RNP-VC

# Aula 9: Aprendizado não supervisionado e Modelos Generativos

**Lucas Pereira, Rafael Teixeira, Lucas Assis, Anderson Soares**
Instituto de Informática
Universidade Federal de Goiás (UFG)

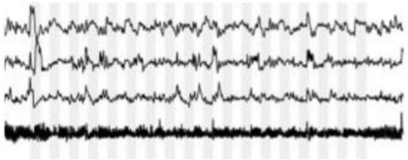
# Sumário

- No último episódio...
- Aprendizado não supervisionado
- Autoencoders
- Modelos Generativos
- Autoregressive Models
- VAEs
- GANs
- No próximo episódio...

# No último episódio...

## Sequências

“This morning I took the dog for a walk.” Texto

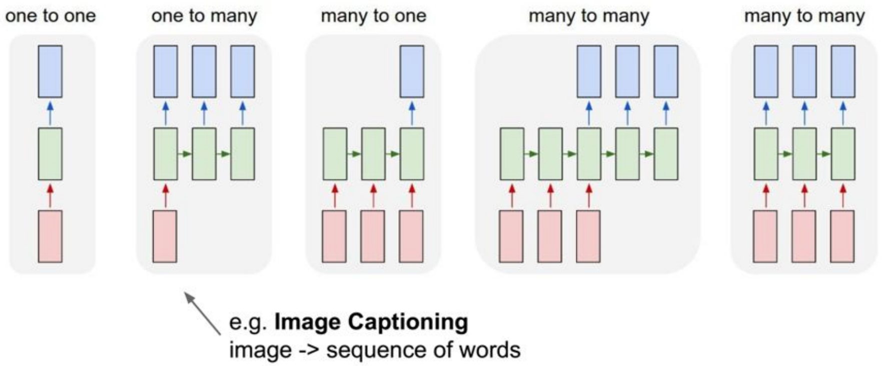


Sinais médicos

Áudio

# No último episódio...

## Recurrent Neural Networks



# Aprendizado não supervisionado

- Até agora…
  - supervisão = label!


$$CE = -\sum_{i}^{C} t_i log(s_i)$$

Dados

Modelo

Custo

GD

Otimização

# Aprendizado não supervisionado

- E se…
  - não supervisionado = não label!

?

?

Dados

Modelo

Custo

GD

Otimização

# Aprendizado não supervisionado

- E se…
  - não supervisionado = não label!

Depende!

Depende!

Dados

Modelo

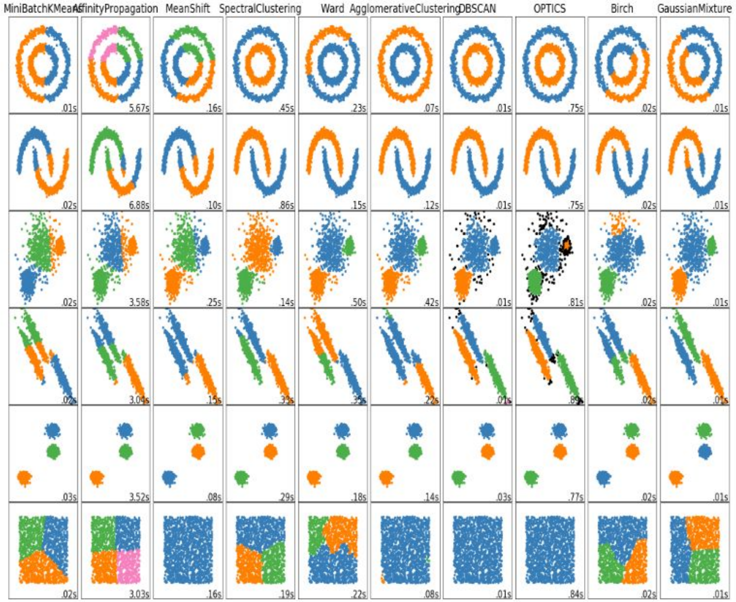
Custo

GD

Otimização

# Aprendizado não supervisionado

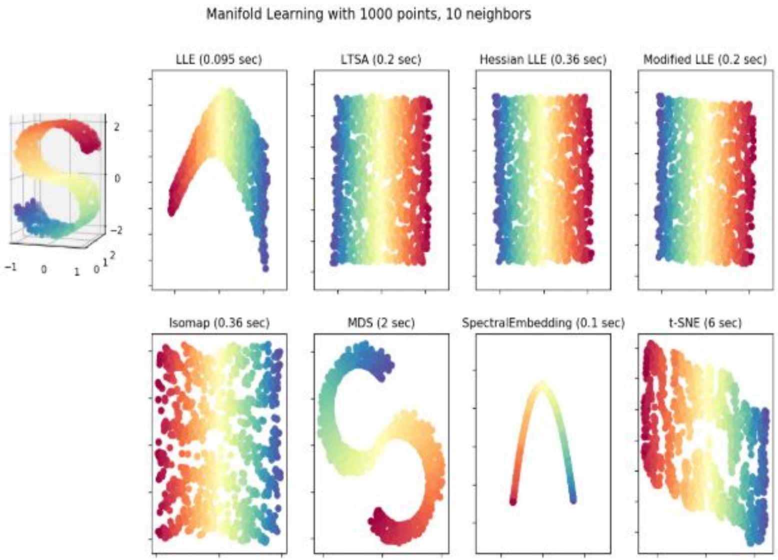
- Tarefas não supervisionadas:
    - Clusterização




