Motivation

• So far we assumed access to plentiful labeled data
• **What if we have limited or no labeled data?**
• Learn from unlabeled data (unsupervised learning)
  – Use structure in data as “labels” (self-supervised learning)
  – Use structure in data to generate similar data (generation)
  – Mine for interesting patterns (discovery)
• Another approach (not discussed): carefully choose which data to label (active learning, human-in-the-loop)
Supervised vs Unsupervised Learning

**Supervised Learning**

**Data:** \((x, y)\)
\(x\) is data, \(y\) is label

**Goal:** Learn a function to map \(x \rightarrow y\)

**Examples:** Classification, regression, object detection, semantic segmentation, image captioning, etc.

**Unsupervised Learning**

**Data:** \(x\)
Just data, **no labels**!

**Goal:** Learn some underlying hidden structure of the data

**Examples:** Clustering, dimensionality reduction, feature learning, density estimation, etc.
Plan for this last lecture

• Self-supervised learning
  – For images
  – For video

• Visual discovery
  – Discovering style-specific elements

• Generation not recognition
  – Theory/technique
  – Applications
Unsupervised Visual Representation Learning by Context Prediction

Carl Doersch, Alexei Efros and Abhinav Gupta

ICCV 2015
ImageNet + Deep Learning

- Image Retrieval
- Detection (RCNN)
- Segmentation (FCN)
- Depth Estimation
- ...

Do we need semantic labels?
Do we need this task?
Do we need this task?
Context as Supervision

[Collobert & Weston 2008; Mikolov et al. 2013]

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 visitors would
Context Prediction for Images

Semantics from a non-semantic task

Relative Position Task

Randomly Sample Patch
Sample Second Patch

8 possible locations

CNN

Classifier

Note: connects *across* instances!

Architecture

Softmax loss
Fully connected

Max Pooling
LRN
Convolution
Max Pooling
LRN
Convolution

Fully connected

Max Pooling
Convolution
Convolution
Convolution
Convolution

Max Pooling
LRN
Max Pooling
LRN
Max Pooling

Tied Weights

Patch 1

Patch 2

What is learned?

<table>
<thead>
<tr>
<th>Input</th>
<th>Ours</th>
<th>ImageNet AlexNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Pre-Training for R-CNN

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

Pre-train on relative-position task, w/o labels


[Girshick et al. 2014]
### VOC 2007 Performance
(pretraining for R-CNN)

<table>
<thead>
<tr>
<th>Method</th>
<th>% Average Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet Labels</td>
<td>54.2</td>
</tr>
<tr>
<td>Ours</td>
<td>46.3</td>
</tr>
<tr>
<td>No Pretraining</td>
<td>40.7</td>
</tr>
</tbody>
</table>

Shuffle and Learn: Unsupervised Learning using Temporal Order Verification

Ishan Misra, C. Lawrence Zitnick, and Martial Hebert

ECCV 2016
Fig. 1: (a) A video imposes a natural temporal structure for visual data. In many cases, one can easily verify whether frames are in the correct temporal order (shuffled or not). Such a simple sequential verification task captures important spatiotemporal signals in videos. We use this task for unsupervised pre-training of a Convolutional Neural Network (CNN). (b) Some examples of the automatically extracted positive and negative tuples used to formulate a classification task for a CNN.
Fig. 2: (a) We sample tuples of frames from high motion windows in a video. We form positive and negative tuples based on whether the three input frames are in the correct temporal order. (b) Our triplet Siamese network architecture has three parallel network stacks with shared weights up to the \( fc7 \) layer. Each stack takes a frame as input, and produces a representation at the \( fc7 \) layer. The concatenated \( fc7 \) representations are used to predict whether the input tuple is in the correct temporal order.
Benefit of unsupervised but in-domain training

Table 2: Mean classification accuracies over the 3 splits of UCF101 and HMDB51 datasets. We compare different initializations and finetune them for action recognition.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Initialization</th>
<th>Mean Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCF101</td>
<td>Random</td>
<td>38.6</td>
</tr>
<tr>
<td></td>
<td>(Ours) Tuple verification</td>
<td><strong>50.2</strong></td>
</tr>
<tr>
<td>HMDB51</td>
<td>Random</td>
<td>13.3</td>
</tr>
<tr>
<td></td>
<td>UCF Supervised</td>
<td>15.2</td>
</tr>
<tr>
<td></td>
<td>(Ours) Tuple verification</td>
<td><strong>18.1</strong></td>
</tr>
</tbody>
</table>
Plan for this last lecture

• Self-supervised learning
  – For images
  – For video

• Visual discovery
  – Discovering style-specific elements

• Generation not recognition
  – Theory/technique
  – Applications
What Makes Paris Look like Paris?

Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, Alexei Efros

SIGGRAPH 2012
One of these is from Paris
Raise your hand if...

...this is Paris

Raise your hand if...

We showed 20 subjects:
- 100 Random Street View Images
- 50 from Paris
- They classified Paris vs non-Paris
- Accuracy: 79%

How do they know?

We showed 20 subjects:
- 100 Random Street View Images
- 50 from Paris
- They classified Paris non-Paris
- Accuracy: 79%

How do they know?

