Generation

Hung-yi Lee 李宏毅
Network as Generator

Simple Distribution $\rightarrow$ Generator $\rightarrow$ Complex Distribution

We know its formulation, so we can sample from it.
Why distribution?

Video Prediction

Real Video

Previous frames

Network

next frame

Source: https://github.com/dyelax/Adversarial_Video_Generation
Why distribution?

**Video Prediction**

Prediction

Network

Previous frames

Source: https://github.com/dyelax/Adversarial_Video_Generation
Why distribution?

Video Prediction

Prediction

Previous frames

Network

Simple Distribution

Source: https://github.com/dyelax/Adversarial_Video_Generation
Why distribution?

(The same input has different outputs.)

• Especially for the tasks needs “creativity”

**Drawing**

Character with red eyes → Network

**Chatbot**

你知道輝夜是誰嗎？→ Network

她是秀知院學生會 ... 她開創了忍者時代 ...
Generative Adversarial Network (GAN)
GAN

• How to pronounce “GAN”?
All Kinds of GAN ...

https://github.com/hindupuravinash/the-gan-zoo

- SeUDA - Semantic-Aware Generative Adversarial Nets for Unsupervised Domain Adaptation and Segmentation
- SG-GAN - Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption (github)
- SG-GAN - Sparsely Grouped Multi-task Generative Adversarial Networks for Facial Attractiveness
- SGAN - Texture Synthesis with Spatial Generative Adversarial Networks
- SGAN - Stacked Generative Adversarial Networks (github)
- SGAN - Steganographic Generative Adversarial Networks
- SGAN - SGAN: An Alternative Training of Generative Adversarial Networks
- SGAN - CT Image Enhancement Using Stacked Generative Adversarial Networks and Task-Specific Segmentation Improvement
- sGAN - Generative Adversarial Training for MRA Image Synthesis Using Multi-Contrast

Anime Face Generation

• Unconditional generation

\[
\begin{align*}
0.3 & 0.1 & -0.3 \\
-0.1 & -0.1 & 0.1 \\
\vdots & \vdots & \vdots \\
-0.7 & 0.7 & 0.9
\end{align*}
\]

Low-dim vector

\[
\begin{align*}
\mathbf{x}
\end{align*}
\]

Generator

\[
\begin{align*}
\mathbf{z}
\end{align*}
\]

Normal Distribution

high-dim vector

Complex Distribution

\[
\begin{align*}
\mathbf{y}
\end{align*}
\]
It is a neural network (that is, a function).

Scalar: Larger means real, smaller value fake.
Basic Idea of GAN

Butterflies are not brown

Butterflies do not have veins

Generator

...........

Discriminator
Basic Idea of GAN

This is where the term “adversarial” comes from.
Basic Idea of GAN

- 寫作敵人，唸做朋友
Algorithm

- Initialize generator and discriminator
- In each training iteration:

**Step 1**: Fix generator G, and update discriminator D

Discriminator learns to assign high scores to real objects and low scores to generated objects.
**Algorithm**

- Initialize generator and discriminator

- In each training iteration:

**Step 2**: Fix discriminator $D$, and update generator $G$

Generator learns to “fool” the discriminator

**Diagram**:
- Large network

- Neural Network (NN)
- Generator update
- Hidden layer
- Discriminator fix
- 0.13
Algorithm

- Initialize generator and discriminator
- In each training iteration:

Sample some real objects:
Generate some fake objects:

Learning $D$

Learning $G$

Update $D$

$1$ $1$ $1$ $1$ $1$

$0$ $0$ $0$ $0$ $0$

$\text{G}$

$\text{D}$
Anime Face Generation

Source of training data: https://zhuanlan.zhihu.com/p/24767059
Anime Face Generation

1000 updates
Anime Face Generation

2000 updates
Anime Face Generation

5000 updates
Anime Face Generation

10,000 updates
Anime Face Generation

20,000 updates
Anime Face Generation

50,000 updates
The faces generated by machine.
In 2019, with StyleGAN ......