A comparison of the clustering algorithms in scikit-learn

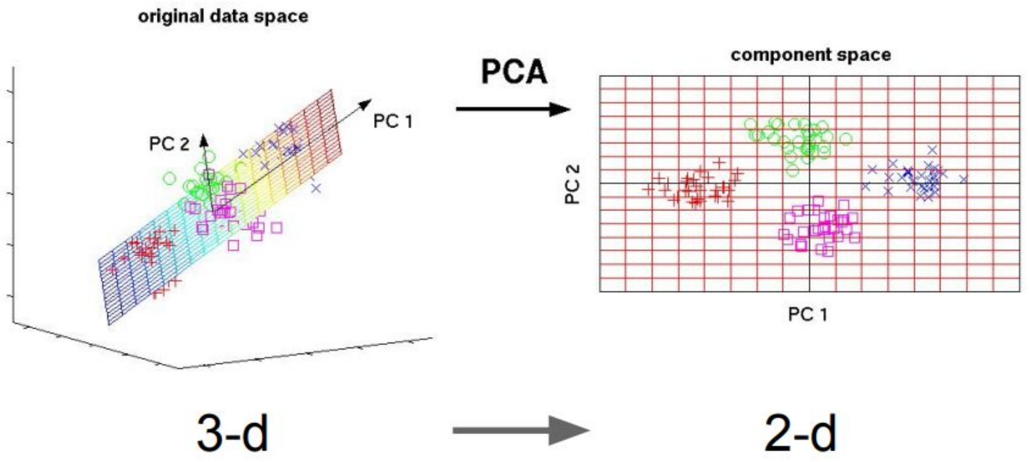
# Aprendizado não supervisionado

- Tarefas não supervisionadas:
  - Clusterização
  - Aprendizado de manifolds



# Aprendizado não supervisionado

- Tarefas não supervisionadas:
    - Clusterização
    - Aprendizado de manifolds
    - Redução de dimensionalidade



# Aprendizado não supervisionado

- Tarefas não supervisionadas:
    - Clusterização
    - Aprendizado de manifolds
    - Redução de dimensionalidade
    - Detecção de outliers
    - Estimativa de densidade de probabilidades
    - Compressão de informações
    - …

# Aprendizado não supervisionado

- Qual a solução mais simples?

?

?

Dados

Modelo

Custo

GD

Otimização

# Autoencoders

- A solução mais simples:

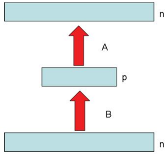



Figure 1: An n/p/n Autoencoder Architecture.

$$\|x - \hat{x}\|^2$$

Dados

Modelo

Custo

GD

Otimização

# Autoencoders

- Pra que?

# Autoencoders

- Transfer Learning

## Treino não supervisionado (Big Data)

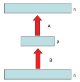


Figure 1: An n/p/n Autoencoder Architecture.

