Motivation

• So far we’ve assumed access to plentiful labeled data
• How is this data obtained?
Crowdsourcing

Workers

Task: Dog?

Answer: Yes

Pay: $0.01

www.mturk.com

Task

Is this a dog?

- Yes
- No

$0.01

Alex Sorokin
Crowdsourcing via games

• The ESP Game
  – Two-player online game
  – Partners don’t know each other and can’t communicate
  – Objective of the game: type the same word
  – The only thing in common is an image

THE **ESP GAME**

**PLAYER 1**

GUESSING: CAR

GUESSING: HAT

GUESSING: KID

SUCCESS!

YOU AGREE ON CAR

**PLAYER 2**

GUESSING: BOY

GUESSING: CAR

SUCCESS!

YOU AGREE ON CAR

Motivation

• So far we’ve assumed access to plentiful labeled data

• **What if we have limited or no labeled data?**

• One approach: learn from unlabeled data (unsupervised learning)
  – Mine for interesting patterns (discovery)
  – Use supervision (labels) inherent in the data (self-supervised learning)

• 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.
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.

Cat
Supervised Learning

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

Goal: Learn a \textit{function} to map \(x \rightarrow y\)

Examples: Classification, regression, object detection, semantic segmentation, image captioning, etc.
Supervised vs Unsupervised Learning

Supervised Learning

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

**Goal:** Learn a function to map x -> y

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

Semantic Segmentation

GRASS, CAT, TREE, SKY

Serena Young
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.

A cat sitting on a suitcase on the floor

Image captioning
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.
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.

K-means clustering

Serena Young
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.

**Autoencoders** (Feature learning)

L2 Loss function:

$$\| x - \hat{x} \|^2$$
Supervised vs Unsupervised Learning

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.
Supervised vs Unsupervised Learning

**Supervised Learning**

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

**Goal:** Learn a *function* to map x -> 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

• Generative models
  – Theory/technique
  – Applications
  – Generating synthetic training data
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 even need semantic labels?

Materials?
Parts?
Pose?
Boundaries?
Geometry?

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 milk, 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 visitors would
Context Prediction for Images

1  2  3

4  5

6  7  8

Semantics from a non-semantic task

Relative Position Task

8 possible locations

Note: connects *across* instances!

What is learned?

<table>
<thead>
<tr>
<th>Input</th>
<th>Ours</th>
<th>ImageNet AlexNet</th>
</tr>
</thead>
</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.

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

• Generative models
  – Theory/technique
  – Applications
  – Generating synthetic training data
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%

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

Approach Summary

1. A Cluster for Every Patch

Our Approach

1. A cluster for every patch
2. Find clusters that are mostly Parisian

Approach Summary

1. A cluster for every patch
2. Find clusters that are mostly Parisian
3. Refine clusters by making them more Parisian

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

• Generative models
  – Theory/technique
  – Applications
  – Generating synthetic training data
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

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
Why Generative Models?

- Realistic samples for artwork, super-resolution, colorization, etc.
- Generative models can be used to enhance training datasets with diverse synthetic data
- Generative models of time-series data can be used for simulation

Adapted from Serena Young
Generative Adversarial Networks

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

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?
Generative Adversarial Networks

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!

Input: Random noise

Output: Sample from training distribution
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
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

D tries to output 1

Differentiable function D

x sampled from data

D tries to output 0

Differentiable function D

x sampled from model

Differentiable function G

Input noise Z
Adversarial Networks Framework

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

Discriminator
Real vs. Fake

[Goodfellow et al. 2014]

Jun-Yan Zhu
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:

$$\min_{\theta_g} \max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log (1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$
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 output** for real data $x$
- **Discriminator output** for generated fake data $G(z)$
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

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)

Serena Young
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**

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

---

**Diagram explanation**
- **Random noise** \( z \)
- **Generator network**
- **Discriminator network**
- **Fake Images** (from generator)
- **Real Images** (from training set)

**After training, use generator network to generate new images**
Generative Adversarial Nets

Samples from the model look amazing!

Radford et al, ICLR 2016
Generative Adversarial Nets: Interpretable Vector Math

Samples from the model

Smiling woman  Neutral woman  Neutral man  Radford et al, ICLR 2016
Generative Adversarial Nets: Interpretable Vector Math

Smiling woman  Neutral woman  Neutral man

Samples from the model

Average Z vectors, do arithmetic

Radford et al, ICLR 2016
Generative Adversarial Nets: Interpretable Vector Math

Samples from the model

Smiling woman  Neutral woman  Neutral man

Average Z vectors, do arithmetic

Smiling Man

Radford et al, ICLR 2016
Generative Adversarial Nets: Interpretable Vector Math

Glasses man   No glasses man   No glasses woman

Radford et al, ICLR 2016  

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
What is in this image?

