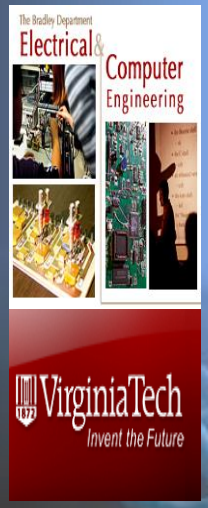


ECE 5984
Deep Reinforcement Learning

Jason J. Xuan, Ph.D.

Department of Electrical & Computer Engineering
Virginia Tech

Syllabus – ECE 5984

- **Instructor**
  - Dr. Jason Jianhua Xuan
  - Phone: (571) 858-3151 (O)
  - E-mail: xuan@vt.edu (For a quicker response, put “5984” in the subject line.)
  - Office location: Virginia Tech Research Center, Arlington, VA
  - Office hour: By appointment – email communication works better.

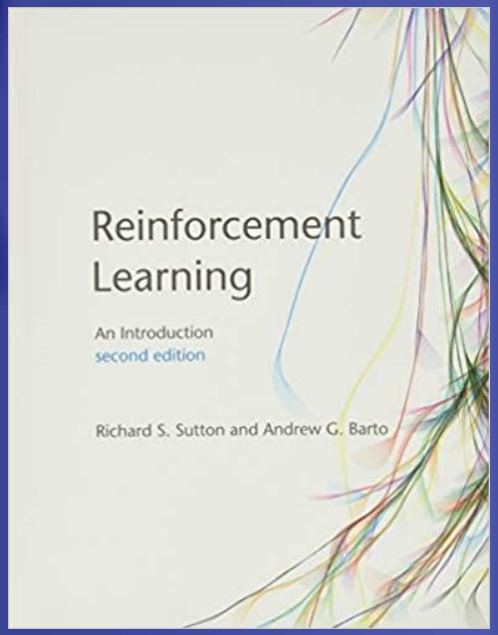
- **On-line information**
  - The Canvas system will be used for most course-related activities: canvas.vt.edu

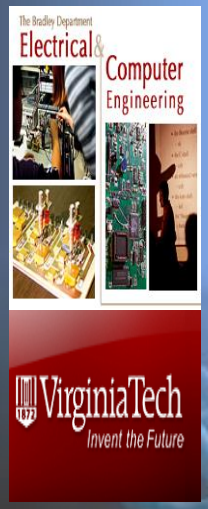
Textbook & Reference

- No Textbook Required
- *Suggested books*
  - Reinforcement Learning: An Introduction

    by Richard S. Sutton and Andrew G. Barto, published by

    A Bradford Book; second edition, 2018 (http://incompleteideas.net/book/the-book-2nd.html)





Grading & Policy

- **Grading**

Homework & Course projects	*30%*
Paper Review	*20%*
Leading Discussion	*10%*
Topic Presentations	*20%*
Final project	*20%*

- **General policies**
  - Students may discuss among themselves general approaches to tackling project problems. The final reports are expected to be the original work of each individual student.
  - Course projects may be done independently or with a partner.

Course Contents

1. Introduction to Reinforcement Learning

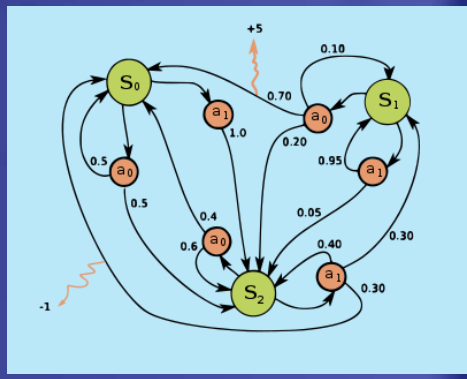
   MDPs, Dynamic Programming, Monte Carlo Methods, Temporal Difference Methods

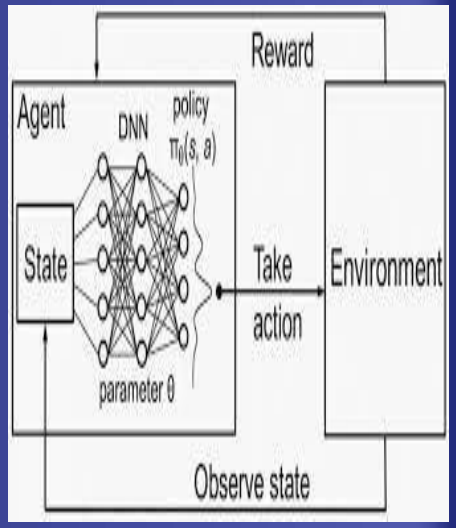
2. Introduction to Deep Learning

   MLPs, CNNs, RNNs

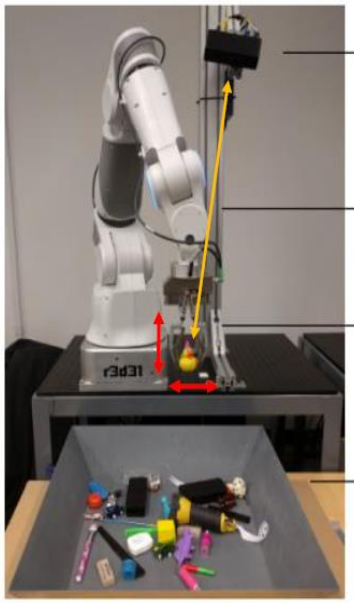
3. Deep Reinforcement Algorithms

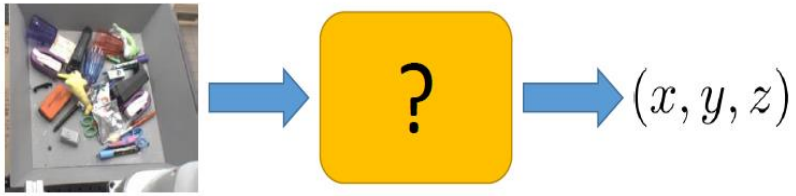
   Deep Q-learning, Double/Dueling DQN, Policy Gradient Methods, Actor-Critic Method





A Machine Learning Approach



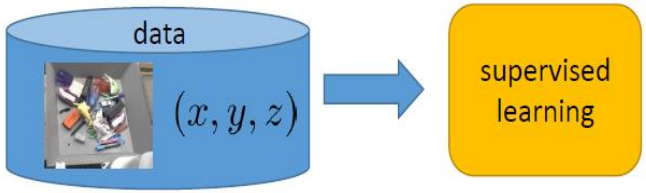


Option 1:

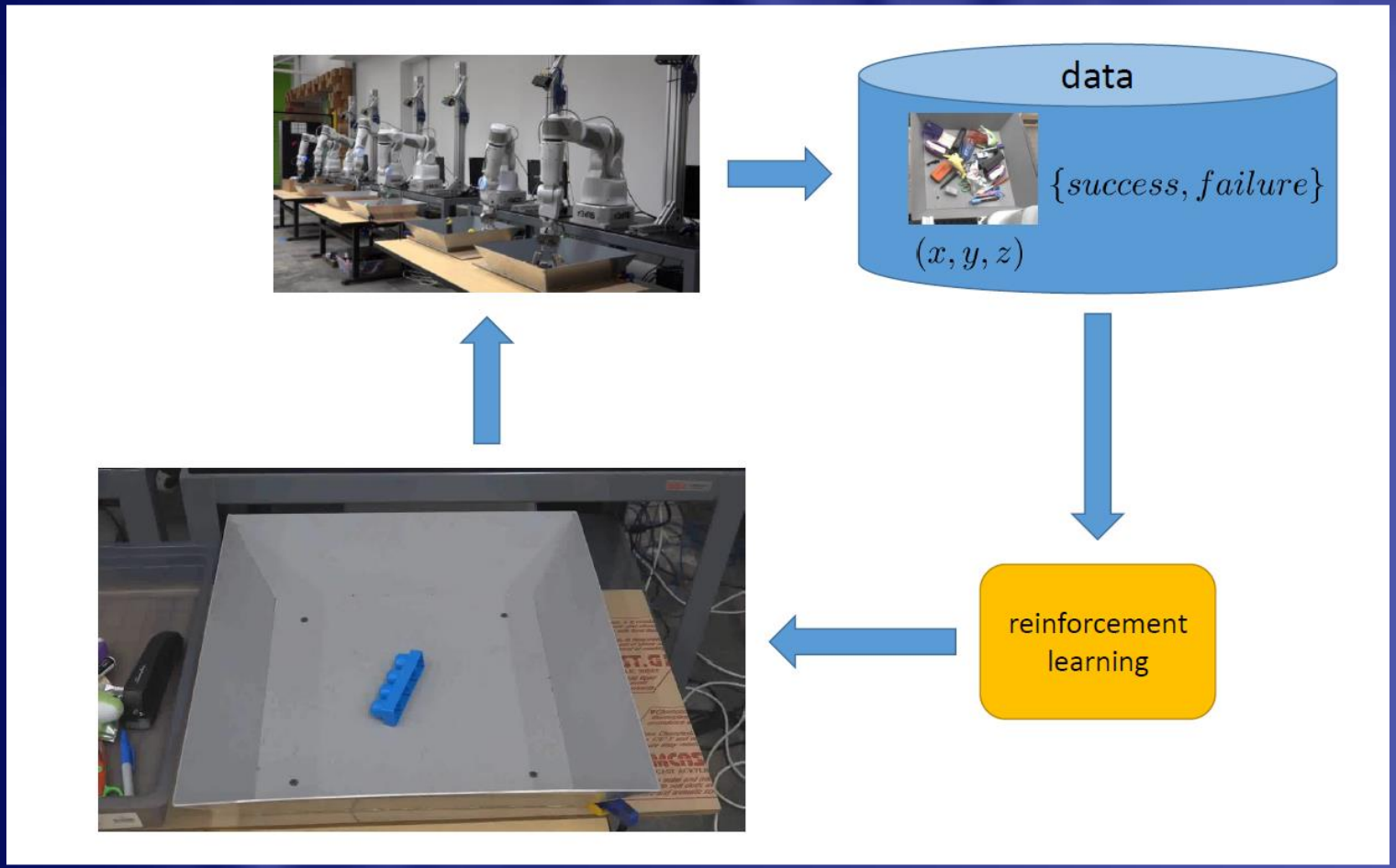
Understand the problem, design a solution

Option 2:

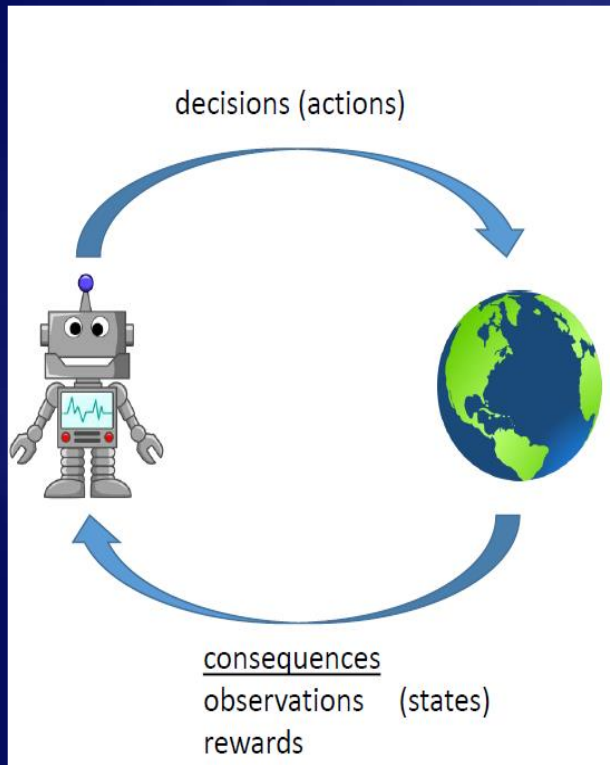
Set it up as a machine learning problem

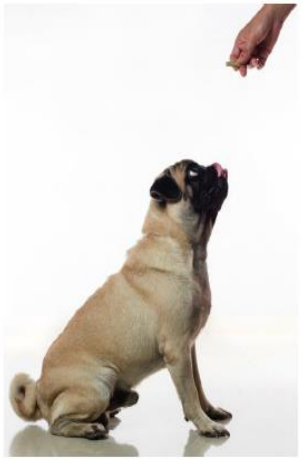


Reinforcement Learning



Reinforcement Learning (cont'd)





