Lecture 18
Representation Learning
18. Representation Learning

- Representations in the brain
- What is learned by a deep net?
- Transfer learning and finetuning
- Unsupervised and self-supervised learning
Observed image

Drawn from memory

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

Drawn from memory

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

Image

“Fish”

“Coral”

Compact mental representation
Representation learning

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]
Convolution is pointwise multiplication in the frequency domain.
Classification units

PIT/AIT

V4/PIT

V2/V4

V1/V2

[Serre, 2014]
Classical object recognition

Feature extractors
- Edges
- Texture
- Colors

Classifier
- Segments
- Parts

“clown fish”
Deep learning

Learned

“clown fish”
What do deep nets internally learn?

X

Image

“image features”
(a vector representation of the image)

“Fish”
Deep Net “Electrophysiology”

[Zeiler & Fergus, ECCV 2014]
[Zhou et al., ICLR 2015]
Gabor-like filters learned by **layer 1**

Image patches that activate each of the **layer 1** filters most strongly

[Zeiler and Fergus, 2014]
Image patches that activate each of the **layer 2** neurons most strongly
Image patches that activate each of the layer 3 neurons most strongly
Image patches that activate each of the **layer 4** neurons most strongly.
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”
Alexnet — [Krizhevsky et al. NIPS 2012]

[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)
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

AlexNet
Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

pool 1

[http://people.csail.mit.edu/torralba/research/drawCNN/drawNet.html]
Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]
Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]

conv 4
Object Detectors Emergence in Deep Scene CNNs

[e.g., Zhou et al., ICLR 2015]

pool 5
Object Detectors Emergence in Deep Scene CNNs

[Zhou et al., ICLR 2015]
Represent image as a vector of neural activations (perhaps representing a vector of detected texture patterns or object parts)
Investigating a representation via similarity analysis

How similar are these two images?

How about these two?

[Kriegeskorte et al. 2008]
Investigating a representation via similarity analysis

Represenational Dissimilarity Matrix

\[ \| h_i - h_j \| \]

Neural activation vector

[Kriegeskorte et al. 2008]
Investigating a representation via similarity analysis

IT Neuronal Units

Deep net (in particular, HMO)

[Yamins et al., PNAS 2014]
Investigating a representation via similarity analysis

Deep nets and the primate brain both learn similar metric spaces.

Deep nets organize visual information similarly to how our brains do!

[Yamins et al., PNAS 2014]
What do deep nets internally learn?

A CNN is a multiscale, hierarchical representation of data.
Transfer learning

“Generally speaking, a good representation is one that makes a subsequent learning task easier.” — Deep Learning, Goodfellow et al. 2016
\[
\{x_i, y_i\}_{i=1}^{N}
\]
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.
Finetuning in practice

- Pretrain a network on task A (often object recognition), resulting in parameters $W$ and $b$

- Initialize a second network with some or all of $W$ and $b$

- Train the second network on task B, resulting in parameters $W'$ and $b'$
Finetuning in practice

Pretraining
Object recognition

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

Finetuning
Place recognition
The “learned representation” is just the weights and biases, so that’s what we transfer
Supervised object recognition

image $X$ \rightarrow \text{Learner} \rightarrow \text{label } Y

"Fish"
Supervised object recognition

image X \rightarrow \text{Learner} \rightarrow \text{“Fish”}
Supervised object recognition

image X \rightarrow \text{Learner} \rightarrow \text{"Fish"}

label Y
Supervised object recognition

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

label Y
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 examples

(aka supervised learning)

Training data

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

\[
f^* = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(x_i), y_i)
\]
Learning without examples
(includes unsupervised learning and reinforcement learning)

\[
\begin{align*}
\text{Data} \\
\{x_1\} \\
\{x_2\} & \rightarrow \text{Learner} \\
\{x_3\} \\
\ldots
\end{align*}
\]
Density modeling

Data → Learner → Density

\( p : \mathcal{X} \rightarrow [0, 1] \)

Clustering

Data → Learner → Clusters

\[ f : \mathcal{X} \rightarrow \{1, \ldots, k\} \]
Representation Learning

\[
\begin{align*}
\{x_1\} & \quad \rightarrow \quad \text{Learner} \\
\{x_2\} & \\
\{x_3\} & \\
\ldots & \\
\end{align*}
\]

\rightarrow \quad \text{Representations}
Unsupervised Representation Learning

Image

X

“Coral”

“Fish”

Compact mental representation
Unsupervised Representation Learning

Image

compressed image code
(vector $\mathbf{z}$)
Unsupervised Representation Learning

“Autoencoder”

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

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

\[ \arg \min_{\mathcal{F}} \mathbb{E}_{\mathbf{X}} [||\mathcal{F}(\mathbf{X}) - \mathbf{X}||] \]
$\hat{X} = \mathcal{F}(X)$
Autoencoder

Data \{x_i\}_{i=1}^{N} \rightarrow \text{Learner}

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

- Hypothesis space
  Neural net with a bottleneck

- Optimizer
  SGD

\rightarrow f
Data compression

Data

\(X\)

\(\hat{X}\)

