# Generative Adversarial Networks (GANs)
From Ian Goodfellow et al.

A short tutorial by :
Binglin, Shashank & Bhargav

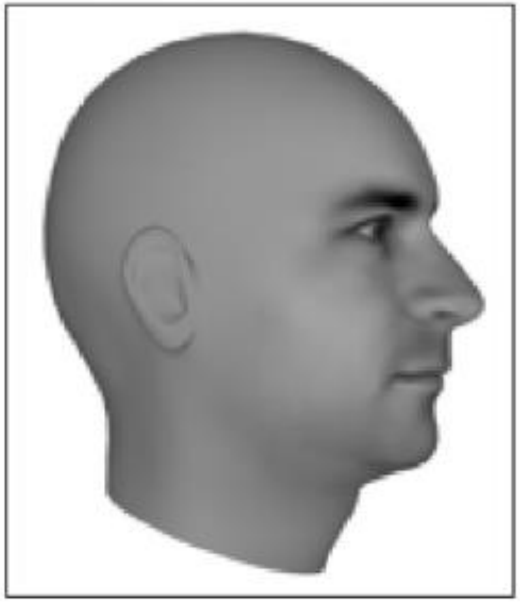
# Mágica das GANs

Qual das imagens é gerada?

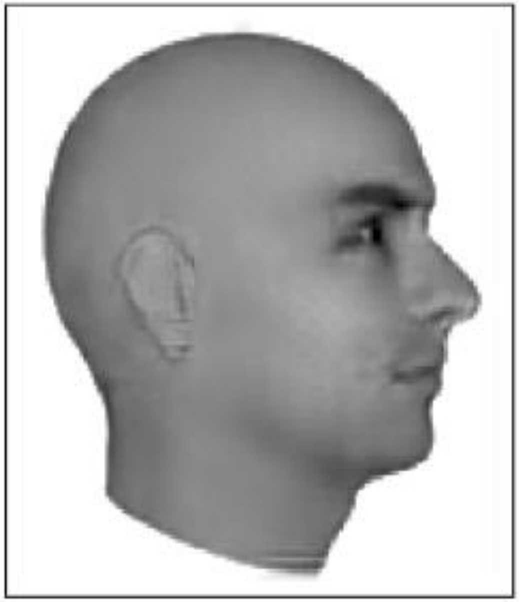
Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *arXiv preprint arXiv:1609.04802* (2016).

