Northeastern University
College of Engineering

EECE 5698 – ST: Reinforcement Learning

Lecture 1: Introduction

Course Instructor: Contact Information

Instructor:

- **Prof. Mahdi Imani, Assistant Professor, Electrical and Computer Engineering Department**
- **Email: m.imani@northeastern.edu**

Course Information:

- **Lecture: Tuesdays and Fridays 9:50 am – 11:30 am**
- **Location: Hayden Hall #221**
- **Lecture mode: The class is in-person and attendance is mandatory.**

Teaching Assistant (TA):

- **Begum Taskazan (taskazan.b@northeastern.edu)**
- **Office Hours:**
  - **In-person: Wednesdays, 12 - 1 pm (Snell Library Colab T (1st Floor))**
  - **Virtual: Fridays, 12 - 1 pm**

Course Objectives

- Define the key features that distinguish reinforcement learning (RL) from artificial intelligence and non-interactive machine learning;
- Understand RL algorithms and build RL models for sequential decision making;
- Understand how to formulate a task as an RL problem, and how to begin implementing a solution;
- Understand how RL fits under the broader umbrella of machine learning, and how it complements deep learning, supervised and unsupervised learning.

Course Outline and Schedule

Let's check the Course Syllabus!

❖ Textbook:

The main textbook for the course will be:

[1] Richard S. Sutton, and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2nd Edition, 2018.

Occasionally we may also use a few additional book chapters and research papers (details on the syllabus). For example:

[2] Csaba. Szepesvári "Algorithms for reinforcement learning." Synthesis lectures on artificial intelligence and machine learning 4.1 (2010): 1-103.

[3] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. "An introduction to deep reinforcement learning." Foundations and Trends® in Machine Learning 11, no. 3-4 (2018): 219-354.

❖ Topics, Exams, Homework and Projects

Date	Topics	HWs/Projects	Ref
L1-Jan 13	Introduction to Reinforcement Learning		
L2-Jan 17	Multi-Armed Bandit	HW1	Ref [1]-Chap 2
L3-Jan 20	Multi-Armed Bandit	Project 1	Ref [1]-Chap 2
L4-Jan 24	Reinforcement Learning Preliminaries		Ref [1]-Chap 3
L5-Jan 27	Reinforcement Learning Preliminaries	HW1 Due	Ref [1]-Chap 3
L6-Jan 31	Bellman Equation and Optimality		Ref [1]-Chap 3
L7-Feb 3	Bellman Equation and Optimality		Ref [1]-Chap 3
L8-Feb 7	Dynamic Programming – Policy Iteration	HW2 P1 Due	Ref [1]-Chap 4
L9-Feb 10	Dynamic Programming – Value Iteration	Project 2	Ref [1]-Chap 4
L10-Feb 14	Policy Iteration – Matrix Form		Ref [1]-Chap 5
L11- Feb 17	Value Iteration – Matrix Form	HW2 Due	Ref [1]-Chap 5
L12- Feb 21	**Exam 1**		
L13- Feb 24	Approximate Dynamic Programming	HW3	Ref [1]-Chap 5
L14- Feb 28	Monte Carlo Learning		Ref [1]-Chap 6
L15-March 3	Temporal Difference Learning	P2 Due	Ref [1]-Chap 6
Spring Break			
L16-March 14	On-Policy Learning	Project 3	Ref [1]-Chap 7
L17- March 17	Off-Policy Learning	HW4 HW3 Due	Ref [1]-Chap 7
L18- March 21	Off-Policy vs. On Policy		Literature
L19- March 24	TD Variations		Literature
L20- March 28	Function Approximations Methods in RL	HW5	Literature
L21- March 31	Least Square Policy Iteration and Neural Fitted Q-Iteration	HW4 Due	Ref [3]-Chap 4
L22-April 4	**Exam 2**		
L23-April 7	Deep Q Networks		Ref [3]-Chap 4
L24-April 11	DQN Variations: Double, Dueling, Prioritized	HW5 Due	Ref [3]-Chap 5
L25-April 14	Policy Gradient Methods	P3 Due	Ref [3]-Chap 5
L26-April 18	Review Session		

Course Evaluations and Grading Policy

- HWs 20%
- Project 1 10%
- Project 2 20%
- Project 3 20%

50%

- Exam 1 15%
- Exam 2 15%

30%

- Which programming language am I allowed to use for the projects?

  Python: Highly Recommended.

  Others: Should be checked with TA first.

- Both exams are closed book.

Important Notes

- Lecture slides will be posted on Canvas after the class.
- Only electronic submission of HWs and projects is accepted.
- Project-related and technical questions should be directed to TA first. However, if you have doubts about notes in lecture notes or do not get answers from TA, feel free to contact me.
- Suggestions are ALWAYS welcome.

How to maximize your learning?

1. Participate in all lectures and ask questions.
2. Use the TA office hours to ask your questions about HWs and projects, debug your code, etc.
3. The HWs are designed to prepare you for exams and projects.
4. Find friends in the class and talk to them about the project, your way of implementation, etc. Please help each other!
5. Review your probability course or ask for more explanations if needed.
6. Look at the textbook materials if anything is unclear (or ask me!).
7. Keep in your mind: the course objectives are learning, learning, learning.

