

# Lecture 1: Introduction to RL

Professor Emma Brunskill

CS234 RL

Winter 2021

- Today the 3rd part of the lecture includes slides from David Silver's introduction to RL slides or modifications of those slides

# Today's Plan

- Overview of reinforcement learning
- Course logistics
- Introduction to sequential decision making under uncertainty

Make good sequences of decisions

Learn to make good sequences of decisions



# Reinforcement Learning

Fundamental challenge in artificial intelligence and machine learning is  
learning to make good decisions under uncertainty

# 2010s: New Era of RL. Atari

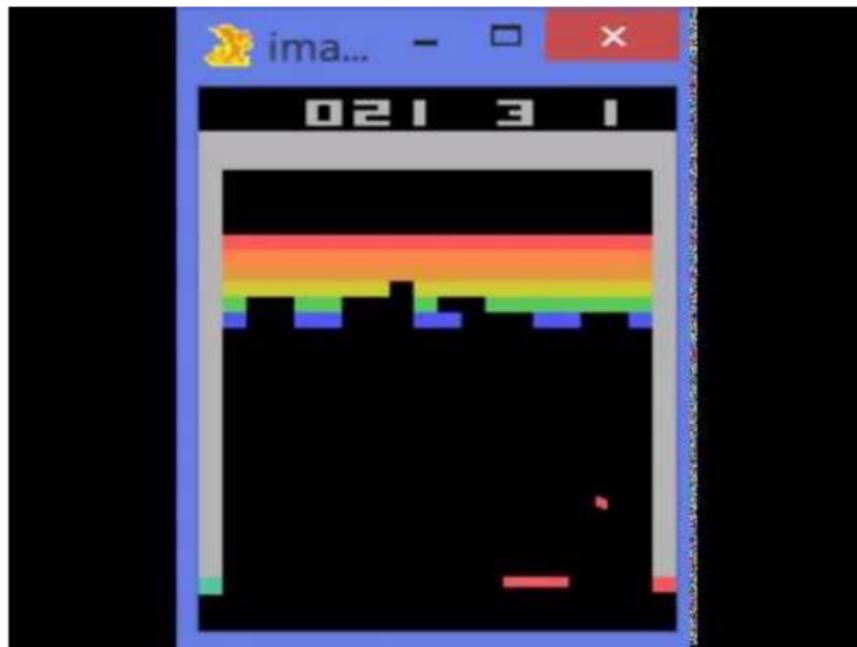


Figure: DeepMind Nature, 2015

# 2010s: New Era of RL. Robotics



Figure: Chelsea Finn, Sergey Levine, Pieter Abbeel

# Expanding Reach. Educational Games



Figure: RL used to optimize Refraction 1, Madel, Liu, Brunskill, Popvic AAMAS 2014.

# Expanding Reach. Health

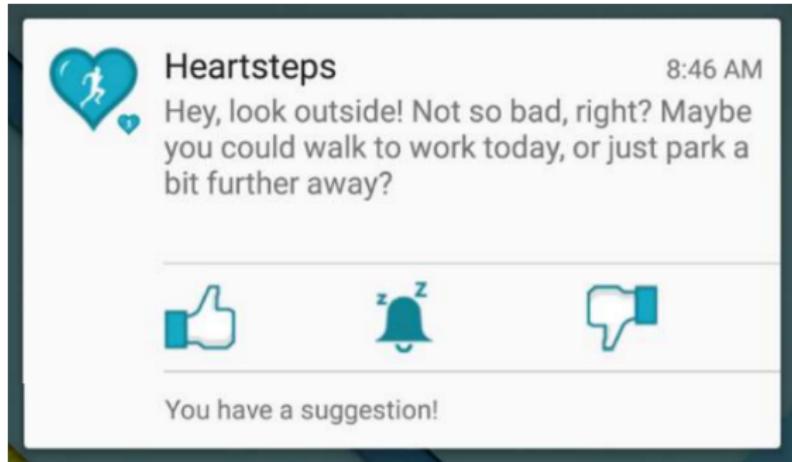


Figure: Personalized HeartSteps: A Reinforcement Learning Algorithm for Optimizing Physical Activity. Liao, Greenwald, Klasnja, Murphy 2019 arxiv

