GENERATIVE ADVERSARIAL NETWORKS - PART II

11785- Introduction to Deep Learning

AKSHAT GUPTA
Spring 2021

Slides Inspired by Benjamin Striner
• GANs Recap
• Understanding Training Issue in GANs
• GAN Training and Stabilization
• Wasserstein GANs
• GANstory - GAN Architectures
CONTENTS

• GANS RECAP
  • Understanding Training Issue in GANs
  • GAN Training and Stabilization
  • Wasserstein GANs
  • GANstory - GAN Architectures
WHAT ARE GANS?

Generative Models
We try to learn the underlying the distribution from which our dataset comes from.
Eg: Variational AutoEncoders (VAE)

Adversarial Training
GANS are made up of two competing networks (adversaries) that are trying beat each other.

GOAL: Generate data from an unlabelled distribution.
WHAT CAN GANS DO?

- Data Augmentation
- Image-to-Image Translation
- Text-to-Image Synthesis
- Single Image Super Resolution
HOW TO TRAIN A GAN?

At $t = 0$, 

- Latent Vector $\rightarrow$ Generator $\rightarrow$ Generated Image (fake image)

- (fake data) $\rightarrow$ Generated Data $\rightarrow$ Discriminator $\rightarrow$ Real/Fake?

- (Real data) $\rightarrow$ Given Training Data $\rightarrow$ Discriminator $\rightarrow$ Real/Fake?

- Binary Classifier
HOW TO TRAIN A GAN?

Which network should I train first?
HOW TO TRAIN A GAN?

Which network should I train first?

Discriminator!
HOW TO TRAIN A GAN?

Which network should I train first?

Discriminator!

But with what training data?
How to train a GAN?

Which network should I train first?

Discriminator!

But with what training data?

The Discriminator is a Binary classifier.
The Discriminator has two class - Real and Fake.
The data for Real class if already given: THE TRAINING DATASET
The data for Fake class? -> generate from the Generator
HOW TO TRAIN A GAN?

What’s next? -> Train the Generator

But how? What’s our training objective?
HOW TO TRAIN A GAN?

What’s next? -> Train the Generator

But how? What’s our training objective?

**Generate images from the Generator**
**such that they are classified incorrectly by the Discriminator!**
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?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D = \text{actual data distribution}$
$P_G = \text{generated data distribution}$
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D =$ actual data distribution
$P_G =$ generated data distribution

$D(X) : \text{Output of the discriminator / Probability that } X \text{ came from actual data distribution } P_D$
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D = \text{actual data distribution}$
$P_G = \text{generated data distribution}$

$D(X) : \text{Output of the discriminator / Probability that } X \text{ came from actual data distribution } P_D$
$G(Z) : \text{Output of the generator/A point from the generated data distribution } P_G$
We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D = \text{actual data distribution}$
$P_G = \text{generated data distribution}$

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

$G(Z)$: Output of the generator/A point from the generated data distribution $P_G$

If $X \sim P_D$, what should happen to the value of $D(X)$?
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D = \text{actual data distribution}$
$P_G = \text{generated data distribution}$

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

$G(Z)$: Output of the generator / A point from the generated data distribution $P_G$

If $X \sim P_D$, what should happen to the value of $D(X)$?

It should be maximized!
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D = \text{actual data distribution}$
$P_G = \text{generated data distribution}$

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

$G(Z)$ : Output of the generator/A point from the generated data distribution $P_G$

If $X \sim P_D$, what should happen to the value of $D(X)$?

It should be maximized!

⇒ $D(X)$ should be maximized
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D$ = actual data distribution
$P_G$ = generated data distribution

$D(X)$ : Output of the discriminator / Probability that $X$ came from actual data distribution $P_D$
$G(Z)$ : Output of the generator / A point from the generated data distribution $P_G$

If $X \sim P_D$, what should happen to the value of $D(X)$?

It should be maximized!

