Generative Adversarial Networks
Part II
11785- Introduction to Deep Learning

CHAORAN ZHANG and DIKSHA AGARWAL
Fall 2021
Contents

• Story So Far
• Training Issue in GANs
• GANs’ Training and Stabilization
• Wasserstein GANs
• Conditional GANs
• Gan’s Progression
Contents

• Story So Far
• Training Issue in GANs
• GANs’ Training and Stabilization
• Wassertein GANs
• Conditional GANs
• Gan’s Progression
WHAT ARE GANS?

Generative Adversarial Networks:

Generative -> Generative Models ->
Learn the underlying distribution, from which our dataset comes from, e.g. VAE

Adversarial -> Adversarial Training ->
Not only made up with generator, but also add an adversarial network, which two trying to beat each other.

Networks -> Neural Networks

GOAL:
Generate data from an unlabelled distribution
HOW TO TRAIN A GAN?

At \( t = 0 \),

Latent Vector \[\rightarrow\] Generator \[\rightarrow\] Generated (Fake image) Image

(Fake data) \[\rightarrow\] Generated Data

(Relal data) \[\rightarrow\] Given Training Data

\[\rightarrow\] Discriminator \[\rightarrow\] Real/Fake?

Binary Classifier
HOW TO TRAIN A GAN?

Step 1: Train the Discriminator *using the current ability* of the Generator.

Step 2: Train the Generator to beat the Discriminator.
HOW TO TRAIN A GAN?

Step 1: Train the Discriminator

Discriminator -> Binary classifier classifying data into real/false.

Real data -> Real data
False data -> Outputs from Generator

Goal:
Chances real data are classified as real data are maximized
Chances fake data are classified as fake data are maximized
Step 2: Train the Generator

Goal:

Chances that generated data are classified incorrectly by Discriminator are maximized
HOW TO TRAIN A GAN?

Discriminator -> $D(X; \theta)$;
Generator -> $G(Z; \theta)$

$P_D$ -> actual data distribution
$P_G$ -> generated data distribution

$D(X)$ : Output of the discriminator
  Probability that $X$ came from actual data distribution $P_D$

$G(Z) \sim P_G$: Output of the generator
HOW TO TRAIN A GAN?

Discriminator -> D(X; θ):

Goal:

<table>
<thead>
<tr>
<th>Chances of real data are classified as real is maximized</th>
<th>Chances of fake data are classified as fake is maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $X \sim P_D$, $D(X)$ is maximized</td>
<td>For $X \sim P_G$, $D(X)$ is minimized</td>
</tr>
</tbody>
</table>
HOW TO TRAIN A GAN?

Discriminator -> $D(X; \theta)$:

Goal:

<table>
<thead>
<tr>
<th>Chances of real data are classified as real is maximized</th>
<th>Chances of fake data are classified as fake is maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $X \sim P_D, D(X)$ is maximized</td>
<td>For $X \sim P_G, D(X)$ is minimized</td>
</tr>
<tr>
<td>For $X \sim P_D, \log(D(X))$ is maximized</td>
<td>For $X \sim P_G, \log(1 - D(X))$ is maximized</td>
</tr>
</tbody>
</table>
HOW TO TRAIN A GAN?

Discriminator -> D(X; θ):

Goal:

<table>
<thead>
<tr>
<th>Chances of real data are classified as real is maximized</th>
<th>Chances of fake data are classified as fake is maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $X \sim P_D, D(X)$ is maximized</td>
<td>For $X \sim P_G, D(X)$ is minimized</td>
</tr>
<tr>
<td>For $X \sim P_D, \log(D(X))$ is maximized</td>
<td>For $X \sim P_G, \log(1 - D(X))$ is maximized</td>
</tr>
<tr>
<td>$E_{X \sim P_D} [\log(D(X))]$ is maximized</td>
<td>$E_{X \sim P_G} [\log(1 - D(X))]$ is maximized</td>
</tr>
</tbody>
</table>
# How to Train a GAN?