# Mágica das GANs

real

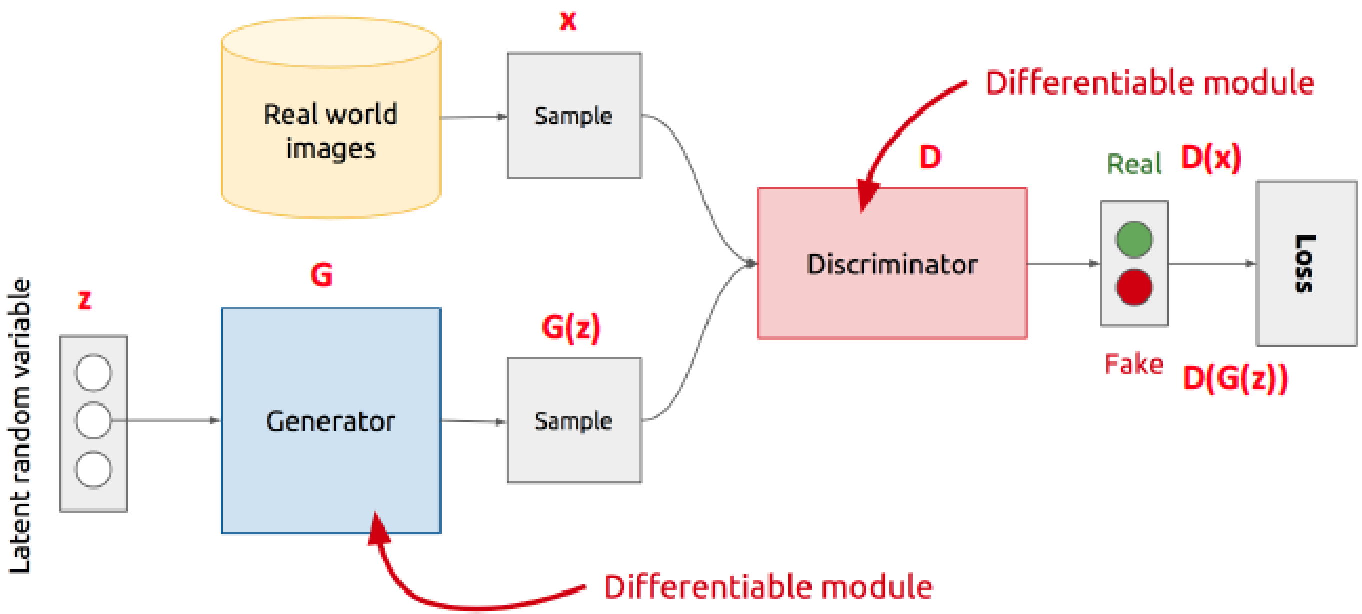


adversarial



Lotter, William, Gabriel Kreiman, and David Cox. "Unsupervised learning of visual structure using predictive generative networks." *arXiv preprint arXiv:1511.06380* (2015).

# Arquitetura da GAN




Z é um ruído Gaussiano (Gaussiano/Uniforme)

Z é pensado como uma representação latente da imagem (*n* dimensional)

adversarial-training-upc-2016

# Formulação da GAN

$$\min_G \max_D V(D,G)$$

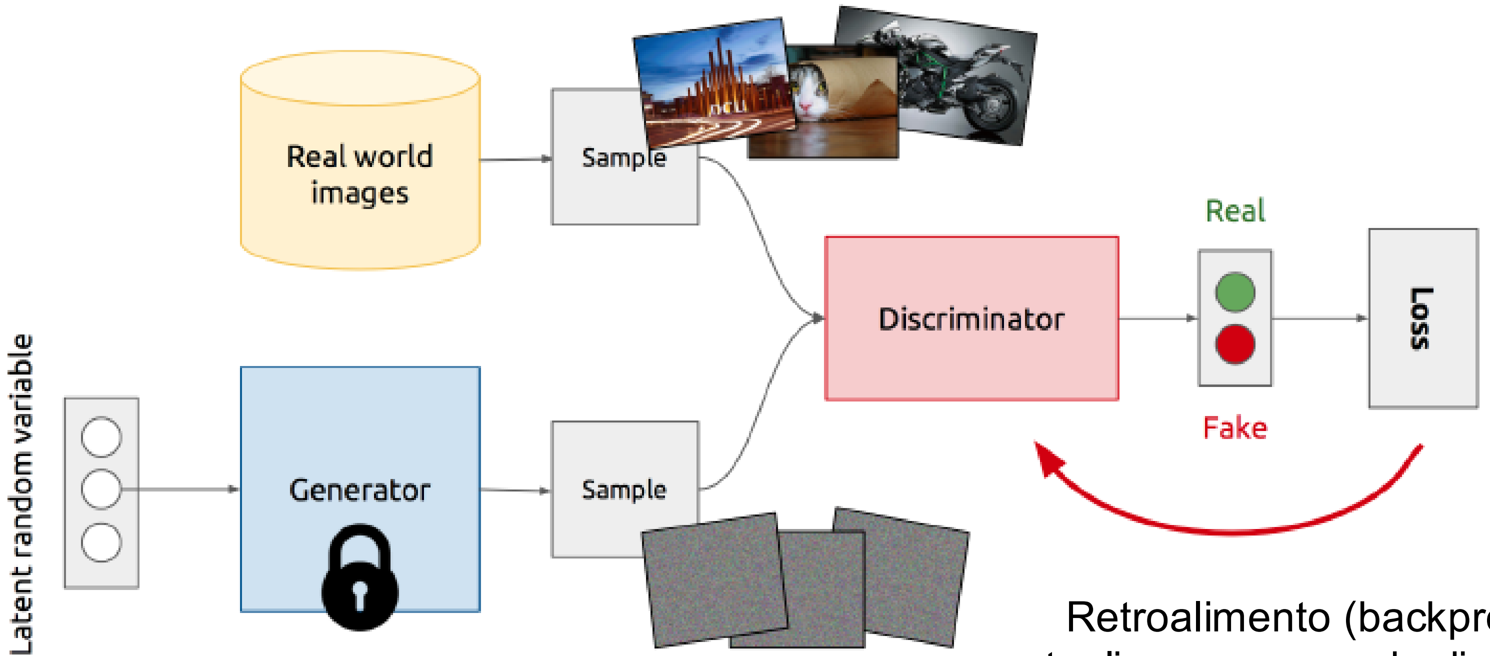
Como um jogo de Min-Max, onde:
O discriminador tenta maximizar sua recompensa (ou seja, reduzir seu erro)
O gerador tenta minimizar a recompensa do discriminador (ou sej,a maximizar sua perda)

