Deep Reinforcement Learning

Danna Gurari
University of Colorado Boulder
Fall 2022

https://home.cs.colorado.edu/~DrG/Courses/NeuralNetworksAndDeepLearning/AboutCourse.html

Review

- Last week:
  - Motivation
  - Efficient learning: curriculum learning
  - Efficient learning: active learning
  - Efficient learning: other considerations
  - Faculty course questionnaire
- Assignments (Canvas):
  - Final project presentations due today
  - Final project report due next week
- Questions?

Today’s Topics

- Definition
- Motivation
- Background: “Markov decision processes” and “policies”
- Method: Policy Gradients for Pong

Today’s Topics

- Definition
- Motivation
- Background: “Markov decision processes” and “policies”
- Method: Policy Gradients for Pong

Definition

Agent takes actions in an environment to maximize the total reward

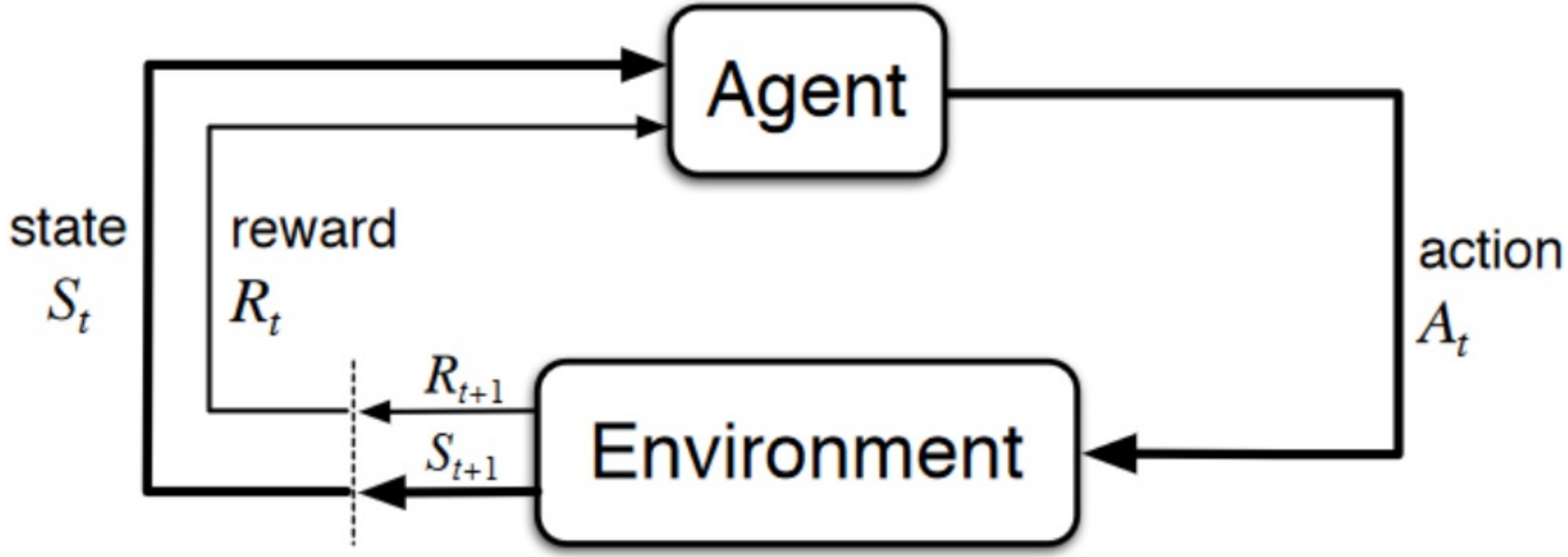
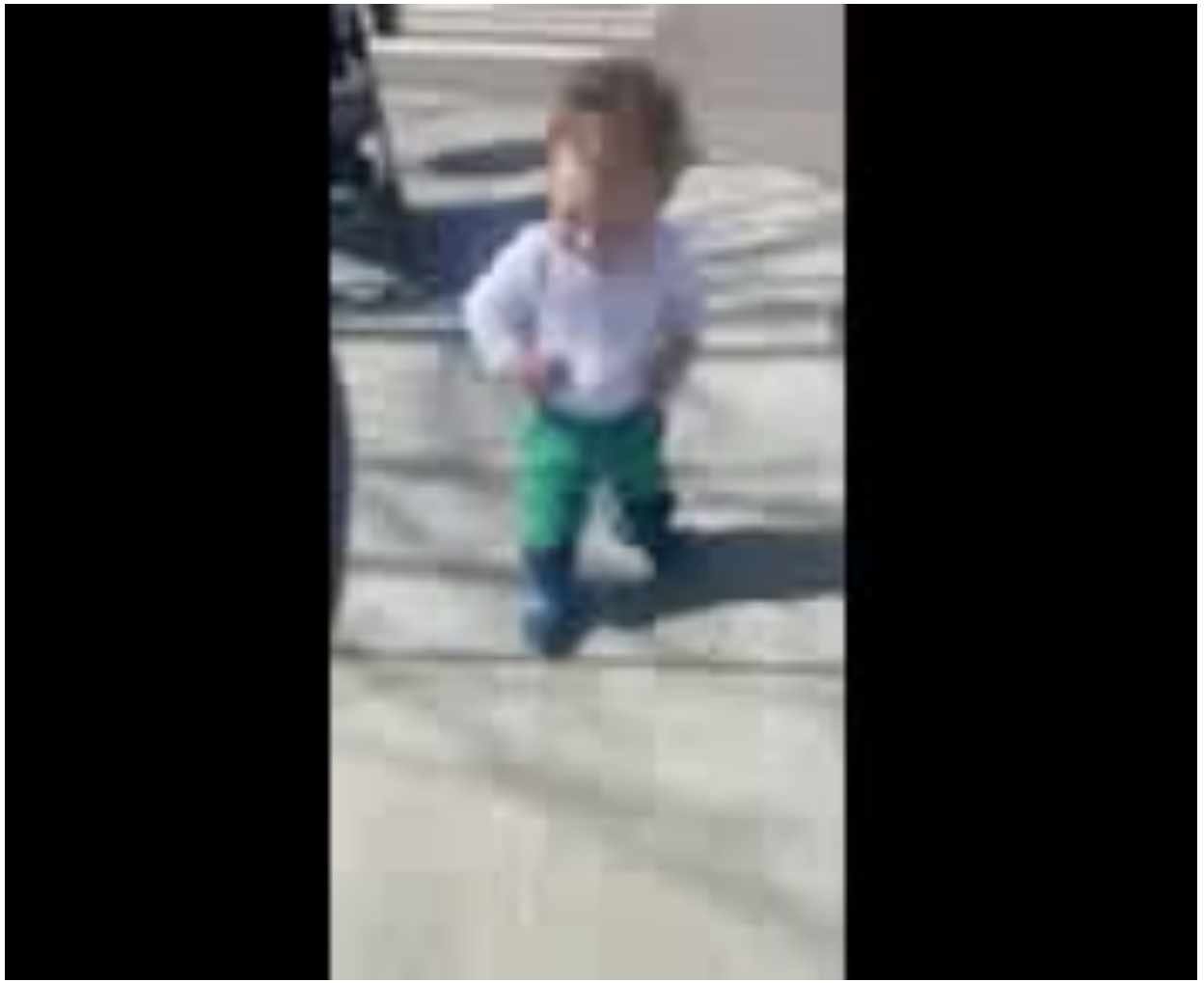
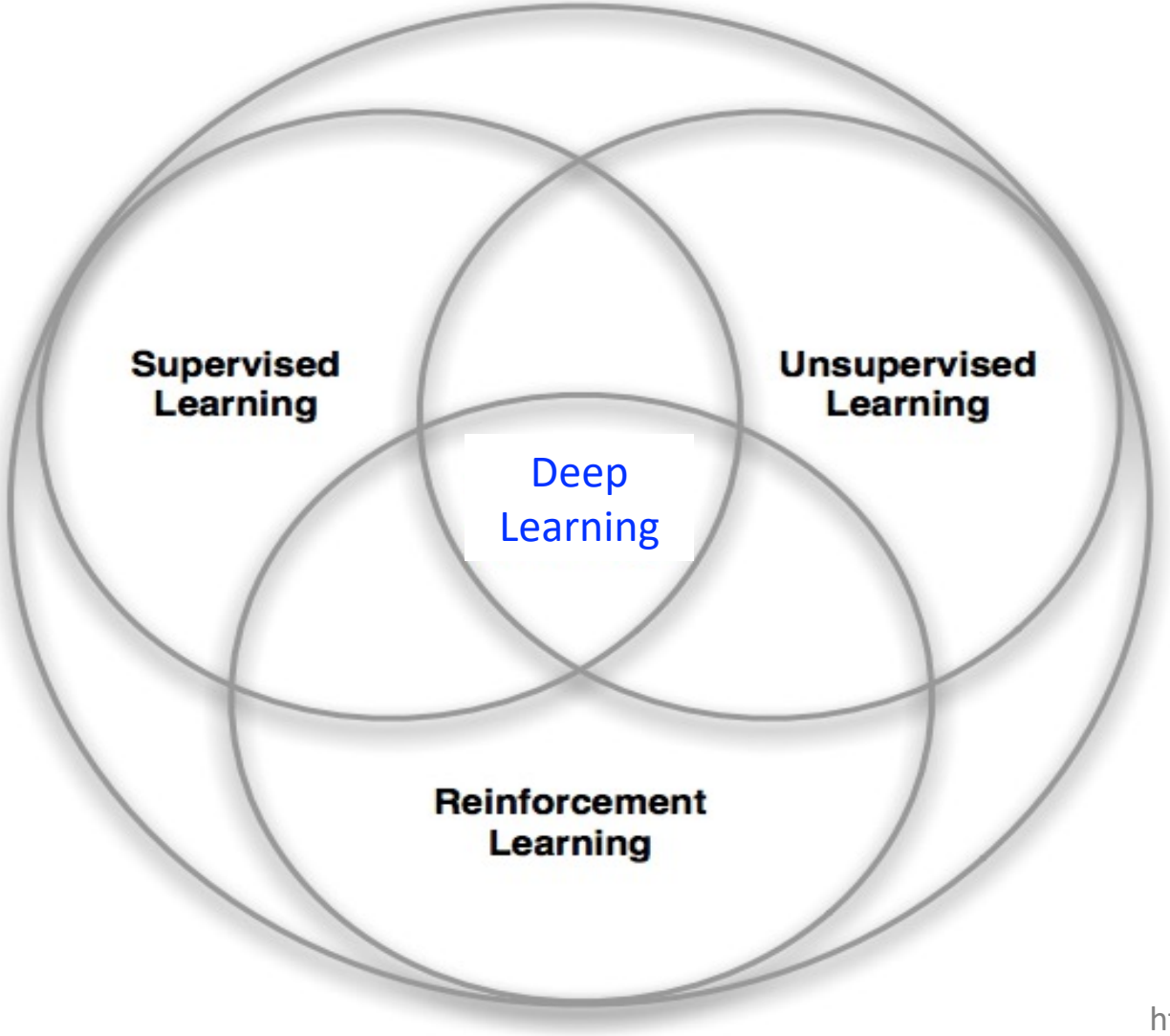


