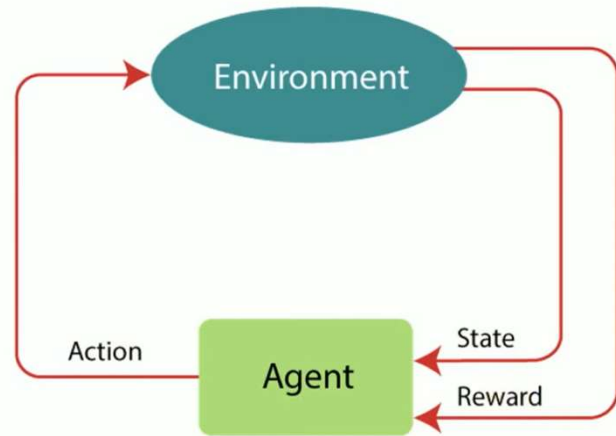


Introduction to Reinforcement Learning

- Reinforcement Learning is a feedback-based Machine learning Approach here an agent learns to which actions to perform by looking at the environment and the results of actions.
- For each correct action, the agent gets positive feedback, and for each incorrect action, the agent gets negative feedback or penalty.



1

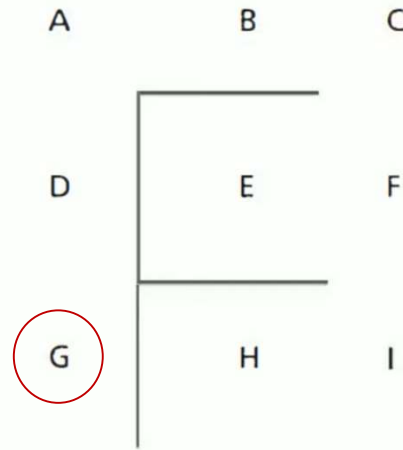
Key components:

- 1.Agent:** The learner or decision-maker that interacts with the environment.
- 2.Environment:** Everything that the agent interacts with and receives feedback from.
- 3.State:** A representation of the environment at a given time.
- 4.Action:** The choices the agent can make.
- 5.Reward:** Feedback from the environment after each action, indicating how good or bad the action was.
- 6.Policy:** The strategy that the agent uses to determine its actions based on the current state.

2

Introduction to Reinforcement Learning

- Reinforcement Learning is a feedback-based Machine learning Approach here an agent learns to which actions to perform by looking at the environment and the results of actions.
- For each correct action, the agent gets positive feedback, and for each incorrect action, the agent gets negative feedback or penalty.



3

Introduction to Reinforcement Learning

- The agent interacts with the environment and identifies the possible actions he can perform.
- The primary goal of an agent in reinforcement learning is to perform actions by looking at the environment and get the maximum positive rewards.
- In Reinforcement Learning, the agent learns automatically using feedbacks without any labeled data, unlike supervised learning.
- Since there is no labeled data, so the agent is bound to learn by its experience only.
- Reinforcement Learning is used to solve specific type of problem where decision making is sequential, and the goal is long-term, such as game-playing, robotics, etc.

4

Introduction to Reinforcement Learning

- There are two types of reinforcement learning - **positive and negative**.
- **Positive reinforcement learning** is a recurrence of behaviour due to positive rewards.
- Rewards increase strength and the frequency of a specific behaviour.
- This encourages to execute similar actions that yield maximum reward.
- Similarly, in **negative reinforcement learning**, negative rewards are used as a deterrent to weaken the behaviour and to avoid it.
- Rewards decreases strength and the frequency of a specific behaviour.

5

Introduction to Reinforcement Learning

- There are two types of reinforcement learning - **positive and negative**.
- **Positive reinforcement learning** is a recurrence of behaviour due to positive rewards.
- Rewards increase strength and the frequency of a specific behaviour.
- This encourages to execute similar actions that yield maximum reward.
- Similarly, in **negative reinforcement learning**, negative rewards are used as a deterrent to weaken the behaviour and to avoid it.
- Rewards decreases strength and the frequency of a specific behaviour.

6

Scope of Reinforcement Learning

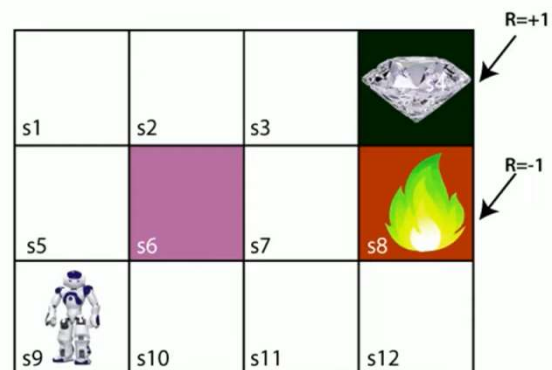
- Reinforcement is not suitable for environments where complete information is available.
- For example, the problems like object detection, face recognition, fraud detection can be better solved using a classifier than by reinforcement learning.

