Reinforcement Learning
Lecture 1

Dr. Apurva Narayan
http://anarayan.com

8-Jan-2024

Plan for today

Motivation for material

What will we learn

Course organization

whoami

Assistant Professor in Computer Science and ECE

Agra, India

PhD, Waterloo, Canada
Adj. Asst. Prof. Waterloo, Canada

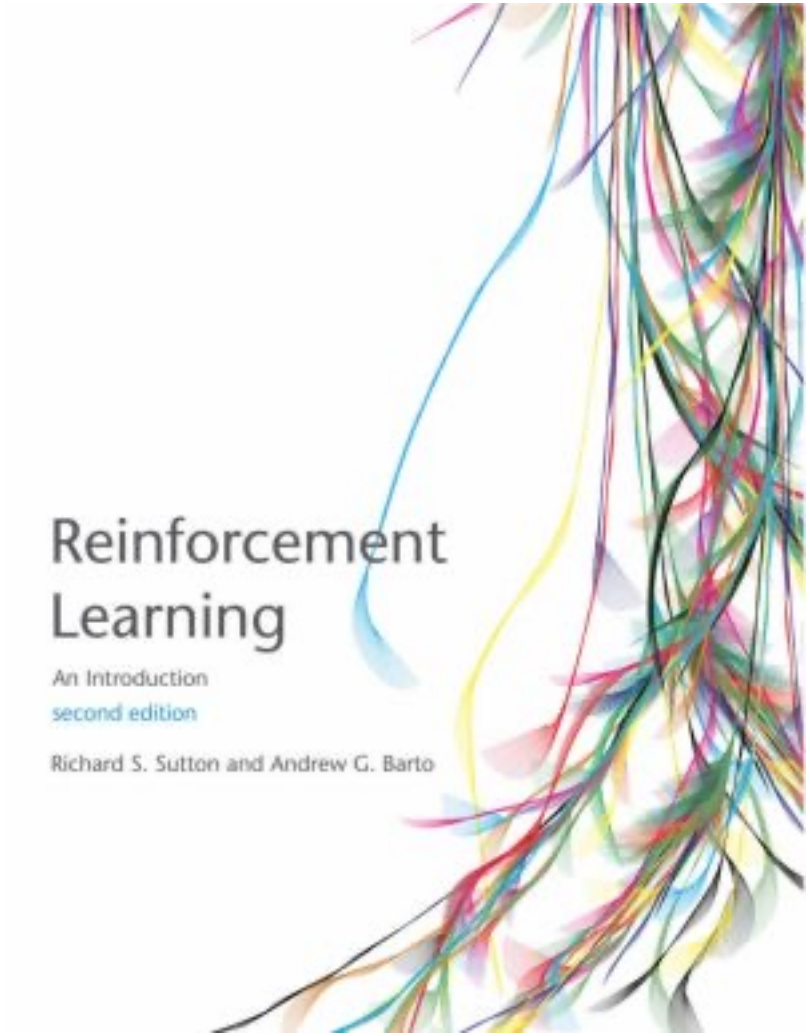
Asst. Prof. UBC, Canada
Affill. Asst. Prof. UBC, Canada

Lead: Intelligent Data Science Lab
http://a-narayan.github.io

Asst. Prof. Western, Canada

Course Structure

Lecture:
Mon: 11:30am – 12:20pm
Fri: 10:30am – 12:20pm
Location:
Mon: B&G 0153
Fri: NCB 114
OWL:
- You should have been added
Evaluation:
- Assignments – 20%
- Midterm – 20%
- Project – 50%
- Project Presentation 10%



Text Books

- An Introduction to Reinforcement Learning, Sutton and Barto, 1998

http://webdocs.cs.ualberta.ca/~sutton/book/the-book.html

Reinforcement Learning

What is Reinforcement Learning?

- Reinforcement learning is also known as
  - Optimal control
  - Approximate dynamic programming
  - Neuro-dynamic programming

- Wikipedia: reinforcement learning is an area of machine learning inspired by behavioural psychology, concerned with how software **agents** ought to take **actions** in an **environment** so as to maximize some notion of cumulative **reward**.

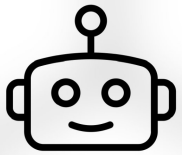
Animal Psychology

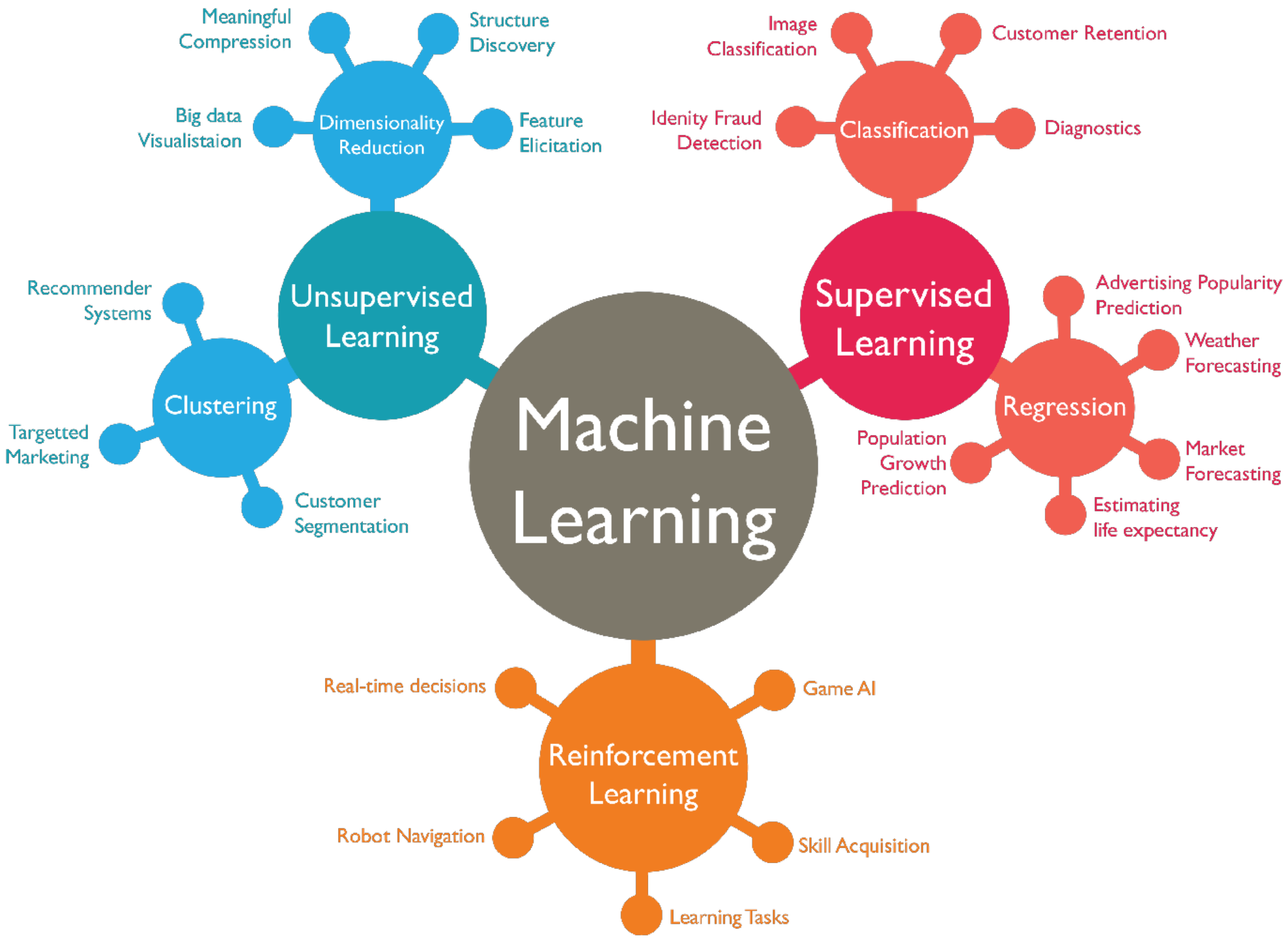
- Negative reinforcements:
  - Pain and hunger
- Positive reinforcements:
  - Pleasure and food

