



GAN - Theory and Applications

Emanuele Ghelfi

Paolo Galeone

Federico Di Mattia

Michele De Simoni

<https://bit.ly/2Y1nqay>

May 4, 2019



PYCONX

Overview

1. Introduction
2. Models definition
3. GANs Training
4. Types of GANs
5. GANs Applications

Introduction

“Generative Adversarial Networks is the most interesting idea in the last ten years in machine learning.”

Yann LeCun, Director, Facebook AI



Generative Adversarial Networks

Two components, the **generator** and the **discriminator**:

- The **generator** G needs to capture the data distribution.
- The **discriminator** D estimates the probability that a sample comes from the training data rather than from G.

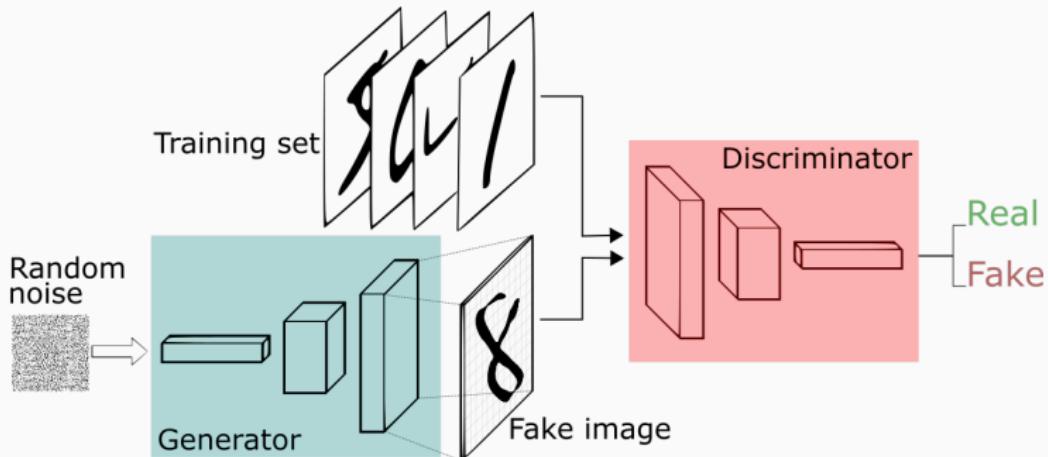


Figure 1: Credits: Silva

Generative Adversarial Networks

GANs game:

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

Generative Adversarial Networks

GANs game:

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

Generative Adversarial Networks

GANs game:

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

GANs - Discriminator

- **Discriminator** needs to:

- Correctly classify **real** data:

$$\max_D \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)]$$

$$D(x) \rightarrow 1$$

- Correctly classify **wrong** data:

$$\max_D \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

$$D(G(z)) \rightarrow 0$$

- The discriminator is an **adaptive loss function**.



**YOU DON'T NEED TO
DESIGN A LOSS FUNCTION**

**IF A DISCRIMINATOR
DESIGNS ONE FOR YOU**

GANs - Generator

- **Generator** needs to **fool** the discriminator:
 - Generate samples similar to the real ones:

$$\min_G \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad D(G(z)) \rightarrow 1$$

GANs - Generator

- **Generator** needs to **fool** the discriminator:

- Generate samples similar to the real ones:

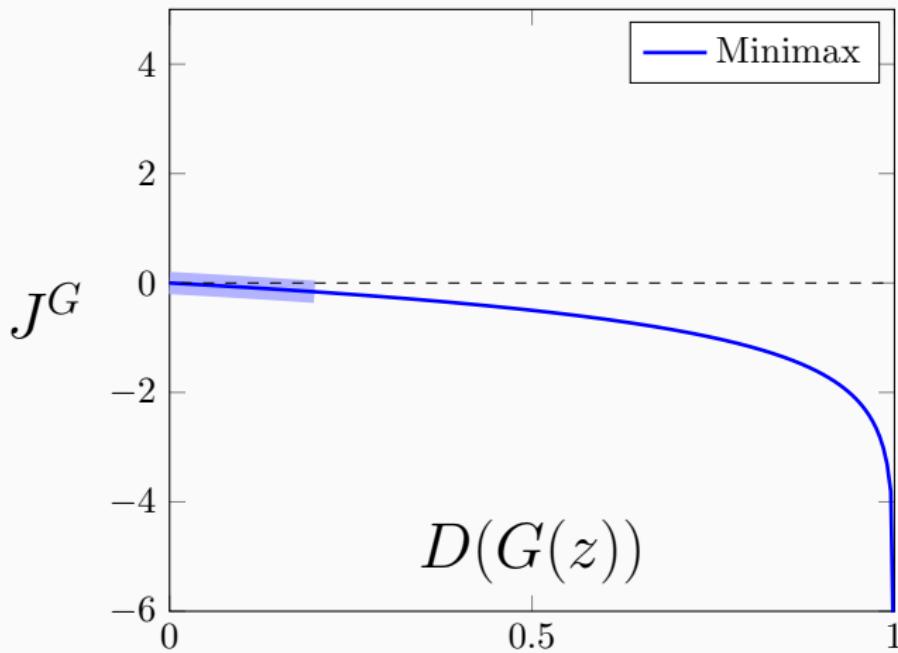
$$\min_G \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad D(G(z)) \rightarrow 1$$