$\Rightarrow D(X)$ should be maximized
$\Rightarrow \log(D(X))$ should be maximized
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D =$ actual data distribution
$P_G =$ generated data distribution

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

$G(Z) :$ Output of the generator / A point from the generated data distribution $P_G$

If $X \sim P_D,$ what should happen to the value of $D(X)?$

It should be maximized!

$\Rightarrow D(X)$ should be maximized

$\Rightarrow \log(D(X))$ should be maximized

$\Rightarrow E_{X \sim P_D}[\log(D(X))]$ should be maximized
HOW TO TRAIN A GAN?

We represent the discriminator by \( D(X; \theta) \)
We represent the generator by \( G(Z; \theta) \)

\( P_D = \) actual data distribution  
\( P_G = \) generated data distribution

\( D(X) \) : Output of the discriminator / Probability that \( X \) came from actual data distribution \( P_D \)
\( G(Z) \) : Output of the generator / A point from the generated data distribution \( P_G \)

If \( X \sim P_D \), what should happen to the value of \( D(X) \)?

It should be maximized!

\( \Rightarrow D(X) \) should be maximized
\( \Rightarrow \log(D(X)) \) should be maximized
\( \Rightarrow E_{X \sim P_D}[\log(D(X))] \) should be maximized

Chances of real data being called real.
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$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) : $ Output of the generator / A point from the generated data distribution $P_G$

If $X = G(Z)$, i.e. $X \sim P_G$, what should happen to the value of $D(X)$?
HOW TO TRAIN A GAN?

We represent the discriminator by \( D(X; \theta) \)
We represent the generator by \( G(Z; \theta) \)

\( P_D = \) actual data distribution
\( P_G = \) generated data distribution

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

\( G(Z) \) : Output of the generator / A point from the generated data distribution \( P_G \)

If \( X = G(Z) \), i.e. \( X \sim P_G \), what should happen to the value of \( D(X) \)?

It should be minimized!
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D = \text{actual data distribution}$
$P_G = \text{generated data distribution}$

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

$G(Z)$ : Output of the generator/A point from the generated data distribution $P_G$

If $X = G(Z)$, i.e. $X \sim P_G$, what should happen to the value of $D(X)$?

It should be minimized!

$\Rightarrow D(X)$ should be minimized

$\Rightarrow \log(D(X))$ should be minimized
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$p_D = \text{actual data distribution}$
$p_G = \text{generated data distribution}$

$D(X) : \text{Output of the discriminator / Probability that } X \text{ came from actual data distribution } p_D$
$G(Z) : \text{Output of the generator / A point from the generated data distribution } p_G$

If $X = G(Z)$, i.e. $X \sim p_G$, what should happen to the value of $D(X)$?

It should be minimized!

$\Rightarrow D(X)$ should be minimized
$\Rightarrow \log(D(X))$ should be minimized
$\Rightarrow \log(1 - D(X))$ should be maximized
HOW TO TRAIN A GAN?

We represent the discriminator by \( D(X; \theta) \)
We represent the generator by \( G(Z; \theta) \)

\[ P_D = \text{actual data distribution} \]
\[ P_G = \text{generated data distribution} \]

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

\( G(Z) \) : Output of the generator / A point from the generated data distribution \( P_G \)

If \( X = G(Z) \), i.e. \( X \sim P_G \), what should happen to the value of \( D(X) \)?

It should be minimized!

\[ \Rightarrow D(X) \text{ should be minimized} \]
\[ \Rightarrow \log(D(X)) \text{ should be minimized} \]
\[ \Rightarrow \log(1-D(X)) \text{ should be maximized} \]
\[ \Rightarrow E_{X \sim P_G}[\log(1-D(X))] \text{ should be maximized} \]
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$PD$ = actual data distribution
$PG$ = generated data distribution
$Pz$ = chosen prior in latent vector space

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

$G(Z)$ : Output of the generator/A point from the generated data distribution $PG$

If $X = G(Z)$, i.e. $X \sim PG$, what should happen to the value of $D(X)$?

