# Lecture 1: Introduction to RL

Professor Emma Brunskill

CS234 RL

- Today the 3rd part of the lecture includes some 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

# Reinforcement Learning

Learning through experience/data to make good decisions under uncertainty

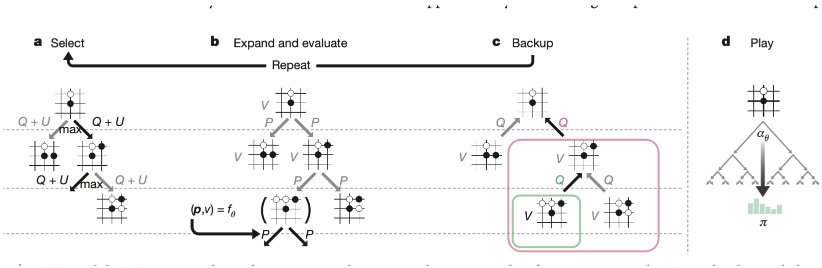
# Reinforcement Learning

- Learning through experience/data to make good decisions under uncertainty
- Essential part of intelligence
- Builds strongly from theory and ideas starting in the 1950s with Richard Bellman

# Reinforcement Learning

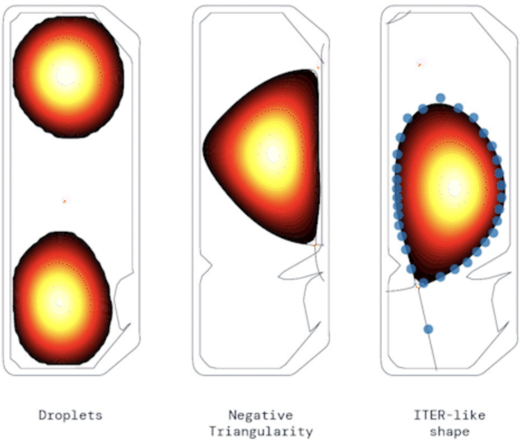
- Learning through experience/data to make good decisions under uncertainty
- Essential part of intelligence
- Builds strongly from theory and ideas starting in the 1950s with Richard Bellman
- A number of impressive successes in the last decade

# Beyond Human Performance on the Board Game Go[1]



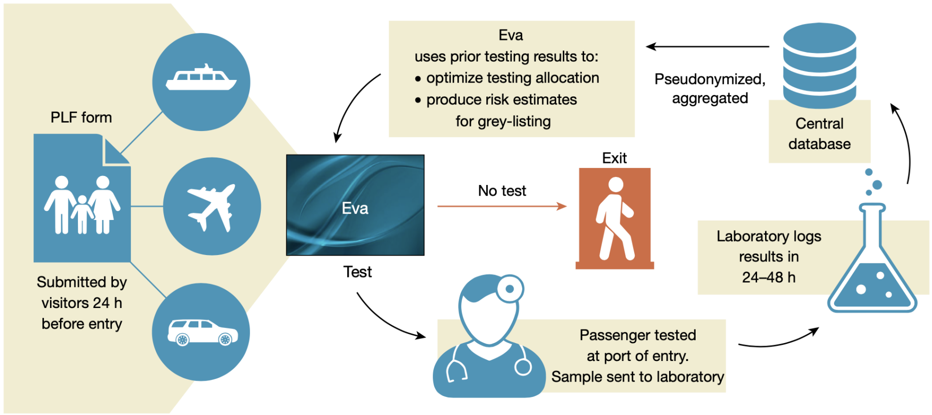

[1]Image credits: Silver et al. Nature 2017
https://www.nature.com/articles/nature24270

# Learning Plasma Control for Fusion Science[2]




[2] DeepMind & SPC/EPFL. Degrave et al. Nature 2022 https://www.nature.com/articles/s41586-021-04301-9

# Efficient and targeted COVID-19 border testing via RL [3]




[3] Bastani et al. Nature 2021
https://www.nature.com/articles/s41586-021-04014-z

# ChatGPT (https://openai.com/blog/chatgpt/ )

behavior cloning
imitation learning

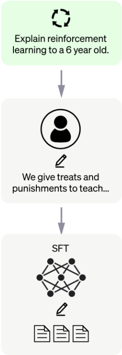
Step 1

**Collect demonstration data and train a supervised policy.**

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.




model of a reward
model based RL

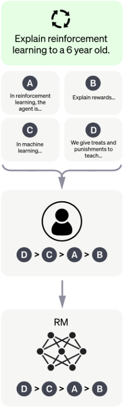
Step 2

**Collect comparison data and train a reward model.**

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.




