Visual Representation Learning

Sanjiban Choudhury
Perception

Lidar
Radar
Camera
Maps

State

Prediction

Decision Making
Past classes

Lidar

Radar

Camera

Maps

Perception

State

Prediction

Decision Making
Upcoming classes!
Markov Decision Process

A mathematical framework for modeling sequential decision making

\[ <S, A, C, \mathcal{T}> \]
How do we estimate state?

A mathematical framework for modeling sequential decision making.
Self-driving
How does a robot build up state?
pose 
\((x, y, \psi)\)
vel 
\((\dot{x}, \dot{y}, \dot{\psi})\)
type 
(pedestrian, car, cyclist)

But we do not observe these directly!
pose \((x, y, \psi)\)

vel \((\dot{x}, \dot{y}, \dot{\psi})\)

type (pedestrian, car, cyclist)

camera

Estimate state from observations

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

\[ s_{t+1} \rightarrow a_{t+1} \rightarrow s_{t+2} \]

\[ z_t \rightarrow s_t \]

\[ z_{t+1} \rightarrow s_{t+1} \]

\[ z_{t+2} \rightarrow s_{t+2} \]

lidar
Learning a state representation

$z_t$

$S_t$
Goal: Learn a state representation

\[ P(s_t \mid z_t, z_{t-1}, z_{t-2}, \ldots) \]
Another example: Manipulation

\[
P(s_t \mid z_t, z_{t-1}, z_{t-2}, \ldots)
\]

What is observation \(z_t\) ?

What is state \(s_t\) ?

From Jeannette Bohg: CS231A Computer Vision: From 3D Reconstruction to Recognition
Another example: Manipulation

\[ P(s_t \mid z_t, z_{t-1}, z_{t-2}, \ldots) \]

What is observation \( z_t \)?
RGB Image + Depth (Kinect)

What is state \( s_t \)?
6D Pose of the object

From Jeannette Bohg: CS231A Computer Vision: From 3D Reconstruction to Recognition
Today’s Class

What is a **visual** representation?

What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by **unsupervised** learning and **self-supervised** learning?
Today’s Class

What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by **unsupervised** learning and **self-supervised** learning?
Observed image

Drawn from memory

References:
[Bartlett, 1932]
[Intraub & Richardson, 1989]
Representation learning

Image

X

“Coral”

“Fish”

Compact mental representation

From MIT 6.8300/6.8301: Advances in Computer Vision
Activity!
Think-Pair-Share!

Think (30 sec): What makes a representation good?

Pair: Find a partner

Share (45 sec): Partners exchange ideas
Representation learning

Good representations are:

1. Compact (*minimal*)
2. Explanatory (*sufficient*)
3. Disentangled (*independent factors*)
4. Hierarchical (*feature reuse*)

5. *Make subsequent problem solving easy*

[See “Representation Learning”, Bengio 2013, for more commentary]
Traditional CV Pipeline

Feature extractors

- Edges
- Texture
- Colors

Classifier

- Segments
- Parts

“clown fish”

From MIT 6.8300/6.8301: Advances in Computer Vision
Can there be more than one representation that is useful?
Exercise: Let’s think about cats!
Represent these cats with a cat detector

Example from CS331B: Representation Learning in Computer Vision
Represent these cats with a cat detector (II)

Example from CS331B: Representation Learning in Computer Vision
Represent these cats with a cat detector (II)

Example from CS331B: Representation Learning in Computer Vision
Represent these cats with a cat detector (III)
Represent these cats with a cat detector (III)
Represent these cats with a cat detector (IV)

Example from CS331B: Representation Learning in Computer Vision
Summary of representations

Color Histograms

Deformable Part based Models (DPM)

Model based Shapes

Histogram of Gradients (HOG)

Example from CS331B: Representation Learning in Computer Vision

Felzenszwalb et al. 2010.
Dalal and Triggs, 2005.