It should be minimized!

$\Rightarrow D(X)$ should be minimized

$\Rightarrow \log(D(X))$ should be minimized

$\Rightarrow \log(1-D(X))$ should be maximized

$\Rightarrow E_{X \sim PD}[\log(1-D(X))]$ should be maximized

$\Rightarrow E_{Z \sim Pz}[\log(1-D(G(Z)))]$ should be maximized

Chances of fake data being called fake.
HOW TO TRAIN A GAN?

We represent the discriminator by \( D(X; \theta) \)
We represent the generator by \( G(Z; \theta) \)

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

\( G(Z) \) : Output of the generator/A point from the generated data distribution \( P_G \)

If \( X \sim P_D \), what should happen to the value of \( D(X) \)?
\[
\Rightarrow E_{X \sim P_D}[\log(D(X))] \text{ should be maximized}
\]

If \( X = G(Z) \), i.e. \( X \sim P_G \), what should happen to the value of \( D(X) \)?
\[
\Rightarrow E_{Z \sim P_Z}[\log(1 - D(G(Z)))] \text{ should be maximized}
\]
HOW TO TRAIN A GAN?

We represent the discriminator by \( D(X; \theta) \)
We represent the generator by \( G(Z; \theta) \)

\( D(X) : \) Output of the discriminator / Probability that \( X \) came from actual data distribution \( P_D \)
\( G(Z) : \) Output of the generator/A point from the generated data distribution \( P_G \)

If \( X \sim P_D \), what should happen to the value of \( D(X) \)?
\[ \Rightarrow \mathbb{E}_{X \sim P_D}[\log(D(X))] \] should be maximized

If \( X = G(Z) \), i.e. \( X \sim P_G \), what should happen to the value of \( D(X) \)?
\[ \Rightarrow \mathbb{E}_{Z \sim P_z}[\log(1 - D(G(Z)))] \] should be maximized

\[ \Rightarrow \text{The discriminator should maximize this sum:} \]
\[ V(D,G) = \mathbb{E}_{X \sim P_D}[\log(D(X))] + \mathbb{E}_{Z \sim P_z}[\log(1 - D(G(Z)))] \]
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D =$ actual data distribution
$P_G =$ generated data distribution
$P_z =$ chosen prior in latent vector space

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

$G(Z) : $ Output of the generator/A point from the generated data distribution $P_G$

The discriminator maximizes this sum:

$$V(D,G) = \mathbb{E}_{X \sim P_D} [\log(D(X))] + \mathbb{E}_{Z \sim P_z} [\log(1 - D(G(Z)))]$$

Chances of real data being called real.

Chances of fake data being called fake.
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$D(X)$ : Output of the discriminator / Probability that $X$ came from actual data distribution $P_D$
$G(Z)$ : Output of the generator/A point from the generated data distribution $P_G$

Given that the discriminator maximizes this sum: $V(D,G) = E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_z}[\log(1 - D(G(Z)))]$

What should the generator do?
HOW TO TRAIN A GAN?

We represent the discriminator by \( D(X; \theta) \)
We represent the generator by \( G(Z; \theta) \)

\( P_D = \) actual data distribution
\( P_G = \) generated data distribution
\( P_z = \) chosen prior in latent vector space

\( D(X) : \) Output of the discriminator / Probability that \( X \) came from actual data distribution \( P_D \)
\( G(Z) : \) Output of the generator/A point from the generated data distribution \( P_G \)

Given that the discriminator maximizes this sum:
\[
V(D,G) = \mathbb{E}_{X \sim P_D}[\log(D(X))] + \mathbb{E}_{Z \sim P_z}[\log(1 - D(G(Z)))]
\]

What should the generator do?

