CS 444: Deep Learning for Computer Vision

D. Hockney, Pool with two figures, 1972

https://slazebni.cs.illinois.edu/spring24/
Overview

• Logistics
• Motivation: The statistical learning viewpoint
• A taxonomy of learning problems
• Topics to be covered in class
How can we build an agent to...

- Play chess?
- Recognize object categories?
- Translate between languages?
- Fly a drone?
How can we build an agent to achieve expertise?

- Good old-fashioned AI (GOFAI) answer: Program expertise into the agent

How can we build an agent to achieve expertise?

- Good old-fashioned AI (GOFAI) answer: Program expertise into the agent

Figure 4-a. "Building" region and "windows".

Figure 4-b. The production for analyzing "windows".
How can we build an agent to achieve expertise?

- Good old-fashioned AI (GOFAI) answer:
  Program expertise into the agent
How can we build an agent to achieve expertise?

• Good old-fashioned AI (GOFAI) answer: Program expertise into the agent
  • Never worked (in general)
How can we build an agent to achieve expertise?

- Good old-fashioned AI (GOFAI) answer: Program expertise into the agent
  - Never worked (in general)
  - Though not without exceptions…

---

**THE PASTRY A.I. THAT LEARNED TO FIGHT CANCER**

In Japan, a system designed to distinguish croissants from bear claws has turned out to be capable of a whole lot more.

By James Somers
March 18, 2021

How can we build an agent to achieve expertise?

- Good old-fashioned AI (GOFAI) answer: Program expertise into the agent
  - Never worked (in general)
  - Though not without exceptions...

**ANNALS OF TECHNOLOGY**

**THE PASTRY A.I. THAT LEARNED TO FIGHT CANCER**

In Japan, a system designed to distinguish croissants from bear claws has turned out to be capable of a whole lot more.

By James Somers
March 18, 2021

https://www.newyorker.com/tech/annals-of-technology/the-pastry-ai-that-learned-to-fight-cancer

not well fed.) But this was all under carefully controlled conditions. In a real bakery, the lighting changes constantly, and BRAIN’s software had to work no matter the season or the time of day. Items would often be placed on the device haphazardly: two pastries that touched looked like one big pastry. A subsystem was developed to handle this scenario. Another subsystem, called “Magnet,” was made to address the opposite problem of a pastry that had been accidentally ripped apart.

be used elsewhere. Today, solving the pastry problem without deep learning would seem impossible; it’s a wonder that, in 2007, when neural networks weren’t a viable option, Kambe even took it on. The system that he and his team managed to build over the following fifteen years must surely be one of the more sophisticated achievements in “classical” computer vision—a fact obscured, perhaps, by its origin in baked goods.
How can we build an agent to achieve expertise?

- Good old-fashioned AI (GOFAI) answer: Program expertise into the agent

- Modern answer: Program into the agent the ability to **improve performance** based on **experience**
  - **Performance** needs to be quantified using some score or metric (loss, reward, etc.)
  - **Experience** comes from *training data or demonstrations*
  - **Improvement** results from the *learning algorithm*
  - Leap of faith: agent that can achieve good performance on training data will *generalize* to never-before-seen inputs
The basic statistical learning framework

Training examples

Training time

Training examples

"apple"
"pear"
"tomato"
"cow"
"dog"
"horse"

Test sample

Test time

Test sample

Inference

Prediction

"apple"
Overview

• Logistics
• Motivation: The statistical learning viewpoint
• A taxonomy of learning problems
# Taxonomy of learning problems

## Type of output
- Classification
- Regression
- Structured prediction
- Dense prediction
- Multi-modal prediction

## Type of supervision
- Fully supervised
- Unsupervised
- Self-supervised or predictive learning

## Training regime
- Batch offline learning
- Online/continual learning
- Active learning
- Reinforcement learning
Type of output: Classification

ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)
Type of output: Regression

Age estimation

<table>
<thead>
<tr>
<th>3</th>
<th>4</th>
<th>7</th>
<th>12</th>
<th>16</th>
<th>21</th>
<th>23</th>
<th>26</th>
<th>29</th>
<th>31</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9</td>
<td>15</td>
<td>16</td>
<td>18</td>
<td>27</td>
<td>30</td>
<td>46</td>
<td>50</td>
<td>52</td>
<td>54</td>
</tr>
</tbody>
</table>

Date prediction

Vittayakorn et al. (2017)

Location prediction

Vo et al. (2017)

Figure source
Type of output: Dense prediction

Semantic segmentation

Image colorization

Zhang et al. (2016)

Depth prediction

Long et al. (2016)

Wang et al. (2017)
Type of output: Structured prediction

Object detection

dog: 0.55

cat: 0.8

Image description

"man in black shirt is playing guitar."
"construction worker in orange safety vest is working on road."
"two young girls are playing with lego toy."

