Introduction to Reinforcement Learning

Kevin Chen and Zack Khan

Outline

Course Logistics

Course Information and Resources

- Course website: cmsc389f.umd.edu (not ready yet)
- Piazza: piazza.com/umd/spring2018/cmsc389f
- Book (optional): Reinforcement Learning, an Introduction by Sutton & Barto, 2018

Prerequisites

Minimum Prerequisites: CMSC216 and CMSC250

Recommended Background:

- Basic Statistics
- Basic Python
- Familiarity with UNIX
- Interest in Reinforcement Learning!

Course Topics

For the full (tentative) schedule of topics, visit cmsc389f.umd.edu

Intuition Theory Application

Lecture 1: Introduction to Reinforcement Learning
Lecture 2: Reinforcement Learning Framework
Lecture 3: Markov Decision Processes
Lecture 4: OpenAI Gym and Universe
Lecture 5: Bellman Expectation Equations
Lecture 6: Optimal Policy through Policy and Value Iteration
Lecture 7: Policy Iteration and Value Iteration in Gridworld
Lecture 8: Model-Free Methods (Monte Carlo)
Lecture 9: Monte Carlo Prediction and Control
Lecture 10: Temporal Difference Learning
Lecture 11: SARSA and Q-Learning
Lecture 12: Value Function Approximation
Lecture 13: Linear Approximation in Mountain Car
Lecture 14: Deep Reinforcement Learning

Assignments

- Weekly problem sets
  - Short and simple
  - Graded on completion
  - Due 1 hour before class (email to cmsc389f@gmail.com)
- One final research project
  - Create an RL implementation or tackle a RL research problem
  - Write up a 3-6 page research paper
  - Focused on exploration, doesn’t need to be too complex

Grading

- Problem Sets: 50%
- Take-home Midterm: 20%
- Research Project: 30%

You'll Be Able To...

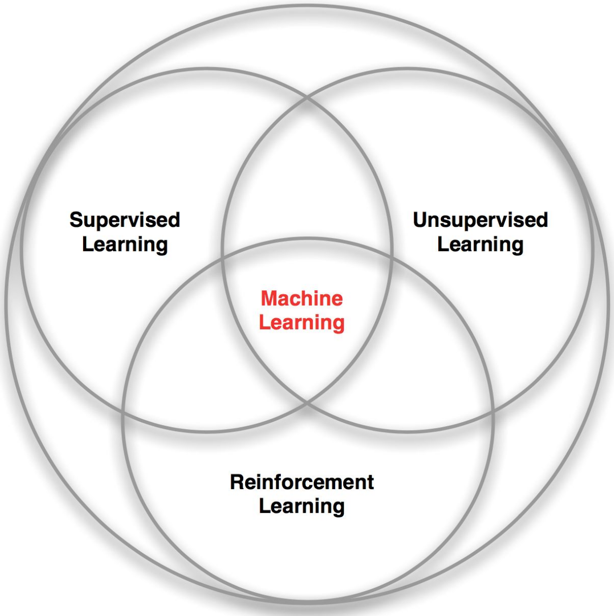
1. Understand modern RL research papers
2. Create your own RL AIs in a variety of games
3. Take further advanced machine learning classes

What is Reinforcement Learning?

Comparison with Other Methods

Three categories of machine learning:

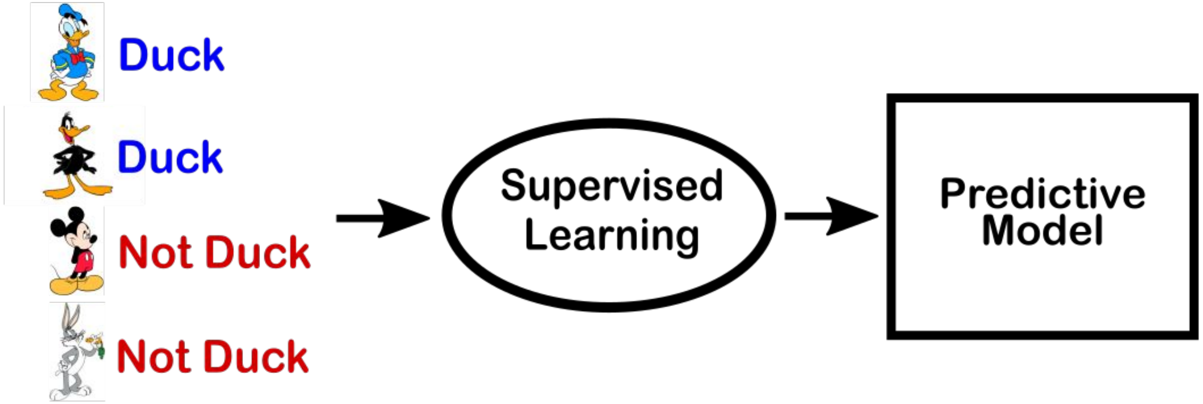
Reinforcement Learning	Supervised Learning	Unsupervised Learning



Silver (2017)

Comparison with Other Methods: Supervised Learning

Supervised Learning: learn a model (a function) to accurately classify data into categories.

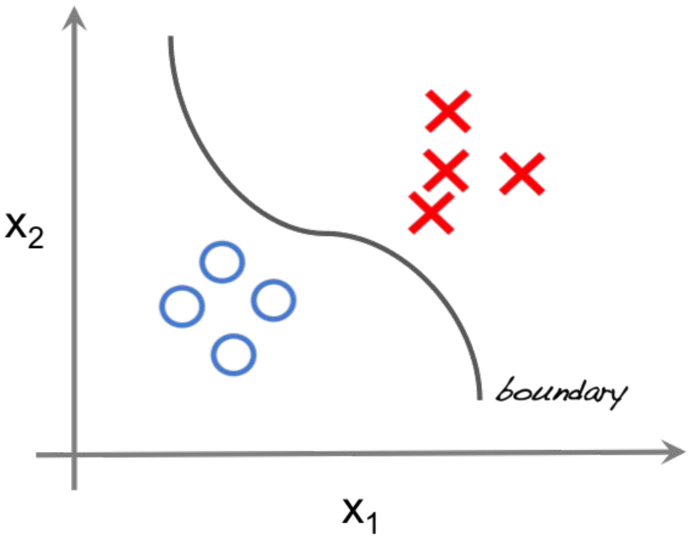


To learn this model, we **teach** our model using data that has already been correctly categorized.

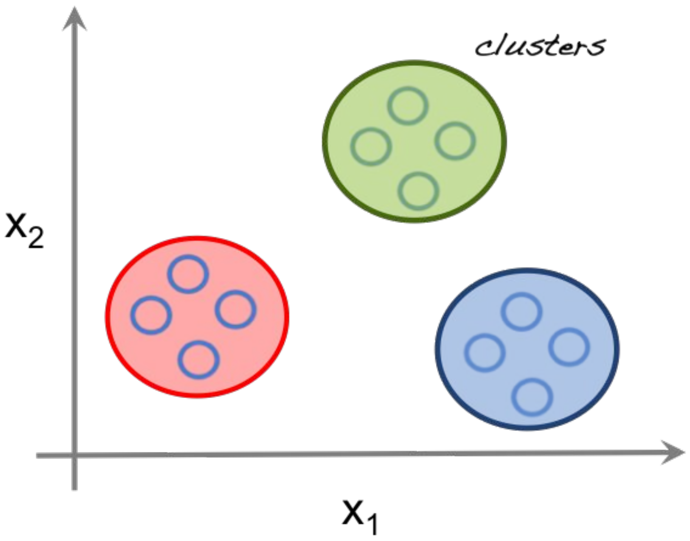
Comparison with Other Methods: Unsupervised Learning

Unsupervised Learning: finding structure and relationships within unlabelled datasets

Supervised learning



Unsupervised learning



Reinforcement Learning

Reinforcement Learning is an area of machine-learning that utilizes the concept of ***learning*** through ***interacting*** with a surrounding environment.