**Generate images from the Generator**
**such that they are classified incorrectly by the Discriminator!**
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D$ = actual data distribution
$P_G$ = generated data distribution
$P_z$ = chosen prior in latent vector space

$D(X)$ : Output of the discriminator / Probability that $X$ came from actual data distribution $P_D$
$G(Z)$ : Output of the generator / A point from the generated data distribution $P_G$

Given that the discriminator maximizes this sum:

$V(D,G) = \mathbb{E}_{X \sim P_D}[\log(D(X))] + \mathbb{E}_{Z \sim P_z}[\log(1 - D(G(Z))))]$

What should the generator do?

Generate images from the Generator such that they are classified incorrectly by the Discriminator!

$\Rightarrow D(G(Z)))$ should be maximized
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D = \text{actual data distribution}$
$P_G = \text{generated data distribution}$
$P_z = \text{chosen prior in latent vector space}$

$D(X) : \text{Output of the discriminator / Probability that } X \text{ came from actual data distribution } P_D$

$G(Z) : \text{Output of the generator / A point from the generated data distribution } P_G$

Given that the discriminator maximizes this sum:

$$V(D,G) = E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_z}[\log(1-D(G(Z)))]$$

What should the generator do?

**Generate images from the Generator**
**such that they are classified incorrectly by the Discriminator!**

$\Rightarrow D(G(Z)))$ should be maximized
$\Rightarrow \log(D(G(Z)))$ should be maximized
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D =$ actual data distribution
$P_G =$ generated data distribution
$P_z =$ chosen prior in latent vector space

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

$G(Z):$ Output of the generator / A point from the generated data distribution $P_G$

Given that the discriminator maximizes this sum:

$$V(D,G) = \mathbb{E}_{X \sim P_D}[\log(D(X))] + \mathbb{E}_{Z \sim P_z}[\log(1 - D(G(Z)))]$$

What should the generator do?

**Generate images from the Generator such that they are classified incorrectly by the Discriminator!**

$\Rightarrow D(G(Z)))$ should be maximized

$\Rightarrow \log(D(G(Z))))$ should be maximized

$\Rightarrow \log(1 - D(G(Z))))$ should be minimized
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$P_D = \text{actual data distribution}$
$P_G = \text{generated data distribution}$
$P_z = \text{chosen prior in latent vector space}$

$D(X) : \text{Output of the discriminator} / \text{Probability that } X \text{ came from actual data distribution } P_D$

$G(Z) : \text{Output of the generator} / \text{A point from the generated data distribution } P_G$

Given that the discriminator maximizes this sum: $V(D,G) = E_{X \sim P_D}[ \log(D(X))] + E_{Z \sim P_z}[ \log(1-D(G(Z)))]$

What should the generator do?

**Generate images from the Generator**
**such that they are classified incorrectly by the Discriminator!**

$\Rightarrow D(G(Z)))$ should be maximized

$\Rightarrow \log(D(G(Z))))$ should be maximized

$\Rightarrow \log(1 - D(G(Z))))$ should be minimized

$\Rightarrow E_{Z \sim P_z}[ \log(1-D(G(Z)))]$ should be minimized
HOW TO TRAIN A GAN?

We represent the discriminator by \( D(X; \theta) \)
We represent the generator by \( G(Z; \theta) \)

\( D(X) \): Output of the discriminator / Probability that \( X \) came from actual data distribution \( P_D \)
\( G(Z) \): Output of the generator / A point from the generated data distribution \( P_G \)

\( P_D = \) actual data distribution
\( P_G = \) generated data distribution
\( P_z = \) chosen prior in latent vector space

Given that the discriminator maximizes this sum:

\[
V(D,G) = \mathbb{E}_{X \sim P_D} [\log(D(X))] + \mathbb{E}_{Z \sim P_z} [\log(1 - D(G(Z)))]
\]

What should the generator do?