Figure Credit: https://towardsdatascience.com/applications-of-reinforcement-learning-in-real-world-1a94955bcd12

Intuition: Learning to Walk by Trial-and Error



Reinforcement Learning in Context



More information than unsupervised learning via rewards but less information than supervised learning's labels

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/intro_RL.pdf

Today’s Topics

- Definition
- Motivation
- Background: “Markov decision processes” and “policies”
- Method: Policy Gradients for Pong

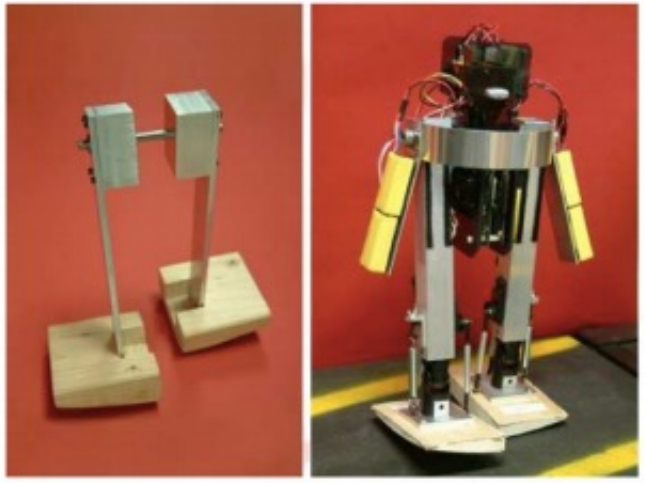
Robotics: Learning to Walk

Learning to Walk in 20 Minutes

Russ Tedrake
Brain & Cognitive Sciences
Center for Bits and Atoms
Massachusetts Inst. of Technology
Cambridge, MA 02139
russt@csail.mit.edu

Teresa Weirui Zhang
Mechanical Engineering
Department
University of California, Berkeley
Berkeley, CA 94270
resa@berkeley.edu

H. Sebastian Seung
Howard Hughes Medical Institute
Brain & Cognitive Sciences
Massachusetts Inst. of Technology
Cambridge, MA 02139
seung@mit.edu



Simulation: Learning to Walk

Demo: https://www.youtube.com/watch?v=gn4nRCC9TwQ

Robotics: Learning to Drive

Autonomous reinforcement learning on raw visual input data in a real world application

Sascha Lange, Martin Riedmiller
Department of Computer Science
Albert-Ludwigs-Universität Freiburg
D-79110, Freiburg, Germany
Email: [slange,riedmiller]@informatik.uni-freiburg.de

Arne Voigtländer
Shoogee GmbH & Co. KG
Krögerweg 16a
D-48155 Münster, Germany
Email: arne@shoogee.com

Fig. 1. The visual slot car racer task. The controller has to autonomously learn to steer the racing car by raw visual input of camera images.

