Generative Adversarial Networks (GANs)

Amisi Fikirini
Elie Mulamba
Salomon Muhirwa
Lionel Nanguep
Omer Elhussien

African Master's in Machine Intelligence, AIMS-Senegal

April 16, 2023

Overview

Overview

1. Introduction
   - Learning approaches: Supervised and Unsupervised.
   - Discriminative and Generative models
2. Generative Adversarial Networks
   - Introduction
   - GAN architecture
   - GAN Types
3. Deep Convolutional GAN
   - DCGAN architecture
   - GAN and Data augmentation
4. Appendix
5. References

Supervised Learning

Our training data set is $\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$. We are interested in learning a function that maps $x \rightarrow y$.
Examples: Regression, classification, object detection, .etc.

Unsupervised Learning

Our training data set is $\mathcal{D} = \{x_i\}_{i=1}^{n}$. We are interested in extracting or detecting patterns in the data set
Examples: Clustering, dimensionality reduction , density estimation, .etc.

Overview

1. Introduction
   - Learning approaches: Supervised and Unsupervised.
   - Discriminative and Generative models
2. Generative Adversarial Networks
   - Introduction
   - GAN architecture
   - GAN Types
3. Deep Convolutional GAN
   - DCGAN architecture
   - GAN and Data augmentation
4. Appendix
5. References

Discriminative model

The model learns $\mathbb{P}(y|x)$. It assigns labels to new data.
Problems: susceptible to outliers.

Generative model

The model learns $\mathbb{P}(x)$. It generates new data points.

Conditional Generative model

The model learns $\mathbb{P}(x|y)$. It assigns labels to new data, reject outliers, and generate new data points conditioned on given input labels.

Overview

- GANs are a deep-learning-based generative models.
- The GAN model architecture involves two parts:
  1. The Generator model which generates new samples.
  2. The Discriminator model which classify whether its input data is real or fake.

GANs rely on a game theoretic approach in which the generator plays a zero-sum game with the discriminator. If the discriminator successfully identifies real from fake data, it is rewarded, or no change is needed. On the contrary, the generator is penalized with extensive parameter updates. The other scenario is when the generator deceives the discriminator. The former will be rewarded, while the latter will be updated.

Overview

1. Introduction
   - Learning approaches: Supervised and Unsupervised.
   - Discriminative and Generative models
2. Generative Adversarial Networks
   - Introduction
   - GAN architecture
   - GAN Types
3. Deep Convolutional GAN
   - DCGAN architecture
   - GAN and Data augmentation
4. Appendix
5. References

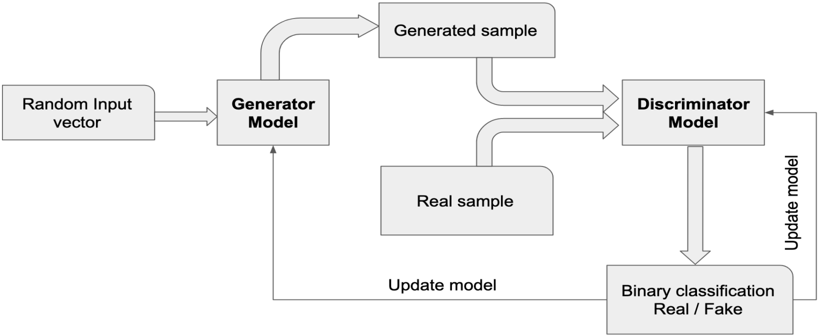

Figure: GAN architecture.

Model assumptions

For the Generator

Let G be a differential function represented by a multilayer perceptron (MLP) with parameter θ_g. Let z be a latent variable drawn from $\mathbb{P}(z)$. Then, $x = G(z)$ is a sample from the generator distribution $\mathbb{P}_G$.

For the Discriminator

Let D be a differential function represented by a MLP with parameter θ_d. Let x_i be a data point drawn from the data distribution $\mathbb{P}_{data}$.

Training time

For the Generator

Train G to convert z into fake data point sampled from $\mathbb{P}_G$ by deceiving the discriminator.

For the Discriminator

Train D to classify data as real or fake.

Training time Cont.

Simultaneously, train G and D with a minimax game

$$\min_G \max_D V(G, D), \quad where \tag{2.1}$$

$$V(G, D) = \left(\mathbb{E}_{x \sim \mathbb{P}_{data}} [\log D(x)] + \mathbb{E}_{z \sim \mathbb{P}(z)} [\log (1 - D(G(z)))]\right).$$

1. The discriminator wants $D(x) = 1$ for real data, and $D(x) = 0$ for fake data.
2. The generator wants $D(x) = 1$ for fake data.