**Generate images from the Generator such that they are classified incorrectly by the Discriminator!**

\( \Rightarrow D(G(Z)) \) should be maximized
\( \Rightarrow \log(D(G(Z))) \) should be maximized
\( \Rightarrow \log(1 - D(G(Z))) \) should be minimized
\( \Rightarrow \mathbb{E}_{Z \sim P_z} [\log(1 - D(G(Z)))] \) should be minimized

Chances of fake data being called fake.
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \phi)$

$D(X)$ : Output of the discriminator / Probability that $X$ came from actual data distribution $P_D$
$G(Z)$ : Output of the generator/A point from the generated data distribution $P_G$

$V(D,G) = \mathbb{E}_{X \sim P_D}[\log(D(X))] + \mathbb{E}_{Z \sim P_z}[\log(1-D(G(Z)))]$

Given that the discriminator maximizes this sum:

What should the generator do?

**Generate images from the Generator such that they are classified incorrectly by the Discriminator!**

$\Rightarrow D(G(Z)))$ should be maximized
$\Rightarrow \log(D(G(Z)))$ should be maximized
$\Rightarrow \log(1- D(G(Z)))$ should be minimized
$\Rightarrow \mathbb{E}_{Z \sim P_z}[\log(1-D(G(Z)))]$ should be minimized
$\Rightarrow \mathbb{E}_{X \sim P_D}[\log(D(X))] + \mathbb{E}_{Z \sim P_z}[\log(1-D(G(Z)))]$ should be minimized
HOW TO TRAIN A GAN?

We represent the discriminator by $D(X; \theta)$
We represent the generator by $G(Z; \theta)$

$D(X) : \text{Output of the discriminator / Probability that } X \text{ came from actual data distribution } P_D$
$G(Z) : \text{Output of the generator/A point from the generated data distribution } P_G$

$P_D = \text{actual data distribution}$
$P_G = \text{generated data distribution}$
$P_z = \text{chosen prior in latent vector space}$

So, in your GAN formulation:

The discriminator maximizes this sum: $V(D,G) = E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_z}[\log(1-D(G(Z)))]$

The generator minimizes this sum: $V(D,G) = E_{X \sim P_D}[\log(D(X))] + E_{Z \sim P_z}[\log(1-D(G(Z)))]$
HOW TO TRAIN A GAN?

We represent the discriminator by \( D(X; \theta) \)
We represent the generator by \( G(Z; \theta) \)

\( P_D = \) actual data distribution
\( P_G = \) generated data distribution
\( P_z = \) chosen prior in latent vector space

\( D(X) : \) Output of the discriminator / Probability that \( X \) came from actual data distribution \( P_D \)
\( G(Z) : \) Output of the generator / A point from the generated data distribution \( P_G \)

So, in your GAN formulation:

The discriminator maximizes this sum:
\[
V(D,G) = \mathbb{E}_{X \sim P_D} \left[ \log(D(X)) \right] + \mathbb{E}_{Z \sim P_z} \left[ \log(1 - D(G(Z))) \right]
\]

The generator minimizes this sum:
\[
V(D,G) = \mathbb{E}_{X \sim P_D} \left[ \log(D(X)) \right] + \mathbb{E}_{Z \sim P_z} \left[ \log(1 - D(G(Z))) \right]
\]

Chances of real data being called real.
Chances of fake data being called fake.
ORIGINAL GAN FORMULATION
The original GAN formulation is the following min-max game

$$\min_G \max_D V(D, G) = \mathbb{E}_X \log D(X) + \mathbb{E}_Z \log(1 - D(G(Z)))$$

- $D$ wants $D(X) = 1$ and $D(G(Z)) = 0$
- $G$ wants $D(G(Z)) = 1$
THE OPTIMAL DISCRIMINATOR

\[ P_D = \text{actual data distribution} \]
\[ P_G = \text{generated data distribution} \]
\[ D(X) = \text{discriminator output} \]

Objective: \( \min_D \max_G V(D, G) = \mathbb{E}_X \log D(X) + \mathbb{E}_Z \log(1 - D(G(Z))) \)

What is the optimal discriminator?