Northeastern University
College of Engineering

EECE 5698 – Special Topics: Reinforcement Learning

Introduction to Reinforcement Learning

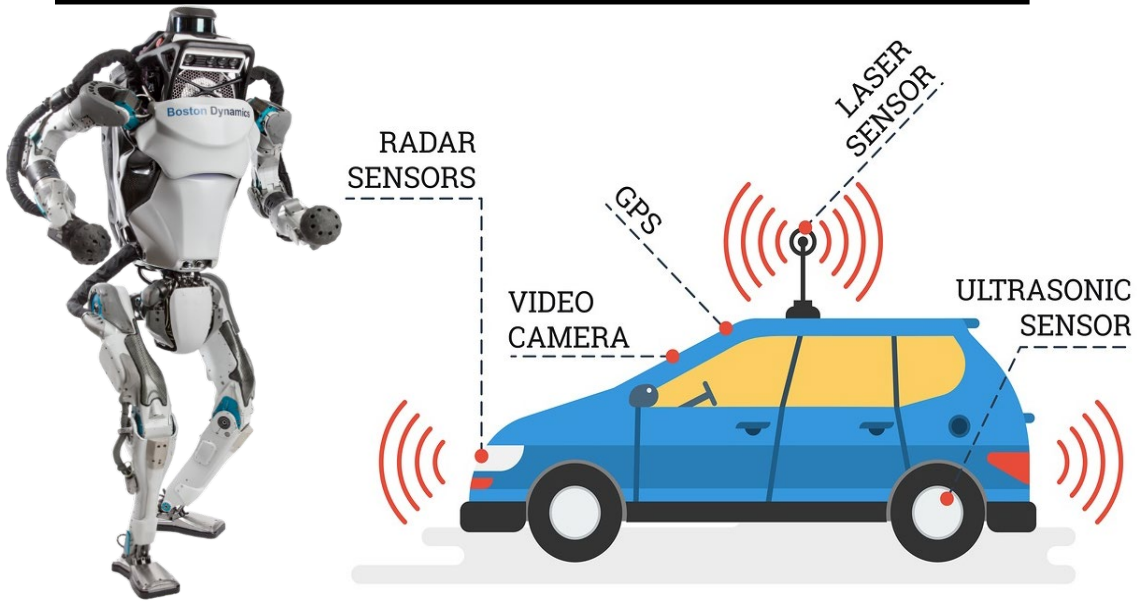
Instructor: Prof. Mahdi Imani

Spring 2023

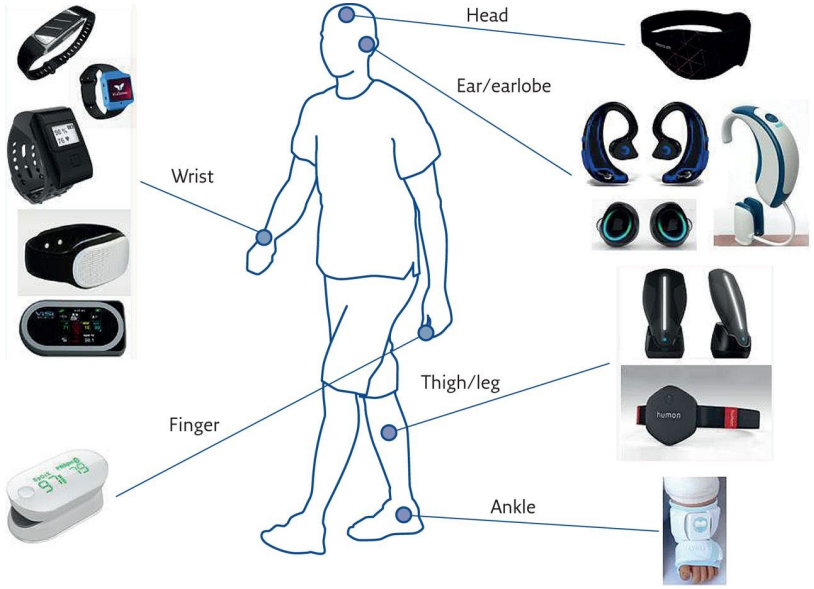
Introduction

Northeastern University
College of Engineering

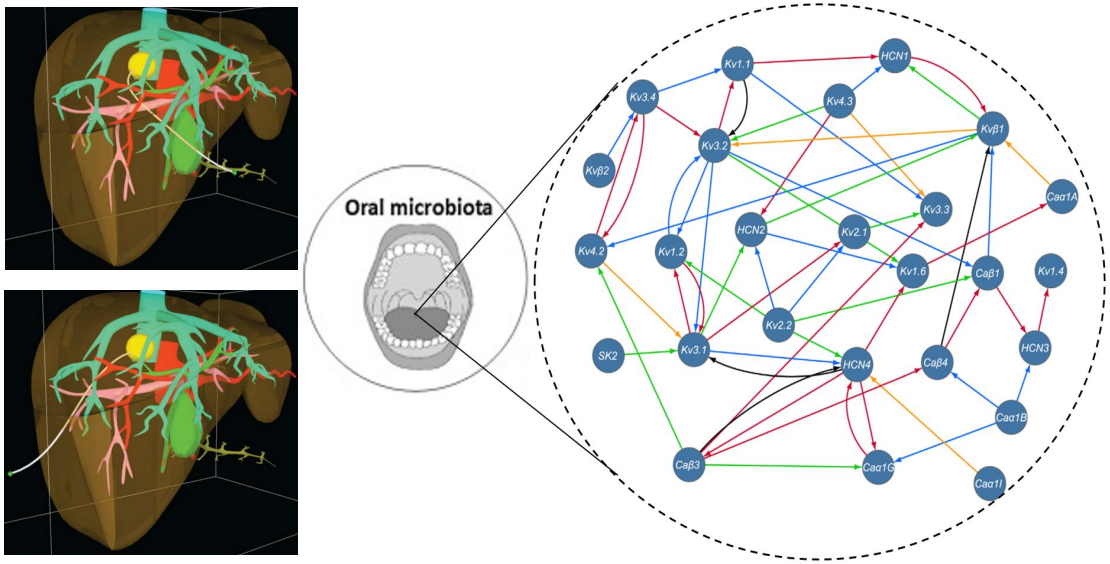
Intelligent Systems



Smart/Wearable Devices



Biomedical Informatics - Healthcare



Cyber-Physical Systems

Supervised Learning

Northeastern University
College of Engineering

Training Data: $(x_k, y_k), k = 1, \ldots, T$

Input: x

Supervised Learning
Model: $f_\theta(.)$

Model Output: $f_\theta(x)$

$$\theta^* = \text{argmin}_\theta \sum_{k=1}^{T} h(f_\theta(x_k) - y_k)$$

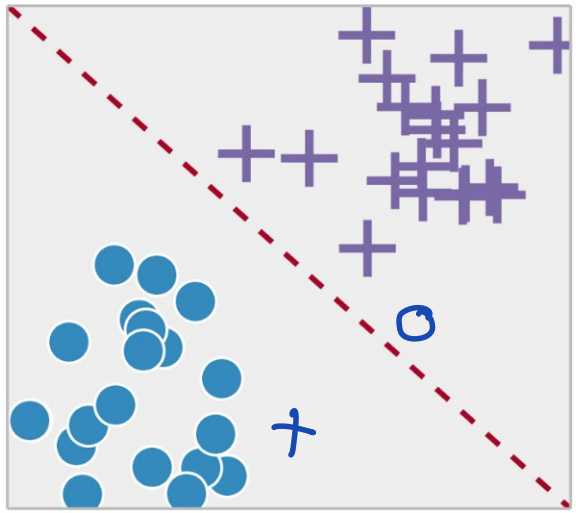
Error

$\theta_1 x + \theta_2 = f_\theta(x)$

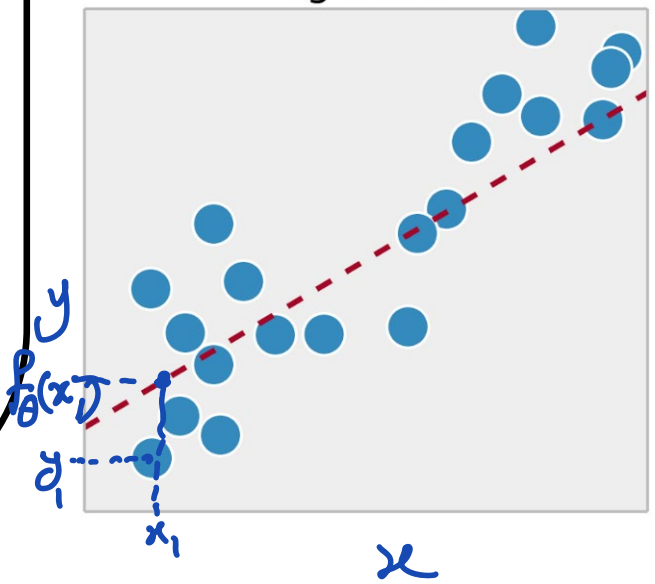
Model Output

Actual Output

Classification



Regression



Unsupervised Learning

Training Data: $x_k, k = 1, \ldots, T$

Input: x

Unsupervised Learning