With great power there must also come – great responsibility  
–*Spiderman comics (though related comments appear in the French National Convention 1793, by Lamb 1817 & Churchill 1906)*

# Reinforcement Learning Involves

- Optimization
- Delayed consequences
- Exploration
- Generalization

# Optimization

- Goal is to find an optimal way to make decisions
  - Yielding best outcomes or at least very good outcomes
- Explicit notion of utility of decisions
- Example: finding minimum distance route between two cities given network of roads

# Delayed Consequences

- Decisions now can impact things much later...
  - Saving for retirement
  - Finding a key in video game Montezuma's revenge
- Introduces two challenges
  - When planning: decisions involve reasoning about not just immediate benefit of a decision but also its longer term ramifications
  - When learning: temporal credit assignment is hard (what caused later high or low rewards?)

# Exploration

- Learning about the world by making decisions
  - Agent as scientist
  - Learn to ride a bike by trying (and failing)
  - Finding a key in Montezuma's revenge
- Censored data
  - Only get a reward (label) for decision made
  - Don't know what would have happened if we had taken red pill instead of blue pill (Matrix movie reference)
- Decisions impact what we learn about
  - If we choose to go to Stanford instead of MIT, we will have different later experiences...

- Policy is mapping from past experience to action
- Why not just pre-program a policy?

# Generalization

- Policy is mapping from past experience to action
- Why not just pre-program a policy?

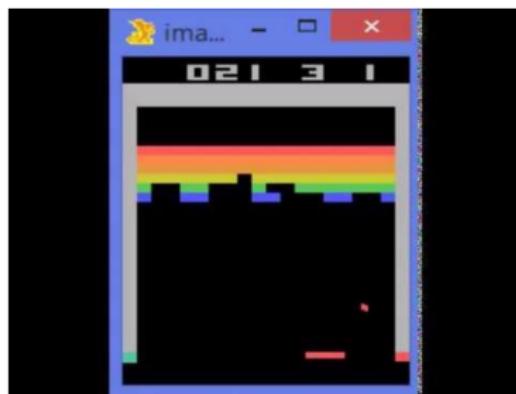


Figure: DeepMind Nature, 2015

- How many possible images are there?
  - $(256^{100 \times 200})^3$

# Reinforcement Learning Involves

- Optimization
- Exploration
- Generalization
- Delayed consequences

# RL vs Other AI and Machine Learning

	AI Planning	SL	UL	RL	IL
Optimization					
Learns from experience					
Generalization					
Delayed Consequences					
Exploration					

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning

# RL vs Other AI and Machine Learning

	AI Planning	SL	UL	RL	IL
Optimization	X				
Learns from experience					
Generalization	X				
Delayed Consequences	X				
Exploration					

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning
- AI planning assumes have a model of how decisions impact environment

# RL vs Other AI and Machine Learning

	AI Planning	SL	UL	RL	IL
Optimization	X				
Learns from experience		X			
Generalization	X	X			
Delayed Consequences	X				
Exploration					

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning
- Supervised learning is provided correct labels

# RL vs Other AI and Machine Learning

	AI Planning	SL	UL	RL	IL
Optimization	X				
Learns from experience		X	X		
Generalization	X	X	X		
Delayed Consequences	X				
Exploration					

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning
- Unsupervised learning is provided no labels

# RL vs Other AI and Machine Learning

	AI Planning	SL	UL	RL	IL
Optimization	X			X	
Learns from experience		X	X	X	
Generalization	X	X	X	X	
Delayed Consequences	X			X	
Exploration				X	

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning
- Reinforcement learning is provided with censored labels

## Sidenote: Imitation Learning

	AI Planning	SL	UL	RL	IL
Optimization	X			X	X
Learns from experience		X	X	X	X
Generalization	X	X	X	X	X
Delayed Consequences	X			X	X
Exploration				X	

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning
- Imitation learning assumes input demonstrations of good policies
- IL reduces RL to SL. IL + RL is promising area

