Improving Generative Adversarial Network (GAN)

Hung-yi Lee
Outline

Basic Idea (Review)

Unified Framework

WGAN

Evaluation

Energy-based GAN

(next time)

Paired Data

Unpaired Data

Transformation
Generation

Using Generative Adversarial Network (GAN)

Drawing?
Basic Idea of GAN

• The data we want to generate has a distribution $P_{data}(x)$
Basic Idea of GAN

• A generator $G$ is a network. The network defines a probability distribution.

\[ z \xrightarrow{\text{generator } G} x = G(z) \]

As close as possible

$P_G(x)$

$P_{\text{data}}(x)$

It is difficult to compute $P_G(x)$

We can only sample from the distribution.

https://blog.openai.com/generative-models/
Basic Idea of GAN

It can be proofed that the loss of the discriminator related to JS divergence.
Basic Idea of GAN

Normal Distribution $\rightarrow$ Generator $\rightarrow$ Discriminatory

It can be proofed that the loss the discriminator related to \textit{JS divergence}.
Intuition

• Discriminator leads the generator

![Diagram of discriminator and generated distribution]
Original GAN

The discriminator is flat in the end.

Source: https://www.youtube.com/watch?v=ebMei6bYeWw (credit: Benjamin Striner)
**Algorithm**

Initialize $\theta_d$ for D and $\theta_g$ for G

In each training iteration:

- Sample $m$ examples $\{x^1, x^2, ..., x^m\}$ from data distribution $P_{data}(x)$
- Sample $m$ noise samples $\{z^1, z^2, ..., z^m\}$ from the prior $P_{prior}(z)$
- Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, ..., \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$
- Update discriminator parameters $\theta_d$ to maximize
  \[
  \tilde{V} = \frac{1}{m} \sum_{i=1}^{m} \log D(x^i) + \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D(\tilde{x}^i)\right)
  \]
  \[
  \theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)
  \]

- Sample another $m$ noise samples $\{z^1, z^2, ..., z^m\}$ from the prior $P_{prior}(z)$
- Update generator parameters $\theta_g$ to minimize
  \[
  \tilde{V} = \frac{1}{m} \sum_{i=1}^{m} \log D(x^i) + \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D(G(z^i))\right)
  \]
  \[
  \theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)
  \]
Outline

Basic Idea (Review)

Unified Framework

Paired Data

Unpaired Data

WGAN

Evaluation

f-divergence

Fenchel Conjugate

Connect to GAN

Energy-based GAN

(next time)
Reference


• One sentence: you can use any f-divergence
Outline

Basic Idea (Review)

Paired Data

Unpaired Data

Unified Framework

WGAN

f-divergence

Fenchel Conjugate

Connect to GAN

Energy-based GAN

(next time)

Generation

Transformation
**f-divergence**  

\[ D_f(P||Q) = \int q(x)f\left(\frac{p(x)}{q(x)}\right) dx \]  

- If \( p(x) = q(x) \) for all \( x \),  
  \[ D_f(P||Q) = \int q(x)f\left(\frac{p(x)}{q(x)}\right) dx = 0 \]  
  - Because \( f \) is convex,  
  \[ \geq f\left(\int q(x)\frac{p(x)}{q(x)} dx\right) \]  
  \[ = f(1) = 0 \]

- If \( P \) and \( Q \) are two distributions, \( p(x) \) and \( q(x) \) are the probability of sampling \( x \).

- \( D_f(P||Q) \) evaluates the difference of \( P \) and \( Q \).

- If \( P \) and \( Q \) are the same distributions, \( D_f(P||Q) \) has the smallest value, which is 0.
**f-divergence**

\[ D_f(P \parallel Q) = \int x q(x) f \left( \frac{p(x)}{q(x)} \right) dx \]

- **f** is convex
- **f(1) = 0**

\[ f(x) = x \log x \]

\[ D_f(P \parallel Q) = \int x q(x) \frac{p(x)}{q(x)} \log \left( \frac{p(x)}{q(x)} \right) dx = \int x p(x) \log \left( \frac{p(x)}{q(x)} \right) dx \]

\[ f(x) = -\log x \]

\[ D_f(P \parallel Q) = \int x q(x) \left( -\log \left( \frac{p(x)}{q(x)} \right) \right) dx = \int x q(x) \log \left( \frac{q(x)}{p(x)} \right) dx \]

\[ f(x) = (x - 1)^2 \]

\[ D_f(P \parallel Q) = \int x q(x) \left( \frac{p(x)}{q(x)} - 1 \right)^2 dx = \int x \frac{(p(x) - q(x))^2}{q(x)} dx \]
Outline