- Decision-making
- Goal-oriented learning

Example: Teaching a dog a trick

How can we teach a Fluffy a trick?

Example: Teaching a dog a trick

How can we teach a Fluffy a trick?

Give Fluffy treats!

Example: Teaching a dog a trick

How can we teach a Fluffy a trick?

Give Fluffy treats!

We teach Fluffy how to best behave in an environment, by giving him treats, so he knows how to adjust his behavior.

Example: Teaching a dog a trick

Takeaway 1: **We found a way of teaching Fluffy behavior!**

Example: Teaching a dog a trick

Takeaway 2: We're not explicitly telling Fluffy what to do.

Fluffy is *learning* what to do, based on *reward* that he encounters.

Example: Teaching a dog a trick

Question: How is Fluffy figuring out how to adjust his behavior based on the reward?

Example: Teaching a dog a trick

Idea: What if we make a software "Fluffy"?

Something that can learn in an environment on its own... (as long as there's reward)

Videos

1. How to Walk: https://www.youtube.com/watch?v=gn4nRCC9TwQ
2. Autonomous Stunt Helicopters: https://www.youtube.com/watch?v=VCdxqn0fcnE&t=5s

The Reinforcement Learning Problem

How should software agents take actions in an environment, to maximize cumulative reward?

Comparison with Other Methods: Overview

Reinforcement Learning	**Supervised Learning**	**Unsupervised Learning**
reward signal affects environment delayed feedback actions affect later data	supervisor doesn't affect environment instant feedback	no supervisor/reward doesn't affect environment no feedback

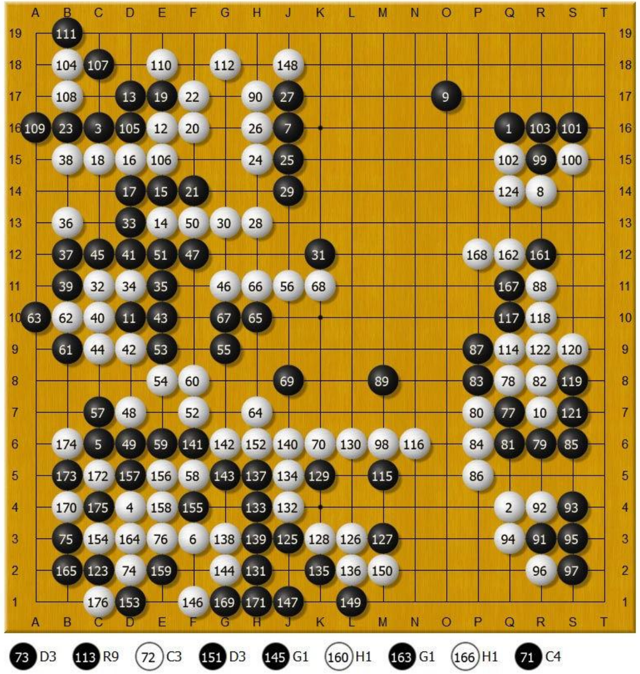
Comparison with Other Methods: Pros/Cons

Con: requires a huge amount of data, often more than Supervised Learning

Con: environments can be hard to describe

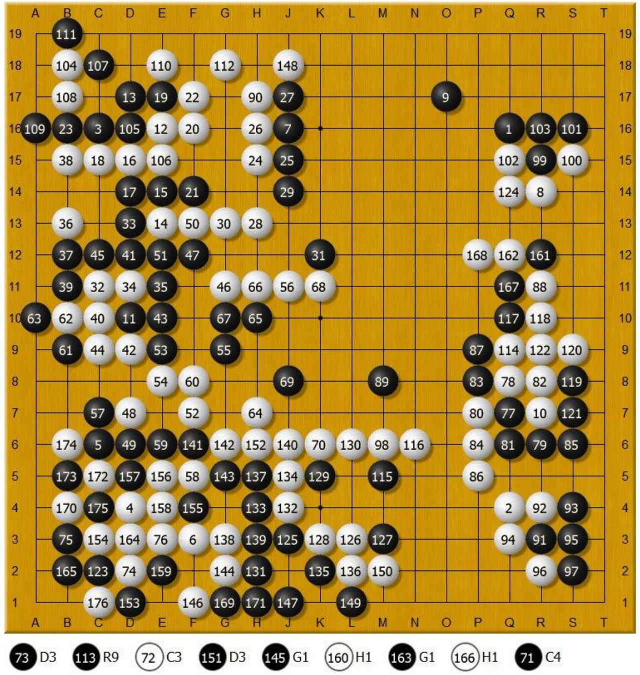
RL is useful when….

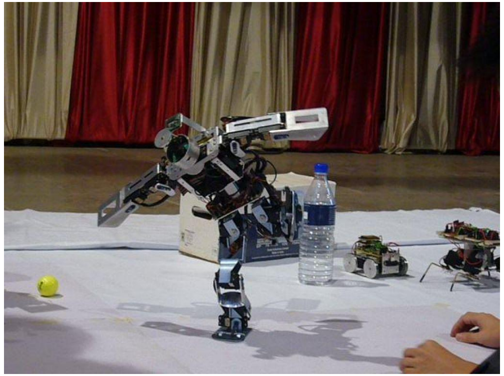
- We do not know the optimal actions to take
- We are dealing with large state spaces. (ex: Go)



Reward Hypothesis

Reward Hypothesis: We can formulate **any** goal as the maximization of some reward





Influences of Reinforcement Learning

Psychology: Law of Effect

"Of **several responses made to the same situation**, those which are **accompanied or closely followed by satisfaction** to the animal will, other things being equal, be more **firmly connected with the situation**, so that, **when it recurs, they will be more likely to recur...** The great the satisfaction or discomfort, the greater the strengthening or weakening of the bond." (Thorndike, 1911, p. 244)

Optimal Control

Finding a control law to achieve some optimality criterion in a system

- Related to reinforcement learning
- Richer history

ENGINEERING

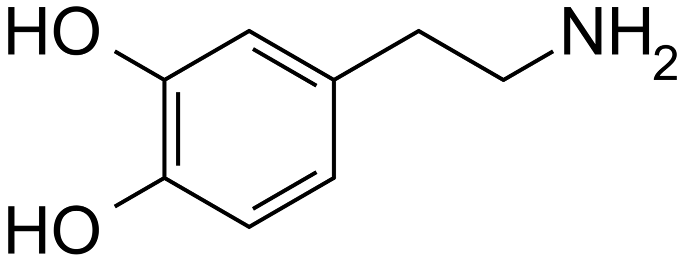
Example: Optimal Control

Example: Say Jim is driving back from I-270 after a long day of classes, and he wants to get home as fast as possible.