$$\|x - \hat{x}\|^2$$

Dados | Modelo | Custo

Otimização

GD

## Fine tuning supervisionado (Small Data)


$$CE = -\sum_{i}^{C} t_i log(s_i)$$

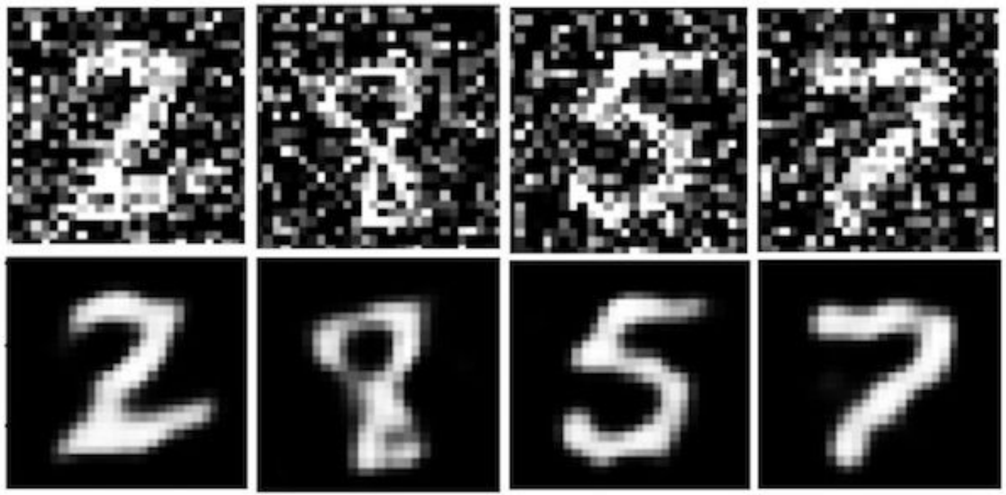
Dados | Modelo | Custo

Otimização

GD

# Autoencoders

- Transfer Learning
- Denoising



# Autoencoders

- Transfer Learning
- Denoising
- Super Resolution

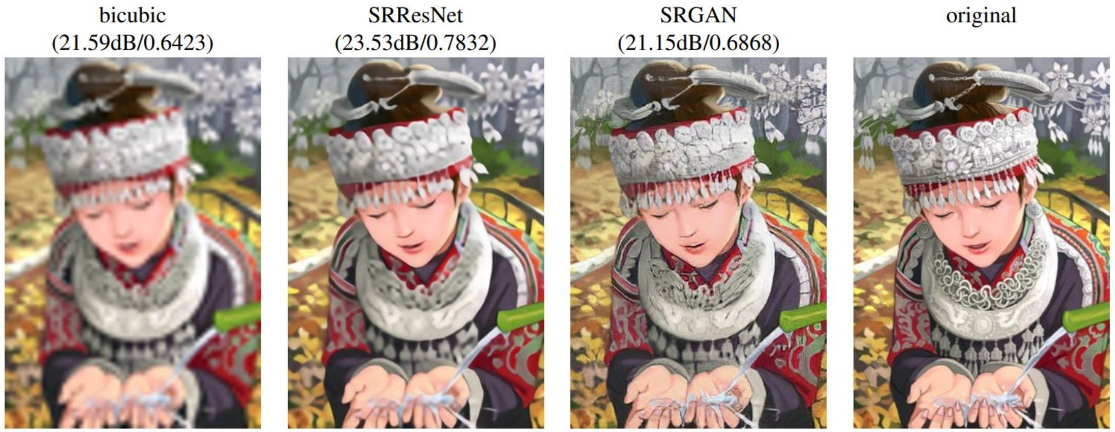



Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

# Autoencoders

- Transfer Learning
- Denoising
- Super Resolution
- …

# Aprendizado não supervisionado

- Uma solução mais complicada?

?

?

Dados

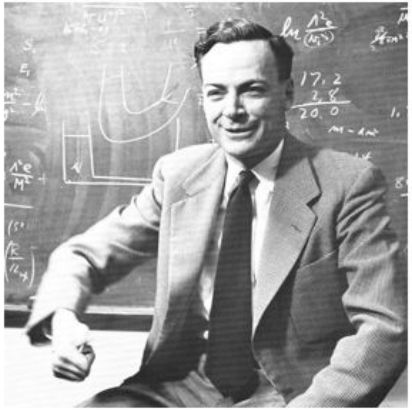
Modelo

Custo

GD

Otimização

# Modelos Generativos

Richard Feynman: “*What I cannot create, I do not understand*”

Generative modeling: “*What I understand, I can* ***create***”

Deep Generative Models

CS236 - Fall 2019

link:
https://deepgenerativemodels.github.io/

# Modelos Generativos

- Problema:
    - Cara(K) x Coroa(C)
        - Dado a sequência: KKCKCCKCKKCCCKKK, qual a probabilidade p(K)? e p(C)?

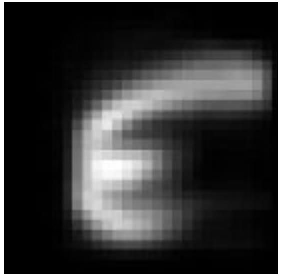
# Modelos Generativos

- Problema:
  - Cara(K) x Coroa(C)
    - Dado a sequência: KKCKCCKCKKCCCKKK, qual a probabilidade p(K)? e p(C)?
    - p(K ou C) = n(K ou C)/n(Total) (-> modelo do histograma!)

# Modelos Generativos

- Problema:
  - MNIST
    - Dado o conjunto de treinamento do MNIST, quais as probabilidades das imagens A, B e C?