Basic Idea (Review)

Unified Framework

Paired Data

Unpaired Data

Evaluation

(f-Divergence)

Fenchel Conjugate

Connect to GAN

Energy-based GAN

(next time)
Fenchel Conjugate

- Every convex function $f$ has a conjugate function $f^*$

\[
f^*(t) = \max_{x \in \text{dom}(f)} \{xt - f(x)\}
\]

\[
f^*(t_1) = \max_{x \in \text{dom}(f)} \{xt_1 - f(x)\}
\]

\[
x_1 t_1 - f(x_1) \quad f^*(t_1) \quad f^*(t_2) = \max_{x \in \text{dom}(f)} \{xt_2 - f(x)\}
\]

\[
x_2 t_1 - f(x_2) \quad x_3 t_2 - f(x_3) \quad f^*(t_2)
\]

\[
x_3 t_1 - f(x_3) \quad x_2 t_2 - f(x_2) \quad x_1 t_2 - f(x_1)
\]

\[
D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx
\]

If $f$ is convex, $f(1) = 0$
Fenchel Conjugate

Every convex function $f$ has a conjugate function $f^*$

$$f^*(t) = \max_{x \in \text{dom}(f)} \{xt - f(x)\}$$

$D_f(P||Q) = \int_{x} q(x)f\left(\frac{p(x)}{q(x)}\right)dx$

$f$ is convex, $f(1) = 0$
Fenchel Conjugate

• Every convex function $f$ has a conjugate function $f^*$

$$f^*(t) = \max_{x \in \text{dom}(f)} \{xt - f(x)\}$$

$$f(x) = x \log x$$

$f^*(t) = \exp(t - 1)$

Something like exponential?
Fenchel Conjugate

• Every convex function \( f \) has a conjugate function \( f^* \)

• \((f^*)^* = f\)

\[
f^*(t) = \max_{x \in \text{dom}(f)} \{xt - f(x)\}
\]

\[
f(x) = x log x \quad \Rightarrow \quad f^*(t) = \exp(t - 1)
\]

\[
f^*(t) = \max_{x \in \text{dom}(f)} \{xt - x log x\}
\]

\[
g(x) = xt - x log x \quad \text{Given } t, \text{ find } x \text{ maximizing } g(x)
\]

\[
t - log x - 1 = 0 \quad x = \exp(t - 1)
\]

\[
f^*(t) = \exp(t - 1) \times t - \exp(t - 1) \times (t - 1) = \exp(t - 1)
\]
Outline

Basic Idea (Review)

Unified Framework

Paired Data

Unpaired Data

WGAND

Evaluation

Generation

f-divergence

Fenchel Conjugate

Connect to GAN

Energy-based GAN

(next time)
Connection with GAN

\[ f^*(t) = \sup_{x \in \text{dom}(f)} \{ xt - f(x) \} \quad \leftrightarrow \quad f(x) = \max_{t \in \text{dom}(f^*)} \{ xt - f^*(t) \} \]

\[ D_f(P\|Q) = \int x q(x) f \left( \frac{p(x)}{q(x)} \right) dx \]

\[ = \int x q(x) \left( \max_{t \in \text{dom}(f^*)} \left\{ \frac{p(x)}{q(x)} t - f^*(t) \right\} \right) dx \]

\[ \approx \max_D \int p(x)D(x)dx - \int q(x)f^*(D(x))dx \]

\[ D_f(P\|Q) \geq \int x q(x) \left( \frac{p(x)}{q(x)} D(x) - f^*(D(x)) \right) dx \]

\[ = \int p(x)D(x)dx - \int q(x)f^*(D(x))dx \]

\( D \) is a function whose input is \( x \), and output is \( t \).
Connection with GAN

\[ D_f(P\|Q) \approx \max_D \int p(x)D(x)dx - \int q(x)f^*(D(x))dx \]

\[ = \max_D \left\{ E_{x\sim P}[D(x)] - E_{x\sim Q}[f^*(D(x))] \right\} \]

Samples from P                           Samples from Q

\[ D_f(P_{data}\|P_G) = \max_D \left\{ E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[f^*(D(x))] \right\} \]