Source of video:
https://www.gwern.net/Faces
Progressive GAN

https://arxiv.org/abs/1710.10196
The first GAN

Today ...... BigGAN

https://arxiv.org/abs/1809.11096
Theory behind GAN
Our Objective

Normal Distribution

$G^* = \underset{G}{\arg \min} \text{Div}(P_G, P_{data})$

Divergence between distributions $P_G$ and $P_{data}$

How to compute the divergence?

c.f. \[ w^*, b^* = \arg \min_{w,b} L \]
Sampling is good enough ......

\[
G^* = \arg \min_G \text{Div}(P_G, P_{data})
\]

Although we do not know the distributions of \(P_G\) and \(P_{data}\), we can sample from them.
Discriminator: $G^* = \arg\min_G \text{Div}(P_G, P_{data})$

$\star$: data sampled from $P_{data}$  $\star\star$: data sampled from $P_G$

Training: $D^* = \arg\max_D V(D, G)$

The value is related to JS divergence.

Objective Function for $D$

$$V(G, D) = E_{y \sim P_{data}}[\log D(y)] + E_{y \sim P_G}[\log(1 - D(y))]$$

$D^* = \arg\max_D V(D, G)$

negative cross entropy

Training classifier: minimize cross entropy
Discriminator

\[ G^* = \arg \min_G Div(P_G, P_{data}) \]

\[ D^* = \arg \max_D V(D, G) \]

Training:

- small divergence: data sampled from \( P_{data} \)
- large divergence: data sampled from \( P_G \)

- hard to discriminate
- easy to discriminate
\[ G^* = \arg \min_G \max_D V(G, D) \]

\[ D^* = \arg \max_D V(D, G) \]

The maximum objective value is related to JS divergence.

- Initialize generator and discriminator
- In each training iteration:
  - **Step 1**: Fix generator \( G \), and update discriminator \( D \)
  - **Step 2**: Fix discriminator \( D \), and update generator \( G \)
### Can we use other divergence?

<table>
<thead>
<tr>
<th>Name</th>
<th>$D_f (P \parallel Q)$</th>
<th>Generator $f(u)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total variation</td>
<td>$\frac{1}{2} \int</td>
<td>p(x) - q(x)</td>
</tr>
<tr>
<td>Kullback-Leibler</td>
<td>$\int p(x) \log \frac{p(x)}{q(x)} , dx$</td>
<td>$u \log u$</td>
</tr>
<tr>
<td>Reverse Kullback-Leibler</td>
<td>$\int q(x) \log \frac{q(x)}{p(x)} , dx$</td>
<td>$- \log u$</td>
</tr>
<tr>
<td>Pearson $\chi^2$</td>
<td>$\int \frac{(q(x) - p(x))^2}{p(x)} , dx$</td>
<td>$(u - 1)^2$</td>
</tr>
<tr>
<td>Neyman $\chi^2$</td>
<td>$\int \frac{(p(x) - q(x))^2}{q(x)} , dx$</td>
<td>$\frac{(1-u)^2}{u}$</td>
</tr>
<tr>
<td>Squared Hellinger</td>
<td>$\int \left( \sqrt{p(x)} - \sqrt{q(x)} \right)^2 , dx$</td>
<td>$(\sqrt{u} - 1)^2$</td>
</tr>
<tr>
<td>Jeffrey</td>
<td>$\int (p(x) - q(x)) \log \left( \frac{p(x)}{q(x)} \right) , dx$</td>
<td>$(u - 1) \log u$</td>
</tr>
<tr>
<td>Jensen-Shannon</td>
<td>$\frac{1}{2} \int p(x) \log \frac{2p(x)}{p(x) + q(x)} + q(x) \log \frac{2q(x)}{p(x) + q(x)} , dx$</td>
<td>$-(u + 1) \log \frac{1+u}{2} + u \log u$</td>
</tr>
<tr>
<td>Jensen-Shannon-weighted</td>
<td>$\int p(x) \pi \log \frac{p(x)(1-\pi)q(x) + (1-\pi)q(x)}{p(x) + (1-\pi)q(x)} , dx$</td>
<td>$\pi u \log u - (1 - \pi + \pi u) \log(1 - \pi + \pi u)$</td>
</tr>
<tr>
<td>GAN</td>
<td>$\int p(x) \log \frac{2p(x)}{p(x) + q(x)} + q(x) \log \frac{2q(x)}{p(x) + q(x)} , dx , dx - \log(4)$</td>
<td>$u \log u - (u + 1) \log(u + 1)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Conjugate $f^*(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total variation</td>
<td>$t$</td>
</tr>
<tr>
<td>Kullback-Leibler (KL)</td>
<td>$\exp(t - 1)$</td>
</tr>
<tr>
<td>Reverse KL</td>
<td>$-1 - \log(-t)$</td>
</tr>
<tr>
<td>Pearson $\chi^2$</td>
<td>$\frac{1}{4} t^2 + t$</td>
</tr>
<tr>
<td>Neyman $\chi^2$</td>
<td>$2 - 2\sqrt{1 - t}$</td>
</tr>
<tr>
<td>Squared Hellinger</td>
<td>$\frac{t}{1-t}$</td>
</tr>
<tr>
<td>Jeffrey</td>
<td>$W(e^{1-t}) + \frac{1}{W(e^{1-t})} + t - 2$</td>
</tr>
<tr>
<td>Jensen-Shannon</td>
<td>$- \log(2 - \exp(t))$</td>
</tr>
<tr>
<td>Jensen-Shannon-weighted</td>
<td>$(1 - \pi) \log \frac{1-\pi}{1-\pi e^{t/\pi}}$</td>
</tr>
<tr>
<td>GAN</td>
<td>$- \log(1 - \exp(t))$</td>
</tr>
</tbody>
</table>

