Homework 1

Exercise 3.1

Devise three example tasks of your own that fit into the MDP framework, identifying for each its states, actions, and rewards. Make the three examples as different from each other as possible. The framework is abstract and flexible and can be applied in many ways. Stretch its limits in some way in at least one of your examples.

1. A conventional supervised image classification problem:

```
States: the image feed to the network.
Actions: the predicted class
Rewards: loss calculated by a certain loss functions
Limitation: the model may be over trained (overfitting)
```

2. Playing cheese

```
States: the actual situation on the game board
Actions: move pieces on the board in accordance with the rules of the game on your turn
Rewards: +1/-1 for win and loss
```

3. An airconditioner that will adjust the temp to a certain degree

```
States: the actual temperature in room
Actions: Increase or decrease the temperature based on the current temperature and the target temperature
Rewards: +1/-1 for lower temp and higher temp
```

Exercise 3.3

Consider the problem of driving. You could define the actions in terms of the accelerator, steering wheel, and brake, that is, where your body meets the machine. Or you could define them farther out-say, where the rubber meets the road, considering your actions to be tire torques. Or you could define them farther in-say, where your brain meets your body, the actions being muscle twitches to control your limbs. Or you could go to a really high level and say that your actions are your choices of where to drive. What is the right level, the right place to draw the line between agent and environment? On what basis is one location of the line to be preferred over another? Is there any fundamental reason for preferring one location over another, or is it a free choice?

Answer:

There is no right level to draw the line between the agent and environment.

The driving problem itself is a complex task composed of multiple layers of RL problems. The line needs to be drawn in terms of what we want to achieve.

A higher level line, such as where your brain meets your body, lead to less computation efficiency. Low level lines, however, may lead to the loss of some details, like the state change of the direction during driving, resulting in some safety issues.

Exercise 3.4

Give a table analogous to that in Example 3.3, but for $p(s', r|s, a)$. It should have columns for s, a, s', r, and $p(s', r|s, a)$, and a row for every 4-tuple for which $p(s', r|s, a) > 0$.

Answer:

The rewards have no probability distribution, which indicates that $p(s'|s, a) = p(s', r|s, a)$

s	a	s'	r	$p(s', r\|s, a)$
$high$	$search$	$high$	$r_{research}$	α
$high$	$search$	low	$r_{research}$	$1-\alpha$
low	$search$	$high$	-3	$1-\beta$
low	$search$	low	$r_{research}$	β
$high$	$wait$	$high$	r_{wait}	1
$high$	$wait$	low	$-$	0
low	$wait$	$high$	$-$	0
low	$wait$	low	r_{wait}	1
low	$recharge$	$high$	0	1
low	$recharge$	low	$-$	0

Investing in Stocks

Let's say you start off at day one, with two stocks S_1 and S_2 to choose to invest in. The return of stock one is uniformly distributed (as a percentage) between [-1,1]. The return of stock two is uniformly distributed (as a percentage) between [-2,8]. Assume you start off on day one with 100 dollars.

Part 1

Write a simulation that runs for one hundred iterations, and randomly invests in stock one with a fifty percent chance, and in stock two with a fifty percent chance. What is your final return on day 100, assuming you start day 1 with 100 dollars? Please attach your code to this assignment.

In [7]:
```
import random

def choose_stock():
    indices = random.randint(0,1)
    if indices == 0:
        return stock_s1()
    else:
        return stock_s2()


def stock_s1():
    return random.randint(-1,1)/100

def stock_s2():
    return random.randint(-2,8)/100

funding = 100
for day in range(100):
    funding = funding*(1+choose_stock())
print(f"the final return on day 100 is: {funding}")
```

```
the final return on day 100 is: 526.2378803121916
```

Part 2

In this problem, what is the state variable(s)? What is the action variable(s)? What are the possible states and actions? What are the possible rewards each day?

Answer:

```
States: which stock we choose
Actions: the principal changes over time
Rewards: the return of the chosen stock
```

Part 3

If you wanted to maximize your mean return, which stock is better to invest in? Hint: The solution is very short, you could write it in a sentence or two.

Answer:

The stock 2 is better to invest in, since it has a higher return expectation E_{return}