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

\[ = \arg \min_G \max_D \left\{ E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[f^*(D(x))] \right\} \]

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

Original GAN has different \( V(G, D) \)

familiar? 😊
Double-loop v.s. Single-step

\[ G^* = \arg \min_G \max_D V(G, D) \quad \Rightarrow \quad G^* = \arg \min_{\theta_G} \max_{\theta_D} V(\theta_G, \theta_D) \]

- Original paper of GAN: double-loop algorithm
  - In each iteration
    - Given a generator \( \theta^t_G \), \( \theta^t_D = \arg \max_{\theta_D} V(\theta^t_G, \theta_D) \)
    - Update the parameters many times to find \( \theta^t_D \)
    - Update generator once:
      - \( \theta^{t+1}_G \leftarrow \theta^t_G - \eta \nabla_{\theta_G} V(\theta^t_G, \theta^t_D) \)

- Paper of f-GAN: Single-step algorithm
  - In each iteration, given \( \theta^t_G \) and \( \theta^t_D \)
    - \( \theta^{t+1}_D \leftarrow \theta^t_D + \eta \nabla_{\theta_D} V(\theta^t_G, \theta^t_D) \)
    - \( \theta^{t+1}_G \leftarrow \theta^t_G - \eta \nabla_{\theta_G} V(\theta^t_G, \theta^t_D) \)
\[
D_f(P_{\text{data}} \parallel P_G) = \max_{D} \{ E_{x \sim P_{\text{data}}} [D(x)] - E_{x \sim P_G} [f^*(D(x))] \}
\]

<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{\pi p(x)+(1-\pi)q(x)}{\pi p(x)+(1-\pi)q(x)} + (1-\pi) \log \frac{q(x)}{\pi 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 - \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}{2} 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} W(e^{1-t}) + \frac{1}{W(e^{1-t})} + t - 2]</td>
</tr>
<tr>
<td>Jeffrey</td>
<td>(- \log(2 - \exp(t)))</td>
</tr>
<tr>
<td>Jensen-Shannon</td>
<td>[(1-\pi) \log \frac{1-\pi}{1-\pi e^{t/\pi}}]</td>
</tr>
<tr>
<td>Jensen-Shannon-weighted</td>
<td>(- \log(1 - \exp(t)))</td>
</tr>
<tr>
<td>GAN</td>
<td></td>
</tr>
</tbody>
</table>

Using the f-divergence you like 😊

Experimental Results

- Approximate a mixture of Gaussians by single mixture

<table>
<thead>
<tr>
<th>train \ test</th>
<th>KL</th>
<th>KL-rev</th>
<th>JS</th>
<th>Jeffrey</th>
<th>Pearson</th>
</tr>
</thead>
<tbody>
<tr>
<td>KL</td>
<td><strong>0.2808</strong></td>
<td>0.3423</td>
<td>0.1314</td>
<td>0.5447</td>
<td>0.7345</td>
</tr>
<tr>
<td>KL-rev</td>
<td>0.3518</td>
<td><strong>0.2414</strong></td>
<td>0.1228</td>
<td>0.5794</td>
<td>1.3974</td>
</tr>
<tr>
<td>JS</td>
<td>0.2871</td>
<td>0.2760</td>
<td><strong>0.1210</strong></td>
<td>0.5260</td>
<td>0.92160</td>
</tr>
<tr>
<td>Jeffrey</td>
<td>0.2869</td>
<td>0.2975</td>
<td>0.1247</td>
<td><strong>0.5236</strong></td>
<td>0.8849</td>
</tr>
<tr>
<td>Pearson</td>
<td>0.2970</td>
<td>0.5466</td>
<td>0.1665</td>
<td>0.7085</td>
<td><strong>0.648</strong></td>
</tr>
</tbody>
</table>
Outline

Basic Idea
(Review)

Unified Framework

Paired Data

WGAN

Unpaired Data

Energy-based GAN

Original version
(weight clipping)

Improved version
(gradient penalty)

Evaluation

Generation

Transformation

(next time)
Reference

- One sentence for WGAN: Using Earth Mover’s Distance to evaluate two distributions
  - Earth Mover’s Distance = Wasserstein Distance
Outline

Basic Idea (Review)

Unified Framework

WGAN

Paired Data

Unpaired Data

Evaluation

Generation

Original version (weight clipping)

Improved version (gradient penalty)

Energy-based GAN

(next time)
Earth Mover’s 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.

$$W(P, Q) = d$$
Earth Mover’s Distance

There are many possible “moving plans”.
Using the “moving plan” with the smallest average distance to define the earth mover’s distance.