Model: $f_{\boldsymbol{\theta}}(.)$

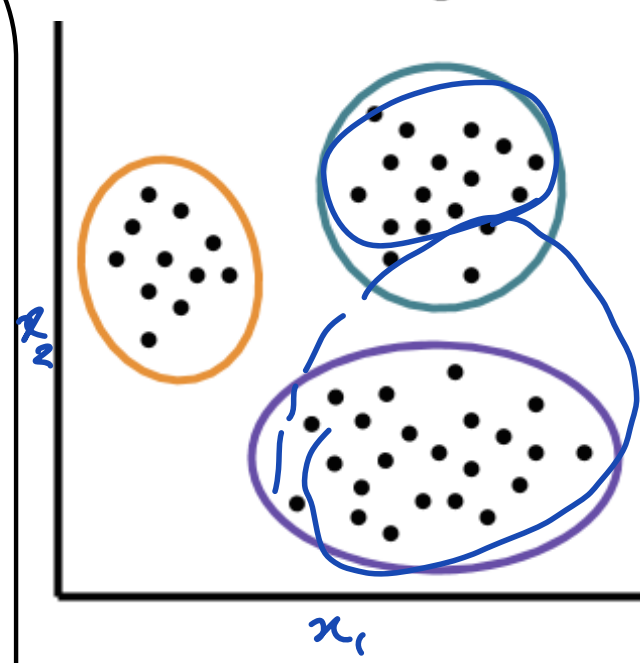
Model Output: $f_{\boldsymbol{\theta}}(x)$

Loss Function

$$\theta^* = argmin_{\boldsymbol{\theta}} \sum_{k=1}^{T} Loss(f_{\boldsymbol{\theta}}(x_k))$$

Model Output

Clustering



Semi-Supervised Learning

Northeastern University
College of Engineering

Training Data: $(s_k, a_k, r_k), k = 1, \dots, T$

Input (State) s

Semi-Supervised Learning
Policy Model: $\pi_{\boldsymbol{\theta}}(.)$

Action: $\pi_{\boldsymbol{\theta}}(s)$

Expected Accumulated Reward

Action

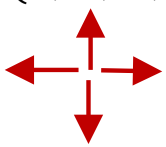
$$\theta^* = \operatorname{argmax}_{\boldsymbol{\theta}} E\left[\sum_{t=1}^{L} r_t \mid \pi_\theta\right] = \operatorname{argmax}_{\boldsymbol{\theta}} E\left[\sum_{t=1}^{L} R(s_{t-1}, a_t = \pi_{\boldsymbol{\theta}}(x_{t-1}))\right]$$

Mov. Empty

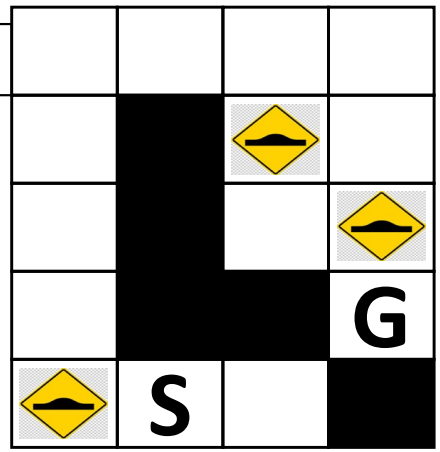
Mov. Bump

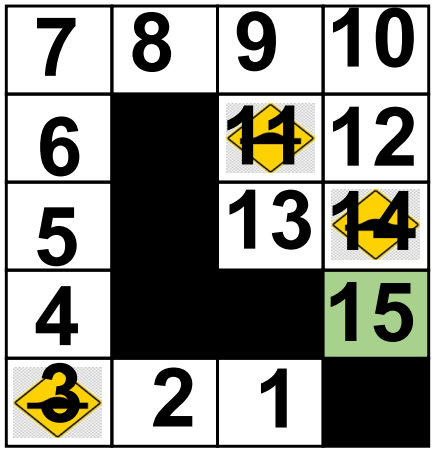
Mov. G

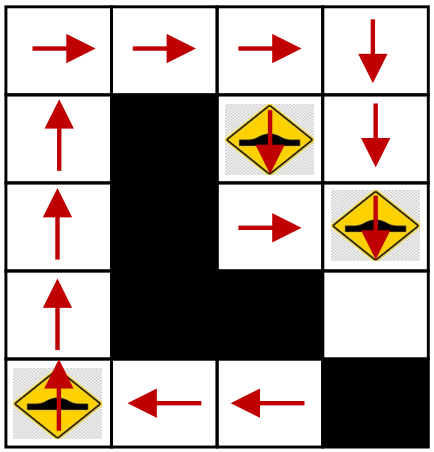
$r \in (-1, -10, 100)$
$s \in (1, 2, \dots, 14, 15)$
$a \in (L, R, U, D)$