A



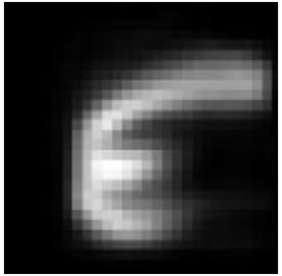
B

C

# Modelos Generativos

- Problema:
    - MNIST
        - Dado o conjunto de treinamento do MNIST, quais as probabilidades das imagens A, B e C?
        - Considerando MNIST **binário**, temos imagens de 28x28 bits para determinar a probabilidade de cada uma, i.e. $2^{28*28}$ probabilidades (**Curse of Dimensionality!!!**)

A



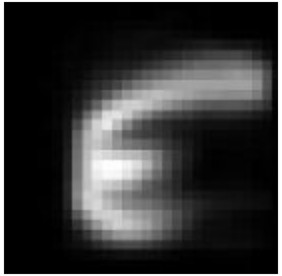
B

C

# Modelos Generativos

- Problema:
  - MNIST
    - Dado o conjunto de treinamento do MNIST, quais as probabilidades das imagens A, B e C?
    - Considerando MNIST **binário**, temos imagens de 28x28 bits para determinar a probabilidade de cada uma, i.e. $2^{28*28}$ probabilidades (**Curse of Dimensionality!!!**)
    - Solução: redes neurais para aproximar $p_{NN}(img) \approx p_{data}(img)$

A



B

C

# Modelos Generativos

- Maximum Likelihood
  - Se o dataset foi coletado, ele tem probabilidade máxima de acontecer.

$$\arg\min_{\theta} \ \text{loss}(\theta, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}) = \frac{1}{n}\sum_{i=1}^{n} -\log p_\theta(\mathbf{x}^{(i)})$$

$$\text{for all } \theta, \quad \sum_{\mathbf{x}} p_\theta(\mathbf{x}) = 1 \quad \text{and} \quad p_\theta(\mathbf{x}) \geq 0 \quad \text{for all } \mathbf{x}$$

# Modelos Generativos

- Maximum Likelihood
  - Se o dataset foi coletado, ele tem probabilidade máxima de acontecer.
  - Vantagens:
    - diferenciável
    - maximum likelihood + SGD = escalável para grandes datasets

$$\arg\min_{\theta} \ \mathrm{loss}(\theta, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}) = \frac{1}{n}\sum_{i=1}^{n} -\log p_\theta(\mathbf{x}^{(i)})$$

$$\text{for all } \theta, \quad \sum_{\mathbf{x}} p_\theta(\mathbf{x}) = 1 \quad \text{and} \quad p_\theta(\mathbf{x}) \geq 0 \quad \text{for all } \mathbf{x}$$

# Modelos Generativos

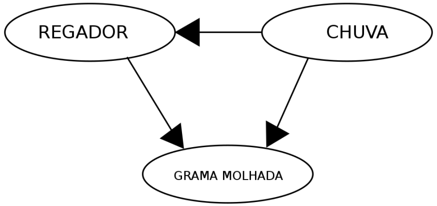
- Maximum Likelihood
  - Se o dataset foi coletado, ele tem probabilidade máxima de acontecer.
  - Vantagens:
    - diferenciável
    - maximum likelihood + SGD = escalável para grandes datasets
  - Desvantagens:
    - não é óbvio como garantir as restrições do problema de otimização!

$$\text{for all } \theta, \quad \sum_{\mathbf{x}} p_\theta(\mathbf{x}) = 1 \quad \text{and} \quad p_\theta(\mathbf{x}) \geq 0 \quad \text{for all } \mathbf{x}$$

# Autoregressive Models

- Redes Bayesianas (≠ redes neurais bayesianas!)
    - Representação do modelo por probabilidades condicionais ao invés de probabilidades absolutas

| CHUVA | REGADOR T | REGADOR F |
|---|---|---|
| F | 0.4 | 0.6 |
| T | 0.01 | 0.99 |




| CHUVA T | CHUVA F |
|---|---|
| 0.2 | 0.8 |

| REGADOR | CHUVA | GRAMA MOLHADA T | GRAMA MOLHADA F |
|---|---|---|---|
| F | F | 0.0 | 1.0 |
| F | T | 0.8 | 0.2 |
| T | F | 0.9 | 0.1 |
| T | T | 0.99 | 0.01 |

# Autoregressive Models

- Hipótese: os pixels de uma imagem podem ser ordenados em uma rede bayesiana.

$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, ..., x_{i-1})$$

Likelihood of image x

Probability of i'th pixel value given all previous pixels

$$\log p(\mathbf{x}) = \sum_{i=1}^{d} \log p(x_i \mid \mathbf{x}_{1:i-1})$$

# Autoregressive Models

- Alguma sugestão?