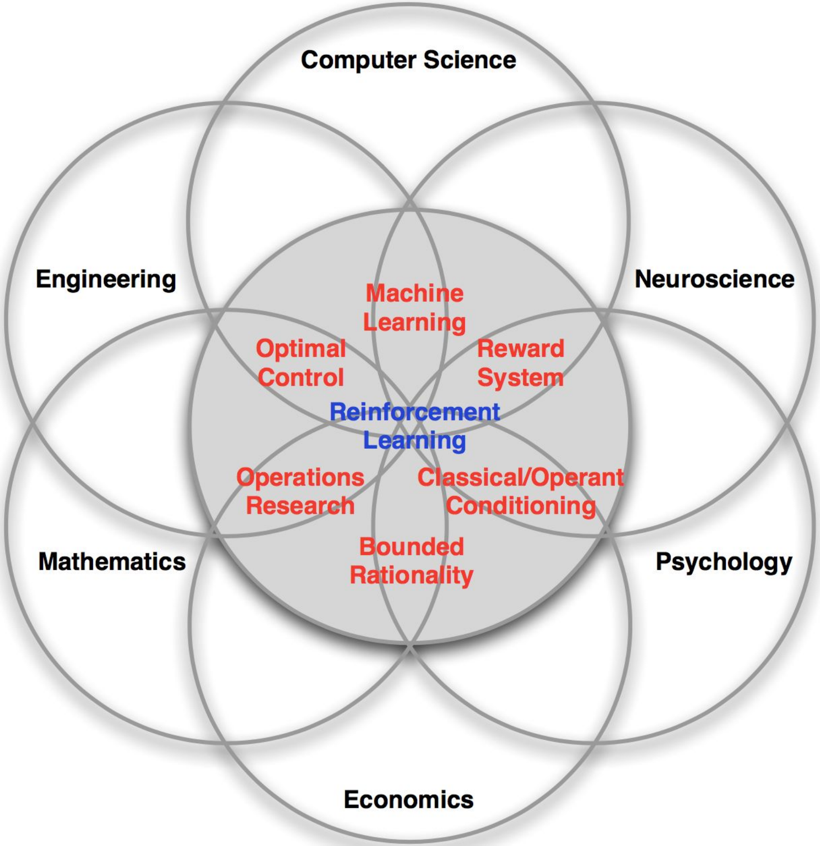
Problem: “How much should Jim accelerate to get home as fast as possible?”.
System: Jim and the road
Optimality criterion: minimization of the Jim’s travel time (under constraints)

Example: Animal Learning

Example: 5-year-old Jim walks into the kitchen. Little Jim sees a glowing red circle on the stove. Little Jim reaches out his hand and touches it. Ouch, that hurt! Little Jim decides to never touch the red-hot stove ever again.



Reinforcement Learning in Context



Silver (2017)

Why Study RL Now?

1. Computation Power
2. Deep Learning
3. New Ideas in Reinforcement Learning

Reinforcement Learning Today

- One of MIT Technology Review's "10 Breakthrough Technologies of 2017".
- Main driver of innovation behind industry titans such as Google DeepMind (AlphaGo), OpenAI (Video Games), and Tesla (Self-Driving Cars)

Examples of RL in the Real World

Google uses RL to decrease energy used in data centres by 40%, finding optimal conditions that optimize energy efficiency.

https://environment.google/projects/machine-learning/

More examples can be found at:
https://www.oreilly.com/ideas/practical-applications-of-reinforcement-learning-in-industry

Agent-environment Framework

Agent-environment Framework

IMPORTANT NOTE: There is ***no actual "learning"*** *described in this section.*
We are only ***setting up the framework*** *in which learning will occur.*

Agent and Environment

Two key parts of an RL system: **Agent** and **Environment**

Agents take **actions** within an environment

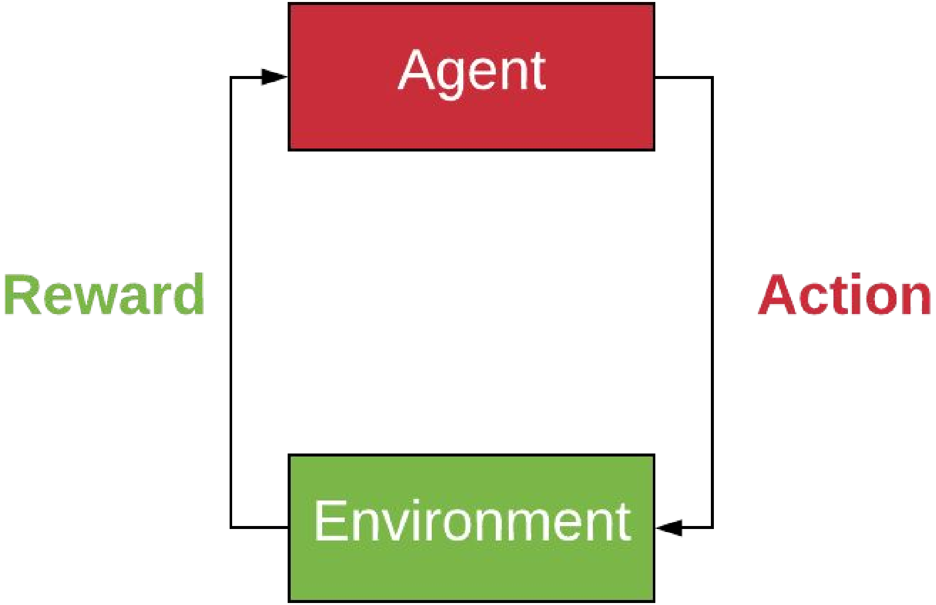
Environment responds to agent actions with **rewards** (or no reward)

Agent and Environment

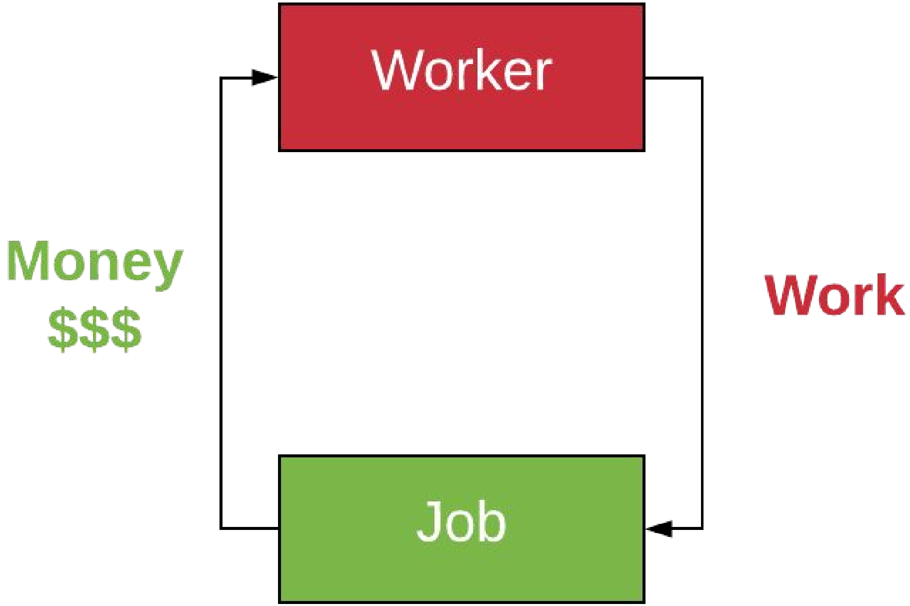
Two key parts of an RL system: **Agent** and **Environment**