$s = 2, a = R \rightarrow r = -1$
$s = 2, a = L \rightarrow r = -10$



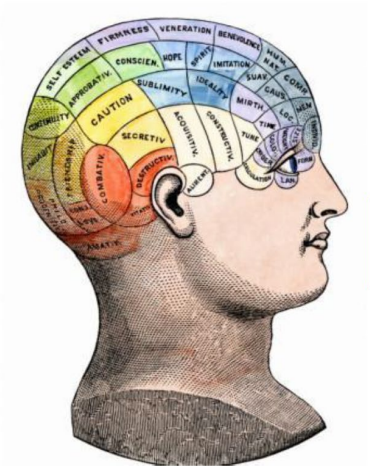


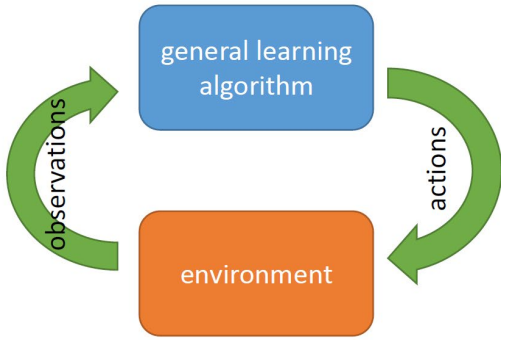


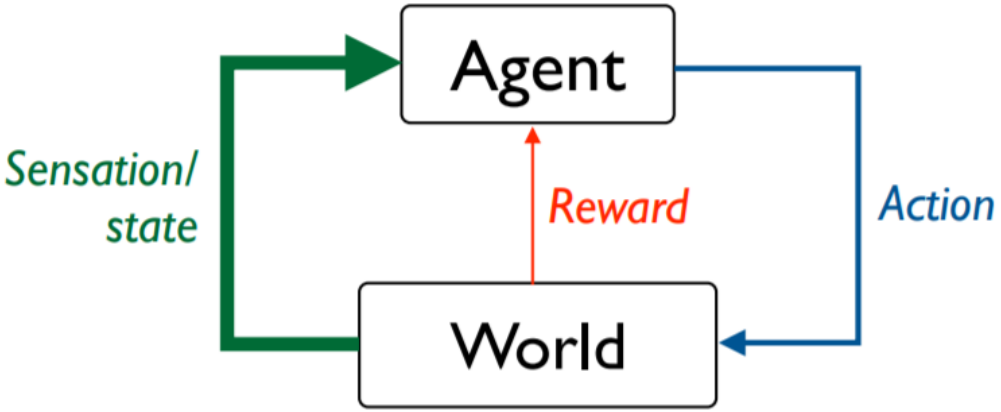
$$\pi^* = \begin{cases} 1 \rightarrow L \\ 2 \rightarrow L \\ 3 \rightarrow U \\ \vdots \\ 13 \rightarrow R \\ 14 \rightarrow D \end{cases}$$

Sequential Learning

Northeastern University
College of Engineering



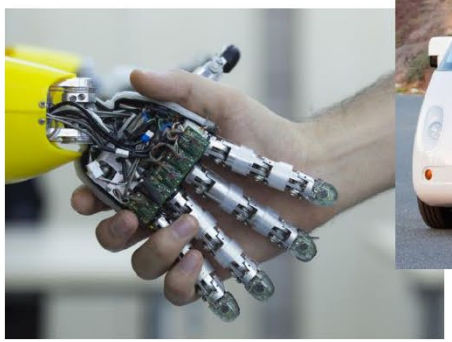




Limited supervision: you know **what** you want, but not **how** to get it (Actions have consequences)

Common Applications

autonomous driving



robotics

language & dialogue
(structured prediction)

business operations

Smart Cities

Sequential Learning: Why NOW?

Northeastern University
College of Engineering

Advances in Technology

Advances in Mathematical Tools

Advances in Computational Capabilities

Hyperdimensional Computing

Mimics the ***functionality*** of the brain

Robust against noise/failure

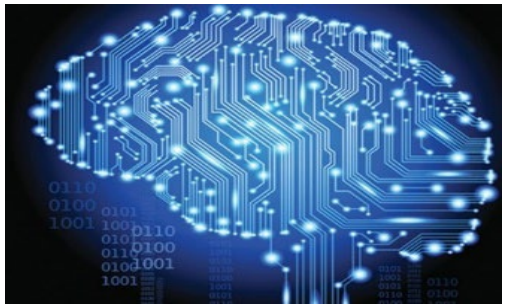


Image Source: https://www.evolvingsol.com/

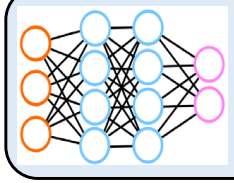
High parallelism

Brain-Like Machine

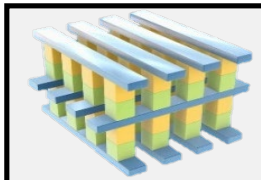
Deep Learning

Mimics ***physical properties***

https://www.information-age.com



Efficient Computing



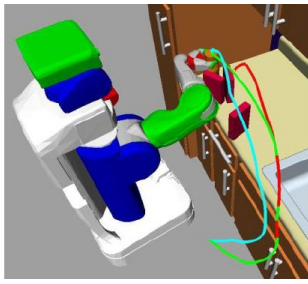
Highly Parallel Platform

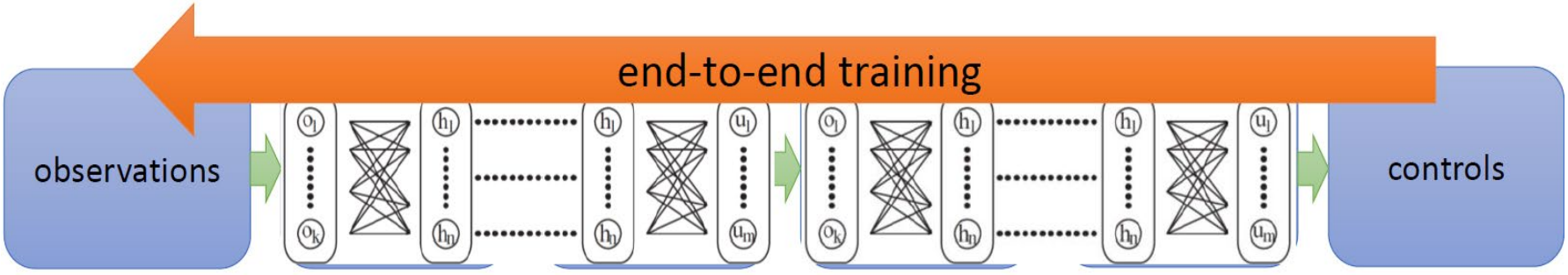
Processing In-Memory (PIM)

Deep Reinforcement Learning

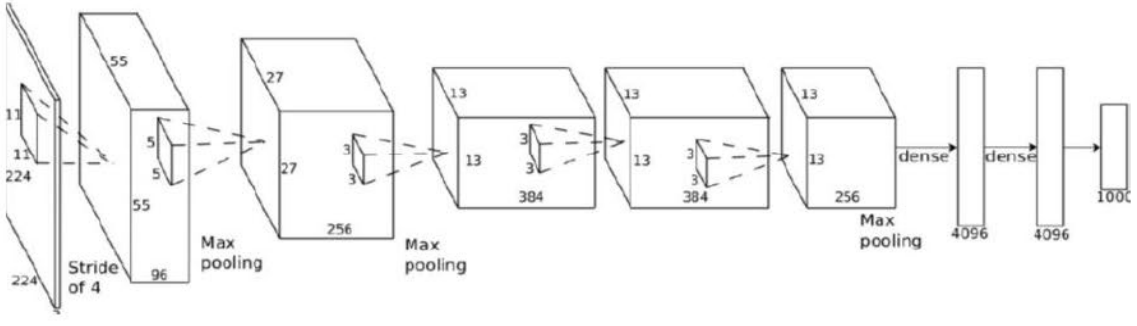
Northeastern University
College of Engineering

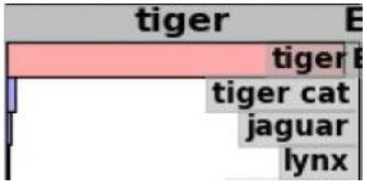
observations → state estimation (e.g. vision) → modeling & prediction → planning → low-level control → controls



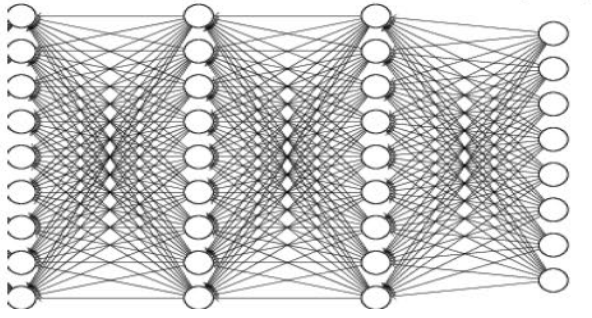


Deep models are what allow RL to solve complex problem end-to-end!