Log Loss (Cross-Entropy)

$$V(D,G) = \boxed{\mathbb{E}_{x\sim p(x)}[\log D(x)]} + \boxed{\mathbb{E}_{z\sim q(z)}\left[\log\left(1 - D(G(z))\right)\right]}$$

Erro para o discriminador Erro para o gerador

# Treinando o Discriminador

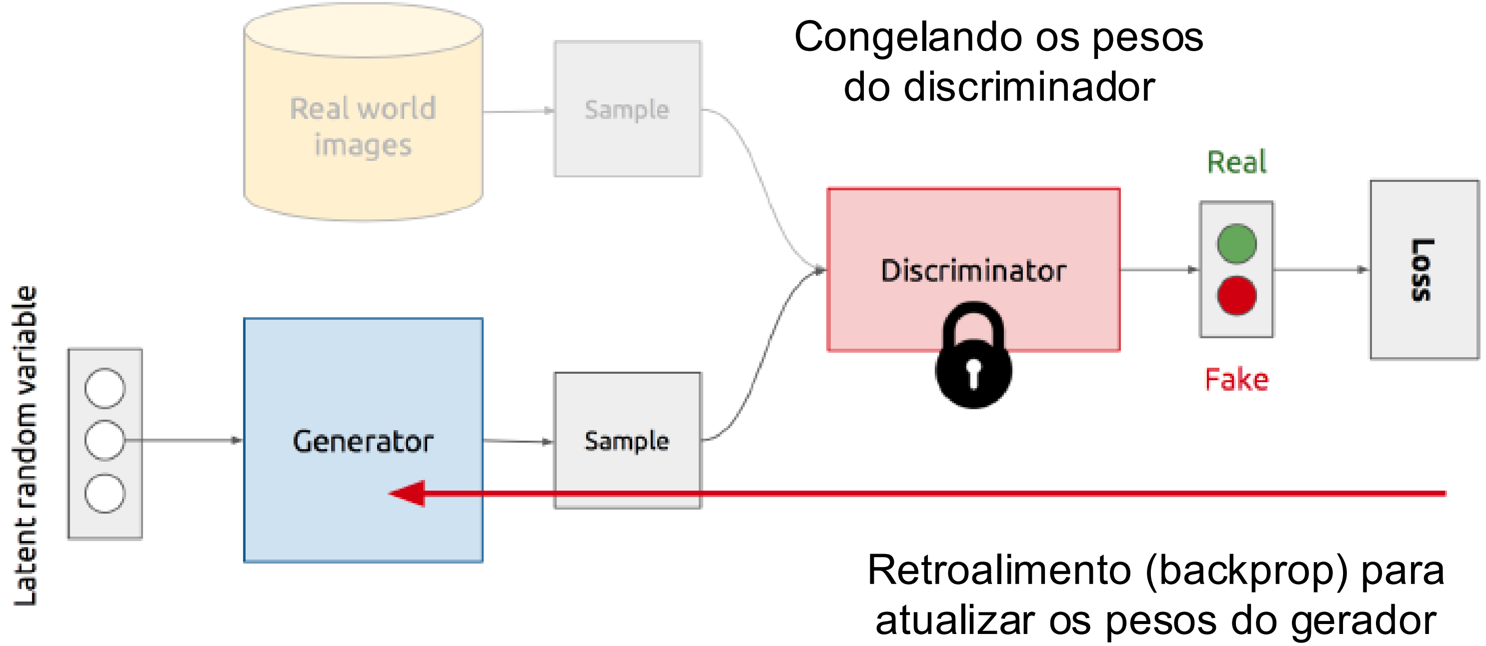



Retroalimento (backprop) para atualizar os pesos do discriminador

Congelando os pesos do gerador

share.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

# Treinando o Gerador




https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

# Algoritmo

atualização do discriminador

**for** number of training iterations **do**

**for** $k$ steps **do**

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

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

**end for**

atualização do gerador

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

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

# Log Loss

- Log Loss é a métrica de classificação mais importante baseada em probabilidades

