

Generative Adversarial Networks - Part 1

11785 - Introduction to Deep Learning Fall 2021

By Manish Mishra and Zhe Chen

Slides Inspired by Akshat Gupta and Benjamin Striner



"This (GANS), and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion"

-Yann LeCun



DeepFake head-swapped Video

<u>https://www.youtube.com/watch?v=34AmKPJNfCg</u>



Contents

- Motivation
- Generative vs Discriminative Models
- GANs vs VAEs
- GANs Introduction
- GANs Theory
- GANs Evaluation



Contents

- Motivation
- Generative vs Discriminative Models
- GANs vs VAEs
- GANs Introduction
- GANs Theory
- GANs Evaluation



Motivation

Generative networks are used to generate samples from an unlabeled dsitribution P(X) given samples X1, ...Xn. For example:

- Learn to generate realistic images given exemplary images
- Learn to generate realistic music given exemplary recordings
- Learn to generate realistic text given exemplary corpus

Let's see some amazing work of GANs



Original paper (GAN, 2014)

Output of original GAN paper, 2014 [GPM⁺14]





c)



7



GANs progression

- Better quality
- High Resolution



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



StarGAN(2018)

Manipulating Celebrity Faces [CCK+17]



Figure 1. Multi-domain image-to-image translation results on the CelebA dataset via transferring knowledge learned from the RaFD dataset. The first and sixth columns show input images while the remaining columns are images generated by StarGAN. Note that the images are generated by a single generator network, and facial expression labels such as angry, happy, and fearful are from RaFD, not CelebA.



Progressive growing of GANs (2018)



Figure 5: 1024×1024 images generated using the CELEBA-HQ dataset. See Appendix F for a larger set of results, and the accompanying video for latent space interpolations.



High fidelity natural images (2019)

Generating High-Quality Images [BDS18]





Contents

Motivation

Generative vs Discriminative Models

- GANs vs VAEs
- GANs Introduction
- GANs Theory
- GANs Evaluation

Discriminative vs Generative Models



Given a distribution of inputs X and labels Y.

Discriminative models

learn conditional distribution P(Y | X)

Generative models learn the joint distribution P(Y, X)

Discriminative vs Generative Models



Given a distribution of inputs X and labels Y.

Discriminative models

- Discriminative models learn conditional distribution P(Y | X)
- Learns decision boundary between classes.
- Limited scope. Can only be used for classification tasks.
- E.g. Logistic regression, SVM etc.

Generative models

- Generative models learn the Generative models learn the joint distribution P(Y, X)
- Learns actual probability distribution of data.
- Can do both generative and discriminative tasks.
- E.g. Naïve Bayes, Gaussian Mixture Model etc.

Harder problem, requires a deeper understanding of the distribution than discriminative models.



Explicit vs Implicit Models

Explicit distribution models

• Calculates $P(x \sim X)$ for all x

Implicit distribution models

• Generate x ~ X



Contents

- Motivation
- Generative vs Discriminative Models
- GANs vs VAEs
- GANs Introduction
- GANs Theory
- GANs Evaluation



Variational autoencoders (VAE)

- Encoder models P(Z|X)
- Decoder models P(X|Z)
- Loss encourages $P(Z|X) \sim Q(Z)$



VAEs vs GANs



VAEs

- Minimizing the KL-divergence
- Minimize a <u>bound</u> on the divergence between generated distribution and target distribution
- Simpler optimization. Trains faster and more reliably
- Results are blurry

GANs

- Minimizing the Jenson-Shannon Divergence
- Minimize divergence between generated distribution and target distribution
- Noisy and difficult optimization
- Sharper results



Contents

- Motivation
- Generative vs Discriminative Models
- GANs vs VAEs
- GANs Introduction
- GANs Theory
- GANs Evaluation



Generative Adversarial Networks



Generative Adversarial Networks

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



Generative Adversarial Networks

Generative Models

We try to learn the underlying the distribution from which our dataset comes from. E.g. Variational Autoencoders (VAE)







Generative Models

We try to learn the underlying the distribution from which our dataset comes from. E.g. Variational Autoencoders (VAE) **Neural Networks**

Adversarial Training

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

Goal: Generate data from an unlabeled distribution.











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



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!





Discriminator

Generator

Step 1: Train the Discriminator <u>using the current ability</u> of the Generator.



Step 2: Train the Generator to beat the Discriminator.

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



Contents

- Motivation
- Generative vs Discriminative Models
- GANs vs VAEs
- GANs Introduction
- GANs Theory
- GANs Evaluation



Generative Adversarial Networks

- Introduced in 2014
- Goal is to model P(X), the distribution of training data
- Model can generate samples from P(X)
- Trained using a pair of models acting as "adversaries"



The Generator

- The generator learns P(X|Z) : Produces realistic looking data X from a latent vector Z
- Z
 - It can be sampled from a known prior, such as a Gaussian
 - It can also be the input to your desired model, such as processed speech segments in Unsupervised Speech Recognition
- Maps a simple known distribution to a complicated data distribution
- Goal: generated distribution, G(z), matches the true data distribution P(X)



The Discriminator

- Trained to tell the difference between real and generated (fake) data
- Loss function / criterion: backpropagates its expectations to the generator
- "Thrown away" after generator is trained



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



What is the optimal discriminator?

