

# Generative Adversarial Networks (GANs)

---

Advanced Institute for Artificial Intelligence – AI2

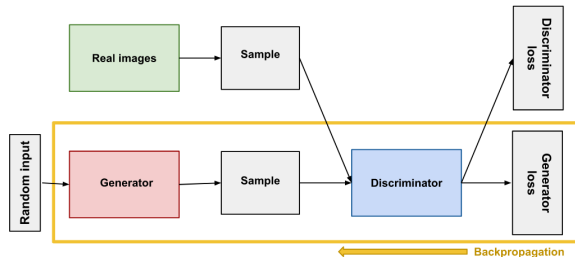
<https://advancedinstitute.ai>

## GANs são um tipo de arquitetura de Redes Neurais para o treinamento de modelos generativos

- **Modelagem Generativa** consiste em criar um modelo que gera novos exemplos seguindo uma "distribuição" específica
  - Por exemplo, gerando fotografias realísticas mas diferentes das fotos utilizadas durante o treinamento

Uma GAN é um modelo generativo treinado utilizando 2 Redes Neurais.

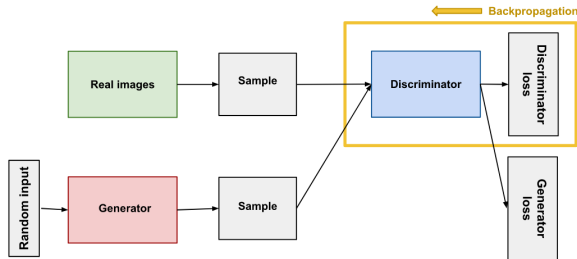
- **Gerador:** aprende a gerar dados plausíveis. As instâncias geradas tornam-se exemplos de treinamento negativo para o discriminador.



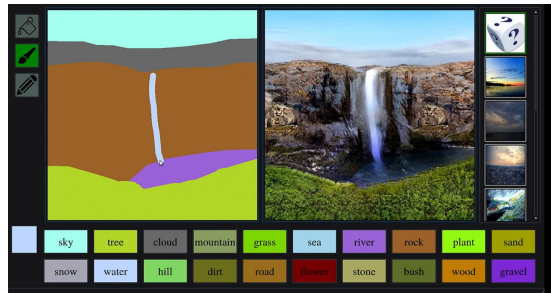
Uma GAN é um modelo generativo treinado utilizando 2 Redes Neurais.

- **Discriminador:** discrimina entre diferentes tipos de instâncias de dados. O discriminador aprende a distinguir os dados **falsos** do gerador dos dados reais. O discriminador penaliza o gerador por produzir resultados implausíveis.

O *Gerador* e o *Discriminador* competem em "um jogo".

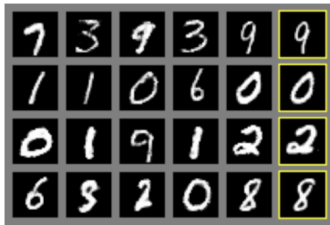


- Após o treinamento, o *Gerador* pode ser utilizado para criar novos exemplos realísticos.



NVIDIA GauGAN:

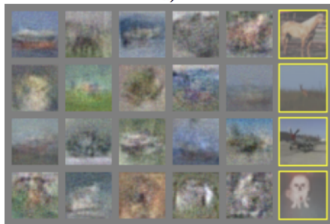
Link encurtado: [advancedinstitute.ai/gaugan](https://advancedinstitute.ai/gaugan)



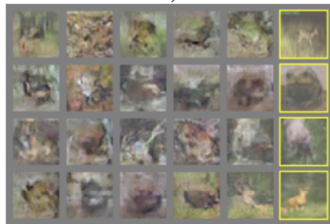
a)



