AIFA: REINFORCEMENT LEARNING

24/03/2025

Koustav Rudra

Overview

- Supervised Learning:
  - Immediate feedback (labels provided for every input)

- Unsupervised Learning:
  - No feedback (no labels provided)

- Reinforcement Learning:
  - Delayed scalar feedback (a number called reward)

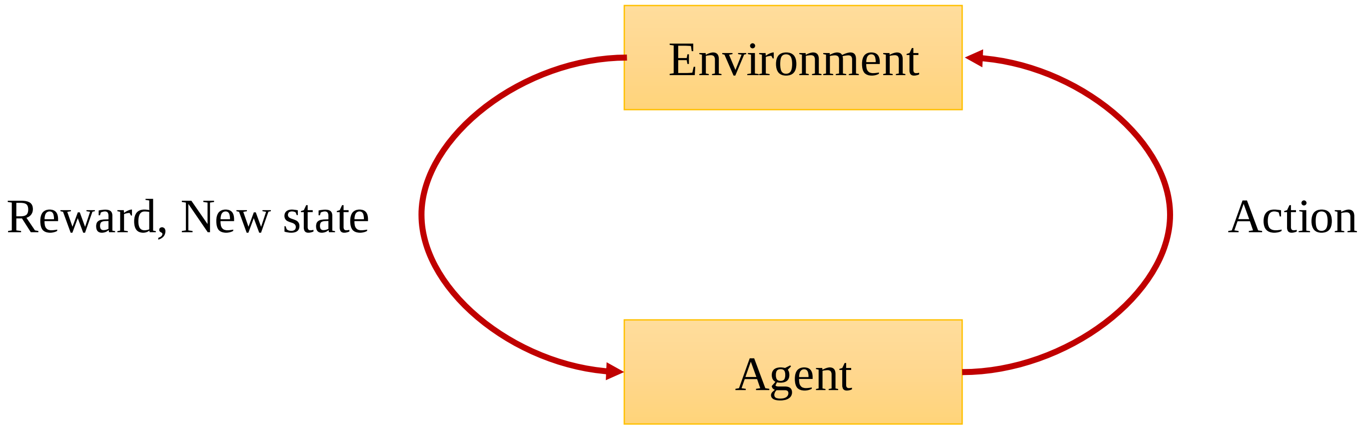
Reinforcement Learning

- RL deals with agents that must sense and act upon their environment
  - This combines classical AI and machine learning techniques
  - It the most comprehensive problem solving

- Example:
  - A robot cleaning the room and recharging its battery
  - How to invest in shares

The RL Framework

- Learn from interaction with environment to achieve a goal



$$s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2} \cdots$$

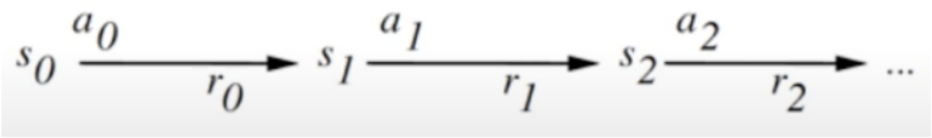
Your action influences the state of the world which determines its reward

Challenges

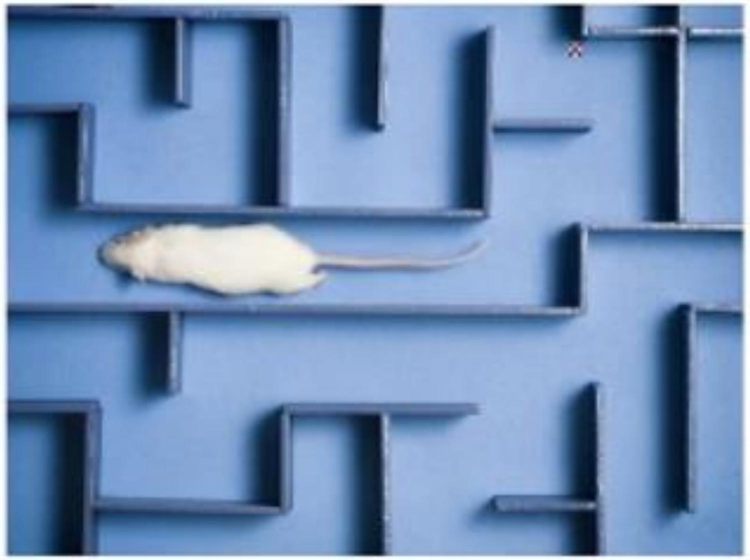
- The outcome of your actions may be uncertain
- You may not be able to perfectly sense the state of the world
- The reward may be stochastic
- The reward may be delayed (finding food in maze)
- You may have no clue (model) about how the world responds to your actions
- You may have no clue (model) of how rewards are being paid off
- The world may change while you try to learn it
- How much time do you need to explore uncharted territory before you exploit what you have learned?