7

Characteristics of Reinforcement Learning

Sequential decision making –

- From figure, it can be seen the path from start to goal is not done in one step.
- It is a sequence of decisions that leads to the goal.
- One wrong move may result in a failure.



8

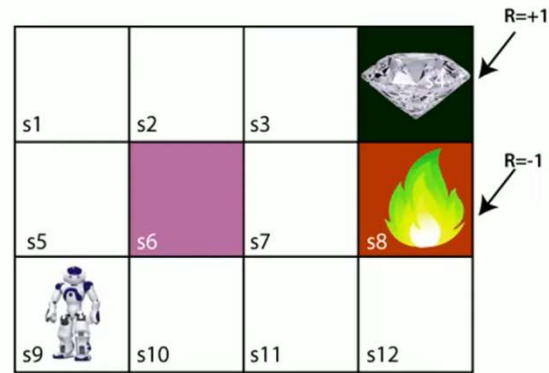
Characteristics of Reinforcement Learning

Delayed feedback –

- Often, rewards are not immediate.
- One must spend many moves to get final success or failure.
- Feedback in terms of reward is often delayed.

Time related –

- All actions are associated with time stamps inherently as all actions are ordered as per the timeline inherently.



9

The Bellman Equation

- The Bellman equation can be written as:





$$V(s) = \max[R(s, a) + \gamma V(s')]$$

- Where,
- $V(s)$ = value calculated at a particular point.
- $R(s, a)$ = Reward at a particular state s by performing an action a .
- γ = Discount factor
- $V(s')$ = The value at the previous state.
- In the above equation, we are taking the max of the complete values because the agent tries to find the optimal solution always.

10

The Bellman Equation





- So now, using the Bellman equation, we will find value at each state of the given environment.
- We will start from the block, which is next to the target block.
- **For S3 block:**
- $V(s_3) = \max[R(s, a) + \gamma V(s')]$,
- here $V(s') = 0$ because there is no further state to move.
- $V(s_3) = \max[R(s, a)] \Rightarrow V(s_3) = \max[1] \Rightarrow V(s_3) = 1.$

V=0.81 s1	V=0.9 s2	V=1 s3	 s4
V=0.73 s5	 s6	s7	 s8
 V=0.66 s9	s10	s11	s12

11

The Bellman Equation





- **For S2 block:**
- $V(s_2) = \max[R(s, a) + \gamma V(s')]$,
- here $\gamma = 0.9, V(s') = 1$, and $R(s, a) = 0$, because there is no reward at this state.
- $V(s_2) = \max[0.9(1)] \Rightarrow V(s_2) = \max[0.9] \Rightarrow V(s_2) = 0.9$

V=0.81 s1	V=0.9 s2	V=1 s3	 s4
V=0.73 s5	 s6	s7	 s8
 V=0.66 s9	s10	s11	s12

12

- **For S9 block:**

- $V(s_9) = \max[R(s, a) + \gamma V(s')]$,
- here $\gamma = 0.9$ (lets), $V(s') = 0.73$, and $R(s, a) = 0$, because there is no reward at this state also.
- $V(s_9) = \max[0.9(0.73)] \Rightarrow V(s_9) = \max[0.66] \Rightarrow V(s_4) = 0.66$





V=0.81 s1	V=0.9 s2	V=1 s3	 s4
V=0.73 s5	 s6	s7	 s8
 V=0.66 s9	s10	s11	s12

13

The Bellman Equation

- **For S7 block:**

- $V(s_7) = \max[R(s, a) + \gamma V(s')]$,
- here $\gamma = 0.9$, $V(s') = 1$ and $R(s, a) = 0$, because there is no reward at this state also.
- $V(s_7) = \max[0.9(1)] \Rightarrow V(s_7) = \max[0.9] \Rightarrow V(s_7) = 0.9$






V=0.81 s1	V=0.9 s2	V=1 s3	 s4
V=0.73 s5	 s6	V=0.9 s7	 s8
 V=0.66 s9	V=0.73 s10	V=0.81 s11	V=0.73 s12

14








The Bellman Equation








- For S12 block:

