Generative AI and LLM

Implicit Generative Models/GANs
CS5202

Course Instructor : Dr. Nidhi Goyal

Mahindra™ University
Global Thinkers. Engaged Leaders.

Lecture Plan

- Implicit Generative models
- Generative adversarial Networks (GANs)

Likelihood-based models assumptions

Assumptions:

- We need to specify the prior distribution. p(z), e.g., the standard Gaussian.
- We need to specify the form of the conditional likelihood p(x|z).
- Typically, people use the Gaussian distribution or a mixture of Gaussians.
- Hence, density nets are the prescribed models because we need to analytically formulate all distributions in advance.

As a result:

The objective function is the (approximated) log-likelihood function.

• We can optimize the objective using gradient-based optimization methods and the autograd tools.

• We can parameterize the conditional likelihood using deep neural networks.

Problem with density networks

- There is no analytical solution (except the case equivalent to the probabilistic PCA).
- We get an approximation of the log-likelihood function.
- We need a lot of samples from the prior to get a reliable approximation of the log-likelihood function.
- It suffers from the curse of dimensionality.

Some Outputs from VAE

Images often blurry,
Not sharp, realistic

What information do we loss in VAE

- **Small Text in Images**
- If an image contains small text, such as signs or product labels, the details may be lost during encoding. Since the VAE reduces resolution, small letters disappear, and the model must “guess” what was originally written, usually resulting in unreadable or distorted text.

What information do we loss in VAE

- **Subtle Facial Features**

Subtle facial details such as wrinkles, freckles, or the fine texture of the skin often disappear in the latent space. Individual strands of hair may disappear, leading to a more uniform and less detailed appearance.

What information do we loss in VAE

- **Fabric and Hair Textures**

Clothes with subtle patterns (such as lace or complex weaves) are often smoothed out.

- **Small Elements in the Background**
- Background objects, such as distant windows, leaves, or birds, may be completely omitted. The model may replace them with a generic texture or blur, reducing the realism of the scene.

Implicit Generative models

- probability distribution is implicitly represented by a model of its sampling process.
  - Generative Adversarial Networks (GANs)

Have you heard about?

TECH AI

All of these faces are fake celebrities spawned by AI

/ New research from Nvidia uses artificial intelligence to generate high-res fake celebs

by James Vincent
Oct 30, 2017, 4:35 PM GMT+5:30

0 Comments

Image Source: All of these faces are fake celebrities spawned by AI | The Verge

Creativity (Creative adversarial Networks)

CAN: Creative Adversarial Networks
Generating "Art" by Learning About Styles and Deviating from Style Norms*

Ahmed Elgammal[1†] Bingchen Liu[1] Mohamed Elhoseiny[2] Marian Mazzone[3]
The Art & AI Laboratory - Rutgers University
[1] Department of Computer Science, Rutgers University, NJ, USA
[2] Facebook AI Research, CA, USA
[3] Department of Art History, College of Charleston, SC, USA

June 23, 2017

Abstract

We propose a new system for generating art. The system generates art by looking at art and learning about style; and becomes creative by increasing the arousal potential of the generated art by deviating from the learned styles. We build over Generative Adversarial Networks (GAN), which have shown the ability to learn to generate novel images simulating a given distribution. We argue that such networks are limited in their ability to generate creative products in their original design. We propose modifications to its objective to make it capable of generating creative art by maximizing deviation from established styles and minimizing deviation from art distribution. We conducted experiments to compare the response of human subjects to the generated art with their response to art created by artists. The results show that human subjects could not distinguish art generated by the proposed system from art generated by contemporary artists and shown in top art fairs.

Source: https://arxiv.org/pdf/1706.07068

Breakthrough in 2014

Generative Adversarial Nets

Ian J. Goodfellow,[*] Jean Pouget-Abadie,[†] Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,[‡] Aaron Courville, Yoshua Bengio[§]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

