Deep Reinforcement Learning

Lecture 1: Introduction

John Schulman

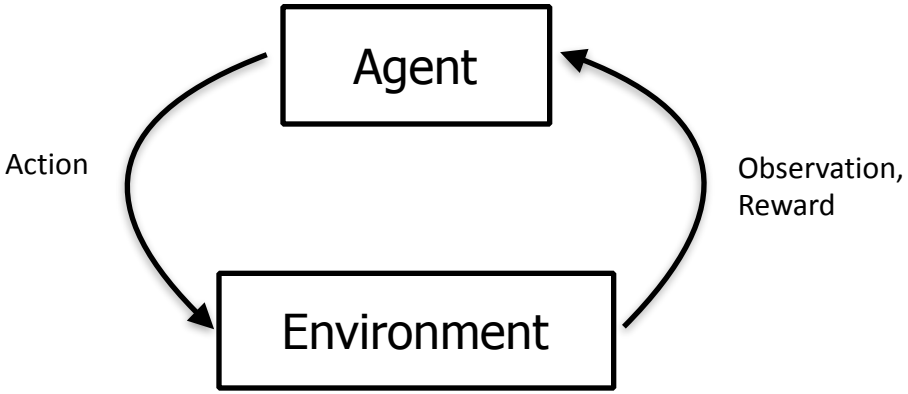
Goal of the Course

- Understand how deep reinforcement learning can be applied in various domains
- Learn about three classes of RL algorithm and how implement with neural networks
  - policy gradient methods
  - approximate dynamic programming
  - search + supervised learning
- Understand the state of deep RL as a research topic

Outline of Lecture

- What is “deep reinforcement learning”
- Where is reinforcement learning deployed?
- Where is reinforcement learning NOT deployed?
  (but could be…)

Sequential Decision Making



Goal: maximize expected total reward

with respect to the ***policy***: *a function from observation history to next action*

Applications

- Robotics:
  - Actions: torque at joints
  - Observations: sensor readings
  - Rewards:
    - navigate to target location

Applications

- Robotics:
  - Actions: torque at joints
  - Observations: sensor readings
  - Rewards:
    - navigate to target location
    - complete manipulation task

Applications

- Business operations
  - Inventory management: how much to purchase of inventory, spare parts
  - Resource allocation: e.g. in call center, who to service first
  - Routing problems: e.g. for management of shipping fleet, which trucks/truckers to assign to which cargo

Applications

- Finance
  - Investment decisions
  - Portfolio design
  - Option/asset pricing

Applications

- E-commerce / media
  - What content to present to users (using click-through / visit time as reward)
  - What ads to present to users (avoiding ad fatigue)

Applications

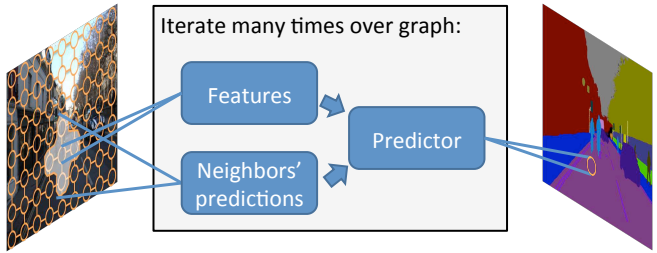
- Medicine
  - What tests to perform, what treatments to provide

Applications

- Structured prediction: algorithm has to make a sequence of predictions, which are fed back into predictor
  - in NLP, text generation & translation, parsing [1,2]
  - multi-step pipelines in vision [3]

[1] Daumé, Hal, et al..*Search-based structured prediction* (2009)

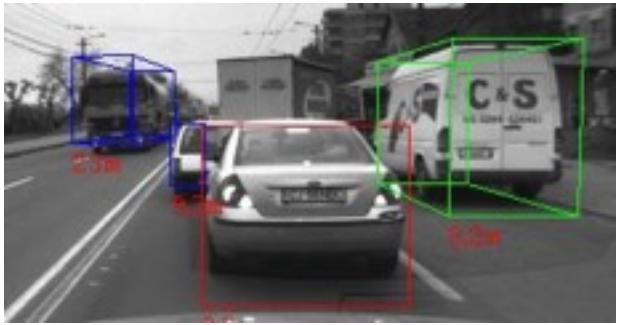
[2] Shi, T et al., *Learning Where to Sample in Structured Prediction,* (2015)