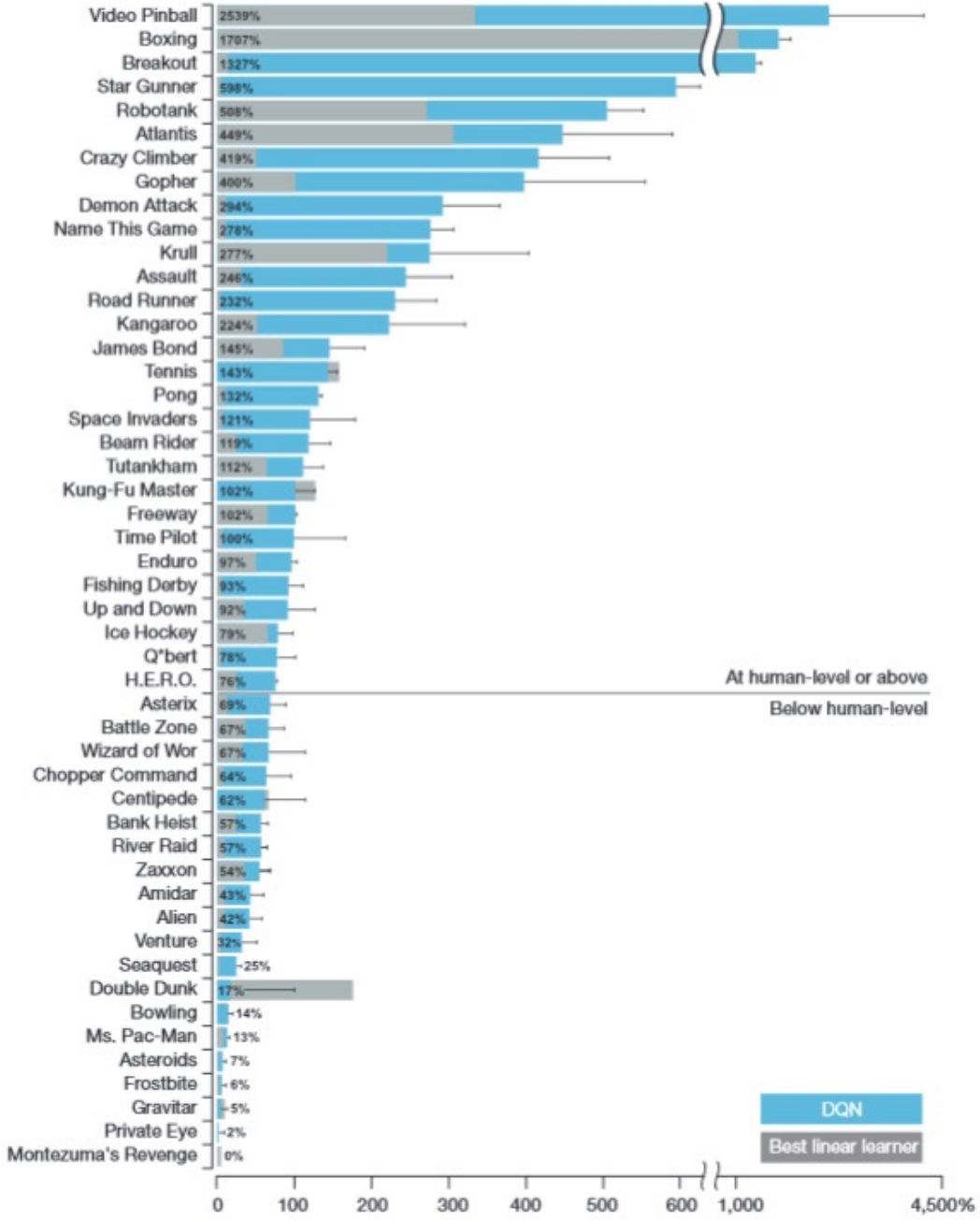
Robotics: Learning Dexterity

Demo: https://www.youtube.com/watch?v=jwSbzNHGflM

Robotics: Learning to Flip Pancakes

Demo: https://www.youtube.com/watch?v=W_gxLKSsSIE&list=PL5nBAYUyJTrM48dViibyi68urttMlUv7e

Games



https://www.tastehit.com/blog/google-deepmind-alphago-how-it-works/

https://web.stanford.edu/class/psych209/Readings/MnihEtAlHassibis15NatureControlDeepRL.pdf

What are other possible reinforcement learning applications?

Today’s Topics

- Definition
- Motivation
- Background: “Markov decision processes” and “policies”
- Method: Policy Gradients for Pong

Theoretical Foundation of RL: Markov Decision Processes (MDP)

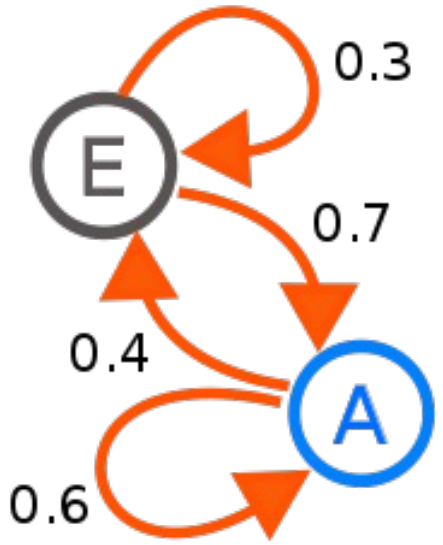
MDP consists of:

Markov process + Markov reward process + Actions

Markov Process (aka – Markov Chain)

- A set of states with transition probabilities defining system dynamics

e.g.,



  - How many states are in the above example?
- Markov property: only current state dictates future system dynamics

https://en.wikipedia.org/wiki/Markov_chain

Theoretical Foundation of RL: Markov Decision Processes (MDP)

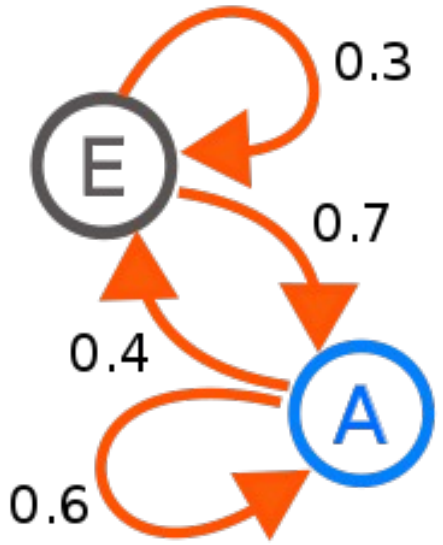
MDP consists of:

Markov process + Markov reward process + Actions

Markov Reward Process

- Additional scalar number associated with each transition (“+” or “–”)

e.g.,



- A discount factor between 0 and 1, gamma, indicates how far in the future rewards are considered to estimate a state’s expected reward