~~NIPS~~
NeurIPS

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

Generative Adversarial Nets

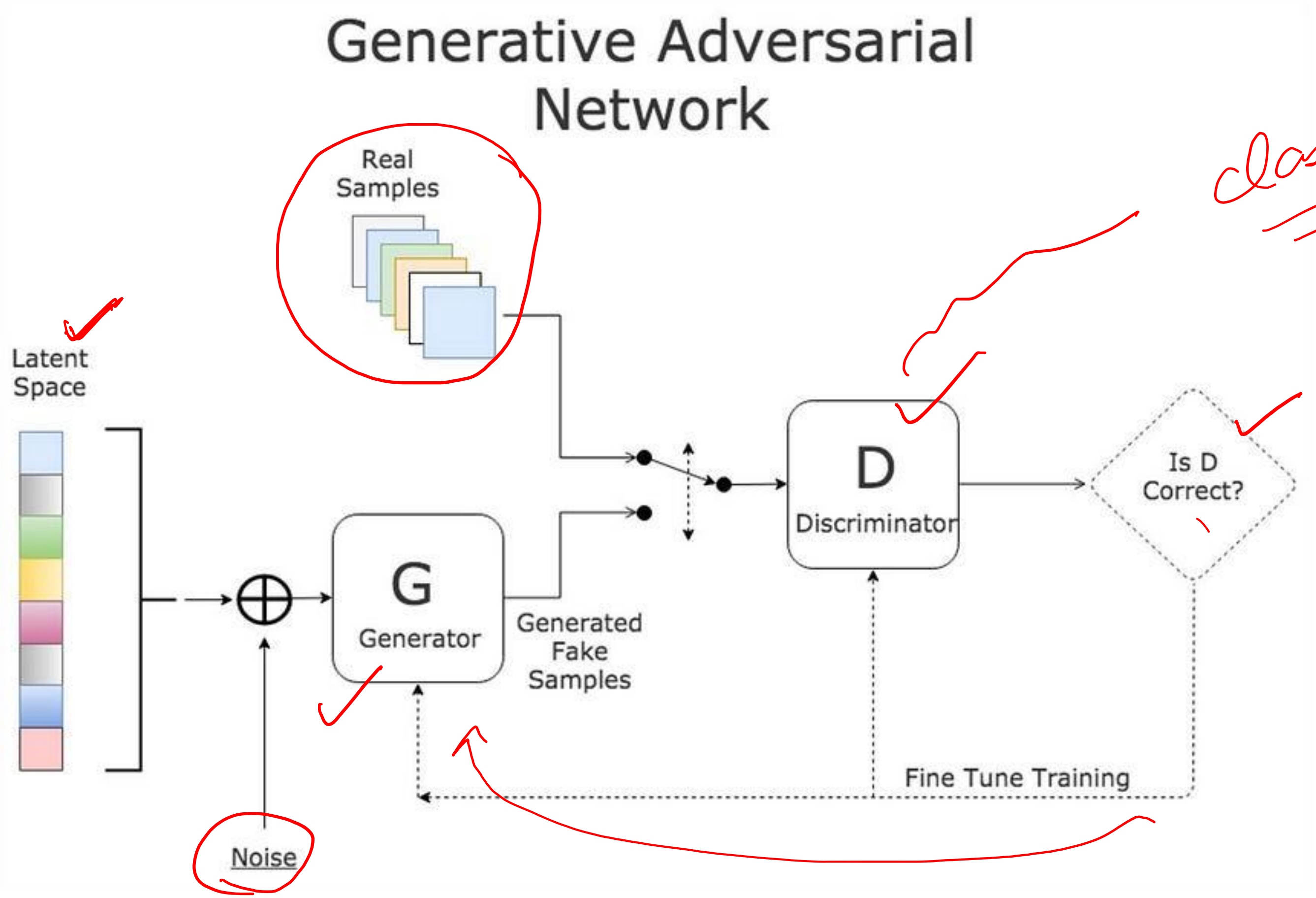
Generative Adversarial Networks (GANs)

- Use two networks:
  - Generator
  - Discriminator
- Learn through competition
- Very realistic outputs

Pros and Cons

- Generate the sharpest images
- Easy to train (since no statistical inference is required), and only back-propogation is needed to obtain gradients
- GANs are difficult to optimize due to unstable training dynamics.
- No statistical inference can be done with them (except here):
  GANs belong to the class of ***direct implicit*** density models; they model ***p(x)*** without explicitly defining the p.d.f.

Generative adversarial Networks



classifier

Generative adversarial Networks

- The Generator (forger) needs to learn how to create data in such a way that the Discriminator isn’t able to distinguish it as fake anymore.

- The competition between these two teams is what improves their knowledge, until the Generator succeeds in creating realistic data.

Training the GAN

- The fundamental steps to train a GAN can be described as following:
  - Sample a **noise** set and a **real-data** set, each with size *m.*
  - Train the **Discriminator** on this data.
  - Sample a different noise **subset** with size *m*.
  - Train the **Generator** on this data.
  - **Repeat** from Step 1.

Generator and Discriminator

$\theta_i \sim (\underbrace{\theta_1, \theta_2})$

$G(z, \theta_1)$ ✓

- A neural network $\boxed{G(\boldsymbol{z}, \boldsymbol{\vartheta_1})}$ is used to model the Generator mentioned above. It's role is mapping input noise variables **z** to the desired data space ***x*** (say images).

$D(x, \theta_2)$

- a second neural network $D(\boldsymbol{x}, \boldsymbol{\vartheta_2})$ models the discriminator and outputs the **probability that the data came from the real dataset,** in the range (0,1).