# How Do We Proceed?

- Explore the world
- Use experience to guide future decisions

# Other Issues

- Where do rewards come from?
  - And what happens if we get it wrong?
- Robustness / Risk sensitivity
- We are not alone...
  - Multi-agent RL

# Break

# Today's Plan

- Overview of reinforcement learning
- **Course structure overview**
- Introduction to sequential decision making under uncertainty

# High Level Learning Goals\*

- Define the key features of RL
- Given an application problem how (and whether) to use RL for it
- Compare and contrast RL algorithms on multiple criteria
- \*For more detailed descriptions, see website

# Course Staff

- Instructor: Emma Brunskill
- CAs: Dilip Arumugam, Jean-Raymond Betterton, Yuqian Cheng, Ramtin Keramati, Haojun Li, Jasdeep Singh, Garrett Thomas, Woody Wang, and Andrea Zanette
- Additional information
  - Course webpage: <http://cs234.stanford.edu>
  - Schedule, Piazza (fastest way to get help), lecture slides
  - Prerequisites, grading details, late policy, see webpage

# Standing on the shoulders of giants...

- A key part of human progress is our ability to learn beyond our own experience
- Enormous variability in the effectiveness of education
- Practice, coupled with prompt feedback, is key
- Use some of our class time to provide opportunities for practice and feedback
- Huge body of evidence which supports that retrieval practice helps increase retention more than many other methods, and can support deep learning: "Refresh your understanding" exercises in many lectures

# Effective Practice Strategies for Learning Class Content

- Keep up with Refresh/Check your understanding exercises
- Do homework
- Attend office hours for help
- Attend problem sessions
- Do past quiz or exam problems for practice without referring to solutions

# Criteria for Doing Well in Class

- **All of you can succeed if you put in the effort**
- We, the class staff, and your fellow classmates, are here to help

# Class Structure

- Course content will be pre-recorded and available by the end of Sunday the week before class
- Monday class time: Optional CA-facilitated watch party for the first lecture will occur during normal class time
- Wednesday class time: Optional CA-lead session to go through worked examples about the material and also practice working on materials in breakouts with CA support
- Friday 6pm Pacific: This is generally when quizzes and homeworks will be due. All quizzes and homeworks will be submitted on gradescope.
- We will drop the lowest score of Quizzes 1-3.

# Learning in a Time of Covid

- I know the pandemic is hard for everyone, and for some of you, an extraordinarily hard time
- I have made some changes to the class to provide additional opportunities to engage in meaningful ways with the material and your classmates, based on discussions with other faculty and CAs about what has worked well in their large remote classes

# Break

# Today's Plan

- Overview of reinforcement learning
- Course logistics
- **Introduction to sequential decision making under uncertainty**

## Refresher Exercise: AI Tutor as a Decision Process

- Student initially does not know addition (easier) nor subtraction (harder)
- AI tutor agent can provide practice problems about addition or subtraction
- AI agent gets rewarded +1 if student gets problem right, -1 if get problem wrong
- Model this as a Decision Process. Define state space, action space, and reward model. What does the dynamics model represent? What would a policy to optimize the expected discounted sum of rewards yield?
- Write down your own answers (5 min) and then (if watching live) discuss in small breakout groups..

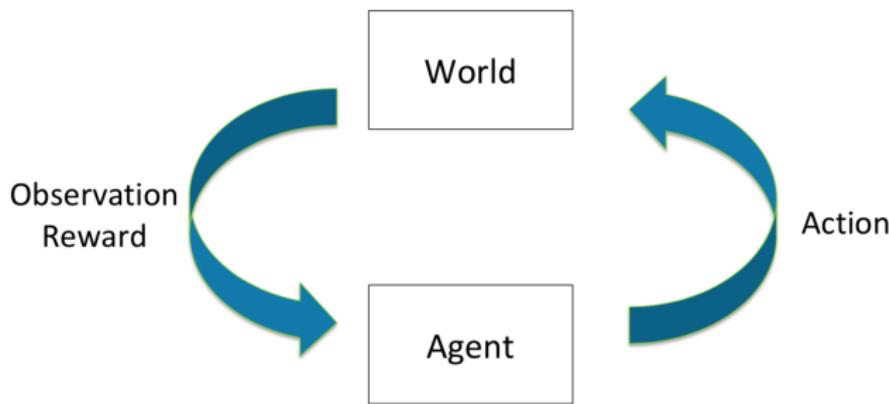
# Refresher Exercise: AI Tutor as a Decision Process