reinforcement
learning
RLHF

Step 3

**Optimize a policy against the reward model using the PPO reinforcement learning algorithm.**

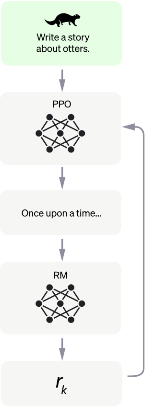
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

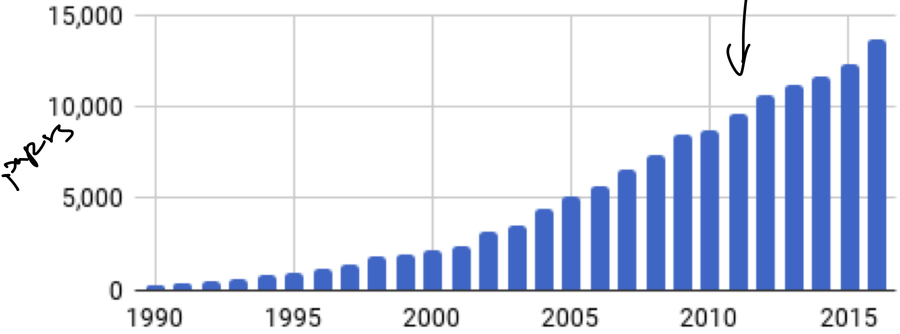
The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



# Huge Increase in Interest[4]

Atari




[4]Figure from Henderson et al. 2018 AAAI
https://arxiv.org/pdf/1709.06560.pdf

# But Also Contentious – How Much Information Does the Machine Need to Predict?

**"Pure" Reinforcement Learning (cherry)**
- The machine predicts a scalar reward given once in a while
- A few bits for some samples

**Supervised Learning (icing)**
- The machine predicts a category or a few numbers for each input
- Predicting human-supplied data
- 10 → 10,000 bits per sample

**Unsupervised/Predictive Learning (cake)**
- The machine predicts any part of its input for any observed part
- Predicts future frames in videos
- Millions of bits per sample

Adapted from Yann LeCun's presentation "A Path to AI"

[5] https://www.youtube.com/watch?v=0unt2Y4qxQo

# Reinforcement Learning (Generally) 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 decision utility
- 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?)

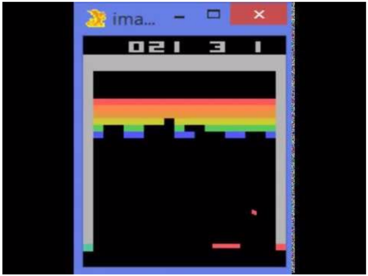
# Exploration

- Learning about the world by making decisions
  - Agent as scientist
  - Learn to ride a bike by trying (and failing)
- Decisions impact what we learn about
  - Only get a reward for decision made
  - Don't know what would have happened for other decision
  - If we choose to go to Stanford instead of MIT, we will have different later experiences...

# Generalization

- Policy is mapping from past experience to action
- Why not just pre-program a policy? 300

$256^{300 \times 400}$




400

Figure: DeepMind Nature, 2015

# 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 has access to the 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 has access to no labels

# 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 typically assumes input demonstrations of good policies
- IL reduces RL to SL. **For many good reasons, IL is very popular.**

# Two Problem Categories Where RL is Particularly Powerful

1. No examples of desired behavior: e.g. because the goal is to go beyond human performance or there is no existing data for a task.
2. Enormous search or optimization problem with delayed outcomes:

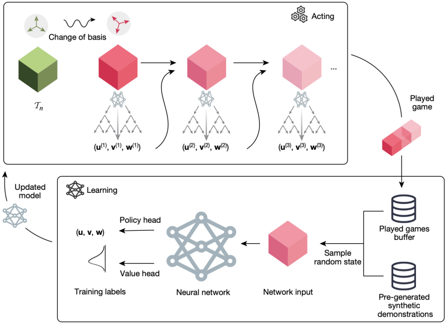



Figure: AlphaTensor. Fawzi et al. 2022

# Course Outline

- Markov decision processes & planning
- Model-free policy evaluation
- Model-free control
- Policy Search
- Offline RL **including RL from Human Feedback and Direct Preference Optimization**
- Exploration
- Advanced Topics

MCTS

# High Level Learning Goals[6]

- Define the key features of RL
- Given an application problem know how (and whether) to use RL for it
- Implement (in code) common RL algorithms
- Understand theoretical and empirical approaches for evaluating the quality of a RL algorithm

[6] For more detailed descriptions, see website

# 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 either 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 discuss in small breakout groups..