- Reinforcements used to train animals

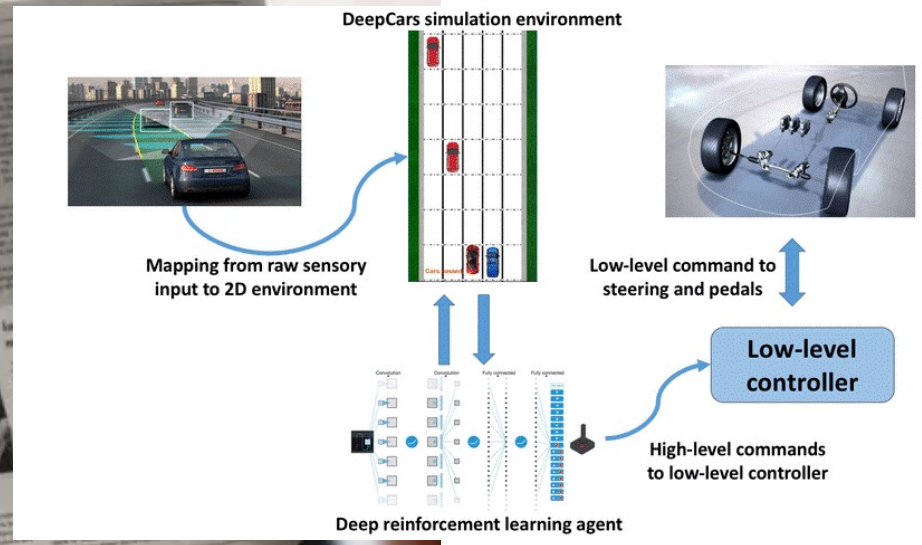
- Let’s do the same with computers!

Faces of Reinforcement Learning

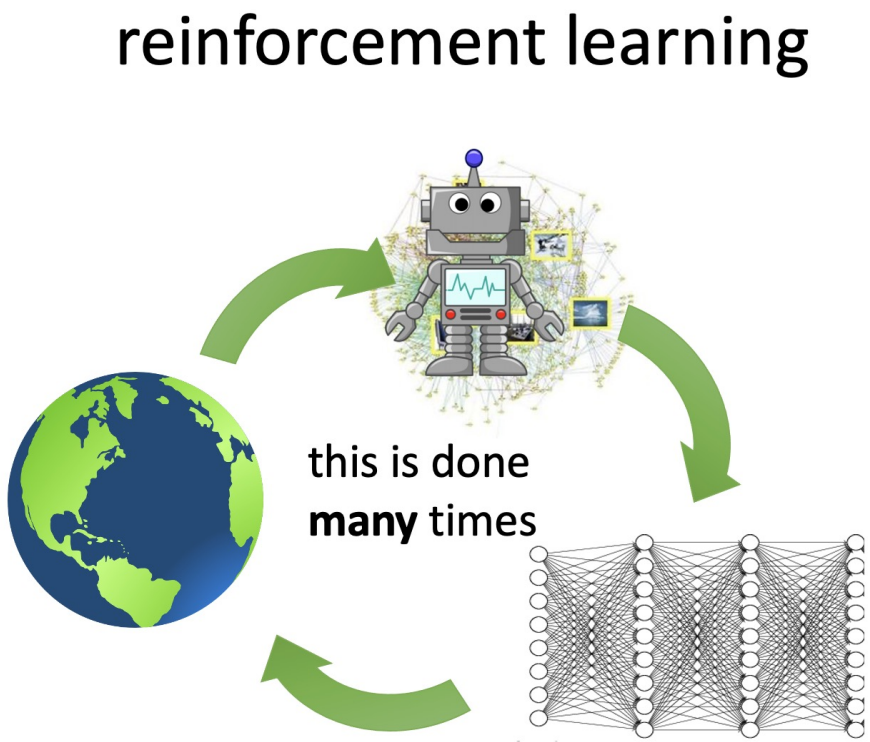


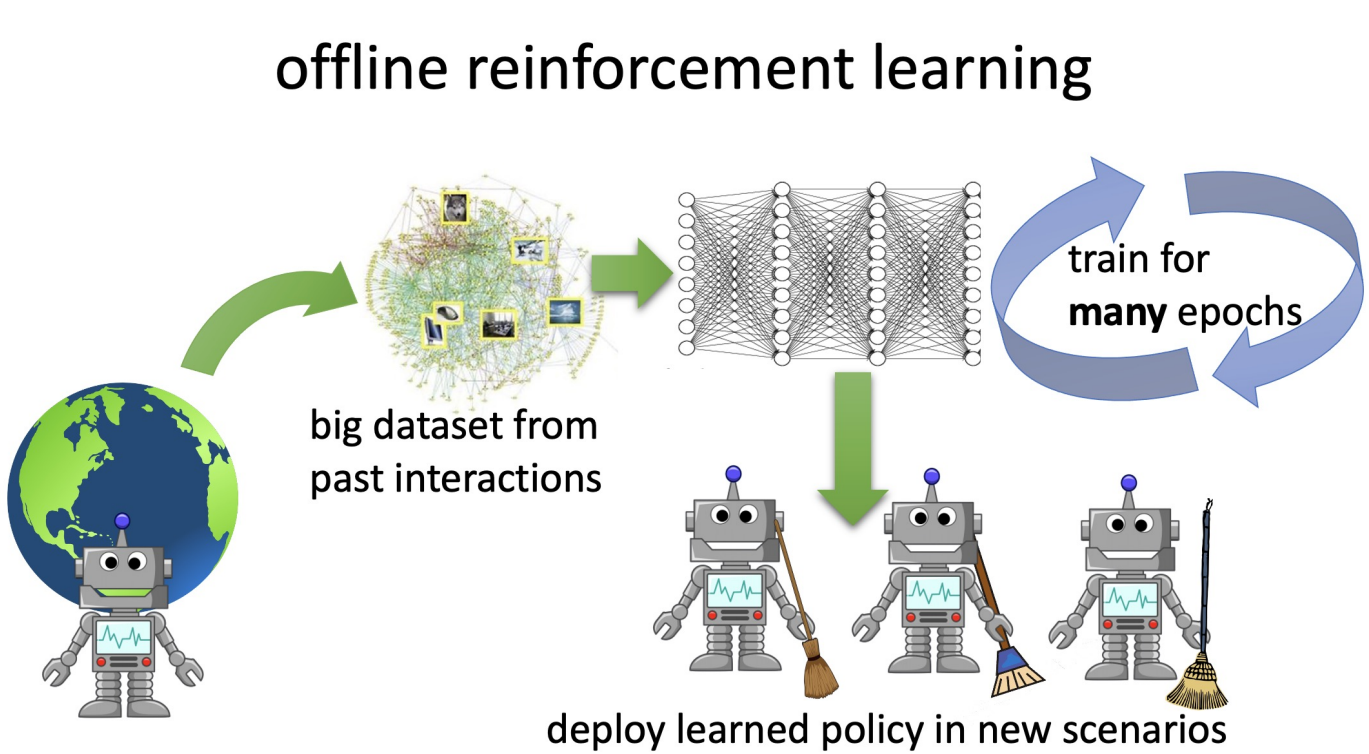

Machine Learning
Unsupervised Learning
Supervised Learning
Reinforcement Learning
Dimensionality Reduction
Clustering
Classification
Regression
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Recommender Systems
Targetted Marketing
Customer Segmentation
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

Reinforcement Learning in the news!



