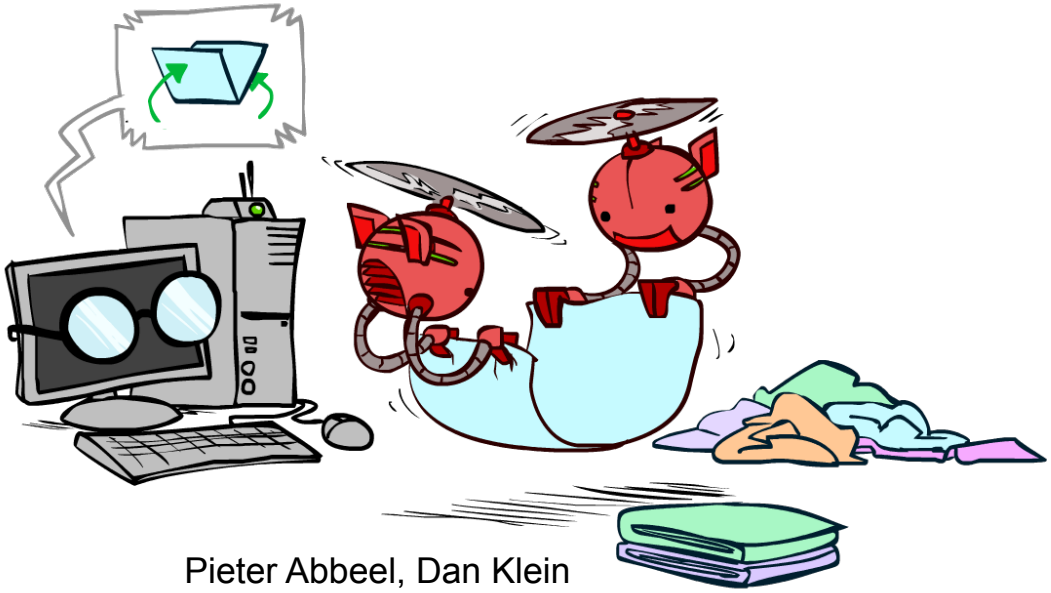
CS 188: Artificial Intelligence

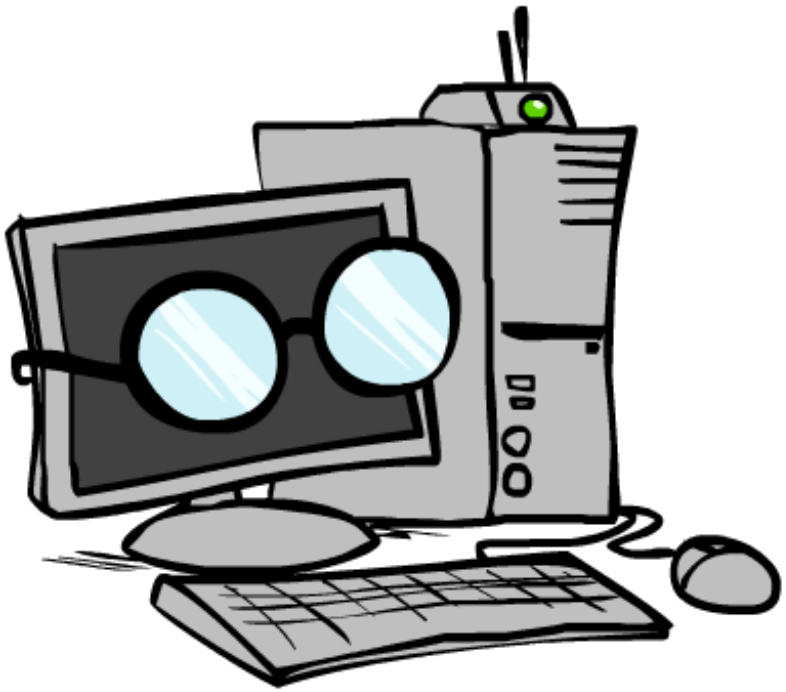
Advanced Applications: Computer Vision and Robotics*