Using the divergence you like 😊

https://arxiv.org/abs/1606.00709
GAN is difficult to train ......

NO PAIN
NO GA N

(I found this joke from 陳柏文’s facebook.)
Tips for GAN
JS divergence is not suitable

• In most cases, $P_G$ and $P_{data}$ are not overlapped.

• 1. The nature of data

  Both $P_{data}$ and $P_G$ are low-dim manifold in high-dim space.
  
  The overlap can be ignored.

• 2. Sampling

  Even though $P_{data}$ and $P_G$ have overlap.
  
  If you do not have enough sampling ......
What is the problem of JS divergence?

JS divergence is always log2 if two distributions do not overlap.

\[
JS(P_{G_0}, P_{data}) = \log 2 \quad JS(P_{G_1}, P_{data}) = \log 2 \quad \cdots \quad JS(P_{G_{100}}, P_{data}) = 0
\]

Intuition: If two distributions do not overlap, binary classifier achieves 100% accuracy.

The accuracy (or loss) means nothing during GAN training.
Wasserstein distance

• Considering one distribution $P$ as a pile of earth, and another distribution $Q$ as the target
• The average distance the earth mover has to move the earth.

$$W(P, Q) = d$$
Wasserstein distance

There are many possible “moving plans”.
Using the “moving plan” with the smallest average distance to define the Wasserstein distance.

Source of image: https://vincentherrmann.github.io/blog/wasserstein/
**What is the problem of JS divergence?**

\[
\begin{align*}
JS(P_{G_0}, P_{data}) &= \log 2 \\
W(P_{G_0}, P_{data}) &= d_0
\end{align*}
\]

\[
\begin{align*}
JS(P_{G_1}, P_{data}) &= \log 2 \\
W(P_{G_1}, P_{data}) &= d_1
\end{align*}
\]

\[
\begin{align*}
&\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ quasi
What is the problem of JS divergence?

$P_{G_0} \iff d_0 P_{data} \iff P_{G_1} \iff d_1 P_{data} \iff \ldots \iff P_{G_{100}} \iff P_{data}$

pigment spot (limpet, *Patella*)

Complex eye (octopus)

https://www.pnas.org/content/104/suppl_1/8567.figures-only
WGANG

Evaluate Wasserstein distance between $P_{data}$ and $P_G$

$$\max_{D \in 1-Lipschitz} \{ E_{y \sim P_{data}}[D(y)] - E_{y \sim P_G}[D(y)] \}$$

D has to be smooth enough. How to fulfill this constraint?

Without the constraint, the training of D will not converge.

Keeping the D smooth forces $D(y)$ become $\infty$ and $-\infty$
\[
\max_{D \in \mathcal{1-Lipschitz}} \left\{ E_{y \sim P_{\text{data}}} [D(y)] - E_{y \sim P_{G}} [D(y)] \right\}
\]

• Original WGAN → Weight
  Force the parameters \( w \) between \( c \) and \(-c\)
  After parameter update, if \( w > c \), \( w = c \); if \( w < -c \), \( w = -c \)

• Improved WGAN → Gradient Penalty
  Keep the gradient close to 1
  
  [Diagram with samples and parameter update]

• Spectral Normalization → Keep gradient norm smaller than 1 everywhere
  
  [Diagram with spectral normalization effect]

https://arxiv.org/abs/1704.00028
https://arxiv.org/abs/1802.05957
GAN is still challenging ...