In both cases, $\boldsymbol{\vartheta_i}$ represents the weights or parameters that define each neural network.

Learning Objective

Discriminator

Generator

x'

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

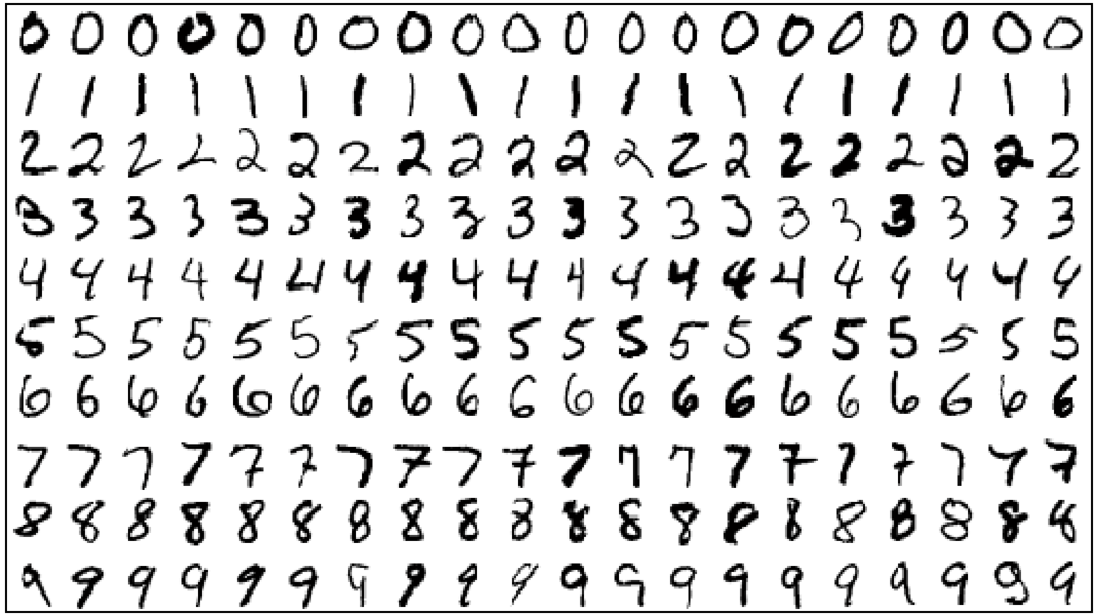
probability distribution of z

probability distribution of input x

minimize the same value

In this minimax game, the generator is trying to maximize it's probability of having it's outputs recognized as real, while the discriminator is trying to minimize this same value.

Implementation



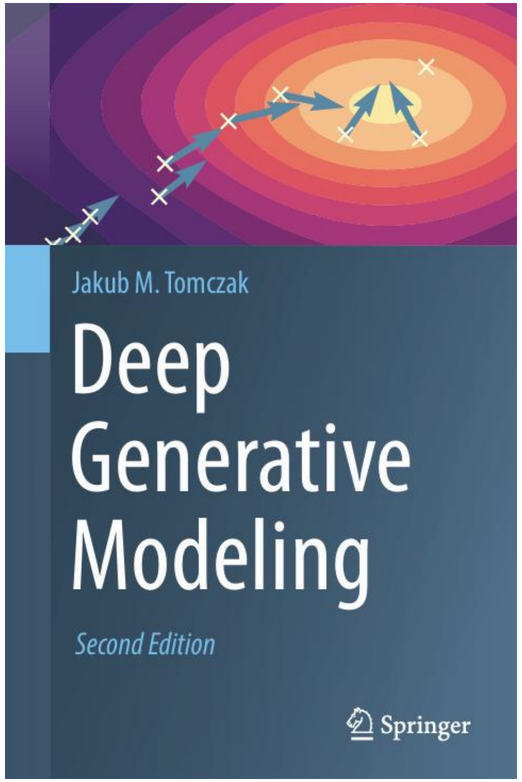
[01_vanilla_gan.ipynb - Colab]

References

- **Vaswani et al., 2017 — *Attention Is All You Need***
  https://arxiv.org/abs/1706.03762
- **Goodfellow et al., 2014 — *Generative Adversarial Networks***
  https://arxiv.org/abs/1406.2661
- **Kingma & Welling, 2013 — *Auto-Encoding Variational Bayes***
  https://arxiv.org/abs/1312.6114
- **Jumper et al., 2021 — *Highly Accurate Protein Structure Prediction with AlphaFold***
  https://www.nature.com/articles/s41586-021-03819-2
- **Shen et al., 2019 — *Deep Image Reconstruction from Human Brain Activity***
  https://www.nature.com/articles/s41593-019-0389-0

Books and lecture notes

Deep Generative Modeling



GitHub

Generative AI & LLMs · Fall 2026