Conditional Generative Adversarial Networks (cGANs)
Conditional GANs (cGANs)

- Gain control of output

- Modeling (e.g., sketch-based modeling, etc.)
  - Add semantic meaning to latent space manifold

- Domain transfer
  - Labels on A -> transfer to B, train network on ‘B’, test on B
  - More later
GAN Manifold

Train Data

Sampled Data -> G(z)
GAN Manifold
GAN Manifold

Linear interpolation in z space: $G(z_0 + t \cdot (z_1 - z_0))$

$G(z_0)$ $G(z_1)$
Conditional GANs (cGANs)
iGANs: Overview

original photo

projection on manifold

Editing UI

transition between the original and edited projection

different degree of image manipulation

Project

Edit Transfer

Slide credit Zhu / [Zhu et al. 16]
iGANs: Overview

- Original photo
- Projection on manifold
- Transition between the original and edited projection
- Different degree of image manipulation

Slide credit Zhu / [Zhu et al. 16]
iGANs: Projecting an Image onto the Manifold

Input: real image $x^R$
Output: latent vector $z$

Optimization

$$z^* = \arg\min_z L(G(z), x^R)$$

Reconstruction loss $L$
Generative model $G(z)$

Slide credit Zhu / [Zhu et al. 16]
iGANs: Projecting an Image onto the Manifold

Input: real image $x^R$
Output: latent vector $z$

Optimization
$$z^* = \arg \min \mathcal{L}(G(z), x^R)$$

Inverting Network $z = P(x)$
$$\theta_P^* = \arg \min_{\theta_P} \sum_{x_n^R} \mathcal{L}(G(P(x^R; \theta_P)), x^R)$$

Auto-encoder with a fixed decoder $G$

Slide credit Zhu / [Zhu et al. 16]
iGANs: Projecting an Image onto the Manifold

Input: real image $x^R$
Output: latent vector $z$

**Optimization**

$$z^* = \arg\min \mathcal{L}(G(z), x^R)$$

**Inverting Network** $z = P(x)$

$$\theta^*_P = \arg\min_{\theta_P} \sum_{x^R_n} \mathcal{L}(G(P(x^R_n; \theta_P)), x^R)$$

**Hybrid Method**

Use the network as initialization for the optimization problem

Slide credit Zhu / [Zhu et al. 16]
iGANs: Overview

original photo

projection on manifold

不同程度的图像操纵

过渡到原始和编辑后的投影

Slide credit Zhu / [Zhu et al. 16]
iGANs: Manipulating the Latent Vector

Objective: \[ z^* = \arg \min_{z \in \mathbb{Z}} \left\{ \sum_g (L_g(G(z)) v_g + \lambda_s \cdot \|z - z_0\|_2^2 \right\} \]

- constraint violation loss \( L_g \)
- user guidance image
- guidance \( v_g \)
- data term
- manifold smoothness
- \( G(z) \)
- \( z_0 \)

Slide credit Zhu / [Zhu et al. 16]
iGANs: Overview

- Original photo
- Projection on manifold
- Editing UI
  - Transition between the original and edited projection
- Different degree of image manipulation

Slide credit Zhu / [Zhu et al. 16]
iGANs: Edit Transfer

Motion \((u, v)+\) Color \((A_{3 \times 4})\): estimate per-pixel geometric and color variation

\[
\int \int \left| I(x, y, t) - A \cdot I(x+u, y+v, t+1) \right|^2 + \sigma_s (\|\nabla u\|^2 + \|\nabla v\|^2) + \sigma_c \|\nabla A\|^2 dxdy
\]

Linear Interpolation in \(z\) space

Slide credit Zhu / [Zhu et al. 16]
Motion \((u, v)\) + Color \((A_{3 \times 4})\): estimate per-pixel geometric and color variation

\[
\int \int \left( \| I(x, y, t) - A \cdot I(x+u, y+v, t+1) \|^2 + \sigma_s (\| \nabla u \|^2 + \| \nabla v \|^2) + \sigma_c \| \nabla A \|^2 \right) dx dy
\]