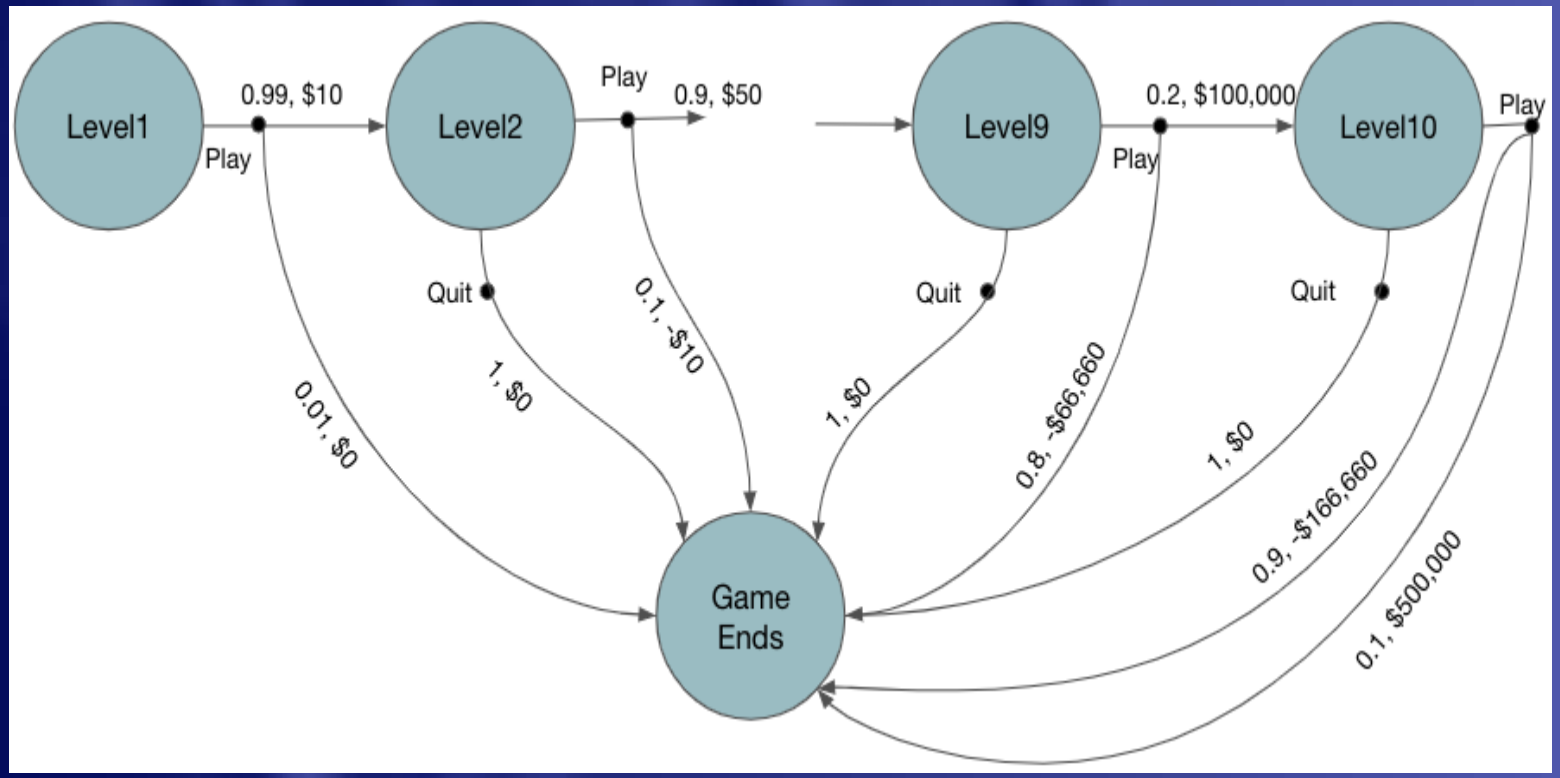
Actions: muscle contractions
Observations: sight, smell
Rewards: food

Actions: motor current or torque
Observations: camera images
Rewards: task success measure (e.g., running speed)

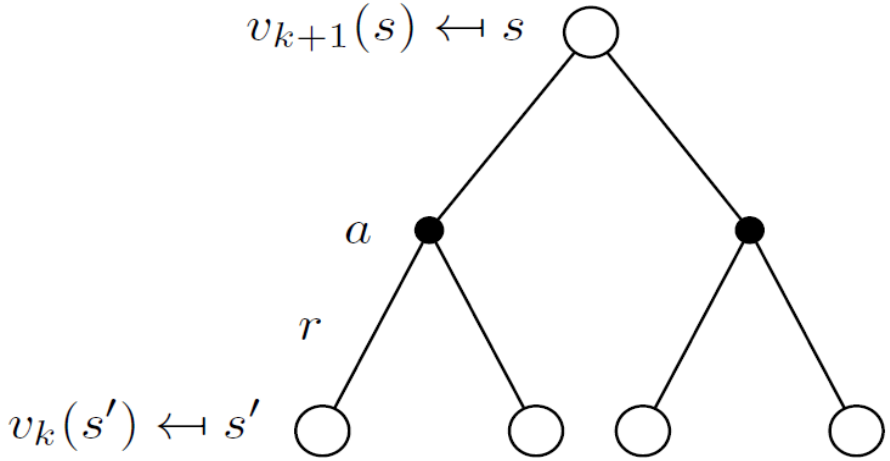
Actions: what to purchase
Observations: inventory levels
Rewards: profit

Markov Decision Process (MDP)

