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?

Translate between languages?

Recognize object categories?

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

![Diagram of "building" region and "windows"](image)

**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…

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
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

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 time

Training examples

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

Training

Learned model

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

Date prediction

Location prediction

Vittayakorn et al. (2017)

Vo et al. (2017)
Type of output: Dense prediction

Semantic segmentation

Image colorization

Zhang et al. (2016)

Depth prediction

Wang et al. (2017)

Long et al. (2016)
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
Overview

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

S. Menon et al. 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?

Generated Code:
```python
def execute_command(image):
    image_patch = ImagePatch(image)
    muffin_patches = image_patch.find("muffin")
    kid_patches = image_patch.find("kid")
    return str(len(muffin_patches) // len(kid_patches))
```

Execution:
```python
muffin_patches = image_patch.find("muffin")
kid_patches = image_patch.find("kid")
> len(muffin_patches) = 8
> len(kid_patches) = 2
> 8/2 = 4
```
Result: 4

Query: Drink with zero alcohol

Generated Code:
```python
def execute_command(image):
    image_patch = ImagePatch(image)
    drink_patches = image_patch.find("drink")
    for drink in drink_patches:
        name = drink_patch.attribute("name")
        alcoholic = drink_patch.attribute("alcoholic")
        if alcoholic == "no":
            return drink
    return None
```

Execution:
```python
> drink_patch = drink_patches[0]
> drink = "tullamore dew"
> alcoholic = "yes"
> drink_patch.attribute("alcoholic") = "yes"
> drink_patch.attribute("alcoholic") = "no"
> drink_patch.attribute("alcoholic") = "yes"
> drink_patch.attribute("alcoholic") = "no"
```
Result: 4

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?

Generated Code:
```python
def execute_command(image):
    image_patch = ImagePatch(image)
    car_patches = image_patch.find("car")
    left_car_patch = car_patches[0]
    right_car_patch = car_patches[1]
    left_car_brand = left_car_patch.attribute("brand")
    left_car_founder = left_car_patch.attribute("founder")
    right_car_brand = right_car_patch.attribute("brand")
    right_car_founder = right_car_patch.attribute("founder")
    print(f"{left_car_founder} of {left_car_brand} might say, 'May our cars continue to push each other to be better and faster!'")
```

Execution:
```python
left_car_patch.attribute("brand") = "Ferruccio Lamborghini"
right_car_patch.attribute("brand") = "Enzo Ferrari"
left_car_founder = "Ferruccio Lamborghini"
right_car_founder = "Enzo Ferrari"
```
Result: "Ferruccio Lamborghini might say, ‘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

D. Kingma and M. Welling, Auto-Encoding Variational Bayes, ICLR 2014
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)
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)
Self-supervised or predictive learning

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

R. Zhang et al., Colorful Image Colorization, ECCV 2016
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**
- **Learned model**
Training regime

Pre-training dataset

Transfer learning

Pre-trained model

Training

Fine-tuning

Fine-tuned model

New dataset

northern waterthrush
yellow warbler
warbling vireo
tree swallow
black tern
scarlet tanager
hooded merganser
green violetear
florida jay
**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*
Training regime

Active learning

Model asks for next training example, annotation

Pool of potential training examples

Update model

Learned model

Challenges: scalability, availability of annotators, difficulty of evaluation
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. Kirillov et al. [Segment anything](https://arxiv.org/abs/2302.10488), 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

---

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/2306.11844), ICCV 2023
Reinforcement learning

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

Overview

• Logistics
• Motivation: 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