Generative Adversarial Networks (GANs)

Bindita Chaudhuri
Unsupervised Learning: Autoencoders

Train such that features can be used to reconstruct original data

\[ \| x - \hat{x} \|^2 \]

Reconstructed input data

Features

Input data

L2 Loss function:

Decoder

Encoder
Unsupervised Learning: Autoencoders

Train such that features can be used to reconstruct original data

L2 Loss function:
$$||x - \hat{x}||^2$$

Reconstructed data

Encoder: 4-layer conv
Decoder: 4-layer upconv

Input data

Reconstructed input data

Features

Decoder

Encoder

Input data

x

z

\hat{x}
Unsupervised Learning: Variational Autoencoders

Sample $x \mid z$ from $x \mid z \sim \mathcal{N}(\mu_{x \mid z}, \Sigma_{x \mid z})$

Decoder network $p_\theta(x \mid z)$

Sample $z$ from $z \mid x \sim \mathcal{N}(\mu_{z \mid x}, \Sigma_{z \mid x})$

Encoder network $q_\phi(z \mid x)$

Input Data $x$
Generative Adversarial Networks: Idea

**Generator**
(Counterfeiter):
Creates fake data from random input
Generative Adversarial Networks: Idea

**Generator** (Counterfeiter): Creates fake data from random input

**Discriminator** (Detective): Distinguish real data from fake data

Looks Fake!

Looks Real!
Generative Adversarial Networks

Real or Fake

Discriminator Network

Fake Images (from generator)

Real Images (from training set)

Generator Network

Random noise Z
Generative Adversarial Networks

Minimax objective function:
\[
\min_{\theta_g} \max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{\text{data}}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log (1 - D_{\theta_d}(G_{\theta_g}(z))) \right]
\]

- Discriminator output for real data \( x \)
- Discriminator output for generated fake data \( G(z) \)
Distributions during training
GAN: Sample Architecture (DC-GAN)
Bidirectional GAN (BiGAN)
Conditional GAN (cGAN)

Generator

Discriminator

Normal distribution $z$

$x = G(c, z)$

$c$: train

$D$: (better)

Scalar
Conditional GAN (cGAN)

Generator

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

Discriminator

True text-image pairs: 

(train, ) 1

(cat, ) 0  (train, image ) 0

 scalar
Pix2Pix: Type of cGAN

L1 loss
CycleGAN: Unsupervised Pix2Pix
CycleGAN: Unsupervised Pix2Pix
CycleGAN Results

Monet ↔ Photos

- Monet → photo
- photo → Monet

Zebras ↔ Horses

- zebra → horse
- horse → zebra

Summer ↔ Winter

- summer → winter
- winter → summer
Progressive Growing of GANs
Progressive GAN Results

Celebrities

Bedrooms

Objects
Application: Neural Style transfer

Image A

Content extractor

Merger

Style extractor

Final result

Image B
Application: 3D GAN

Z

512x4x4x4 → 256x8x8x8 → 128x16x16x16 → 64x32x32x32 → G(z) in 3D Voxel Space 64x64x64