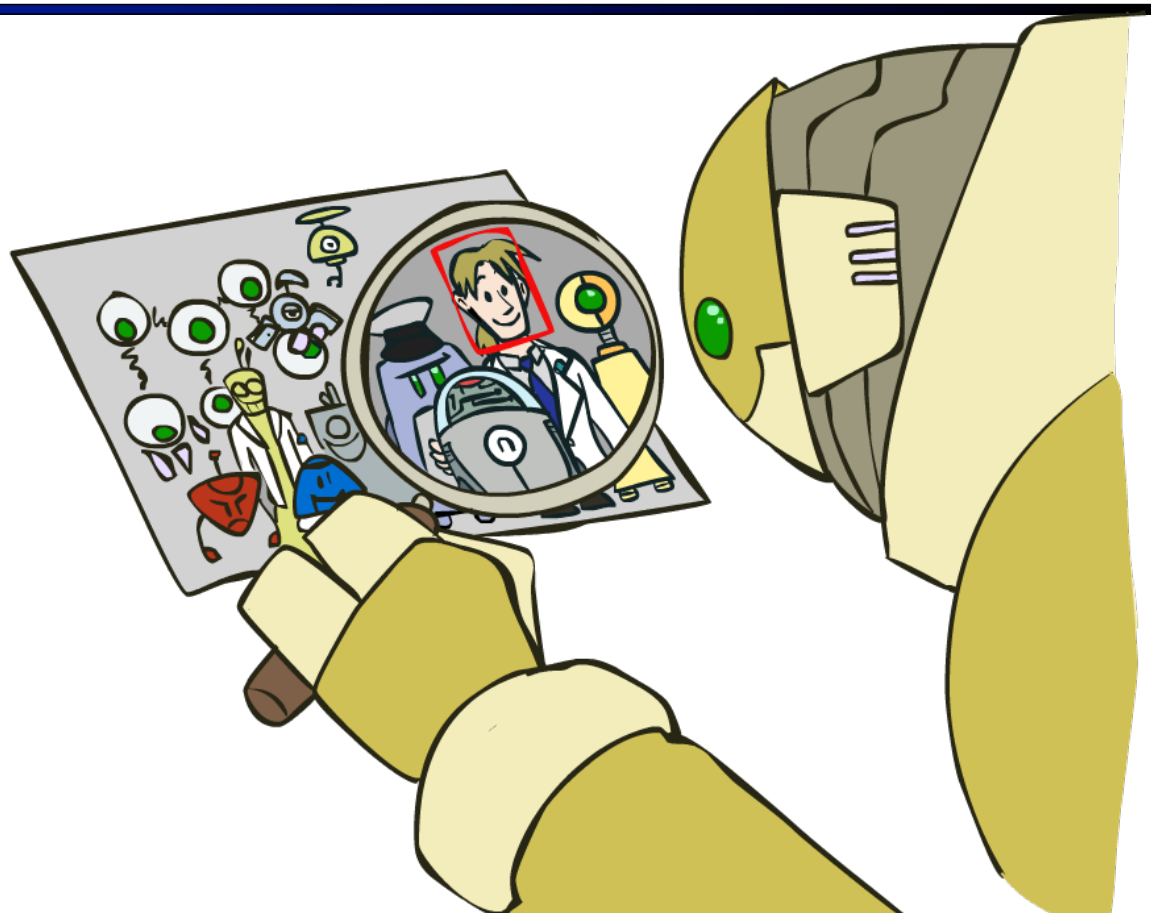
Pieter Abbeel, Dan Klein

University of California, Berkeley

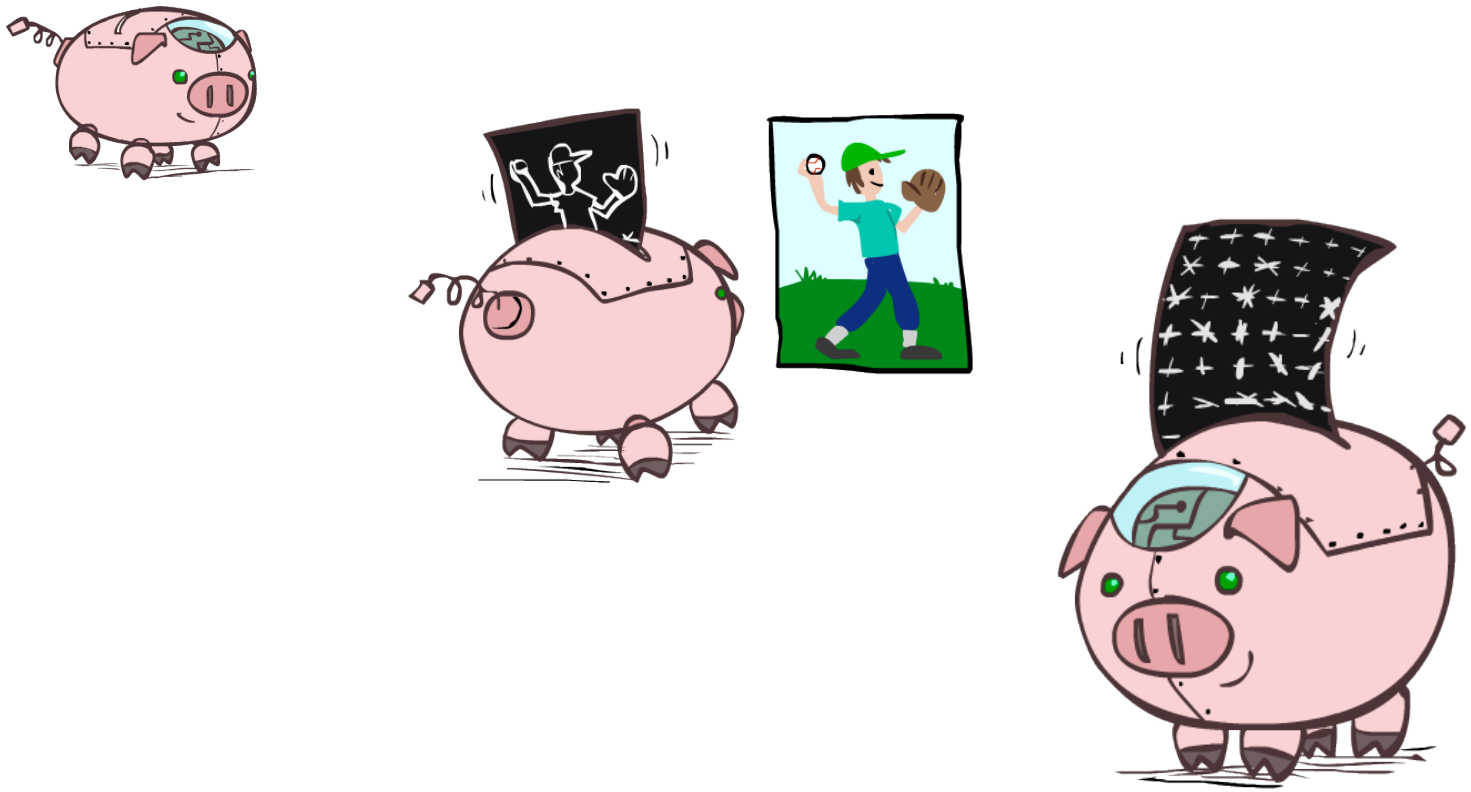
Computer Vision



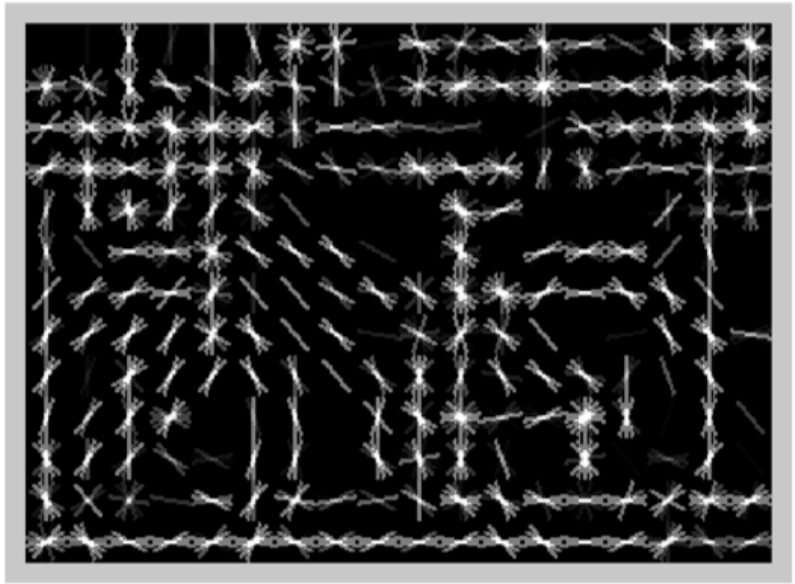
Object Detection



Object Detection Approach 1: HOG + SVM



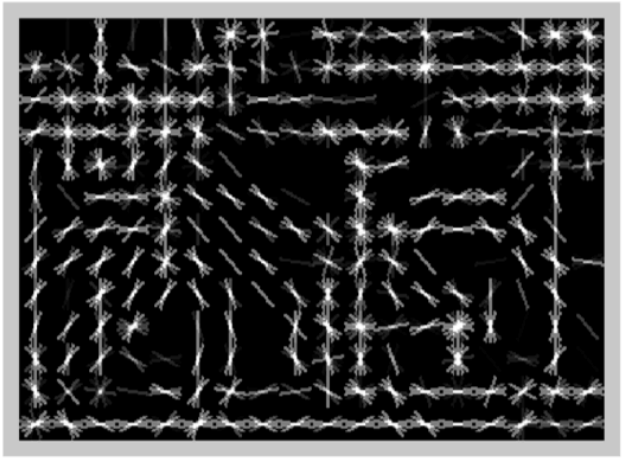
Features and Generalization



[Dalal and Triggs, 2005]

Features and Generalization

Image



HoG

Training

- Round 1
  - Training set =
    - Positive examples: from labeling
    - Negative examples: random patches

  → preliminary SVM
- Round 2 ("bootstrapping" or "mining hard negatives")
  - Training set =
    - Positive examples: from labeling
    - Negative examples: patches that have score >= -1

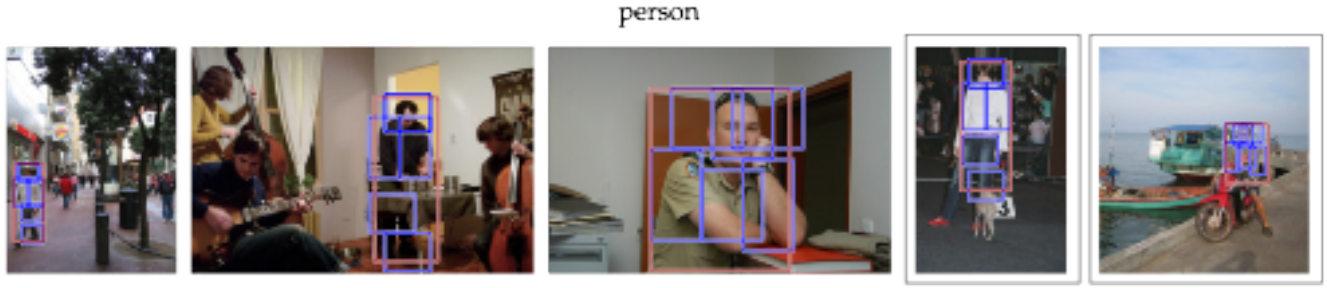
  → final SVM

State-of-the-art Results

[Girschik, Felzenszwalb, McAllester]

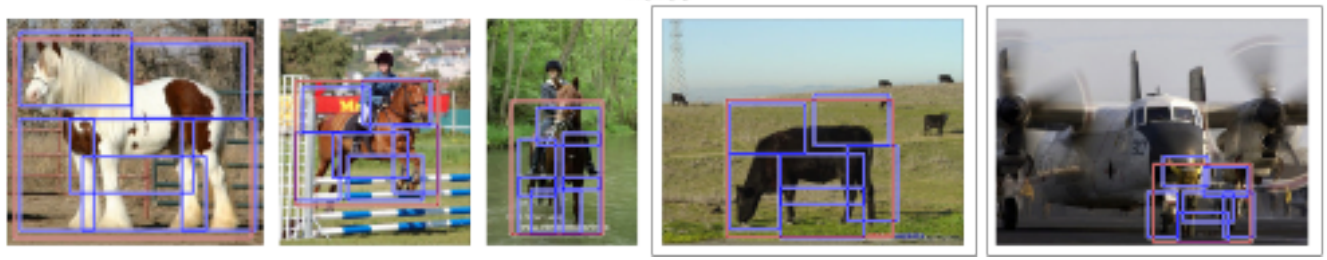
State-of-the-art Results

person



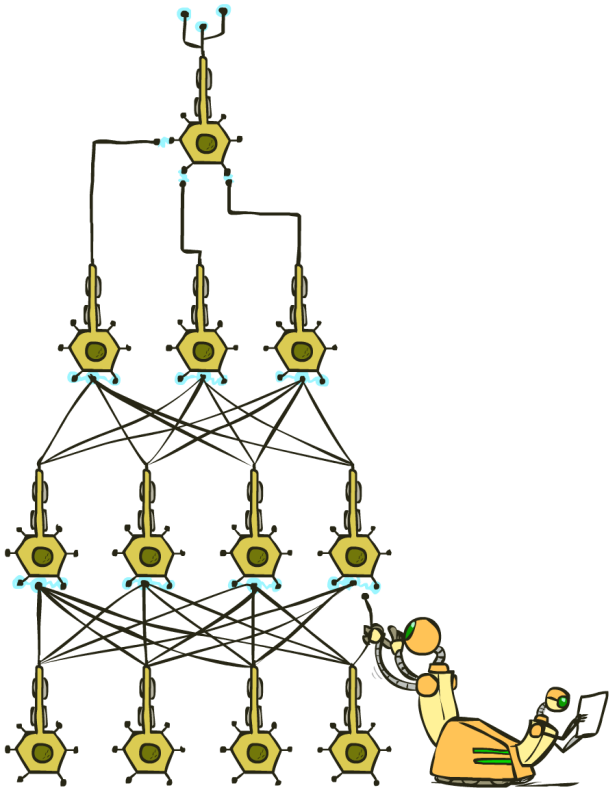
car

horse



[Girschik, Felzenszwalb, McAllester]

Object Detection Approach 2: Deep Learning



How Many Computers to Identify a Cat?

HOME PAGE | TODAY'S PAPER | VIDEO | MOST POPULAR | U.S. Edition

Subscribe: Digital / Home Delivery | Log In | Register Now | Help

The New York Times — Business Day — Technology

Search All NYTimes.com

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION | ARTS | STYLE | TRAVEL | JOBS | REAL ESTATE | AUTOS

How Many Computers to Identify a Cat? 16,000

An image of a cat that a neural network taught itself to recognize.