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

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

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

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

\[ D(X) = \frac{P_D(X)}{P_G(X) + P_D(X)} \]
THE OPTIMAL GENERATOR

Objective: \( \min_G \max_D V(D, G) = \mathbb{E}_X \log D(X) + \mathbb{E}_Z \log(1 - D(G(Z))) \)

\[= \mathbb{E}_{X \sim P_D} \log D(X) + \mathbb{E}_{X \sim P_G} \log(1 - D(X)) \]
\[= \mathbb{E}_{P_D} \log \frac{P_D(X)}{P_G(X) + P_D(X)} + \mathbb{E}_{P_G} \log \frac{P_D(X)}{P_G(X) + P_D(X)} \]
\[= \text{JSD}(P_D|P_G) - \log 4 \]
THE OPTIMAL GENERATOR

What is the optimal generator?

$$\min_G JSD(P_D \| P_G) - \log 4$$

Minimize the Jensen-Shannon divergence between the real and generated distributions (make the distributions similar)
MIN-MAX STATIONARY POINT

- Stationary points need not be stable (depends on the exact GANs formulation and other factors)
CONTENTS

• GANs Recap

• UNDERSTANDING TRAINING ISSUES IN GANS

• GAN Training and Stabilization

• Wasserstein GANs

• GANstory - GAN Architectures
WHY IS THERE NO STATIC OPTIMAL DISCRIMINATOR?

\[
\min_D \max_G V(D, G) = \mathbb{E}_X \log D(X) + \mathbb{E}_Z \log(1 - D(G(Z)))
\]

• Discriminator indicates the direction in which generator should move relative to the current generator

• For a given fixed discriminator, the optimal generator outputs \( \text{argmax } D(X) \) for all \( z \sim Z \)

• Cannot train generator without training discriminator first
CAUSES OF OPTIMIZATION ISSUES

• Simultaneous updates require a careful balance between players

• Stationary point exists but there’s no guarantee of reaching it

• If discriminator is undertrained, it guides the generator in the wrong direction

• If discriminator is overtrained, it is too hard and generator cannot make much progress
FACTORS AFFECTING ADVERSARIAL BALANCE

- Different optimizers, learning rates, batch size
- Different architectures, depths, number of parameters
- Training discriminator and generator for different number of iterations
ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

CASE - 1:1 play rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)

• What is your best strategy?

• What is your probability of winning?
ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

CASE - II: I play rock-paper-scissors with a probability of

(0.33, 0.33, 0.33)

• What is your optimal strategy?

• What is your probability of winning?
Player A plays rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)
ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

(0.36, 0.32, 0.32)

• GLOBAL OPTIMUM: Both players play uniformly with (0.33, 0.33, 0.33)
Player A plays rock-paper-scissors with a probability of 

\((0.36, 0.32, 0.32)\)

- If player B optimizes all the way, its optimal strategy is always paper \((0, 1, 0)\)
ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

\[(0.36, 0.32, 0.32)\]

- If player B optimizes all the way, its optimal strategy is always paper \((0,1,0)\)
- Now player A should play only scissors \((0,0,1)\)
ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

\[(0.36, 0.32, 0.32)\]

• If player B optimizes all the way, its optimal strategy is always paper \((0,1,0)\)

• Now player A should play only scissors \((0,0,1)\)

• Now player B should only play rock \((1,0,0)\)
ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of 

\[(0.36, 0.32, 0.32)\]

- If player B optimizes all the way, its optimal strategy is always paper \((0, 1, 0)\)
- Now player A should play only scissors \((0, 0, 1)\)
- Now player B should only play rock \((1, 0, 0)\)
- Now player A should only play paper \((0, 1, 0)\)
ADVERSARIAL BALANCE IN TWO PLAYER GAMES: ROCK-PAPER-SCISSORS

Player A plays rock-paper-scissors with a probability of

$(0.36, 0.32, 0.32)$

- If player B optimizes all the way, its optimal strategy is always paper.
- Now player A should play only scissors
- Now player B should only play rock
- Now player A should only play paper
- ……
TRAINING ISSUES IN GAS

