Lecture 14: Representation learning
Announcements

- Project proposal due after spring break
- We’ll post description on website
- New idea? Turn in half-page description
- Premade project idea? Just tell us which one
Today

- What representations do neural nets learn?
- Transfer learning
- Unsupervised learning
Observed image  

Drawn from memory

[Bartlett, 1932]
[Intraub & Richardson, 1989]

Source: Isola, Freeman, Torralba
Observed image  

Drawn from memory

[Bartlett, 1932]  
[Intraub & Richardson, 1989]

Source: Isola, Freeman, Torralba
“I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees.”
— Max Wertheimer, 1923
Representation learning

X

Image

“Coral”

“Fish”

Compact mental representation

Source: Isola, Freeman, Torralba
Good representations are:

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

[See “Representation Learning”, Bengio 2013, for more commentary]
Transfer learning

“Generally speaking, a good representation is one that makes a subsequent learning task easier.” — *Deep Learning*, Goodfellow et al. 2016

Source: Isola, Freeman, Torralba
Often, what we will be “tested” on is to learn to do a new thing.
Finetuning starts with the representation learned on a previous task, and adapts it to perform well on a new task.

Source: Isola, Freeman, Torralba
Finetuning

Pretraining
Object recognition

- dolphin
- cat
- grizzly bear
- angel fish
- chameleon
- clown fish
- iguana
- elephant

Finetuning
Place recognition

Source: Isola, Freeman, Torralba
The “learned representation” is just the weights and biases, so that’s what we transfer.

Source: Isola, Freeman, Torralba
Finetuning

- Pretrain a network on task A (often object recognition), resulting in parameters $W$
- Initialize a second network with some or all of $W$
- Train the second network on task B, resulting in parameters $W'$
- Why would we expect this to work?
Visualizing representations
What do deep nets internally learn?

Image features (a vector representation of the image)

Source: Isola, Freeman, Torralba
Deep net “electrophysiology”

[Zeiler & Fergus, ECCV 2014]
[Zhou et al., ICLR 2015]

Source: Isola, Freeman, Torralba
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

Source: Isola, Freeman, Torralba
Image patches that activate each of the **layer 2** neurons most strongly

Source: Isola, Freeman, Torralba
Image patches that activate each of the **layer 3** neurons most strongly

Source: Isola, Freeman, Torralba
Image patches that activate each of the layer 4 neurons most strongly

Source: Isola, Freeman, Torralba
Image patches that activate each of the layer 5 neurons most strongly.
CNNs learned the classical visual recognition pipeline

Edges
Texture
Colors

Segments
Parts

“clown fish”

Source: Isola, Freeman, Torralba
What do deep nets internally learn?

A CNN is a multiscale, hierarchical representation of data.

Representations!

Source: Isola, Freeman, Torralba
Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

- For each unit (neuron) in network, find which images it is most selective for (cause it to have highest activation)

- Find which pixels in these images are responsible by occluding regions and seeing which pixels, when occluded, cause activation to change the most

- Use a network trained on scene recognition

Source: Isola, Freeman, Torralba
Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

pool 1

[http://people.csail.mit.edu/torralba/research/drawCNN/drawNet.html]

Source: Isola, Freeman, Torralba
Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

pool 2

Source: Isola, Freeman, Torralba
Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

conv 4

Source: Isola, Freeman, Torralba
Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

pool 5

Source: Isola, Freeman, Torralba
Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

Source: Isola, Freeman, Torralba
Layer 1 representation

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

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

Source: Isola, Freeman, Torralba

Layer 6 representation

- structure, construction
- covering
- commodity, trade good, good
- conveyance, transport
- invertebrate
- bird
- hunting dog
Transferring CNN features

“Dog”

Object recognition net
Transferring CNN features

Object recognition net
Simple feature transfer

\[ y = \sigma(\mathbf{Wz} + \mathbf{b}) \]

Logistic regression:

\[ y \rightarrow \text{"Rainforest"} \]
Transferring CNN features

Hand-crafted features

CNN features pretrained on ImageNet + linear classifier [Donahue et al. 2013]

[5.0\%]

[Xiao et al., CVPR 2010]
Finetuning for object detection

- ImageNet pretraining speeds up object detection training.
- No change in accuracy for large datasets
- Big performance gains for small/medium datasets though (e.g. 1K examples per class)!