Figure source: https://en.wikipedia.org/wiki/Markov_chain

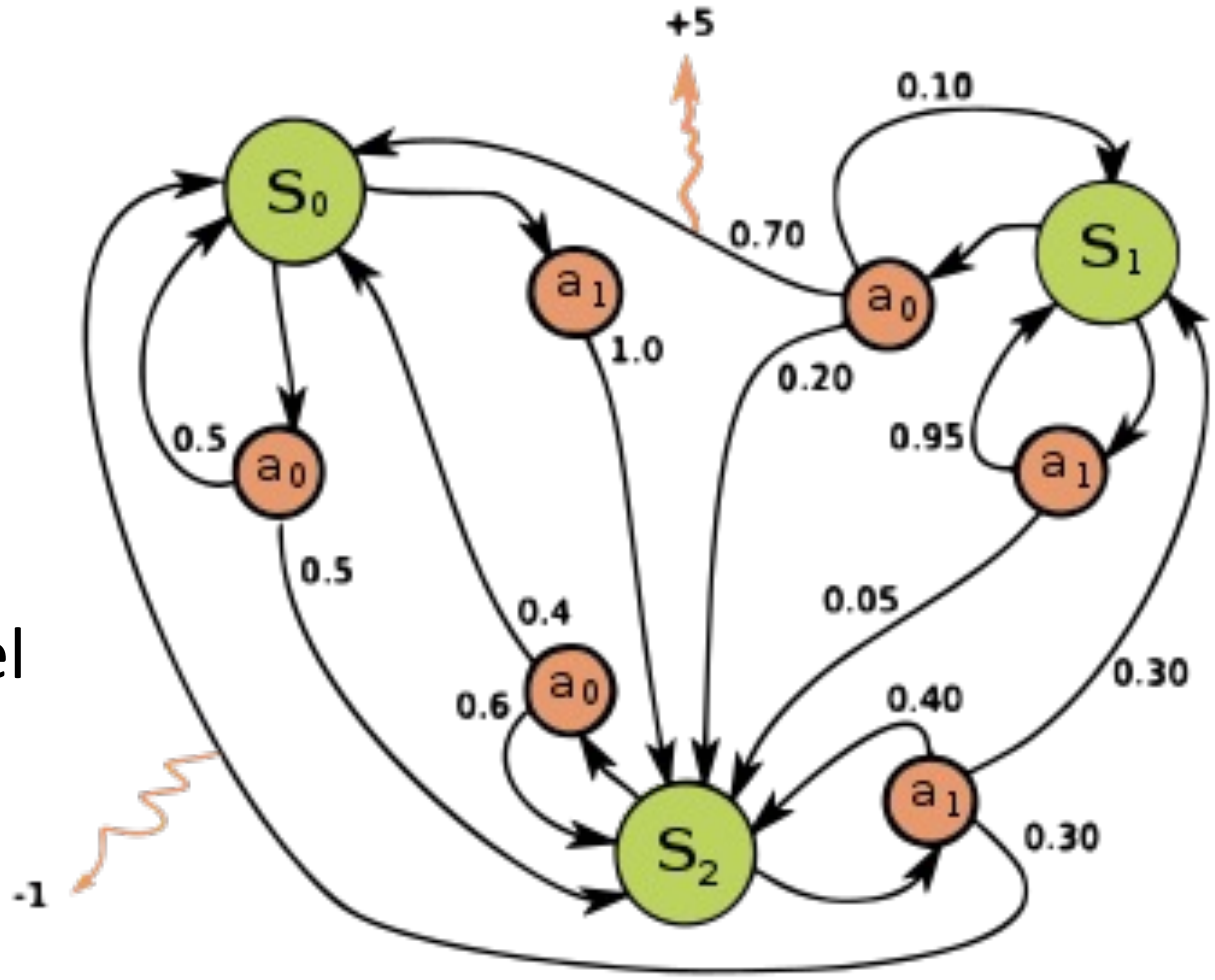
Theoretical Foundation of RL: Markov Decision Processes (MDP)

MDP consists of:

Markov process + Markov reward process + Actions

Third Ingredient of "Actions" Leads to a Markov Decision Process

- At each time step, the chosen action influences what will become the next state
- Probability allows for randomness (e.g., turn car wheel to go right on icy patch but car slips and continues straight)



https://en.wikipedia.org/wiki/Markov_decision_process

Policy

- Rules that dictate how an agent behaves
- Defined using a probability distribution over potential actions so there is randomness in the agent's behavior
- RL goal: find a good policy

Today's Topics

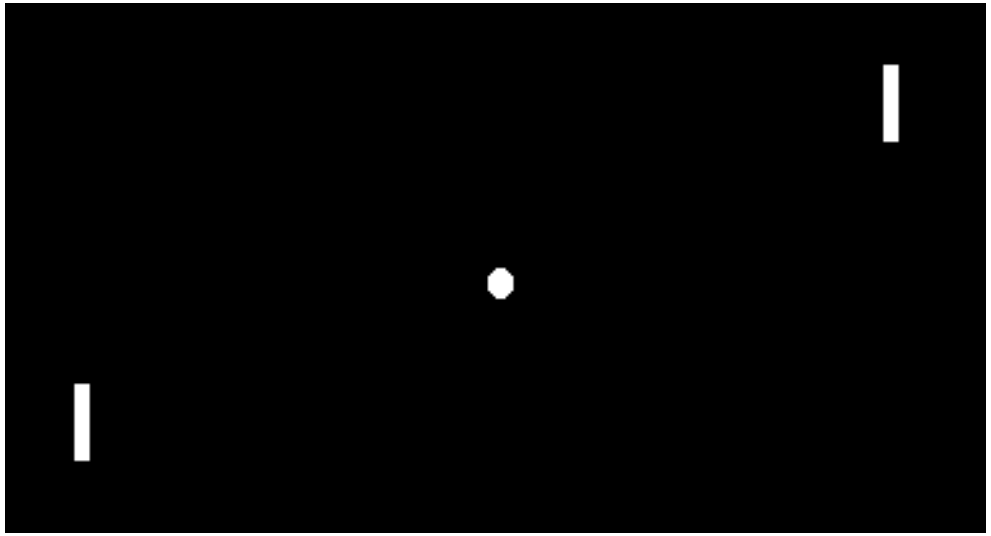
- Definition
- Motivation
- Background: "Markov decision processes" and "policies"
- Method: Policy Gradients for Pong

Basic Ingredients for RL Methods

1. Observations of environment
2. Possible actions
3. Rewards

Basic Ingredients for RL Methods; e.g., Pong

1. **Observations of environment**:

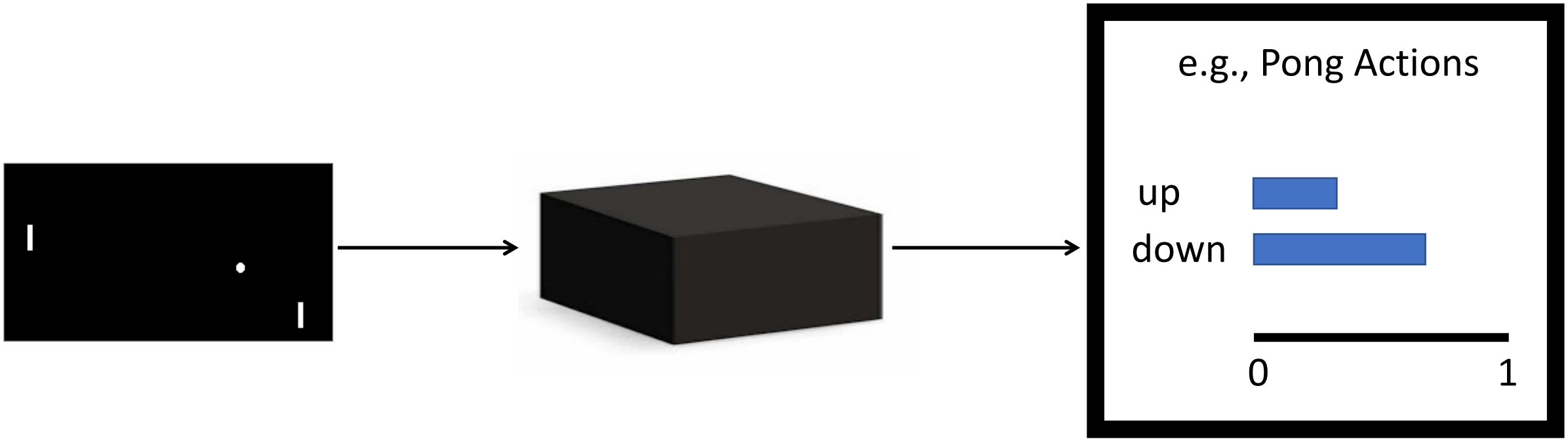


2. **Possible actions**: “up” and “down” paddle movements
3. **Rewards**: -1 if missed the ball; +1 reward if ball goes past opponent; 0 otherwise

Goal: Maximize rewards computing optimal “up” and “down” paddle movements

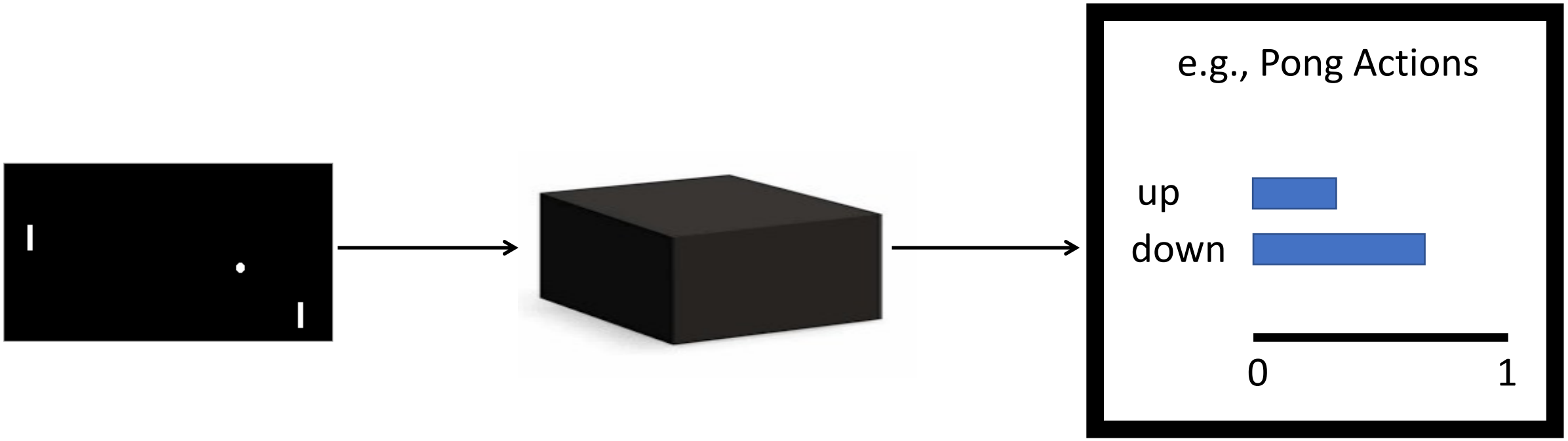
http://karpathy.github.io/2016/05/31/rl/

Policy Gradients: Approach



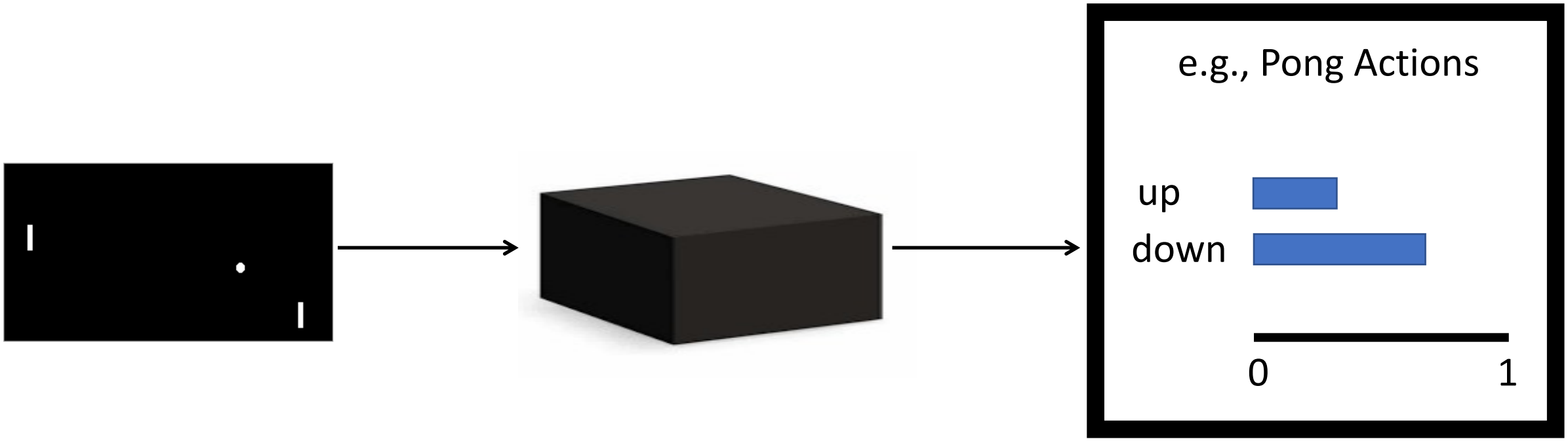
Policies (i.e., rules dictating how an agent behaves) are represented using a probability for each possible action

Policy Gradients: Approach



Neural network trained to increase probability of actions leading to a good total reward and decrease probability of actions leading to a bad total reward

