

Generative Adversarial Networks

The man with no name

Stefano Gualandi

Università di Pavia, Dipartimento di Matematica

email: stefano.gualandi@unipv.it

twitter: [@famo2spaghi](https://twitter.com/famo2spaghi)

blog: <http://stegua.github.com>

web: <http://matematica.unipv.it/gualandi/>

Outline

1 Introduction

2 Deep Learning

Which is a fake photo?

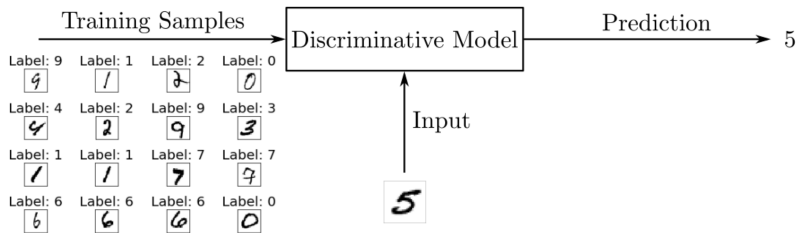


Which is a fake photo?



<https://thispersondoesnotexist.com/>

Using Deep NNs as Discriminative Models



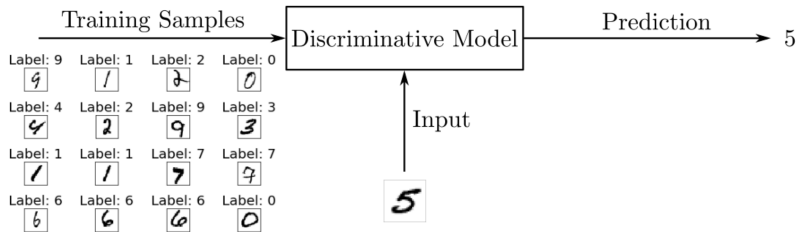
Step 1: Train-test split of the dataset.

Step 2: Choice of architecture.

Step 3: Training.

Step 4: Testing.