- State:
- Actions:
- Reward model:
- Meaning of dynamics model:

## Refresher Exercise: AI Tutor as a Decision Process

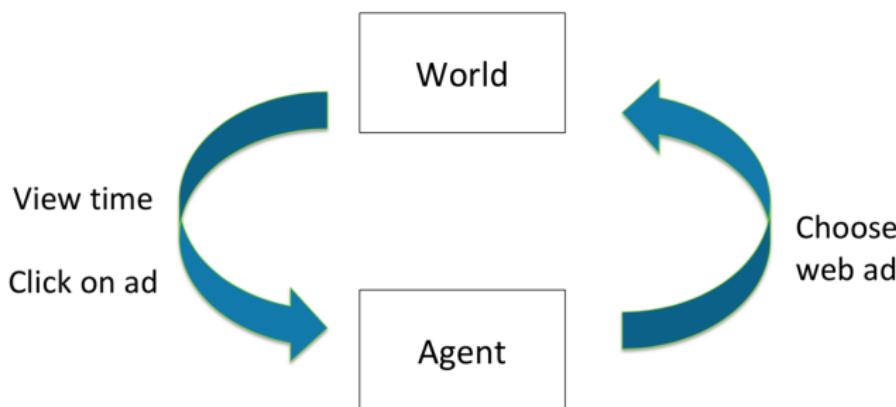
- Student initially does not know addition (easier) nor subtraction (harder)
- Teaching agent can provide activities about addition or subtraction
- Agent gets rewarded for student performance: +1 if student gets problem right, -1 if get problem wrong
- Which items will agent learn to give to max expected reward? Is this the best way to optimize for learning? If not, what other reward might one give to encourage learning?

# Sequential Decision Making



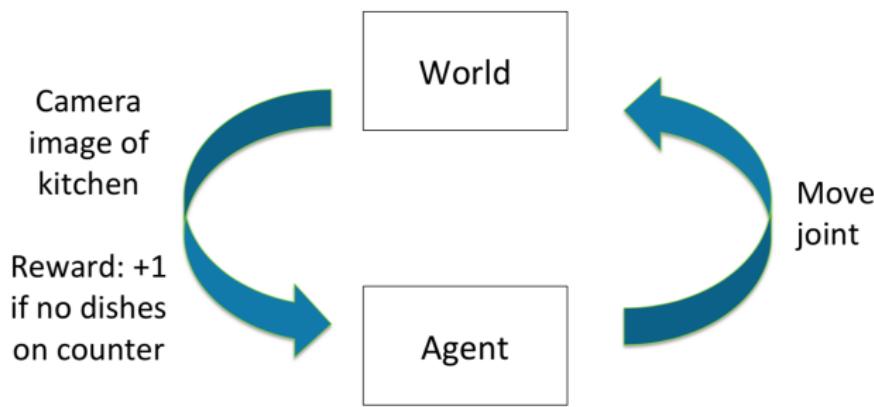
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

## Example: Web Advertising



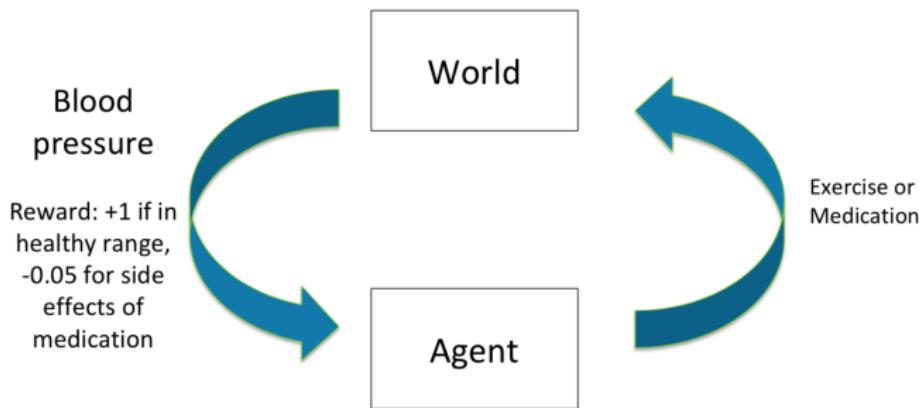
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