Source of image: https://vincentherrmann.github.io/blog/wasserstein/
Earth Mover’s Distance

Using the “moving plan” with the smallest average distance to define the earth mover’s distance.

There many possible “moving plans”.

Source of image: https://vincentherrmann.github.io/blog/wasserstein/
A “moving plan” is a matrix. The value of the element is the amount of earth from one position to another.

Average distance of a plan $\gamma$:

$$B(\gamma) = \sum_{x_p, x_q} \gamma(x_p, x_q) \| x_p - x_q \|$$

Earth Mover’s Distance:

$$W(P, Q) = \min_{\gamma \in \Pi} B(\gamma)$$

The best plan
Why Earth Mover’s Distance?

\[ D_f(P_{\text{data}} \parallel P_G) \]

\[ W(P_{\text{data}}, P_G) \]

\[ d_0 \]

\[ d_{50} \]

\[ P_{G_0} \]

\[ P_{\text{data}} \]

\[ P_{G_50} \]

\[ P_{G_{100}} \]

\[ JS(P_{G_0}, P_{\text{data}}) = \log 2 \]

\[ JS(P_{G_{50}}, P_{\text{data}}) = \log 2 \]

\[ JS(P_{G_{100}}, P_{\text{data}}) = 0 \]

\[ W(P_{G_0}, P_{\text{data}}) = d_0 \]

\[ W(P_{G_{50}}, P_{\text{data}}) = d_{50} \]

\[ W(P_{G_{100}}, P_{\text{data}}) = 0 \]
Back to the GAN framework

\[ D_f(P_{\text{data}} \| P_G) \overset{\rightarrow}{=} W(P_{\text{data}}, P_G) \]

\[ = \max_D \{ E_{x \sim P_{\text{data}}} [D(x)] - E_{x \sim P_G} [f^*(D(x))] \} \]

\[ W(P_{\text{data}}, P_G) \]

\[ = \max_{D \in 1-\text{Lipschitz}} \{ E_{x \sim P_{\text{data}}} [D(x)] - E_{x \sim P_G} [D(x)] \} \]

**Lipschitz Function**

\[ \| f(x_1) - f(x_2) \| \leq K \| x_1 - x_2 \| \]

1-Lipschitz?

Output change

Input change

K=1 for "1 – Lipschitz"

Do not change fast
Back to the GAN framework

\[ W(P_{\text{data}}, P_G) = \max_{D \in 1-Lipschitz} \left\{ E_{x \sim P_{\text{data}}} [D(x)] - E_{x \sim P_G} [D(x)] \right\} \]

\[
\begin{align*}
D(x_1) & = +\infty k + d \\
D(x_2) & = -\infty k
\end{align*}
\]

\[ W(P_{\text{data}}, P_G) = d \]

\[ \|D(x_1) - D(x_2)\| \leq \|x_1 - x_2\| \]

Blue: \(D(x)\) for original GAN
Green: \(D(x)\) for WGAN

WGAN will provide gradient to push \(P_G\) towards \(P_{\text{data}}\)
Back to the GAN framework

\[ W(K, (P_{data}, P_G)) = \max_{D \in 1-Lipschitz} \left\{ E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)] \right\} \]

How to use gradient descent to optimize?

**Weight clipping:**
Force the weights \( w \) between \( c \) and \( -c \)

After parameter update,
if \( w > c \), then \( w = c \); if \( w < -c \), then \( w = -c \)

We only ensure that
\[ ||D(x_1) - D(x_2)|| \leq K||x_1 - x_2|| \]

For some \( K \)