[3] S. Ross, *Interactive Learning for Sequential Decisions and Predictions*, 2013

RL vs Other Learning Problems

- Supervised learning: classification / regression
  - given observation, predict label, maximize reward function R(observation, label)



object detection

speech recognition

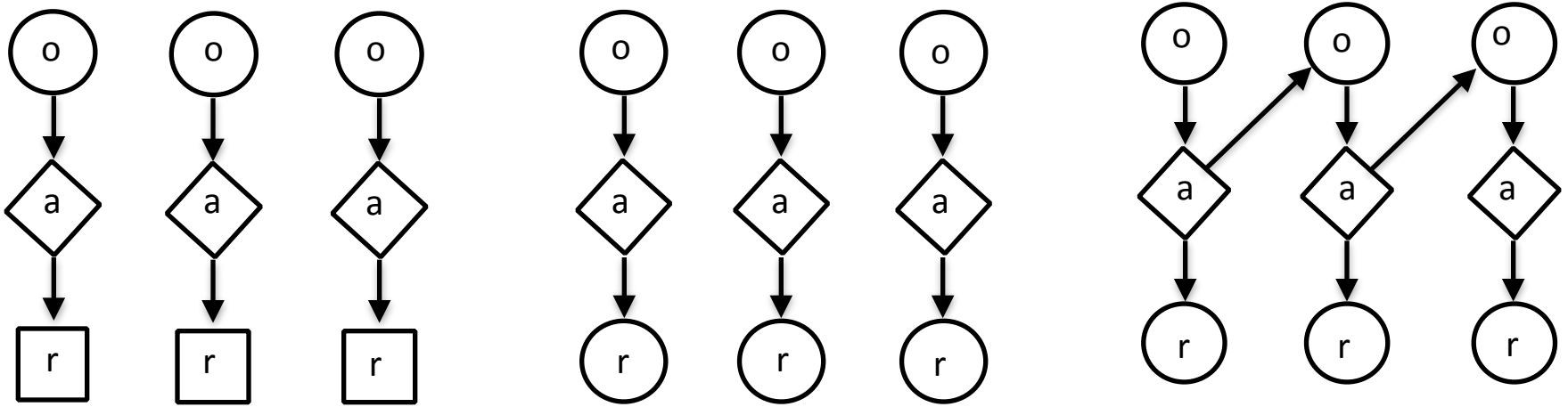
RL vs Other Learning Problems

- Contextual Bandits
  - given observation, output action, receive reward, with unknown and stochastic dependence on action and observation
  - e.g., advertising

RL vs Other Learning Problems

- Reinforcement learning
  - given observation, output action, receive reward, with unknown and stochastic dependence on action and observation
  - AND we perform a sequence of actions, and states depend on previous actions

RL vs. Other Learning Problems



Supervised learning ⊂ Contextual bandits ⊂ Reinforcement learning

deterministic node

decision node

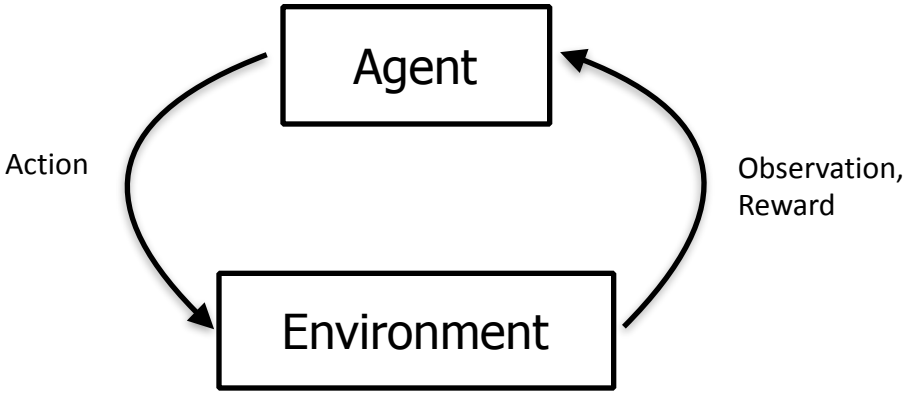
stochastic node

How is RL different from Supervised Learning, In Practice?