- $V(s_{12}) = \max[R(s, a) + \gamma V(s')]$,
- here $\gamma = 0.9$, $V(s') = 0.81$ and $R(s, a) = 0$, because there is no reward at this state also.
- $V(s_{12}) = \max[0.9(0.81)] \Rightarrow V(s_{12}) = \max[0.73] = 0.73$

V=0.81 s1	V=0.9 s2	V=1 s3	 s4
V=0.73 s5	 s6	V=0.9 s7	 s8
 V=0.66 s9	 V=0.73 s10	V=0.81 s11	V=0.73 s12

15

V=0.81 s1	 V=0.9 s2	V=1 s3	 s4
 V=0.73 s5	 s6	s7	 s8
 V=0.66 s9	V=0.59 s10	 V=0.53 s11	V=0.48 s12

V=0.81 s1	V=0.9 s2	V=1 s3	 s4
V=0.73 s5	 s6	 V=0.9 s7	 s8
 V=0.66 s9	V=0.73 s10	 V=0.81 s11	 V=0.73 s12

16

Q- Learning Algorithm

Q learning algorithm

For each s, a initialize the table entry $\hat{Q}(s, a)$ to zero.

Observe the current state s

Do forever:

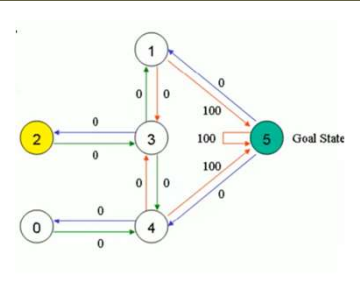
- Select an action a and execute it
- Receive immediate reward r
- Observe the new state s'
- Update the table entry for $\hat{Q}(s, a)$ as follows:

$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

- $s \leftarrow s'$

s is the current state, a is the action taken, r is the reward received, s' is the next state, and a' represents all possible next actions from s' .

17



$\gamma = 0.8$

$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

$3 \rightarrow 1$

$S = 3$
 $a = 1, 2, 4$
 $r = 0$
 $s' = 1$
 $a' = 3, 5$

$$\hat{Q}(3, 1) = 0 + 0.8(0 + 100) \Rightarrow 80$$

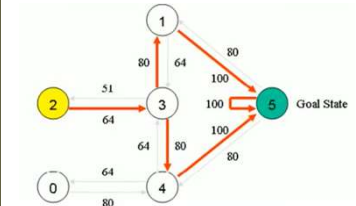
$\hat{Q}(3, 1) = 80$

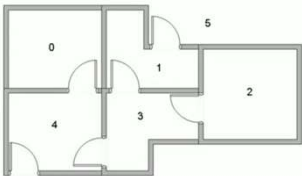
$1 \rightarrow 5$

$S = 1$
 $a = 3, 5$
 $r = 100$
 $s' = 5$
 $a' = 1, 4$

$$\hat{Q}(1, 5) = 100 + 0.8(0 + 0) \Rightarrow 100$$

$\hat{Q}(1, 5) = 100$





18

Challenges of Reinforcement Learning

- **Reward design** is a big challenge as in many games, as determining the rewards and its value is a challenge.
- **Absence of a model** is a challenge - Games like chess have fixed board and rules. But, many games do not have any fixed environment or rules. There is no underlying model as well. So, simulation must be done to gather experience.
- **Partial observability of states** - Many states are fully observable. Imagine a scenario in a weather forecasting where the uncertainty or partial observability exists as complete information about the state is simply not available.

19

Challenges of Reinforcement Learning

- **Time consuming operations** - More state spaces and possible actions may complicate the scenarios, resulting in more time consumption.
- **Complexity**- Many games like GO are complicated with much larger board configuration and many possibilities of actions. So, labelled data is simply not available. This adds more complexity to the design of reinforcement algorithms.

20

Advantages of Naive Bayes Classifier

- Easy to implement and computationally efficient.
- Effective in cases with a large number of features.
- Performs well even with limited training data.
- It performs well in the presence of categorical features.
- For numerical features data is assumed to come from normal distributions

Disadvantages of Naive Bayes Classifier

- Assumes that features are independent, which may not always hold in real-world data.
- Can be influenced by irrelevant attributes.
- May assign zero probability to unseen events, leading to poor generalization.

21

Applications of Reinforcement Learning

There are many applications of RL. Some of the application domains where reinforcement learning is used are listed below:

1. Industrial automation
2. Resource management applications to allocate resource
3. Traffic light controller to reduce congestion of traffic
4. Personalized recommendation systems like news
5. Bidding for advertisement
6. Driverless cars

22

Thank You