Agents take **actions** within an environment

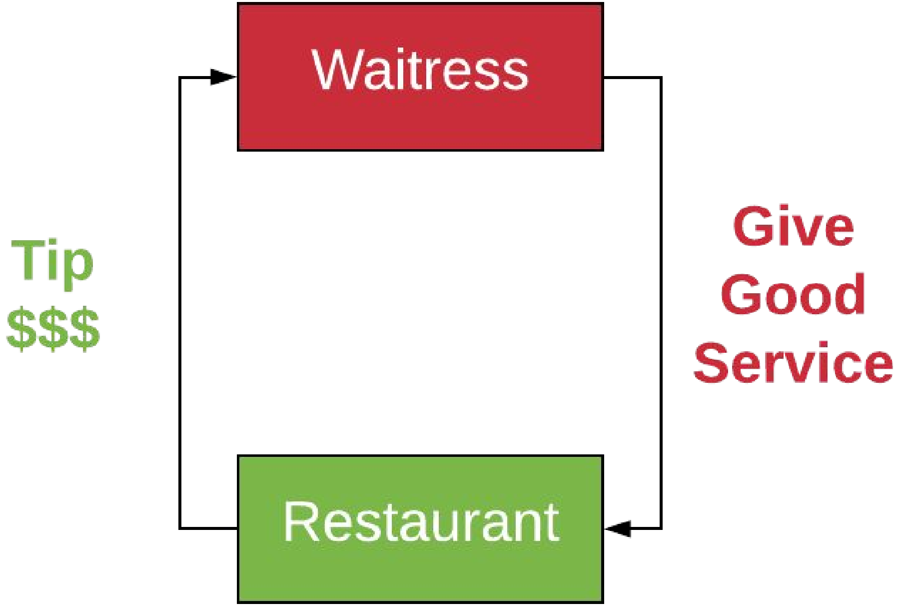
Environment responds to agent actions with **rewards** (or no reward)