[He et al. 2018]
How do we learn good representations?
Supervised object recognition

image X → Learner → "Fish" → label Y

Source: Isola, Freeman, Torralba
Supervised object recognition

Learner

image X → "Fish" 

label Y

Source: Isola, Freeman, Torralba
Supervised object recognition

image X $\rightarrow$ Learner $\rightarrow$ “Fish”

Source: Isola, Freeman, Torralba
Supervised object recognition

image X → Learner → “Duck”

Source: Isola, Freeman, Torralba
**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

Source: Isola, Freeman, Torralba
Learning from examples
(aka \textit{supervised learning})

Training data

\[
\begin{align*}
\{x_1, y_1\} & \rightarrow \quad \text{Learner} \\
\{x_2, y_2\} & \rightarrow \quad \rightarrow \quad f : X \rightarrow Y \\
\{x_3, y_3\} \\
\cdots
\end{align*}
\]

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

Source: Isola, Freeman, Torralba
Representation Learning

\[
\{x_1\} \quad \rightarrow \quad \text{Learner} \quad \rightarrow \quad \text{Representations}
\]

Source: Isola, Freeman, Torralba
Self-supervised learning

Common trick:

- Convert “unsupervised” problem into “supervised” empirical risk minimization

- Do so by cooking up “labels” (prediction targets) from the raw data itself

- Designing new algorithms still takes a lot of trial and error.

Source: Isola, Freeman, Torralba
Unsupervised Representation Learning

Image

X

“Coral”

“Fish”

Compact mental representation

Source: Isola, Freeman, Torralba
Unsupervised Representation Learning

Source: Isola, Freeman, Torralba
Unsupervised Representation Learning

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

“Autoencoder”

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

Source: Isola, Freeman, Torralba
Autoencoder

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

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

Source: Isola, Freeman, Torralba
$X \xrightarrow{\mathcal{F}} \hat{X} = \mathcal{F}(X)$

“Fish”

“Coral”

Source: Isola, Freeman, Torralba
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

\[ \rightarrow f \]

Source: Isola, Freeman, Torralba
Data compression

Source: Isola, Freeman, Torralba
Label prediction

e.g., image classification

Source: Isola, Freeman, Torralba
Data prediction
aka “self-supervised learning”

Source: Isola, Freeman, Torralba
Grayscale image: L channel
\[ X \in \mathbb{R}^{H \times W \times 1} \]

Color information: ab channels
\[ \hat{Y} \in \mathbb{R}^{H \times W \times 2} \]

Source: Isola, Freeman, Torralba

[Zhang, Isola, Efros, ECCV 2016]
Visualizing units

Source: Isola, Freeman, Torralba

[Zeiler & Fergus, ECCV 2014]
[Zhou et al., ICLR 2015]
Stimuli that drive selected neurons (conv5 layer)

- faces
- dog faces
- flowers
Classification performance
ImageNet Task [Russakovsky et al. 2015]

Source: Isola, Freeman, Torralba
Represent image as a vector of neural activations
(perhaps representing a vector of detected texture patterns or object parts)

Source: Isola, Freeman, Torralba
X2vec methods are also called embeddings of X, e.g., a word embedding.
Words with similar meanings should be near each other
word2vec

Words with similar meanings should be near each other

Proxy: words that are used in the same context tend to have similar meanings

words with similar contexts should be near each other

Source: Isola, Freeman, Torralba
Next to the 'sofa' is a desk, and a 'person' is sitting behind it.

'sofa'
'armchair'
'bench'
'chair'
'deck chair'
'ottoman'
'seat'
'stool'
'swivel chair'
'loveseat'
...

'person'
'man'
'woman'
'child'
'teenager'
'girl'
'boy'
'baby'
'daughter'
'son'
...