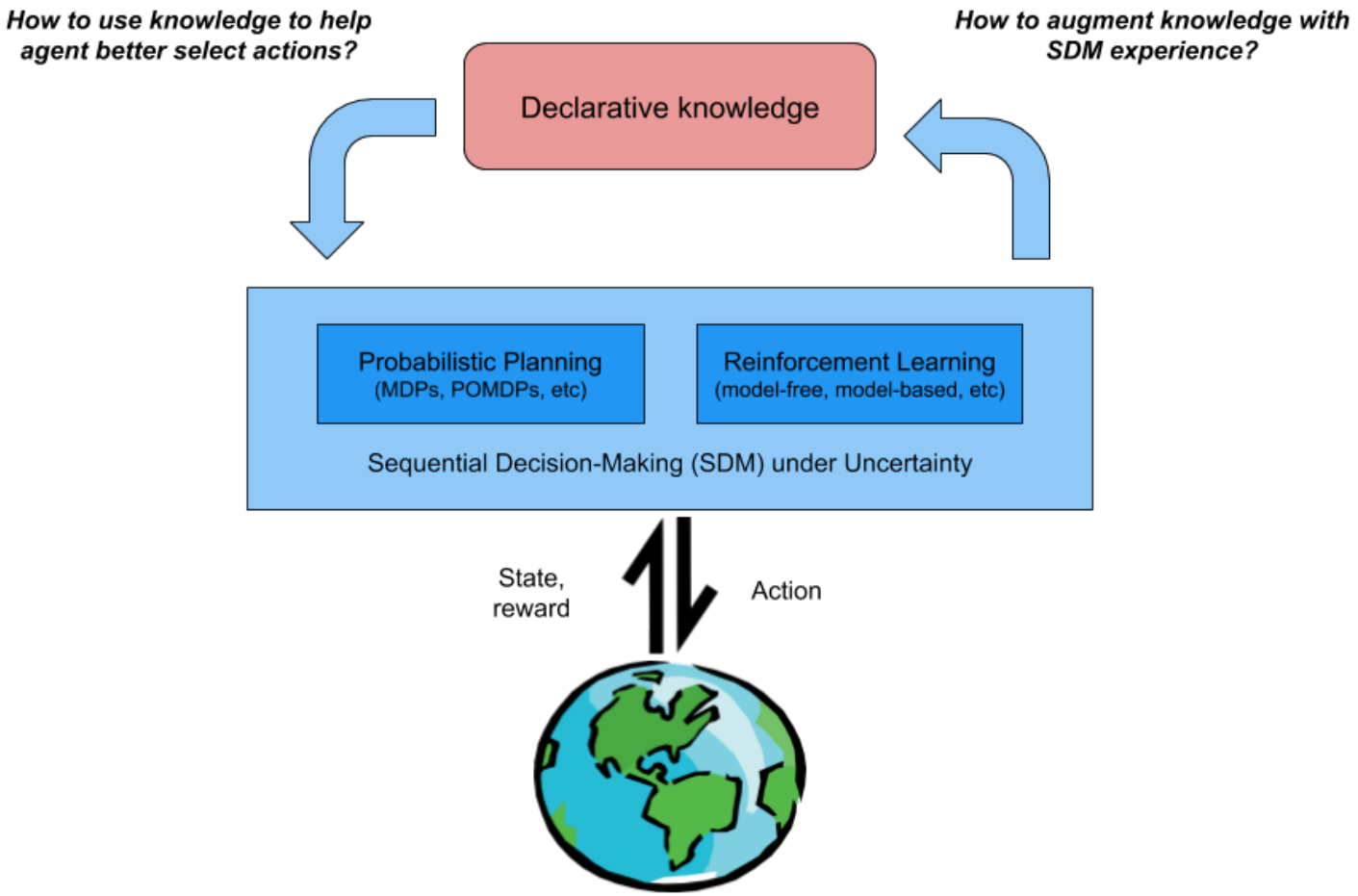
Reinforcement Learning Online Vs Offline





Foundations Are Important

Sequential Decision Making under Uncertaitny



Characteristics of RL

What makes reinforcement learning different from other machine learning paradigms?

- There is **no supervisor**, only a *reward* signal
- Feedback is **delayed**, not instantaneous
- **Time** really matters (sequential, non i.i.d data)
- Agent's **actions affect** the subsequent data it receives

Examples of RL

Optimize

- Process planning
- Job shop scheduling
- Yield management
- Supply chain
- Demand forecasting
- Warehouse operations optimization (picking)
- Production coordination
- Fleet logistics
- Product design
- Facilities location
- Camera Tuning
- Search ordering
- Agriculture
- Network optimization
- DDoS attack prevention
- Service availability

Control

- Robotics
- Wind Turbine Control
- HVAC
- Autonomous vehicles
- Factory automation
- Smart grids
- Machine Tuning

Monitor and Maintain

- Quality control
- Fault detection and isolation
- Predictive maintenance
- Inventory monitoring
- Supply chain risk management

Examples of RL

- Fly stunt manoeuvres in a helicopter
- Defeat the world champion at Backgammon
- Manage an investment portfolio Control a power station
- Make a humanoid robot walk
- Play many different Atari games better than humans


ima...
021 3 1

Problem Set Up - RL

- A reward R_t is a scalar feedback signal
- Indicates how well agent is doing at step t
- The agent's job is to maximise cumulative reward

Reinforcement learning is based on the reward hypothesis

Definition (Reward Hypothesis)