Using Deep NNs as Discriminative Models



Step 1: **Train-test split of the dataset.**

Step 2: **Choice of architecture.**

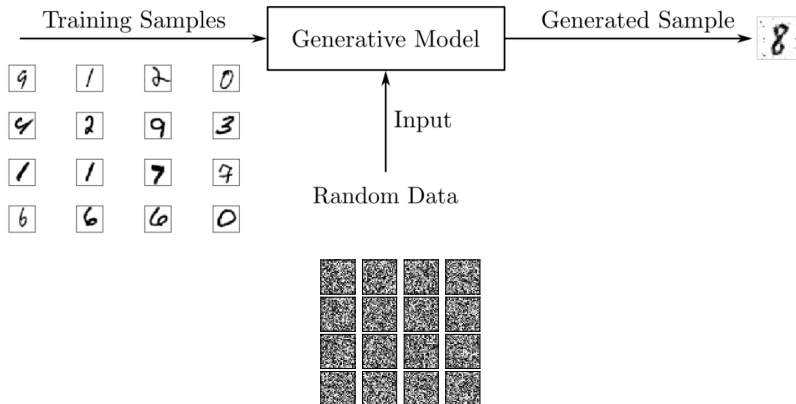
Step 3: **Training.**

Step 4: **Testing.**

This is an example of **Supervised Learning**

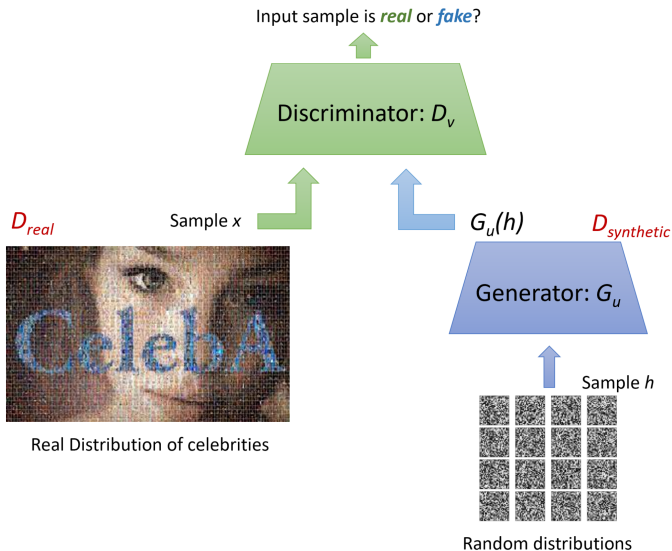
Using Deep NNs as Generative Models

This is an example of **Unsupervised Learning**

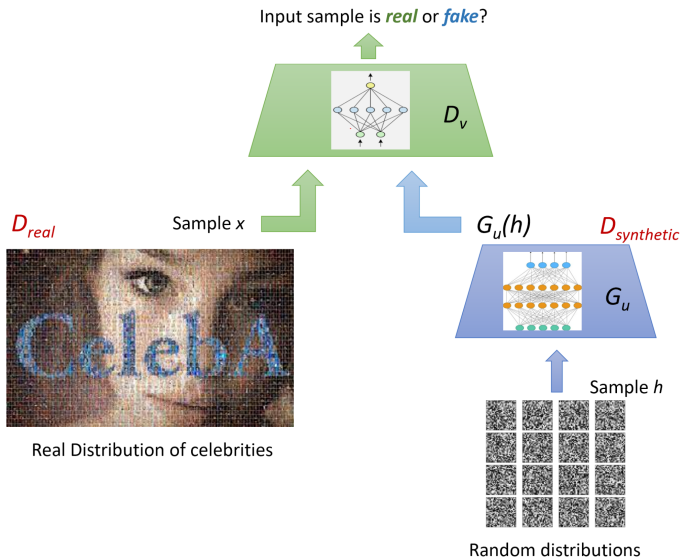


Picture from: <https://realpython.com/generative-adversarial-networks/>

The GAN game

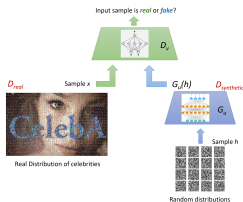


The GAN game



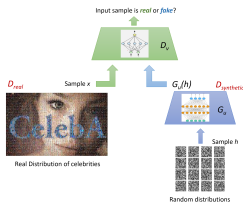
Generators and Discriminators

- $\{G_u \mid u \in \mathcal{U} \subset \mathbb{R}^p\}$, where $G_u : \mathbb{R}^\ell \rightarrow \mathbb{R}^d$, is the **class of generators with parameters u** . $\mathcal{U}$ is a subset of the unit ball.

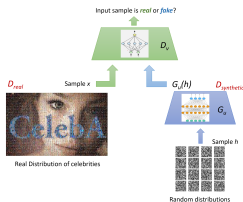


Generators and Discriminators

- $\{G_u \mid u \in \mathcal{U} \subset \mathbb{R}^p\}$, where $G_u : \mathbb{R}^\ell \rightarrow \mathbb{R}^d$, is the **class of generators with parameters u** . $\mathcal{U}$ is a subset of the unit ball.
- The generator G_u defines a distribution $\mathcal{D}_{G_u}$: generate h and then apply G_u to get $x = G_u(h)$ of $\mathcal{D}_{G_u}$.