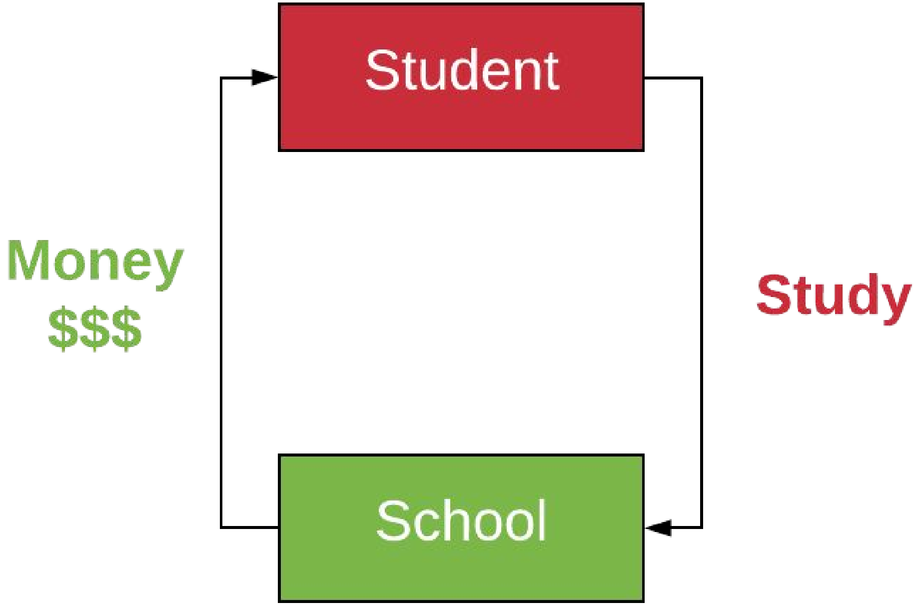
Example 1



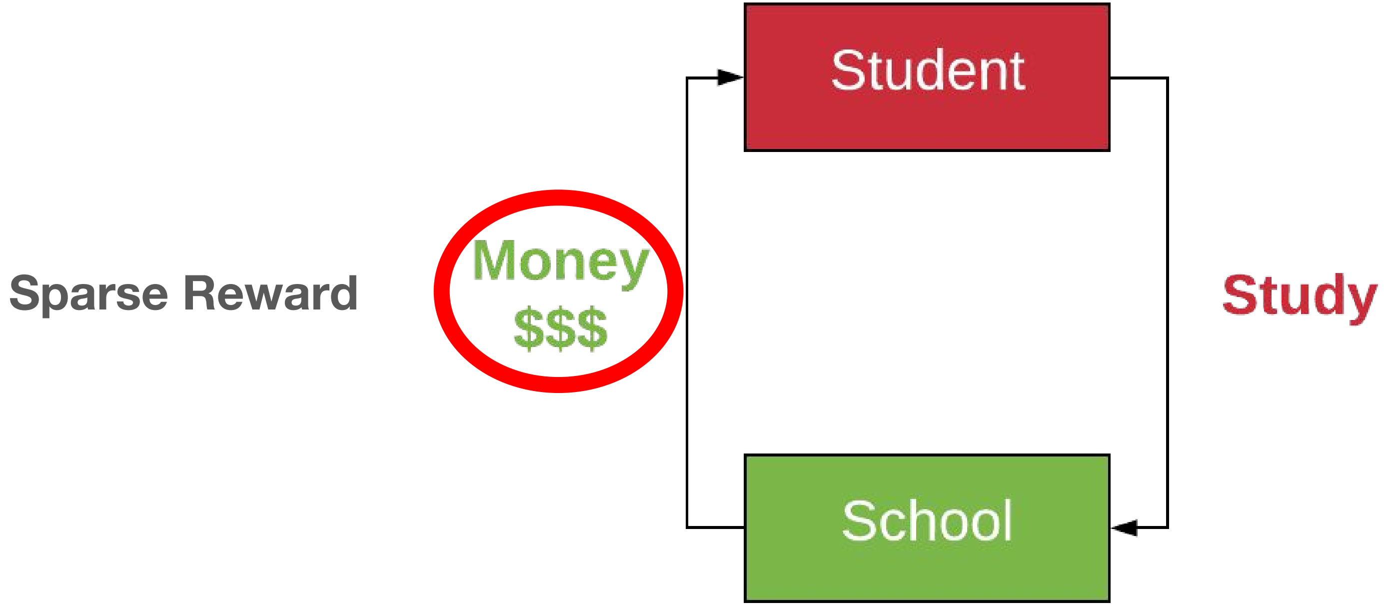
Example 2



Example 3

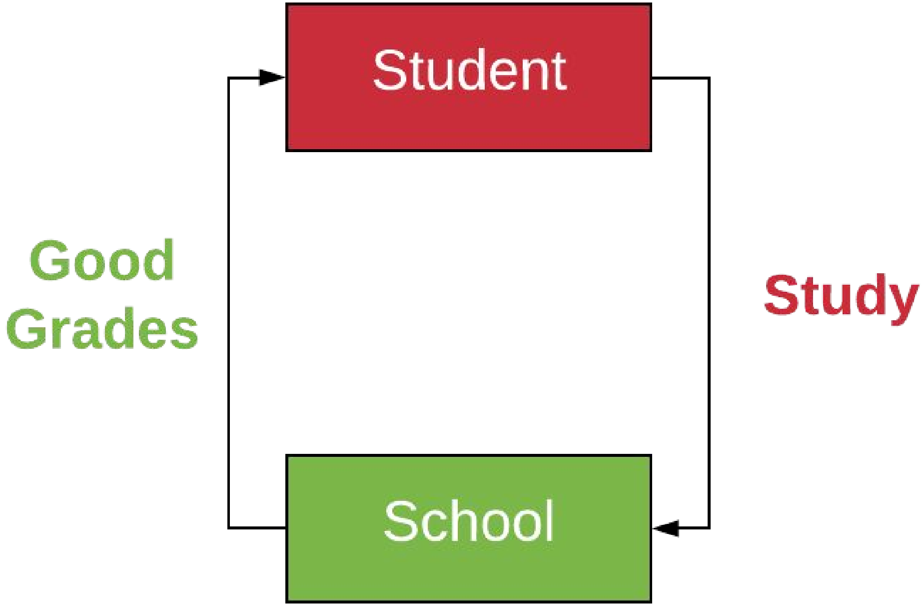


Example 3



Money is not rewarded until far in the future, too far for us to predict. Since we do not see this reward very often, we call this a Sparse Reward, which should be avoided

Example 4



Grades would be a more efficient reward as the rewards come in more frequently in relation to the action of studying

Agent-environment Framework II

Agent and Environment II

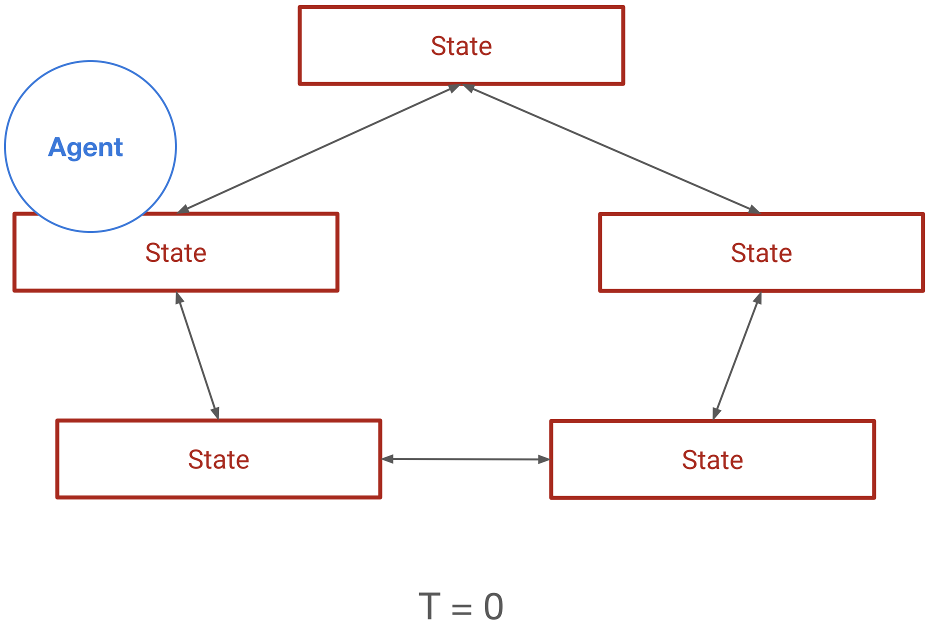
Environment can be represented as a set of **states** that the agent exists in.

When an agent takes an action, it will **move into a new state.**

Agent and Environment II

Environment can be represented as a set of **states** that the agent exists in.

When an agent takes an action, it will **move into a new state**, and receive a reward.

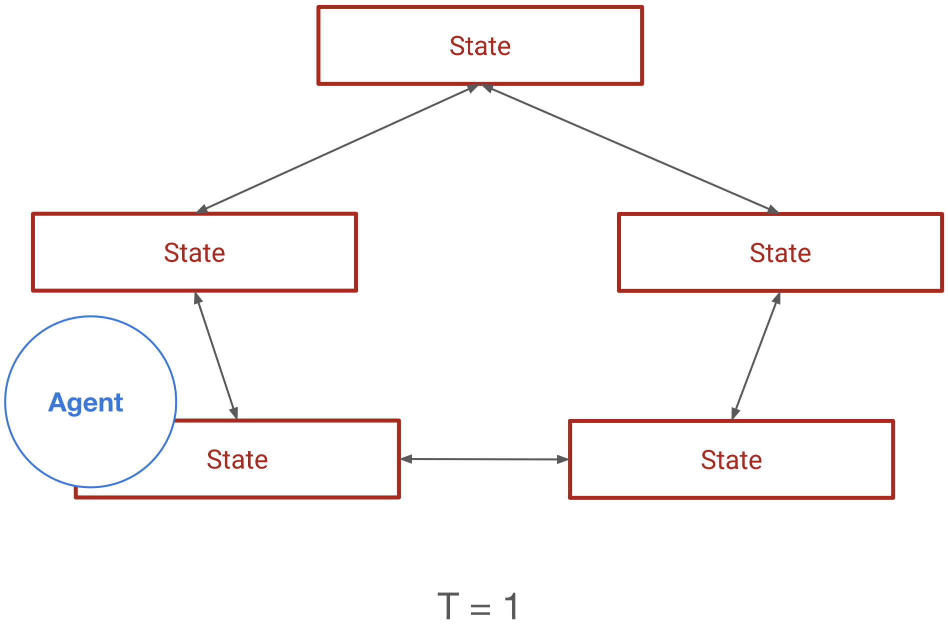


Agent and Environment II

Environment can be represented as a set of **states** that the agent exists in.

When an agent takes an action, it will **move into a new state**, and receive a reward.

To model time: after every action, time t increases by 1

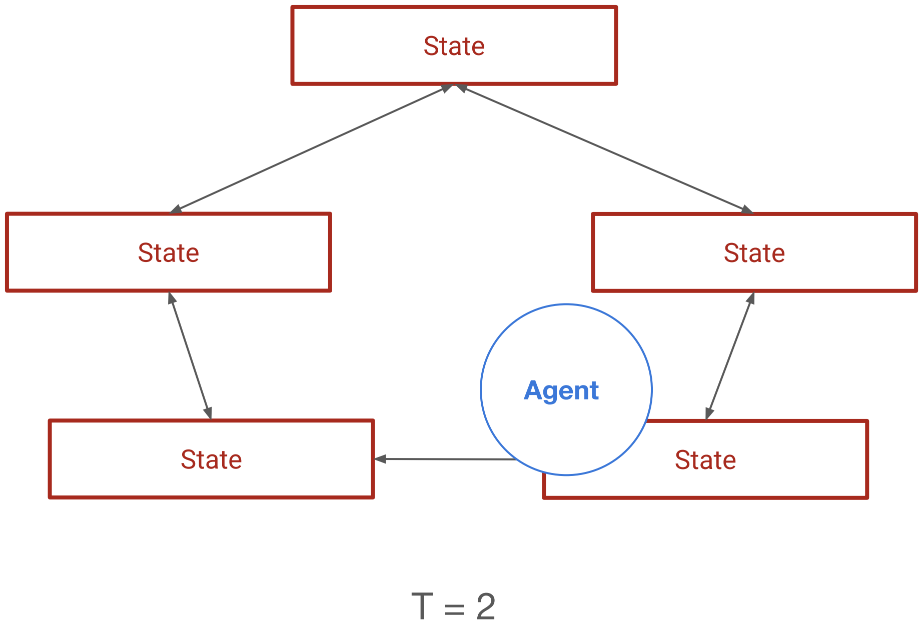


Agent and Environment II

Environment can be represented as a set of **states** that the agent exists in.