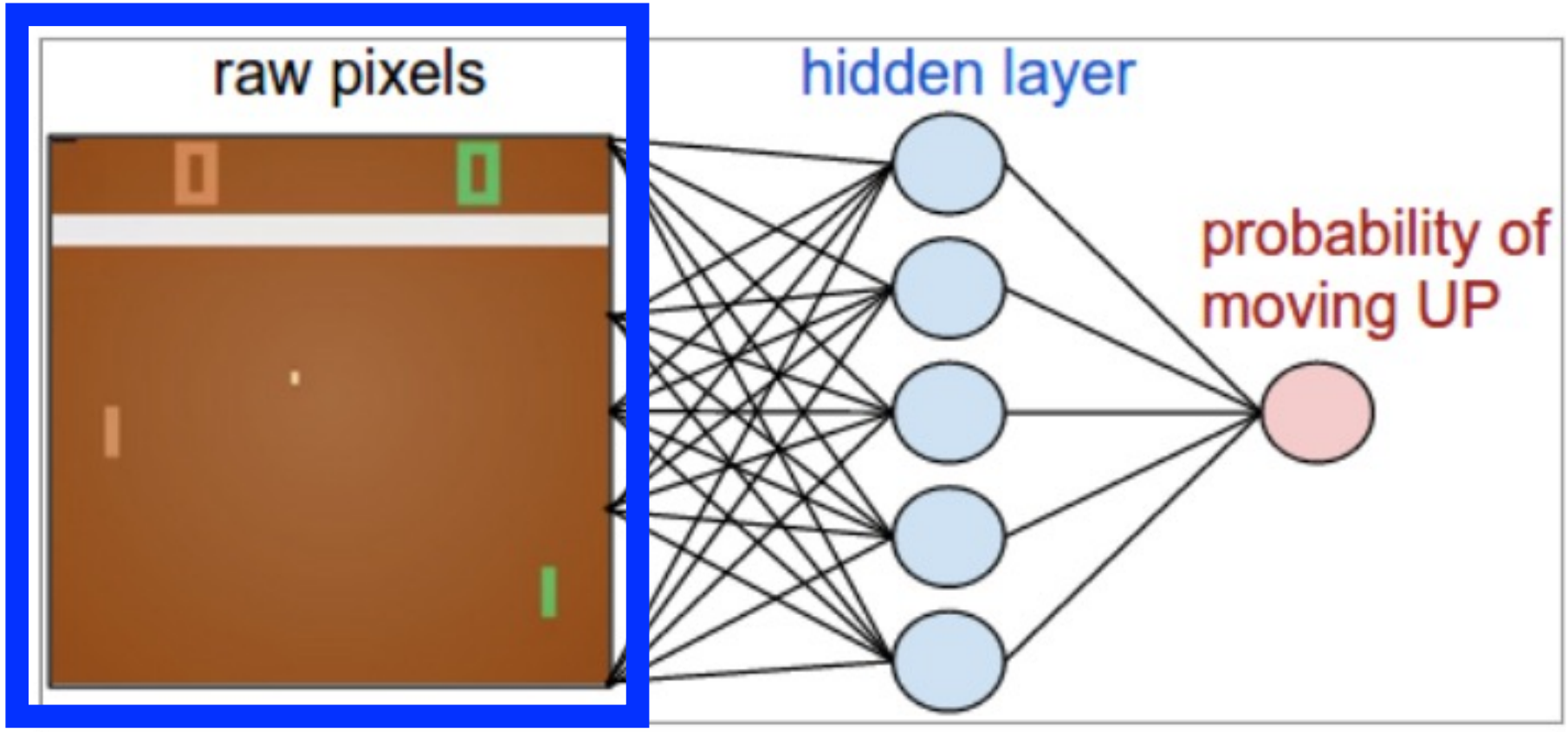
Policy Gradients: Approach



How does this approach support “exploration”?

e.g., Learning Pong (2-layer NN with 200 hidden units)

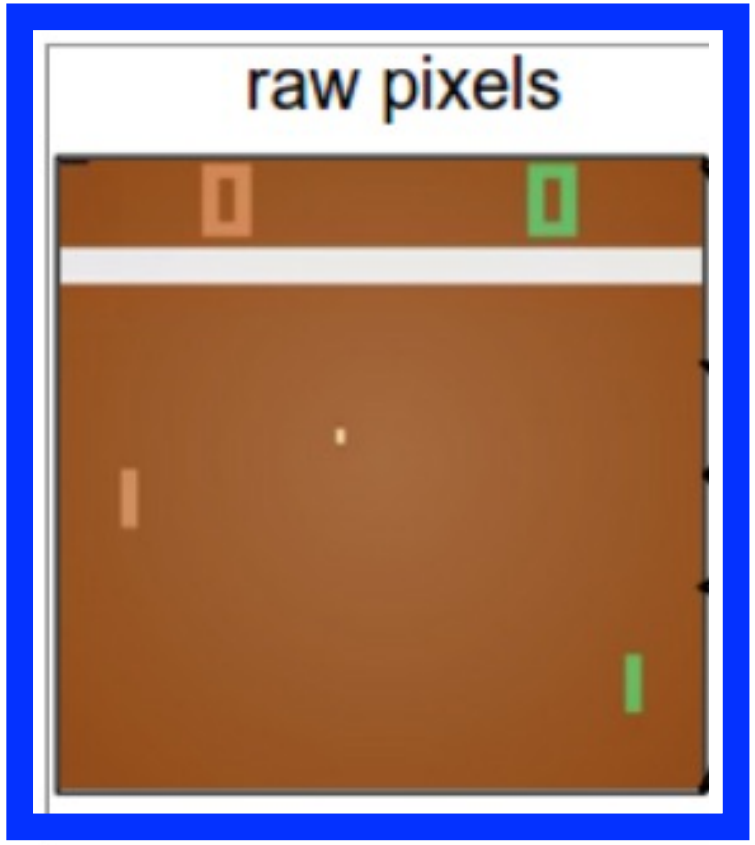
Given game state (as image), decide if to move paddle up (vs down)



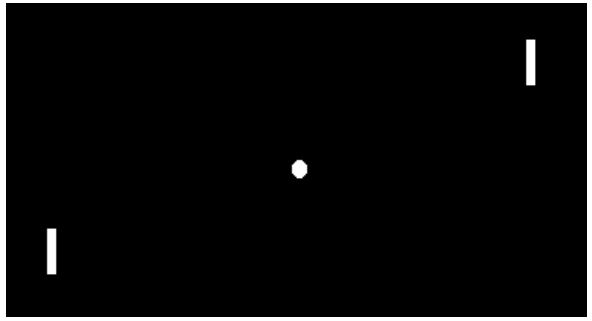
http://karpathy.github.io/2016/05/31/rl/

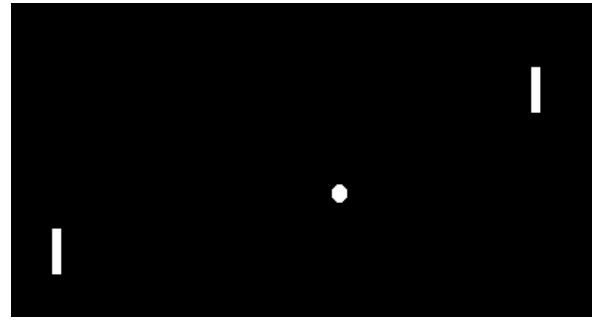
e.g., Learning Pong (2-layer NN with 200 hidden units)

How to capture motion in game state (i.e., image)?