**Discriminator \( \rightarrow D(X; \theta) \):**

**Goal:**

<table>
<thead>
<tr>
<th>Chances of real data are classified as real is maximized</th>
<th>Chances of fake data are classified as fake is maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For ( X \sim P_D, D(X) ) is maximized</td>
<td>For ( X \sim P_G, D(X) ) is minimized</td>
</tr>
<tr>
<td>For ( X \sim P_D, \log(D(X)) ) is maximized</td>
<td>For ( X \sim P_G, \log(1 - D(X)) ) is maximized</td>
</tr>
<tr>
<td>( E_{X \sim P_D}[\log(D(X))] ) is maximized</td>
<td>( E_{Z \sim P_Z}[\log(1 - D(G(Z)))] ) is maximized</td>
</tr>
</tbody>
</table>


Discriminator -> D(X; θ):

Goal:

<table>
<thead>
<tr>
<th>Chances of real data are classified as real is maximized</th>
<th>Chances of fake data are classified as fake is maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $X \sim P_D$, $D(X)$ is maximized</td>
<td>For $X \sim P_G$, $D(X)$ is minimized</td>
</tr>
<tr>
<td>For $X \sim P_D$, $\log(D(X))$ is maximized</td>
<td>For $X \sim P_G$, $\log(1 - D(X))$ is maximized</td>
</tr>
<tr>
<td>$E_{X \sim P_D}[\log(D(X))]$ is maximized</td>
<td>$E_{Z \sim P_Z}[\log(1 - D(G(Z)))]$ is maximized</td>
</tr>
</tbody>
</table>

$$\max_{\theta_D} E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_Z}[\log(1 - D(G(Z)))]$$
HOW TO TRAIN A GAN?

Generator -> $G(X; \theta)$:

Goal:

Chances that generated data are classified incorrectly by Discriminator are maximized

For $Z \sim P_Z, D(G(Z))$ is maximized
HOW TO TRAIN A GAN?

Generator -> G(X; θ):

Goal:

<table>
<thead>
<tr>
<th>Chances that generated data are classified incorrectly by Discriminator are maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For Z ~ P_Z, D(G(Z)) is maximized</td>
</tr>
<tr>
<td>For Z ~ P_Z, log(D(G(Z))) is maximized</td>
</tr>
</tbody>
</table>
HOW TO TRAIN A GAN?

Generator -> G(X; θ):

Goal:

<table>
<thead>
<tr>
<th>Chances that generated data are classified incorrectly by Discriminator are maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $Z \sim P_Z, D(G(Z))$ is maximized</td>
</tr>
<tr>
<td>For $Z \sim P_Z, \log(1 - D(G(Z)))$ is minimized</td>
</tr>
</tbody>
</table>
HOW TO TRAIN A GAN?

Generator -> G(X; θ):

Goal:

<table>
<thead>
<tr>
<th>Chances that generated data are classified incorrectly by Discriminator are maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $Z \sim P_Z, D(G(Z))$ is maximized</td>
</tr>
<tr>
<td>For $Z \sim P_Z, \log(1 - D(G(Z)))$ is minimized</td>
</tr>
<tr>
<td>$E_{Z \sim P_Z} [\log(1 - D(G(Z)))]$ is minimized</td>
</tr>
</tbody>
</table>
### HOW TO TRAIN A GAN?

**Generator -> G(X; \theta):**

**Goal:**

<table>
<thead>
<tr>
<th>Chances that generated data are classified incorrectly by Discriminator are maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $Z \sim P_Z, D(G(Z))$ is maximized</td>
</tr>
<tr>
<td>For $Z \sim P_Z, \log(1 - D(G(Z)))$ is minimized</td>
</tr>
<tr>
<td>$E_{Z \sim P_Z}[\log(1 - D(G(Z)))]$ is minimized</td>
</tr>
<tr>
<td>Some Mysterious Constant + $E_{Z \sim P_Z}[\log(1 - D(G(Z)))]$ is minimized</td>
</tr>
</tbody>
</table>
**HOW TO TRAIN A GAN?**

Generator $\rightarrow G(X; \theta)$:

**Goal:**

- Chances that generated data are classified incorrectly by Discriminator are maximized

<table>
<thead>
<tr>
<th>Chances that generated data are classified incorrectly by Discriminator are maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $Z \sim P_Z, D(G(Z))$ is maximized</td>
</tr>
<tr>
<td>For $Z \sim P_Z, \log(1 - D(G(Z)))$ is minimized</td>
</tr>
<tr>
<td>$E_{Z \sim P_Z} [\log(1 - D(G(Z)))]$ is minimized</td>
</tr>
<tr>
<td>$E_{X \sim P_D} [\log(D(X))] + E_{Z \sim P_Z} [\log(1 - D(G(Z)))]$ is minimized</td>
</tr>
</tbody>
</table>
HOW TO TRAIN A GAN?