When an agent takes an action, it will **move into a new state**, and receive a reward.

To model time: after every action, time t increases by 1

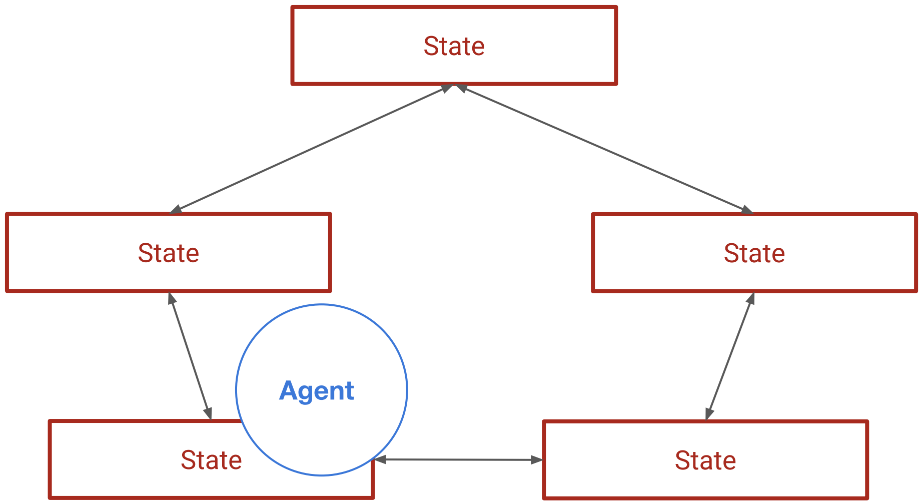


Agent and Environment II

Environment can be represented as a set of **states** that the agent exists in.

When an agent takes an action, it will **move into a new state**, and receive a reward.

To model time: after every action, time t increases by 1



T = 3

Agent Behavior

What if we tell the agent which **actions** to take, based on the **state** that they are in?

Agent Behavior

Example:

If the paddle is in a state where it is below the maximum height, take the “move up” action

Agent Behavior

Example:
If the paddle is in a state where it is below the maximum height, take the “move up” action

This is an AI!

Agent Behavior

Example:
If the paddle is in a state where it is below the maximum height, take the “move up” action

This is an AI! *(a really dumb one)*

Agent Behavior

Example 2:

If the paddle is in a state where it is **below** the ball, we say take the “move up” action

If the paddle is in a state where it is **above** the ball, we say take the “move down” action

Agent Behavior

Example 2:

If the paddle is in a state where it is **below** the ball, we say take the "move up" action

If the paddle is in a state where it is **above** the ball, we say take the "move down" action

This is also an AI! (a smart one)

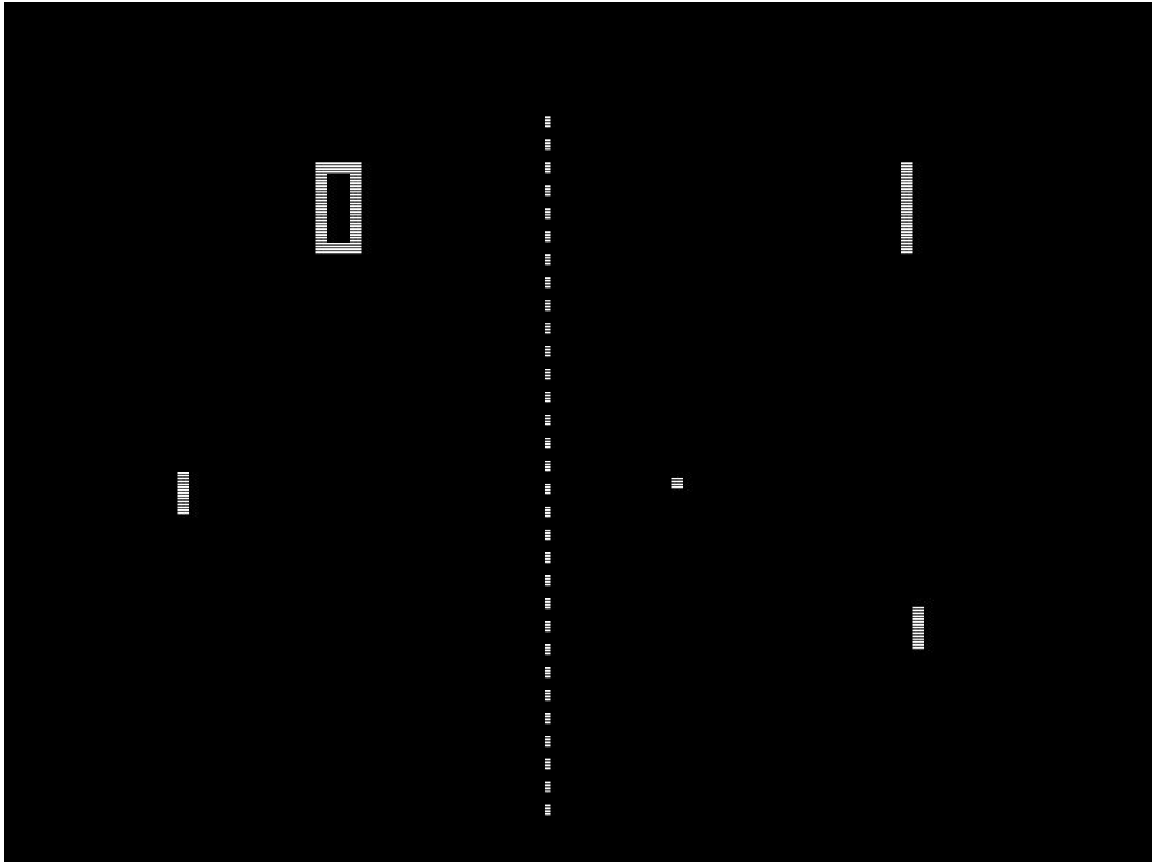
Agent Behavior

What if we tell the agent which **actions** to take, based on the **state** that they are in?
Answer: We get an AI!

What if we tell the agent which **actions** to take, based on the **state** that they are in, in such a way that those actions will result in **maximizing** reward?
Answer: We get a smart AI!

Figuring out how to do the above is what Reinforcement Learning is about!