Quais são as probabilidades corrigidas?

| ID | Actual | Predicted Probabilities | Corrected Probabilities | Log |
|---|---|---|---|---|
| ID6 | 1 | 0.94 | 0.94 | -0.02687 |
| ID1 | 1 | 0.9 | 0.9 | -0.04576 |
| ID7 | 1 | 0.78 | 0.78 | -0.10791 |
| ID8 | 0 | 0.56 | 0.44 | -0.35655 |
| ID2 | 0 | 0.51 | 0.49 | -0.3098 |
| ID3 | 1 | 0.47 | 0.47 | -0.3279 |
| ID4 | 1 | 0.32 | 0.32 | -0.49485 |
| ID5 | 0 | 0.1 | 0.9 | -0.04576 |

Média negativa

Como se vê, os valores de log são negativos. Para lidar com isso, se toma o negativo da média dos log de probabilidades

$$log\ loss = -1/N \sum_{i=1}^{N} (log\,(P i\,))$$

# Log Loss

- Log Loss é a métrica de classificação mais importante baseada em probabilidades

1. Calcula-se as probabilidades
2. Tira-se o log das probabilidades
3. Calcula-se o negativo da média do passo 2

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Binary Cross-Entropy / Log Loss

*yi* representa a classe real e *p(yi)* a probabilidade daquela classe.

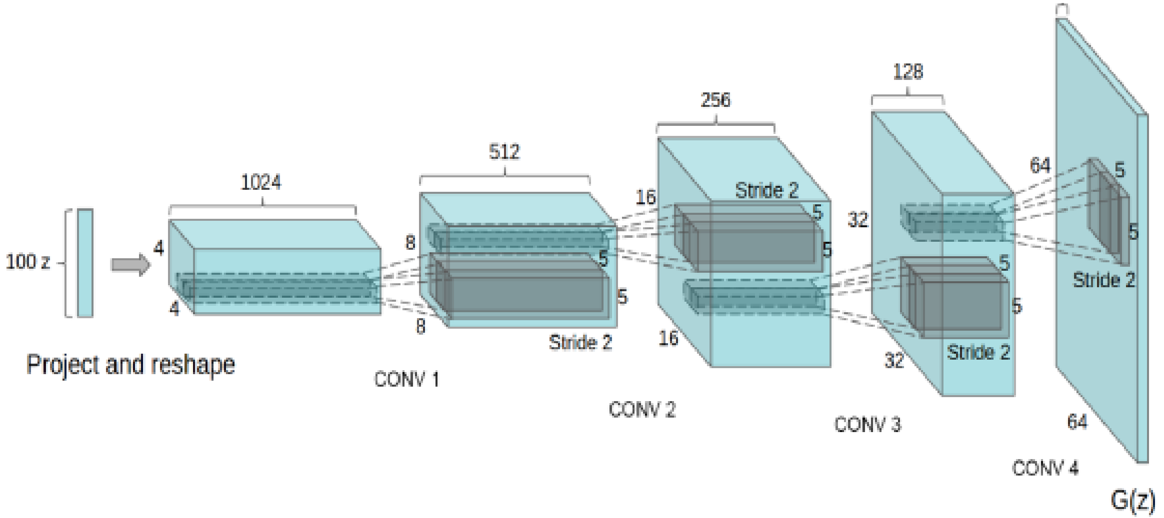
*p(yi) é a probabilidade da classe 1*
*1-p(yi) é a probabilidade da classe 0*

# Log Loss

- Log Loss é a métrica de classificação mais importante baseada em probabilidades

https://www.analyticsvidhya.com/blog/2020/11/binary-cross-entropy-aka-log-loss-the-cost-function-used-in-logistic-regression/

