Generative Adversarial Networks (GANs)
Outline

• **Part 1: Introduction to GANs**

• **Part 2: Applications of GANs**
Part 1

- Motivation for Generative Models
- From Adversarial Training to GANs
- GAN’s Architecture
- GAN’s objective
- DCGANs
GANs

- **Generative**
  - Learn a generative model

- **Adversarial**
  - Trained in an adversarial setting

- **Networks**
  - Use Deep Neural Networks
Why Generative Models?

- **We’ve only seen discriminative models so far**
  - Given an image $X$, predict a label $Y$
  - Estimates $P(Y|X)$

- **Discriminative models have several key limitations**
  - Can’t model $P(X)$, i.e. the probability of seeing a certain image
  - Thus, can’t sample from $P(X)$, i.e. can’t generate new images

- **Generative models (in general) cope with all of above**
  - Can model $P(X)$
  - Can generate new images
Magic of GANs...

Which one is Computer generated?

Magic of GANs...

User edits

Generated images

[Images and diagrams related to GANs showing user edits and corresponding generated images]
Adversarial Examples

\[ K(X + v) \neq K(X), \]
where \( K \) is a classifier, \( X \) is input image, \( v \) is perturbation.

Intriguing properties of neural networks, Szegedy et al. - 2013
Adversarial Training

**We can:**
- We can generate adversarial samples to fool a discriminative model
- We can use those adversarial samples to make models robust
- We then require more effort to generate adversarial samples
- Repeat this and we get better discriminative model

**GANs extend that idea to generative models:**
- Generator: generate fake samples, tries to fool the Discriminator
- Discriminator: tries to distinguish between real and fake samples
- Train them against each other
- Repeat this and we get better Generator and Discriminator
• **$Z$** is some random noise (Gaussian/Uniform).
• **$Z$** can be thought as the latent representation of the image.
Training Discriminator

Real world images → Sample → Discriminator → Real/Fake → Loss

Latent random variable → Generator → Sample → Discriminator

Backprop error to update discriminator weights
Training Generator

Latent random variable

Real world images

Sample

Generator

Sample

Discriminator

Real

Fake

Loss

Backprop error to update generator weights

https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training
Generator in action
GAN’s formulation

$$\min \max_{G} \max_{D} V(D, G)$$

- It is formulated as a minimax game, where:
  - The Discriminator is trying to maximize its reward $V(D, G)$
  - The Generator is trying to minimize Discriminator’s reward (or maximize its loss)

$$V(D, G) = \mathbb{E}_{x \sim p(x)}[\log D(x)] + \mathbb{E}_{z \sim q(z)}[\log(1 - D(G(z)))]$$
Faces

CIFAR

**DCGAN**: Bedroom images
Deep Convolutional GANs (DCGANs)

Generator Architecture

Key ideas:

• Replace FC hidden layers with Convolutions
  • **Generator**: Fractional-Strided convolutions

• Use Batch Normalization after each layer

• **Inside Generator**
  • Use ReLU for hidden layers
  • Use Tanh for the output layer
Latent vectors capture interesting patterns...
Celebs...
The Cool Stuff...

3D Faces

(a) Azimuth (pose)

(b) Elevation

(c) Lighting

(d) Wide or Narrow

Cool Stuff (contd.)

3D Chairs

(a) Rotation

(b) Width
Part 2

- Conditional GANs
- Applications
  - Image-to-Image Translation
  - Text-to-Image Synthesis
  - Face Aging
- Summary
Conditional GANs

MNIST digits generated conditioned on their class label.

\[
\begin{align*}
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] & \rightarrow \text{Digit 0} \\
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0] & \rightarrow \text{Digit 1} \\
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0] & \rightarrow \text{Digit 2} \\
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0] & \rightarrow \text{Digit 3} \\
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0] & \rightarrow \text{Digit 4} \\
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0] & \rightarrow \text{Digit 5} \\
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0] & \rightarrow \text{Digit 6} \\
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0] & \rightarrow \text{Digit 7} \\
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0] & \rightarrow \text{Digit 8} \\
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1] & \rightarrow \text{Digit 9}
\end{align*}
\]

Figure 2 in the original paper.
Conditional GANs

- Simple modification to the original GAN framework that conditions the model on additional information for better multi-modal learning.

- Lends to many practical applications of GANs when we have explicit supervision available.
Part 2

- Conditional GANs
- Applications
  - Image-to-Image Translation
  - Text-to-Image Synthesis
  - Face Aging
- Summary
Image-to-Image Translation

The idea

Figure 2: Training a conditional GAN to map edges→photo. The discriminator, $D$, learns to classify between fake (synthesized by the generator) and real \{edge, photo\} tuples. The generator, $G$, learns to fool the discriminator. Unlike an unconditional GAN, both the generator and discriminator observe the input edge map.
conditional GAN vs GAN

\( \mathcal{L}_{GAN} = \mathbb{E}_{x,y}[\log D(y)] + \mathbb{E}_{x,y}[\log(1 - D(G(x, z)))] \)

\( \mathcal{L}_{cGAN} = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,y}[\log(1 - D(x, G(x, z)))] \)

\( \mathcal{L}_{L1} = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1] \)

\( G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \)
Image-to-Image Translation

- Architecture: DCGAN-based architecture
- Training is conditioned on the images from the source domain.
- Conditional GANs provide an effective way to handle many complex domains without worrying about designing structured loss functions explicitly.

Figure 2 in the original paper.

Live demo

- https://affinelayer.com/pixsrv/
Text-to-Image Synthesis

Motivation

Given a text description, generate images closely associated.

Uses a conditional GAN with the generator and discriminator being condition on “dense” text embedding.
Text-to-Image Synthesis

Positive Example: Real Image, Right Text

Negative Examples: Real Image, Wrong Text
Fake Image, Right Text

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. “Generative adversarial text to image synthesis”. ICML
Face Aging with Conditional GANs

- Differentiating Feature: Uses an Identity Preservation Optimization using an auxiliary network to get a better approximation of the latent code \((z^*)\) for an input image.
- Latent code is then conditioned on a discrete (one-hot) embedding of age categories.

Figure 1 in the original paper.

Figure 3 in the original paper.
Summary

- GANs are generative models that are implemented using two stochastic neural network modules: **Generator** and **Discriminator**.
- **Generator** tries to generate samples from random noise as input.
- **Discriminator** tries to distinguish the samples from Generator and samples from the real data distribution.
- Both networks are trained adversarially (in tandem) to fool the other component. In this process, both models become better at their respective tasks.
Why use GANs for Generation?

- Can be trained using back-propagation for Neural Network based Generator/Discriminator functions.
- Sharper images can be generated.
- Faster to sample from the model distribution: *single* forward pass generates a *single* sample.