Pong Example

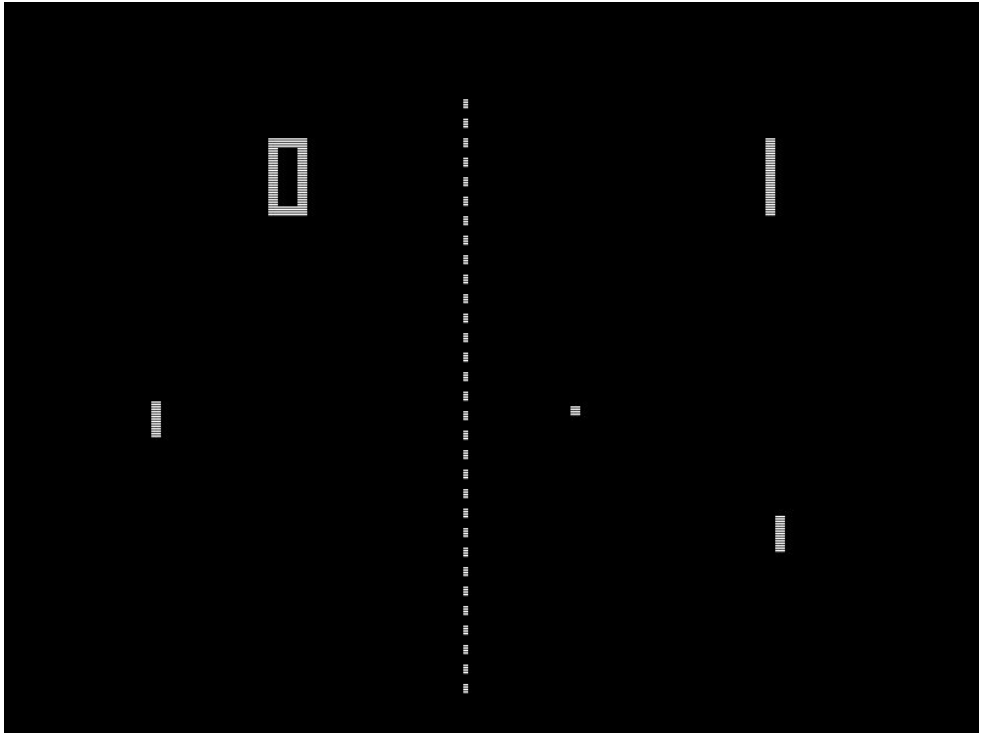


Pong Example

Environment: Pong Game (clock, game physics, etc)

Environment Reward: Scoring a Point

Goal: Winning the Game



Pong Example

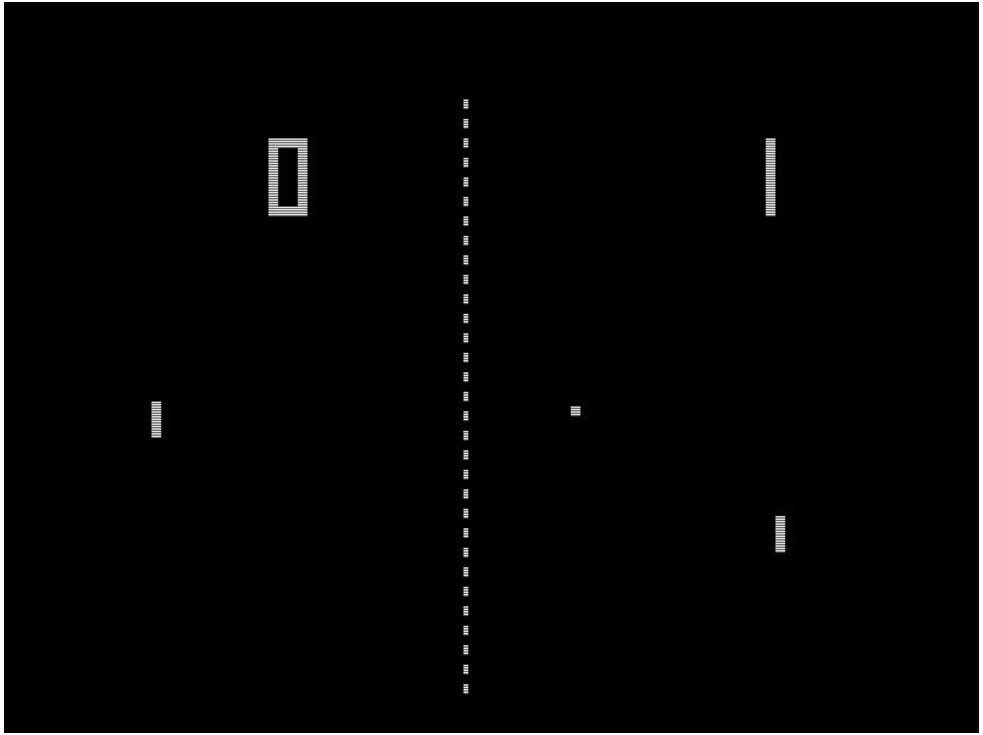
Environment: Pong Game (clock, game physics, etc)

Environment Reward: Scoring a Point

Goal: Winning the Game

Agent: Paddle

Agent Actions: Move up, Move down



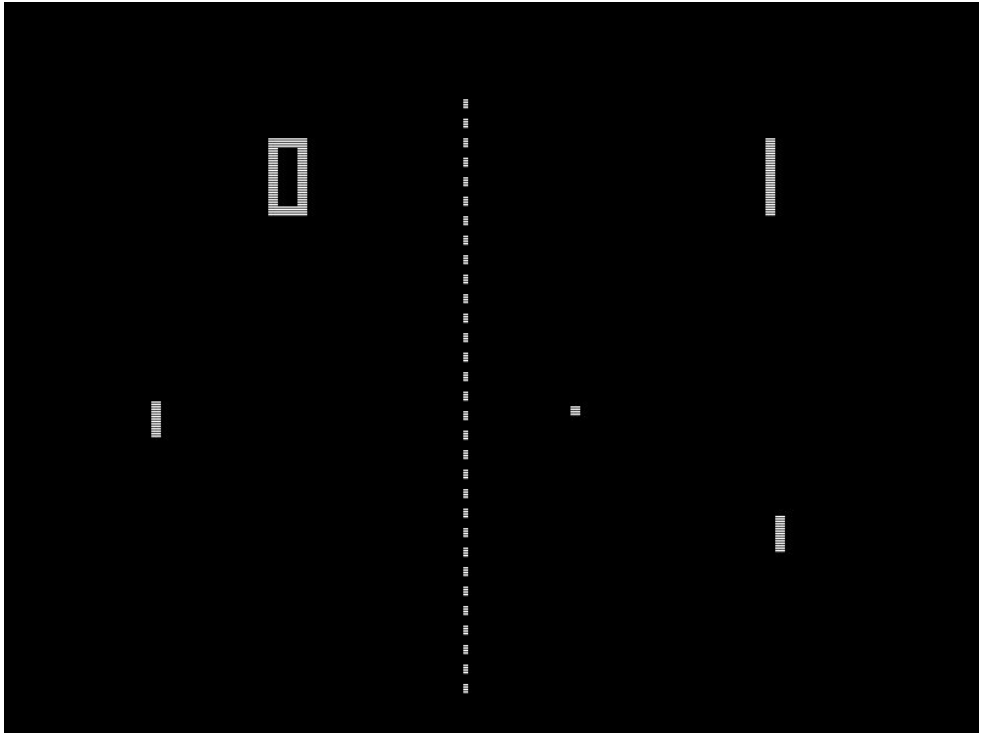
Agent and Environment

Goal of Reinforcement Learning: Figure out which actions the agent can take in the environment, to maximize some cumulative reward, in order to achieve a goal