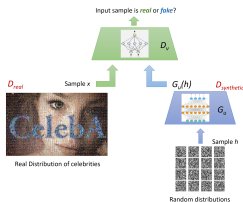


Generators and Discriminators



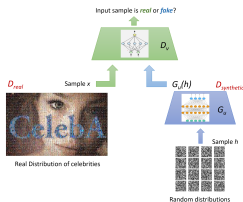
- $\{G_u \mid u \in \mathcal{U} \subset \mathbb{R}^p\}$, where $G_u : \mathbb{R}^\ell \rightarrow \mathbb{R}^d$, is the **class of generators with parameters u** . $\mathcal{U}$ is a subset of the unit ball.
- The generator G_u defines a distribution $\mathcal{D}_{G_u}$: generate h and then apply G_u to get $x = G_u(h)$ of $\mathcal{D}_{G_u}$.
- $\{D_v \mid v \in \mathcal{V}\}$, where $D_v : \mathbb{R}^d \rightarrow [0..1]$, is the **class of discriminators with parameters v** .

Generators and Discriminators



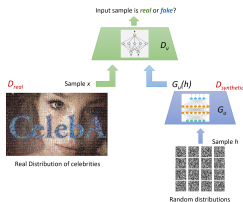
- $\{G_u \mid u \in \mathcal{U} \subset \mathbb{R}^p\}$, where $G_u : \mathbb{R}^\ell \rightarrow \mathbb{R}^d$, is the **class of generators with parameters u** . $\mathcal{U}$ is a subset of the unit ball.
- The generator G_u defines a distribution $\mathcal{D}_{G_u}$: generate h and then apply G_u to get $x = G_u(h)$ of $\mathcal{D}_{G_u}$.
- $\{D_v \mid v \in \mathcal{V}\}$, where $D_v : \mathbb{R}^d \rightarrow [0..1]$, is the **class of discriminators with parameters v** .
- $D_v(x)$ is the probability that x comes from $\mathcal{D}_{real}$.

Generators and Discriminators



- $\{G_u \mid u \in \mathcal{U} \subset \mathbb{R}^p\}$, where $G_u : \mathbb{R}^\ell \rightarrow \mathbb{R}^d$, is the **class of generators with parameters u** . $\mathcal{U}$ is a subset of the unit ball.
- The generator G_u defines a distribution $\mathcal{D}_{G_u}$: generate h and then apply G_u to get $x = G_u(h)$ of $\mathcal{D}_{G_u}$.
- $\{D_v \mid v \in \mathcal{V}\}$, where $D_v : \mathbb{R}^d \rightarrow [0..1]$, is the **class of discriminators with parameters v** .
- $D_v(x)$ is the probability that x comes from $\mathcal{D}_{real}$.
- $1 - D_v(x)$ is the probability that x comes from $\mathcal{D}_{G_u}$.

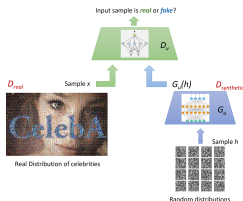
Generators and Discriminators



- $\{G_u \mid u \in \mathcal{U} \subset \mathbb{R}^p\}$, where $G_u : \mathbb{R}^\ell \rightarrow \mathbb{R}^d$, is the **class of generators with parameters u** . $\mathcal{U}$ is a subset of the unit ball.
- The generator G_u defines a distribution $\mathcal{D}_{G_u}$: generate h and then apply G_u to get $x = G_u(h)$ of $\mathcal{D}_{G_u}$.
- $\{D_v \mid v \in \mathcal{V}\}$, where $D_v : \mathbb{R}^d \rightarrow [0..1]$, is the **class of discriminators with parameters v** .
- $D_v(x)$ is the probability that x comes from $\mathcal{D}_{real}$.
- $1 - D_v(x)$ is the probability that x comes from $\mathcal{D}_{G_u}$.
- We assume G_u and D_v are L -Lipschitz with respect to the parameters. For all $u, u' \in \mathcal{U}$ and any input h :