# Autoregressive Models

- PixelRNN

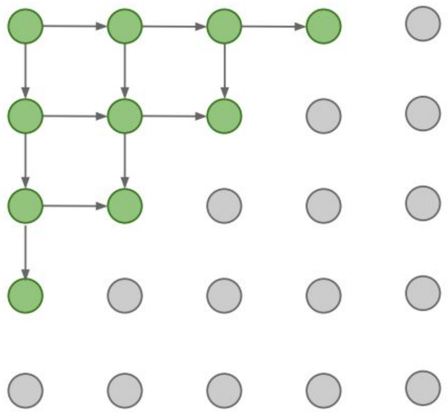
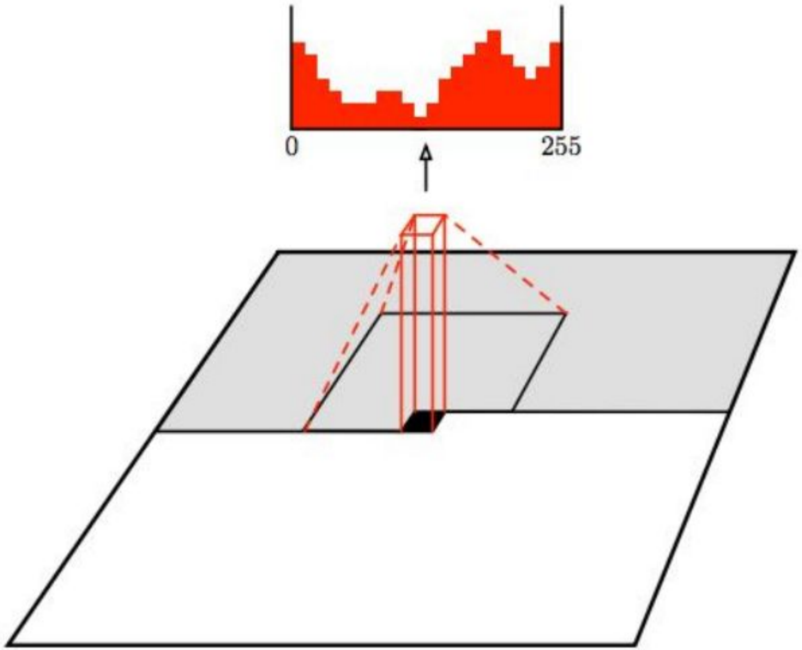



*Figure 1.* Image completions sampled from a PixelRNN.

# Autoregressive Models

- PixelCNN



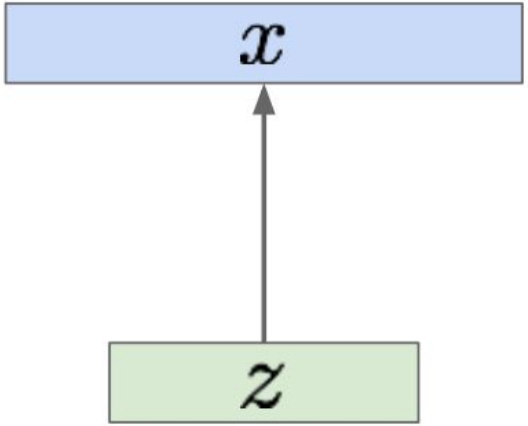
# Autoregressive Models

- PixelRNN e PixelCNN
  - Vantagens:
    - Faz a inferência de $p_{NN}$(img) de forma paralelizável e rápida
    - Modelo explícito de p(img) = fácil avaliação!
  - Desvantagens:
    - Geração de novas amostras é sequencial pixel a pixel = processo lento!

# Variational Autoencoders (VAEs)

- E se...

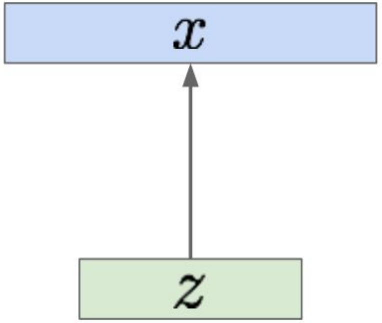
$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$



# Variational Autoencoders (VAEs)

- Problema?

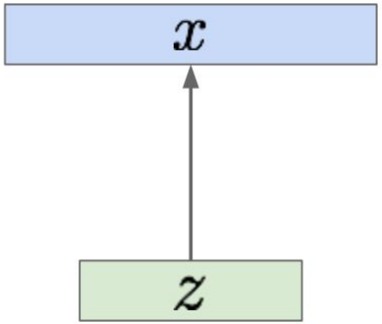
$$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$$



# Variational Autoencoders (VAEs)

- Problema:

$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$



# Variational Autoencoders (VAEs)

- Solução:

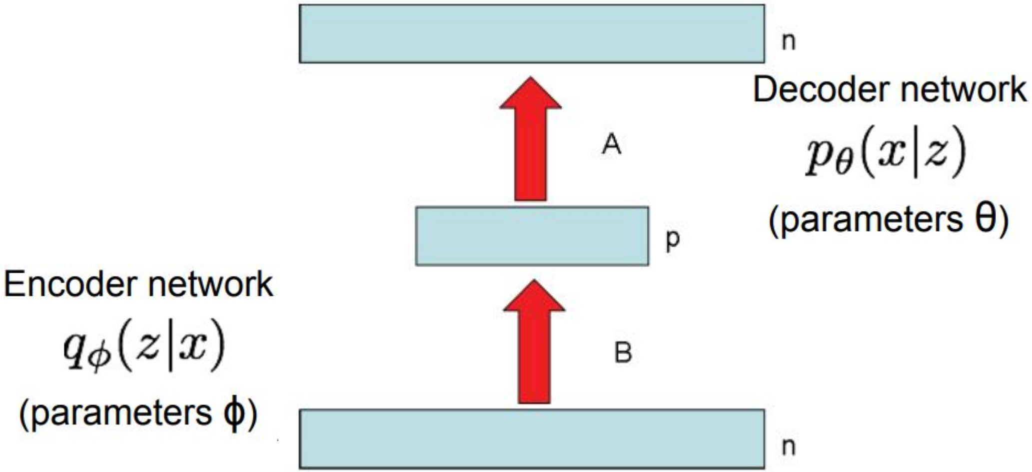



Figure 1: An n/p/n Autoencoder Architecture.

# Variational Autoencoders (VAEs)

- Variational Lower Bound (Evidence Lower Bound - ELBO)

$$\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad (\text{Multiply by constant}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Logarithms}) \\
&= \boxed{\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\boxed{\geq 0}}
\end{aligned}$$

We want to maximize the data likelihood

**Tractable lower bound** which we can take gradient of and optimize! ($p_\theta(x|z)$ differentiable, KL term differentiable)

# Variational Autoencoders (VAEs)

- Variational Lower Bound (Evidence Lower Bound - ELBO)

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Logarithms)}$$

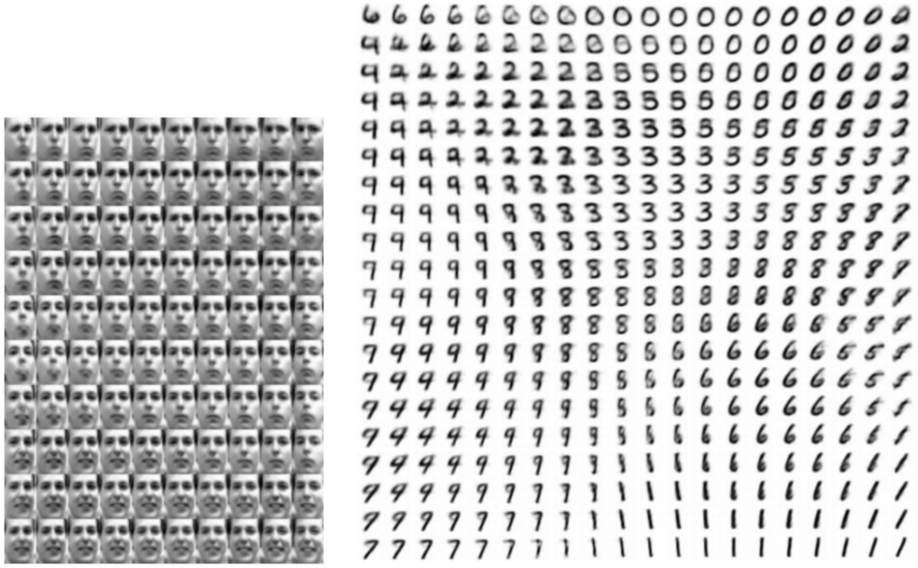
$$= \boxed{\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\boxed{\geq 0}}$$

We want to maximize the data likelihood

**Tractable lower bound** which we can take gradient of and optimize! ($p_\theta(x|z)$ differentiable, KL term differentiable)

# Variational Autoencoders (VAEs)

- Resultados



(a) Learned Frey Face manifold

(b) Learned MNIST manifold

Figure 4: Visualisations of learned data manifold for generative models with two-dimensional latent space, learned with AEVB. Since the prior of the latent space is Gaussian, linearly spaced coordinates on the unit square were transformed through the inverse CDF of the Gaussian to produce values of the latent variables $\mathbf{z}$. For each of these values $\mathbf{z}$, we plotted the corresponding generative $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ with the learned parameters $\boldsymbol{\theta}$.