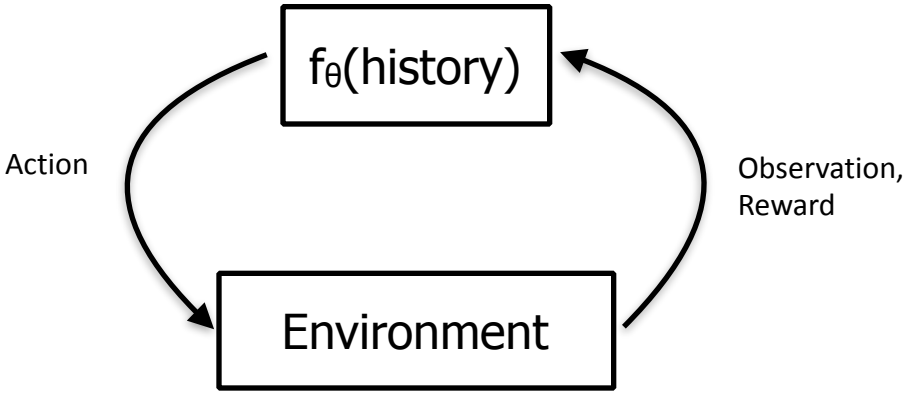
- State distribution is affected by policy
  - Need for exploration
  - Leads to instability in many algorithms

- Can’t use past data — online learning is not straightforward

What is “Deep RL”?

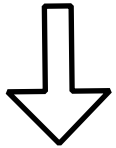


What is “Deep RL”?



Deep RL: Algorithm Design Criteria

- Algorithm learns a parameterized function f_θ
- Algorithm does not depend on parameterization, just that loss is differentiable wrt θ
- Optimize using gradient-based algorithms, using gradient estimators $\nabla_\theta \text{Loss}$



- computational complexity is linear in θ
- sample complexity is (in a sense) independent of θ

Nonlinear/Nonconvex Learning

- Supervised learning: just an unconstrained minimization of differentiable objective
  - minimize_θ Loss(X_{train}, y_{train})
  - easy to get convergence to local minimum
- Reinforcement learning: no differentiable objective to optimize!
  - actual objective E[total reward] is an expectation over random variables of unknown system
  - Approximate Dynamic Programming methods e.g. Q-learning: NOT gradient descent on fixed objective, NOT guaranteed to converge

Deep RL Allows Unified Treatment of Problem Classes

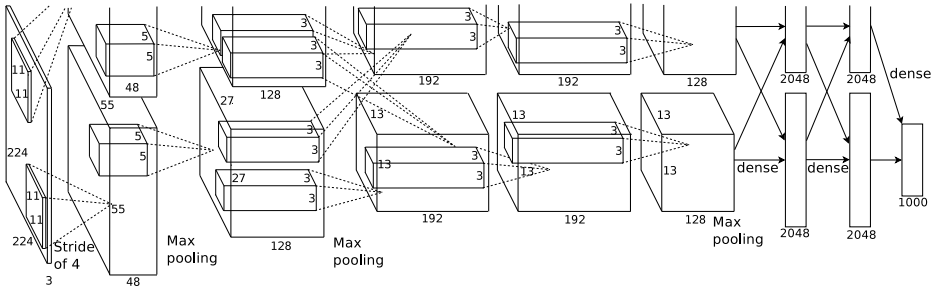
- No difference between Markov Decision Process (MDP) and Partially Observed Markov Decision Process (POMDP)
- Not much difference between discrete and continuous state/action setting
- No difference between finite-horizon, infinite horizon discounted, and average-cost setting
  - we're always just ignoring long-term dependencies

Deep RL Frontier

- Opportunity for theoretical / conceptual advances
  - How to explore state space
  - How to have a policy that involves actions with different timescales, or has subgoals (hierarchy)
  - How to combine reinforcement learning with unsupervised learning

Deep RL Frontier

- Opportunity for empirical/engineering advances
  - Pre-2012, object recognition state-of-the-art used hand-engineered features + learned linear classifiers + hand-engineered grouping mechanism
  - Now entire computer vision field uses deep neural networks for feature extraction, and moving towards end-to-end optimization of entire pipeline



[KSH2012] Krizhevsky, Sutskever, & Hinton., *ImageNet Classification with Deep Convolutional Neural Networks*, 2012