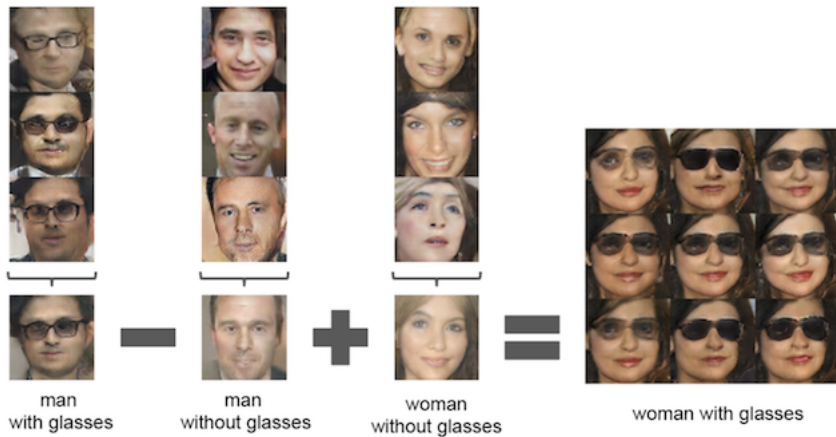
b)



c)



d)



Input



Output



Input



Output





Real

Input

Ours

NN



# Como as GANs são treinadas?

- Buscamos a convergência de um modelo em um conjunto de dados minimizando da função Loss no conjunto de dados de treinamento.
- O objetivo é equilíbrio entre a perda do gerador e do discriminador.
- Função de perda "padrão": Non-Saturating GAN Loss
- Loss no Discriminador: maximizar a probabilidade atribuída a imagens reais e falsas.
- Matematicamente, o discriminador busca maximizar a probabilidade para imagens reais e a inversa da probabilidade das imagens fakes.  
maximizar  $\log D(x) + \log(1 - D(G(z)))$

**Leitura recomendada:** Are GANs Created Equal? A Large-Scale Study

<https://arxiv.org/abs/1711.10337>

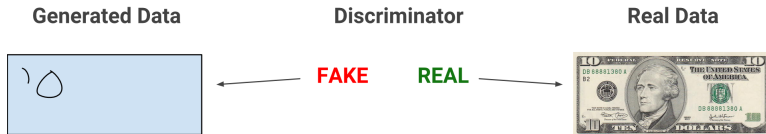
- Com essa função, o *Discriminador* tenta aprender a distinguir exemplos sintéticos dos reais.
- Em contraste, o **Gerador** tenta gerar exemplos sintéticos indistinguíveis dos reais.
- A forma mais simples de modelar essa dinâmica é através de um *jogo de soma zero*.

$$\mathcal{L}_D = -\mathcal{L}_G \tag{1}$$

- O que significa que o *Gerador* vai tentar enganar o *Discriminador*.

# O que acontece durante o treinamento?

## Início



## Após um tempo treinando



## Após convergência



## O Treinamento geralmente é realizado em ciclos:

- 1 Treina o  $D(x)$  para que este consiga classificar samples
- 2 Treina  $G(x)$  para que gere samples que enganem  $D$ .

# Problema do Desequilíbrio

- Um ponto crucial no design e na implementação de GANs é como resolver o problema do desequilíbrio entre o **Gerador** e o **Discriminador**.
- **Se o gerador for bem-sucedido:** o discriminador terá uma precisão de **50%** (O mesmo que lançar uma moeda).
- **Se o discriminador for bem-sucedido:** O gradiente diminui a tal ponto que o gerador deixa de aprender.



# Problema do Desequilíbrio

- **Colapse mode:** Ocorre geralmente quando o discriminador está preso em um mínimo local e não consegue distinguir entre uma entrada real e uma saída do gerador. Uma opção é usar uma função de Loss diferente.
- **Problemas de convergência:** Isso pode acontecer em casos de desequilíbrio. Uma opção é adicionar ruído as entradas do discriminador





Treinar GANs é muito desafiador, dado que é difícil para o gerador descobrir a "direção" em que seus updates devem ir para enganar o discriminador.

**Dicas Avançadas:** <https://github.com/soumith/ganhacks>