GAN and Feature Representation

Hung-yi Lee
Outline

• GAN+Autoencoder

• InfoGAN

• BiGAN
GAN + Autoencoder
Photo Editing

Photo Editing

We can tune $z$ to edit image $x$

How to modify a specific attribute?

https://www.youtube.com/watch?v=kPEIJJsQr7U
GAN+Autoencoder

• We have a generator (input $z$, output $x$)
• However, given $x$, how can we find $z$?
  • Learn an encoder (input $x$, output $z$)

**Diagram:**
- Encoder: Input $x$, Output $z$
- Generator (Decoder): Input $z$, Output $x$
- Discriminator: Different structures?

**Note:**
- The encoder is initialized ($\text{init}$) and the generator is fixed ($\text{fixed}$).
- The goal is to make the output as close as possible to the input.
as close as possible
# Attribute Representation

### CelebA

<table>
<thead>
<tr>
<th>Image</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="female_image.png" alt="Female Image" /></td>
<td>Arched eyebrows, attractive, brown hair, heavy makeup, high cheekbones, mouth slightly open, no beard, pointy nose, smiling, straight hair, wearing earrings, wearing lipstick, young.</td>
</tr>
<tr>
<td><img src="male_image.png" alt="Male Image" /></td>
<td>5 o’clock shadows, attractive, bags under eyes, big lips, big nose, black hair, bushy eyebrows, male, no beard, pointy nose, straight hair, young.</td>
</tr>
</tbody>
</table>

\[
Z_{male} = \frac{1}{N_1} \sum_{x \in male} En(x) - \frac{1}{N_2} \sum_{x' \notin male} En(x')
\]

**Female image**

\[ x \rightarrow En(x) + Z_{male} = z' \rightarrow Gen(z') \]

**Male image**
Find the Attributes

\[ z \cdot \frac{z_{\text{male}}}{\|z_{\text{male}}\|} = 0.76 \]
Basic Idea

space of z
Back to $z$

- **Method 1**
  
  $z^* = \arg \min_z L(G(z), x^T)$

  Difference between $G(z)$ and $x^T$
  
  - Pixel-wise
  - By another network

  Gradient Descent

- **Method 2**

  as close as possible

- **Method 3**

  Using the results from **method 2** as the initialization of **method 1**
Back to z - Results
Editing Photos

- $z_0$ is the code of the input image

$$z^* = \underset{z}{\text{arg min}} \ U(G(z)) + \lambda_1 \|z - z_0\|^2 - \lambda_2 D(G(z))$$

Not too far away from the original image

Using discriminator to check the image is realistic or not

Does it fulfill the constraint of editing?
Editing Photos - Results

(a) User constraints $v_g$ at different update steps

(b) Updated images according to user edits

(c) Linear interpolation between $G(z_0)$ and $G(z_1)$
Final System

$G(z_0)$  \quad \text{Linear interpolation between } G(z_0) \text{ and } G(z_1)  \quad G(z_1)

User Edits

Original \quad \text{Edit transfer sequence on the original photo} \quad \text{Result}

$G(z_0)$  \quad \text{Linear interpolation between } G(z_0) \text{ and } G(z_1)  \quad G(z_1)

User Edits

Original \quad \text{Edit transfer sequence on the original photo} \quad \text{Result}
VAE-GAN

- Minimize reconstruction error
- $z$ close to normal

- Minimize reconstruction error
- Cheat discriminator

- Discriminate real, generated and reconstructed images


Discriminator provides the similarity measure
Algorithm

• Initialize En, De, Dis
• In each iteration:
  • Sample M images $x^1, x^2, \cdots, x^M$ from database
  • Generate M codes $\tilde{z}^1, \tilde{z}^2, \cdots, \tilde{z}^M$ from encoder
    • $\tilde{z}^i = En(x^i)$
  • Generate M images $\tilde{x}^1, \tilde{x}^2, \cdots, \tilde{x}^M$ from decoder
    • $\tilde{x}^i = En(\tilde{z}^i)$
  • Sample M codes $z^1, z^2, \cdots, z^M$ from prior $P(z)$
  • Generate M images $\hat{x}^1, \hat{x}^2, \cdots, \hat{x}^M$ from decoder
    • $\hat{x}^i = En(z^i)$
  • Update En to decrease $\|\tilde{x}^i - x^i\|$, decrease $KL(P(\tilde{z}^i | x^i) || P(z))$
  • Update De to decrease $\|\tilde{x}^i - x^i\|$, increase $Dis(\tilde{x}^i)$ and $Dis(\hat{x}^i)$
  • Update Dis to increase $Dis(x^i)$, decrease $Dis(\tilde{x}^i)$ and $Dis(\hat{x}^i)$

Another kind of discriminator:

Discriminator

real  gen  recon  
\[\downarrow\  \uparrow\  \uparrow\  \uparrow\]
\[\downarrow\]
\[x\]
VAE+GAN - Sample

VAE

VAE_{DisI}

VAE/GAN

GAN

blurry

sharp

sharp
<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>VAE</th>
<th>VAE_{Dist}</th>
<th>VAE/GAN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output</strong></td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

GAN cannot do reconstruction
InfoGAN
What is InfoGAN?