$$\|G_u(h) - G_{u'}(h)\| \leq L \|u - u'\|.$$

Generators and Discriminators



- $\{G_u \mid u \in \mathcal{U} \subset \mathbb{R}^p\}$, where $G_u : \mathbb{R}^\ell \rightarrow \mathbb{R}^d$, is the **class of generators with parameters u** . $\mathcal{U}$ is a subset of the unit ball.
- The generator G_u defines a distribution $\mathcal{D}_{G_u}$: generate h and then apply G_u to get $x = G_u(h)$ of $\mathcal{D}_{G_u}$.
- $\{D_v \mid v \in \mathcal{V}\}$, where $D_v : \mathbb{R}^d \rightarrow [0..1]$, is the **class of discriminators with parameters v** .
- $D_v(x)$ is the probability that x comes from $\mathcal{D}_{real}$.
- $1 - D_v(x)$ is the probability that x comes from $\mathcal{D}_{G_u}$.
- We assume G_u and D_v are L -Lipschitz with respect to the parameters. For all $u, u' \in \mathcal{U}$ and any input h :

$$\|G_u(h) - G_{u'}(h)\| \leq L \|u - u'\|.$$

Training a GAN: a min-max game

The training of a GAN consists in optimizing the objective function:

$$\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} \mathbb{E}_{x \sim \mathcal{D}_{real}} [\log D_v(x)] + \mathbb{E}_{x \sim \mathcal{D}_{G_u}} [\log(1 - D_v(x))] \quad (1)$$

Objective function

Training a GAN: a min-max game

The training of a GAN consists in optimizing the objective function:

$$\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} \mathbb{E}_{x \sim \mathcal{D}_{real}} [\log D_v(x)] + \mathbb{E}_{x \sim \mathcal{D}_{G_u}} [\log(1 - D_v(x))] \quad (2)$$

Notice that $\log x \rightarrow -\infty$ as $x \rightarrow 0$ (numerical issues).

Objective function

Training a GAN: a min-max game

The training of a GAN consists in optimizing the objective function:

$$\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} \mathbb{E}_{x \sim \mathcal{D}_{real}} [\log D_v(x)] + \mathbb{E}_{x \sim \mathcal{D}_{G_u}} [\log(1 - D_v(x))] \quad (2)$$

Notice that $\log x \rightarrow -\infty$ as $x \rightarrow 0$ (numerical issues).

We can replace $\log(\cdot)$ with a monotone function $\phi : [0..1] \rightarrow \mathbb{R}$:

$$\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} \mathbb{E}_{x \sim \mathcal{D}_{real}} [\phi(D_v(x))] + \mathbb{E}_{x \sim \mathcal{D}_{G_u}} [\phi(1 - D_v(x))] \quad (3)$$

We call $\phi(\cdot)$ the *measuring function*.

Objective function

Training a GAN: a min-max game

The training of a GAN consists in optimizing the objective function:

$$\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} \mathbb{E}_{x \sim \mathcal{D}_{real}} [\log D_v(x)] + \mathbb{E}_{x \sim \mathcal{D}_{G_u}} [\log(1 - D_v(x))] \quad (2)$$

Notice that $\log x \rightarrow -\infty$ as $x \rightarrow 0$ (numerical issues).

We can replace $\log(\cdot)$ with a monotone function $\phi : [0..1] \rightarrow \mathbb{R}$:

$$\min_{u \in \mathcal{U}} \max_{v \in \mathcal{V}} \mathbb{E}_{x \sim \mathcal{D}_{real}} [\phi(D_v(x))] + \mathbb{E}_{x \sim \mathcal{D}_{G_u}} [\phi(1 - D_v(x))] \quad (3)$$

We call $\phi(\cdot)$ the *measuring function*.

In practice, we have m training samples $x_1, \dots, x_m \sim \mathcal{D}_{real}$, and instead of the expectation, we use the *empirical distribution*:

$$\frac{1}{m} \sum_{i=1}^m \phi(D_v(x_i))$$

Optimal Discriminator

Training the discriminator

Under the assumption to have access to **infinite** number of examples for $\mathcal{D}_{real}$, the optimal discriminator function $D : \mathbb{R}^d \rightarrow [0..1]$ is equal to:

$$D(x) = \frac{P_{real}}{P_{real} + P_{G_u}(x)},$$

where P_{real} is the density of x in the real distribution.

