Generative Adversarial Network

Khushboo Thaker
kmt81@pitt.edu
Exponential Growth in GAN Papers

Explosive growth—All the named GAN variants cumulatively since 2014. Credit: Bruno Gavranović

Ian Goodfellow
• Why Generative Modeling?
• Existing Generative Models – A review
• Properties of GAN
• GAN Framework
• Minimax Play for GAN
• Why GAN training is Hard?
• Tricks to train GAN
• Examples of some common extension to GAN
• Conclusion and future reading
Generative Modeling

- Input is Training examples and output is some representation of probability distribution which defines this example space.

  - Un-Supervised
    Data – X
    Goal – Learn Hidden structure of data

- Supervised
  Data – X, y
  Goal – Learn mapping from X -> Y
Why Generative Modeling?

\[ P(X), P(X,Y), P(X|Y) \]

- Noisy Input
- Simulated Data
- Features Representative of Data
- Prediction of Future State
- Missing Data
- Semi-Supervised Learning
Maximum Likelihood based Models

\[ P(x) \]

\[ \theta^* = \arg \max_{\theta} \mathbb{E}_{x \sim P_{data}} \log P(x/\theta) \]

Maximum likelihood tries increase the likelihood of data given the parameters.
Tractable Model - PixelRNN / PixelCNN / WaveNet

Fully visible belief Network

- Generate image pixels from corner
- Training Faster
- Generation Slow / Sequential
- Cannot generate samples based on some latent code

\[
p(x) = \prod_{i=1}^{n} p(x_i | x_1, x_2, ..., x_{i-1})
\]

Maximum Likelihood based Training

Khushboo Thaker
Non Tractable Model - Variational Approximation
Variational Auto-encoder

- Model is able to achieve high likelihood
- Model is not asymptotically consistent unless $q$ is perfect
- Samples tend to have lower quality

\[
\log p(x) \geq \log p(x) - D_{KL}(q(z) \| p(z \mid x)) \\
= E_{z \sim q} \log p(x, z) + H(q)
\]
Non Tractable Model - MCMC Approximation

Boltzmann Machine

- Energy Function based models
- Markov chains don’t work for long sequences
- Hard to scale on large dataset

\[ p(x, h) = \exp\left( -E(x, h) \right) | Z \]

\[ Z = \sum_{x, h} \exp\left( -E(x, h) \right) \]
Where do GANs fall?

• Can Use Latent Information while sample generation
• Asymptotically consistent (claims to recover true distribution)
• No Markov Chain assumption
• Samples produced are high quality
Generated Samples - GAN
Next Video Frame Prediction

- Sharp image
- Better estimation of Ear position
- Much crisp eyes
Generative Adversarial Networks

Generator

Discriminator
Generative Adversarial Networks

Quote from the original paper on GANs:

"The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine articles."

-Goodfellow et. al., "Generative Adversarial Networks" (2014)
**Classic GAN Framework**

\[ x = G(z, \theta^{(G)}) \]

\( Z \) – random Noise
(latent representation of data)
\( Zd \leq Xd \)

Training Discriminator
Training Generator

\[ x = G(z, \theta^{(G)}) \]

[Diagram showing the training process of a generator network with real-world images and a discriminator for adversarial training.]
Mini-max Game Approach

\[
\min_G \max_D - J^D
\]

\[
J^D = -\frac{1}{2} \mathbb{E}_{x \sim P_{\text{data}}} \log D(x) - \frac{1}{2} \mathbb{E}_z \log \left(1 - D(G(z))\right)
\]

\[
J^G = -J^D
\]

- Generator minimizes the log-probability of the discriminator being correct
- Resembles Jensen-Shannon divergence
- Saddle Point of discriminators loss
Mini-max Game Approach

Nash Equilibrium / Saddle Point

\[
\frac{\partial J^D}{\partial D(X)} = 0 \quad \Rightarrow \quad D^*(x) = \frac{P_{\text{data}}(x)}{P_{\text{data}}(x) + P_{\text{model}}(x)}
\]

\[
p_{\text{data}}(x) = P_{\text{model}}(x) \quad \forall x
\]

\[
D^*(x) = \frac{1}{2} \quad \forall x
\]

- Generator minimizes the log-probability of the discriminator being correct
- Resembles Jensen-Shannon divergence
- Saddle Point of discriminators loss