- Generator and Discriminator needs to match each other (棋逢敌手)

  Generate fake images to fool discriminator

  Cannot fool the discriminator ...

  Fail to improve ...

  Tell the difference between real and fake

  Fail to improve ...

  I cannot tell the difference ......
More Tips

• Tips from Soumith
  • https://github.com/soumith/ganhnacks

• Tips in DCGAN: Guideline for network architecture design for image generation
  • https://arxiv.org/abs/1511.06434

• Improved techniques for training GANs
  • https://arxiv.org/abs/1606.03498

• Tips from BigGAN
  • https://arxiv.org/abs/1809.11096
GAN for Sequence Generation

Non-differentiable ...

Discriminator

Generator

max or sample

update

Decoder

unchanged

score

unchanged

Non-differentiable ...
**GAN for Sequence Generation**

Reinforcement learning (RL) is involved ......

RL is difficult to train  \quad GAN is difficult to train

Sequence Generation GAN (RL+GAN)
GAN for Sequence Generation

- Usually, the generator are fine-tuned from a model learned by other approaches.
- However, with enough hyperparameter-tuning and tips, ScarchGAN can train from scratch.

Training language GANs from Scratch

Generative Models

• This lecture: Generative Adversarial Network (GAN)

Full version

https://www.youtube.com/playlist?list=PLJV_el3uVTsMq6JEFPW35BCiOQTsoqwNw
More Generative Models

Variational Autoencoder (VAE)  
https://youtu.be/8zomhgKrsmQ

FLOW-based Model  
https://youtu.be/uXY18nzdSsM
Possible Solution?

Using typical learning approaches?

Conditional Generation
Text-to-image

Generator

\( x \)
- red eyes
- black hair

\( y \)
- yellow hair
- dark circles

\( z \)
- red hair, green eyes
- blue hair, red eyes
Conditional GAN

\[ x: \text{Red eyes} \rightarrow G \rightarrow \text{Image} \]

\[ y = G(c, z) \]

\[ y \text{ is real image or not} \]

\[ y \rightarrow D \quad \text{(original)} \rightarrow \text{scalar} \]

Generator will learn to generate realistic images .... But completely ignore the input conditions.

Real images:

1

Generated images:

0
Conditional GAN

$x$: Red eyes

Normal distribution $z$

$\mathbf{D}$ (better)

$\mathbf{G}$

Image

$y = G(c, z)$

$y$ is realistic or not + $x$ and $y$ are matched or not

True text-image pairs:

(red eyes, ) 1

(red eyes, ) 0

(red eyes, ) 0

Image
Conditional GAN

\[ y = G(c, z) \]

Image translation, or \textit{pix2pix}
Conditional GAN

https://arxiv.org/abs/1611.07004

Testing:

input  supervised  GAN  GAN + supervised
Conditional GAN

\[ x: \text{sound} \rightarrow \mathbf{G} \rightarrow \text{Image} \]

"a dog barking sound"

**Training Data Collection**

- video

Conditional GAN

- Sound-to-image

The images are generated by Chia-Hung Wan and Shun-Po Chuang.

https://wjohn1483.github.io/audio_to_scene/index.html
Conditional GAN
Talking Head Generation

Conditional GAN

Multi-label Image Classifier = Conditional Generator

https://arxiv.org/abs/1811.04689
Learning from Unpaired Data
Learning from Unpaired Data

Deep Network

\[ x \rightarrow y \]

\( x^1, x^3, x^5, x^9 \) \text{ (unpaired)}

\( y^2, y^4, y^8, y^{10} \) \text{ (unpaired)}

HW3: pseudo labeling
HW5: back translation

\text{Still need some paired data}
Learning from Unpaired Data

Can we learn the mapping without any paired data?