All goals can be described by the maximisation of expected cumulative reward

Examples of RL

Fly stunt manoeuvres in a helicopter

- +ve reward for following desired trajectory
- −ve reward for crashing

Defeat the world champion at Backgammon

- +/−ve reward for winning/losing a game

Manage an investment portfolio

- +ve reward for each $ in bank

Control a power station

- +ve reward for producing power
- −ve reward for exceeding safety thresholds

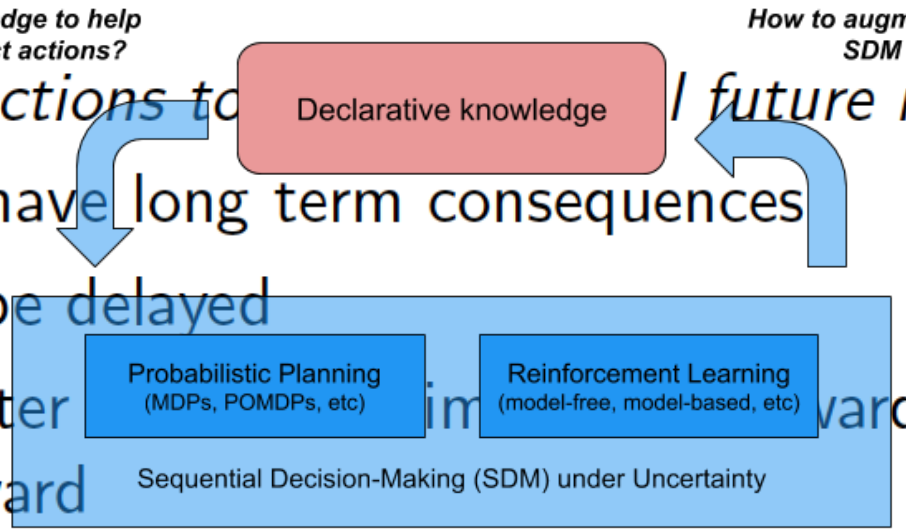
Make a humanoid robot walk

- +ve reward for forward motion
- −ve reward for falling over

Play many different Atari games better than humans

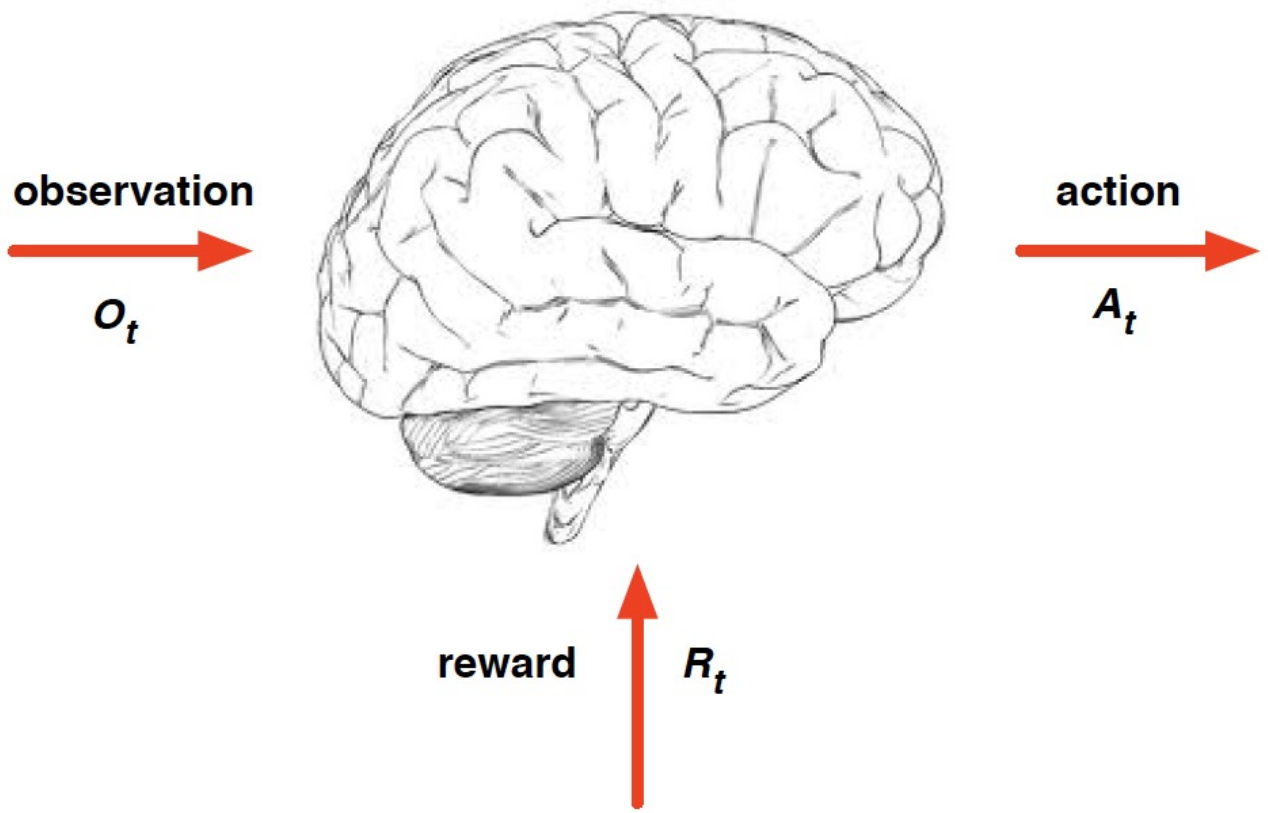
- +/−ve reward for increasing/decreasing score

Sequential Decision Making

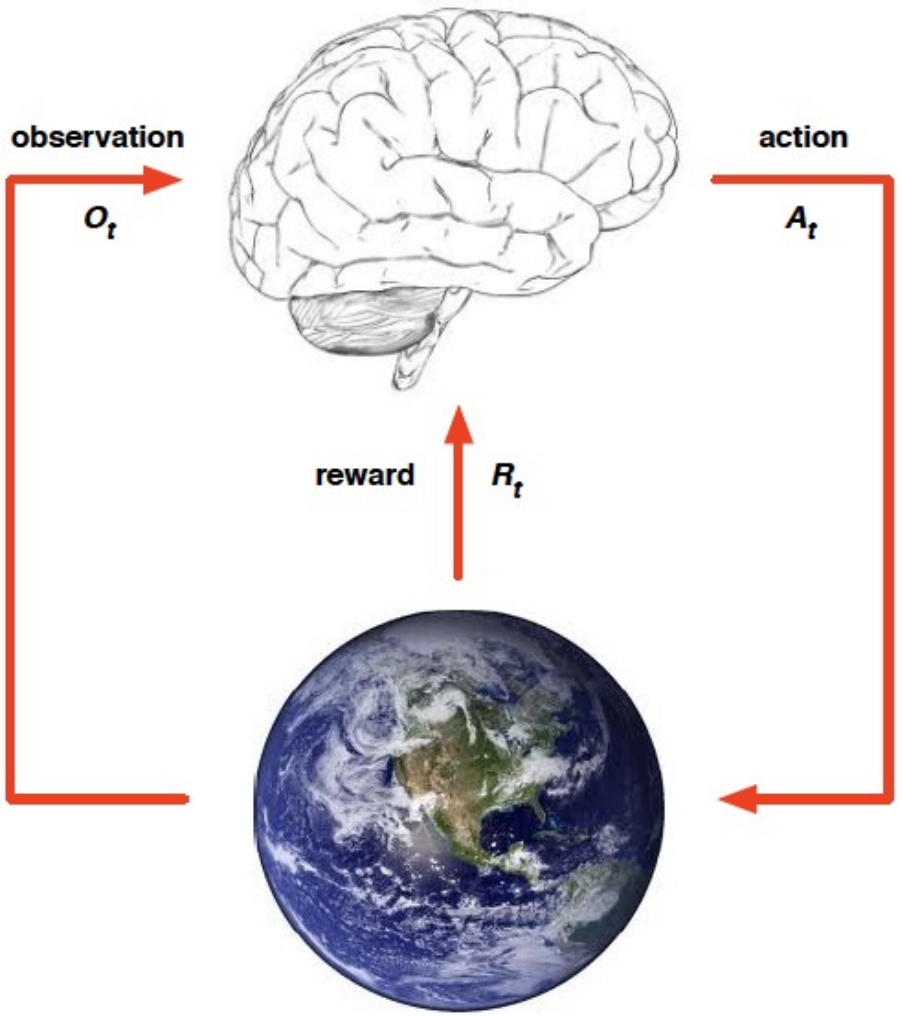


- Goal: *select actions to* *l future reward*
- Actions may have long term consequences
- Reward may be delayed
- It may be better *im* *vard* to gain more long-term reward
- Examples:
  - A financial investment (may take months to mature)
  - Refuelling a helicopter *revent* a crash in several hours)
  - Blocking opponent mo* *t help winning chances many moves from now)

Agent and Environment



Agent and Environment



- At each step t the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step

Operations research

- Example: vehicle routing

- **Agent:** vehicle routing software
- **Environment:** stochastic demand
- **State:** vehicle location, capacity and depot requests
- **Action:** vehicle route
- **Reward:** - travel costs

Robotic Control

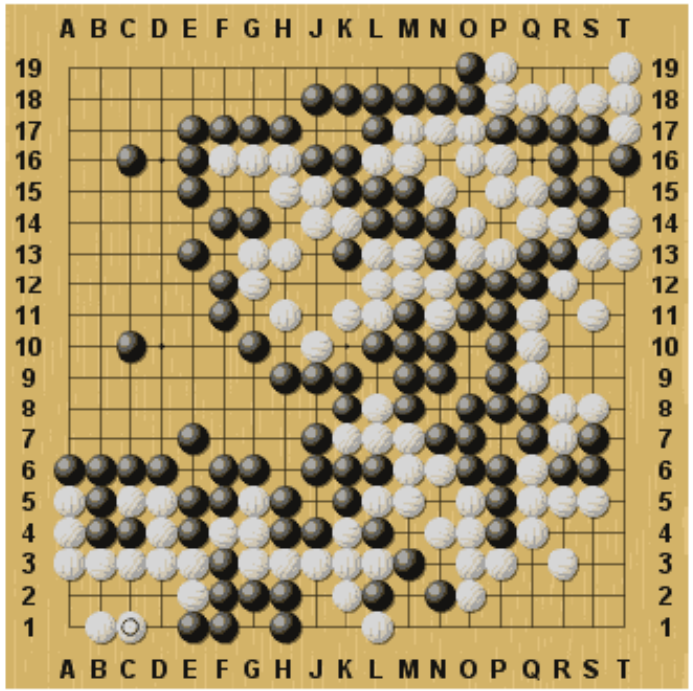
- Example: helicopter control

- **Agent:** controller
- **Environment:** helicopter
- **State:** position, orientation, velocity and angular velocity
- **Action:** collective pitch, cyclic pitch, tail rotor control
- **Reward:** - deviation from desired trajectory