"girl in pink dress is jumping in air."
"black and white dog jumps over bar."
"young girl in pink shirt is swinging on swing."

Karpathy & Fei-Fei (2015)
Dense + structured prediction

Object detection + instance segmentation

Keypoint detection

K. He, G. Gkioxari, P. Dollar, and R. Girshick, Mask R-CNN, ICCV 2017
Multi-modal prediction

Tasks
- Image Classification
- Object Detection
- Semantic Segmentation
- Depth Estimation
- Surface Normal Estimation
- Segment-based Image Generation
- Image Inpainting
- Pose Estimation
- Relationship Detection
- Image Captioning
- Visual QA
- Referring Expressions
- Situation Recognition
- Text-based Image Generation
- Visual Commonsense
- Classification in context
- Region Captioning
- GLUE Benchmark tasks
- Reading comprehension
- Natural Language Inference
- Grounded Commonsense Inference

Multi-modal prediction

Query $q$

“Which pet is in the top left?”

Visual Input $x$

$\text{ViperGPT}$

Code LLM $\pi$

API Specification

```python
def count(image, object_name) -> list[torch.Tensor]:
    # Returns object names in image

def compute_depth(image) -> torch.Tensor:
    # Returns estimated depth

def exists(image, object_name) -> bool:
    # Returns True if object_name is in image

def is_query(text) -> text:
    # Returns similarity between text and image

def verify_property(image, object_name, property) -> bool:
    # Returns True if the object has the property
```

Generated Code $\gamma$

```python
def process_query_function(image):
    image_patch = ImagePatch(image)
    pets = image_patch.find("pet")
    pets_sorted = ...
    return result
```

Code Execution $\phi$

Python Interpreter + API Implementation

Result: “Shiba Inu”

S. Menon et al. $\text{ViperGPT: Visual Inference via Python Execution for Reasoning}$, ICCV 2023
Multi-modal prediction

**Query:** How many muffins can each kid have for it to be fair?

```python
def execute_command(image):
    image_patch = ImagePatch(image)
muffin_patches = image_patch.find("muffin")
overlapping_areas = len(muffin_patches)
result = overlapping_areas // len(kid_patches)
return result
```

**Query:** Drink with zero alcohol

```python
def execute_command(image):
    image_patch = ImagePatch(image)
drink_patches = image_patch.find("drink")
for drink_patch in drink_patches:
    drink_name = drink_patch.simple_query("What is this?")
alcoholic = drink_patch.simple_query("Does the (drink_name) have alcohol?")
if alcoholic == "no":
    return drink_patch
```

**Query:** What would the founder of the brand of the car on the left say to the founder of the brand of the car on the right?

```python
def execute_command(image):
    image_patch = ImagePatch(image)
car_patches = image_patch.find("car")
car_patches.sort(key=lambda car: car.horizental_center)
left_car = car_patches[0]
right_car = car_patches[1]
left_car_brand = left_car.simple_query("What is the brand of this car?")
right_car_brand = right_car.simple_query("What is the brand of this car?")
left_car_founder = left_car.simple_query("Who is the founder of (left_car_brand)?")
right_car_founder = right_car.simple_query("Who is the founder of (right_car_brand)?")
return left_car_founder + right_car_founder
```

Result: "Ferruccio Lamborghini might say, "It's been an honor to be a rival of yours for so many years, Enzo. May our cars continue to push each other to be better and faster!""

S. Menon et al. ViperGPT: Visual Inference via Python Execution for Reasoning, ICCV 2023
Taxonomy of learning problems

• **Type of output**
  - Classification
  - Regression
  - Structured prediction
  - Dense prediction
  - Multi-modal prediction

• **Type of supervision**
  - Fully supervised
  - Unsupervised
  - Self-supervised or predictive learning
Type of supervision

• Traditional (over-simplified) dichotomy

**Supervised learning:**
clean, complete training labels for the task of interest

**Unsupervised learning:** no labels
Unsupervised learning

• Given: large collection of unlabeled data
• Goal: ???
Unsupervised learning

- Clustering
  - Discover groups of “similar” data points

Unsupervised learning

• **Dimensionality reduction, manifold learning**
  • Discover a lower-dimensional surface on which the data lives