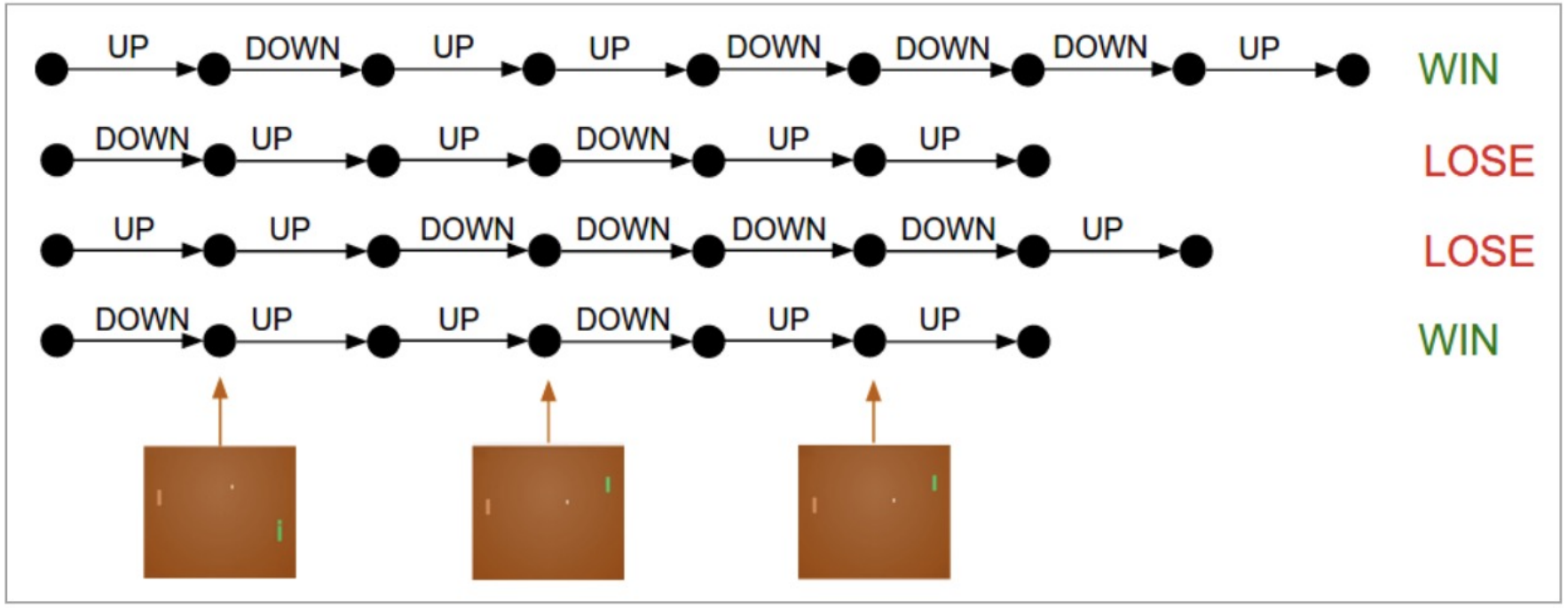
Use difference image
(i.e., subtract frame current from last frame)





http://karpathy.github.io/2016/05/31/rl/

e.g., Learning Pong: Training Protocol

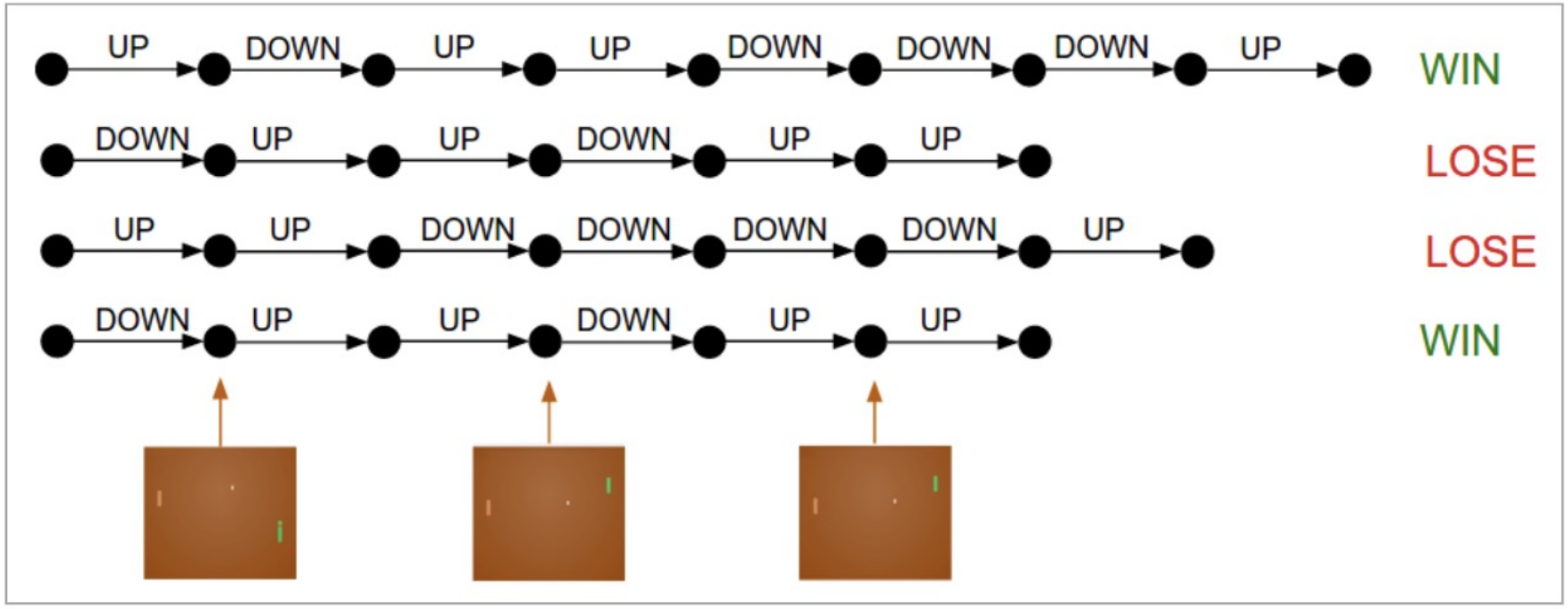


Assume 100 games played with 200 images/game

How many (action) decisions were made?