Optimal Discriminator

Training the discriminator

Under the assumption to have access to **infinite** number of examples for $\mathcal{D}_{real}$, the optimal discriminator function $D : \mathbb{R}^d \rightarrow [0..1]$ is equal to:

$$D(x) = \frac{P_{real}}{P_{real} + P_{G_u}(x)},$$

where P_{real} is the density of x in the real distribution.

REMARK: This allows removing, in some cases, the maximum operation from the objective function used for the training.

Training Algorithm (Goodfellow et al 2014)

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 $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Sample minibatch of m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
- Update the discriminator by ascending its stochastic gradient:

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

end for

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

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

end for

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

GAN Advantages

- 1 **Sharp Images:** GANs produce sharper images than other generative models.

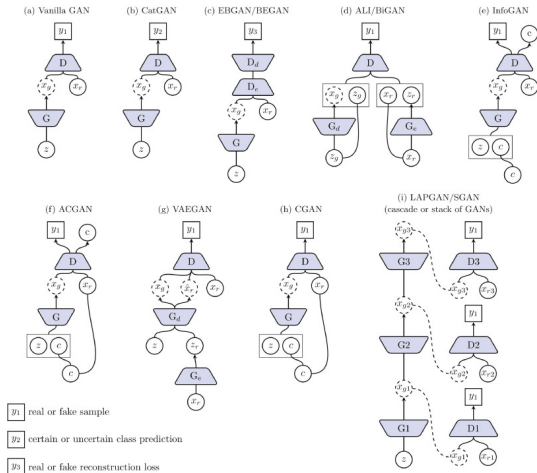
GAN Advantages

- 1 Sharp Images:** GANs produce sharper images than other generative models.
- 2 Configurable size:** The latent random variable size is unrestricted, enriching the generator search space.

GAN Advantages

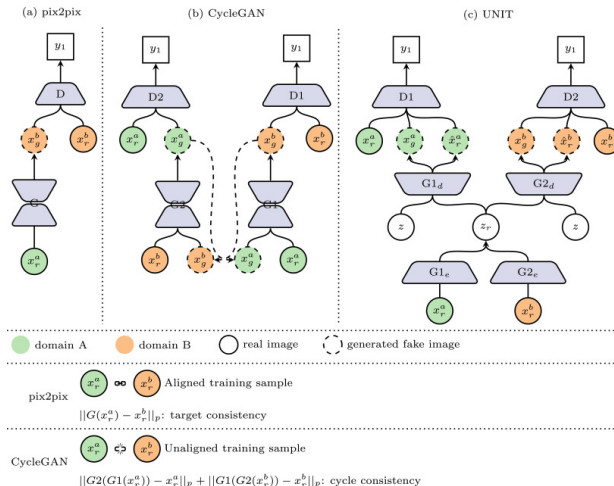
- 1 Sharp Images:** GANs produce sharper images than other generative models.
- 2 Configurable size:** The latent random variable size is unrestricted, enriching the generator search space.
- 3 Versatile generator:** The GAN framework can support many different generator networks (*it's a GAN zoo nowadays...*)

GAN Zoo



Yi, X., Walia, E. and Babyn, P., 2019. *Generative adversarial network in medical imaging: A review*. Medical image analysis, 58, p.101552.

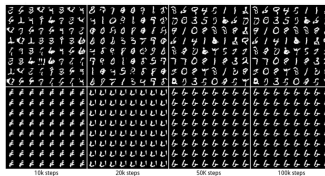
GAN Zoo



Yi, X., Walia, E. and Babyn, P., 2019. *Generative adversarial network in medical imaging: A review*. Medical image analysis, 58, p.101552.