By JOHN MARKOFF
Published: June 25, 2012

MOUNTAIN VIEW, Calif. — Inside Google's secretive X laboratory, known for inventing self-driving cars and augmented reality glasses, a small group of researchers began working several years ago on a simulation of the human brain.

There Google scientists created one of the largest neural networks for machine learning by connecting 16,000 computer processors, which they turned loose on the Internet to learn on its own.

Presented with 10 million digital images found in YouTube videos, what did Google's brain do? What millions of humans do with YouTube: looked for cats.

Multimedia

Business Day Live | Google's Brain

FACEBOOK
TWITTER
GOOGLE+
E-MAIL
SHARE
PRINT
REPRINTS

Log in to see what your friends are sharing on nytimes.com. Privacy Policy | What's This?
Log In With Facebook

What's Popular Now

Restaurant Review: Guy's American Kitchen & Bar in Times Square

Occupy Offshoot Aims to Erase People's Debts

Today's Headlines Daily E-Mail

Sign up for a roundup of the day's top stories, sent every morning.
Sign Up
See Sample | Privacy Policy

Subscribe to Technology RSS Feeds

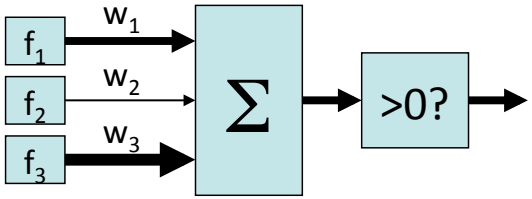
Technology News
Internet
Start-Ups
Business Computing
Companies
Bits Blog
Personal Tech
Pogue's Posts

MOST E-MAILED
MOST VIEWED

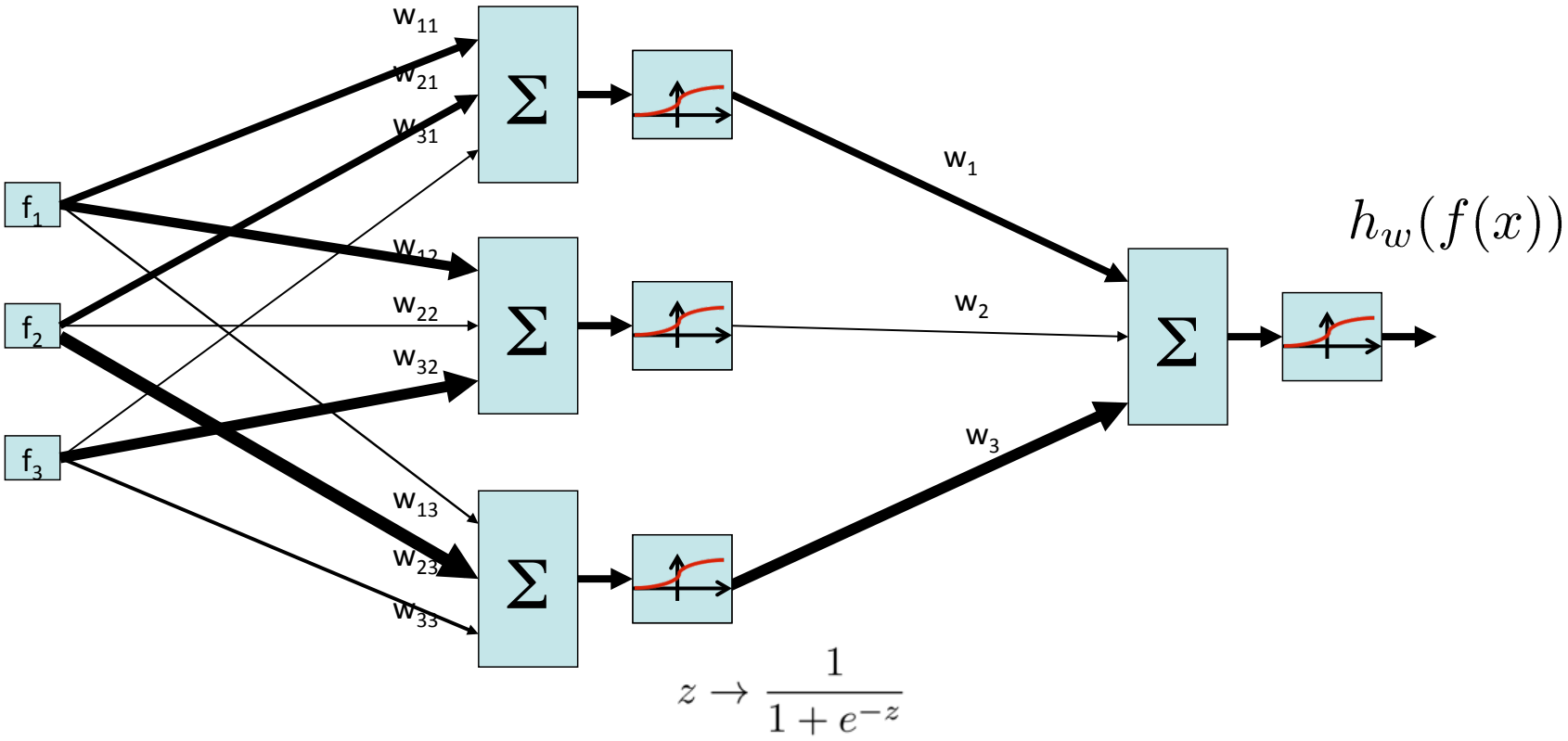
"Google Brain"

[Le, Ng, Dean, et al, 2012]