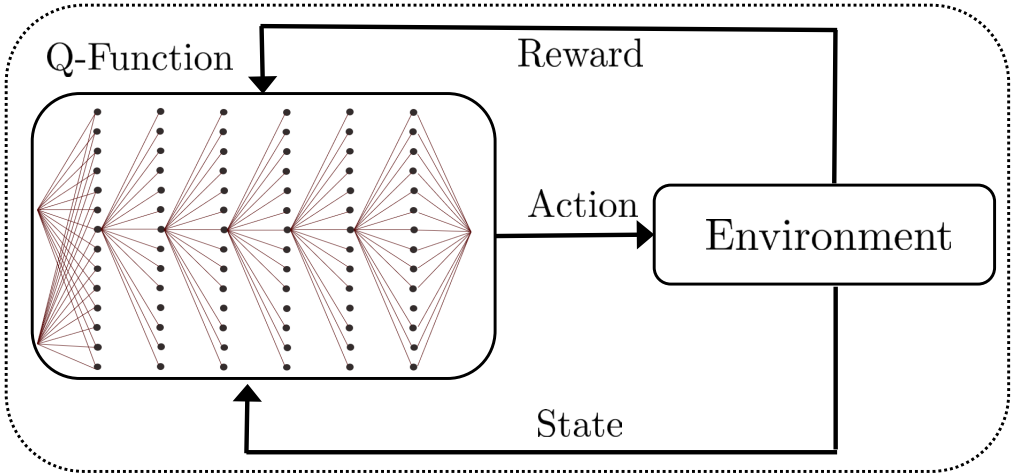
Action (run away)



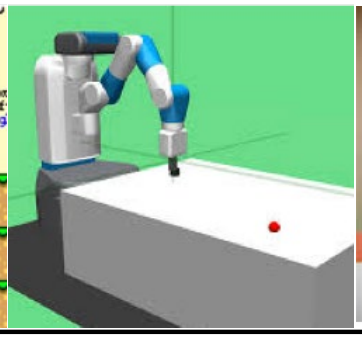
What can deep RL do well now?

Northeastern University
College of Engineering

- Acquire high degree of proficiency in domains governed by simple, known rules
- Learn simple skills with raw sensory inputs, given enough experience
- Learn from imitating enough human-provided expert behavior



Abundant-Interaction Domains



Atari games:

Q-learning:

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, et al. "Playing Atari with Deep Reinforcement Learning". (2013).

Policy gradients:

J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. "Trust Region Policy Optimization". (2015).
V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, et al. "Asynchronous methods for deep reinforcement learning". (2016).

Real-world robots:

Guided policy search:

S. Levine*, C. Finn*, T. Darrell, P. Abbeel. "End-to-end training of deep visuomotor policies". (2015).

Q-learning:

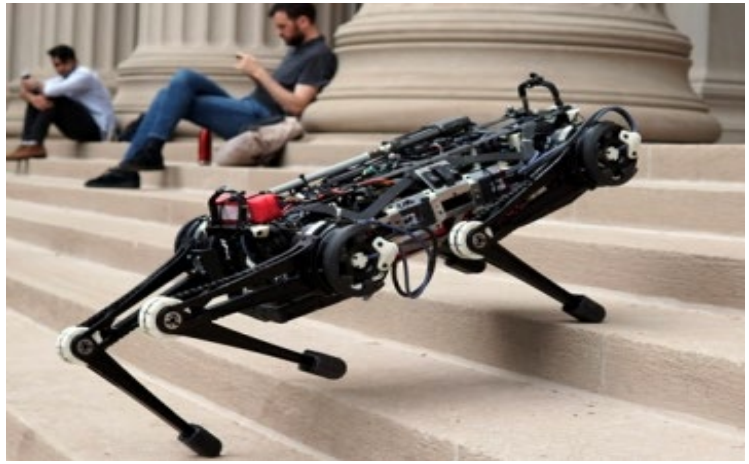
S. Gu*, E. Holly*, T. Lillicrap, S. Levine. "Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates". (2016).

Beating Go champions:

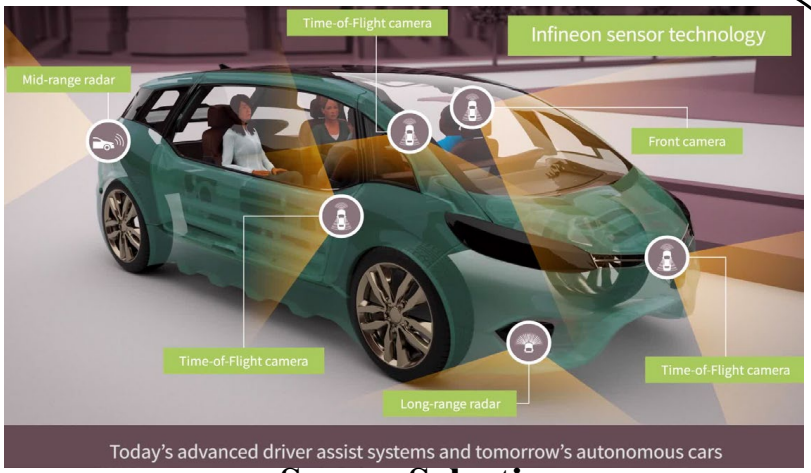
Supervised learning + policy gradients + value functions + Monte Carlo tree search:

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, et al. "Mastering the game of Go with deep neural networks and tree search". Nature (2016).