# GANs Convolucionais Profundas (DCGANs)




**Key ideas**:

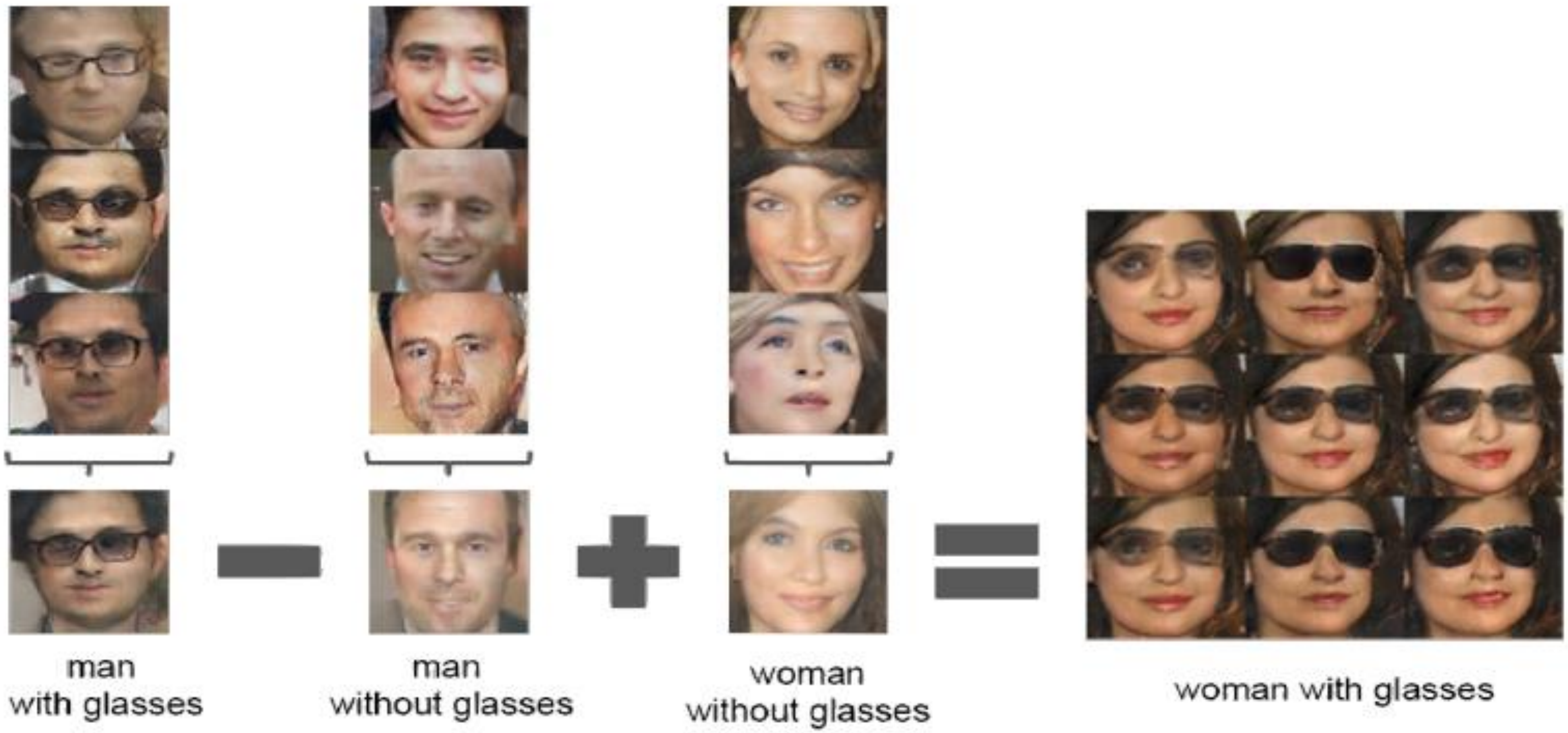
- Replace FC hidden layers with Convolutions
  - **Generator:** Fractional-Strided convolutions
- Use Batch Normalization after each layer
- **Inside Generator**
  - Use ReLU for hidden layers
  - Use Tanh for the output layer

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

# DCGAN: Bedroom images

Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

# Latent vectors capture interesting patterns...




Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv:1511.06434 (2015).

# A Friendly Introduction to Generative Adversarial Networks (GANs)

https://www.youtube.com/watch?v=8L11aMN5KY8

- Luis Serrano. Serrano.Academy