An Example: Quiz Game Show



Dynamic Programming

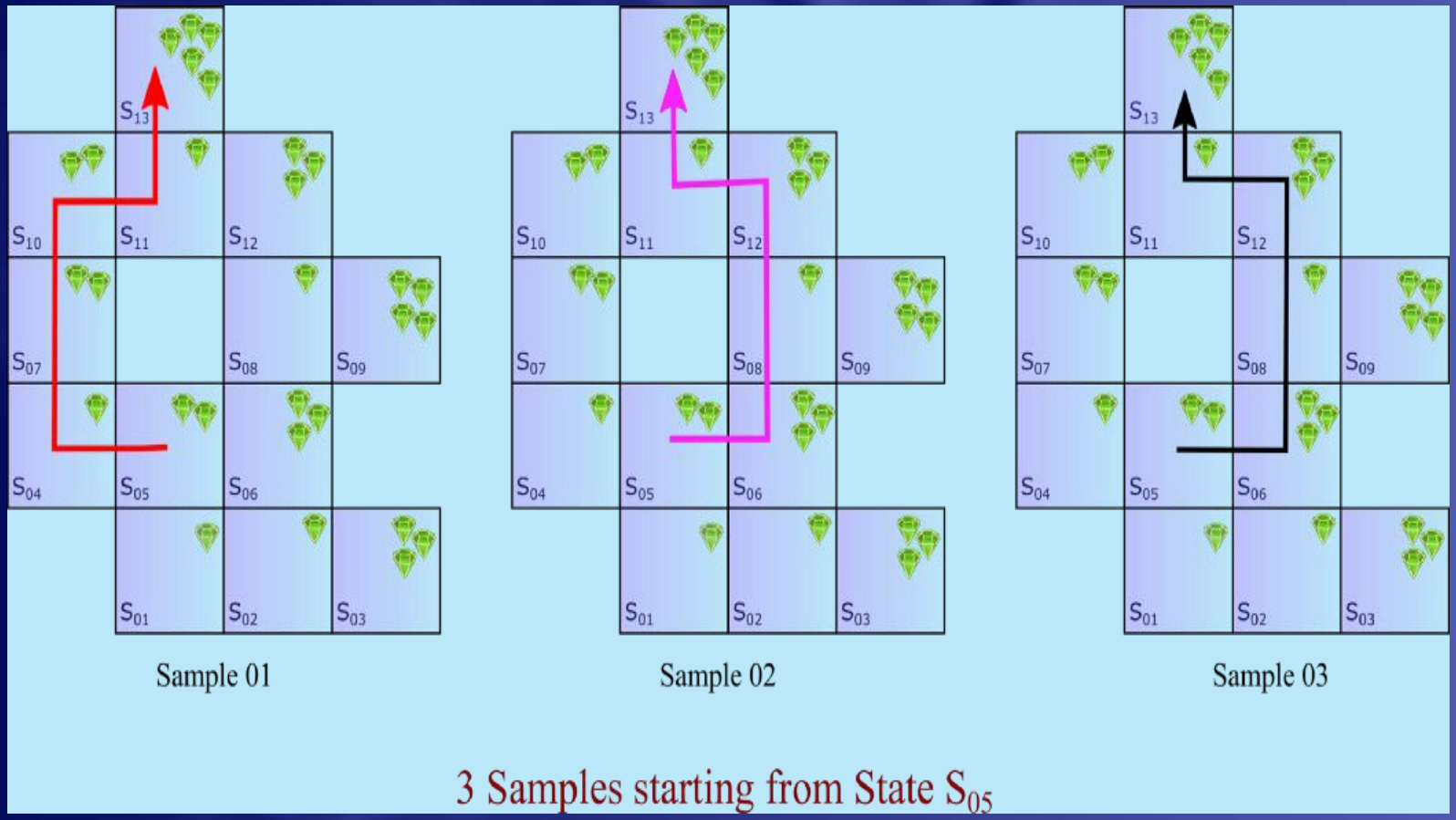


$$v_{k+1}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_k(s') \right)$$

$$\mathbf{v}^{k+1} = \boldsymbol{\mathcal{R}}^{\boldsymbol{\pi}} + \gamma \boldsymbol{\mathcal{P}}^{\boldsymbol{\pi}} \mathbf{v}^k$$

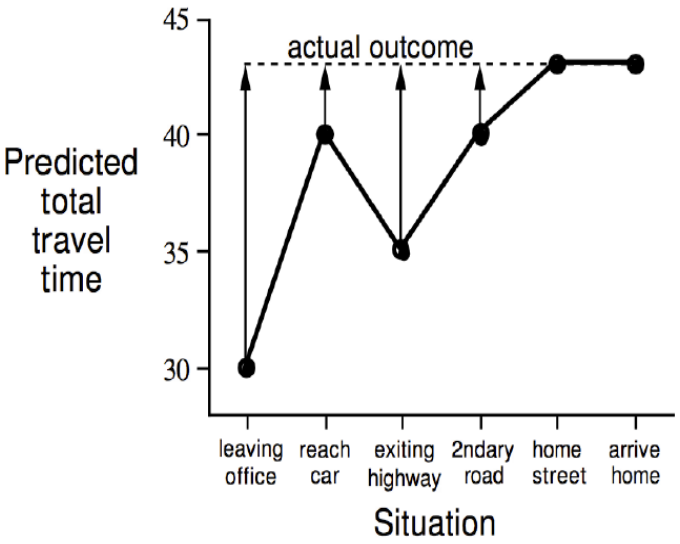
Monte Carlo Method:

An Example: Gem Collection

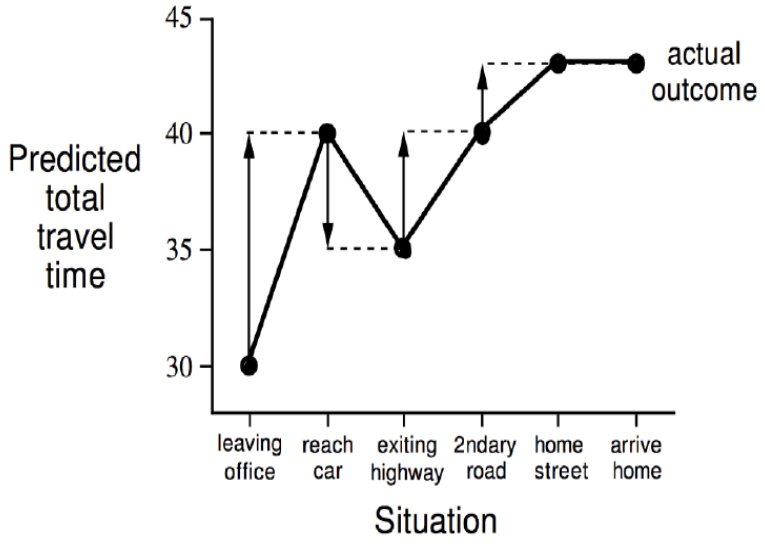


Temporal Difference

Changes recommended by Monte Carlo methods (α=1)



Changes recommended by TD methods (α=1)



Sarsa: On-Policy Control

Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$

Repeat (for each episode):

  Initialize S

  Choose A from S using policy derived from Q (e.g., ε-greedy)

  Repeat (for each step of episode):

    Take action A, observe R, S'

    Choose A' from S' using policy derived from Q (e.g., ε-greedy)

    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S',A') - Q(S,A)\right]$

    $S \leftarrow S'$; $A \leftarrow A'$;

  until S is terminal

Q-learning: Algorithm

Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$

Repeat (for each episode):

  Initialize S

  Repeat (for each step of episode):

    Choose A from S using policy derived from Q (e.g., ε-greedy)

    Take action A, observe R, S'

    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$

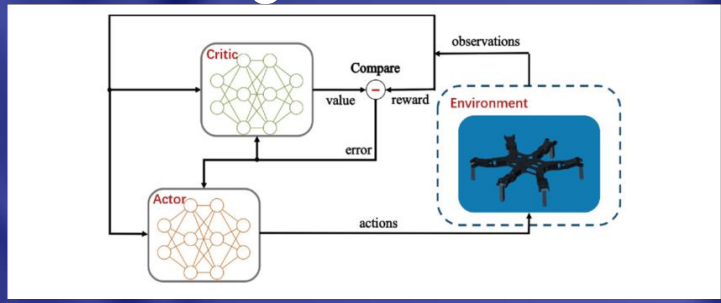
    $S \leftarrow S'$;

  until S is terminal

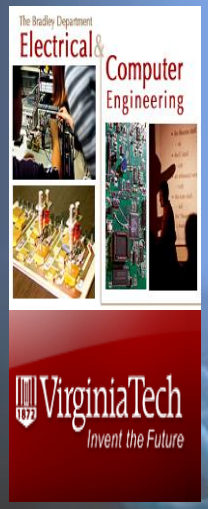
Course Contents (cont'd)

