Lecture 26: Robot Learning Overview and Deep Learning Frontiers

Danfei Xu
Administrative

Remember to fill CIOS evaluation!

Poster session Dec 5th 5pm-6:30pm
• Bring your poster. We will provide easels.
• You will be given an easel number the day of the event.
• The TAs will start by grading half of the posters in the first 45 min, and the other half in the second 45 min.
• You will know which batch you are in at the event.
• Check out other posters if your batch is not being graded.
• We will have pizza and dessert available
• We will announce a **best project award** at the end of the poster session.
• The event is open to the GT community. Expect many attendees, so bring your best work. And tell your friends to come too!
Past & present: robots in factories & semi-structured environments
Future: robots everywhere!
How we program these robots today …
Manual programming is not enough!

diverse tasks

messy environments
The Moravec's paradox

Moravec's paradox is the observation ... contrary to traditional assumptions, reasoning requires very little computation, but sensorimotor and perception skills require enormous computational resources. (Wikipedia)

Marvin Minsky: "In general, we're least aware of what our minds do best" ... "we're more aware of simple processes that don't work well than of complex ones that work flawlessly".
Can we teach robots through data / examples?

state input → Control Policy → action output
Can we teach robots through data / examples?

Very useful, but expensive to acquire in the physical world

Data source: experience from trial-and-error
Can we teach robots through data / examples?

Many possible data sources & formats!
Only if we can have flexible ML methods that can learn from them all ...
Deep Learning for Robotics

Deep Neural Networks

state input

Deep Learning is NOT all you need!

The ALVINN project at CMU (Pomerleau 1988)

“Robot Transformer (RT1)” from Google Robotics (2023)
Deep Learning is NOT all your need

Robots today have some deep learning components, but nothing is fully "end-to-end".

The “control policy” of a learning robot for e-commerce fulfillment.
Covariant AI (video source)
Deep Learning is NOT all your need

Robots today have some deep learning components, but nothing is fully "end-to-end".

The perception pipeline of an autonomous driving stack

NVIDIA (image source)
Robot Learning

Robot learning is a research field at the intersection of machine learning and robotics. It studies techniques allowing a robot to acquire novel skills or adapt to its environment through learning algorithms. (Wikipedia)

More concise version:

Principles, algorithms, and systems that allow robots to improve by learning from data.

Robot Learning research today (2023): what and how to learn.
Robot Learning: ML don’t need to (and shouldn’t) be applied to everything!

The reason that we want to use machine learning is to deal with variation, noise, and things that are hard to model.

Unlike computer vision and natural language understanding, robotics often deal with physics, which we know well. So we don’t need to learn everything!

Both a challenge and an opportunity for robot learning: how to best combine what we know and what we need to learn.
State of Robot Learning Research

**Mastery**: be able to solve tasks that are hard / infeasible to solve by manual programming.

**Scaling**: apply a method / framework to a broad range of tasks by scaling up data sources.

**Generalization**: solve new tasks in new environments and scenarios; show emerging behaviors that are not in the training data.
State of Robot Learning Research

**Mastery**: be able to solve tasks that are hard / infeasible to solve by manual programming (*successes in some domains*).

**Scaling**: apply a method / framework to a broad range of tasks by scaling up data sources (*ongoing progress*).

**Generalization**: solve new tasks in new environments and scenarios; show emerging behaviors that are not in the training data (*holy grail, no real progress yet*).
Examples of mastering hard tasks

Sim-to-real Reinforcement Learning

Source: OpenAI

Source: ETH Zurich
Examples of scaling up data sources

RT1: Imitation learning from 130k demonstrations collected over the course of 17 months

https://robotics-transformer.github.io/assets/rt1.pdf
No where near generalizable decision making!

https://www.ted.com/talks/marc_raibert_meet_spot_the_robot_dog_that_can_run_hop_and_open_doors?language=en
It’s a great time to work on robot learning!

general-purpose learning algorithms

general-purpose robot hardware

more compute
# Deep Learning for Robotics (CS 8803-DLM): an overview

<table>
<thead>
<tr>
<th>2D/3D Perception and Grasping</th>
<th>Act without Models: Reinforcement Learning and Imitation Learning</th>
<th>Model-based Decision Making: Learning for Planning and Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning to grasp: DexNet family</td>
<td>Model-free RL: TRPO, SAC, DDPG</td>
<td>Model-based RL</td>
</tr>
<tr>
<td>Learning to grasp: visual affordances and action-as-perception</td>
<td>Offline Reinforcement Learning</td>
<td>Learning Planning Representations</td>
</tr>
<tr>
<td>VLM for Manipulation</td>
<td>Imitation Learning: Behavior Cloning, Learning from human data</td>
<td>Learning Control Representations</td>
</tr>
<tr>
<td>Tactile Sensing</td>
<td>Imitation Learning: Inverse RL, Generative Adversarial Imitation, Sim-to-real transfer</td>
<td>Task and Motion Planning</td>
</tr>
<tr>
<td>Multimodal Representation Learning</td>
<td>Curiosity and Exploration</td>
<td>Learning for Task and Motion Planning</td>
</tr>
<tr>
<td></td>
<td>Human-in-the-loop Robot Learning</td>
<td>Language Model for Robotics</td>
</tr>
</tbody>
</table>
Frontiers of Deep Learning