"Auto-encoder"

\[ z = z' + c \]

Predict the code \( c \) that generates \( x \)

Parameter sharing (only the last layer is different)

Discriminator

Scalar
Motivation

- **c** must have clear influence on **x**, so the encoder can recover **c** from **x**
- **c** will be easy to interpret

A specific dimension **c_i** cannot cooperate with other feature dimensions to have influence.
(a) Varying $c_1$ on InfoGAN (Digit type)  
(b) Varying $c_1$ on regular GAN (No clear meaning)  
(c) Varying $c_2$ from $-2$ to $2$ on InfoGAN (Rotation)  
(d) Varying $c_3$ from $-2$ to $2$ on InfoGAN (Width)
BiGAN

Jeff Donahue, Philipp Krähenbühl, Trevor Darrell, “Adversarial Feature Learning”, ICLR, 2017

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, Aaron Courville, “Adversarially Learned Inference”, ICLR, 2017
BiGAN

Encoder

Decoder

Discriminator

Image x
(real)

code z

Image x
(generated)

(from prior distribution)

code z

to encoder or decoder?

Image x
(code z)
Algorithm

• Initialize encoder $E_n$, decoder $D_e$, discriminator $D_{is}$
• In each iteration:
  • Sample $M$ images $x^1, x^2, \ldots, x^M$ from database
  • Generate $M$ codes $\tilde{z}^1, \tilde{z}^2, \ldots, \tilde{z}^M$ from encoder
    • $\tilde{z}^i = E_n(x^i)$
  • Sample $M$ codes $z^1, z^2, \ldots, z^M$ from prior $P(z)$
  • Generate $M$ codes $\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^M$ from decoder
    • $\tilde{x}^i = D_e(z^i)$
  • Update $D_{is}$ to increase $D_{is}(x^i, \tilde{z}^i)$, decrease $D_{is}(\tilde{x}^i, z^i)$
  • Update $E_n$ and $D_e$ to decrease $D_{is}(x^i, \tilde{z}^i)$, increase $D_{is}(\tilde{x}^i, z^i)$
 encoder

Decoder

Discriminator

Image x

code z

(from prior distribution)

Image x

code z

from encoder

or decoder?

Encoder

Decoder

Discriminator

Image x

code z

Evaluate the difference between P and Q

To make P and Q the same

Optimal encoder and decoder:

\[
\begin{align*}
\text{En}(x') &= z' \\
\text{De}(z') &= x' \\
\text{De}(z'') &= x'' \\
\text{En}(x'') &= z''
\end{align*}
\]

For all \( x' \)

For all \( z'' \)
BiGAN

Optimal encoder and decoder:

- $\text{En}(x') = z'$ \quad $\text{De}(z') = x'$ \quad \text{For all } x'$
- $\text{De}(z'') = x''$ \quad $\text{En}(x'') = z''$ \quad \text{For all } z''

How about?
Concluding Remarks

- GAN+Autoencoder
- InfoGAN
- BiGAN
Next Time:
Energy-based GAN
Original Idea

- Discriminator leads the generator

![Diagram showing the relationship between discriminator and generator distributions]
Original Idea

• When the data distribution and generated distribution is the same.
• The output of discriminator will be flat everywhere.
• However, discriminator is often used in pre-training.
  • It contains useful information.
• We always use the discriminator obtained in the last iteration as the initialization of the next step.
Energy-based Model

• We want to find an evaluation function $F(x)$
  • Input: object $x$ (e.g. images), output: scalar (how good $x$ is)
  • Real $x$ has high $F(x)$
  • $F(x)$ can be a network
• We can find good $x$ by $F(x)$:
  • Generate $x$ with large $F(x)$

• How to find $F(x)$?

![Diagram showing the evaluation function process]
Energy-based GAN

- We want to find an evaluation function \( F(x) \)
- How to find \( F(x) \)?

In the end ......
Energy-based Model

• Preview: Framework of structured learning (Energy-based Model)
  • ML Lecture 21: Structured Learning - Introduction
    • https://www.youtube.com/watch?v=5OYu0vxXEv8
  • ML Lecture 22: Structured Learning - Linear Model
    • https://www.youtube.com/watch?v=HfPw40JPays
  • ML Lecture 23: Structured Learning - Structured SVM
    • https://www.youtube.com/watch?v=YjvGVVrCrhQ
  • ML Lecture 24: Structured Learning - Sequence Labeling
    • https://www.youtube.com/watch?v=o9FPSqobMys
• Graphical model & Gibbs sampling