# Example: Robot Unloading Dishwasher



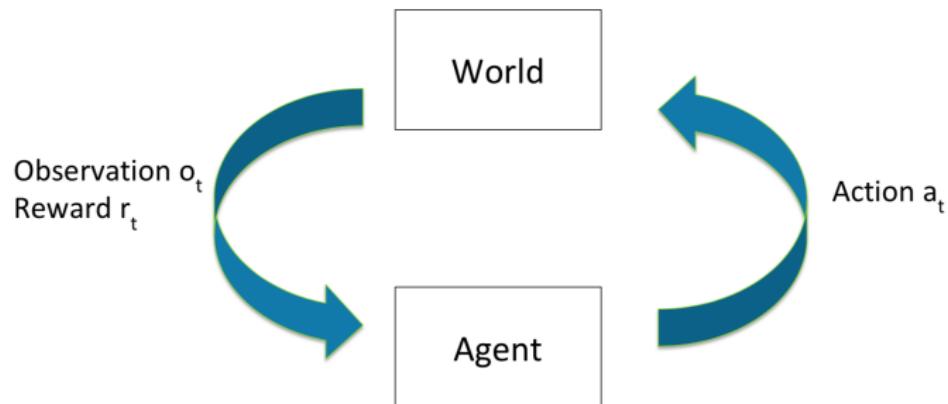
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

# Example: Blood Pressure Control



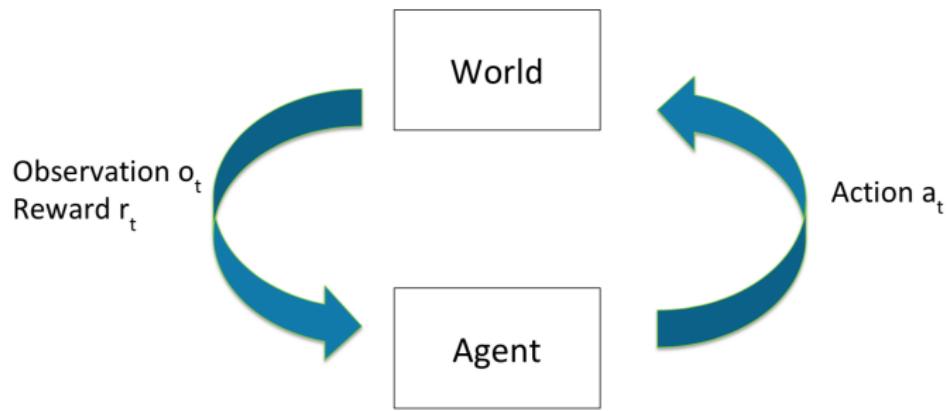
- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

# Sequential Decision Process: Agent & the World (Discrete Time)



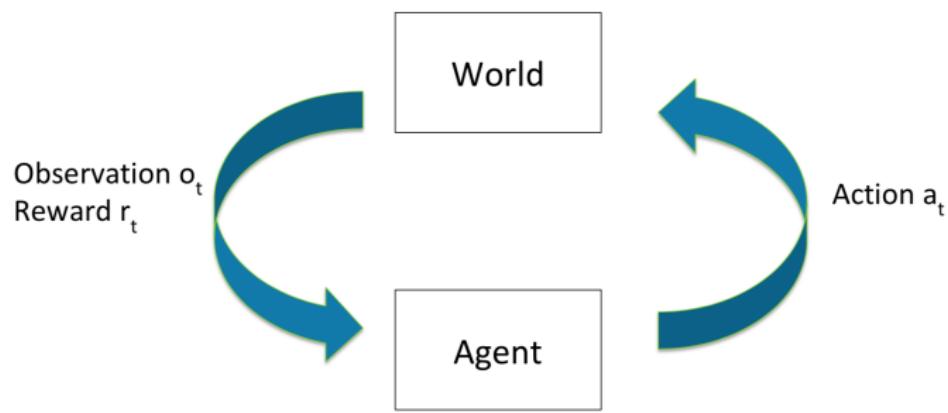
- Each time step  $t$ :
  - Agent takes an action  $a_t$
  - World updates given action  $a_t$ , emits observation  $o_t$  and reward  $r_t$
  - Agent receives observation  $o_t$  and reward  $r_t$