• Oscillations

• Mode Collapse: Generates a small subspace but does not cover the entire distribution (https://www.youtube.com/watch?v=ktxhiKhWoEE)
CONTENTS

• GANs Recap

• Understanding Training Issue in GANs

• GAN TRAINING AND STABILIZATION

• Wasserstein GANs

• GANstory - GAN Architectures
IMPROVED TECHNIQUES FOR TRAINING GANS (2016)

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

$$\| \mathbb{E}_x D(X) - \mathbb{E}_z D(G(Z)) \|^2_2$$
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
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

• GANs Recap

• Understanding Training Issue in GANs

• GAN Training and Stabilization

• WASSERSTEIN GANS

• GANstory - GAN Architectures
WASSERSTEIN DISTANCE

• The distance between probability distributions

• Intuitively, each distribution is viewed as a unit amount of earth (soil)

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

• Also called earth mover’s distance

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

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

Red points, Blue points represent two different distributions.
THE GAME OF DISTANCE MEASURES
THE GAME OF DISTANCE MEASURES

VAE
THE GAME OF DISTANCE MEASURES

VAE

KL Divergence
THE GAME OF DISTANCE MEASURES

VAE

KL Divergence

GANs
THE GAME OF DISTANCE MEASURES

VAE

KL Divergence

GANs

Jenson-Shannon Divergence
THE GAME OF DISTANCE MEASURES

- VAE
  - KL Divergence
- GANs
  - Jenson-Shannon Divergence
- WGANs
  - Wasserstein Distance
KL-DIVERGENCE

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

If $\theta \neq 0$, $KL(P\|Q) = \int_x p(x) \log \frac{p(x)}{q(x)} = \infty$

If $\theta = 0$, $KL(P\|Q) = \int_x p(x) \log \frac{p(x)}{q(x)} = 0$

Not differentiable w.r.t $\theta$
Let $\theta$ be the distance between the two peaks of the distribution.

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

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

Not differentiable w.r.t $\theta$
WASSERSTEIN DISTANCE

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

Differentiable w.r.t \( \theta \)!!
Figure 1: These plots show $\rho(\mathbb{P}_\theta, \mathbb{P}_0)$ 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.
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.*
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
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
WGAN

\[
\min_G \max_D \mathbb{E}_{x \sim P_r} \left[ D(x) \right] - \mathbb{E}_{\tilde{x} \sim P_g} \left[ D(\tilde{x}) \right]
\]

Kantorovich-Rubinstein duality

D should be a 1-Lipschitz function

Weight clipping:
• Restrict weights between [-c, c]
A function is 1-Lipschitz if its gradients are at most 1 everywhere.
A function is 1-Lipschitz if its gradients are at most 1 everywhere.

Gradient penalty introduces a softer constraint on gradients.
CONTENTS

• GANs Recap

• Understanding Training Issue in GANs

• GAN Training and Stabilization

• Wasserstein GANs

• GANSTORY - GAN ARCHITECTURES
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

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

- Original GANs Paper
- Conditional GANs
- DCGAN
- LapGAN

Timeline:
- 2014
- 2015
- 2016
- 2017
- 2018
- 2019
- 2020
GANS PROGRESSION

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

GANS PROGRESSION

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

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

- **Original GANs Paper**
  - 2014

- **Conditional GANs**
  - 2015

- **LapGAN**
  - 2016

- **Improved Techniques for Training GANS**
  - 2017

- **WGAN**
  - 2018

- **WGAN-GP**
  - 2019

- **STARGAN**
  - 2020
GANS PROGRESSION

- Original GANs Paper (2014)
- Conditional GANs
- LapGAN (2015)
- DCGAN
- Improved Techniques for Training GANS (2016)
- WGAN (2017)
- WGAN-GP (2018)
- STARGAN
- BigGAN (2019)
- Improved Techniques for Training GANs (2020)
QUESTIONS?