Do not truly find function \( D \) maximizing the function.
Algorithm of WGAN

- In each training iteration:
  - Sample m examples \( \{x^1, x^2, \ldots, x^m\} \) from data distribution \( P_{data}(x) \)
  - Sample m noise samples \( \{z^1, z^2, \ldots, z^m\} \) from the prior \( P_{prior}(z) \)
  - Obtaining generated data \( \{\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^m\} \), \( \tilde{x}^i = G(z^i) \)
  - Update discriminator parameters \( \theta_d \) to maximize
    \[
    \tilde{V} = \frac{1}{m} \sum_{i=1}^{m} D(x^i) - \frac{1}{m} \sum_{i=1}^{m} D(\tilde{x}^i)
    \]
    \[
    \theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)
    \]
    Weight clipping
  - Sample another m noise samples \( \{z^1, z^2, \ldots, z^m\} \) from the prior \( P_{prior}(z) \)
  - Update generator parameters \( \theta_g \) to minimize
    \[
    \tilde{V} = \frac{1}{m} \sum_{i=1}^{m} \log D(x^i) - \frac{1}{m} \sum_{i=1}^{m} D(G(z^i))
    \]
    \[
    \theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)
    \]
CNN generator:

W-GAN  GAN

CNN generator (no batch normalization, bad structure):

W-GAN  GAN

MLP generator:

W-GAN  GAN
**Vertical**

\[
W(P_{\text{data}}, P_G) = \max_{D \in 1-\text{Lipschitz}} \left\{ E_{x \sim P_{\text{data}}} [D(x)] - E_{x \sim P_G} [D(x)] \right\}
\]

https://arxiv.org/abs/1701.07875
Improved WGAN

\[ W(P_{data}, P_G) \]

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

A differentiable function is 1-Lipschitz if and only if it has gradients with norm less than or equal to 1 everywhere.

\[ D \in 1 - \text{Lipschitz} \iff \| \nabla_x D(x) \| \leq 1 \text{ for all } x \]

\[ W(P_{data}, P_G) \approx \max_D \{ E_{x \sim P_{data}} [D(x)] - E_{x \sim P_G} [D(x)] \]  

\[ - \lambda \int_x \max(0, \| \nabla_x D(x) \| - 1) dx \} \]

Prefer \( \| \nabla_x D(x) \| \leq 1 \text{ for all } x \)

\[ -\lambda E_{x \sim P_{penalty}} [\max(0, \| \nabla_x D(x) \| - 1)] \}

Prefer \( \| \nabla_x D(x) \| \leq 1 \text{ for } x \text{ sampling from } x \sim P_{penalty} \)
Improved WGAN

\[ W(P_{data}, P_G) \approx \max_D \{ E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)] \] \]
\[ - \lambda E_{x \sim P_{penalty}}[\max(0, \| \nabla_x D(x) \| - 1)] \}\]

“Given that enforcing the Lipschitz constraint everywhere is intractable, enforcing it only along these straight lines seems sufficient and experimentally results in good performance.”

Only give gradient constraint to the region between \( P_{data} \) and \( P_G \) because they influence how \( P_G \) moves to \( P_{data} \)
**Improved WGAN**

\[
W(P_{data}, P_G) \approx \max_D \{ E_{x \sim P_{data}} [D(x)] - E_{x \sim P_G} [D(x)] - \lambda E_{x \sim P_{penalty}} [\max(0, \|\nabla_x D(x)\| - 1)] \}
\]

“One may wonder why we penalize the norm of the gradient for differing from 1, instead of just penalizing large gradients. The reason is that the optimal critic ... actually has gradients with norm 1 almost everywhere under Pr and Pg”

(check the proof in the appendix)

“Simply penalizing overly large gradients also works in theory, but experimentally we found that this approach converged faster and to better optima.”
Improved WGAN

https://arxiv.org/abs/1704.00028
DCGAN: G: CNN, D: CNN

LSGAN: G: CNN (no normalization), D: CNN (no normalization)

Original WGAN: G: CNN (tanh), D: CNN (tanh)

Improved WGAN: G: CNN (tanh), D: CNN (tanh)
DCGAN
G: MLP, D: CNN

LSGAN

Original
WGAN
G: CNN (bad structure), D: CNN

Improved
WGAN
G: 101 layer, D: 101 layer
Sentence Generation

- good bye.

Consider this matrix as an “image”
Sentence Generation

- Real sentence
  
  A binary classifier can immediately find the difference.