# History: Sequence of Past Observations, Actions & Rewards



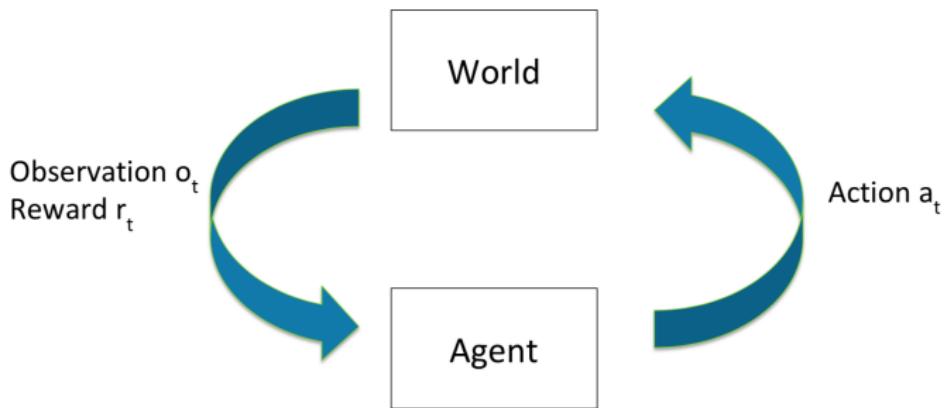
- History  $h_t = (a_1, o_1, r_1, \dots, a_t, o_t, r_t)$
- Agent chooses action based on history
- State is information assumed to determine what happens next
  - Function of history:  $s_t = (h_t)$

# World State



- This is true state of the world used to determine how world generates next observation and reward
- Often hidden or unknown to agent
- Even if known may contain information not needed by agent

# Agent State: Agent's Internal Representation



- What the agent / algorithm uses to make decisions about how to act
- Generally a function of the history:  $s_t = f(h_t)$
- Could include meta information like state of algorithm (how many computations executed, etc) or decision process (how many decisions left until an episode ends)

# Markov Assumption

- Information state: sufficient statistic of history
- State  $s_t$  is Markov if and only if:

$$p(s_{t+1}|s_t, a_t) = p(s_{t+1}|h_t, a_t)$$

- Future is independent of past given present

# Markov Assumption for Prior Examples

- Information state: sufficient statistic of history
- State  $s_t$  is Markov if and only if:

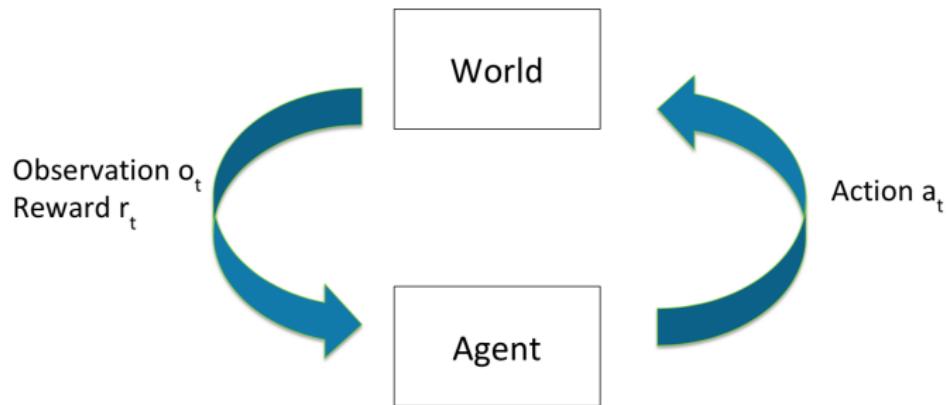
$$p(s_{t+1}|s_t, a_t) = p(s_{t+1}|h_t, a_t)$$

- Future is independent of past given present
- Hypertension control: let state be current blood pressure, and action be whether to take medication or not. Is this system Markov?
- Website shopping: state is current product viewed by customer, and action is what other product to recommend. Is this system Markov?

