Generative Adversarial Networks

Riley Harper, Rohan Phadke

University of North Carolina at Chapel Hill

October 10, 2024

1. History
   - Ian Goodfellow
   - Challenges Before GANs
2. Theory
3. Impact

History

1. Ian Goodfellow
2. The Idea
3. Challenges Before GANs

Ian Goodfellow

Ian Goodfellow

A renowned researcher in machine learning. He introduced Generative Adversarial Networks (GANs) in 2014 while working on his Ph.D. at the University of Montreal under the supervision of Yoshua Bengio.

Ian Goodfellow

Ian Goodfellow

A renowned researcher in machine learning. He introduced Generative Adversarial Networks (GANs) in 2014 while working on his Ph.D. at the University of Montreal under the supervision of Yoshua Bengio.

Career

Goodfellow's contributions to research extend well beyond GANs. He has worked at OpenAI, Google Brain, and Apple, advancing AI technology in various fields and was listed as a coauthor on the seminal Deep Learning textbook.

Data Generation Problems

Before GANs, generating data that was both high-quality and diverse was a significant problem in machine learning. Models like autoencoders and Boltzmann Machines were commonly used but were limited in their ability to produce realistic outputs, especially in high-dimensional data like images.

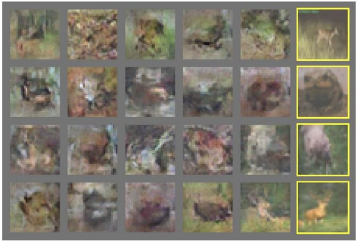
Data Generation Problems

Before GANs, generating data that was both high-quality and diverse was a significant problem in machine learning. Models like autoencoders and Boltzmann Machines were commonly used but were limited in their ability to produce realistic outputs, especially in high-dimensional data like images.

GANs' Contribution

GANs introduced a revolutionary approach to high-dimensional data generation by setting up a competition between two multi-layer perceptrons - a generator and a discriminator - resulting in the generation of more realistic data.

Data Generation



Pre-GAN Methods

Monte Carlo Markov Chains (MCMC)

Used for sampling data in high-dimensional spaces, but slow and computationally expensive, especially for images.

Pre-GAN Methods

Monte Carlo Markov Chains (MCMC)

Used for sampling data in high-dimensional spaces, but slow and computationally expensive, especially for images.

Restricted Boltzmann Machines (RBMs)

Popular for unsupervised learning, but struggled with scalability and training inefficiencies.

GANs' Advantage

Why GANs Replaced MCMC and RBMs

- **Faster and more scalable:** MCMC is slow, requiring many iterations to sample data. GANs generate data, meaning they will scale better with large datasets.
- **End-to-end differentiability:** GANs train both networks using gradient descent, making training smoother, whereas RBMs rely on approximations, which complicates training and slows convergence.
- **Better data generation:** GANs' adversarial process leads to sharper, more realistic images, while MCMC and RBMs struggle with high-dimensional data like images.

Theory

1. Intuition
2. Generator
3. Discriminator
4. Training Process

Intuition

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.

Intuition

Counterfeiting Team

(akin to the Generator)

Police

(akin to the Discriminator)

Generator: Produces a sample in an attempt to mimic real data

1. The input to the Generator is noise: $p_z(z)$
2. The Generator learns the distribution p_g over training data x
3. A mapping to data space is represented as $G(z; \theta_g)$
4. G is a differentiable function represented by a multi-layer neural network with parameters θ_g
5. G is trained to minimize $[\log(1 - D(G(z)))]$

Discriminator:
Estimates the probability that a sample came from the real data.

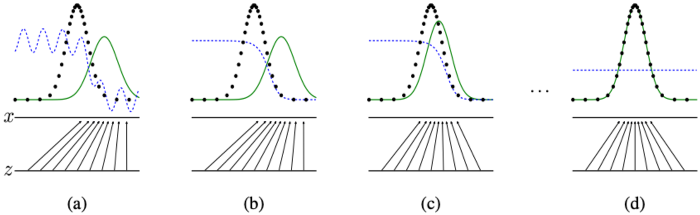
1. The Discriminator is also a multi-layer neural network with a single scalar output: $D(x; \theta_d)$
2. $D(x)$ represents the probability that x came from the data rather than p_g
3. D is trained to maximize the probability of assigning the correct label. It has to correctly differentiate between the training data and the generated samples.

Optimization

Simultaneous Min-Max Optimization

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

Optimization



The output produced by the generator (Green) slowly starts to mimic the real training data (Black) after training iterations.

Optimization

Global Optimum

This minimax game has a global optimum $p_g = p_{\text{data}}$. This is the point at which the generator mimics the distribution of the training data. At this point, **the discriminator is unable to differentiate between the two distributions, so $D(x) = \frac{1}{2}$.**

Training Process

During training, we alternate between k steps of optimizing the Discriminator (D) and one step of optimizing the Generator (G). This process ensures that D is kept close to its optimal solution, provided that G changes slowly enough.

Training Process

If the Discriminator guesses correctly, then the Generator is updated. If the Discriminator guesses incorrectly, then the Discriminator is updated.

Training Process

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**
  for k steps **do**
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for
  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Impact

1. DCGAN
2. CycleGAN
3. StyleGAN

DCGAN: Unsupervised Representation Learning with Deep Convolutional GANs

Overview

DCGAN is an extension of GANs that incorporates convolutional layers to improve the quality of generated images. It also serves as a baseline for many image generation tasks.

Impact

Introduced the idea of using convolutional neural networks in GANs and significantly improved image generation tasks. It has been cited over 18,000 times to date.

CycleGAN: Unpaired Image-to-Image Translation

Overview

CycleGAN enables image translation between domains without requiring paired datasets. This innovation is useful in tasks such as style transfer, where images from one domain can be transformed into another.

Impact

CycleGAN opened new possibilities for unsupervised learning in image translation. It has been cited 24,000 to date.

StyleGAN: A Style-Based Generator Architecture for GANs

Overview

StyleGAN introduced a new generator architecture for GANs that allows better control over the generated images' styles at different resolutions; it's particularly well known for producing high-quality, photorealistic faces.

Impact

StyleGAN's architecture has been widely adopted for generating highly detailed images. It has been cited over 11,000 times to date. In fact you may have even seen StyleGAN used before, try visiting thispersondoesnotexist.com.

Questions?

References in Discussion

Brownlee, J. (2019).
What Are Generative Adversarial Networks (GANs)?
Accessed: 2024-09-15.

Goodfellow, I. (2018).
What are Generative Adversarial Networks? Ian Goodfellow explains.
Accessed: 2024-09-15.

Goodfellow, I. et al. (2014).
Generative Adversarial Nets.
Advances in Neural Information Processing Systems, 27.

Karras, T., Laine, S., and Aila, T. (2019).
A Style-Based Generator Architecture for Generative Adversarial Networks.
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4401–4410.

Radford, A., Metz, L., and Chintala, S. (2015).
Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.
arXiv preprint arXiv:1511.06434.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017).
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks.
Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2223–2232.