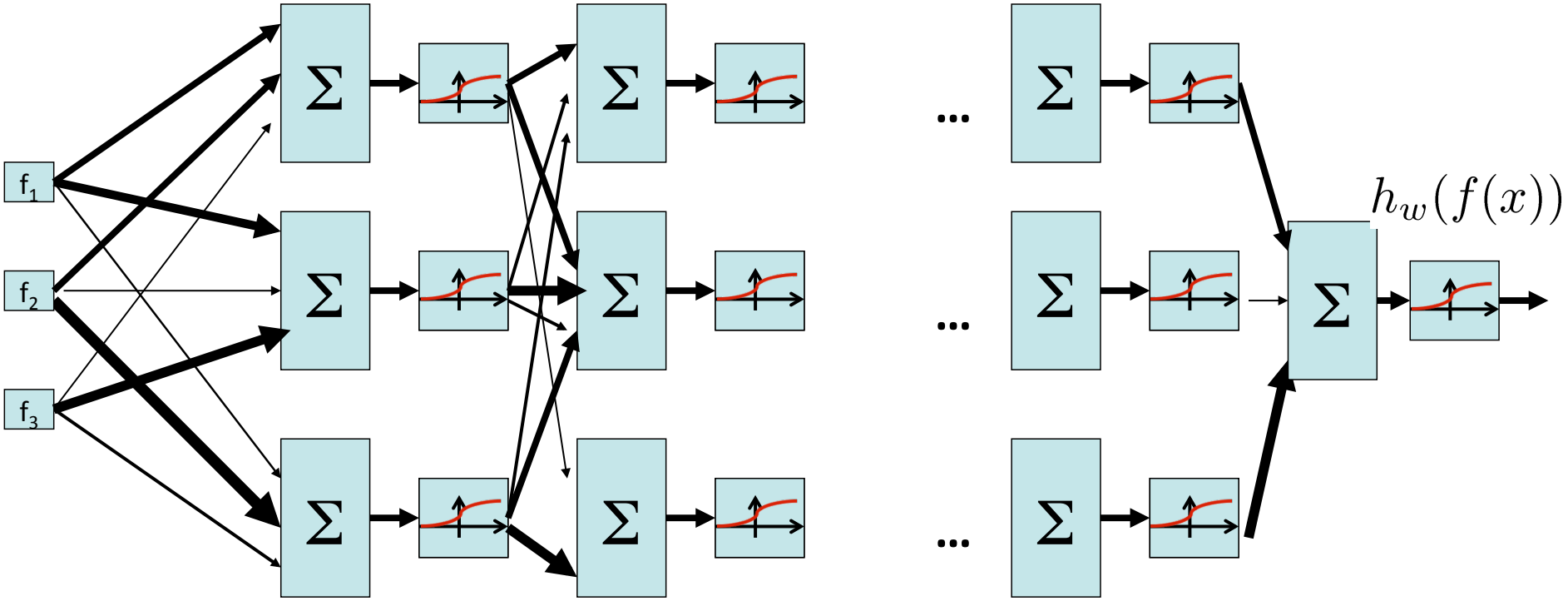
Perceptron



Two-Layer Neural Network

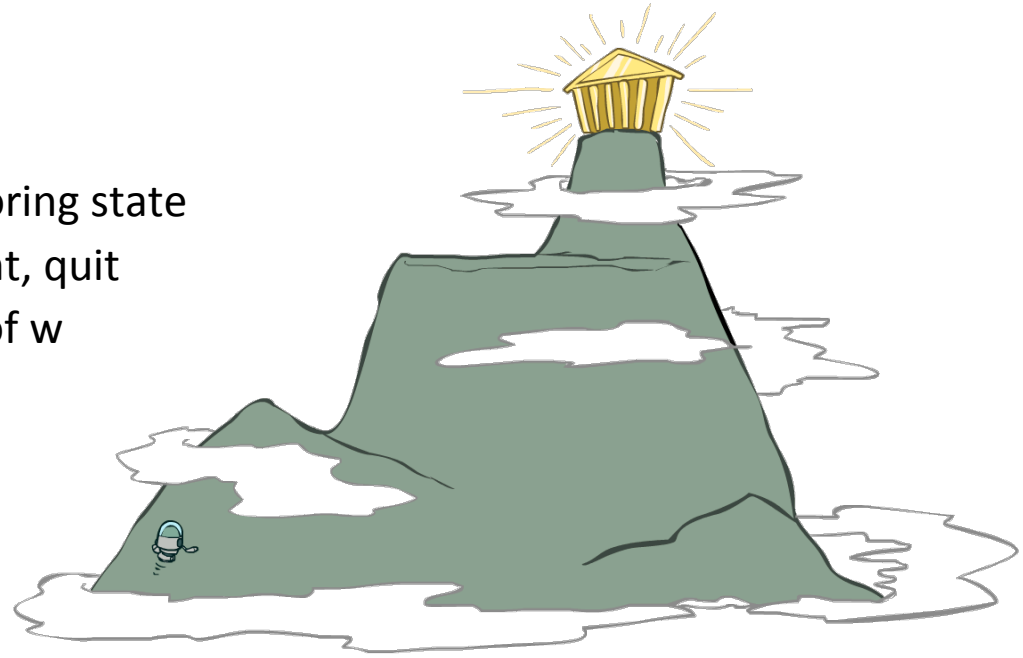


N-Layer Neural Network



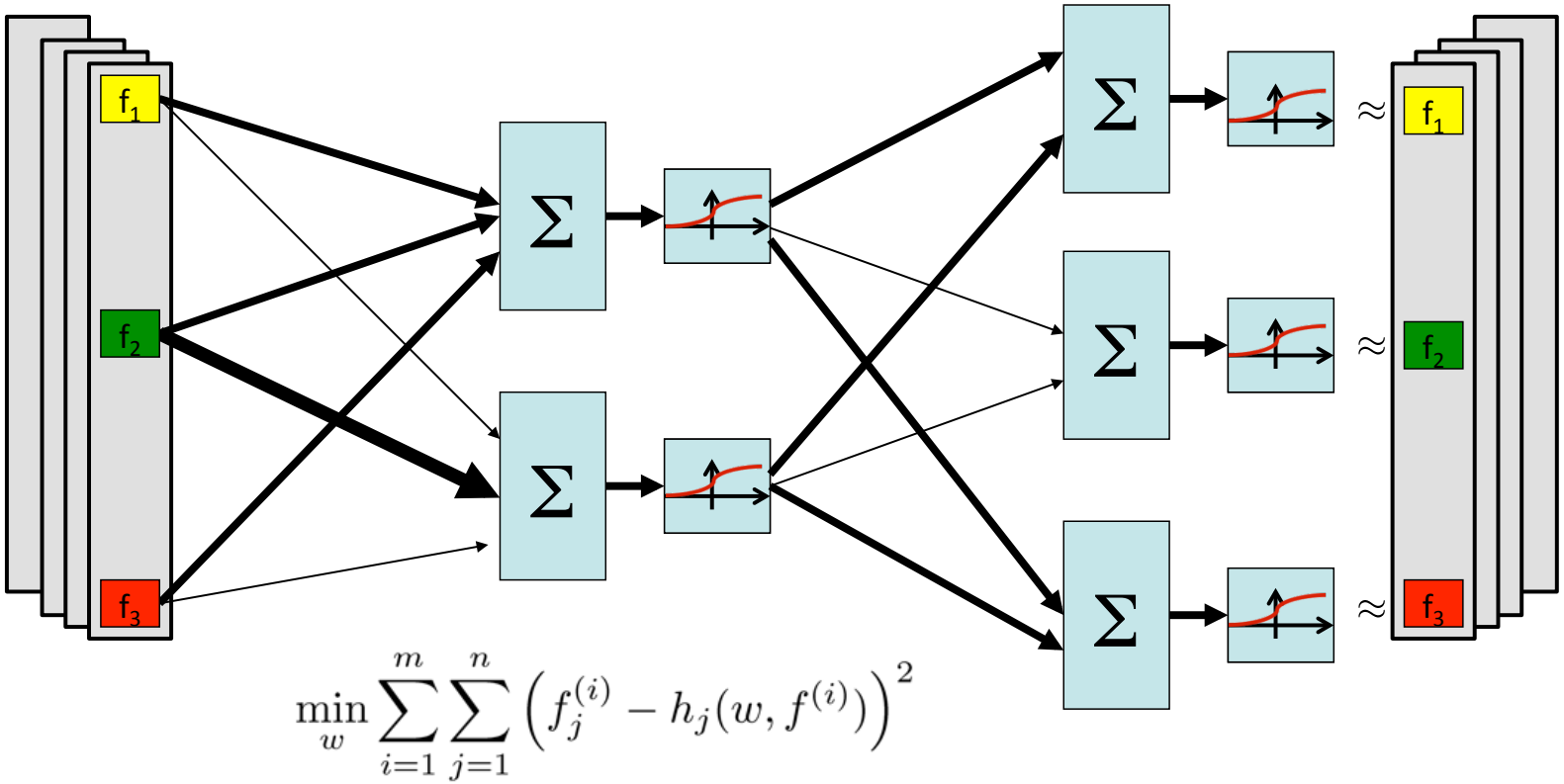
Hill Climbing

- Simple, general idea:
  - Start wherever
  - Repeat: move to the best neighboring state
  - If no neighbors better than current, quit
  - Neighbors = small perturbations of w
- Property
  - Many local optima



--> How to find a good local optimum?

Auto-Encoder (Crude Idea Sketch)

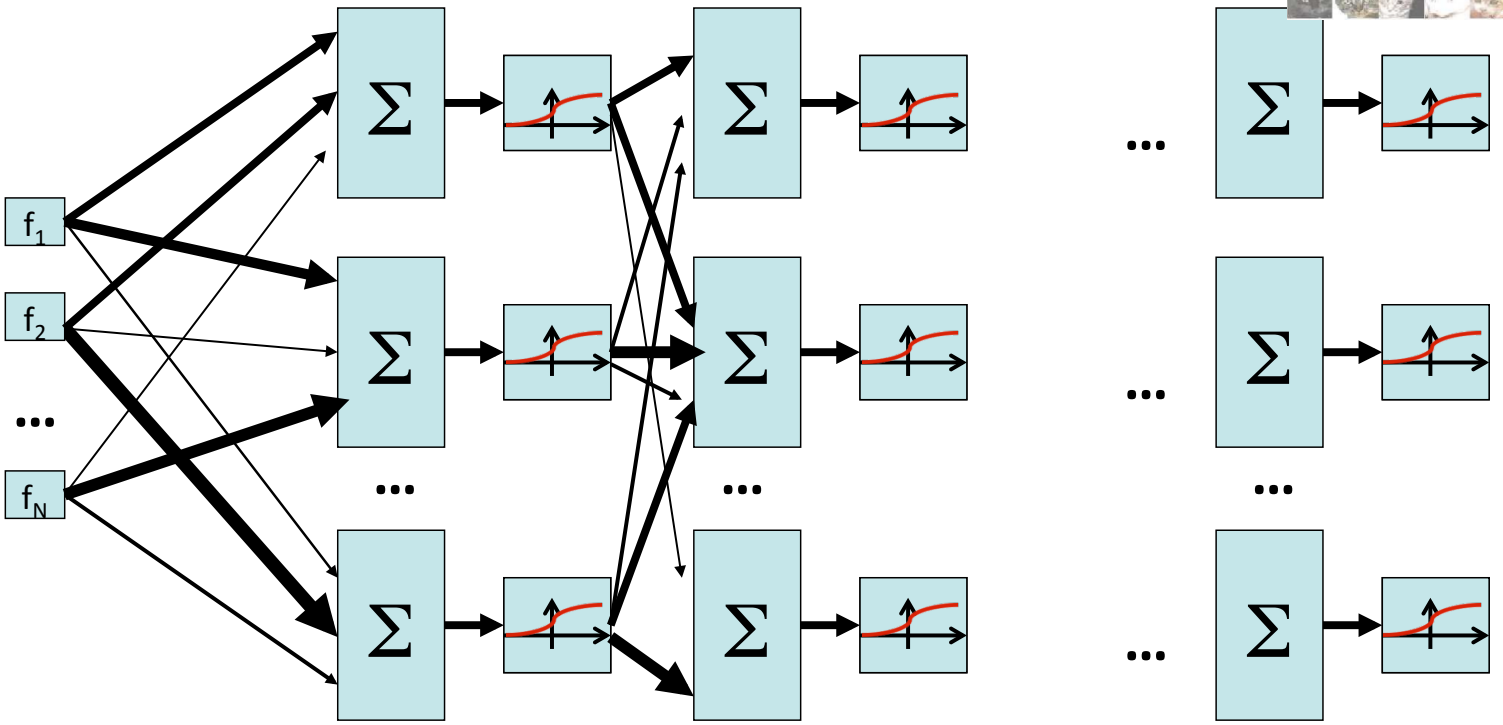


$$\min_{w} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(f_j^{(i)} - h_j(w, f^{(i)}) \right)^2$$