Where is RL Deployed Today

- Operations research (see, e.g., [1])
  - Inventory / storage
    - Power grid: when to buy new transformers. Each costs $5M, but failure leads to much bigger costs
    - How much of items to purchase and keep in stock
  - Resource allocation
    - Fleet management: assign cargos to truck drivers, locomotives to trains
    - Queueing problems: which customers to serve first in call center

RL in Robotics

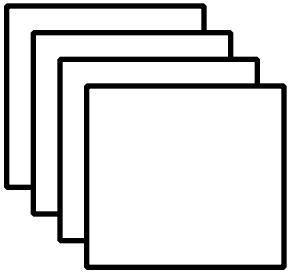
- Most industrial robotic systems perform a fixed motion repeatedly with simple or no perception.

Automotive body paint line.
(Courtesy of Kawasaki Robotics (USA) Inc.)

- *Iterative Learning Control* [1] is used in some robotic systems — using model of dynamics, correct errors in trajectories. But these systems still use simple or no perception

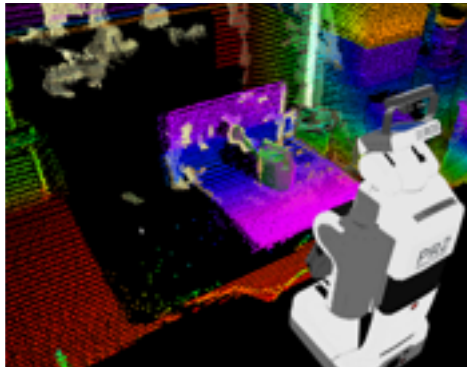
[1] Bristow, Douglas, Marina Tharayil, and Andrew G. Alleyne. *A survey of iterative learning control*

Classic Paradigm for Vision-Based Robotics



Sensor data
images / lidar

state estimation,
integration

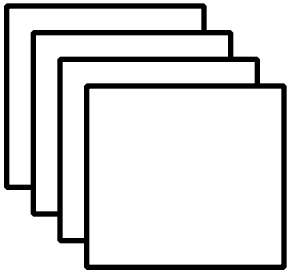


World model

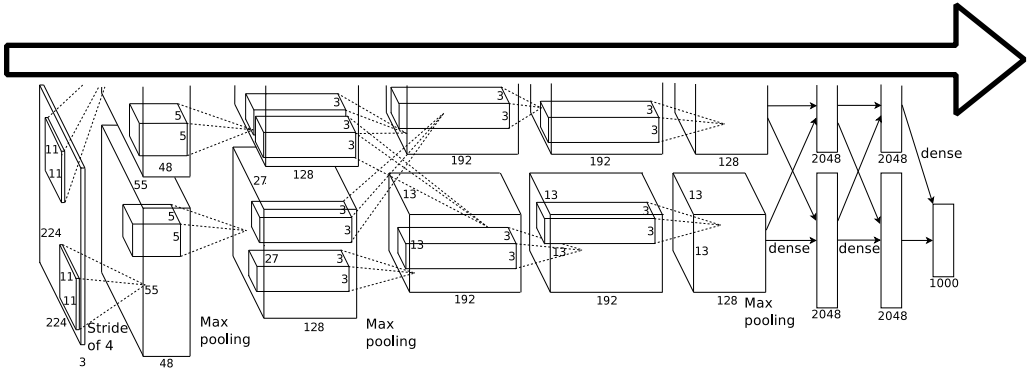
Motion planning
+ control

Motor
commands

Future paradigm?