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

$$= \int_X [P_D(X) \log D(X) + P_G(X) \log(1 - D(X))] dX$$

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

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

$$(1 - D(X))P_D(X) = D(X)P_G(X)$$

$$D(X) = \frac{P_D(X)}{P_G(X) + P_D(X)}$$

$$D(X) = \text{discriminator output}$$

$$P_D = \text{PDF of actual data distribution}$$

34

 $P_G = PDF$ of generated data distribution



$$D(X) = \frac{P_D(X)}{P_D(X) + P_G(X)}$$

D(X) = discriminator output

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

Case 1: Bad Generator

"There's no way the input X = G(Z) looks like my data"

 $P_D(X) = 0, P_G(X) = 1$ D(X) = 0



$$D(X) = \frac{P_D(X)}{P_D(X) + P_G(X)}$$

D(X) = discriminator output

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

Case 2: Good Generator

"I cannot tell the difference between X = G(Z) and my data"

 $P_D(X) = 1, P_G(X) = 1$ D(X) = 0.5



The Optimal Generator

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

Minimize:

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

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

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

D(X) = discriminator outputG(Z) = generator output

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



Jensen-Shannon Divergence

$$m(X) = \frac{P_D(X) + P_G(X)}{2}$$

$$JSD(P_D||P_G) = \frac{1}{2}KLD(P_D||m(X)) + \frac{1}{2}KLD(P_G||m(X))$$

JSD Properties	KLD Properties $\mathbb{E}_{X \sim P_G(X)} \log \frac{P_G(X)}{P_D(X)}$
Symmetrical	Non-symmetrical
Bounded between 0 and log(2)	No upper bound. Lower bounded by 0

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

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

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



The Optimal Generator

$$\min_{G} 2 \cdot \mathrm{JSD}(P_D || P_G) - \log 4$$

Minimize the Jensen-Shannon divergence between the real and generated distributions

Make the distributions similar



Min-Max Stationary Point

- There exists a stationary point:
 - If the generated data exactly matches the real data, the discriminator outputs 0.5 for all inputs
 - If discriminator outputs 0.5, the gradients for the generator is flat, so generator does not learn
- Stationary points need not be stable (depends on the exact GANs formulation and other factors)
 - Generator may overshoot some values or oscillate around the optimum
 - A discriminator with unlimited capacity can still assign an arbitrarily large distance to 2 similar distributions



Min-Max Optimization

- Generator and the discriminator need to be trained simultaneously
 - If discriminator is undertrained, it provides sub-optimal feedback to the generator
 - If the discriminator is overtrained, there is no local feedback for marginal improvements





Step 1: Train the Discriminator <u>using the current ability</u> of the Generator. Step 2: Train the Generator to beat the Discriminator.

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



GAN Stability

- GANs can be very sensitive to hyperparameters
- There are many variations of GANs that attempt to make the stationary point more stable



https://avg.is.tuebingen.mpg.de/projects/convergence-and-stability-of-gan-training



Perceptual Loss

- Although an idealized discriminator just calculates the JS divergence, a real discriminator calculates something much more complicated
- The discriminator can be a loss function that better gauge similarity from humans' perspective
 - E.g. discriminator loss is shift invariant to inputs if it's a convolutional neural network
 - It can be better than the L₂ loss used in flavors of VAE



The Good, the Bad, and the Ugly

- Good GANs can produce awesome, crisp results for many problems
- Bad GANs have stability issues and open theoretical questions
- Many ugly (ad-hoc) tricks and modifications to get GANs to work correctly



Contents

- Motivation
- Generative vs Discriminative Models
- GANs vs VAEs
- GANs Introduction
- GANs Theory
- GANs Evaluation



GANs Evaluation

- Some tasks like unsupervised speech recognition have a clearly-defined metric
 - An Generator-equivalent model can be trained in a supervised manner
- Other tasks like generating realistic-looking images is not as easily quantified as a task like correctly labeling images
- Possible evaluation methods for generated distributions
 - Human Evaluation
 - Approximate Test Set likelihood
 - Evaluate with Discriminative Network



Human Evaluation

- Expensive, time-consuming, non-reproducible
- Yet maybe the only justifiable way to claim that the generated images are realistic
- Maybe it's not so bad with MechanicalTurk



Evaluate with Discriminative Network

- Inception Score
 - Use a discriminative network (originally based on Inception v3 Architecture) to classify generated images
 - Inception should produce a variety of labels
 - Each label should have high confidence (low entropy)



Questions?