- Non saturating objective (Goodfellow et al., 2014):

$$\min_G \mathbb{E}_{z \sim p_z(z)} [-\log(D(G(z)))]$$

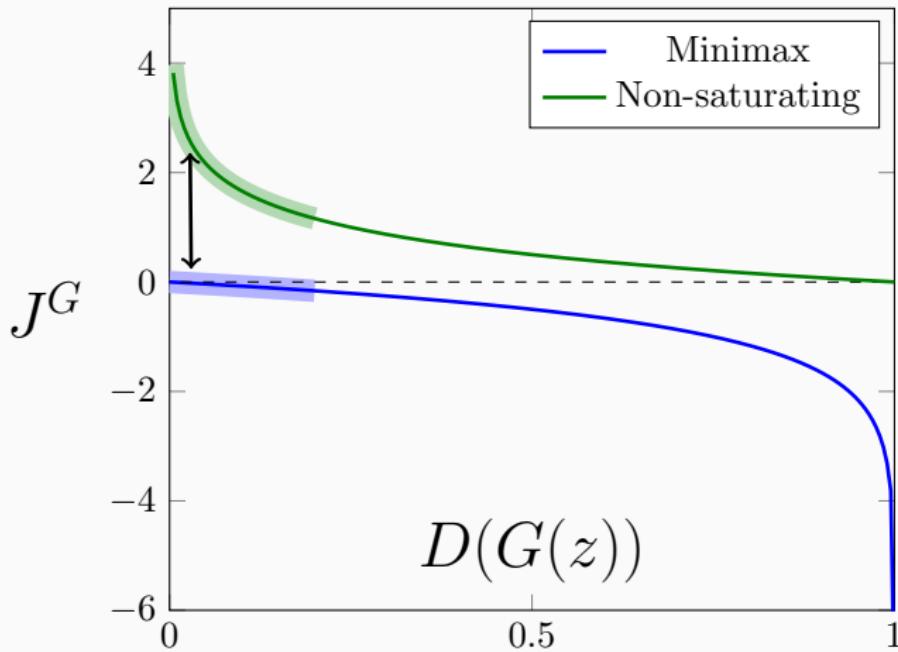
GANs - Generator Objectives

- Minimax: $\log(1 - D(G(z)))$



GANs - Generator Objectives

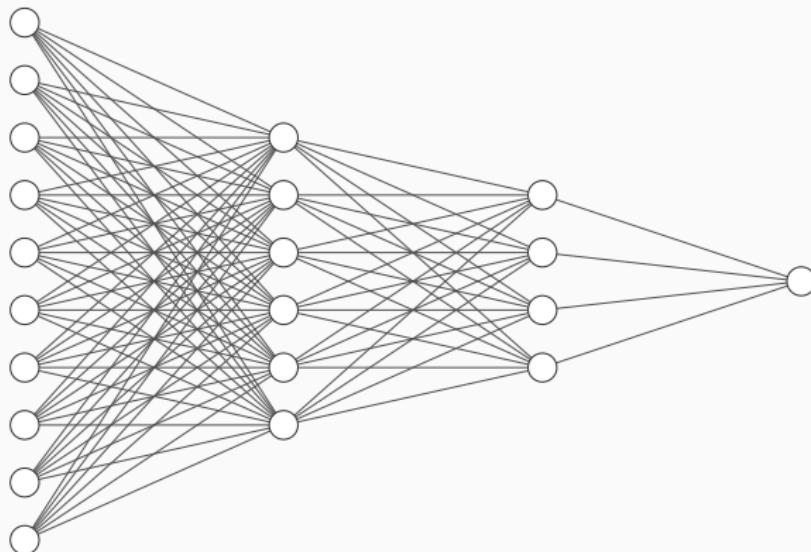
- Minimax: $\log(1 - D(G(z)))$
- Non-saturating: $-\log(D(G(z)))$