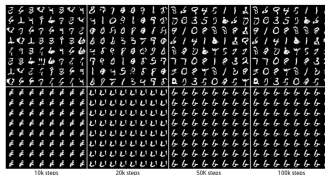
GAN Issues

- 1 Mode collapse:** The generator tends to produce a specific pattern which fools the discriminator.



GAN Issues

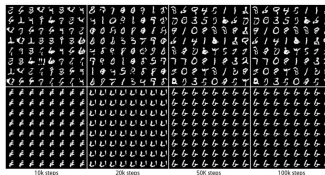
- 1 Mode collapse:** The generator tends to produce a specific pattern which fools the discriminator.



- 2 Vanishing gradients:** The training step of the generator backpropagates very low gradients (no info!).

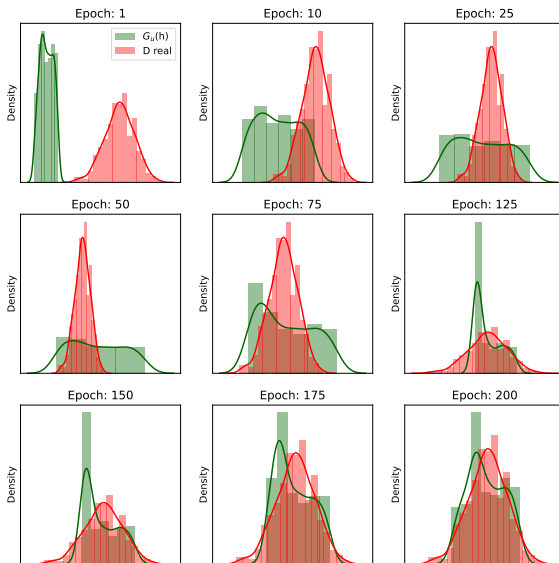
GAN Issues

- Mode collapse:** The generator tends to produce a specific pattern which fools the discriminator.

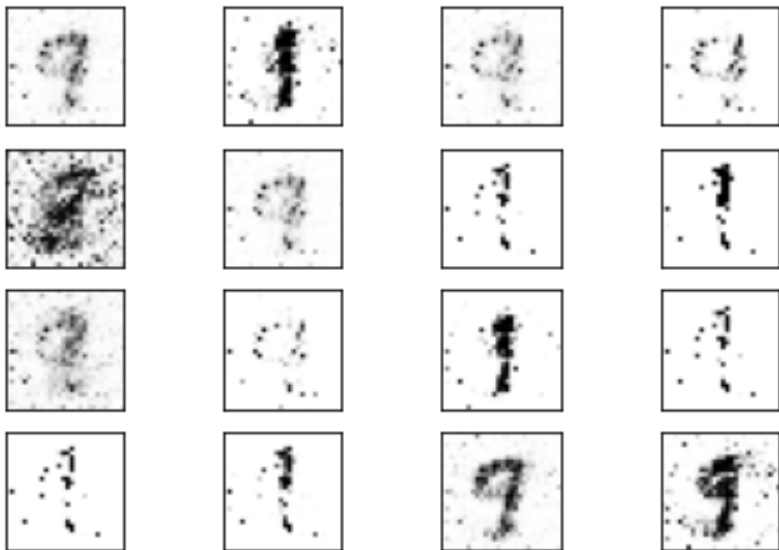


- Vanishing gradients:** The training step of the generator backpropagates very low gradients (no info!).
- Instability:** The parameters u and v oscillate during training and they do not converge toward an equilibrium (low quality images in output).

Google Colab 1: Learn a Normal Distribution



Google Colab 2: Learn MNIST Digits



Main References

- Generalization and Equilibrium in Generative Adversarial Nets (GANs).
<https://www.youtube.com/watch?v=V7TliSCq0wI>
- Generative Adversarial Networks: Build Your First Models
<https://realpython.com/generative-adversarial-networks/>