# Refresher Exercise: AI Tutor as a Decision Process

history (observ, question, reward... )

how good student is at add & subtr
(.9, .4)

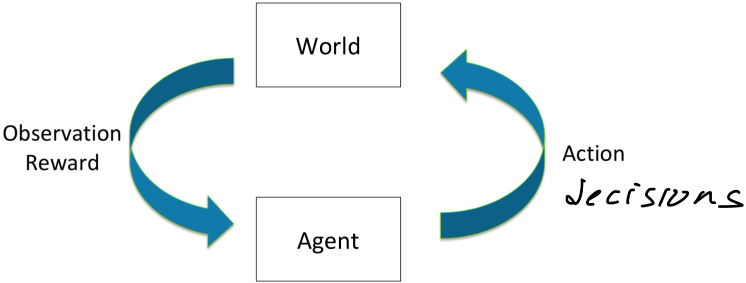
- State:
- Actions: addition question or subt
- Reward model: +1 if student gets right
- Meaning of dynamics model:

agent max its reward should only
give easy questions

# Refresher Exercise: AI Tutor as a Decision Process

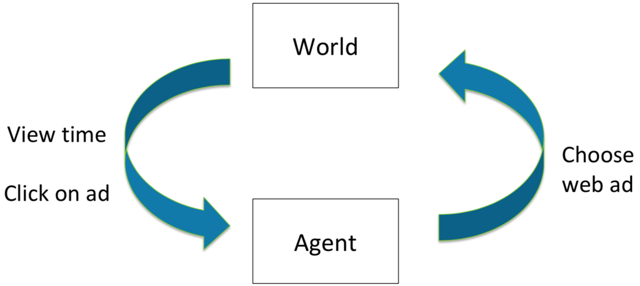
- Student initially does not know either addition (easier) nor subtraction (harder)
- Teaching agent can provide activities about addition or subtraction
- Agent gets rewarded for student performance: +1 if student gets problem right, -1 if get problem wrong
- Which items will agent learn to give to 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



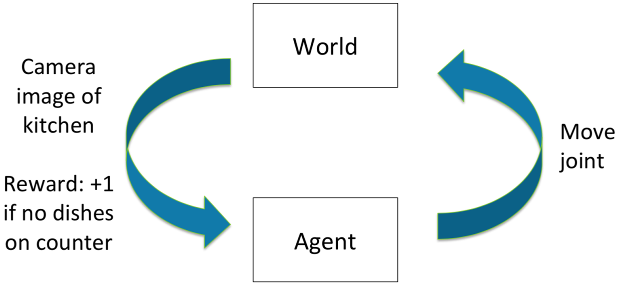

- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

# Example: Web Advertising



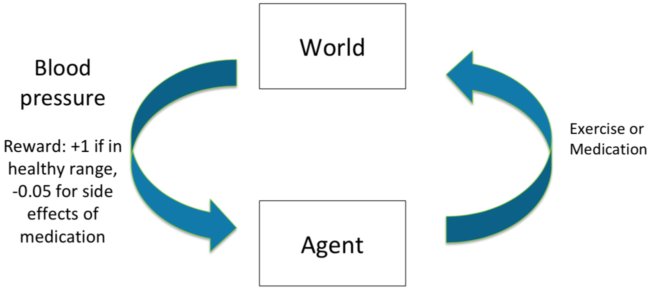

- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

# Example: Robot Unloading Dishwasher



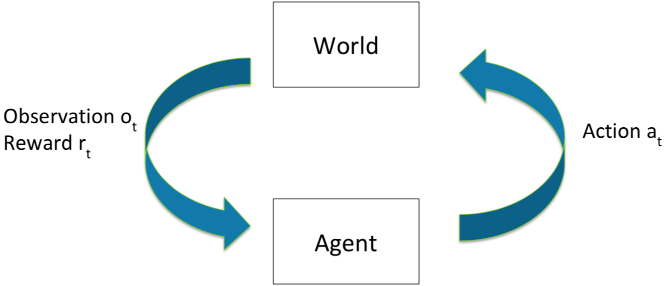

- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

# Example: Blood Pressure Control



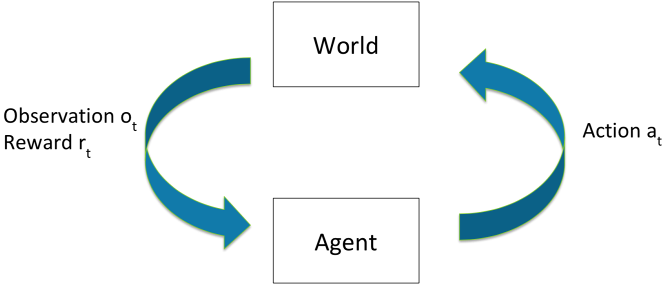

- Goal: Select actions to maximize total expected future reward
- May require balancing immediate & long term rewards

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