GENERAL

- Last class.
- Project Reports Due Next Tuesday Night!
- Created a repository to submit project reports.
  - Include your report as report.pdf (keep this name consistent).
  - Don’t forget to git add, git commit, and git push.
  - Don’t include code, but hang on to it!
- Keys for PSET 5 can be picked up after next week. Will post location/times on piazza.

CONDITIONAL GANS

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Jun-Yan Zhu*  Taesung Park*  Phillip Isola  Alexei A. Efros
UC Berkeley
In ICCV 2017
**CONDITIONAL GANS**

Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Jun-Yan Zhu*  Taesung Park*  Phillip Isola  Alexei A. Efros

UC Berkeley

In ICCV 2017

* Indicates equal contribution

---

**ADVERSARIAL LOSS**

- Train with

\[
\begin{align*}
L(G) &= \|G(x) - y\|^2 - \lambda \log D(G(x)) \\
L(D) &= -\log(1 - D(G(x))) - \log D(y)
\end{align*}
\]

- The GAN loss is unconditional, but there is also a reconstruction loss.
- So the loss says, be close to the true answer, but make your output resemble natural images.

---

**VARIATIONAL AUTO-ENCODERS**

Regular Auto-encoder:

\[ X \rightarrow \text{Encoder} \rightarrow Z \rightarrow \text{Decoder} \rightarrow \hat{X} \]

Train with \( \|X - \hat{X}\|^2 \)

Variational Auto-encoder:

\[ X \rightarrow \text{Encoder} \rightarrow \mu, \Sigma \rightarrow \text{Sample} \rightarrow Z \rightarrow \text{Decoder} \rightarrow \hat{X} \]

Train with \( \|X - \hat{X}\|^2 \)
VARIATIONAL AUTO-ENCODERS

How do you back-propagate through sampling?

Variational Auto-encoder

\[ X \xrightarrow{\text{Encoder}} \mu, \Sigma \xrightarrow{\text{Sample}} Z \xrightarrow{\text{Decoder}} \hat{X} \]

Train with \[ ||X - \hat{X}||^2 + KL(\mathcal{N}(\mu, \Sigma) || \mathcal{N}(0, I)) \]

At Test Time,

\[ 0, I \xrightarrow{\text{Sample}} Z \xrightarrow{\text{Decoder}} \hat{X} \]

Train with \[ ||X - \hat{X}||^2 + KL(\mathcal{N}(\mu, \Sigma) || \mathcal{N}(0, I)) \]

VARIATIONAL AUTO-ENCODERS

CONDITIONAL VARIATIONAL AUTO-ENCODERS

- Variants that also propose an input based target distribution instead of \( \mathcal{N}(0, I) \)

Esser et al., A Variational U-Net for Conditional Appearance and Shape Generation
Kohl et al., A Probabilistic U-Net for Segmentation of Ambiguous Images
**CONDITIONAL VARIATIONAL AUTO-ENCODERS**

- Can often generate more diverse samples than GANs.

<table>
<thead>
<tr>
<th>GT</th>
<th>pix2pix[12]</th>
<th>our (reconst.)</th>
<th>our (random samples)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="shoes.png" alt="Shoes" /></td>
<td><img src="shoes_pix2pix.png" alt="Shoes" /></td>
<td><img src="shoes_reconst.png" alt="Shoes" /></td>
<td><img src="shoes_random.png" alt="Shoes" /></td>
</tr>
<tr>
<td><img src="purses.png" alt="Purses" /></td>
<td><img src="purses_pix2pix.png" alt="Purses" /></td>
<td><img src="purses_reconst.png" alt="Purses" /></td>
<td><img src="purses_random.png" alt="Purses" /></td>
</tr>
</tbody>
</table>

---

**UN-SUPERVISED LEARNING**

- For a lot of tasks, it is hard to collect enough training data.
- We saw for the stereo example, how you can have an indirect supervision.
- But in other cases, you have to use transfer learning.
  - Train a network on a large dataset for a related task for which you have ground truth.
  - Remove last layer, and use / finetune feature extractor for new task.
- Researchers are exploring tasks made specifically to transfer from.

---

**UN-SUPERVISED LEARNING**

- Pre-train by learning to add color

Larsson, Maire, Shakhnarovich, CVPR 2017.

---

**UN-SUPERVISED LEARNING**

- Pre-train by solving jigsaw puzzles

Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles

Melodi Nreouei and Paolo Favaro

Institute for Informatics
University of Bologna

[me@me.eu](mailto:me@me.eu) favaro@unibo.it
**UN-SUPERVISED LEARNING**

- Pre-train by predicting sound from video

**Visually Indicated Sounds**

Andrew Owens  Philip Isola  Josh McDermott  Antonio Torralba  Edward H. Adelson  William T. Freeman

(Examples of sounds from the Greatest Hits dataset. Click each image to play.)

**DOMAIN ADAPTATION**

- Generate synthetic training data using renderers.
- But networks trained on synthetic data need not generalize to real data.
- (In fact, they may not transfer from high-quality Flickr data to cell-phone camera data)

**Problem Setting**

- Have input-output training pairs of $(x', y)$ from source domain: renderings/high-quality images/...
- Have only inputs $x$ from target domain: where we actually want to use this.
- Train a network so that features computed from $x'$ and $x$ have the same distribution ...
  i.e., use GANs!

---

**DOMAIN ADAPTATION**

- Adversarial Discriminative Domain Adaptation
  - Source images + labels
  - Class label

**Adversarial Adaptation**

- Source images
- Target images
- Discriminator

**Testing**

- train image
- CAN
- target image
- class label

---

**DOMAIN ADAPTATION**

- Digits adaptation
  - MNIST
  - USPS
  - SVHN

- Cross-modality adaptation (NYUD)
  - RGB
  - IRLA
That's all folks!

- We’ve covered what forms the foundations of state-of-the-art vision algorithms
  - Will help you read, understand, and implement vision papers
- But things are changing rapidly: not just new solutions, but new problems
- So keep reading! (we’ll cover some of this in 659A)

We hope that you have...

- An understanding of the basic math and programming tools to approach vision problems
- Are as surprised as we are that humans and animals are able to solve this so easily

Remaining Time

Ending class early: please use this time to fill out course evaluations.