# Why is Markov Assumption Popular?

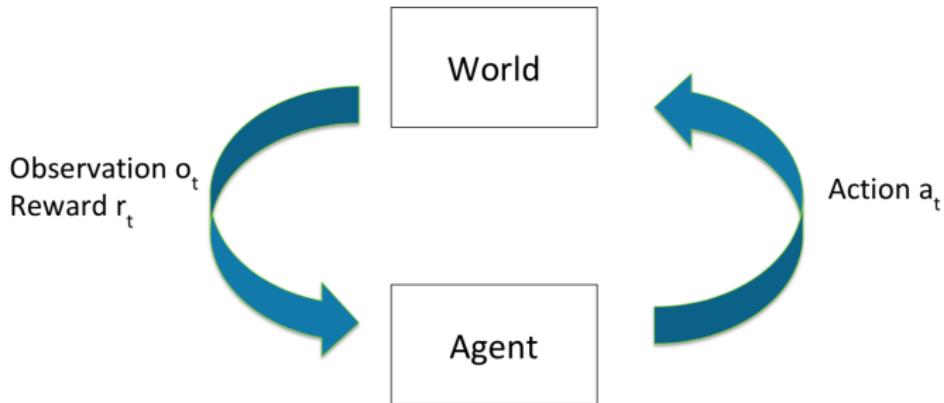
- Can always be satisfied
  - Setting state as history always Markov:  $s_t = h_t$
- In practice often assume most recent observation is sufficient statistic of history:  $s_t = o_t$
- State representation has big implications for:
  - Computational complexity
  - Data required
  - Resulting performance

# Full Observability / Markov Decision Process (MDP)



- Environment and world state  $s_t = o_t$

# Types of Sequential Decision Processes



- Is state Markov? Is world partially observable? (POMDP)
- Are dynamics deterministic or stochastic?
- Do actions influence only immediate reward or reward and next state?

# Example: Mars Rover as a Markov Decision Process

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$
						

Figure: Mars rover image: NASA/JPL-Caltech

- States: Location of rover ( $s_1, \dots, s_7$ )
- Actions: TryLeft or TryRight
- Rewards:
  - +1 in state  $s_1$
  - +10 in state  $s_7$
  - 0 in all other states

# RL Algorithm Components

- Often includes one or more of: Model, Policy, Value Function

# MDP Model

- Agent's representation of how world changes given agent's action
- Transition / dynamics model predicts next agent state

$$p(s_{t+1} = s' | s_t = s, a_t = a)$$

- Reward model predicts immediate reward

$$r(s_t = s, a_t = a) = \mathbb{E}[r_t | s_t = s, a_t = a]$$

# Example: Mars Rover Stochastic Markov Model

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$
$\hat{r} = 0$						

- Numbers above show RL agent's reward model
- Part of agent's transition model:
  - $0.5 = P(s_1|s_1, \text{TryRight}) = P(s_2|s_1, \text{TryRight})$
  - $0.5 = P(s_2|s_2, \text{TryRight}) = P(s_3|s_2, \text{TryRight}) \dots$
- Model may be wrong

# Policy

- Policy  $\pi$  determines how the agent chooses actions
- $\pi : S \rightarrow A$ , mapping from states to actions
- Deterministic policy:

$$\pi(s) = a$$

- Stochastic policy:

$$\pi(a|s) = Pr(a_t = a | s_t = s)$$

# Example: Mars Rover Policy

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$
						

- $\pi(s_1) = \pi(s_2) = \dots = \pi(s_7) = \text{TryRight}$
- Quick check: is this a deterministic policy or a stochastic policy?