Pong Example

Agent: "Move paddle up"

Environment: "Move paddle into new state"



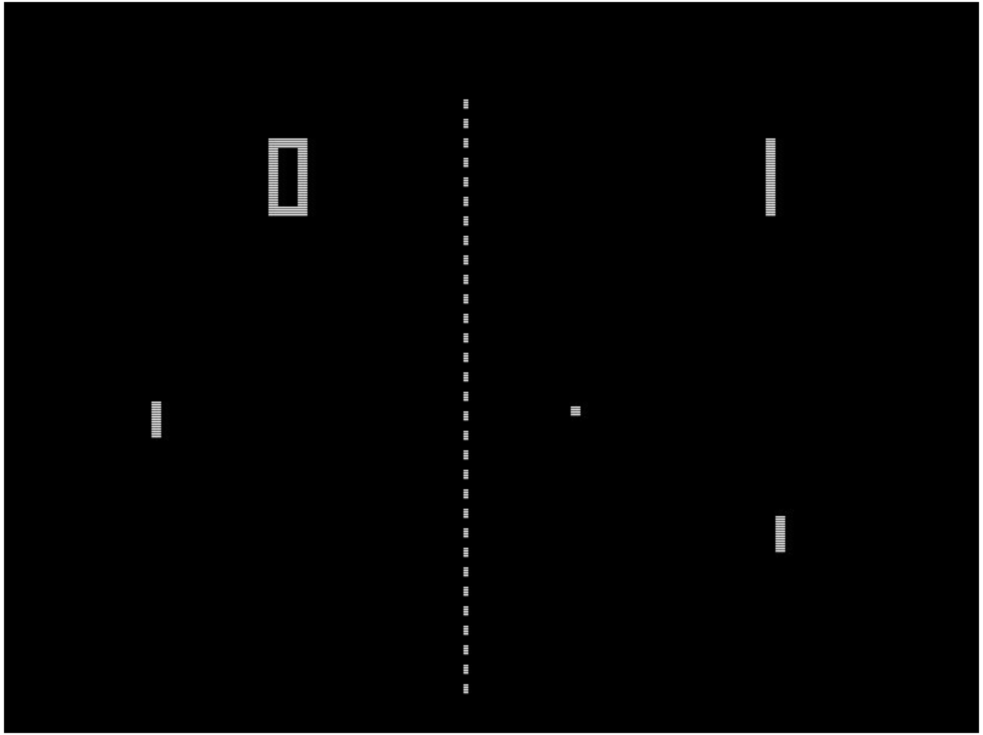
Pong Example

Agent: “Move paddle up”

Environment: “Move paddle into new state”

New State:

- One pixel above
- Time increases by 1



Pong Example

Example:
Paddle is in State 1: (height 6, time 0)

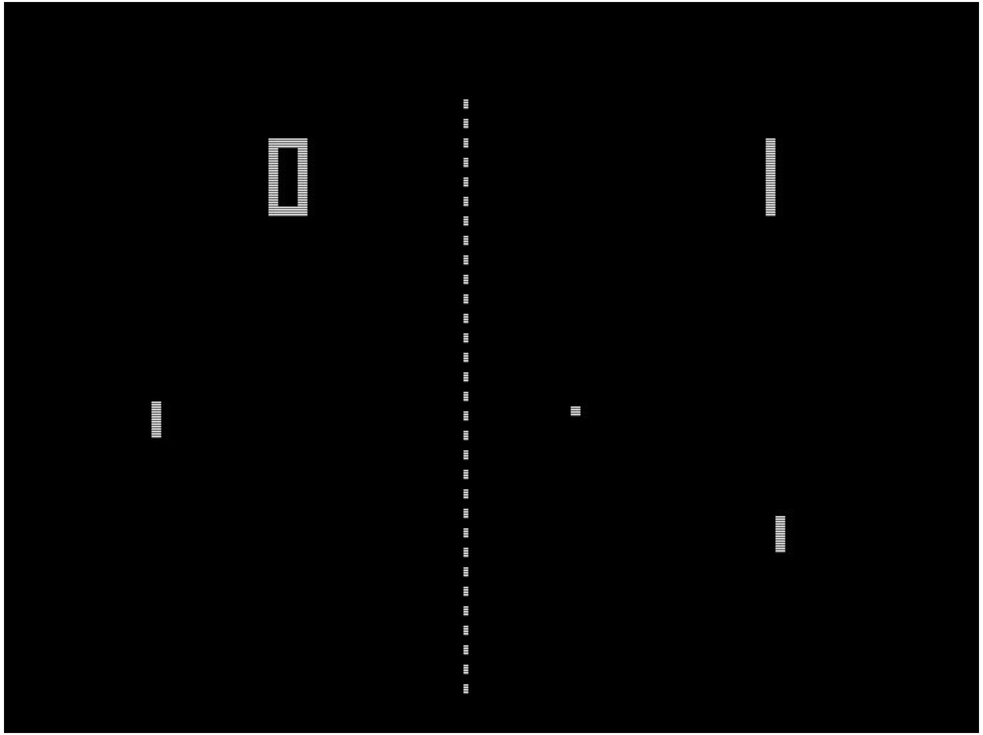
Paddle takes action: “Move up”
Environment moves Paddle to State 2

Paddle is in State 2: (height 7, time 1)

Paddle takes action: “Move down”
Environment moves Paddle to State 3

Paddle is in State 3: (height 6, time 2)

NOTE: State numbering is arbitrary



Summary

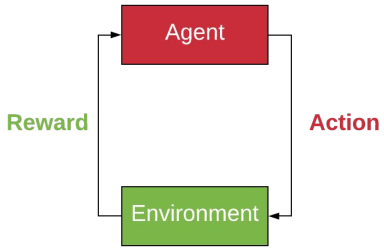
1. Reinforcement Learning (RL) is about an agent maximizing reward by interacting with its surrounding environment

2. RL has distinct advantages over other AI methods, but often requires more data or understanding of the problem/situation

3. Agents take **actions** within an environment. Environment responds with **rewards** (or no reward)

4. After an action, the agent moves into a new **state** of the environment

5. Figuring out **how to tell an agent what actions to take, in order to maximize reward**, is the key to reinforcement learning and creating a good AI



What's Next

Next week, we'll learn build on our understanding of the Reinforcement Learning Framework

Then, we'll start formalizing the concept of states, rewards, etc., mathematically

After that, we'll start to construct a solution for how to solve the Reinforcement Learning Problem

HOMEWORK:

Join Piazza!

Problem Set 1 is out on the website! Due by next class, send solutions to cmsc389f@gmail.com

Additional Resources

Machine Learning at Maryland

- Undergraduate Journal Club (Feb. 7th, 6:00pm, Location: TBD)

Machine Learning Faculty

- Computer Vision Department, Computational Linguistics (CLIP) Department, etc