Unsupervised Conditional Generation
Learning from Unpaired Data
Cycle GAN

Domain $\mathcal{X}$

Become similar to domain $\mathcal{Y}$

Domain $\mathcal{Y}$

Input image belongs to domain $\mathcal{Y}$ or not

$G_{\mathcal{X} \rightarrow \mathcal{Y}}$

$D_{\mathcal{Y}}$ scalar
Cycle GAN

Domain $\mathcal{X}$

ignore input

$G_{\mathcal{X} \rightarrow \mathcal{Y}}$

Become similar to domain $\mathcal{Y}$

Domain $\mathcal{Y}$

Input image belongs to domain $\mathcal{Y}$ or not

$D_{\mathcal{Y}}$ scalar
Cycle GAN

\[ G_{\mathcal{X} \rightarrow \mathcal{Y}} \quad \text{as close as possible} \]
\[ D_{\mathcal{Y}} \]

Lack of information for reconstruction

Domain \( \mathcal{Y} \)

Input image belongs to domain \( \mathcal{Y} \) or not

scalar
Cycle GAN

- $G_{\mathcal{X} \rightarrow \mathcal{Y}}$
- $G_{\mathcal{Y} \rightarrow \mathcal{X}}$

Cycle consistency

as close as possible

“Related” to input, so possible to reconstruct

Domain $\mathcal{Y}$

Input image belongs to domain $\mathcal{Y}$ or not

$D_{\mathcal{Y}}$ scalar
Cycle GAN

\[ G_{\mathcal{X} \rightarrow \mathcal{Y}} \rightarrow G_{\mathcal{Y} \rightarrow \mathcal{X}} \]

as close as possible

Cycle consistency

\[ D_{\mathcal{Y}} \rightarrow G_{\mathcal{Y} \rightarrow \mathcal{X}} \]

\[ D_{\mathcal{X}} \rightarrow G_{\mathcal{X} \rightarrow \mathcal{Y}} \]

scalar: belongs to domain \( \mathcal{X} \) or not

\[ G_{\mathcal{Y} \rightarrow \mathcal{X}} \rightarrow G_{\mathcal{X} \rightarrow \mathcal{Y}} \]
Disco GAN
https://arxiv.org/abs/1703.05192

Dual GAN
https://arxiv.org/abs/1704.02510

Cycle GAN
https://arxiv.org/abs/1703.10593
StarGAN

https://arxiv.org/abs/1711.09020
SELFIE2ANIME

https://selfie2anime.com/
https://arxiv.org/abs/1907.10830
Text Style Transfer

你真笨 (negative)

你真聰明 (positive)

Seq2seq
Text Style Transfer

Circle GAN

积极或不是？

Discriminator

Seq2seq
generate

你真笨
(negative)

你真聪明
(positive)

Seq2seq

transform

你真笨
(negative)

minimize the reconstruction error
Text Style Transfer

- From negative sentence to positive one

胃疼，沒睡醒，各種不舒服

我都想去上班了，真夠賤的！

暈死了，吃燒烤、竟然遇到個變態狂

我肚子痛的厲害
Unsupervised Abstractive Summarization
https://arxiv.org/abs/1810.02851

Unsupervised Translation
https://arxiv.org/abs/1710.04087
https://arxiv.org/abs/1710.11041

Unsupervised ASR
https://arxiv.org/abs/1804.00316
https://arxiv.org/abs/1812.09323
https://arxiv.org/abs/1904.04100
Evaluation of Generation
Quality of Image

- Human evaluation is expensive (and sometimes unfair/unstable).
- How to evaluate the quality of the generated images automatically?

\[ y \rightarrow \text{Off-the-shelf Image Classifier} \rightarrow P(c|y) \]

Concentrated distribution means higher visual quality

e.g., Inception net, VGG, etc.

class 1
class 2
class 3
Diversity - Mode Collapse

☆ : real data
★☆ : generated data
Diversity - Mode Dropping

🌟 : real data
🌟 : generated data

Generator at iteration $t$

Generator at iteration $t+1$

(BEGAN on CelebA)
Diversity

\[ \begin{align*}
P(c|y^1) & \\
& = \frac{1}{N} \sum_{n} P(c|y^n)
\end{align*} \]

\[ P(c|y^2) \]

\[ P(c|y^3) \]

low diversity
Diversity

$P(c|y^1)$

$P(c|y^2)$

$P(c|y^3)$

Inception Score (IS):
Good quality, large diversity $\rightarrow$ Large IS

$P(c) = \frac{1}{N} \sum_{n} P(c|y^n)$

Uniform means higher variety

What is the problem here? 😊
Fréchet Inception Distance (FID)

Fréchet Inception Distance (FID) is a measure of the similarity between real and generated image distributions. It is calculated as the Fréchet distance between the two Gaussians that best match the real and generated distributions. The Fréchet distance is defined as the distance between the means of the two distributions, weighted by the covariance matrix.