From Jeannette Bohg: CS231A Computer Vision: From 3D Reconstruction to Recognition
Traditional CV Pipeline

Feature extractors

- Edges
- Texture
- Colors

Segments

Parts

“clown fish”

From MIT 6.8300/6.8301: Advances in Computer Vision
Learned CV Pipeline

From MIT 6.8300/6.8301: Advances in Computer Vision
Today’s Class

What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by **unsupervised** learning and **self-supervised** learning?
Today’s Class

What is a visual representation?
What makes for a good representation?

How do we learn a visual representation?

How do we train a representation by unsupervised learning and self-supervised learning?
Convolutional Neural Network
Video from ThreeBlue1Brown:  https://www.youtube.com/watch?v=KuXjwB4LzSA&t=738s
Video from ThreeBlue1Brown: [https://www.youtube.com/watch?v=KuXjwB4LzSA&t=738s](https://www.youtube.com/watch?v=KuXjwB4LzSA&t=738s)
Training a CNN Classifier

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html
CIFAR-10 Dataset
CIFAR-10 Dataset

Input: Image

Output: Class Probabilities

Truck!

0.8
Model Architecture

Input:
Image

Output:
Logits for class (1-10)

classes = ('plane', 'car', 'bird', 'cat',
'deer', 'dog', 'frog', 'horse', 'ship', 'truck')
class Net(nn.Module):
    def __init__(self):
        super().__init__()
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)

INPUT
3x32x32

c1: feature maps
6@28x28

c3: f. maps 16@10x10
s4: f. m
s2: f. maps
6@14x14

Model Architecture
Model Architecture

class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
Model Architecture

class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)

        self.conv3 = nn.Conv2d(16, 32, 5)
        self.pool2 = nn.MaxPool2d(2, 2)
        self.conv4 = nn.Conv2d(32, 64, 5)

        self.fc1 = nn.Linear(64*5*5, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)

        self.dropout = nn.Dropout2d(0.25)
        self.dropout2 = nn.Dropout2d(0.5)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool2(F.relu(self.conv2(x)))
        x = self.pool2(F.relu(self.conv3(x)))
        x = self.pool2(F.relu(self.conv4(x)))
        x = x.view(-1, 64*5*5)
        x = F.relu(self.fc1(x))
        x = self.dropout(x)
        x = F.relu(self.fc2(x))
        x = self.dropout2(x)
        x = self.fc3(x)
        return x
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        self.fc2 = nn.Linear(120, 84)
Model Architecture

class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)
Go through rest of the tutorial, train a classifier, and try to improve accuracy
(Hint: Increase the width of the network)

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html
Visualizing and Understanding CNNs
[Zeiler and Fergus, 2014]

Gabor-like filters learned by layer 1

Image patches that activate each of the layer 1 filters most strongly
Image patches that activate several of the \textbf{layer 2} neurons most strongly
Image patches that activate several of the layer 3 neurons most strongly
Image patches that activate several of the layer 4 neurons most strongly.

[Zeiler and Fergus, 2014]
Image patches that activate several of the **layer 5** neurons most strongly
CNNs learned the classical visual recognition pipeline!

From MIT 6.8300/6.8301: Advances in Computer Vision
Today’s Class

What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by **unsupervised** learning and **self-supervised** learning?
What is a visual representation?
What makes for a good representation?
How do we learn a visual representation?
How do we train a representation by unsupervised learning and self-supervised learning?
Supervised object recognition

From MIT 6.8300/6.8301: Advances in Computer Vision
Supervised object recognition

image X \rightarrow \text{Learner} \rightarrow \text{“Fish”}

label Y

From MIT 6.8300/6.8301: Advances in Computer Vision
Supervised object recognition

image X \rightarrow \text{Learner} \rightarrow \text{“Fish”}

label Y

From MIT 6.8300/6.8301: Advances in Computer Vision
Supervised object recognition

image X \rightarrow \text{Learner} \rightarrow \text{“Duck”}

From MIT 6.8300/6.8301: Advances in Computer Vision
The real world is not labeled!

Imagine a robot having to clean this kitchen.