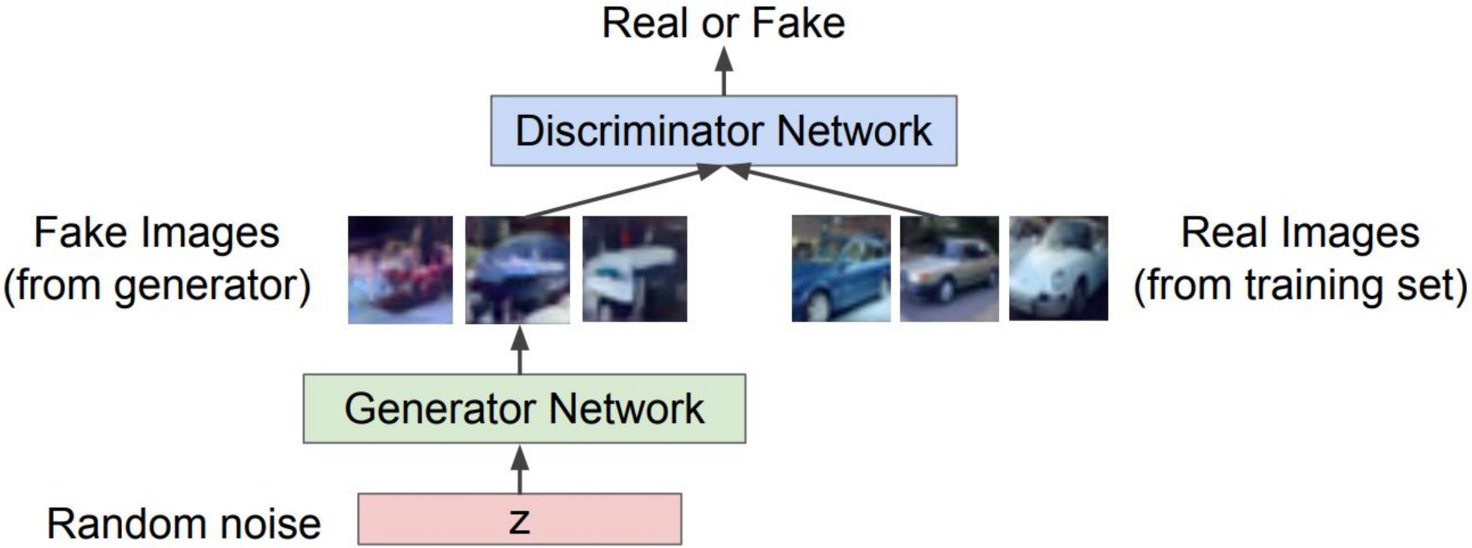
# Generative Adversarial Networks (GANs)

- E se…

    - Não interessa p(img), mas me interessa a possibilidade de gerar imagens plausíveis!

# Generative Adversarial Networks (GANs)

- Abordagem adversária
  - Otimização segundo teoria dos jogos



# Generative Adversarial Networks (GANs)

- Abordagem adversária
  - Otimização segundo teoria dos jogos
  - Na teoria:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

2. **Gradient descent** on generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

  - Na prática:

2. **Instead: Gradient ascent** on generator, different objective

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

# Generative Adversarial Networks (GANs)

- Abordagem adversária
  - Otimização segundo teoria dos jogos
  - Na teoria:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[ \mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

2. **Gradient descent** on generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

  - Na prática:

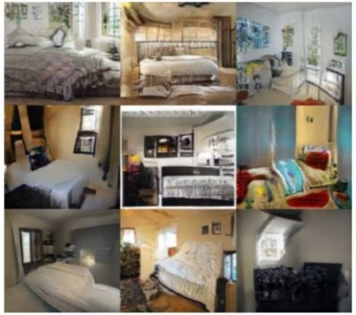
2. **Instead: Gradient ascent** on generator, different objective

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

# Generative Adversarial Networks (GANs)

## 2017: Explosion of GANs

### Better training and generation



LSGAN, Zhu 2017.

Wasserstein GAN, Arjovsky 2017. Improved Wasserstein GAN, Gulrajani 2017.

Progressive GAN, Karras 2018.

# Generative Adversarial Networks (GANs)

## 2017: Explosion of GANs

See also: https://github.com/soumith/ganhacks for tips and tricks for trainings GANs