Data
Label prediction

Data $X$ \rightarrow \text{Label} \rightarrow y

e.g., image classification
Data prediction
aka “self-supervised learning”

\[ \hat{X}_2 \]
Grayscale image: L channel
\[ X \in \mathbb{R}^{H \times W \times 1} \]

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

[Zhang, Isola, Efros, ECCV 2016]
Deep Net “Electrophysiology”

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

Supervised
- implausible label

"cow"
Target

Input

Self-Supervised
- derives label from a co-occurring input to another modality

Input 1

moo

Input 2


[see also “Six lessons from babies”, Smith and Gasser 2005]
“Multiview” self-supervised learning

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

\[
\text{arg min}_f \sum_i D(f(g(x_i)) - h(x_i))
\]

Distance function

\( g \) and \( h \) are two “views” of the data \( x \), e.g., two different image channels

\( \rightarrow f \)
The allegory of the cave
Ambient Sound Provides Supervision for Visual Learning

Andrew Owens  Jiajun Wu  Josh McDermott
William Freeman  Antonio Torralba
Predicting ambient sound
What did the model learn?

Unit #90 of 256

Strongest responses in dataset

Visualization method from (Zhou 2015)

[Slide credit: Andrew Owens]
Is the code informative about object class $y$?

Logistic regression:

$$y = \sigma(Wz + b)$$
Layer 1 representation

Layer 6 representation

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

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

[Visualization technique: t-sne, van der Maaten & Hinton, 2008]
Classification performance
ImageNet Task [Russakovsky et al. 2015]

Accuracy

autoencoder
colorization

Layer
conv1 pool1 conv2 pool2 conv3 conv4 conv5 pool5

Raw Data
Predicted Color Channels
Reconstructed Data
Raw Grayscale Channel

$X$
$\hat{X}$
$X_1$
$X_2$

conv3
predicted
colorization

raw
data
reconstructed
data
autoencoder
colorization
conv1 pool1 conv2 pool2 conv3 conv4 conv5 pool5

Classification performance
ImageNet Task [Russakovsky et al. 2015]
How to represent words as numbers

“Fish” ➔ [1,0,0,0,0,0,0,…]

“Shark” ➔ [0,1,0,0,0,0,0,…]

“Whale” ➔ [0,0,1,0,0,0,0,…]

“Water” ➔ [0,0,0,1,0,0,0,…]

“Cat” ➔ [0,0,0,0,1,0,0,…]

“Couch” ➔ [0,0,0,0,0,1,0,…]

“Sun” ➔ [0,0,0,0,0,0,1,…]
Represent image as a vector of neural activations
(perhaps representing a vector of detected texture patterns or object parts)
word2vec

“Elephant” → dense vector representation of word

one-hot vector representation of word

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

“Meaning is use” — Wittgenstein
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'
...
word2vec

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**

word2vec

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

Hidden layer

Soft-max classifier

Encoder

Decoder

word2vec, training

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

$p = \frac{e^{x_i}}{\sum_j e^{x_j}}$

word2vec, training

- In training maximize log-likelihood over the training set:

\[
\sum_{t=1}^{T} \sum_{i=-c}^{c} \log p(w_{t+i} | w_t)
\]

\(T \ldots \) training set size
\(c \ldots \) context window size

\[
p = \frac{e^{xi}}{\sum_j e^{xj}}
\]

At test time, $w$ is our word embedding. The encoding is just a look up table.

Algebraic operations with the vector representation of words

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

Closest nearest neighbor to \( X \) is \( \text{vector(“Rome”) } \)
Examples from https://www.tensorflow.org/tutorials/representation/word2vec
Unsupervised visual representation learning by context prediction

[Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015]
Context as Supervision
[Collobert & Weston 2008; Mikolov et al. 2013]

"house, where the professor lived without his wife and child; or so he said jokingly sometimes: “Here’s where I live. My house.” His daughter often added, without resentment, for the visitor’s information, “It started out to be for me, but it’s really his.” And she might reach in to bring forth an inch-high table lamp with fluted shade, or a blue dish the size of her little fingernail, marked “Kitty” and half full of eternal saline, but she was sure to replace these, after they had been admired, pretty near exactly where they had been. The little house was very orderly, and just big enough for all it contained, though to some tastes the bric-à-brac in the parlor might seem excessive. The daughter’s preference was for the store-bought gimmicks and appliances, the toasters and carpet sweepers of Lilliput, but she knew that most adult vis..."
Context Prediction as Supervision

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

[Slide credit: Carl Doersch]
Relative Position Task

Randomly Sample Patch
Sample Second Patch

8 possible locations

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

Input  Nearest Neighbors

Note: connects *across* instances!

[Slide credit: Carl Doersch]
Goal: Set up a pre-training scheme to induce a “useful” representation
How Much Information is the Machine Given during Learning?

- **“Pure” Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - A few bits for some samples

- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - 10→10,000 bits per sample

- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - Millions of bits per sample

[Slide Credit: Yann LeCun]
Summary

1. Deep nets learn *representations*, just like our brains do.

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, which is great since labels are expensive and limiting.

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