- Generated
  
  No overlap

Can never be 1-of-N

WGAN is helpful
Improved WGAN successfully generating sentences
感謝 李仲翊 同學提供實驗結果
輸出 32 個字 (包含標點)

- 升雲白遲丹齋取，此酒新巷市入頭。黃道故海歸中後，不驚入得韻子門。
- 據口容章蕃翎翎，邦貸無遊隔將毬。外蕭曾臺遙出畧，此計推上呂天夢。
- 新來寶伎泉，手雪泓臺蓑。曾子花路魏，不謀散薦船。
- 功持牧度機邈爭，不躚官嬉牧涼散。不迎白旅今掩冬，盡蘸金祇可停。
- 玉十洪沄爭春風，溪子風佛挺橫鞋。盤盤稅焰先花齋，誰過飄鶴一丞憧。
- 海人依野庇，為阻例沉迥。座花不佐樹，弟闌十名儂。
- 入維當興日世瀕，不評皺。頭醉空其杯，騭園凋送頭。
- 鉢笙動春枝，寶叅潔長知。官爲宻爛去，絆粒薛一靜。
- 吾涼腕不楚，縱先待旅知。楚人縱酒待，一蔓飄聖猜。
- 折幕故癘應韻子，徑頭霜瓊老徑徑。尚錯春鐫熊親梅，去吹依能九將香。
- 通可矯目鬩須浄，丹迄挈花一抵嫖。外子當目中前醒，迎日幽筆鈎弧前。
- 庭愛四樹人庭好，無衣服仍繡秋州。更怯風流欲鴉雲，帛陽舊據科研院。
More about Discrete Output

- **SeqGAN**
  - Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu, SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, AAAI, 2017

- **Boundary seeking GAN**

- **Gumbel-Softmax**
  - Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, Yoshua Bengio, Maximum-Likelihood Augmented Discrete Generative Adversarial Networks, arXiv preprint, 2017
Outline

Basic Idea (Review)

Unified Framework

WGAN

Evaluation

Energy-based GAN

(next time)

Transformation

Generation
Conditional GAN

- **Text to image** by traditional supervised learning

\[ \text{Text: “train”} \]

\[ \begin{align*}
& \text{Target of NN output} \\
& \text{A blurry image!} \\
& \text{as close as possible} \\
\end{align*} \]
Conditional GAN

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

Text: “train”

A blurry image!
Conditional GAN

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

Prior distribution \( z \)

\( x \) is realistic or not

Image

\( c: \) train

\( x \) is realistic or not + c and \( x \) are matched or not

Positive example:

Negative example:

Positive example:

Negative example:
Image-to-image

\[ x = G(c, z) \]
Image-to-image

- Traditional supervised approach

Testing:

It is blurry because it is the average of several images.
Image-to-image

- Experimental results

Testing:

input  close  GAN  GAN + close

$z$  \rightarrow  G  \rightarrow  Image  \rightarrow  D  \rightarrow  scalar
Outline

Basic Idea (Review)

Unified Framework

WGAN

Energy-based GAN

Generation

Paired Data

Unpaired Data

Transformation

Evaluation (next time)
Unpaired Transformation - Cycle GAN, Disco GAN

Transform an object from one domain to another without paired data.
Cycle GAN

https://arxiv.org/abs/1703.10593
https://junyanz.github.io/CycleGAN/

Domain X

Ignore input

$G_{X \rightarrow Y}$

Become similar to domain Y

Domain Y

$D_Y$ -> scalar

Not what we want

Input image belongs to domain Y or not
Cycle GAN

$G_{X \rightarrow Y}$

as close as possible

$Lack of information for reconstruction$

$D_Y$

Input image belongs to domain Y or not

$G_{Y \rightarrow X}$

scalar

Domain X

Domain Y
Cycle GAN

\[ G_X \rightarrow Y \quad \rightarrow \quad G_Y \rightarrow X \]

as close as possible

\[ D_Y \]

scalar: belongs to domain Y or not

\[ D_X \]

scalar: belongs to domain X or not

as close as possible
Unpaired Transformation – 真人動畫化

• http://qiita.com/Hi-king/items/8d36d9029ad1203aac55

http://www.iis.ee.ic.ac.uk/cxiong/database.html

把真人頭像變成動漫人物
Unpaired Transformation – 真人動畫化

• Using the code:  
  https://github.com/Hi-king/kawaii_creator

• It is not cycle GAN, Disco GAN
Outline

Basic Idea (Review)

Unified Framework

WGAN

Generation

Energy-based GAN

(next time)

Paired Data

Unpaired Data

Transformation

Evaluation

Generation
Review

• Role of discriminator: lead the generator

When the data distribution and generated distribution is the same, the discriminator will be useless (flat).
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 $F(x)$?

\[ x \rightarrow \text{Evaluation Function} \rightarrow \text{scalar} \]
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