Models definition

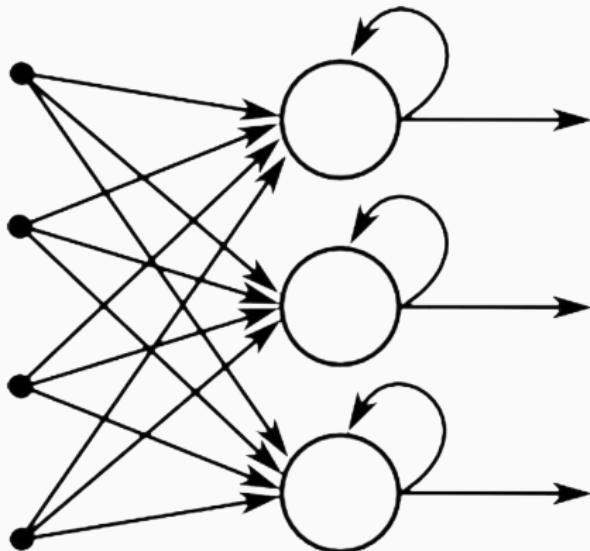
GANs - Models definition

- Different architectures for different data types.
 - Tuple of numbers? **Fully Connected Neural Networks**



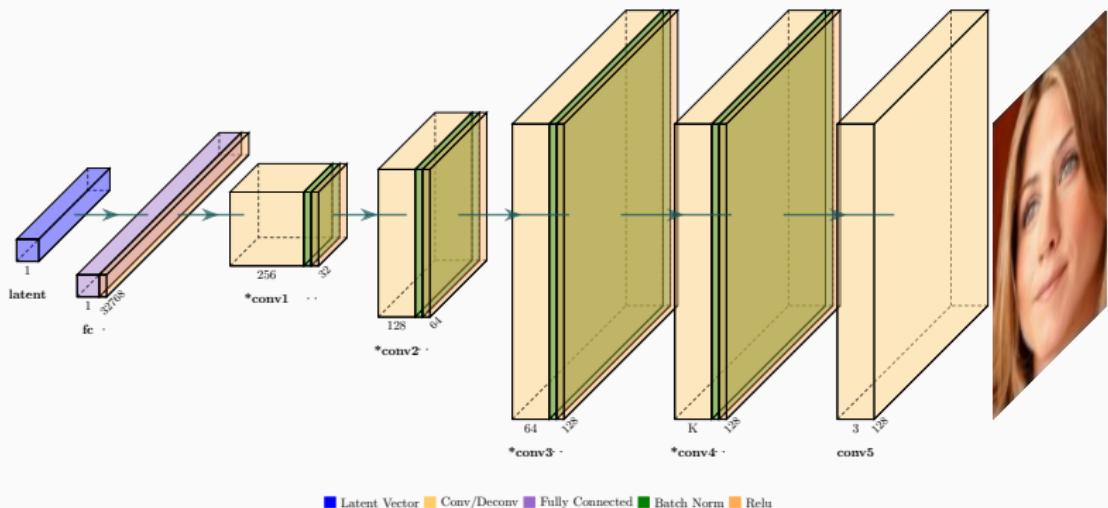
GANs - Models definition

- Different architectures for different data types.
 - Text or sequences? Recurrent Neural Networks



GANs - Models definition

- Different architectures for different data types.
 - Images? **Convolutional Neural Networks**



GANs Training

GANs - Training

- D and G are **competing** against each other.
- **Alternating** execution of training steps.
- Use **minibatch stochastic gradient descent/ascent**.



GANs - Training - Discriminator

How to **train** the **discriminator**?

Repeat from 1 to **k**:

1. Sample minibatch of m noise samples $z^{(1)}, \dots, z^{(m)}$ from $p_z(z)$

GANs - Training - Discriminator

How to **train** the **discriminator**?

Repeat from 1 to **k**:

1. Sample minibatch of m noise samples $z^{(1)}, \dots, z^{(m)}$ from $p_z(z)$
2. Sample minibatch of m examples $x^{(1)}, \dots, x^{(m)}$ from $p_{data}(x)$

GANs - Training - Discriminator

How to **train** the **discriminator**?