Khushboo Thaker

21
Vanishing Gradient Problem with Generator

\[ J^D = -\frac{1}{2} \mathbb{E}_{x \sim P_{data}} \log D(x) - \frac{1}{2} \mathbb{E}_z \log \left( 1 - D(G(z)) \right) \]

Gradient goes to 0 if D is confident, i.e., \( D(G(z)) \rightarrow 0 \)

As can be seen that whenever the discriminator becomes very confident the loss value will be zero

Nothing to improve for Generator
Heuristic Non Saturating Game

\[ J^D = -\frac{1}{2} \mathbb{E}_{x \sim P_{\text{data}}} \log D(x) - \frac{1}{2} \mathbb{E}_z \log \left( 1 - D(G(z)) \right) \]

\[ J^G = -\frac{1}{2} \mathbb{E}_z \log D(G(z)) \]

Generator maximizes the log probability of the discriminator’s mistake.

Does not change when discriminator is successful.
Comparison of Generator Losses

• Generators cost is a function $D(G(z))$

\[
J^G = -J^D \\
J^G = -\frac{1}{2} \mathbb{E}_z \log D(G(z)) \\
J^G = -\frac{1}{2} \mathbb{E}_z \exp(\sigma^{-1} D(G(z)))
\]
Outline

• Why Generative Modeling?
• Existing Generative Models – A review
• Properties of GAN
• GAN Framework
• Minimax Play for GAN
• Why GAN training is Hard?
• Tricks to train GAN
• Examples of some common extension to GAN
• Conclusion and future reading

https://www.youtube.com/watch?v=mObnwR-u8pc
Why GAN are hard to train?
Non-Convergence

D & G nullifies each others learning in every iteration
Train for a long time – without generating good quality samples

\[ V(x, y) = xy \]
\[ x = 0, \quad y = 0 \]

\[ V(x(t), y(t)) = x(t)y(t) \]
\[ \frac{\partial x}{\partial t} = -y(t) \]
\[ \frac{\partial y}{\partial t} = x(t) \]

\[ \frac{\partial^2 y}{\partial t^2} = \frac{\partial x}{\partial t} = -y(t) \]

\[ x(t) = x(0)\cos(t) - y(0)\sin(t) \]
\[ y(t) = x(0)\cos(t) - y(0)\sin(t) \]

• Differential Equation’s solution has sinusoidal terms
• Even with a small learning rate, it will not converge
• Discrete time gradient descent can spiral outward for large step size
Mode Collapse

http://www.youtube.com/watch?v=ktxhiKhWoEE&t=0m30s

Generator excels in a subspace but does not cover entire real distribution

Why GAN are hard to train?

• Generator keeps generating similar images – so nothing to learn

• Maintain trade-off of generating more accurate vs high coverage samples

• The two learning tasks need to have balance to achieve stability

• If Discriminator is not sufficiently trained – it can worse generator

• If Discriminator is over-trained - will produce no gradients
Tricks to Train GAN

• One sided label smoothing
• Historical generated batches
• Feature Matching
• Batch Normalization
• Regularizing discriminator gradient in region around real data (DRAGAN)
One Side Label Smoothening

• Generator is very sensitive to Discriminators output
• Prevents discriminator to give high gradients
• Does-not reduce accuracy.
• Increase confidence
• Only smooth positive samples

\[ J^D = -\frac{1}{2} \mathbb{E}_{x \sim \text{data}} 0.9 \log D(x) - \frac{1}{2} \mathbb{E}_z \log \left(1 - D(G(z))\right) \]

Historical generated batches

Help stabilize discriminator training at early stages

Don’t Let discriminator forget what it already learned

Feature Matching

\[ \| E_{x \sim \text{pdata}} f(x) - E_{z \sim \text{pmodel}} f(G(z)) \|_2^2 \]

- Generated images must match statistics of real images
- Discriminator defines the statistics
- Generator is trained such that the expected value of statistics matches the expected value of real statistics
- Generator tries to minimize the L2 distance in expected values in some arbitrary space
- Discriminator defines that arbitrary space
Batch Normalization

• Construct different mini-batches for real and fake

• Each mini-batch needs to contain only all real images or all generated images.