- 2008 (Andrew Ng): automated helicopter wins acrobatic competition against humans

Game Playing

- Example: Go (one of the oldest and hardest board games)



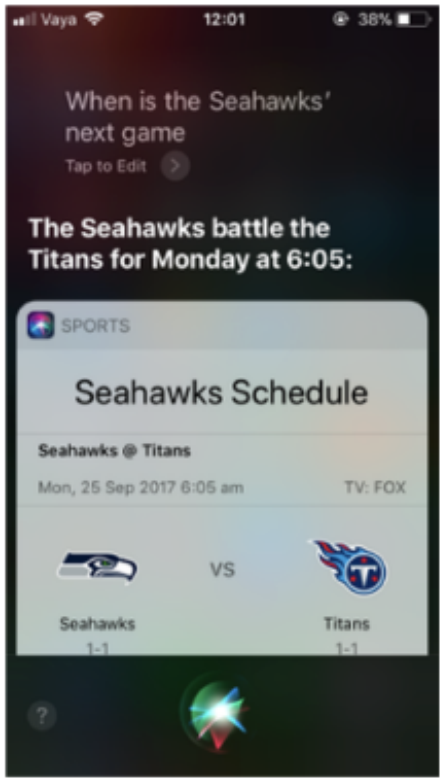
- **Agent:** player
- **Environment:** opponent
- **State:** board configuration
- **Action:** next stone location
- **Reward:** +1 win / -1 loose

- 2016: AlphaGo defeats top player Lee Sedol (4-1)
  - Game 2 move 37: AlphaGo plays unexpected move (odds 1/10,000)

Conversational agent

- **Agent:** virtual assistant
- **Environment:** user
- **State:** conversation history
- **Action:** next utterance
- **Reward:** points based on task completion, user satisfaction, etc.

- Today: active area of research



Computational Finance

- Automated trading

- **Agent:** trading software
- **Environment:** other traders
- **State:** price history
- **Action:** buy/sell/hold
- **Reward:** amount of profit

Example: how to purchase a large # of shares in a short period of time without affecting the price

- Comprehensive, but challenging form of machine learning
  - Stochastic environment
  - Incomplete model
  - Interdependent sequence of decisions
  - No supervision
  - Partial and delayed feedback

- **Long term goal**: lifelong machine learning

History and State

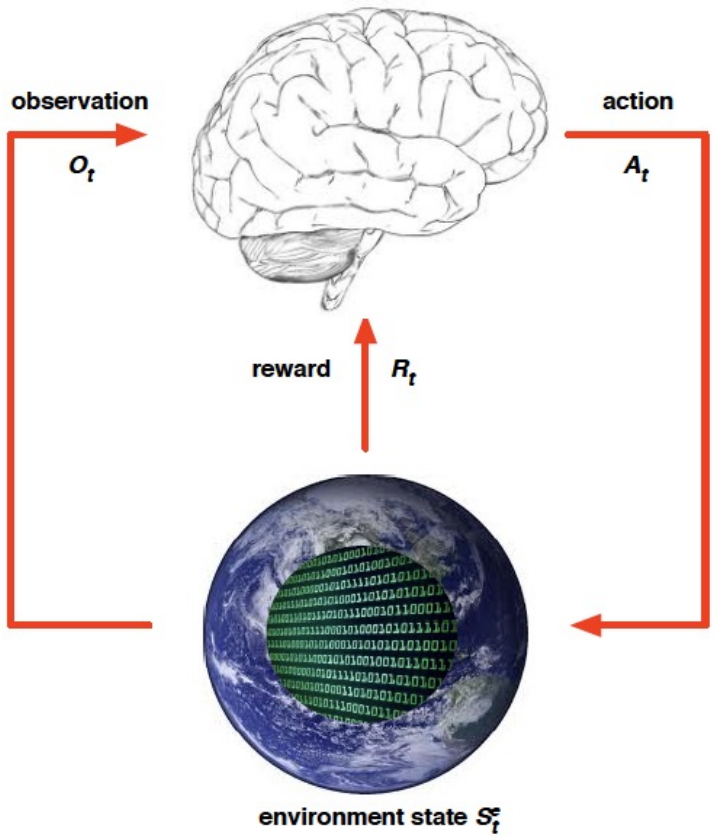
- The history is the sequence of observations, actions, rewards

$$H_t = O_1, R_1, A_1, ..., A_{t-1}, O_t, R_t$$

- i.e. all observable variables up to time t
- i.e. the sensorimotor stream of a robot or embodied agent
- What happens next depends on the history:
  - The agent selects actions
  - The environment selects observations/rewards
- State is the information used to determine what happens next
- Formally, state is a function of the history:

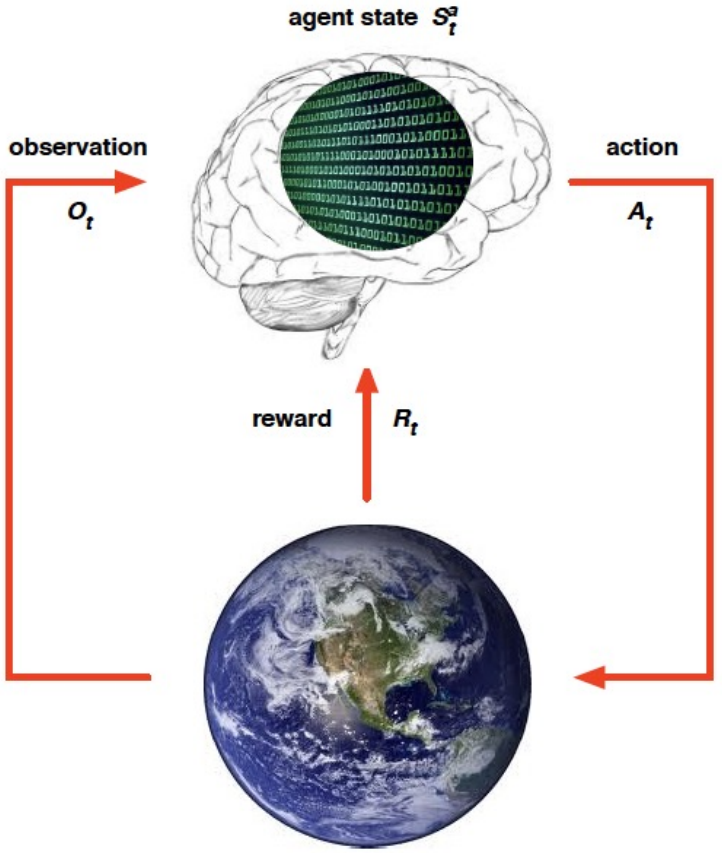
$$S_t = f(H_t)$$

Environment State



- The environment state S_t^e is the environment's private representation
- i.e. whatever data the environment uses to pick the next observation/reward
- The environment state is not usually visible to the agent
- Even if S_t^e is visible, it may contain irrelevant information

Agent State



- The agent state S_t^a is the agent's internal representation
- i.e. whatever information the agent uses to pick the next action
- i.e. it is the information used by reinforcement learning algorithms
- It can be any function of history:

$$S_t^a = f(H_t)$$

Information State

An information state (a.k.a. Markov state) contains all useful information from the history.