4. Advanced Deep Reinforcement Algorithms

- Deterministic Policy Gradients;

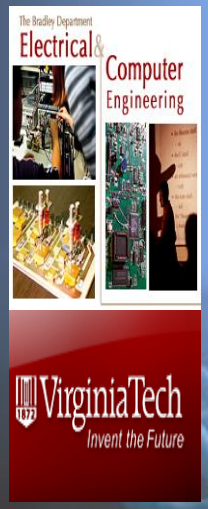


- Distributional Reinforcement Learning;
- Policy Gradients with Action-Dependent Baselines;
- Path-Consistency Learning;
- Exploration - Intrinsic Motivation & Unsupervised Reinforcement Learning;
- Model-Based Reinforcement Learning;
- Reinforcement Learning in the Real World



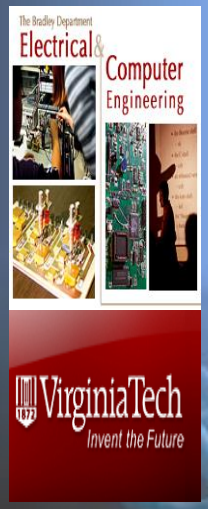
Requirement – Class Participation

- Read the assigned papers before each class
- Actively participate in discussions in class.
- If you are unable to attend a specific class, please let me know ahead of time via email (and have a good excuse).



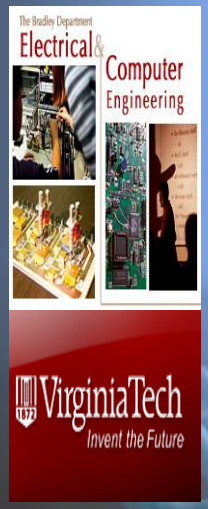
Requirement – Paper Review

- One page review of the selected paper
- Write in your own words
- Due date
  - 2 PM, the day of the class (i.e. on Tuesdays and Thursdays).
- Submission via Canvas



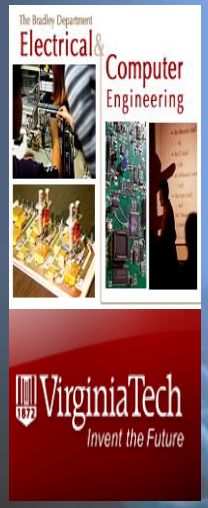
Paper Review – Suggested Structure

- Short summary of the paper
- Main contributions
- Strengths and weaknesses?
- Are the experiments convincing?
- How could the work be extended?
- Additional comments, including unclear points, open research questions, and applications.



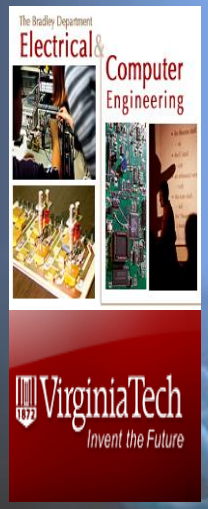
Requirement – Leading Discussion

- ~ One of you will be assigned to argue for the paper
- ~ One of you will be assigned to argue against the paper
- Come prepared with 5 points



Requirement – Topic Presentation

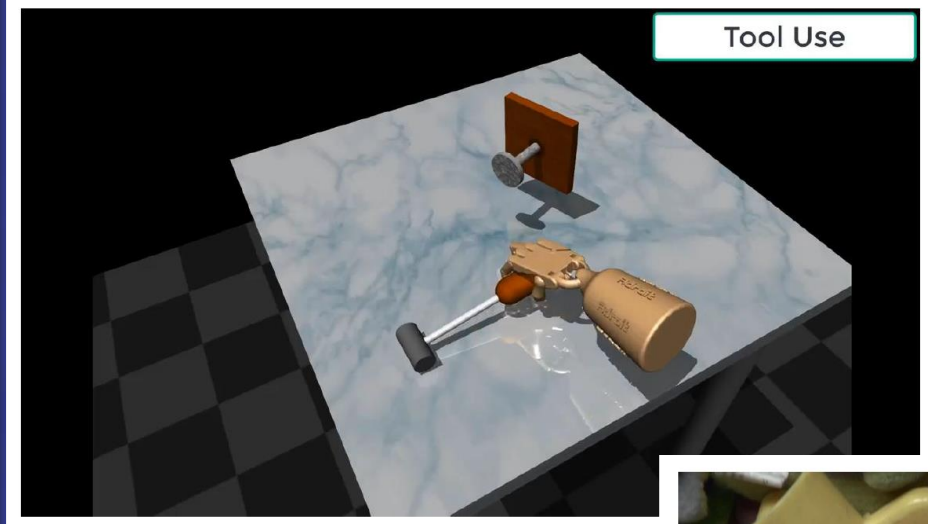
- 30 minutes in-class presentation
- Submit a complete set of slides prior the talk
- IMPORTANT: Don’t present papers – present the topic!



Topic Presentation - Structure

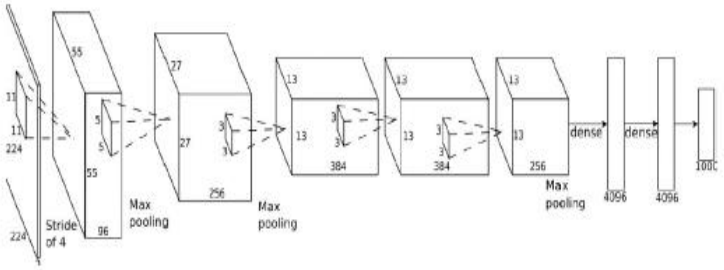
- High-level topic overview
- Main motivation
- Clear statement of the problem
- Overview of the technical approach
- Strengths/weaknesses of the approach
- Overview of the experimental evaluation
- Strengths/weaknesses of evaluation
- Discussion: future direction, links to other work