The Task

- To learn an optimal policy that maps states of the world to actions of the agent
  - For example: If this patch of room is dirty, I clean it. If my battery is empty, I recharge it.
  - $\pi : S \rightarrow A$



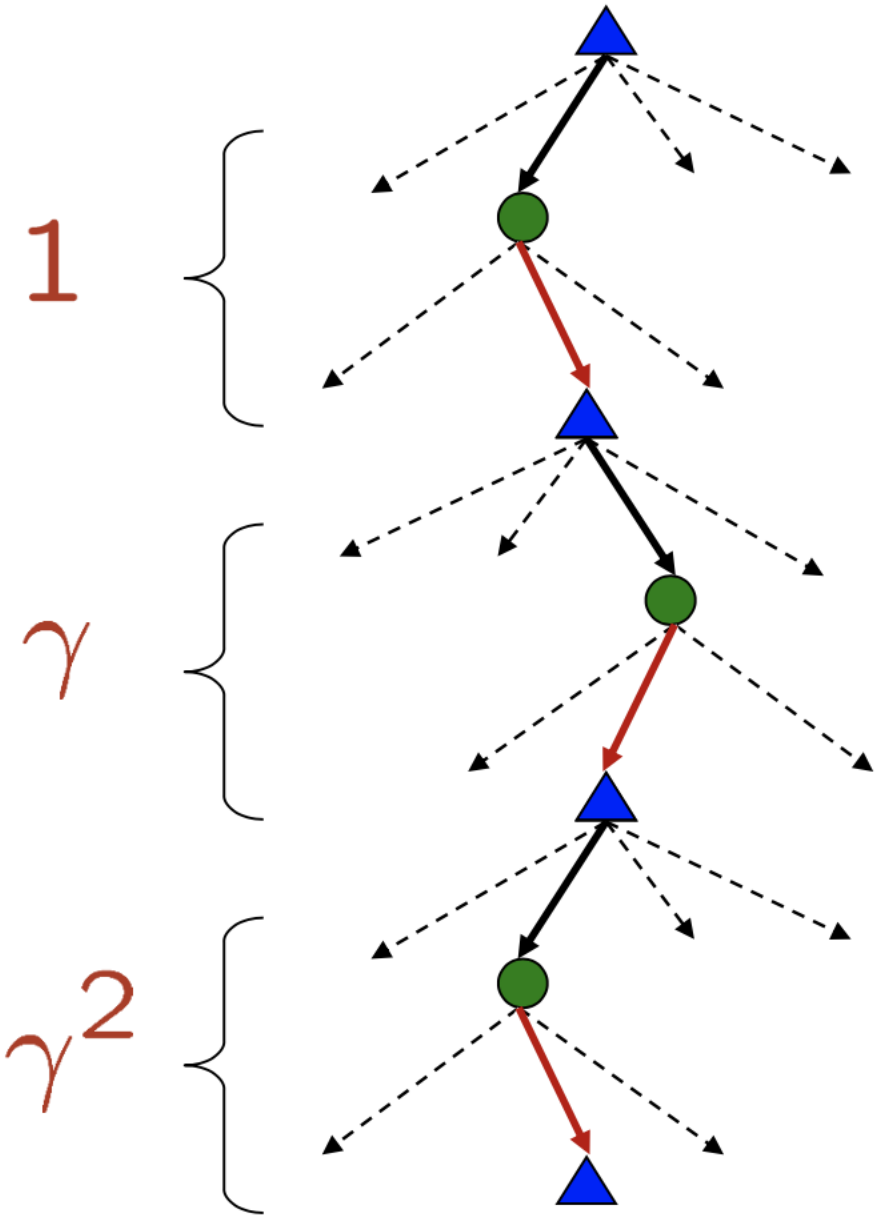
- What is it that the agent tries to optimize?
  - The total future discounted reward
  - $V^{\pi}(S_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$
  - $V^{\pi}(S_t) = \sum_{i=0}^{\infty} \gamma^i r_{t+i}, 0 \leq \gamma < 1$
  - Immediate reward is worth more than future reward



- What would happen to a mouse in a maze with γ=0

Discounting

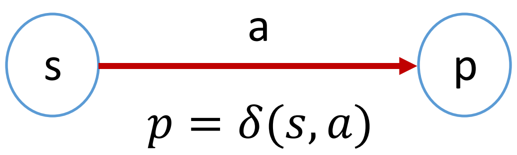
- How to discount?
  - Each time we descend a level, we multiply in the discount once
- Why discount?
  - Reward now is better than later
  - Also helps our algorithms converge
- Example: discount of 0.5
  - U([1,2,3]) = 1*1 + 0.5*2 + 0.25*3
  - U([1,2,3]) < U([3,2,1])



Value Function

- Suppose we have access to the optimal value function that computes the total future discounted reward $V^*(s)$

- $\pi^*(s) = \max_a [r(s,a) + \gamma V^*(\delta(s,a))]$



- We assume that we know what the reward will be if we perform action "a" in state "s": $r(s,a)$