Definition

A state S_t is Markov if and only if

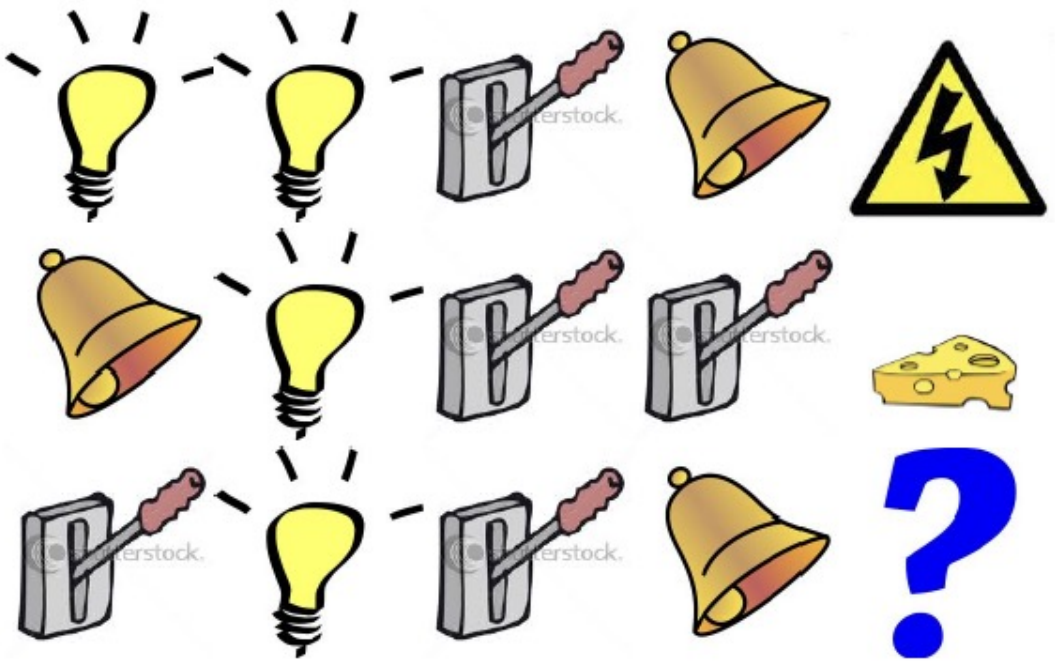
$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, ..., S_t]$$

- "The future is independent of the past given the present"

$$H_{1:t} \to S_t \to H_{t+1:\infty}$$

- Once the state is known, the history may be thrown away
- i.e. The state is a sufficient statistic of the future
- The environment state S_t^e is Markov
- The history H_t is Markov

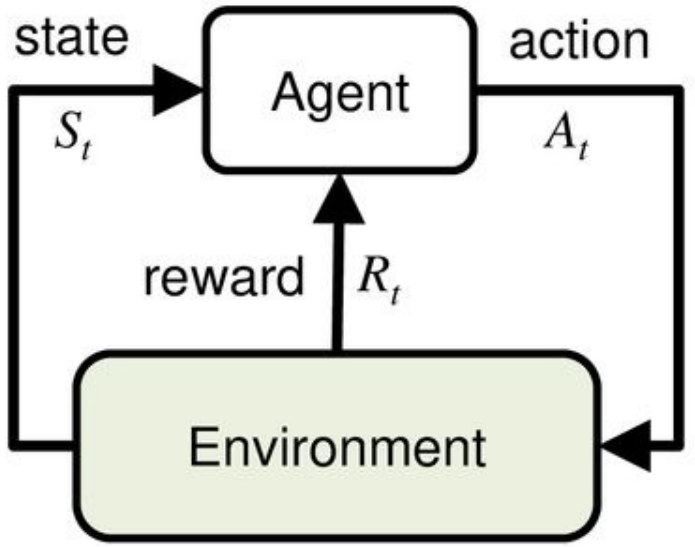
Decision Making Example



- What if agent state = last 3 items in sequence?
- What if agent state = counts for lights, bells and levers?
- What if agent state = complete sequence?

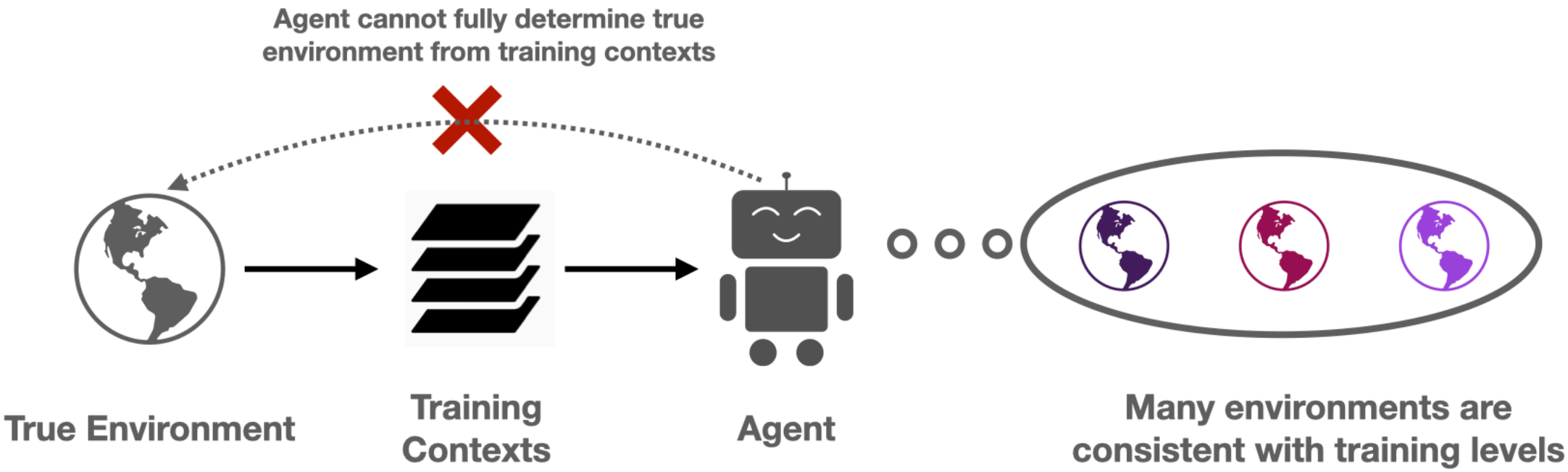
Fully Observable Environment

- Full observability:
  - agent directly observes environment state
  - Agent state = environment state = information state
  - Formally, this is a Markov decision process (MDP)



Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Partially Observable Environment



Major Components of an RL Agent

- An RL agent may include one or more of these components:
  - Policy: agent's behaviour function
  - Value function: how good is each state and/or action
  - Model: agent's representation of the environment

Policy

- A policy is the agent's behaviour
- It is a map from state to action, e.g.
- Deterministic policy: $a = \pi(s)$
- Stochastic policy: $\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$

Value Function

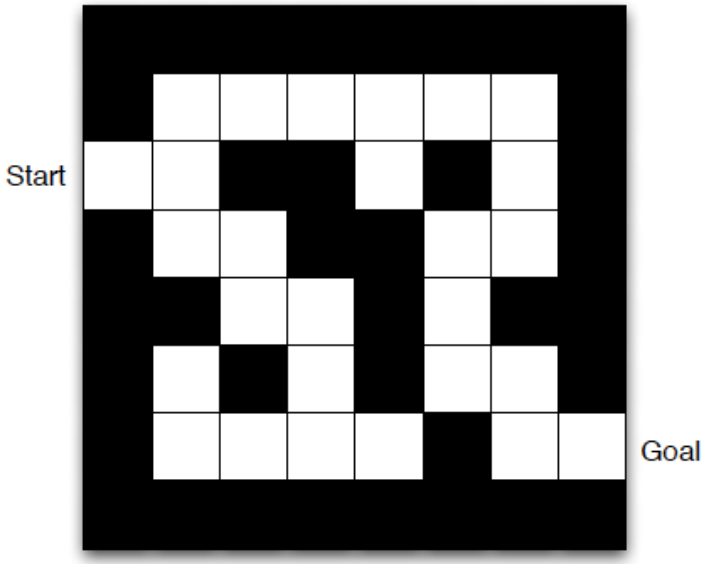
- Value function is a prediction of future reward
- Used to evaluate the goodness/badness of states
- And therefore to select between actions, e.g.

