GAN - Theory and Applications

Michele de Simoni
Paolo Galeone
Emanuele Ghelfi
Federico di Mattia

March 12, 2019

Generative Adversarial Networks

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

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

- The **generator** G, aim is to capture the data distribution.
- The **discriminator** D, estimates the probability that a sample came from the training data rather than from G.

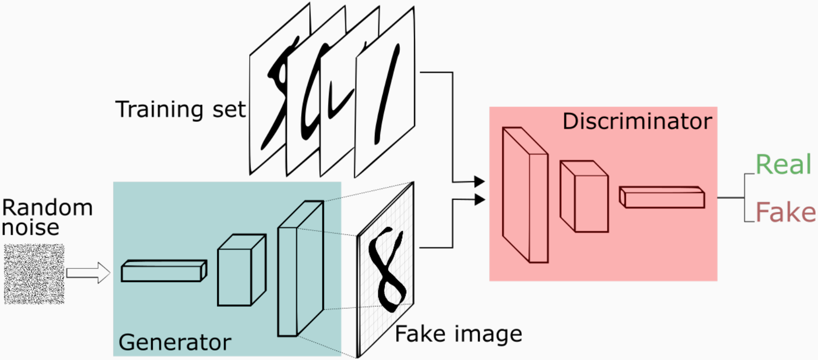


Figure 1: Credits: Reference

Generative Adversarial Networks

Generator and Discriminator compete against each other, playing the following **zero sum min-max game** with value function $V_{GAN}(D, G)$:

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

Generative Adversarial Networks

Generator and Discriminator compete against each other, playing the following **zero sum min-max game** with value function $V_{GAN}(D, G)$:

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

Generative Adversarial Networks

Generator and Discriminator compete against each other, playing the following **zero sum min-max game** with value function $V_{GAN}(D, G)$:

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

GANs - Discriminator

Intuitive explanation:

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

$$\boxed{\max_D \mathop{\mathbb{E}}_{x \sim p_{data}(x)} [\log D(x)].} \tag{2}$$

  - Correctly classify wrong data:

$$\boxed{\max_D \mathop{\mathbb{E}}_{z \sim p_z(z)} [\log(1 - D(G(z)))].} \tag{3}$$

GANs - Generator

Intuitive explanation:

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

$$\boxed{\min_{G} \mathop{\mathbb{E}}_{z \sim p_z(z)} [\log(1 - D(G(z)))].} \tag{4}$$

GANs - Generator

Intuitive explanation:

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

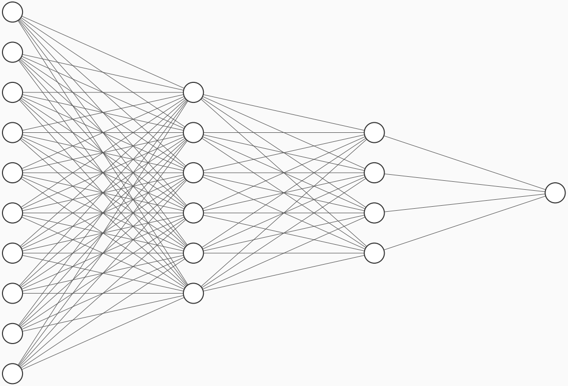
$$\boxed{\min_{G} \mathop{\mathbb{E}}_{z\sim p_z(z)}[\log(1 - D(G(z)))].} \tag{4}$$

  - Saturates easily [**?**].
  - Change loss for generator:

$$\boxed{\max_{G} \mathop{\mathbb{E}}_{z\sim p_z(z)}[\log(D(G(z)))].} \tag{5}$$

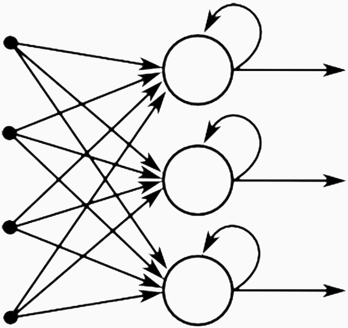
GANs - Models definition

- Both D and G can be parametrized functions (Neural Networks).
- Different architectures to reach different aims.
  - Tuple of numbers?



GANs - Models definition

- Both D and G can be parametrized functions (Neural Networks).
- Different architectures to reach different aims.
    - Text or sequences?



GANs - Models definition

- Both D and G can be parametrized functions (Neural Networks).
- Different architectures to reach different aims.
    - Images?