Generator -> G(X; θ):

Goal:

<table>
<thead>
<tr>
<th>Chances that generated data are classified incorrectly by Discriminator are maximized</th>
</tr>
</thead>
<tbody>
<tr>
<td>For $Z \sim P_Z, D(G(Z))$ is maximized</td>
</tr>
<tr>
<td>For $Z \sim P_Z, \log(1 - D(G(Z)))$ is minimized</td>
</tr>
<tr>
<td>$E_{Z \sim P_Z} [\log(1 - D(G(Z)))]$ is minimized</td>
</tr>
<tr>
<td>$\min_{\theta_G} E_{X \sim P_D} [\log(D(X))] + E_{Z \sim P_Z} [\log(1 - D(G(Z)))]$</td>
</tr>
</tbody>
</table>
HOW TO TRAIN A GAN?

Discriminator -> D(X; θ):

$$\max_{\theta_D} E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_Z}[\log(1 - D(G(Z)))]$$

Generator -> G(X; θ):

$$\min_{\theta_G} E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_Z}[\log(1 - D(G(Z)))]$$
HOW TO TRAIN A GAN?

Put it together:

GANs’ objective is formulated as:

$$\min_{\theta_G} \max_{\theta_D} E_{X \sim P_D} [\log(D(X))] + E_{Z \sim P_Z} [\log(1 - D(G(Z)))]$$
HOW TO TRAIN A GAN?

GANs’ objective:

\[
\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{X \sim P_D}[\log(D(X))] + \mathbb{E}_{Z \sim P_Z}[\log(1 - D(G(Z)))]
\]

\[
f := \mathbb{E}_{X \sim P_D} \log D(X) + \mathbb{E}_{X \sim P_G} \log(1 - D(X))
\]

\[
\frac{\partial f}{\partial D(X)} = \frac{P_D(X)}{D(X)} - \frac{P_G(X)}{1 - D(X)} = 0
\]

\[
\frac{P_D(X)}{D(X)} = \frac{P_G(X)}{1 - D(X)}
\]

\[
(1 - D(X))P_D(X) = D(X)P_G(X)
\]

\[
D(X) = \frac{P_D(X)}{P_G(X) + P_D(X)}
\]
HOW TO TRAIN A GAN?

GANs’ objective:

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{X \sim P_D} [\log(D(X))] + \mathbb{E}_{Z \sim P_Z} [\log(1 - D(G(Z)))]$$

$$f = \mathbb{E}_{X \sim P_D} \log D(X) + \mathbb{E}_{X \sim P_G} (1 - \log D(X))$$

$$= \mathbb{E}_{X \sim P_D} \log \frac{P_D(X)}{P_D(X) + P_G(X)} + \mathbb{E}_{X \sim P_G} \log \frac{P_G(X)}{P_D(X) + P_G(X)}$$

$$= 2 \cdot \text{JSD}(P_D || P_G) - \log 4$$

$$\min_{\theta_G} f = \min_{\theta_G} 2 \cdot \text{JSD}(P_D || P_G) - \log 4$$
HOW TO TRAIN A GAN?

GANs’ objective:

\[
\min_{\theta_G} 2 \cdot JSD(P_D || P_G) - \log 4
\]

Minimize JSD between \(P_D\) and \(P_G\)
Story So Far

GANs’ objective:

$$\min_{\theta_G} 2 \times JSD(P_D||P_G) - \log 4$$

Minimize JSD between $P_D$ and $P_G$

Min – Max Stationary points exists and need not be stable
Contents

• Story So Far

• Training Issue in GANs

• GANs’ Training and Stabilization

• Wassertein GANs

• Conditional GANs

• Gan’s Progression
ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

CASE - 1: Player A plays rock-paper-scissors with a probability of (0.36, 0.32, 0.32)
What is your best strategy?
What is your probability of winning?
CASE - 1: Player A plays rock-paper-scissors with a probability of (0.36, 0.32, 0.32)
What is your best strategy?
What is your probability of winning?
Ans: Player B will choose the strategy as paper.
Ans: 36% winning probability
CASE - 2: Player A plays rock-paper-scissors with a probability of (0.33, 0.33, 0.33)
What is your best strategy?
What is your probability of winning?
ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

CASE - 2: Player A plays rock-paper-scissors with a probability of (0.33, 0.33, 0.33)

What is your best strategy?
What is your probability of winning?