The formula for FID is:

$$FID = \sqrt{\text{tr}(\Sigma_1) + \text{tr}(\Sigma_2) - 2\text{tr}(\Sigma_{12})}$$

where $\Sigma_1$ and $\Sigma_2$ are the covariance matrices of the real and generated distributions, respectively, and $\Sigma_{12}$ is the covariance matrix between the two distributions.

The smaller the FID, the more similar the real and generated distributions are. This makes FID a useful metric for evaluating the quality of generated images in generative modeling tasks.
### GAN

#### Discriminator Loss

<table>
<thead>
<tr>
<th>GAN</th>
<th>$\mathcal{L}_D^{GAN}$</th>
<th>$\mathcal{L}_G^{GAN}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM GAN</td>
<td>$-\mathbb{E}<em>{x \sim p_d}[\log(D(x))] + \mathbb{E}</em>{\hat{x} \sim p_g}[\log(1 - D(\hat{x}))]$</td>
<td>$-\mathcal{L}_D^{GAN}$</td>
</tr>
<tr>
<td>NS GAN</td>
<td>$\mathcal{L}_{NSGAN}^{GAN} = \mathcal{L}_D^{GAN}$</td>
<td>$\mathcal{L}<em>{NSGAN}^{GAN} = \mathbb{E}</em>{\hat{x} \sim p_g}[\log(D(\hat{x}))]$</td>
</tr>
<tr>
<td>WGAN</td>
<td>$\mathcal{L}<em>D^{WGAN} = -\mathbb{E}</em>{x \sim p_d}[D(x)] + \mathbb{E}_{\hat{x} \sim p_g}[D(\hat{x})]$</td>
<td>$\mathcal{L}_G^{WGAN} = -\mathcal{L}_D^{WGAN}$</td>
</tr>
<tr>
<td>WGAN GP</td>
<td>$\mathcal{L}_D^{WGAN} = \mathcal{L}<em>D^{WGAN} + \lambda \mathbb{E}</em>{\hat{x} \sim p_g}[|\nabla D(\alpha x + (1 - \alpha)\hat{x})|_2 - 1)^2]$</td>
<td>$\mathcal{L}<em>G^{WGAN} = -\mathbb{E}</em>{\hat{x} \sim p_g}[D(\hat{x})]$</td>
</tr>
<tr>
<td>LS GAN</td>
<td>$\mathcal{L}<em>D^{LSGAN} = -\mathbb{E}</em>{x \sim p_d}[(D(x) - 1)^2] + \mathbb{E}_{\hat{x} \sim p_g}[D(\hat{x})^2]$</td>
<td>$\mathcal{L}<em>G^{LSGAN} = -\mathbb{E}</em>{\hat{x} \sim p_g}[(D(\hat{x}) - 1)^2]$</td>
</tr>
<tr>
<td>DRAGAN</td>
<td>$\mathcal{L}_D^{DRAGAN} = \mathcal{L}<em>D^{GAN} + \lambda \mathbb{E}</em>{\hat{x} \sim p_d + N(0, \sigma)}[|\nabla D(\hat{x})|_2 - 1)^2]$</td>
<td>$\mathcal{L}_G^{DRAGAN} = -\mathcal{L}_D^{NS GAN}$</td>
</tr>
<tr>
<td>BEGAN</td>
<td>$\mathcal{L}<em>D^{BEGAN} = \mathbb{E}</em>{x \sim p_d}[|x - AE(x)|<em>1] - k_t \mathbb{E}</em>{\hat{x} \sim p_g}[|\hat{x} - AE(\hat{x})|_1]$</td>
<td>$\mathcal{L}<em>G^{BEGAN} = \mathbb{E}</em>{\hat{x} \sim p_g}[|\hat{x} - AE(\hat{x})|_1]$</td>
</tr>
</tbody>
</table>

### FIT: Smaller is better

Are GANs Created Equal? A Large-Scale Study

https://arxiv.org/abs/1711.10337
We don’t want memory GAN.

Real Data

Generated Data

Same as real data ...

Generated Data

Simply flip real data ...
Pros and cons of GAN evaluation measures

Concluding Remarks

- Introduction of Generative Models
- Generative Adversarial Network (GAN)
- Theory behind GAN
- Tips for GAN
- Conditional Generation
- Learning from unpaired data
- Evaluation of Generative Models
Q&A