Our Goal:

Given a large geo-tagged image dataset, we automatically discover **visual elements** that characterize a geographic location

*Why might this be a useful task?*

Our Hypothesis

• The visual elements that capture Paris:
  – Frequent: Occur often in Paris
  – Discriminative: Are not found outside Paris

Step 1: Nearest Neighbors for Every Patch
Using normalized correlation of HOG features as a distance metric

Step 2: Find the Parisian Clusters by Sorting

Paris: A Few Top Elements

Elements from Prague

Elements from London

Elements from Barcelona

In the U.S.

Elements from San Francisco

Elements from Boston

Plan for this last lecture

• Self-supervised learning
  – For images
  – For video

• Visual discovery
  – Discovering style-specific elements

• Generation not recognition
  – Theory/technique
  – Applications
Generative Models

Training data $\sim p_{\text{data}}(x)$

Generated samples $\sim p_{\text{model}}(x)$

Want to learn $p_{\text{model}}(x)$ similar to $p_{\text{data}}(x)$
Generative Models

Training data ~ $p_{data}(x)$

Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Addresses density estimation, a core problem in unsupervised learning

Several flavors:
- Explicit density estimation: explicitly define and solve for $p_{model}(x)$
- Implicit density estimation: learn model that can sample from $p_{model}(x)$ w/o explicitly defining it
Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.

Q: What can we use to represent this complex transformation?
Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.

Q: What can we use to represent this complex transformation?

A: A neural network!

Output: Sample from training distribution

Input: Random noise

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014
Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

**Generator network**: try to fool the discriminator by generating real-looking images

**Discriminator network**: try to distinguish between real and fake images
Training GANs: Two-player game

**Generator network**: try to fool the discriminator by generating real-looking images

**Discriminator network**: try to distinguish between real and fake images

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014
Adversarial Networks Framework

Generator
\[ x \sim G(z) \]

Discriminator
Real vs. Fake

[Goodfellow et al. 2014]

Jun-Yan Zhu
Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

**Generator network**: try to fool the discriminator by generating real-looking images

**Discriminator network**: try to distinguish between real and fake images

Train jointly in **minimax game**

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log (1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$
Training GANs: Two-player game

**Generator network**: try to fool the discriminator by generating real-looking images

**Discriminator network**: try to distinguish between real and fake images

Train jointly in **minimax game**

Minimax objective function:

\[
\min_{\theta_g} \max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log (1 - D_{\theta_d}(G_{\theta_g}(z))) \right]
\]

Discriminator outputs likelihood in (0,1) of real image

Discriminator output for real data \(x\)

Discriminator output for generated fake data \(G(z)\)
Training GANs: Two-player game

Ian Goodfellow et al., “Generative Adversarial Nets”, NIPS 2014

**Generator network**: try to fool the discriminator by generating real-looking images

**Discriminator network**: try to distinguish between real and fake images

Train jointly in **minimax game**

Minimax objective function:

\[
\min_{\theta_g} \max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]
\]

- Discriminator \((\theta_d)\) wants to **maximize objective** such that \(D(x)\) is close to 1 (real) and \(D(G(z))\) is close to 0 (fake)
- Generator \((\theta_g)\) wants to **minimize objective** such that \(D(G(z))\) is close to 1 (discriminator is fooled into thinking generated \(G(z)\) is real)
Training GANs: Two-player game

Minimax objective function:

\[
\min_{\theta_g} \max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]
\]

Alternate between:

1. **Gradient ascent** on discriminator

   \[
   \max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]
   \]

2. **Gradient descent** on generator

   \[
   \min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))
   \]
Training GANs: Two-player game

Generator network: try to fool the discriminator by generating real-looking images

Discriminator network: try to distinguish between real and fake images

Real or Fake

<table>
<thead>
<tr>
<th>Discriminator Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fake Images (from generator)</td>
</tr>
<tr>
<td>Real Images (from training set)</td>
</tr>
</tbody>
</table>

Generator network

Random noise

After training, use generator network to generate new images

Serena Young
Samples from the model look amazing!

Radford et al, ICLR 2016

Serena Young
Generative Adversarial Nets: Interpretable Vector Math

Samples from the model

Average features, do arithmetic

Smiling woman  Neutral woman  Neutral man  =  Smiling Man

Radford et al, ICLR 2016

Adapted from Serena Young
Generative Adversarial Nets: Interpretable Vector Math

Glasses man  No glasses man  No glasses woman

Radford et al, ICLR 2016

Woman with glasses

Serena Young
Celebrities Who Never Existed

Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018
StarGAN

Choi et al., “StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation”, CVPR 2018
GANs for Privacy (Action Detection)

Identity: Jessica
Action: Applying Make-up on Lips

Identity: ???
Action: Applying Make-up on Lips

Ren et al., “Learning to Anonymize Faces for Privacy Preserving Action Detection”, arxiv 2018
Artificial Fashion: vue.ai
Edges \rightarrow Images

Edges from [Xie & Tu, 2015]
Sketches $\rightarrow$ Images

Data from [Eitz, Hays, Alexa, 2012]
#edges2cats

[Christopher Hesse]

Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

https://affinelayer.com/pixsrv/
Changing artistic style

Input | Monet | Van Gogh | Cezanne | Ukiyo-e

Pix2pix / CycleGAN
Changing seasons