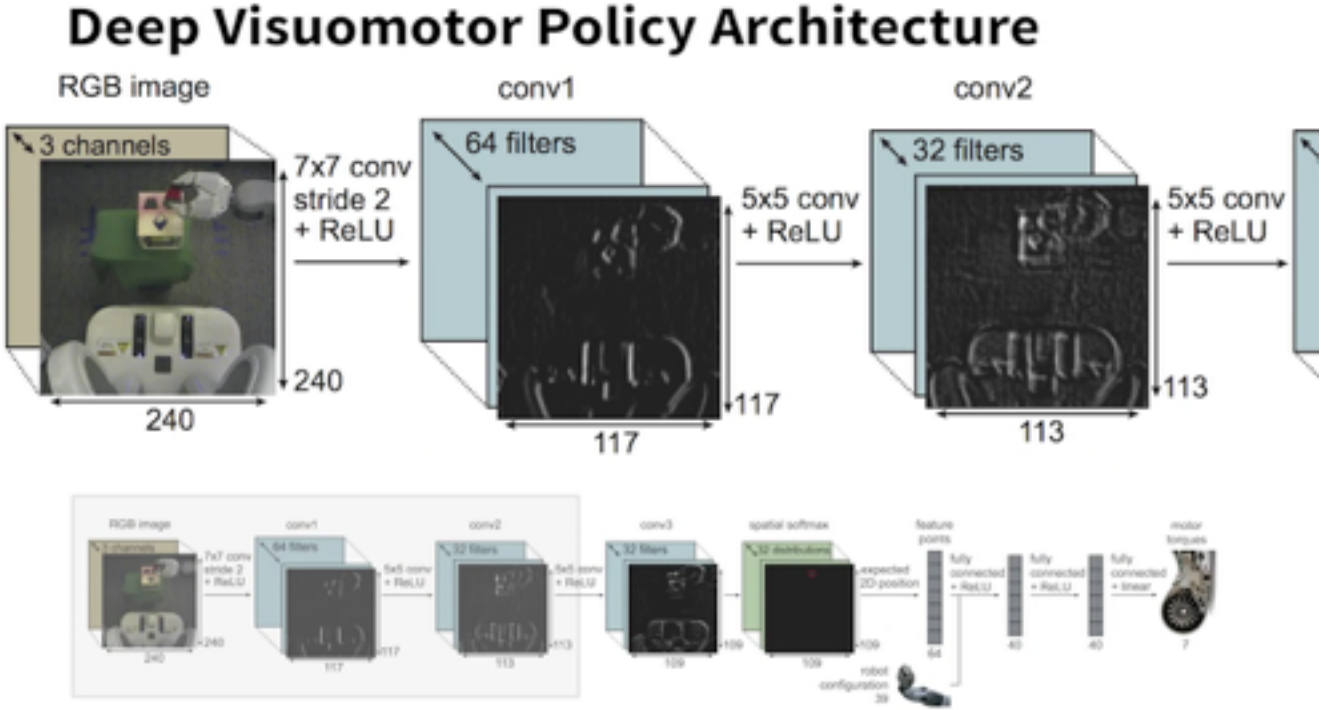
Sensor data
images / lidar



Deep neural net

Motor
commands

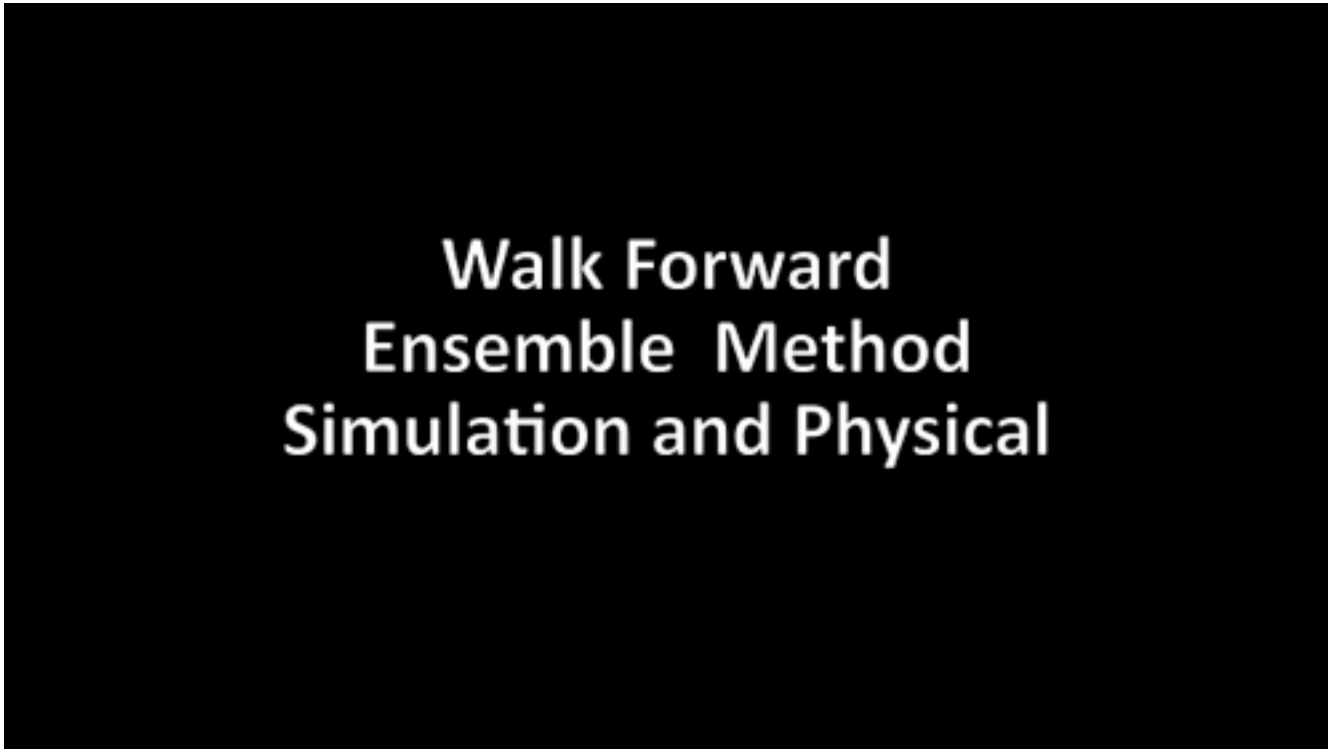
Frontiers in Robotic Manipulation