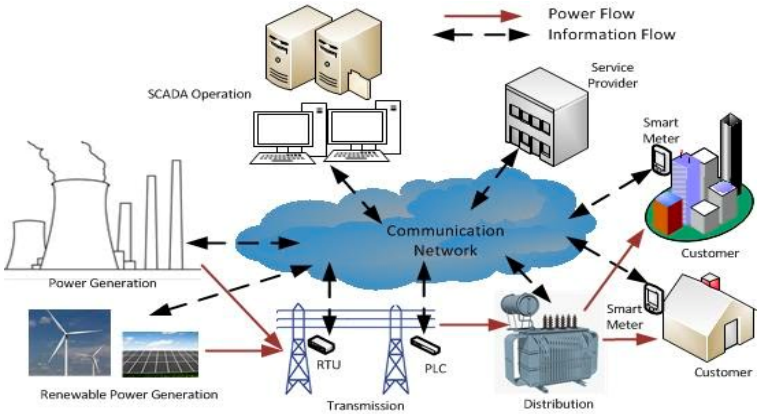
Other RL Applications



Quickest Modeling



Sensor Selection



Quickest Identification

Reliable/Safe Decision Making

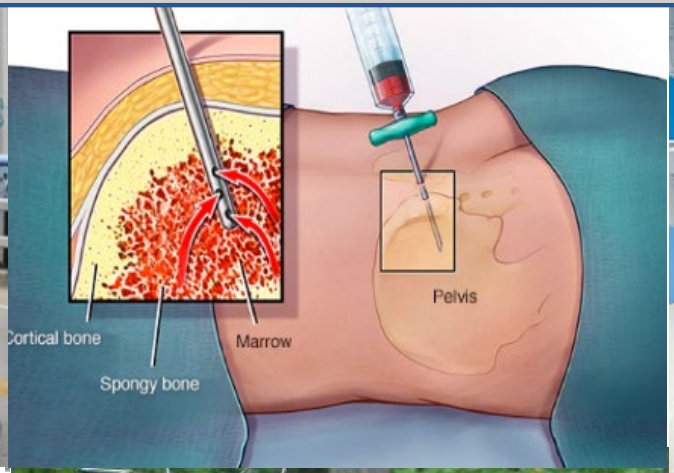
Practical Considerations

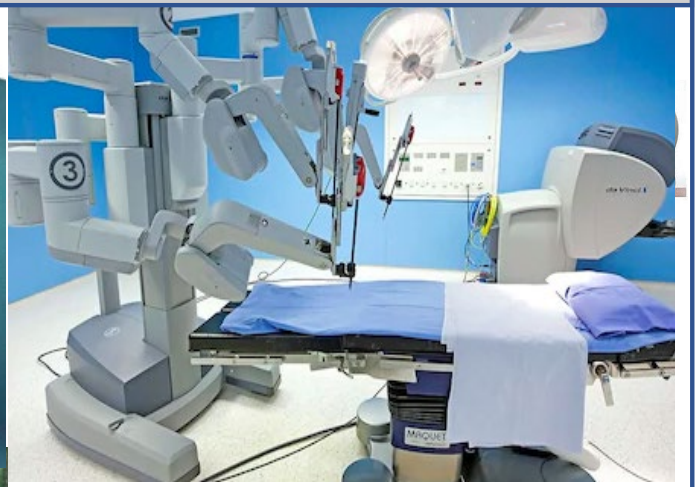
- Reliability
- Scalability
- Online Learning
- Efficiency
- Finite-Horizon
- Infinite-Horizon

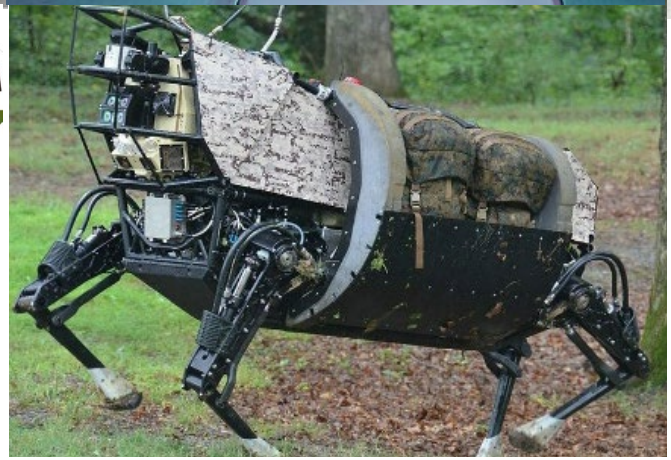
RL Challenges

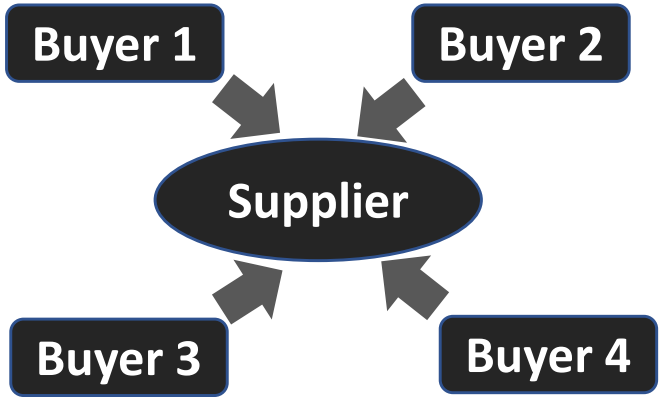
Northeastern University
College of Engineering

Limited-Interaction Domains





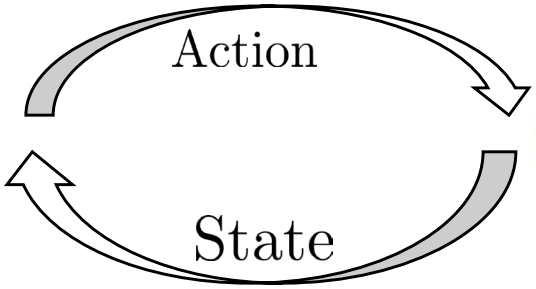




Practical Considerations

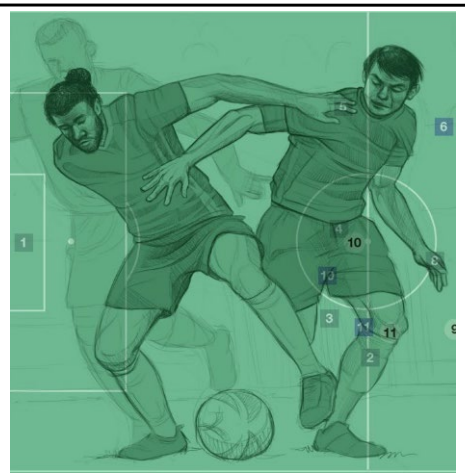
- ✓ Online/Real-Time Decision Making
- ✓ Non-Stationary Environments
- ✓ Unknown Dynamics and Lack of Access to Perfect Simulator
- ✓ Get Human/Experts Out of Loop
- ✓ Partial-Observability of Systems (e.g., Sensor or Technological Noise)
- ✓ Constraints (e.g., Ethical, Economical, Safety, Physical)

Other RL Impacts



Apprenticeship learning (Learning Policy in Mind of Experts)