Repeat from 1 to **k**:

1. Sample minibatch of m noise samples $z^{(1)}, \dots, z^{(m)}$ from $p_z(z)$
2. Sample minibatch of m examples $x^{(1)}, \dots, x^{(m)}$ from $p_{data}(x)$
3. Update **D**:

$$J = \underbrace{\frac{1}{m} \sum_{i=1}^m \log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))}_{D \text{ performance}}$$

$$\theta_d = \theta_d + \lambda \nabla_{\theta_d} J$$

GANs - Training - Generator

How to **train** the **generator**?

Update executed **only once** after **D** updates:

1. Sample minibatch of m noise samples $z^{(1)}, \dots, z^{(m)}$ from $p_z(z)$

GANs - Training - Generator

How to **train** the **generator**?

Update executed **only once** after **D** updates:

1. Sample minibatch of m noise samples $z^{(1)}, \dots, z^{(m)}$ from $p_z(z)$
2. Update **G**:

$$\mathbf{J} = \underbrace{\frac{1}{m} \sum_{i=1}^m \log(\mathbf{D}(\mathbf{G}(z^{(i)})))}_{\text{G performance}}$$

$$\theta_{\mathbf{g}} = \theta_{\mathbf{g}} + \lambda \nabla_{\theta_{\mathbf{g}}} \mathbf{J}$$

GANs - Training - Considerations

- Optimizers: Adam, Momentum, RMSProp.
- **Arbitrary number** of steps or epochs.
- Training is completed when D is **completely fooled** by G.
- Goal: reach a **Nash Equilibrium** where the best D can do is random guessing.

Types of GANs

Types of GANs

Two big families:

- **Unconditional** GANs (just described).
- **Conditional** GANs (Mirza and Osindero, 2014).

Conditional GANs

- Both G and D are **conditioned** on some extra information \mathbf{y} .
- In **practice**: perform conditioning by feeding \mathbf{y} into D and G .

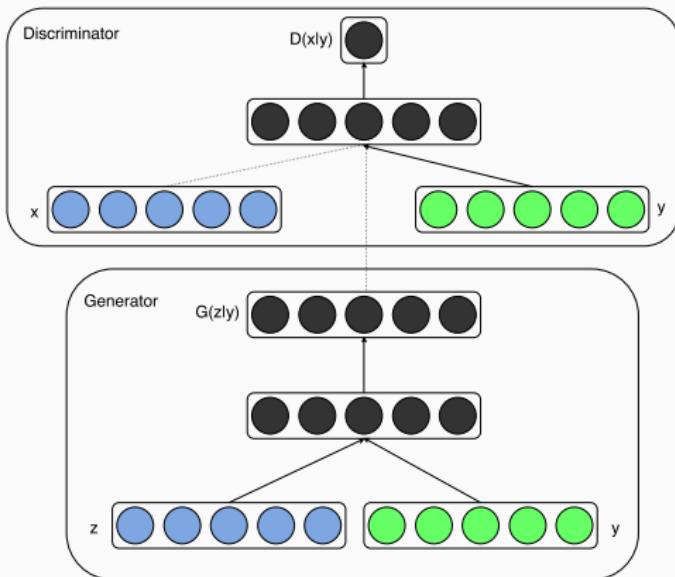


Figure 2: From Mirza and Osindero (2014)

Conditional GANs

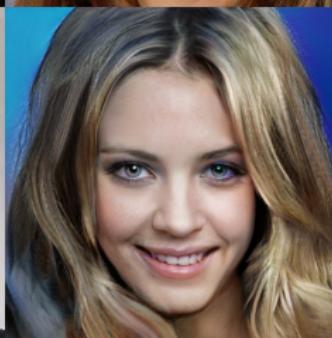
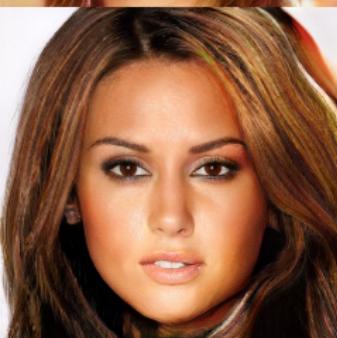
The GANs game becomes:

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

Notice: the same representation of the condition has to be presented to both network.