# Value Function

- Value function  $V^\pi$ : expected discounted sum of future rewards under a particular policy  $\pi$

$$V^\pi(s_t = s) = \mathbb{E}_\pi[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \dots | s_t = s]$$

- Discount factor  $\gamma$  weighs immediate vs future rewards
- Can be used to quantify goodness/badness of states and actions
- And decide how to act by comparing policies

# Example: Mars Rover Value Function

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$
$V^\pi(s_1) = +1$	$V^\pi(s_2) = 0$	$V^\pi(s_3) = 0$	$V^\pi(s_4) = 0$	$V^\pi(s_5) = 0$	$V^\pi(s_6) = 0$	$V^\pi(s_7) = +10$

- Discount factor,  $\gamma = 0$
- $\pi(s_1) = \pi(s_2) = \dots = \pi(s_7) = \text{TryRight}$
- Numbers show value  $V^\pi(s)$  for this policy and this discount factor

# Types of RL Agents

- Model-based
  - Explicit: Model
  - May or may not have policy and/or value function
- Model-free
  - Explicit: Value function and/or policy function
  - No model

# RL Agents

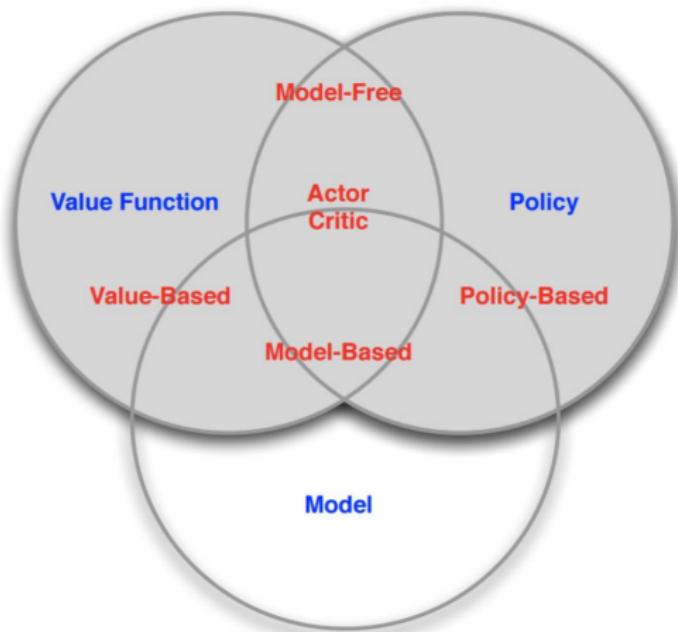
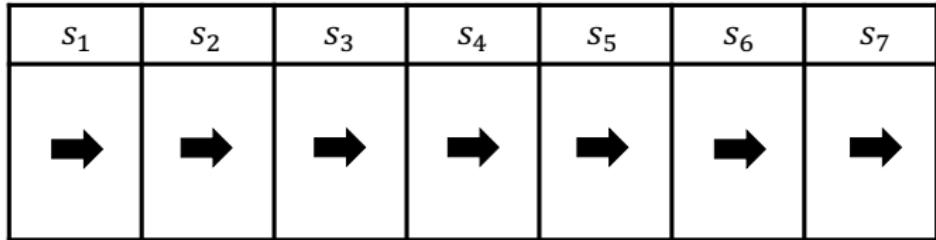


Figure: Figure from David Silver's RL course

# Evaluation and Control

- Evaluation
  - Estimate/predict the expected rewards from following a given policy
- Control
  - Optimization: find the best policy

# Example: Mars Rover Policy Evaluation



- $\pi(s_1) = \pi(s_2) = \dots = \pi(s_7) = \text{TryRight}$
- Discount factor,  $\gamma = 0$
- What is the value of this policy?

$$V^\pi(s_t = s) = \mathbb{E}_\pi[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots | s_t = s]$$

- Answer:

$$V^\pi(s_t = s) = r(s)$$

# Course Outline

- Markov decision processes & planning
- Model-free policy evaluation
- Model-free control
- Reinforcement learning with function approximation & Deep RL
- Policy Search
- Exploration
- Advanced Topics

See website for more details