\(G(z_0)\)

Linear Interpolation in \(z\) space

\(G(z_1)\)

Slide credit Zhu / [Zhu et al. 16]
iGANs: Edit Transfer

Motion \((u, v)\) + Color \((A_{3\times4})\): estimate per-pixel geometric and color variation

\[
\int \int \| I(x, y, t) - A \cdot I(x+u, y+v, t+1) \|^2 + \sigma_s (\| \nabla u \|^2 + \| \nabla v \|^2) + \sigma_c \| \nabla A \|^2 dxdy
\]

- **data term**
- **spatial reg**
- **color reg**

\[G(z_0)\]
\[G(z_1)\]

Input
Result

Linear Interpolation in \(z\) space
cGANs: Interactive GANs

Interactive GANs: projection to GAN embedding

[https://github.com/junyanz/iGAN](https://github.com/junyanz/iGAN) [Zhu et al. 16.]
cGANs: Interactive GANs

https://github.com/junyanz/iGAN [Zhu et al. 16.]
cGANs: Interactive GANs

https://github.com/junyanz/iGAN [Zhu et al. 16.]
Mapping in Latent Space is Difficult!

- Semantics are missing
- In most cases, no labels available
- Ideally, need some unsupervised disentangled rep.

(a) Azimuth (pose)  
(b) Presence or absence of glasses

InfoGAN [Chen et al. 16]
Paired vs Unpaired Setting

Paired

\[ x_i, y_i \]

\[ \{ \}, \{ \}, \{ \}, \{ \}, \{ \} \]

Unpaired

\[ X, Y \]

\[ \{ \}, \{ \}, \{ \}, \{ \}, \{ \} \]
pix2pix: Image-to-Image Translation

slides credit: Isola / Zhu
\[
\min_G \max_D \mathbb{E}_{z,x} [\log D(G(z)) + \log(1 - D(x))] 
\]

slides credit: Isola / Zhu

[Goodfellow et al. 2014]
\[
\min_G \max_D \mathbb{E}_{x,y} [\log D(G(x)) + \log(1 - D(y))] 
\]
\[
\min_G \max_D \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right]
\]
\[
\min_G \max_D \mathbb{E}_{x,y} [\log D(G(x)) + \log(1 - D(y))] 
\]
\[
\min_{G} \max_{D} \mathbb{E}_{x,y} \left[ \log D(x, G(x)) + \log(1 - D(x, y)) \right]
\]

match joint distribution \( p(G(x), y) \sim p(x, y) \)

slides credit: Isola / Zhu
Edges $\rightarrow$ Images

Edges from [Xie & Tu, 2015]

slides credit: Isola / Zhu
pix2pix: Paired Setting

• Great when we have ‘free’ training data

• Often called self-supervised

• Think about these settings 😊
Sketches → Images

Trained on Edges → Images

Data from [Eitz, Hays, Alexa, 2012]

slides credit: Isola / Zhu
#edges2cats

[Christopher Hesse]

- @gods_tail

- Ivy Tasi @ivymyt

- @matthematician

Vitaly Vidmirov @vvid

slides credit: Isola / Zhu

https://affinelayer.com/pixsrv/
Data from maps.google.com
slides credit: Isola / Zhu
BW → Color

Data from [Russakovsky et al. 2015]

slides credit: Isola / Zhu
Ideas behind Pix2Pix

- $L = L_{GAN} + \lambda L_1$ (makes it more constraint)

- Unet / skip connections for preserving structure

- Noise only through dropout
  - cGANs tend to learn to ignore the random vector $z$
  - Still want probabilistic model
Ideas behind Pix2Pix

• L1 or L2 loss for low frequency details

• GAN discriminator for high frequency details

-> PatchGAN
  – GAN discriminator applied only to local patches
  – It’s fully-convolutional; i.e., can run on arbitrary image sizes
Pix2PixHD