Unsupervised learning

- Learning the data distribution
  - **Density estimation**: Find a function that approximates the probability density of the data (i.e., value of the function is high for “typical” points and low for “atypical” points)
  - An extremely hard problem for high-dimensional data…
Unsupervised learning

- Learning the data distribution
  - **Learning to sample:** Produce samples from a data distribution that mimics the training set

**Generative adversarial networks** (GANs)
Unsupervised learning

- Learning the data distribution
  - **Learning to sample:** Produce samples from a data distribution that mimics the training set

**Denoising diffusion probabilistic models** (DDPMs)

![Diagram of denoising diffusion probabilistic models](image)
Unsupervised learning

- Learning the data distribution
  - **Learning to sample**: Produce samples from a data distribution that mimics the training set

  **Denoising diffusion probabilistic models** (DDPMs)

  [Images of various samples generated by DALL-E 2]
Self-supervised or predictive learning

- Use part of the data to predict other parts of the data
  - Example: Image colorization

Self-supervised or predictive learning

- Use part of the data to predict other parts of the data
  - Example: Future prediction


Self-supervised or predictive learning

• Use part of the data to predict other parts of the data
  • Example: Grasp prediction

Self-supervised or predictive learning

- Use part of the data to predict other parts of the data
  - Example: Next/masked word prediction
Self-supervised or predictive learning

- Use part of the data to predict other parts of the data
  - Example: Masked patch prediction

K. He et al. Masked autoencoders are scalable vision learners. CVPR 2022
Taxonomy of learning problems

• **Type of output**
  • Classification
  • Regression
  • Structured prediction
  • Dense prediction
  • Multi-modal prediction

• **Type of supervision**
  • Fully supervised
  • Unsupervised
  • Self-supervised or predictive learning

• **Training regime**
  • Batch offline learning
  • Online/continual learning
  • Active learning
  • Reinforcement learning
Training regime

**Offline learning**

Training dataset

"apple"
"pear"
"tomato"
"cow"
"dog"
"horse"

Process the entire training set (typically in multiple passes)

Challenges: static dataset, high storage, memory requirements
Training regime

Transfer learning

Pre-training dataset

New dataset

Pre-trained model

Fine-tuning

Fine-tuned model
Training regime

Online learning, continual learning

Stream of training data
Update model based on continuous stream of data, do not revisit samples
Challenges: changes in the data distribution, catastrophic forgetting

Update model
Learned model
Training regime

Model asks for next training example, annotation

Pool of potential training examples

Active learning

Challenges: scalability, availability of annotators, difficulty of evaluation

Update model

Learned model
Today’s trend: **Data engines**

Task: Promptable segmentation

Data engine

Segment Anything 1B (SA-1B):
- 1+ billion masks
- 11 million images
- privacy respecting
- licensed images

A. Kirilllov et al. [Segment anything](https://arxiv.org/abs/2112.03005), ICCV 2023
Today’s trend: *Data engines*

- **Data engine steps:**
  1. **Pre-training** using public datasets
  2. **Assisted manual stage:** interactive segmentation with SAM assisting annotators
  3. **Semi-automatic stage:** SAM generates confident masks, annotators add masks to improve diversity
  4. **Fully automatic stage:** SAM generates ~100 masks per image starting with a grid of points

A. Kirillov et al. [Segment anything](https://arxiv.org/abs/2306.12721), ICCV 2023
Reinforcement learning

- Learning for an agent that can affect the world through actions

Update policy
Reinforcement learning: Examples

DeepMind's AlphaGo

DeepMind's Atari system

Sensorimotor learning
Sensorimotor learning

Figure 1: Our robot can traverse a variety of challenging terrain in indoor and outdoor environments, urban and natural settings during day and night using a single front-facing depth camera. The robot can traverse curbs, stairs and moderately rocky terrain. Despite being much smaller than other commonly used legged robots, it is able to climb stairs and curbs of a similar height. Videos at https://vision-locomotion.github.io

Sensorimotor learning

Overview

• Logistics
• The statistical learning viewpoint
• A taxonomy of learning problems
• Topics to be covered in class
Topics to be covered in class

ML basics, linear classifiers

Multilayer neural networks, backpropagation

Convolutional networks for classification

Networks for detection, dense prediction

Self-supervised learning

Generative models: GANs, image-to-image translation, diffusion models

Deep reinforcement learning

Models for sequence data

Transformers, large language models, transformers for vision