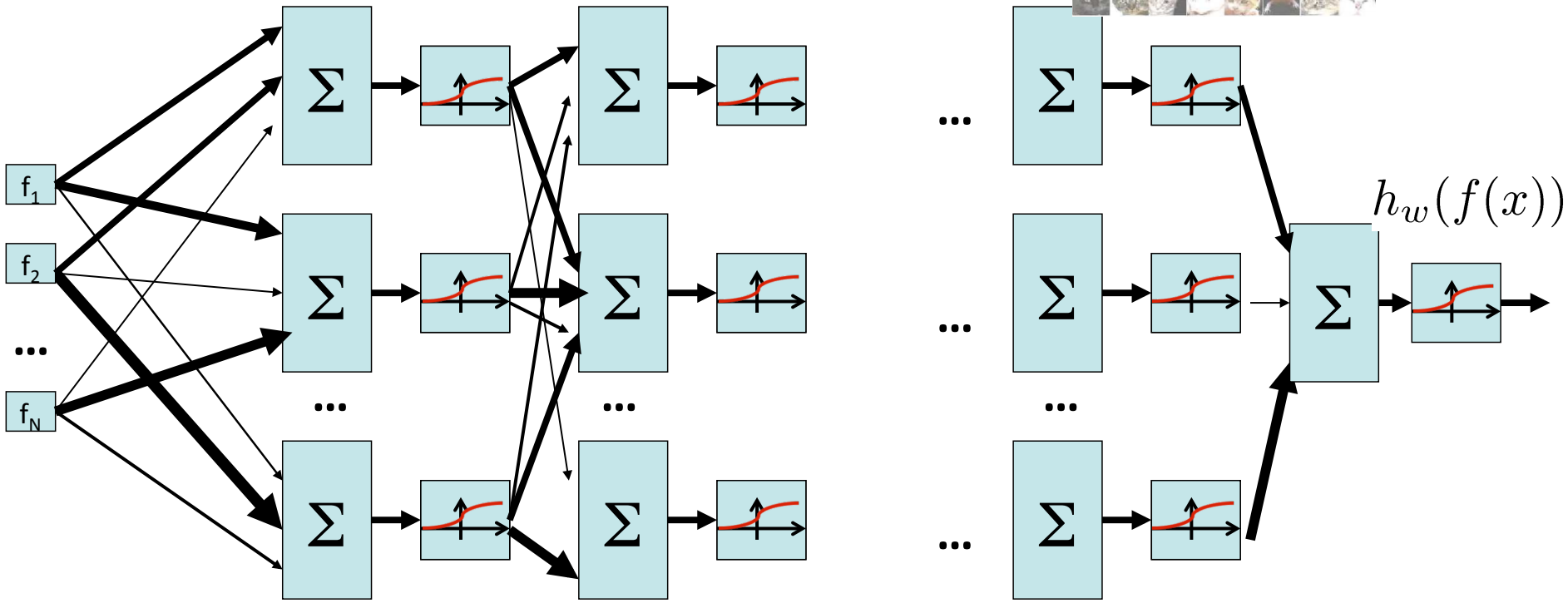
Training Procedure: Stacked Auto-Encoder

- Auto-encoder
  - Layer 1 = “compressed” version of input layer
- Stacked Auto-encoder
  - For every image, make a compressed image (= layer 1 response to image)
  - Learn Layer 2 by using compressed images as input, and as output to be predicted
  - Repeat similarly for Layer 3, 4, etc.
- Some details left out
  - Typically in between layers responses get agglomerated from several neurons (“pooling” / “complex cells”)

Final Result: Trained Neural Network

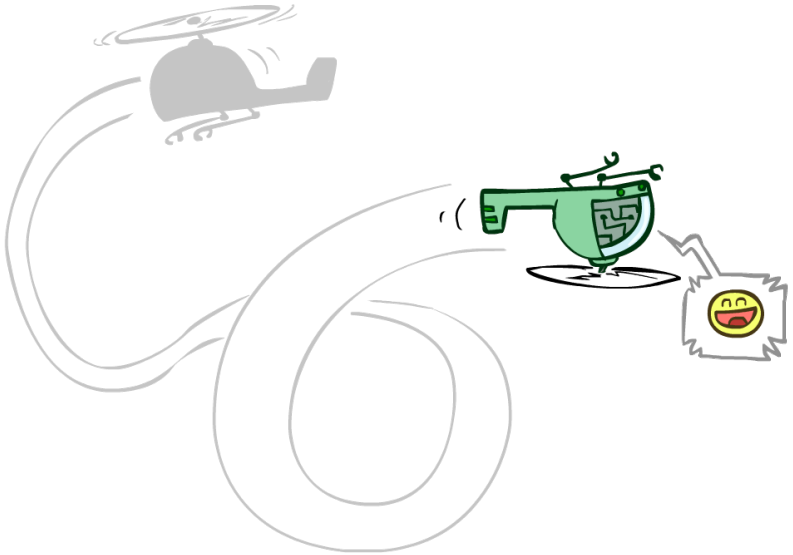


Final Result: Trained Neural Network



Robotics

Robotic Helicopters



Motivating Example

- How do we execute a task like this?

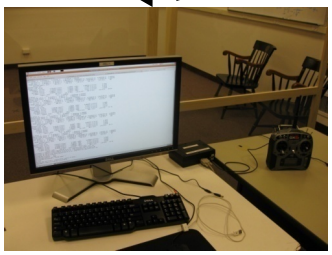
Autonomous Helicopter Flight

- Key challenges:
  - Track helicopter position and orientation during flight
  - Decide on control inputs to send to helicopter

Autonomous Helicopter Setup

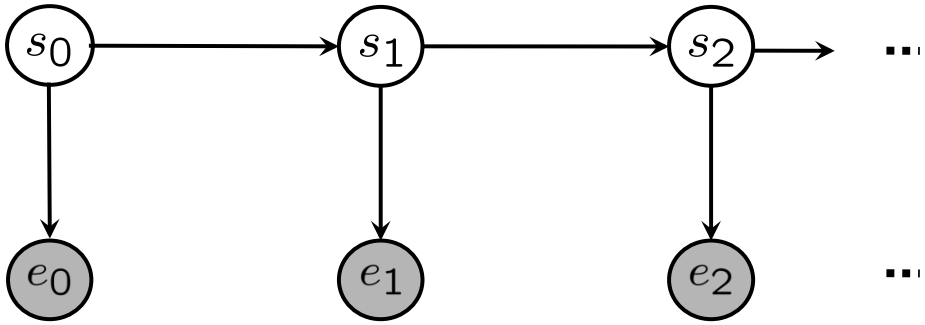
Position

On-board inertial measurement unit (IMU)



Send out controls to helicopter

HMM for Tracking the Helicopter



- State: $s = (x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})$
- Measurements: [observation update]
  - 3-D coordinates from vision, 3-axis magnetometer, 3-axis gyro, 3-axis accelerometer
- Transitions (dynamics): [time elapse update]
  - $s_{t+1} = f(s_t, a_t) + w_t$ f: encodes helicopter dynamics, w: noise

Helicopter MDP

- State: $s = (x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})$

- Actions (control inputs):
  - a_{lon} : Main rotor longitudinal cyclic pitch control (affects pitch rate)
  - a_{lat} : Main rotor latitudinal cyclic pitch control (affects roll rate)
  - a_{coll} : Main rotor collective pitch (affects main rotor thrust)
  - a_{rud} : Tail rotor collective pitch (affects tail rotor thrust)

- Transitions (dynamics):
  - $s_{t+1} = f(s_t, a_t) + w_t$

  [f encodes helicopter dynamics]

  [w is a probabilistic noise model]

- Can we solve the MDP yet?

Problem: What's the Reward?

- Reward for hovering:

$$
\begin{aligned}
R(s) \quad = \quad & -\alpha_x (x - x^*)^2 \\
& -\alpha_y (y - y^*)^2 \\
& -\alpha_z (z - z^*)^2 \\
& -\alpha_{\dot{x}} \dot{x}^2 \\
& -\alpha_{\dot{y}} \dot{y}^2 \\
& -\alpha_{\dot{z}} \dot{z}^2
\end{aligned}
$$

Hover

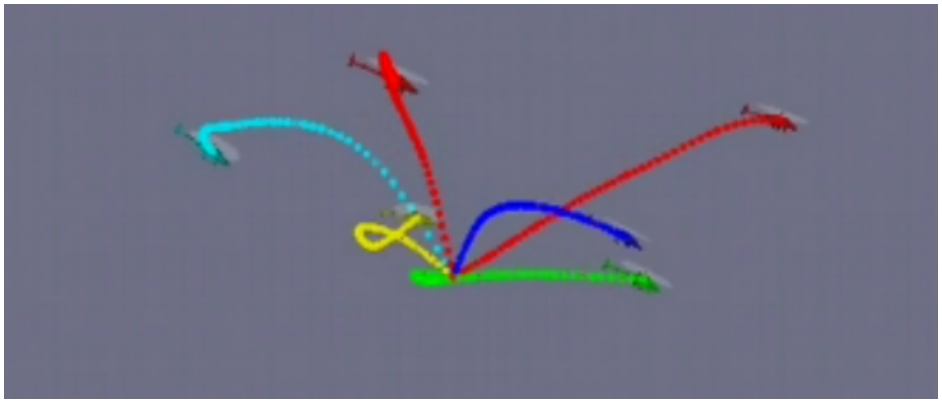
[Ng et al, 2004]

Problem: What's the Reward?

