GAN - Theory and Applications

Emanuele Ghelfi
Paolo Galeone
Federico Di Mattia
Michele De Simoni

April 12, 2019

> "Adversarial Training (also called GAN for Generative Adversarial Networks) is the most interesting idea in the last 10 years of ML."
>
> — Yann LeCun

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.

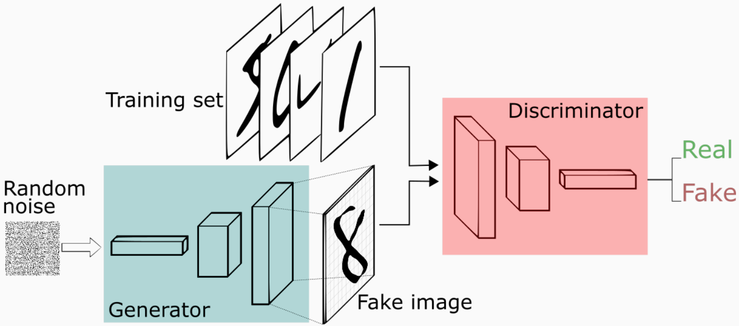


Figure 1: Credits: Silva

Generative Adversarial Networks

GANs game:

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

Generative Adversarial Networks

GANs game:

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

Generative Adversarial Networks

GANs game:

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

GANs - Discriminator

- **Discriminator** needs to:
  - Correctly classify real data:

$$\boxed{\max_{D} \mathop{\mathbb{E}}_{x \sim p_{data}(x)} [\log D(x)].} \qquad D(x) \to 1$$

  - Correctly classify wrong data:

$$\boxed{\max_{D} \mathop{\mathbb{E}}_{z \sim p_{z}(z)} [\log(1 - D(G(z)))].} \qquad D(G(z)) \to 0$$

- The discriminator is an **adaptive loss function** that gets discarded once the generator has been trained.

GANs - Generator

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

$$\boxed{\min_G \mathop{\mathbb{E}}_{z \sim p_z(z)}[\log(1 - D(G(z)))].} \qquad D(G(z)) \to 1$$

GANs - Generator

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

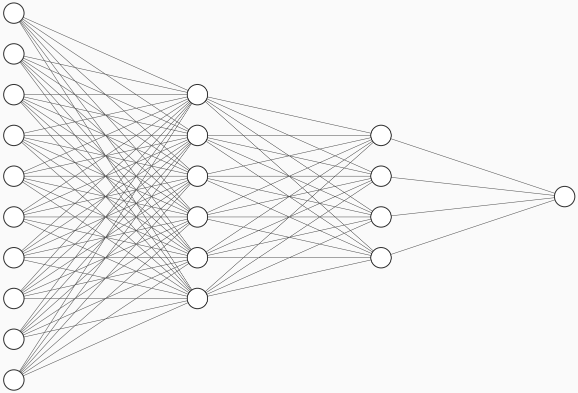
$$\boxed{\min_G \mathop{\mathbb{E}}_{z \sim p_z(z)} [\log(1 - D(G(z)))].} \qquad D(G(z)) \rightarrow 1$$

  - Saturates easily (Goodfellow et al., 2014):

$$\boxed{\max_G \mathop{\mathbb{E}}_{z \sim p_z(z)} [\log(D(G(z)))].} \tag{2}$$

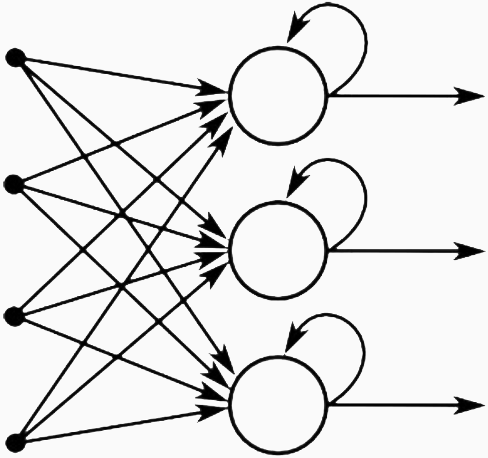
GANs - Models definition

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



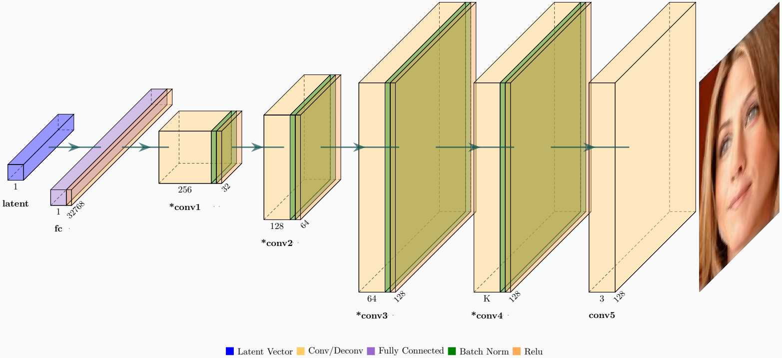
GANs - Models definition

- 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

- Discriminator and generator 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 $z^{(1)}, \ldots, z^{(m)}$ from $p_g(z)$

