Theory Section: Reinforcement Learning

Question 1 – Value Function

Setup

- **Episode length:** 3 words
- **Vocabulary:** {I, like, pizza} → 3 possible words
- **Reward:**
  - Only the exact sequence **"I like pizza"** gives **+10**
  - All other 3-word sequences give **0**
- **Policy:** uniformly random → each word chosen with probability **1/3**
- **Discount:** $\gamma = 1$
- **Reward is given only at the final (third) word**

Because the reward is only at the end and $\gamma = 1$:

$$V(s) = \mathbb{E}[R_t \mid s_t = s] = 10 \cdot \Pr(\text{ending at "I like pizza"} \mid s)$$

1. List all states of length 1 and 2

Length-1 states (3):

- [I]
- [like]
- [pizza]

Length-2 states (3×3 = 9):

- [I, I]
- [I, like]
- [I, pizza]
- [like, I]
- [like, like]
- [like, pizza]
- [pizza, I]
- [pizza, like]
- [pizza, pizza]

2. Number of terminal states and how many give non-zero reward

A terminal state is a full 3-word sequence.

Total possible sequences:

$$3^3 = 27$$

Only **one** gives reward ≠ 0:

- **I like pizza → +10**

Thus:

- **Total terminal states:** 27
- **Terminal states with non-zero reward: 1**

3. Compute the value function $V(s)$

Recall:

$$V(s) = 10 \cdot \Pr(\text{success} \mid s)$$

Where success = generating "I like pizza".

(c) State

$$s_2 = [I, like]$$

Only one word left. Policy chooses uniformly among {I, like, pizza}.

To complete correctly, the last word must be **pizza**:

$$\Pr(\text{success} \mid s_2) = \frac{1}{3}$$
$$V(s_2) = 10 \cdot \frac{1}{3} = \frac{10}{3} \approx 3.33$$

(d) State

$$s_3 = [I, pizza]$$

This sequence can **never** become "I like pizza", because the second word is already incorrect.

$$\Pr(\text{success} \mid s_3) = 0$$
$$V(s_3) = 10 \cdot 0 = 0$$

(b) State

$$s_1 = [I]$$

Second word choices (each with prob. 1/3):

- like → leads to state s_2 that can still succeed
- I → impossible to recover
- pizza → impossible to recover

Thus:

$$\Pr(\text{success} \mid s_1) = \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9}$$
$$V(s_1) = 10 \cdot \frac{1}{9} = \frac{10}{9} \approx 1.11$$

(a) Initial empty state

$$s_0 = []$$

First word must be **I** for any chance of success:

$$\frac{1}{3}\ (\text{first step}) \times \frac{1}{9}\ (\text{success from } s_1) = \frac{1}{27}$$
$$V(s_0) = 10 \cdot \frac{1}{27} = \frac{10}{27} \approx 0.37$$

Summary of Value Function

State	Value
$s_0 = []$	$V = 10/27 \approx 0.37$
$s_1 = [I]$	$V = 10/9 \approx 1.11$
$s_2 = [I, like]$	$V = 10/3 \approx 3.33$
$s_3 = [I, pizza]$	$V = 0$

Question 2 – Q-Learning Update

Same environment as before.

Current state:

$$s = [I]$$

Q-values:

Action	Q([I], action)
I	1.0
like	1.0
pizza	0.5

We take the action:

$$a = \text{“like”}$$

Next state:

$$s' = [I, like]$$

Q-values for next state:

- Q([I,like], I) = 1.0
- Q([I,like], like) = 0.5
- Q([I,like], pizza) = 2.0

Learning rate:

$$\alpha = 0.5$$

Discount:

$$\gamma = 1$$

Immediate reward:

$$r = 0$$

(because reward only at end)

1. Q-learning update rule

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

2. Compute the updated Q([I], like)

First compute TD error:

$$\delta = r + \gamma \max_{a'} Q(s',a') - Q(s,a)$$
$$= 0 + 1 \cdot 2.0 - 1.0 = 1.0$$

Apply learning rate α:

$$Q_{\text{new}}([I],\text{like}) = 1.0 + 0.5 \cdot 1.0 = 1.5$$

Final Answer

$$\boxed{Q([I],\text{like})_{\text{updated}} = 1.5}$$