Topics we didn’t get time to cover:

• Vision Transformers
• Graph Neural Nets
• Metric learning
• AutoML
• 3D perception & reconstruction
• Memory modeling
• Few-shot / meta learning
• Neural Radiance Field (NeRF) / implicit representations
• Adversarial learning and robustness
• Continual / lifelong learning
• Visual reasoning
• Neural Theorem Proving
• Neural Program Induction / Synthesis
• MLSys
• Many topics in NLP …
Topics we didn’t get time to cover:

- Vision Transformers
- Graph Neural Nets
- Metric learning
- AutoML
- **3D perception & reconstruction**
- Memory modeling
- Few-shot / meta learning
- **Neural Radiance Field (NeRF) / implicit representations**
- Adversarial learning and robustness
- Continual / lifelong learning
- Visual reasoning
- Neural Theorem Proving
- Neural Program Induction / Synthesis
- MLSys
- Many topics in NLP …
3D Perception

3D Object Detection / Pose Estimation

Input Image → 2D Recognition

height
roll
pitch
width

length

3D Object Reconstruction

Input Image

2D Recognition

→ sofa

chair

3D Meshes

3D Voxels

3D Object Reconstruction
3D Perception

Many possible ways to represent the 3D world ...

- Depth Map
- Voxel Grid
- Pointcloud
- Mesh
- Implicit Surface

Each representation requires different neural network architectures!

Figure credit: Justin Johnson
3D Perception

3D Convolution for Voxel-based 3D Reconstruction

3D Perception

(Simplified) PointNet architecture for 3D point cloud classification
Neural Radiance Field
Neural Radiance Field: View Synthesis
Volume Rendering


https://coronarenderer.freshdesk.com/support/solutions/articles/12000045276-how-to-use-the-corona-volume-grid-
Volume Rendering: Ray Marching
Volume Rendering: Ray Marching

\((r, g, b)\)
Volume Rendering: Ray Marching

$$(r, g, b, \delta, \sigma)$$

$$(R, G, B)$$
Volume Rendering: Ray Marching

\[(r, g, b, \delta, \sigma)\]

\[(R, G, B)\]
Volume Rendering: Ray Marching

\((r, g, b, \delta, \sigma)\)

\((R, G, B)\)

Compositing
Volume Rendering: Ray Marching

\[(x, y, z, \theta, \phi) \xrightarrow{F_Q} (r, g, b, \delta, \sigma)\]
Neural Radiance Field

5D Input Position + Direction

\[(x, y, z, \theta, \phi)\] → \(F_\theta\) → \((RGB\sigma)\)

Output Color + Density

Very slow to train & render!
Requires many tricks to render high-quality images
One model per scene
Instant NeRF

Müller et al., 2022
Frontiers of Deep Learning

Topics we didn’t get time to cover:

• Vision Transformers
• **Graph Neural Nets**
• Metric learning
• AutoML
• 3D perception & reconstruction
• Memory modeling
• Few-shot / meta learning
• **Neural Radiance Field (NeRF) / implicit representations**
• Adversarial learning and robustness
• Continual / lifelong learning
• Visual reasoning
• Neural Theorem Proving
• Neural Program Induction / Synthesis
• MLSys
• Many topics in NLP …
Homogenization of Deep Learning

Homogenization is the **consolidation** of methodologies for building machine learning systems across a wide range of applications.

**Example:** The Transformer Models (Vaswani et al., 2017)

Transformer Models originally designed for NLP

Almost identical model (Visual Transformers) can be applied to Computer Vision tasks
Lack of interpretability

Why did the robot do that?

I’m turning left here because ...

I will turn left next.
What have we learned this semester?

<table>
<thead>
<tr>
<th>Deep Learning Fundamentals</th>
<th>Neural Network Components and Architectures</th>
<th>Applications &amp; Learning Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear classification &amp; kNNs</td>
<td>Hardware &amp; software</td>
<td>Object Detection</td>
</tr>
<tr>
<td>Loss functions</td>
<td>Convolutions</td>
<td>Semantic &amp; instance</td>
</tr>
<tr>
<td>Optimization</td>
<td>Convolution Neural Networks</td>
<td>Segmentation</td>
</tr>
<tr>
<td>Optimizers</td>
<td>Pooling</td>
<td>Reinforcement Learning</td>
</tr>
<tr>
<td>Backpropagation</td>
<td>Activation functions</td>
<td>Large-language Models</td>
</tr>
<tr>
<td>Computation Graph</td>
<td>Batch normalization</td>
<td>Variational Autoencoders</td>
</tr>
<tr>
<td>Multi-layer</td>
<td>Transfer learning</td>
<td>Diffusion Models</td>
</tr>
<tr>
<td>Perceptrons</td>
<td>Data augmentation</td>
<td>Generative Adversarial Nets</td>
</tr>
<tr>
<td></td>
<td>Architecture design</td>
<td>Self-supervised Learning</td>
</tr>
<tr>
<td></td>
<td>RNN/LSTMs</td>
<td>Vision-Language Models</td>
</tr>
<tr>
<td></td>
<td>Attention &amp; Transformers</td>
<td>VLM for Robotics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Graph Neural Networks</td>
</tr>
</tbody>
</table>
Thank you!