### "The GAN Zoo"

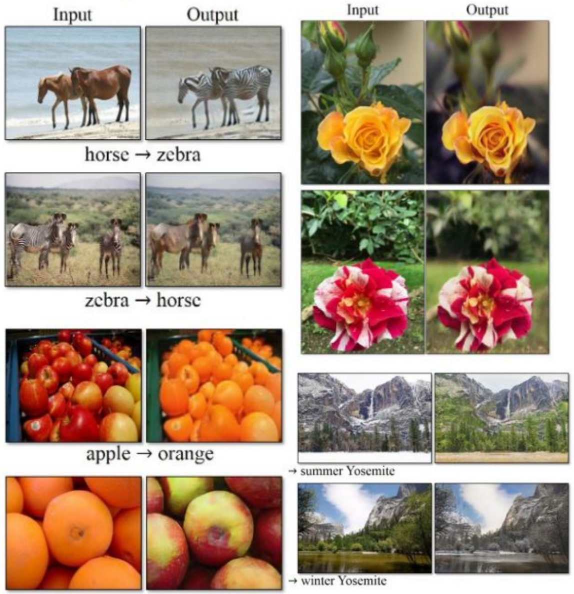
- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- AdaGAN - AdaGAN: Boosting Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference
- AM-GAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- Bayesian GAN - Deep and Hierarchical Implicit Models
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- CGAN - Conditional Generative Adversarial Nets
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks
- CCGAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CoGAN - Coupled Generative Adversarial Networks
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- DTN - Unsupervised Cross-Domain Image Generation
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DR-GAN - Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- EBGAN - Energy-based Generative Adversarial Network
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- GAWWN - Learning What and Where to Draw
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data
- Geometric GAN - Geometric GAN
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending
- IAN - Neural Photo Editing with Introspective Adversarial Networks
- iGAN - Generative Visual Manipulation on the Natural Image Manifold
- IcGAN - Invertible Conditional GANs for image editing
- ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
- Improved GAN - Improved Techniques for Training GANs
- InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

https://github.com/hindupuravinash/the-gan-zoo

# Generative Adversarial Networks (GANs)

## 2017: Explosion of GANs

### Source->Target domain transfer




CycleGAN. Zhu et al. 2017.

### Text -> Image Synthesis


Reed et al. 2017.

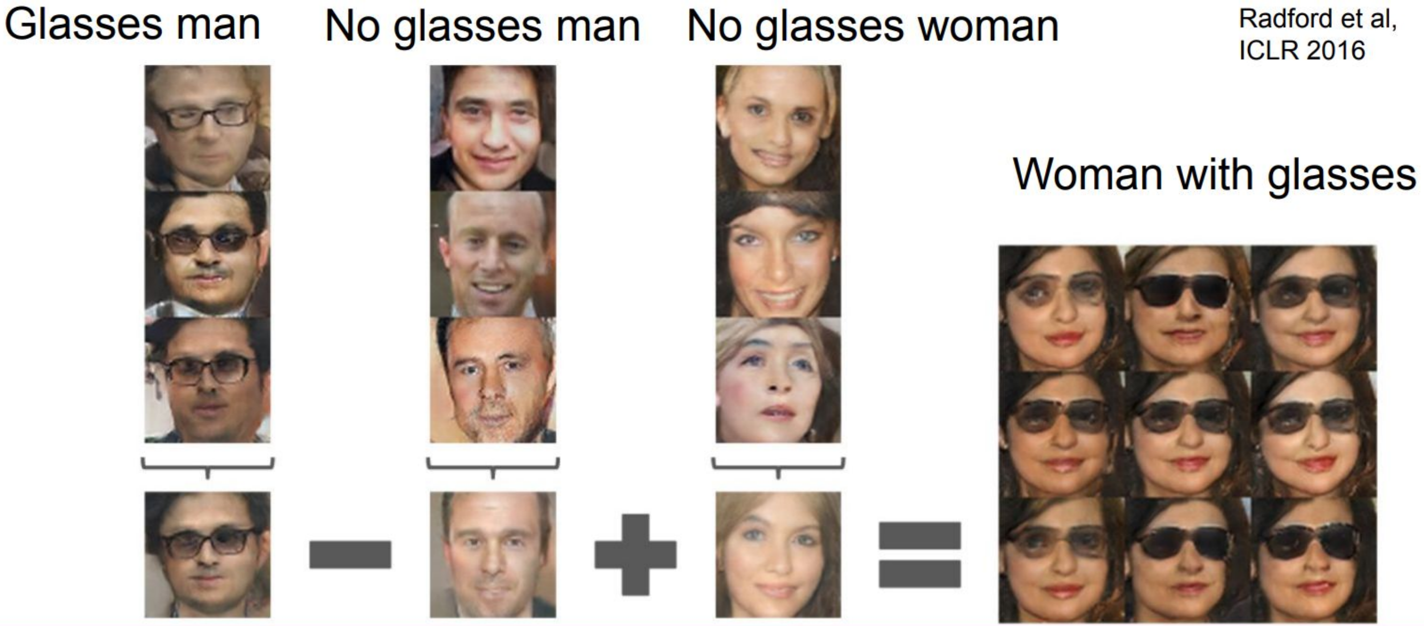
### Many GAN applications

Pix2pix. Isola 2017. Many examples at https://phillipi.github.io/pix2pix/

# Generative Adversarial Networks (GANs)

## Generative Adversarial Nets: Interpretable Vector Math

Radford et al, ICLR 2016



# Generative Adversarial Networks (GANs)

- BigGAN

Brock et al., 2019

# Generative Adversarial Networks (GANs)

Brock et al., 2019

# No próximo episódio...

- ?