Frontiers in Robotic Locomotion

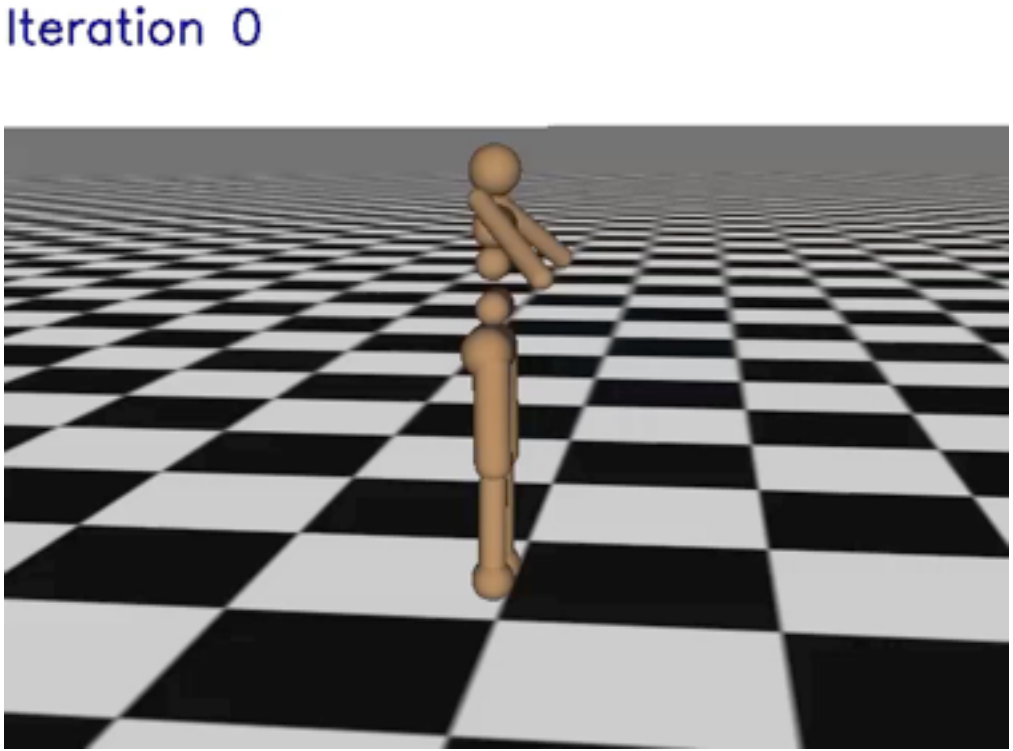
Mordatch et al., *Interactive Control of Diverse Complex Characters with Neural Networks*, Under review (2015)