http://karpathy.github.io/2016/05/31/rl/

e.g., Learning Pong: Training Protocol

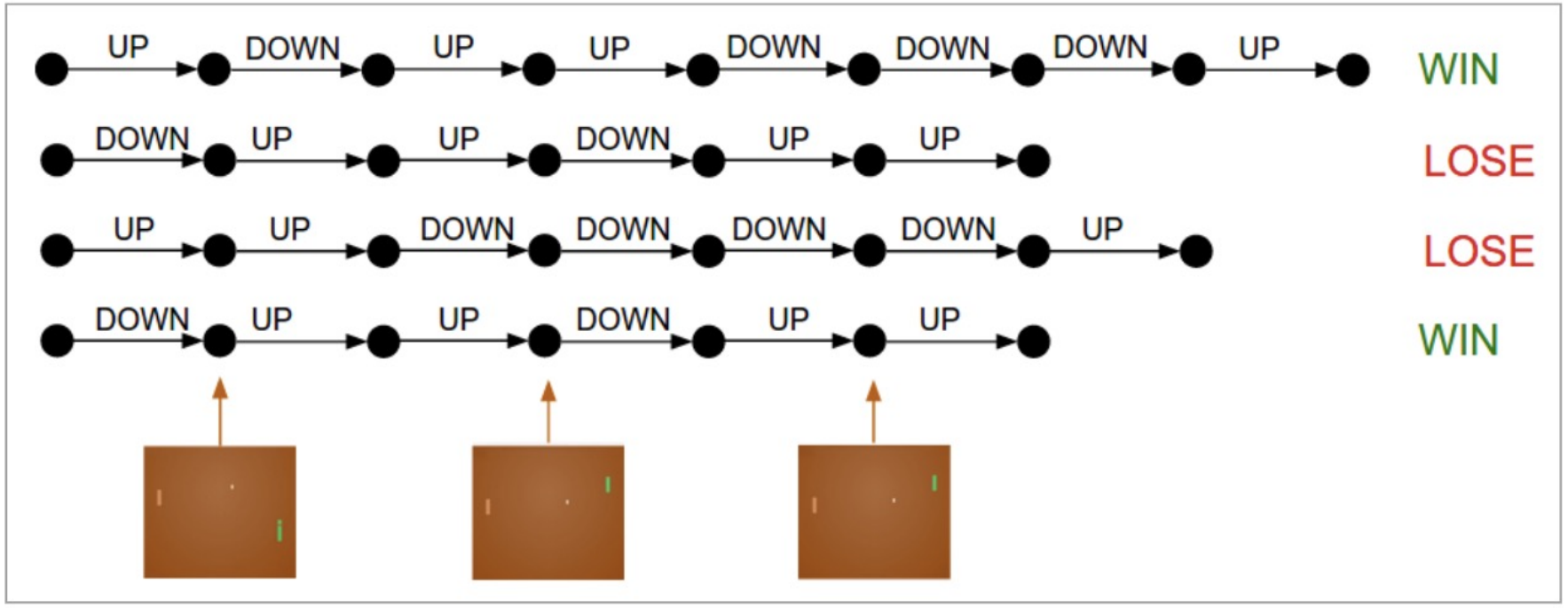


Assume 100 games played with 200 images/game; 12 games won & 88 lost

- How many winning decisions were made?
  - 2,400 (i.e., 12 x 200)
- How many losing decisions were made?
  - 17,600 (i.e., 88 x200)

http://karpathy.github.io/2016/05/31/rl/

e.g., Learning Pong: Training Protocol



After each set of 100 games, gradient updated to encourage actions that eventually lead to good outcomes (i.e., 2,400 winning up/down actions) and discourage actions that eventually lead to bad outcomes discouraged (17,600 losing up/down actions)

http://karpathy.github.io/2016/05/31/rl/

e.g., Pong Model: RL Model vs Pong’s AI Model

Demo: https://www.youtube.com/watch?time_continue=16&v=YOW8m2YGtRg

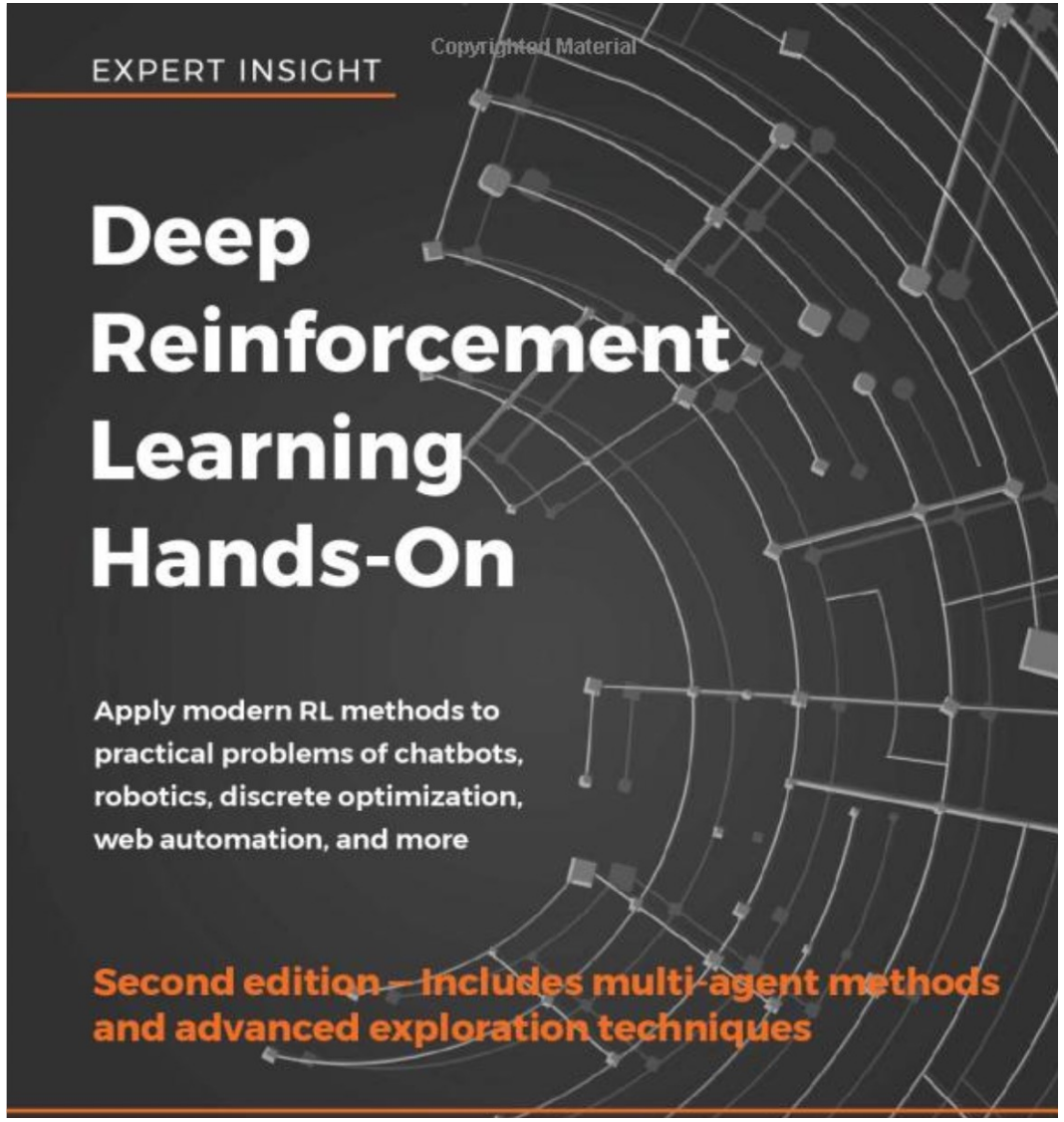
Why Reinforcement Learning is Difficult

- Agent must infer what it did wrong/right and so how performance can be maintained/improved based on (delayed) rewards; e.g., chess

- Agent needs to strike the appropriate balance between exploration and exploitation; e.g., order one's favorite food vs something new

Want to Learn More?

- CU Boulder course (currently taught by Alessandro Roncone)
- SW dev environment: OpenAI Gym
- Hands on training:



Today’s Topics

- Definition
- Motivation
- Background: “Markov decision processes” and “policies”
- Method: Policy Gradients for Pong


The End