Training time Cont.

Algorithm 1: GAN training algorithm

---

for *number of training iterations* **do**
  loop instructions
  (update D) $D = D + \alpha_D \frac{\partial V}{\partial D}$,
  (update G) $G = G - \alpha_G \frac{\partial V}{\partial G}$.

Proposition

For a fixed G *, the optimal discriminator is,*

$$D_G^*(x) = \frac{\mathbb{P}_{data}(x)}{\mathbb{P}_{data}(x) + \mathbb{P}_G(x)}.$$

Theorem

Let

$$C(G) = \mathbb{E}_{x\sim\mathbb{P}_{data}}\left[\log\frac{\mathbb{P}_{data}(x)}{\mathbb{P}_{data}(x) + \mathbb{P}_G(x)}\right] + \mathbb{E}_{x\sim\mathbb{P}_G}\left[\log\frac{\mathbb{P}_G(x)}{\mathbb{P}_{data}(x) + \mathbb{P}_G(x)}\right].$$

The global minimum of $C(G) = -\log 4$ *is achieved if and only if* $\mathbb{P}_G = \mathbb{P}_{data}$.

For proofs, check the appendix.

Overview

1. Introduction
   - Learning approaches: Supervised and Unsupervised.
   - Discriminative and Generative models
2. Generative Adversarial Networks
   - Introduction
   - GAN architecture
   - GAN Types
3. Deep Convolutional GAN
   - DCGAN architecture
   - GAN and Data augmentation
4. Appendix
5. References

GAN is a growing field with a huge number of publications each week. In this work, we will mention some of them, while an up-to-date list could be accessed at this link.

Types of GAN

1. Deep Convolutional GAN (DC-GAN).
2. Improved loss functions:
    - Wasterstein GAN (WGAN)
    - WGAN with gradient penalty.
3. Conditional GANS:
    - BigGAN.
    - Unpaired image-to-image translation: CycleGAN.

Overview

DC-GAN is a direct extension of the GAN. It uses convolutional and convolutional-transpose layers in the discriminator and generator, respectively.

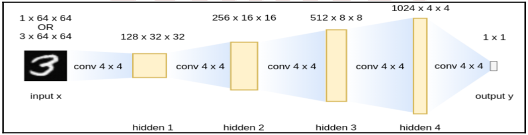

Figure: Discriminator architecture.

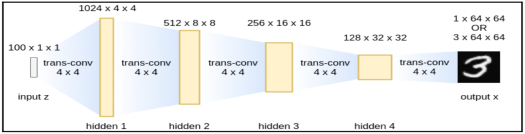

Figure: Generator architecture.

Overview

- Why is GAN receiving this massive interest during the previous years?
    - To answer this question, we must introduce the Data augmentation concept.
    - Data augmentation works by creating new, artificial, but plausible examples from the input problem domain. It contributes to better model performance- by increasing the model skill and providing regularizing effect- and reduces generalization error.

Generative successful models such as GANs will provide an alternative and potentially more domain-specific approach for Data augmentation.

Appendix

For the proofs, please check this link.
For the GitHub repository, please check this link.

References I

[1] Ian Goodfellow et al. "Generative adversarial nets ian". In: *Mining of Massive Datasets; Cambridge University Press: Cambridge, UK* (2014).

[2] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (2014).

[3] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[4] Ian Goodfellow. "Nips 2016 tutorial: Generative adversarial networks". In: *arXiv preprint arXiv:1701.00160* (2016).

References II

[5] Jason Brownlee. *Generative adversarial networks with python: deep learning generative models for image synthesis and image translation*. Machine Learning Mastery, 2019.

[6] Jakub Langr and Vladimir Bok. *GANs in action: deep learning with generative adversarial networks*. Manning Publications, 2019.

[7] John Hany and Greg Walters. *Hands-On Generative Adversarial Networks with PyTorch 1. x: Implement next-generation neural networks to build powerful GAN models using Python*. Packt Publishing Ltd, 2019.

References III

[8] Kuan Wei. *Understand Transposed Convolutions*. https://towardsdatascience.com/understand-transposed-convolutions-and-build-your-own-transposed-convolution-layer-from-scratch-4f5d97b2967. [Online; accessed 16-April-2023]. 2020.

[9] Michigan Online. *Generative Models II*. https://web.eecs.umich.edu/~justincj/slides/eecs498/498_FA2019_lecture20.pdf. [Online; accessed 16-April-2023]. 2021.

[10] Nathan Inkawhich. *DCGAN Tutorial*. https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html. [Online; accessed 16-April-2023]. 2023.

AIMS African Institute for Mathematical Sciences SENEGAL