Frontiers in Robotic Locomotion



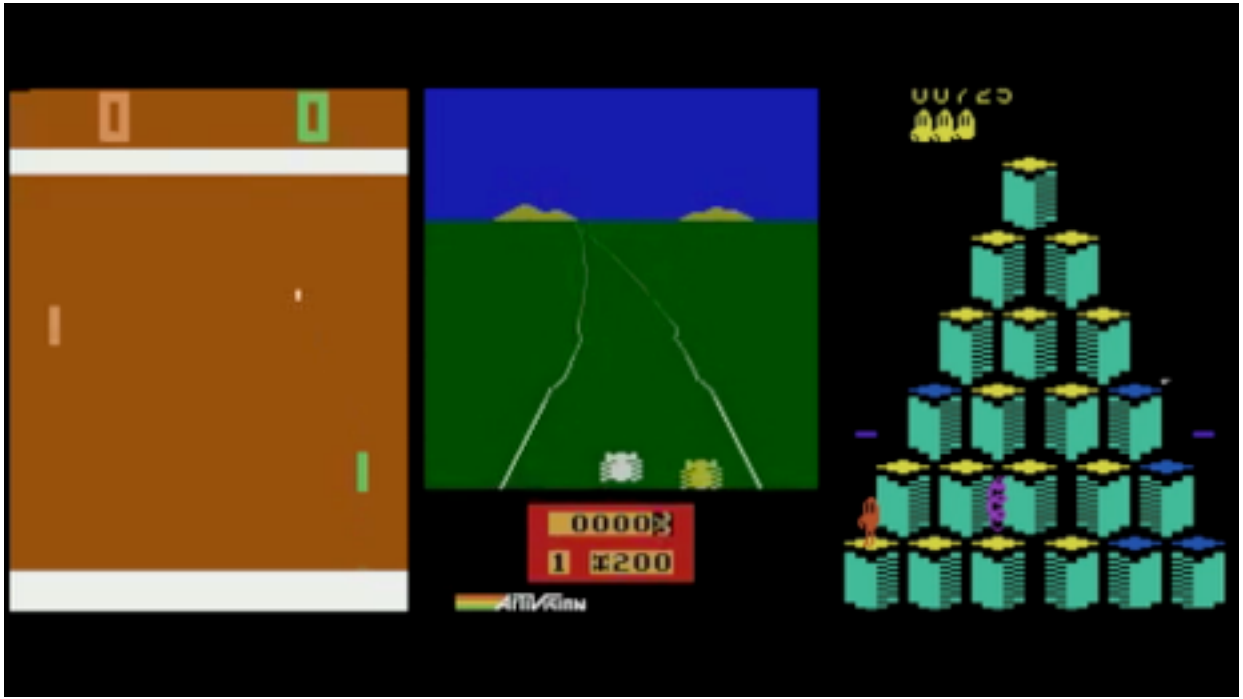
Mordatch, Igor, Kendall Lowrey, and Emanuel Todorov. *Ensemble-CIO: Full-Body Dynamic Motion Planning that Transfers to Physical Humanoids.*

Frontiers in Locomotion



Schulman, Moritz, Levine, Jordan, Abbeel (2015)
High-Dimensional Continuous Control Using Generalized Advantage Estimation

Atari Games



Schulman, Levine, Moritz, Jordan, Abbeel (2015) *Trust Region Policy Optimization*

Where Else Could Deep RL Be Applied?

Outline for Next Lectures

- Mon 8/31: MDPs
- Weds 9/2: neural nets and backprop
- Mon 9/9: policy gradients

Brushing up on RL: refs

- MDP review
  - Sutton and Barto, ch 3 and 4
- See Andrew Ng's thesis, ch 1-2 for a nice concise review of MDPs

Reinforcement Learning Textbooks

- Sutton & Barto, Reinforcement Learning: An Introduction
  - informal, prefers online algorithms
- Bertsekas, Dynamic Programming and Optimal Control
  - Vol 1. ch 6: survey of some of the most useful practical approaches for control, e.g. MPC, rollout algs
  - Vol 2 (*Approximate Dynamic Programming,* 3ed): linear and otherwise tractable methods for solving for value functions, policy iteration algs
- Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming
  - Exact methods for solving MDPs, including modified policy iteration
- Czepesvari, Algorithms for Reinforcement Learning
  - Theory on online algorithms
- Powell, Approximate Dynamic Programming
  - great on OR applications

Thanks

- Next class is Monday, August 31st
- We'll cover MDPs
- First homework will be released
  - uses python+numpy+ipython