GANs - Training - Discriminator

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

Repeat from 1 to **k**:

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

GANs - Training - Discriminator

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

Repeat from 1 to **k**:

1. Sample minibatch of m noise samples $z^{(1)}, \ldots, z^{(m)}$ from $p_g(z)$
2. Sample minibatch of m examples $x^{(1)}, \ldots, x^{(m)}$ from $p_{data}(x)$
3. Train the **discriminator** by stochastic gradient **ascent**:

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

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

GANs - Training - Generator

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

The update is executed **only once** and only after the turn of the discriminator is completed:

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

GANs - Training - Generator

How to **train** the **generator**?
The update is executed **only once** and only after the turn of the discriminator is completed:

1. Sample minibatch of m noise samples $z^{(1)}, \ldots, z^{(m)}$ from $p_g(z)$
2. Train the **generator** by stochastic gradient **ascent**:

$$\nabla_{\theta_g} \underbrace{\frac{1}{m} \sum_{i=1}^{m} \underbrace{\log(D(G(z^{(i)})))}_{\text{Generator loss}}}_{\text{Loss estimation}}$$

GANs - Training - Considerations

- Optimizers: Adam, Momentum, RMSProp.
- Training phase can last for an **arbitrary number** of steps or epochs.
- Training is completed when the discriminator is **completely fooled** by the generator.
- 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 y.
- In **practice**: perform conditioning by feeding y into the discriminator and generator.

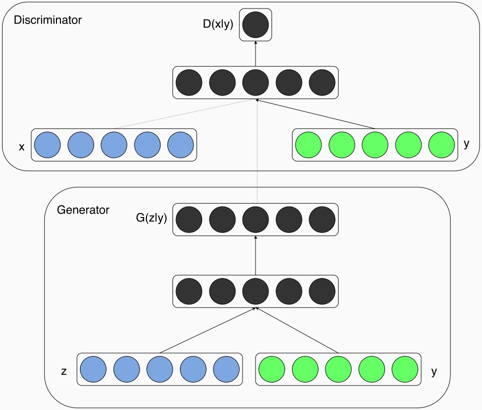


Figure 2: From Mirza and Osindero (2014)

Conditional GANs

The GANs game becomes:

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

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

GANs Applications

Face Generation - Karras et al. (2017)

Domain Translation - Isola et al. (2016)

Input

Generated

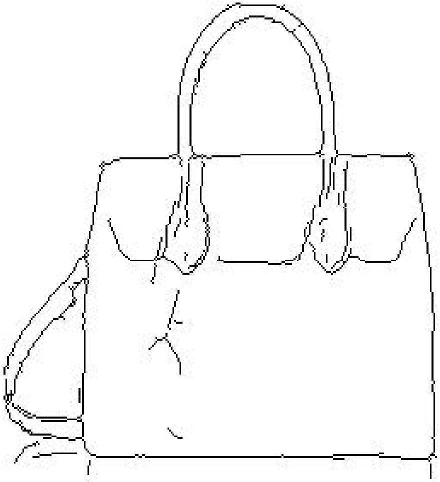
Domain Translation - Isola et al. (2016)

Input

Generated

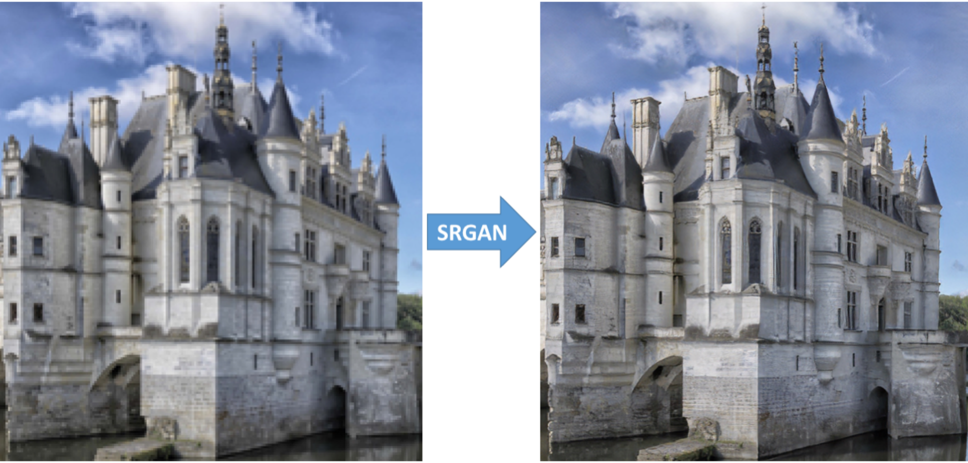
Domain Translation - Isola et al. (2016)

Input



Generated

Image Super Resolution - Ledig et al. (2016)



LG Image

Generated Image

Thank you for your attention!

Questions?

References

[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)

[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)

[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)

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