$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s \right]$$

Model

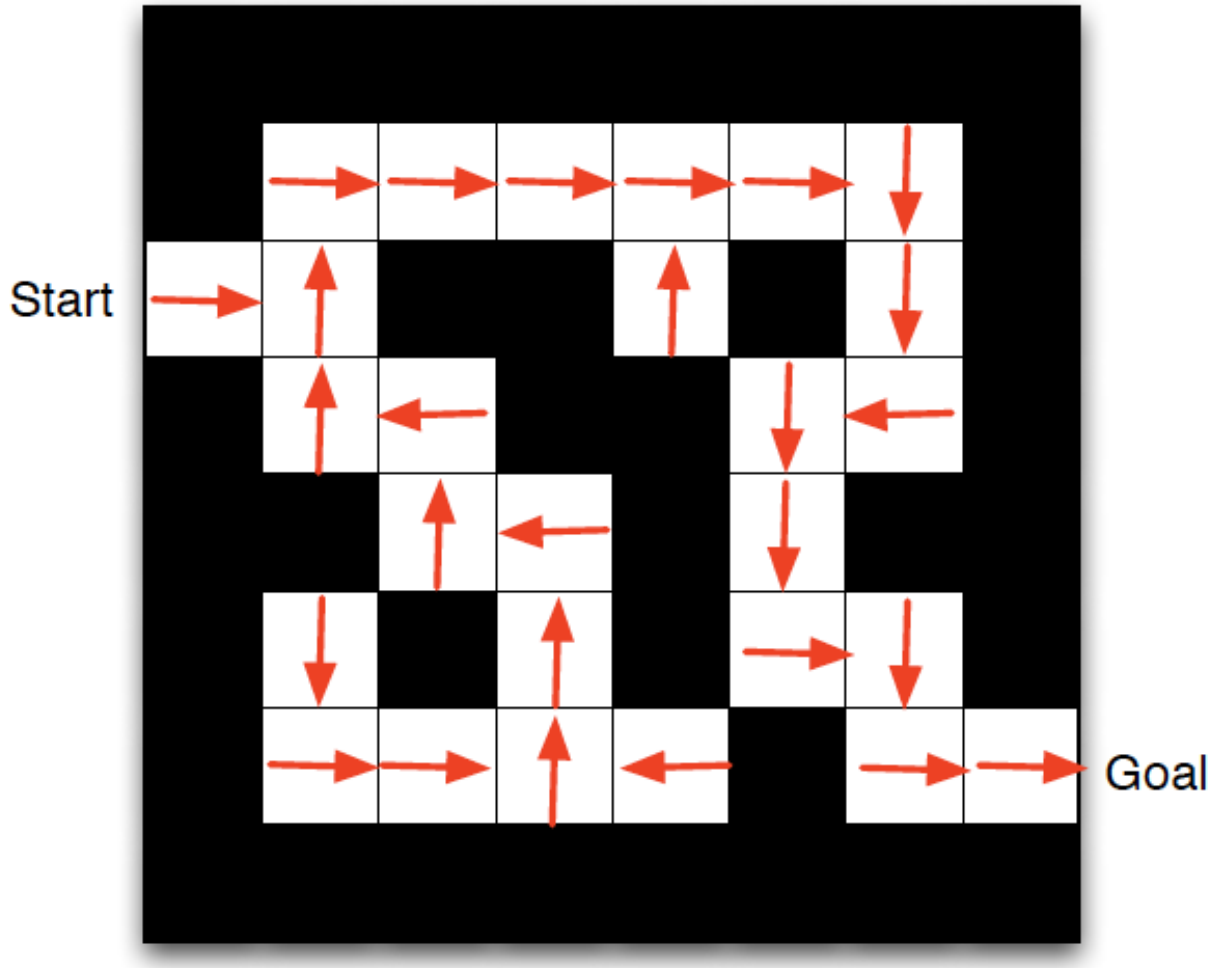
- A model predicts what the environment will do next
- $\mathcal{P}$ predicts the next state
- $\mathcal{R}$ predicts the next (immediate) reward, e.g.

