Vision for embodied agents

Bill Freeman, Antonio Torralba, Phillip Isola
6.819 / 6.869
Announcements

- Pset 7 due this Thursday
- Be working on final project!
- Presentations 12/6 - 12/11 — Sign up online
- Writeups due 12/12
Intro to reinforcement learning (RL)

Code

Curated list of papers

Advice on how to do research in RL

[Another good blog: http://karpathy.github.io/2016/05/31/rl/]
Agent observation raw pixels

Indoor map overview

Silver et al., 2016

Jaderberg et al. 2018
The whole purpose of visual perception, in humans, is to make good motor decisions.

We are **sensorimotor** systems.
Intelligent agents

Agent

Observations

Environment

Actions
Intelligent agents

**Policy**

\[ \pi : s_t \rightarrow a_t \]

**Environment**

\[ f : s_t, a_t \rightarrow s_{t+1} \]
Recipe for deep learning in a new domain

1. Transform your data into numbers (e.g., a vector)

2. Transform your goal into an equation (objective function)

3. #1 and #2 specify the “learning problem”

4. Use a generic optimizer (SGD) and an appropriate architecture (e.g., CNN or RNN) to solve the learning problem
How to represent a state? How to represent policy?

state: pixels!

policy: action classifier
Learning from examples
(aka *supervised learning*)

Training data

\[
\begin{align*}
\{x_1, y_1\} & \\
\{x_2, y_2\} & \\
\{x_3, y_3\} & \\
\ldots
\end{align*}
\]

\[
f^* = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(x_i), y_i)
\]

\[
f : X \to Y
\]
Imitation learning

(still just **supervised learning**, applied to learn **policies**)
Imitation learning

\[ S \] \to \begin{cases} \text{d3 \texttt{b} b4+} \\ \text{f3 \texttt{c} c6} \\ \text{0-0 \texttt{x} xc3} \end{cases} \to \begin{cases} \text{a} \end{cases} \]

**Learner**

**Objective**

\[ \pi(s) = \text{softmax}(g_\theta(s)) \]

\[ \mathcal{L}(a, \pi(s)) = H(a, \pi(s)) \]

**Hypothesis space**

Convolutional neural net

**Optimizer**

Stochastic gradient descent
Learning without examples

(includes unsupervised learning and reinforcement learning)

Data

\[
\begin{align*}
\{x_1\} \\
\{x_2\} & \rightarrow \text{Learner} \\
\{x_3\} \\
\ldots
\end{align*}
\]
Unsupervised Representation Learning

\[ \{x_1\} \rightarrow \text{Learner} \rightarrow \text{Representations} \]
Reinforcement learning

What’s a good policy? (what’s the learning objective?)
Reinforcement learning

Policy

Observations

Rewards

Environment

Actions

Learn a policy that takes actions that maximize reward
**Imitation learning**

- Hand-curated training data
  + Instructive examples
  + Follows a curriculum
- Expensive
- Limited to teacher’s knowledge

**Reinforcement learning**

- No training data, have to play around and collect the data *yourself*
  + No need for labeled data
  + Can learn things no human knows how to do
- Less instructive
- No curriculum
- Have to explore
Reinforcement learning

Policy

\[ \pi : s_t \rightarrow a_t \]

Environment

\[ f : s_t, a_t \rightarrow s_{t+1} \]

State, Reward

\[ s_{t+1}, r_t \]

Actions

\[ a_t \]
Reinforcement learning

Learn a policy that takes actions that maximize reward

\[ \pi^* = \arg \max_{\pi} R(\tau) \]
This is called a **Markov decision process (MDP)**.
Reinforcement learning

Data

\[ \mathcal{T} \rightarrow \]

Learner

Objective

\[ R(\tau) \]

Hypothesis space
These days: deep net

Optimizer

\[ \mathcal{P} \rightarrow \]

Environment

Can't, in general, backprop through env!
Environment is not differentiable! — How to optimize?

Idea #1 (trial and error):

**Policy gradients**: Run a policy for a while. See what actions led to high rewards. Increase their probability.

[Adapted from Andrej Karpathy: http://karpathy.github.io/2016/05/31/rl/]
**Policy gradients**: Run a policy for a while. See what actions led to high rewards. Increase their probability.

[Adapted from Andrej Karpathy: http://karpathy.github.io/2016/05/31/rl/]

Supervised Learning
(correct label is provided)

Reinforcement Learning

[Adapted from Andrej Karpathy: http://karpathy.github.io/2016/05/31/rl/]
Environment is not differentiable! — How to optimize?

Policy gradients

1. Start with an arbitrary initial policy

2. **Rollout** this stochastic policy a bunch of times, sampling different random actions each time

3. Update your policy to place higher probability on actions that led to higher returns

Mathematically, this approximates gradient ascent on policy parameters, so as to maximize reward.
Intelligent agents

Agent

Observations

Environment

Actions

Why vision?
Why vision?

1. Human-like intelligence (and animal-like), relies heavily on vision

>30% of the human cortex?
http://www.kyb.tuebingen.mpg.de/research/dep/lo/visual-perception.html

[See Animal Eyes by Michael Land and Dan Nilsson]
Why vision?

2. Eyes are good sensors

Farther away things look smaller

Get details on stuff that we can immediately interact with, rough summary of more distant context
Why vision?

2. Eyes are good sensors

Laser rangefinder
Why vision?

2. Eyes are good sensors

Laser rangefinder
2. Eyes are good sensors
Why vision?

2. Universal interface
Why vision?

4. The brain’s *model building* system

[Kanazawa, Tulsiani, et al., ECCV 2018]
Model-based intelligence

If vision can give us a good representation/model of the world, then planning and control should be easy.

Yann LeCun’s cake
ATARI Games

~10-50 million interactions!

AlphaGo

21 million games!

[Slide adapted from Pulkit Agrawal]