How does it learn about all the objects in the scene?
Supervised computer vision

Hand-curated training data
+ Informative
- Expensive
- Limited to teacher’s knowledge

Vision in nature

Raw unlabeled training data
+ Cheap
- Noisy
- Harder to interpret
Learning from labels

(aka \textit{supervised learning})

Training data

\[
\begin{align*}
\{x^{(1)}, y^{(1)}\} \\
\{x^{(2)}, y^{(2)}\} & \rightarrow \text{ Learner } \\
\{x^{(3)}, y^{(3)}\} & \\
\ldots
\end{align*}
\]

\[
f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(x^{(i)}), y^{(i)})
\]
Learning without labels

(includes **unsupervised learning** and **reinforcement learning**)

\[
\begin{align*}
\text{Data} & \quad \rightarrow \quad \text{Learner} \quad \rightarrow \quad \text{?}
\end{align*}
\]

\[
\{x^{(1)}\} \\
\{x^{(2)}\} \\
\{x^{(3)}\} \\
\ldots
\]

From MIT 6.8300/6.8301: Advances in Computer Vision
Unsupervised Learning
Unsupervised Representation Learning

Image

X

“Coral”

“Fish”

Compact mental representation

From MIT 6.8300/6.8301: Advances in Computer Vision
Unsupervised Representation Learning

From MIT 6.8300/6.8301: Advances in Computer Vision
Unsupervised Representation Learning

Image $X$ → compressed image code (vector $z$) → Reconstructed image $\hat{X}$

"Autoencoder"

[e.g., Hinton & Salakhutdinov, Science 2006]
Autoencoder

\[ \hat{X} = \mathcal{F}(X) \]

\[ \arg \min_{\mathcal{F}} \mathbb{E}_X [||\mathcal{F}(X) - X||] \]

From MIT 6.8300/6.8301: Advances in Computer Vision
\[
\hat{X} = F(X)
\]
Autoencoder

Data
\{x^{(i)}\}_{i=1}^{N} \rightarrow

Learner

Objective
\mathcal{L}(f(x), x) = \|f(x) - x\|_2^2

Hypothesis space
Neural net with a bottleneck

Optimizer
SGD

\rightarrow f
Reconstructed
image

\[ \hat{X} = \mathcal{F}(X) \]
Is the code informative about object class $y$?

Logistic regression:

$$y = \sigma(Wz + b)$$

From MIT 6.8300/6.8301: Advances in Computer Vision
Layer 1 representation

Layer 6 representation

[DeCAF, Donahue, Jia, et al. 2013]

[Visualization technique: t-sne, van der Maaten & Hinton, 2008]
Data compression

Data → X → Data

From MIT 6.8300/6.8301: Advances in Computer Vision
Data prediction
aka “self-supervised learning”

From MIT 6.8300/6.8301: Advances in Computer Vision
Today’s Class

What is a **visual** representation?
What makes for a **good** representation?

How do we learn a visual representation?

How do we train a representation by **unsupervised** learning and **self-supervised** learning?
Why learn representations?

**Reinforcement Learning** (Cherry)
Predicting a scalar reward given once in a while
A few bits for some samples

**Supervised Learning** (Chocolate Coat)
Predicting category or vector of scalars per input as provided by human labels.
10-10k bits per sample

**Unsupervised / Self-Supervised Learning** (Cake)
Predicting parts of observed input or predicting future observations or events
Millions of bits per sample

Visualisation Idea by Yann LeCun
Photo by Kristina Paukshtite from Pexels
Representation learning

Good representations are:

1. Compact (minimal)
2. Explanatory (sufficient)
3. Disentangled (independent factors)
4. Hierarchical (feature reuse)
5. Make subsequent problem solving easy

[See 'Representation Learning', Bengio 2013, for more commentary]

Model Architecture

Input:
- Image

Output:
- Logits for class (1-10)

Autoencoder

\[ \text{arg min}_{\hat{F}} \mathbb{E}_{X}[||\hat{F}(X) - X||] \]

\[ \hat{X} = \hat{F}(X) \]

From MIT 6.830/6.8301: Advances in Computer Vision.