nothing
        else:
            newstate = state + 5
    # down (any action >= 3 is treated as 'down')
    else:
        if state == 11 or state == 12 or state < 5:
            newstate = state
        else:
            newstate = state - 5

    return newstate
```

2. Implement a reward function

```
def reward(state, action, newstate):
    # return immediate reward
    r = 0
    if newstate == 14:
        r = 1
    elif newstate == 9:
        r = -1
    return r
```

3. Implement a step function

```
def step(state, action):
    # return tuple (newstate, reward, done)
    newstate = transition(state, action)
    r = reward(state, action, newstate)
    done = r != 0 # in this environment, we're done once we get a non-0 reward
    return (newstate, r, done)
```

4. Implement a function that randomly selects an action, implement reset

```
def sampleaction():
    # return a random action
    action = random.randint(0, 3)
    return action

def reset():
    # return the start state
    # (this is the function - it returns a tuple (0, 0, False))
    return (0, 0, False)
```

5. Implement a function that renders the current state of the environment

```
def render(state):
    # render the state of the environment
    print('#'*19+'##')
    for i in range(2,-1,-1):
        s = '#'
        w = '#' if i == 0 else '-'
        for j in range(5):
            pos = i*5 + j
            if pos == state:
                c = ' * '
```

```

        elif pos == 6 or pos == 7:
            c = '###'
        else:
            c = '   '
        s = s + c + '|' if j < 4 else s + c
    s += '#'
    print(s)
    print('#'+w*19+'#')

```

6. Putting it all together

```

stepcounter = 0
summedreward = 0
for i in range(10):
    print(i)
    state, r, done = reset()
    while not done:
        action = sampleaction()
        newstate, r, done = step(state, action)
        stepcounter = stepcounter + 1
        state = newstate
        render(state)
    summedreward += r

print()
print(f'Average number of steps per run {stepcounter/10}')
print(f'Total reward over 10 runs: {summedreward}')

```