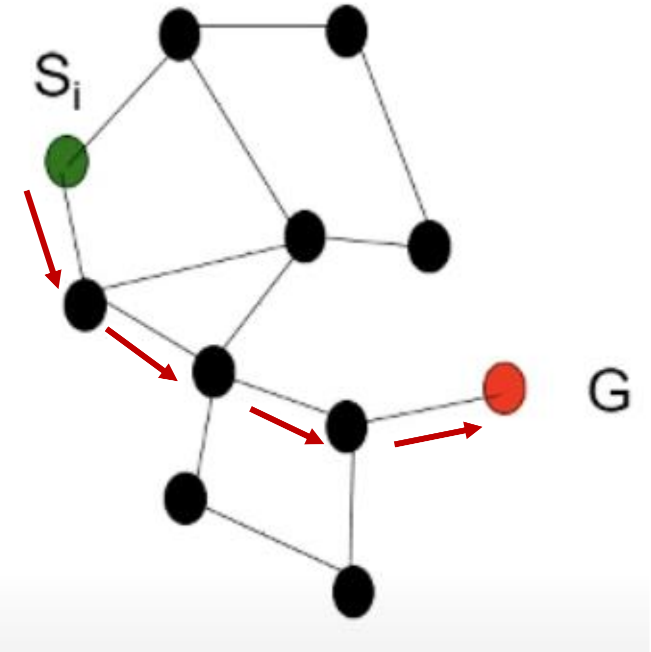
- We also assume we know what the next state of the world will be if we perform action "a" in state "s":
  - $s_{t+1} = \delta(s_t, a)$

deterministic

Example 1

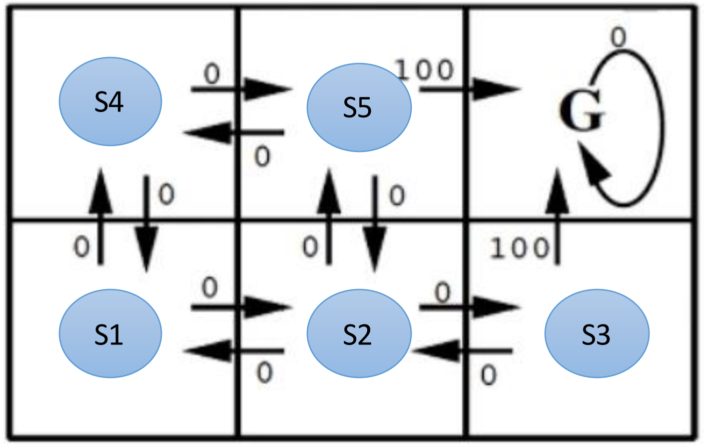
- Consider some complicated graph, and we would like to find the shortest path from a node S_i to a goal node G

- Traversing an edge will incur costs – this is the edge weight
- The value function encodes the total remaining distance to the goal node from any node s, that is:
  - $v(s) = {}^{1}/_{distance}$ to goal from s

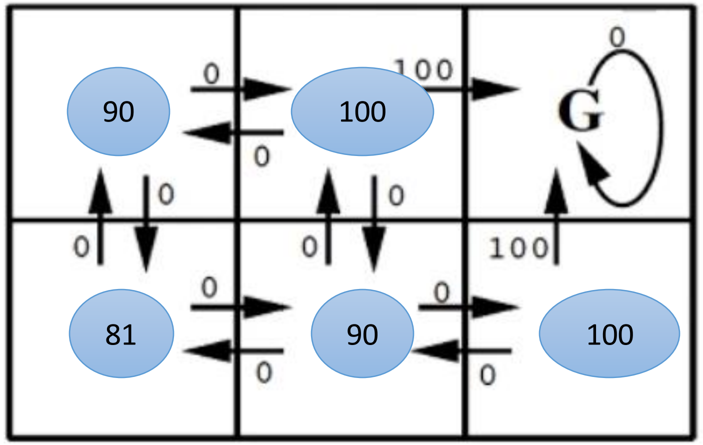


- If we know v(s) the problem is trivial
  - Simply choose the node with highest $v(s)$

Example 2



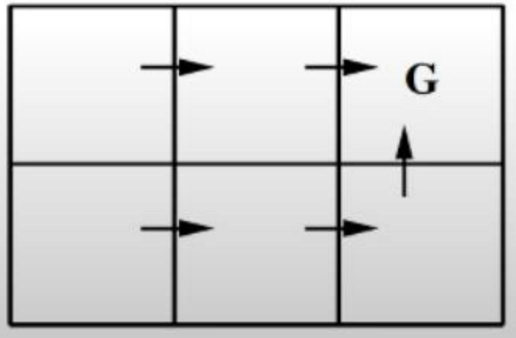
r(s,a) : immediate reward values



$V^*(s)$

$$V^*(s) = values$$

$$V^*(s1) = 0 + 0.9 \times 0 + (0.9)^2 100 = 81$$



One optimal policy

$$\pi^*(s) = \max_a [r(s,a) + \gamma V^*(\delta(s,a))]$$

Thank You