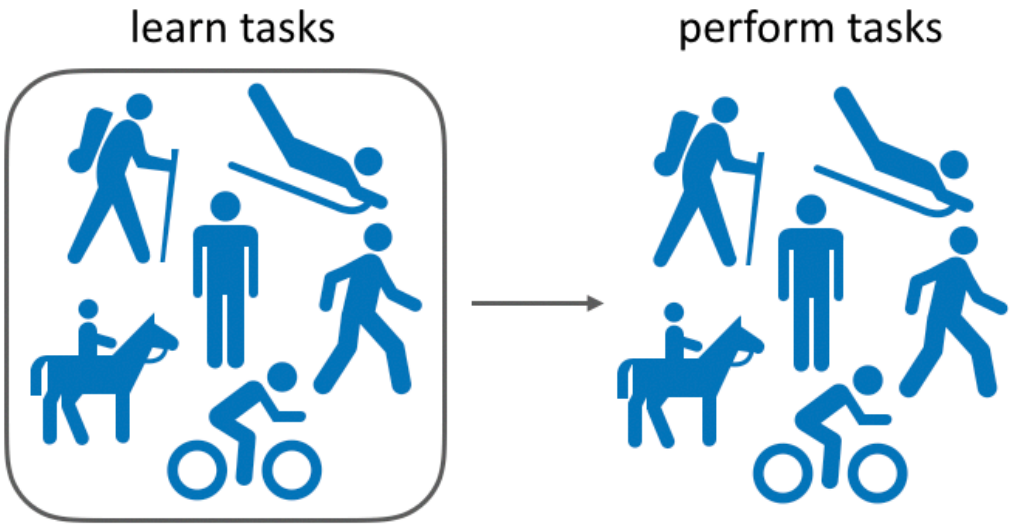
Imitation learning



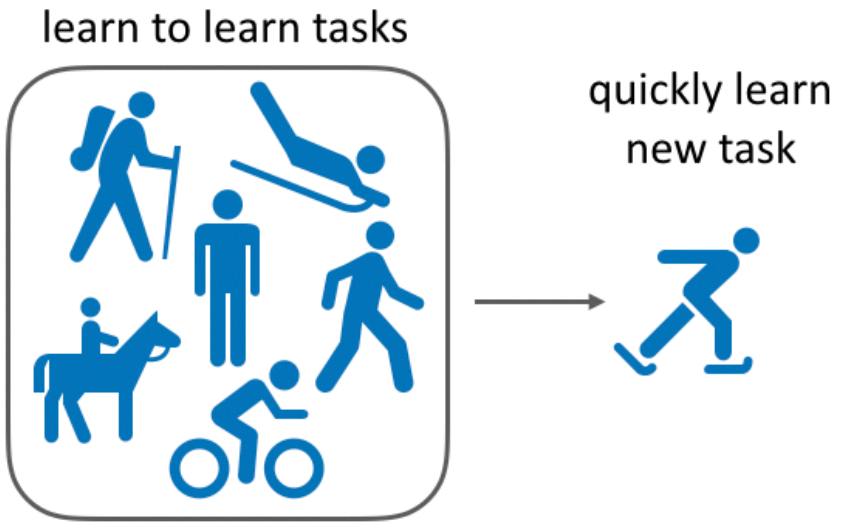
Other Topics:

Northeastern University
College of Engineering

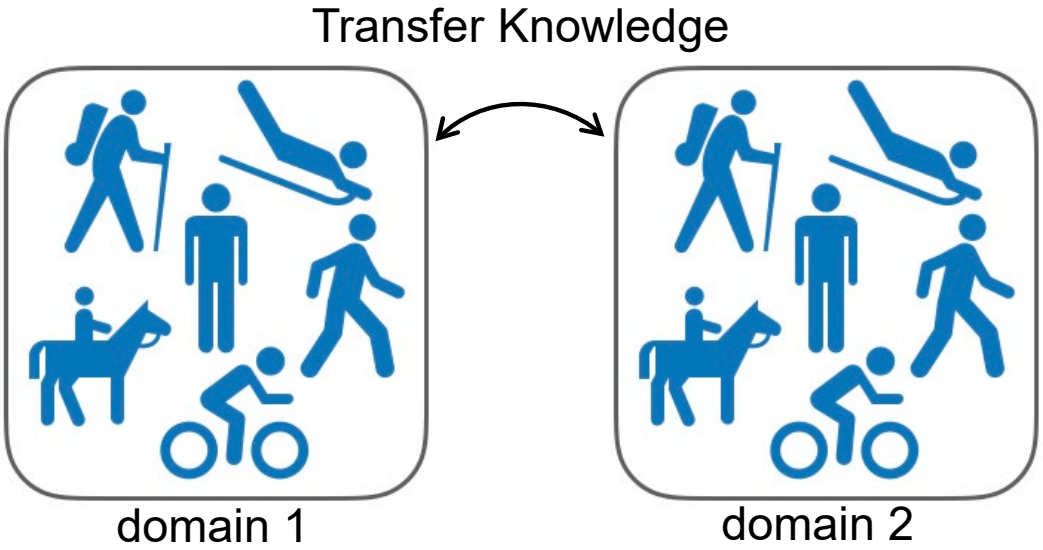
Multi-Task Reinforcement Learning



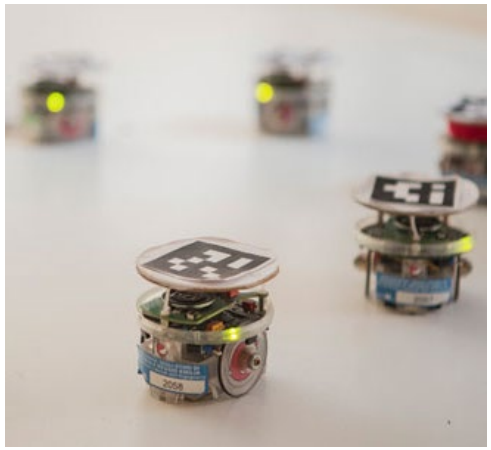
Meta Reinforcement Learning



Transfer Reinforcement Learning



Multi-Agent and Multi-Goal Reinforcement Learning

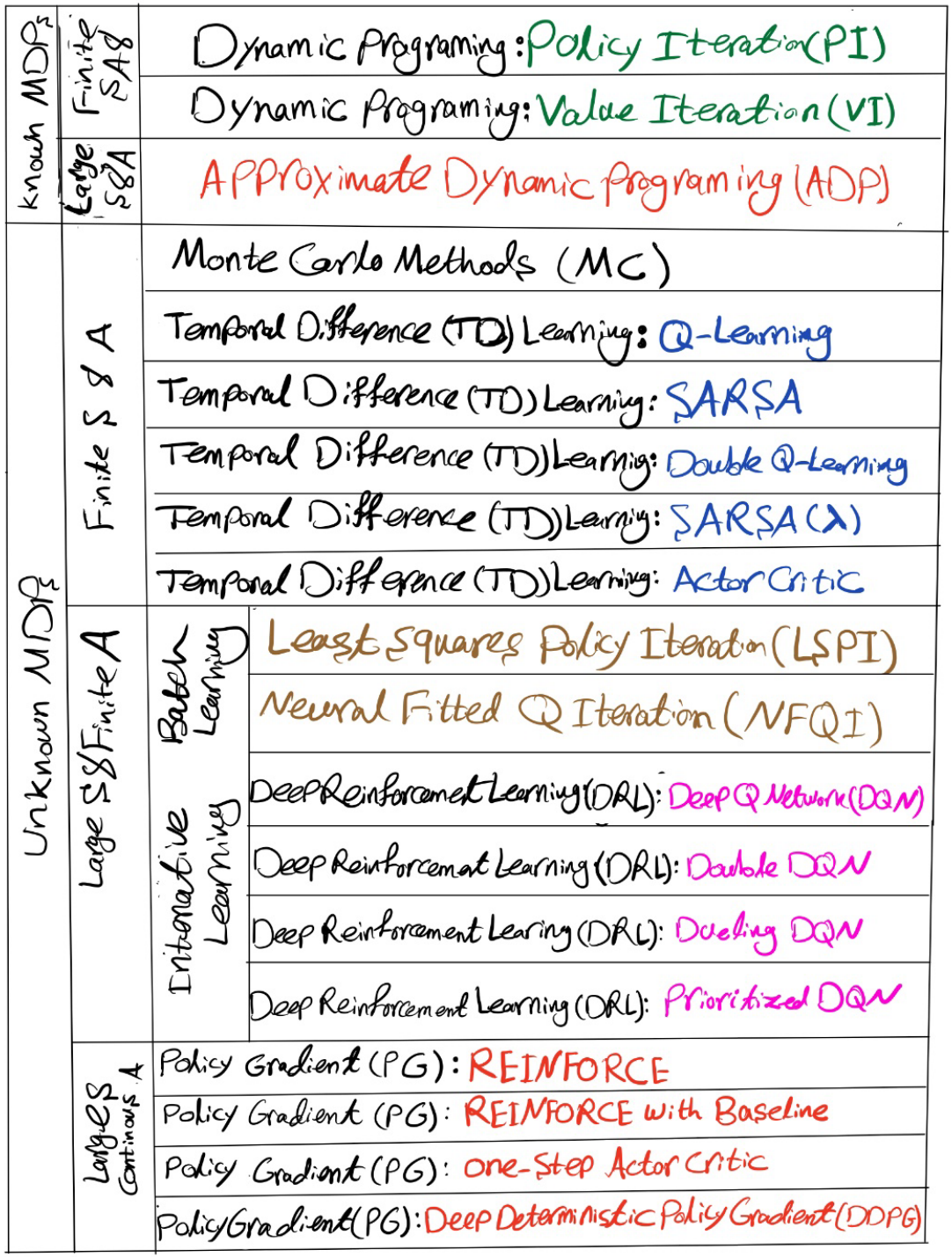


Main Topics/Recurring Issues:

- Learning (by trial and error)
- Planning (search, reason, thought, cognition)
- Prediction (evaluation functions, knowledge)
- Control (action selection, decision making)

What's coming?

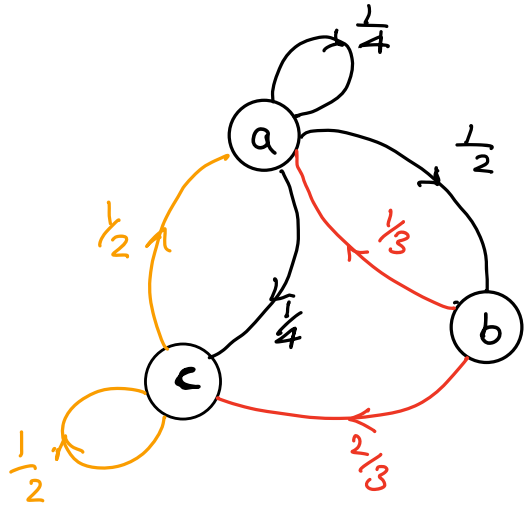
Northeastern University
College of Engineering



- Markov process

- Multi-Arm Bandits
    - Introduction
    - Exploration - Exploitation Delima

State Transition Diagram



$x_k \in \{a, b, c\}$ → any given k: state

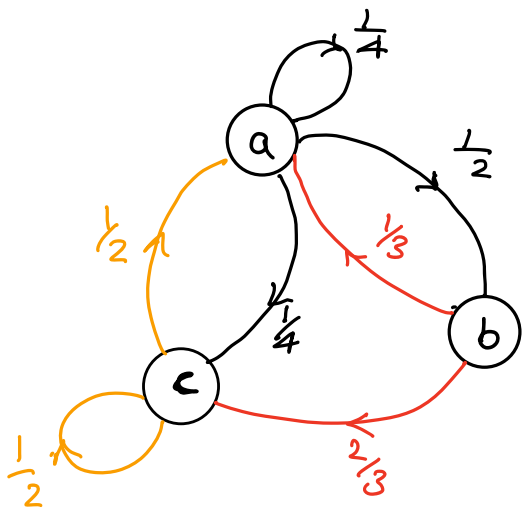
→ known

$$P(x_k = c \mid \underbrace{x_{k-1}}_{b}, \underbrace{x_{k-2}}_{a}, \underbrace{x_{k-3}}_{a}, \dots, x_0) = P(x_k \mid x_{k-1}) = \frac{2}{3}$$

(I) $P(A) = \sum_{b \in B} P(A, B = b) \Rightarrow P(A \mid c) = \sum_{b \in B} P(A, B = b \mid c)$

(II) $P(A, B) = P(A \mid B)\, P(B)$

(III) $P(A, B \mid c) = P(A \mid B, c)\, P(B \mid c)$



$$P(X_2 = b \mid X_1 = a, X_0 = c) = \frac{1}{2}$$

$$P(X_2 = b \mid X_0 = c) \overset{(I)}{=} \sum_{i \in X_1} P(X_2 = b, X_1 = i \mid X_0 = c)$$

$$= P(X_2 = b, X_1 = a \mid X_0 = c)$$

$$+ P(X_2 = b, X_1 = b \mid X_0 = c)$$

$$+ P(X_2 = b, X_1 = c \mid X_0 = c)$$

$$\overset{(II)}{=} P(X_2 = b \mid X_1 = a, X_0 = c)\, P(X_1 = a \mid X_0 = c)$$

$$+ P(X_2 = b \mid X_1 = b, X_0 = c)\, P(X_1 = b \mid X_0 = c)$$

($P(X_2 = b \mid X_1 = a, X_0 = c) = 1/2$; $P(X_1 = a \mid X_0 = c) = 1/2$)

$$+ P(X_2 = b \mid X_1 = c, X_0 = c)\, P(X_1 = c \mid X_0 = c)$$

$$= \frac{1}{4}$$