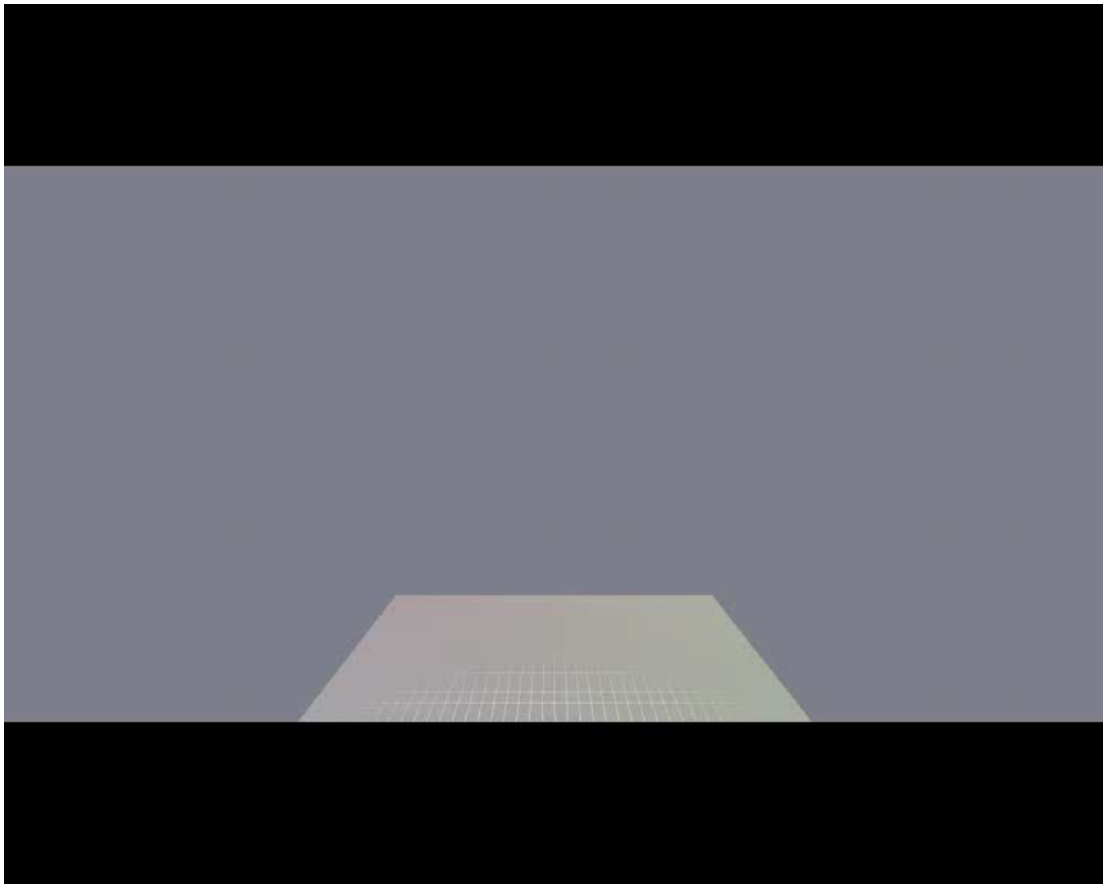
- Rewards for "Flip"?
  - Problem: what's the target trajectory?
  - Just write it down by hand?

Flips (?)

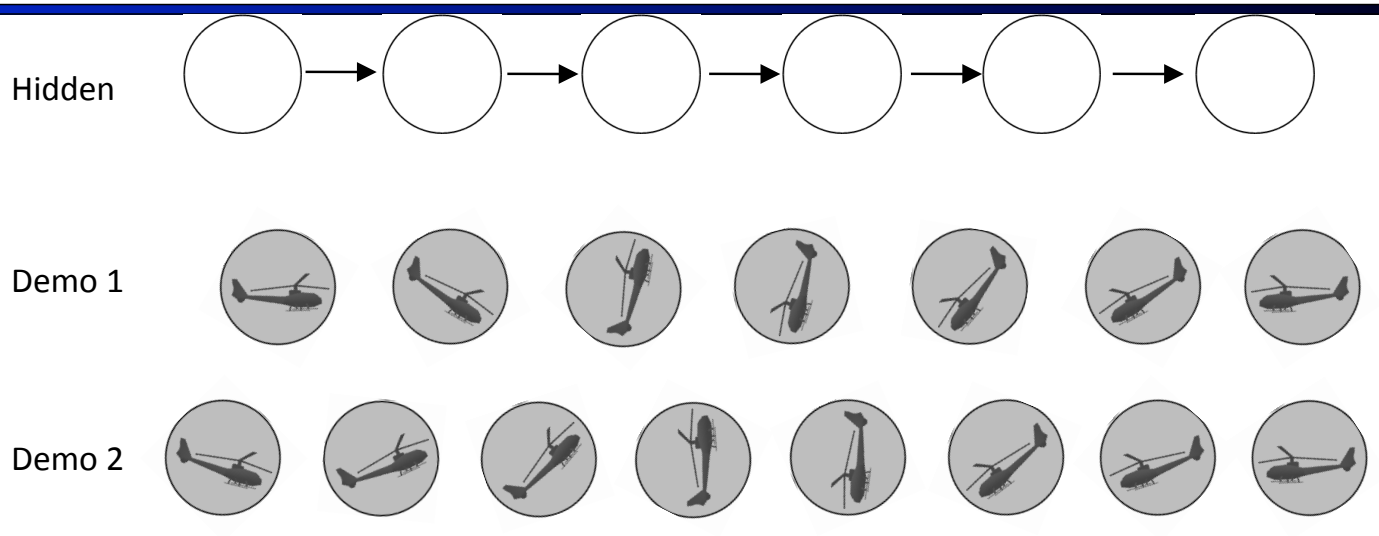
Helicopter Apprenticeship?



Demonstrations



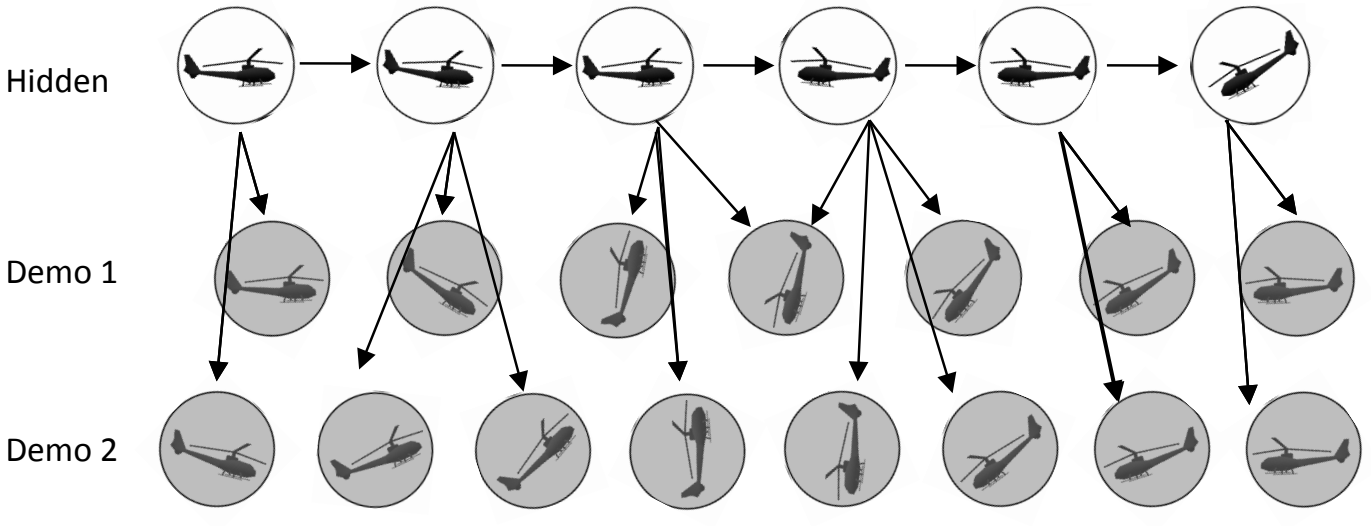
Learning a Trajectory



- HMM-like generative model
  - Dynamics model used as HMM transition model
  - Demos are observations of hidden trajectory
- Problem: how do we align observations to hidden trajectory?