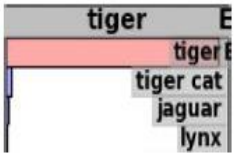
Complex Tasks

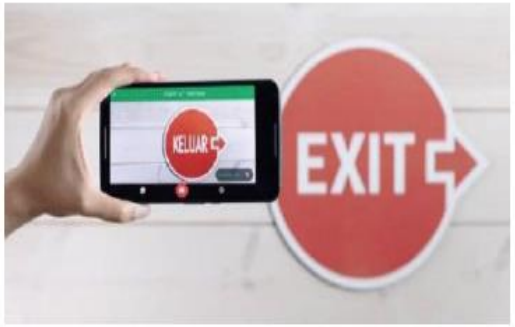


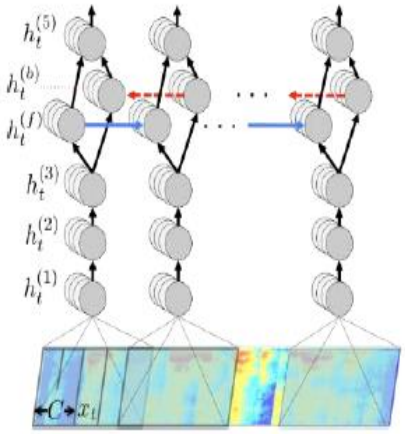
In the real world

Deep Learning: Unstructured Environments

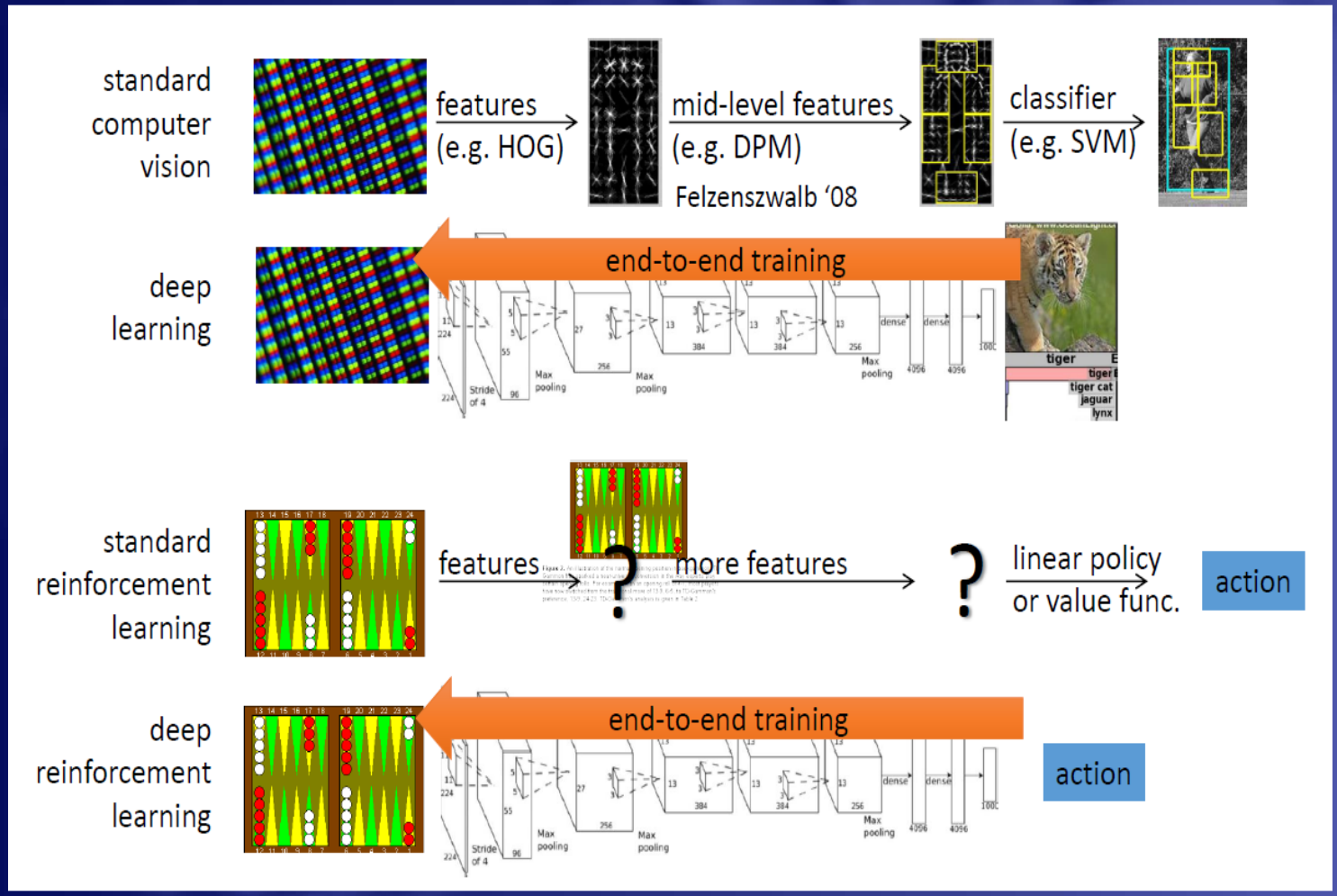






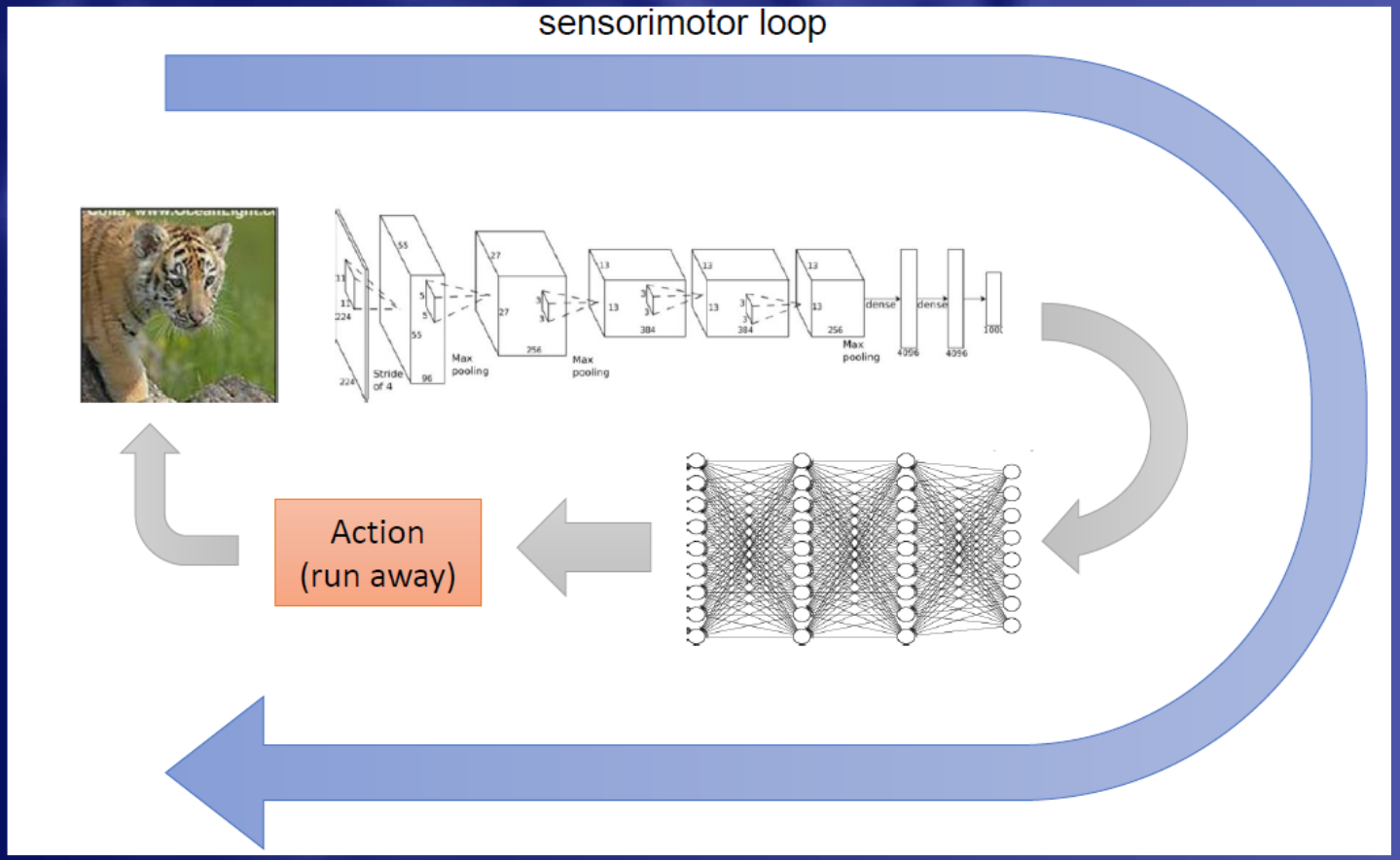


Deep Reinforcement Learning

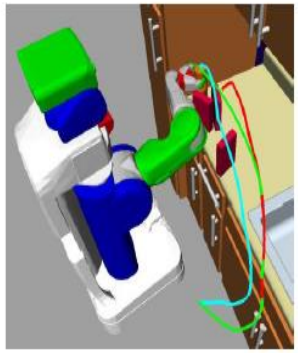


End-to-End Learning: Sequential Decision Making

- Perception & Action – A Sensorimotor Loop

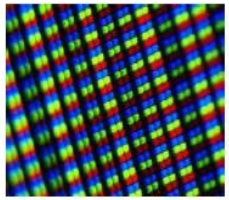


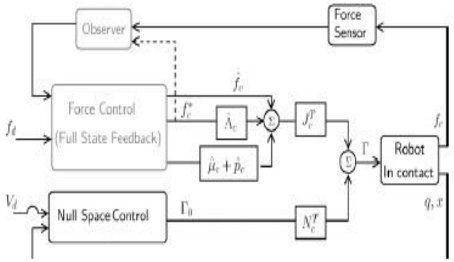
An Example



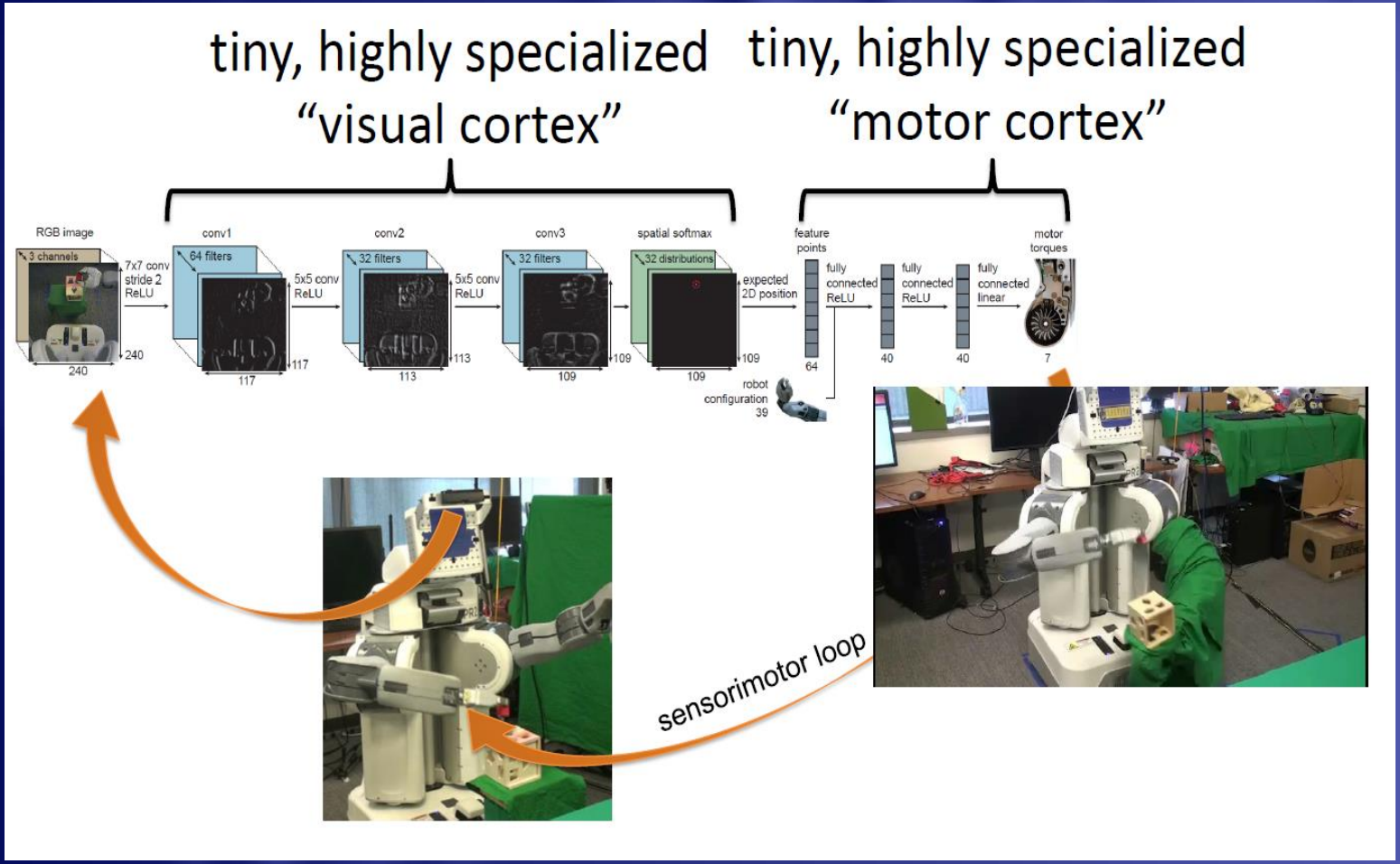