- Expand the pix2pix idea to multi-scale

- Coarse-to-fine generator + discriminator

- G's and D's are the same but since they operate on different resolutions, they have effectively a larger receptive field

[Wang et al. 18]
Pix2PixHD

2x downsampling

Residual blocks

[Prof. Leal-Taixé and Prof. Niessner]

[Wang et al. 18]
Pix2PixHD

• Use of multi-scale discriminators

\[
\min_G \max_{D_1,D_2,D_3} \sum_{k=1,2,3} L_{GAN} (G, D_k)
\]

• Can make various combinations of stacking discriminator and generator
  – E.g., have a single G and downsample generated and real images – or have intermediate real images (cf. ProGAN)

[Wang et al. 18]
Pix2PixHD

Input labels

Synthesized image

[Wang et al. 18]
Pix2PixHD
Pix2PixHD (interactive results)
Paired

$\{x_i\}, \{y_i\}$

Label $\leftrightarrow$ photo: per-pixel labeling

Horse $\leftrightarrow$ zebra: how to get zebras?
- Expensive to collect pairs.
- Impossible in many scenarios.

slides credit: Isola / Zhu
Paired

\[ x_i, y_i \]

\[
\begin{Bmatrix}
\text{shoe}, \text{brown shoe}, \text{black boot}, \text{white shoe}
\end{Bmatrix}
\]

Unpaired

\[ X, Y \]

\[
\begin{Bmatrix}
\text{horse}, \text{zebra}, \text{horse}, \text{zebra}
\end{Bmatrix}
\]

slides credit: Isola / Zhu
No input-output pairs!

slides credit: Isola / Zhu
Discriminator

\[ x \xrightarrow{G} G(x) \xrightarrow{\mathcal{D}} \text{Real!} \]

Generator

slides credit: Isola / Zhu
GANs doesn’t force output to correspond to input
mode collapse!

slides credit: Isola / Zhu
Cycle-Consistent Adversarial Networks

\[ X \xrightarrow{D_Y} Y \]

slides credit: Isola / Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]
Cycle-Consistent Adversarial Networks

[Mark Twain, 1903]

[slides credit: Isola / Zhu]

[Zhu*, Park*, Isola, and Efros, ICCV 2017]
Cycle Consistency Loss

\[ \text{Cycle Consistency Loss} = D_Y(G(x)) + \|F(G(x)) - x\|_1 \]

\[ G(x) \rightarrow \hat{Y} \rightarrow F(G(x)) \rightarrow \hat{x} \]

Reconstruction error

\[ \|F(G(x)) - x\|_1 \]

slides credit: Isola / Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]
Cycle Consistency Loss

\[
\begin{align*}
    &x & & \xrightarrow{G} & \hat{y} & & \xrightarrow{F} & \hat{x} \\
    \text{Reconstruction error} & & & & & & & &
    \|F(G(x)) - x\|_1
\end{align*}
\]

[Zhu*, Park*, Isola, and Efros, ICCV 2017]

slides credit: Isola / Zhu
Cycle Consistency Loss

\[ D_Y(G(x)) \]

\[ D_G(F(x)) \]

Reconstruction error

\[ \|F(G(x)) - x\|_1 \]

\[ \|G(F(y)) - y\|_1 \]

slides credit: Isola / Zhu

[Zhu*, Park*, Isola, and Efros, ICCV 2017]
Cycle GAN - Overview

https://junyanz.github.io/CycleGAN/ [Zhu et al. 17]
Monet's paintings → photos

slides credit: Isola / Zhu
Next Lectures

• Next Monday 23rd,
  – Xmas GANs
  – No Lecture

• Next Lecture -> Jan 13th

• Keep working on the projects!
See you next year 😊

Input (Dataset of Tree's) -> Computationally Expensive Math -> Output (Fattest Path)