I parked the **car** in a nearby street. It is a red **car** with two doors, …

I parked the **vehicle** in a nearby street…

I parked the **car** in a nearby street. It is a red **car** with two doors, …

word2vec

Word = ‘car’

Hidden layer
Soft-max classifier

Encoder
Decoder

Output prob. That each word is in the context of the input word

word2vec, training

Output prob. That each word is in the context of the input word

\[ p = \frac{e^{x_i}}{\sum_j e^{x_j}} \]

Algebraic operations with the vector representation of words

\[ X = \text{Vector(“Paris”) – vector(“France”) + vector(“Italy”)} \]

Closest nearest neighbor to \( X \) is \( \text{vector(“Rome”)} \)
Context as Supervision
[Collobert & Weston 2008; Mikolov et al. 2013]
Context Prediction as Supervision

[Slide credit: Carl Doersch]
Semantics from a non-semantic task

[Slide credit: Carl Doersch]
Relative Position Task

8 possible locations

Randomly Sample Patch
Sample Second Patch

[Slide credit: Carl Doersch]
Patch Embedding (representation)

Input

Nearest Neighbors

Note: connects *across* instances!

[Slide credit: Carl Doersch]
Revisiting autoencoders

Is reconstruction necessary?

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

\[ z^T z \rightarrow \text{High} \]
\[ z^T x_1 \rightarrow \text{Low} \]

[Wu et al., Instance discrimination 2018], [He et al. Momentum contrastive learning 2019]
Contrastive learning

Maximize:

\[
\frac{\exp\{\mathbf{z}^\top \mathbf{z}\}}{\exp\{\mathbf{z}^\top \mathbf{z} + \sum_i \mathbf{z}^\top \mathbf{x}_i\}}
\]

Equivalent to softmax loss with each image as a category

[Wu et al., Instance discrimination 2018], [He et al. Momentum contrastive learning 2019]
Contrastive learning

Can build invariance. Compare to *warped* images.

\[
\text{exp}\left\{ z^\top \tilde{z} \right\} \over \text{exp}\left\{ z^\top \tilde{z} + \sum_i z^\top x_i \right\}
\]

[Wu et al., Instance discrimination 2018], [He et al. Momentum contrastive learning 2019]
Contrastive learning

Can build invariance.
Compare to \textit{warped} images.

\[
\frac{\exp\{\mathbf{z}^\top \tilde{\mathbf{z}}\}}{\exp\{\mathbf{z}^\top \tilde{\mathbf{z}} + \sum_i \mathbf{z}^\top \mathbf{x}_i\}}
\]

[Wu et al., Instance discrimination 2018], [He et al. Momentum contrastive learning 2019]
Contrastive learning

(a) Original  (b) Crop and resize  (c) Crop, resize (and flip)  (d) Color distort. (drop)  (e) Color distort. (jitter)

(f) Rotate \{90^\circ, 180^\circ, 270^\circ\}  (g) Cutout  (h) Gaussian noise  (i) Gaussian blur  (j) Sobel filtering

From [Chen et al., SinCLR, 2020]
Performance snapshot

ImageNet linear classification
Object detection finetuning

Comparable in many cases to supervised pretraining!
**Egomotion**

- Agrawal et al. ICCV 2015.
- Jayaraman et al. ICCV 2015.

**Context**

- Pathak et al. CVPR 2016.
- Noroozi and Favaro. ECCV 2016.
- Doersch et al. ICCV 2015.

**Video**

- Wang et al. ICCV 2015.
- Pathak et al. CVPR 2017.

**Generative Modeling**


**Autoencoders**


**Denoising Autoencoders**


**Goal:** Set up a pre-training scheme to induce a “useful” representation.
1. Deep nets learn *representations*

2. This is useful because representations transfer — they act as prior knowledge that enables quick learning on new tasks

3. Representations can also be learned without labels

4. Without labels there are many ways to learn representations. We saw:
   1. representations as compressed codes
   2. representations that are predictive of missing data

Source: Isola, Freeman, Torralba
Next time: Sight, sound, and touch