robotic control pipeline

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

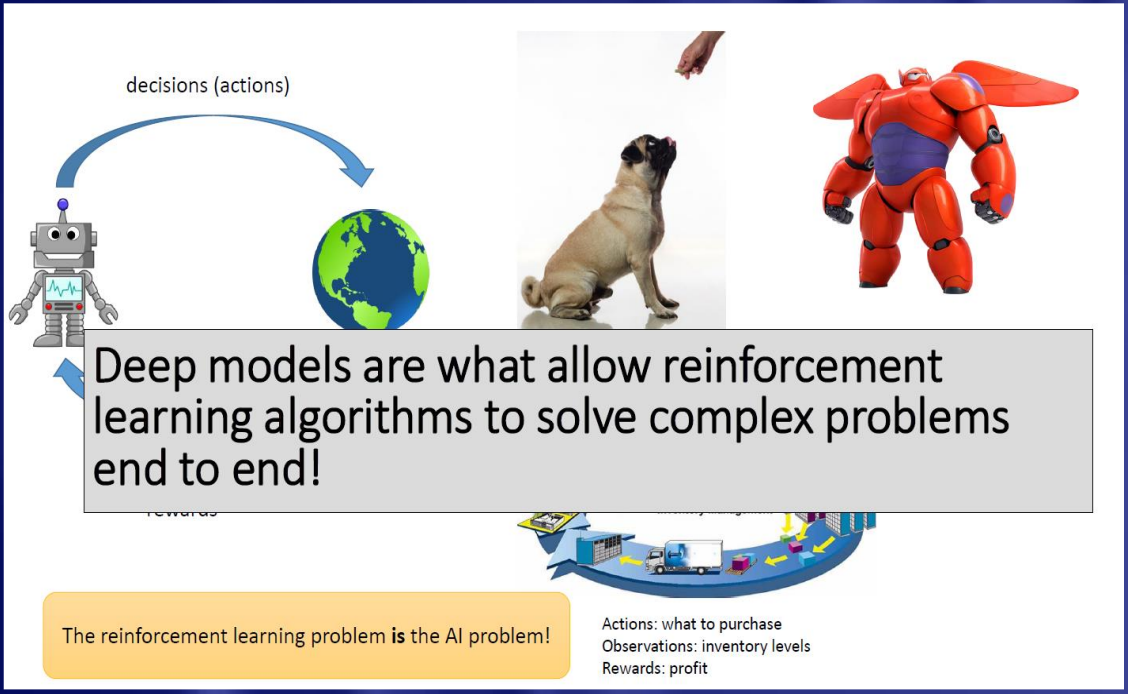




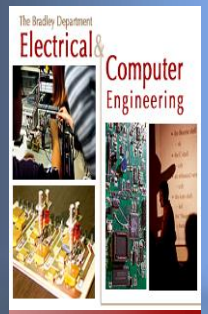
An Example (cont'd)



Deep Reinforcement Learning



- Deep = can process complex sensory input
  - ...and also compute really complex functions
- Reinforcement learning = can choose complex actions



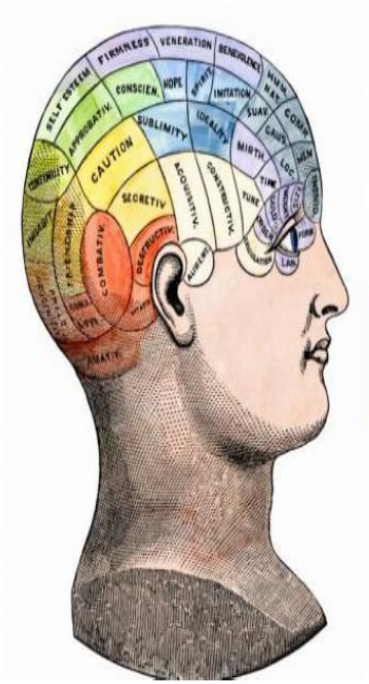
Forms of Supervision?

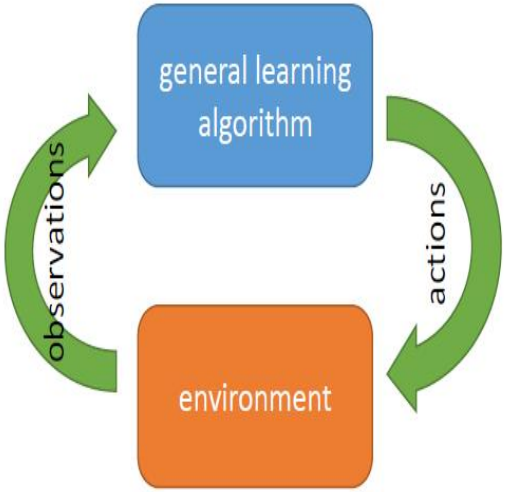
- Learning from demonstrations
  - Directly copying observed behavior
  - Inferring rewards from observed behavior (inverse reinforcement learning)
- Learning from observing the world
  - Learning to predict
  - Unsupervised learning
- Learning from other tasks
  - Transfer learning
  - Meta-learning: learning to learn

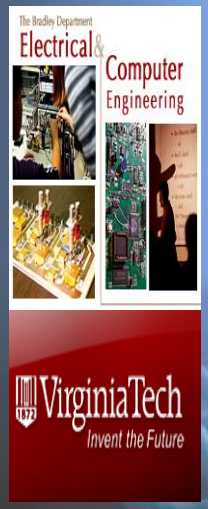
(General) Artificial Intelligence

Instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain.

- Alan Turing





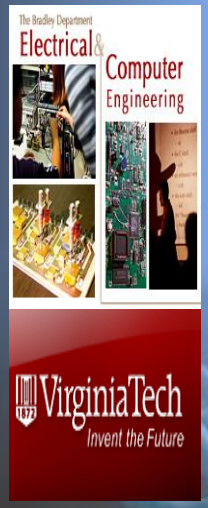


Assignment

- Homework Assignment #0 (Preparation)
  - Your first reinforcement learning project:

Getting Started with Gym:

https://gym.openai.com/docs/



Question

- Comments are more than welcome!
- About yourself?