• Makes samples within a batch less dependent
DRAGAN

• Failed GANs typically have extreme gradients/sharp peaks around real data

• Regularize GANs to reduce the gradient of the discriminator in a region around real data

\[ \lambda \cdot E_{x \sim p_{data}, \delta \sim N(0,cI)} \left[ ||\Delta_x D(x + \delta)|| - k \right]^2 \]
Few variations of GAN

• Conditional GAN
• LapGAN
• DCGAN
• CatGAN
• InfoGAN
• AAE
• DRAGAN
• IRGAN
Conditional GANs - $P(X | Y)$

- Generator Learns $P(X | Z, Y)$
- Discriminator Learns $P(L | X, Y)$
- Much better samples

$$J^D = -\frac{1}{2} \mathbb{E}_{x \sim P_{data}} \log \mathbb{D}(x|y) - \frac{1}{2} \mathbb{E}_{z} \log \left(1 - \mathbb{D}(G(z|y))\right)$$


Khushboo Thaker 37
Each row is conditioned on a different label. You can use a single neural network to generate all 10 digits by telling it what digit to generate.
• Multiple Convolutional Layers
• Batch Normalization
• Strides with Convolution
• Leaky ReLUs

Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution $Z$ is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a $64 \times 64$ pixel image. Notably, no fully connected or pooling layers are used.
InfoGAN

• Rewards Disentanglement – (individual dimensions capturing key attributes of images)

• $Z$ – partitioned into two parts
  - $z$ – capture slight variation in the images
  - $y$ – captures the main attributes of the images

Mutual Information – maximizing mutual information
Between the code and generator output
\[ \text{InfoGAN} \quad \min_G \max_D V_I(D, G) = V(D, G) - \lambda I(c; G(z, c)) \]

\[ I(c; G(z, c)) = H(c) - H(c|G(z, c)) \]

\[ = E_{x \sim G(z,c)} \left[ D_{KL} (P || Q) + E_{c' \sim p(c|x)} \left[ \log Q(c'|x) \right] \right] + H(c) \]

\[ \geq E_{x \sim G(z,c), c \sim p(c), \log Q(c|x)} + H(c) \]
BiGANs

- Encoder
- Decoder
- Discriminator
LapGANs

- To Scale GAN for large image
- Laplacian pyramid function is used to generate different scales of image

LapGAN

Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution $Z$ is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a $64 \times 64$ pixel image. Notably, no fully connected or pooling layers are used.

DCGAN

- Multiple Convolutional Layers
- Batch Normalization
- Strides with Convolution
- Leaky ReLUs

Figure 4: **Manipulating latent codes on 3D Chairs:** In (a), we show that the continuous code captures the pose of the chair while preserving its shape, although the learned pose mapping varies across different types; in (b), we show that the continuous code can alternatively learn to capture the widths of different chair types, and smoothly interpolate between them. For each factor, we present the representation that most resembles prior supervised results [7] out of 5 random runs to provide direct comparison.
Adversarial Autoencoder (GAN + VAE)
GAN for Text

• GANs for Language Generation (Yu et al. 2017)
• GANs for MT (Yang et al. 2017)
• GANs for Dialogue Generation (Li et al. 2016)
• GANs for fake news detection (Yang et al. 2017)
• GANs for Information Retrieval
GAN and RL connection

• GANs – Inverse Reinforcement Learning
• GANs - Imitate Learning
• GANs – actor critic framework

• REINFORCE - Policy Gradient Based learning
• Gumbel Softmax
Conclusion

• GAN is an active area of research
• GAN architecture is flexible to support variety of learning problems
• GAN does not guarantee to converge
• GAN is able to capture perceptual similarity and generates better images than VAE
• Needs a lot of work in theoretic foundation of Network
• Evaluation of GAN is still an open research (Theis et. al)
Important Papers to dig into GAN

• **NIPS 2016 Tutorial:** - [Ian Goodfellow](#)


• [https://github.com/soumith/ganhacks#authors](https://github.com/soumith/ganhacks#authors)


• [https://www.araya.org/archives/1183](https://www.araya.org/archives/1183)
Startup code, Tools and Tricks

• https://github.com/soumith/ganhacks#authors


• https://jhui.github.io/2017/03/05/Generative-adversarial-models/
References

• Deep Learning Book
• GAN paper: https://arxiv.org/abs/1701.00160
• GAN slides: http://slazebni.cs.illinois.edu/spring17/lec11_gan.pdf
• GAN Tutorial: https://www.youtube.com/watch?v=HGYYEUSm-0Q
Not the end..

Explosive growth—All the named GAN variants cumulatively since 2014. Credit: Bruno Gavranović
Thank You for Listening

Questions ?