Ans: Any strategy will work.
Ans: 33%-win chance
Global optimum: Both players play uniformly with (0.33, 0.33, 0.33)
CASE - 2: Player A plays rock-paper-scissors with a probability of (0.33, 0.33, 0.33)
What is your best strategy?
What is your probability of winning?
Ans: Any strategy will work.
Ans: 33%-win chance
Global optimum: Both players play uniformly with (0.33, 0.33, 0.33)
CASE - 1: Player A plays rock-paper-scissors with a probability of 
(0.36, 0.32, 0.32)  
Now if player B optimizes all the way its optimal strategy  
is to choose paper first (0,1,0)  
Seeing this player, A will now choose scissor (0,0,1)  
Seeing this player B will now choose rock (1,0,0)  

................. This will keep on going and no stabilization  
can be achieved.
TRAINING ISSUES IN GAN

The two training issues in GAN are as follows:

Oscillations
Mode Collapse: Generates a small subspace but does not cover the entire distribution.

You tube video: https://www.youtube.com/watch?v=ktxhiKhWoEE
IMPROVED TECHNIQUES FOR TRAINING GAN

A collection of interesting techniques and experiments:

Feature Matching
Minibatch Discrimination
Historical Averaging
One-sided Label Smoothing
Virtual Batch Normalization
FEATURE MATCHING

Statistics of generated images should match statistics of real images

- Discriminator produces multidimensional output, a “statistic” of the data
- Generator trained to minimize $L_2$ between real and generated data
- Discriminator trained to maximize $L_2$ between real and generated data

$$\left\| \mathbb{E}_X D(X) - \mathbb{E}_Z D(G(Z)) \right\|_2^2$$
MINIBATCH DISCRIMINATION

Discriminator can look at multiple inputs at once and decide if those inputs come from the real or generated distribution

- GANs frequently collapse to a single point
- Discriminator needs to differentiate between two distributions
- Easier task if looking at multiple samples
HISTORICAL AVERAGING

Dampen oscillations by encouraging updates to converge to a mean

- GANs frequently create a cycle or experience oscillations
- Add a term to reduce oscillations that encourages the current parameters to be near a moving average of the parameters

\[ \left\| \theta - \frac{1}{t} \sum_{i}^{t} \theta_i \right\|_2^2 \]
ONE-SIDED LABEL SMOOTHING

Don’t over-penalize generated images

- Label smoothing is a common and easy technique that improves performance across many domains
  - Sigmoid tries hard to saturate to 0 or 1 but can never quite reach that goal
  - Provide targets that are $\epsilon$ or $1 - \epsilon$ so the sigmoid doesn’t saturate and overtrain

- Experimentally, smooth the real targets but do not smooth the generated targets when training the discriminator
VIRTUAL BATCH NORMALIZATION

Use batch normalization to accelerate convergence

- Batch normalization accelerates convergence
- However, hard to apply in an adversarial setting
- Collect statistics on a fixed batch of real data and use to normalize other data
Contents

• Story So Far
• Training Issue in GANs
• GANs’ Training and Stabilization
  • Wassertein GANs
• Conditional GANs
• Gan’s Progression
Contents

• Story So Far
• Training Issue in GANs
• GANs’ Training and Stabilization
• Wassertein GANs
• Conditional GANs
• Gan’s Progression
Recap

VAE

KL Divergence

GANs

JS Divergence
KL-DIVERGENCE

$$KL(p\|q) = \int_x p(x) \log \frac{p(x)}{q(x)}$$

Let $\theta$ be the distance between the two peaks of the distribution

- If $\theta \neq 0$, $KL(P\|Q) = 1 \ \log(1/0) = \infty$
- If $\theta = 0$, $KL(P\|Q) = 1 \ \log(1/1) = 0$

Not differentiable w.r.t $\theta$
JENSON-SHANON DIVERGENCE

Let $\theta$ be the distance between the two peaks of the distribution

If $\theta \neq 0$, $JSD(P||Q) = 0.5 \times \left( \log\left(\frac{1}{0.5}\right) + \log\left(\frac{1}{0.5}\right) \right) = \log 4$

If $\theta = 0$, $JSD(P||Q) = 0.5 \times \left( \log\left(\frac{1}{1}\right) + \log\left(\frac{1}{1}\right) \right) = 0$

Not differentiable w.r.t $\theta$
Both KLD and JSD do not tell how far we currently are w.r.t. the true distribution.