(Yeh et al., 2016)
Generative modeling reveals a face

(Yeh et al., 2016)
Creative Adversarial Networks

CAN: Top ranked by human subjects

(Elgammal et al., 2017)
Artificial Fashion: vue.ai

Ian Goodfellow
Celebrities Who Never Existed

Karras et al., “Progressive Growing of GANs for Improved Quality, Stability, and Variation”, ICLR 2018
GANs for Privacy (Action Detection)

Ren et al., “Learning to Anonymize Faces for Privacy Preserving Action Detection”, arxiv 2018
2017: Year of the GAN

Better training and generation

Source->Target domain transfer

Text -> Image Synthesis
this small bird has a pink breast and crown, and black primaries and secondaries. this magnificent fellow is almost all black with a red crest, and white cheek patch.

Reed et al. 2017.

Many GAN applications


Conditional GANs

$\mathbf{X} \xrightarrow{\mathbf{G}} \mathbf{G}(\mathbf{x}) \xrightarrow{\mathbf{D}} \text{real or fake \textit{pair} ?}$

Adapted from Jun-Yan Zhu
Edges → Images

Edges from [Xie & Tu, 2015]

Pix2pix / CycleGAN
Sketches $\rightarrow$ Images

Trained on Edges $\rightarrow$ Images

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

Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

Pix2pix / CycleGAN

https://affinelayer.com/pixsrv/
Paired $x_i$, $y_i$

Unpaired $X$, $Y$

Jun-Yan Zhu
Cycle Consistency

Discriminator $D_Y$: $L_{GAN}(G(x), y)$
Real zebras vs. generated zebras

Zhu et al., “Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks”, ICCV 2017
Discriminator $D_X$: $L_{GAN}(F(y), x)$
Real horses vs. generated horses

Discriminator $D_Y$: $L_{GAN}(G(x), y)$
Real zebras vs. generated zebras

Zhu et al., “Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks”, ICCV 2017
Forward cycle loss: $\|F(G(x)) - x\|_1$
Cycle Consistency

Forward cycle loss: \( \| F(G(x)) - x \|_1 \)

Zhu et al., “Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks”, ICCV 2017
Cycle Consistency

Forward cycle loss: \( \|F(G(x)) - x\|_1 \)

Backward cycle loss: \( \|G(F(y)) - y\|_1 \)

Zhu et al., “Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks”, ICCV 2017
Monet’s paintings → photos
Monet’s paintings → photos
Pix2pix / CycleGAN
#CycleGAN

Monet → Thomas Kinkade @David Fouhey

Resurrecting Ancient Cities @ Jack Clark

Birds @Matt Powell

Bear → Panda @Matt Powell
#CycleGAN

Portrait to Dollface
@Mario Klingemann

Colorizing legacy photographs
@Mario Klingemann
StarGAN

Choi et al., “StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation”, CVPR 2018
Generating Training Data

Unlabeled Real Images

Synthetic

Refiner

Refined

Shrivastava et al., “Learning From Simulated and Unsupervised Images Through Adversarial Training”, CVPR 2017
Generating Training Data

Figure 1: (a) Examples from our novel Precarious Pedestrian Dataset of dangerous, but rare pedestrian scenes.

Huang and Ramanan, “Expecting the Unexpected: Training Detectors for Unusual Pedestrians With Adversarial Imposters”, CVPR
Generating Training Data

Varol et al., “Learning From Synthetic Humans”, CVPR 2017
Recognizing atypical objects

- Computer vision models cannot recognize objects in unusual modalities well

*Thomas and Kovashka, in submission to ECCV 2018*
Recognizing atypical objects

• Training a network with two source modalities helps it generalize to novel target modalities

• We can obtain good secondary modalities for free:
  • Automatic transformation of a photo to a cartoon/sketch (style transfer by Johnson et al. ECCV 2016)

Thomas and Kovashka, in submission to ECCV 2018
Recognizing atypical objects

• Train a network with additional modalities:
  • The outlines of objects (segmentation)
  • Use a convolutional neural network to transform a photo into a cartoon/sketch

Thomas and Kovashka, in submission to ECCV 2018
Recognizing atypical objects

- We outperform prior domain adaptation methods
  - Including a generous modification of Bousmalis et al. 2017 ~ our approach but with GAN instead of style transfer

<table>
<thead>
<tr>
<th>Method</th>
<th>PACS: Art, Cartoons, Sketches (Li 2017)</th>
<th>Sketchy Database (Sangkloy 2016)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Adaptation</td>
<td>0.388605333</td>
<td>0.093055</td>
</tr>
<tr>
<td>Ganin &amp; Lempitsky 2015</td>
<td>0.537944333</td>
<td>0.112329</td>
</tr>
<tr>
<td>Castrejon et al. 2016</td>
<td>0.400783</td>
<td>0.085107</td>
</tr>
<tr>
<td>Ghifary et al. 2016</td>
<td>0.462287276</td>
<td>0.22547</td>
</tr>
<tr>
<td>Long et al. 2016</td>
<td>0.566674</td>
<td>0.303163</td>
</tr>
<tr>
<td>Bousmalis et al. 2017 + domain confusion</td>
<td>0.547297333</td>
<td>0.284765</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>0.587813333</strong></td>
<td><strong>0.326453</strong></td>
</tr>
<tr>
<td>Supervised (upper bound)</td>
<td>0.904064333</td>
<td>0.822074</td>
</tr>
</tbody>
</table>

*Thomas and Kovashka, in submission to ECCV 2018*