Abbeel, Coates, Ng, IJRR 2010

Probabilistic Alignment using a Bayes' Net



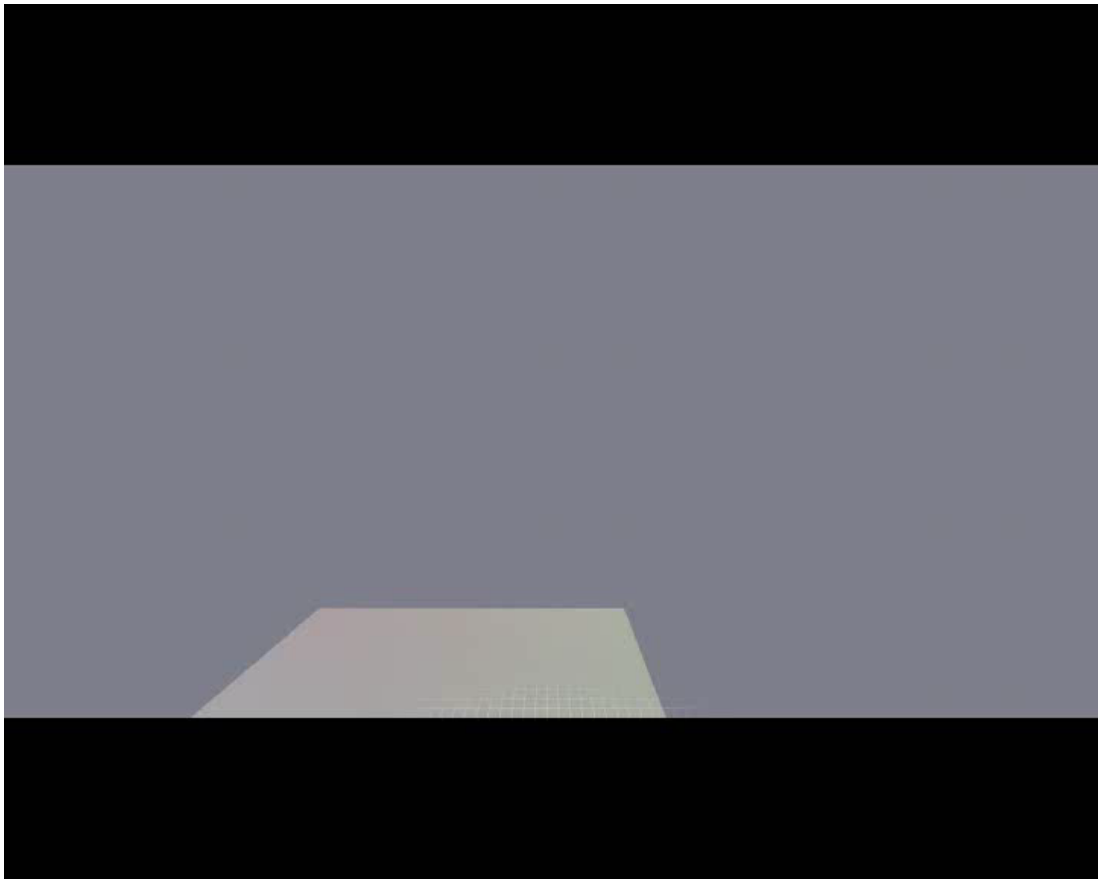
- Dynamic Time Warping

  (Needleman&Wunsch 1970, Sakoe&Chiba, 1978)

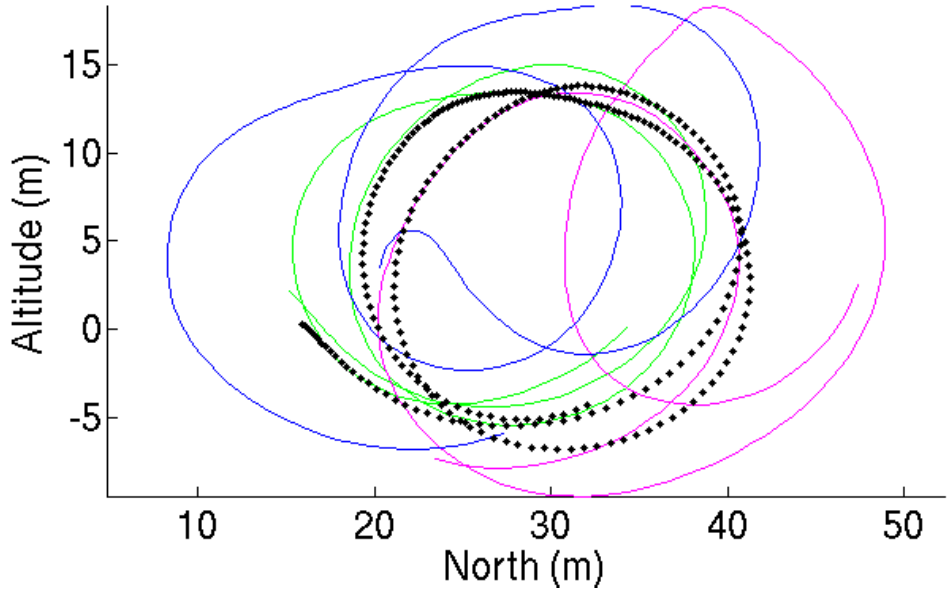
- Extended Kalman filter / smoother

Abbeel, Coates, Ng, IJRR 2010

Aligned Demonstrations



Alignment of Samples

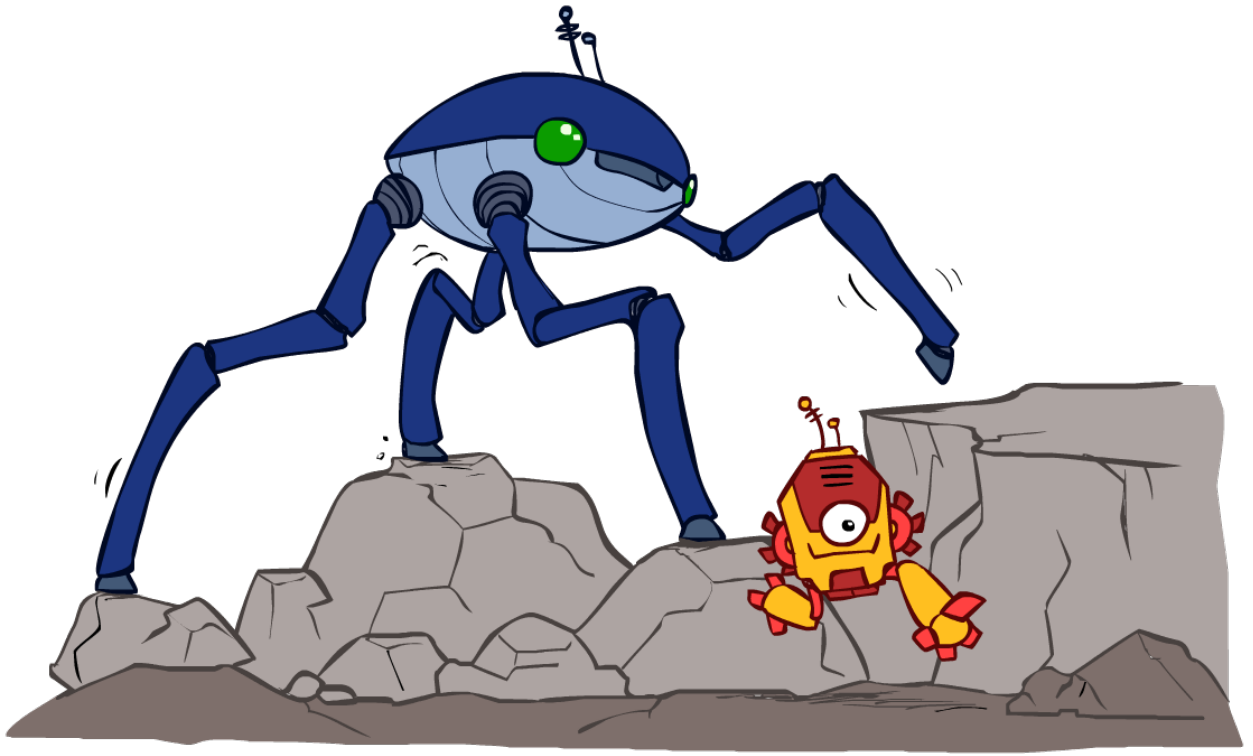


- Result: inferred sequence is much cleaner!

Final Behavior

[Abbeel, Coates, Quigley, Ng, 2010]

Legged Locomotion



Quadruped

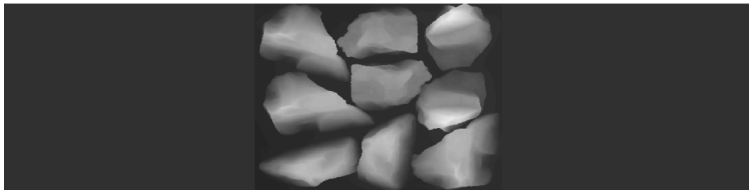
- Low-level control problem: moving a foot into a new location
  → search with successor function ~ moving the motors
- High-level control problem: where should we place the feet?
  - Reward function R(x) = w . f(s) [25 features]

[Kolter, Abbeel & Ng, 2008]

Experimental setup

- Demonstrate path across the "training terrain"

- Run apprenticeship to learn the reward function
- Receive "testing terrain"---height map.



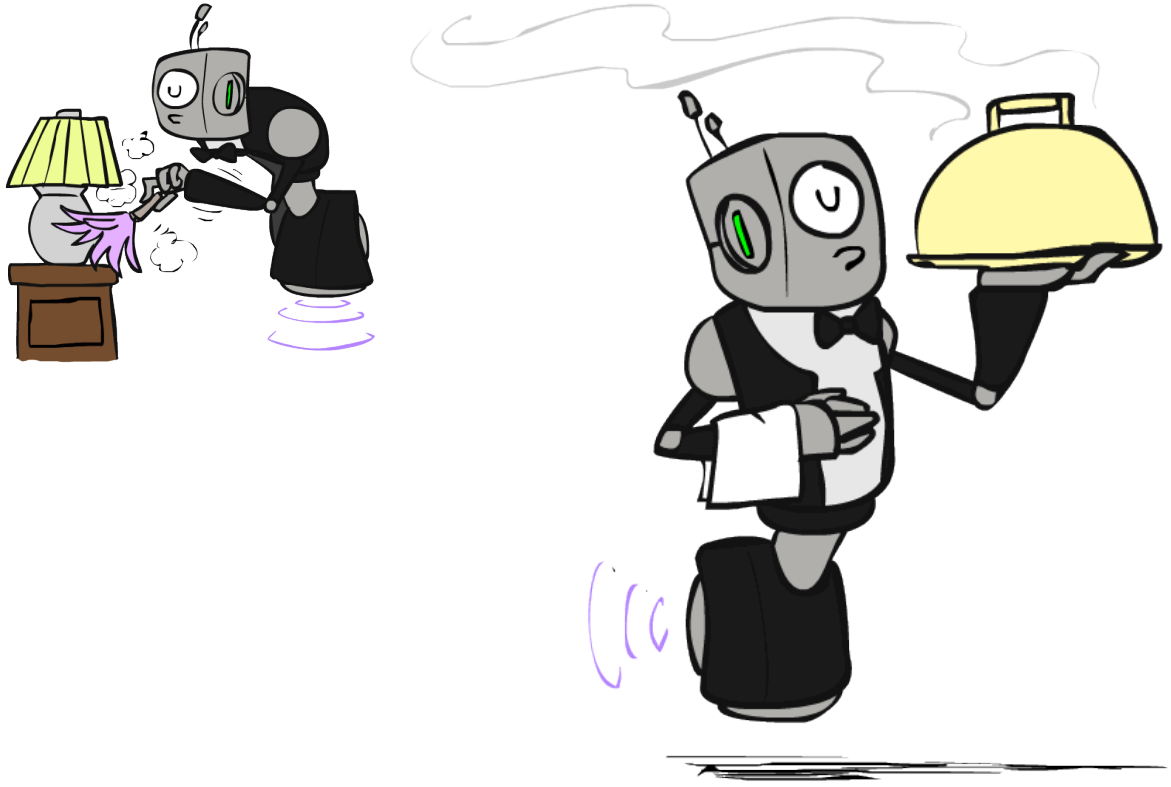
- Find the optimal policy with respect to the *learned reward function* for crossing the testing terrain.