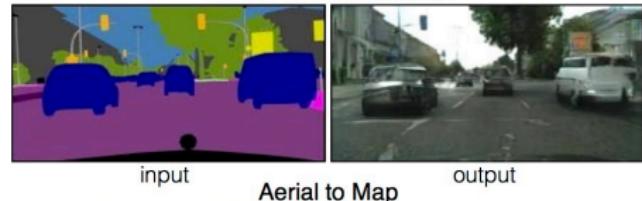
GANs Applications

Unconditional - Face Generation - Karras et al. (2017)



Conditional - Domain Translation - Isola et al. (2016)

Labels to Street Scene



input

output

Aerial to Map



input

output

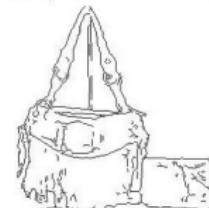
Input



Ground truth



Output



Conditional - Semantic Image Synthesis - Park et al. (2018)

Conditional - Image Super Resolution - Ledig et al. (2016)



SRGAN



Real-world GANs

- Semi-Supervised Learning (Salimans et al., 2016)
- Image Generation (almost all GAN papers)
- Image Captioning
- Anomalies Detection (Zenati et al., 2018)
- Program Synthesis (Ganin et al., 2018)
- Genomics and Proteomics (Killoran et al., 2017) (De Cao and Kipf, 2018)
- Personalized GANufactoring (Hwang et al., 2018)
- Planning

References

- [De Cao and Kipf 2018] DE CAO, Nicola ; KIPF, Thomas: MolGAN: An Implicit Generative Model for Small Molecular Graphs. (2018). – (2018)
- [Ganin et al. 2018] GANIN, Yaroslav ; KULKARNI, Tejas ; BABUSCHKIN, Igor ; ESLAMI, S. M. A. ; VINYALS, Oriol: Synthesizing Programs for Images Using Reinforced Adversarial Learning. (2018). – (2018)
- [Goodfellow et al. 2014] GOODFELLOW, Ian J. ; POUGET-ABADIE, Jean ; MIRZA, Mehdi ; Xu, Bing ; WARDE-FARLEY, David ; OZAIR, Sherjil ; COURVILLE, Aaron ; Bengio, Yoshua: Generative Adversarial Networks. (2014). – (2014)

[Hwang et al. 2018] HWANG, Jyh-Jing ; AZERNIKOV, Sergei ; EFROS, Alexei A. ; Yu, Stella X.: Learning Beyond Human Expertise with Generative Models for Dental Restorations. (2018). – (2018)

[Isola et al. 2016] ISOLA, Phillip ; ZHU, Jun-Yan ; ZHOU, Tinghui ; EFROS, Alexei A.: Image-to-Image Translation with Conditional Adversarial Networks. (2016). – (2016)

[Karras et al. 2017] KARRAS, Tero ; AILA, Timo ; LAINE, Samuli ; LEHTINEN, Jaakko: Progressive Growing of GANs for Improved Quality, Stability, and Variation. (2017). – (2017)

[Killoran et al. 2017] KILLORAN, Nathan ; LEE, Leo J. ; DELONG, Andrew ; DUVENAUD, David ; FREY, Brendan J.: Generating and Designing DNA with Deep Generative Models. (2017). – (2017)

[Ledig et al. 2016] LEDIG, Christian ; THEIS, Lucas ; HUSZAR, Ferenc ; CABALLERO, Jose ; CUNNINGHAM, Andrew ; ACOSTA, Alejandro ;AITKEN, Andrew ; TEJANI, Alykhan ; TOTZ, Johannes ; WANG, Zehan ; SHI, Wenzhe: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. (2016). – (2016)

[Mirza and Osindero 2014] MIRZA, Mehdi ; OSINDERO, Simon: Conditional Generative Adversarial Nets. (2014). – (2014)

[Park et al. 2018] PARK, Taesung ; LIU, Ming-Yu ; WANG, Ting-Chun ; ZHU, Jun-Yan: Semantic Image Synthesis with Spatially-Adaptive Normalization. (2018). – (2018)

[Salimans et al. 2016] SALIMANS, Tim ; GOODFELLOW, Ian ; ZAREMBA, Wojciech ; CHEUNG, Vicki ; RADFORD, Alec ; CHEN, Xi: Improved Techniques for Training GANs. (2016). – (2016)

[Silva] SILVA, Thalles: *An Intuitive Introduction to Generative Adversarial Networks (GANs)*

[Zenati et al. 2018] ZENATI, Houssam ; Foo, Chuan S. ; LECOUAT, Bruno ; MANEK, Gaurav ; CHANDRASEKHAR, Vijay R.: Efficient GAN-Based Anomaly Detection. (2018). – (2018)