Maze Example

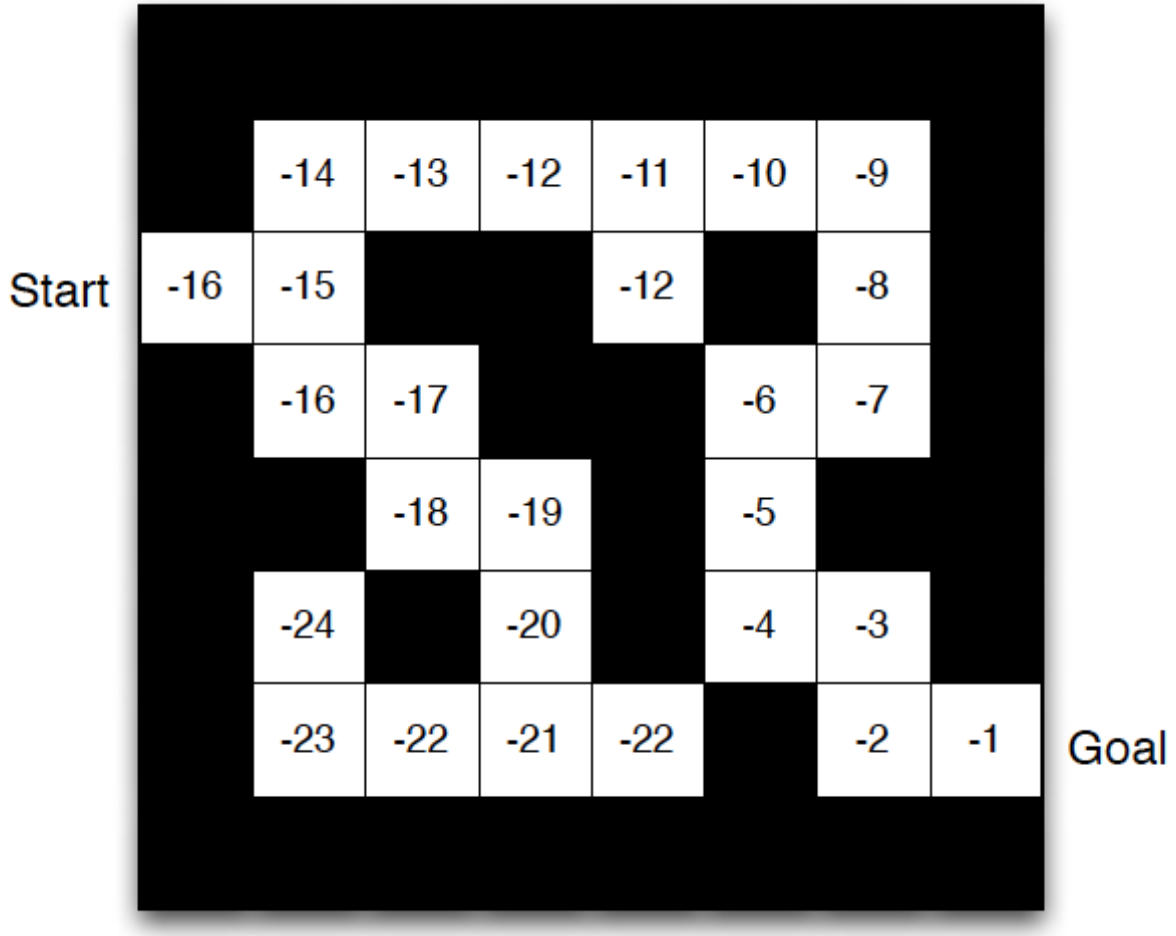


- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agent's location

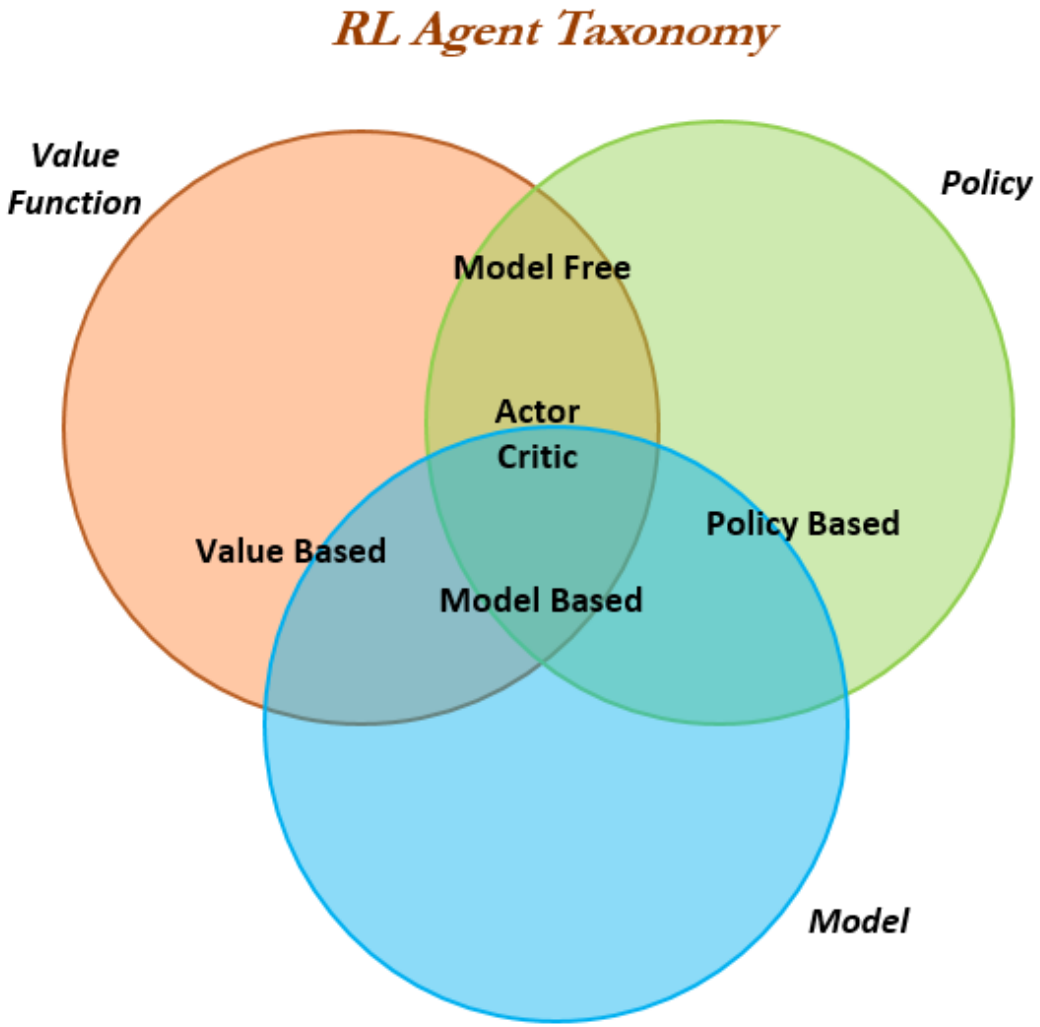
Maze Example: Policy



Maze Example: Value Function



Categorizing RL Agents

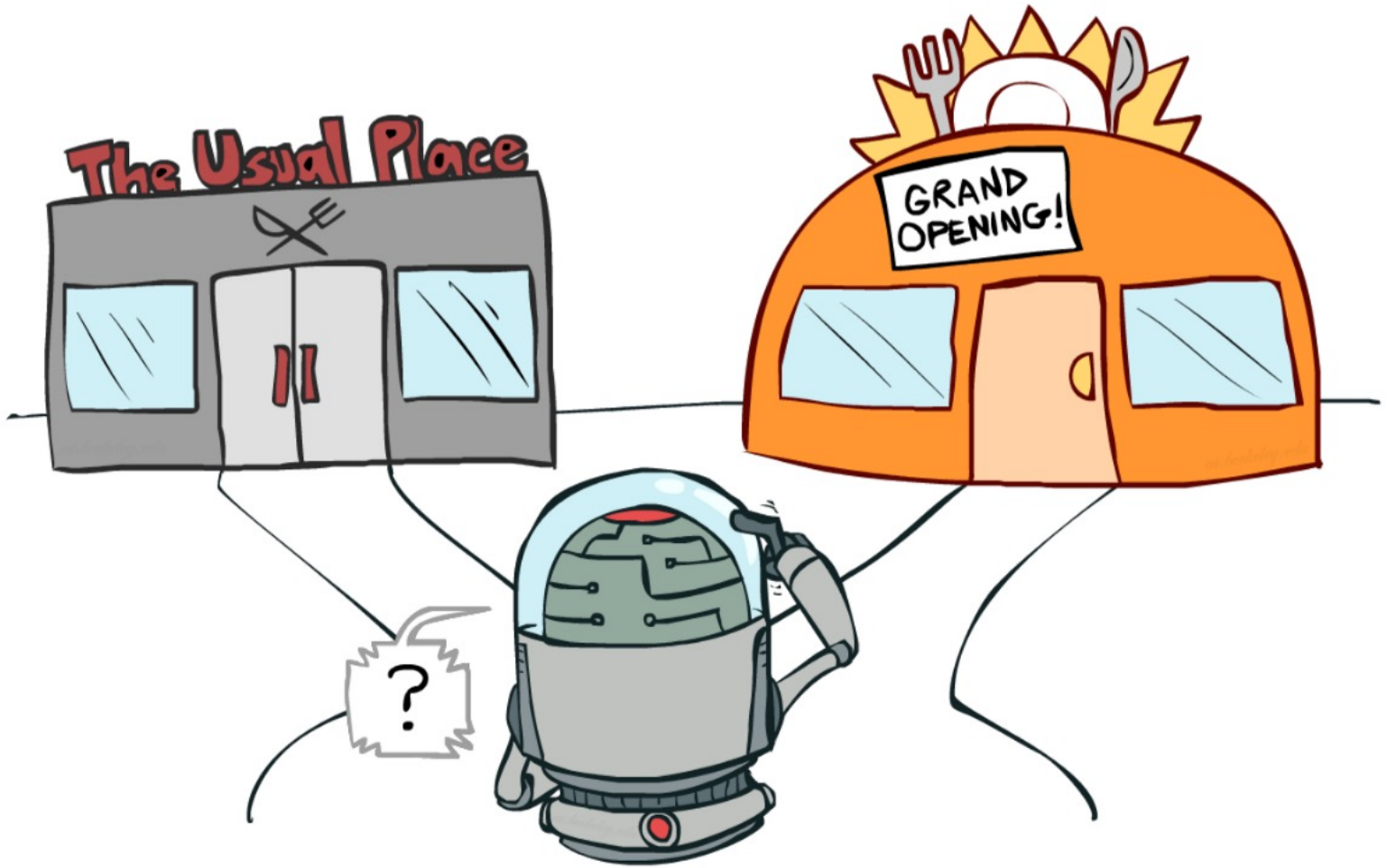


Fundamental Problem in Decision Making

Two fundamental problems in sequential decision making

- Reinforcement Learning:
  - The environment is initially unknown
  - The agent interacts with the environment
  - The agent improves its policy
- Planning:
  - A model of the environment is known
  - The agent performs computations with its model (without any external interaction)
  - The agent improves its policy
  - a.k.a. deliberation, reasoning, introspection, pondering, thought, search

Exploration - Exploitation



Exploration – Exploitation – Examples

- Restaurant Selection
  - Exploitation Go to your favourite restaurant
  - Exploration Try a new restaurant
- Online Banner Advertisements
  - Exploitation Show the most successful advert
  - Exploration Show a different advert
- Oil Drilling
  - Exploitation Drill at the best known location
  - Exploration Drill at a new location
- Game Playing
  - Exploitation Play the move you believe is best
  - Exploration Play an experimental move

Next Lecture

- MDPs