[Kolter, Abbeel & Ng, 2008]

Without learning

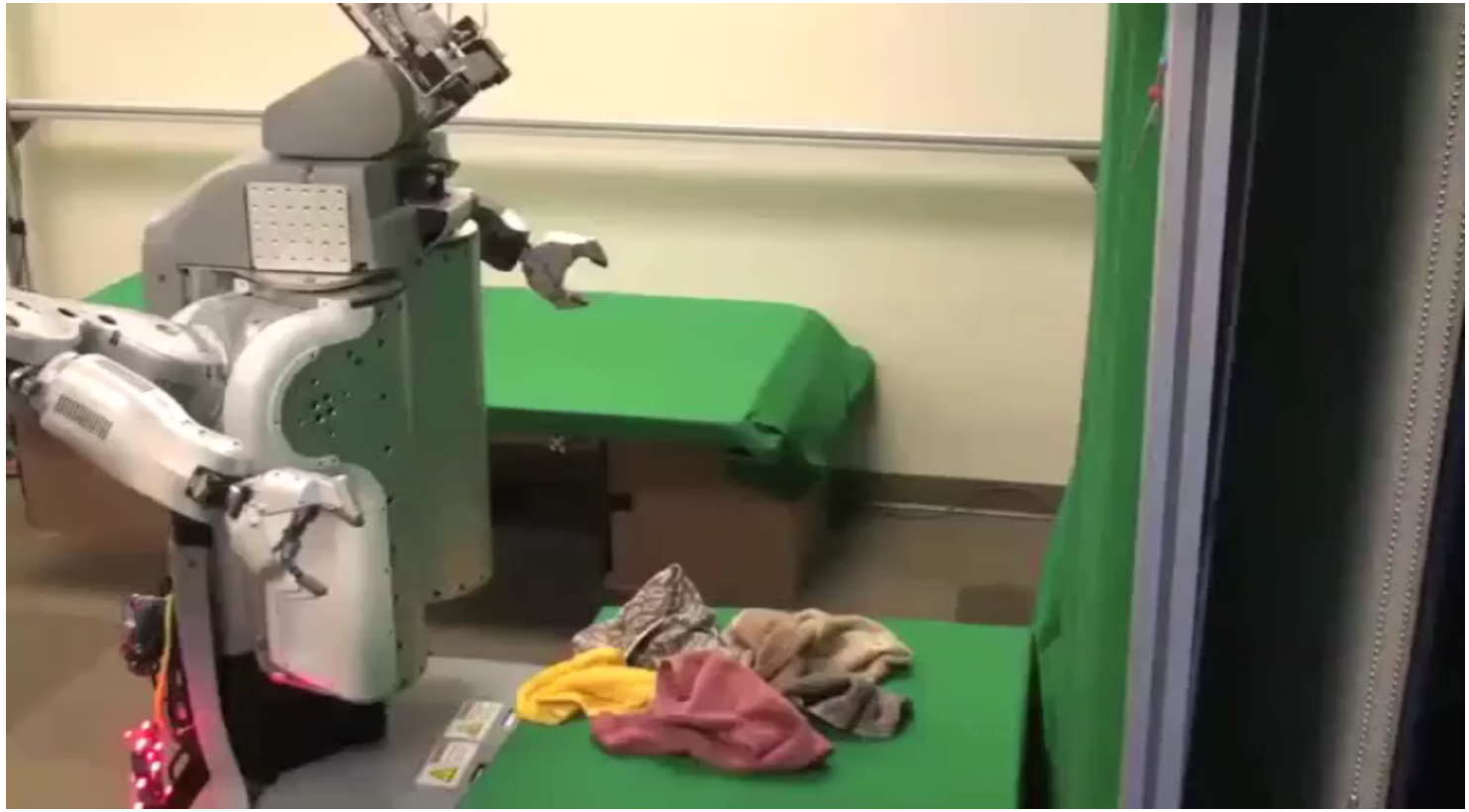
With learned reward function

Personal Robotics

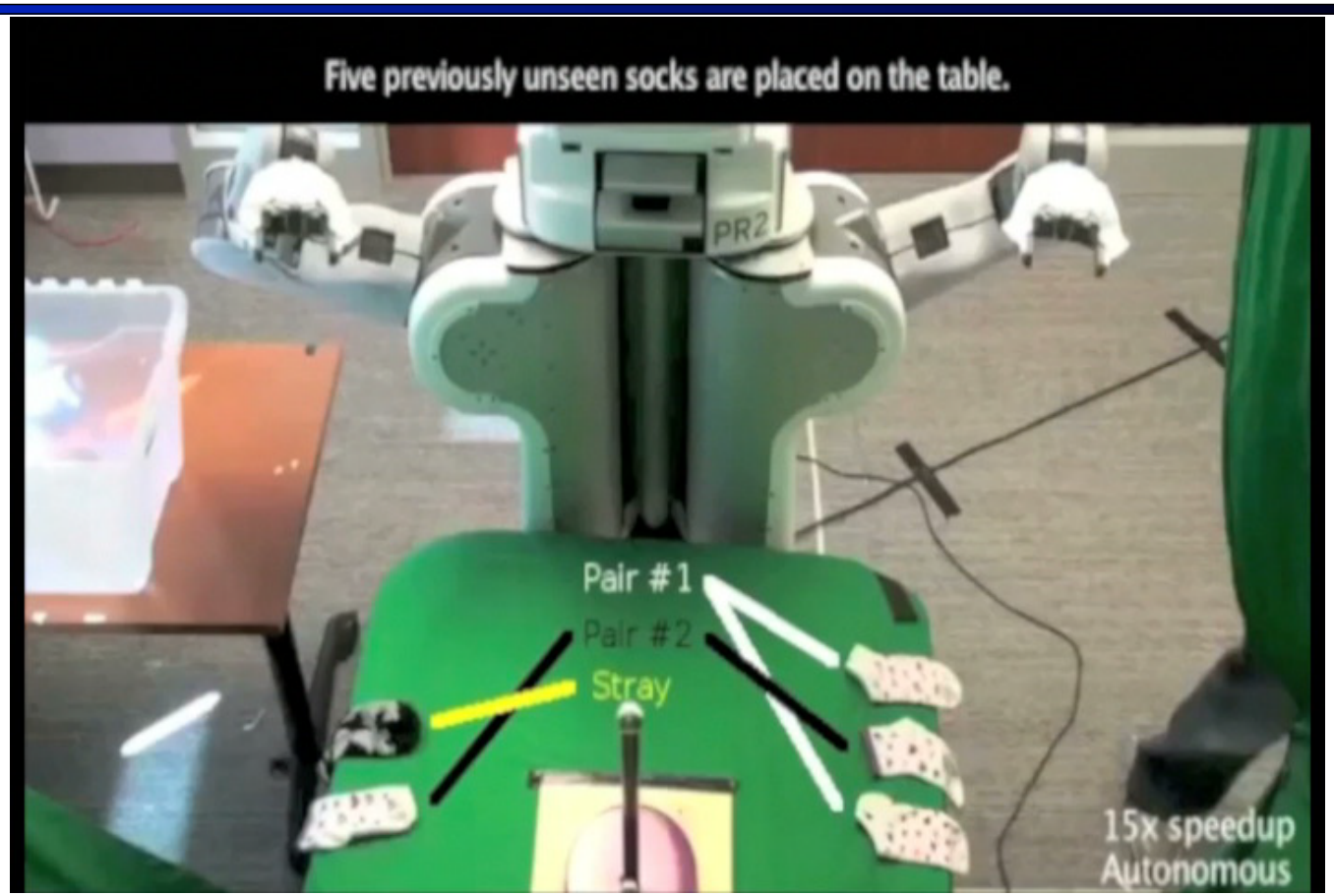


PR1 (tele-op)

PR2 (autonomous)



PR2 (autonomous)



Darpa Robotics Challenge

- Disaster response (e.g., Fukushima)
  - E.g., Get into car, drive it, get out, open door, enter building, climb ladder, traverse industrial walkway, use tool to break a panel, locate and close a valve, replace a cooling pump
- Competition / Prizes
  - Simulation competition (June 2013)
    - Prize: Petman
  - Real robot (petman) competition (November 2014)
    - Prize: $ 2M

Next Time

- AI for games
- Final Contest results
- Where to go next to learn more about AI