And by the way, they are not differentiable w.r.t. the distance $\theta$

And we desire something could tell us how far we currently are w.r.t. the true distribution.

And maybe differentiable w.r.t. the distance $\theta$
WASSERSTEIN DISTANCE

• The distance between probability distributions

• Intuitively: Minimum cost of turning one pile of dirt into another pile of dirt, when both distributions are treated as pile of dirt.

• The total $\Sigma$ mass $\times$ mean distance required to transform one distribution to another

Red points, Blue points represent two different distributions.
WASSERSTEIN DISTANCE

\[ W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y) \sim \gamma} \left[ \| x - y \| \right] \]

Red points, Blue points represent two different distributions.
WASSERSTEIN DISTANCE

\[ W(P, Q) = | \theta | \]

Differentiable w.r.t \( \theta \) !!
WASSERSTEIN (EM) VS JSD

- Distance value is not constant for non-overlapping distributions
- Differentiable w.r.t $\theta$

Figure 1: These plots show $\rho(P_\theta, P_0)$ as a function of $\theta$ when $\rho$ is the EM distance (left plot) or the JS divergence (right plot). The EM plot is continuous and provides a usable gradient everywhere. The JS plot is not continuous and does not provide a usable gradient.
Story So Far

- VAE
  - KL Divergence
- GANs
  - Jenson-Shanon Divergence
- WGANs
  - Wasserstein Distance
WGAN

\[
\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{x \sim P_r} [D(x)] - \mathbb{E}_{\tilde{x} \sim P_g} [D(\tilde{x})]
\]

Kantorovich-Rubinstein duality

D should be a 1-Lipschitz function:
A function is K-Lipschitz if its gradients are at most K everywhere.

Done by weight clipping:
Restrict weights between [-c, c]
WGAN-GP

\[ L = \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)] + \lambda \mathbb{E}_{\hat{x} \sim \mathbb{P}_\hat{x}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]. \]

Original critic loss  
Our gradient penalty

Gradient penalty introduces a softer constraint on gradients
Contents

• Story So Far

• Training Issue in GANs

• GANs’ Training and Stabilization

• Wassertein GANs

• **Conditional GANs**

• Gan’s Progression
Story So Far

What’s the input of Generator?
Story So Far

What’s the input of Generator?
Z ~ Pz
Supposed we have got trained a vanilla GAN/ WGAN; Luckily it works and generate great results;

I want to use the generator to generate my selfie, what to do?
Intuitively, the Discriminator wants to will only give the real data that fit the condition information high value;

The Generator wants to generate fake data that fool the discriminator.

What should be the condition information $y$?

\[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z|y)))).
\]
Conditional GAN

Quite flexible:
- $y \rightarrow$ one-hot, real images;
- $y$’s representation;
- Output of discriminator; $\rightarrow$ one score / two scores

Applications:
- Text-to-image
- Image-to-image
- Speech Enhancement
- Video Generation
Contents

• Story So Far
• Training Issue in GANs
• GANs’ Training and Stabilization
• Wassertein GANs
• Conditional GANs
• Gan’s Progression
GANS PROGRESSION

- Better quality
- High Resolution

https://twitter.com/goodfellow_ian/status/1084973596236144640?lang=en
GANS PROGRESSION

Original GANs Paper

GANS PROGRESSION

- Original GANs Paper
- LapGAN
- Conditional GANs
GANS PROGRESSION
GANS PROGRESSION

- Original GANs Paper
- LapGAN
- Improved Techniques for Training GANS
- Conditional GANs
- DCGAN

2014 - 2020
GANS PROGRESSION

- Original GANs Paper
- LapGAN
- Improved Techniques for Training GANS
- WGAN-GP
- Conditional GANs
- WGAN
- DCGAN
GANS PROGRESSION

- Original GANs Paper
- LapGAN
- Improved Techniques for Training GANS
- WGAN-GP
- Conditional GANs
- WGAN
- DCGAN
- STARGAN
QUESTIONS?