



UNIVERSIDADE
FEDERAL DO CEARÁ



Aprendizagem de Máquina Probabilística

César Lincoln Cavalcante Mattos

2025

Agenda

- ① Modelos generativos
- ② Deep Latent Variable Model
- ③ Variational autoencoder
 - Reparameterization trick
 - Otimização do ELBO
- ④ Tópicos adicionais
- ⑤ Referências

Agenda

- ① Modelos generativos
- ② Deep Latent Variable Model
- ③ Variational autoencoder
 - Reparameterization trick
 - Otimização do ELBO
- ④ Tópicos adicionais
- ⑤ Referências

Modelos generativos

- **Modelos discriminativos:** aprendem preditores diretamente.
- **Modelos generativos:** aprendem a distribuição conjunta de todas as variáveis envolvidas.
 - Intenção de simular o processo em que os dados são gerados.
 - Distribuições de interesse podem ser obtidas via marginalizações e regra de Bayes.
- Modelos generativos podem ser usados em diversos contextos (supervisionados / não supervisionados) de forma isolada ou em conjunto com modelos discriminativos.

Modelo completamente observado

- Considere uma observação \mathbf{x} cuja distribuição é aproximada por um modelo com parâmetros $\boldsymbol{\theta}$ (e.g. uma rede neural profunda):

$$\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x}).$$

- Considerando um dataset disponível $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, temos:

$$\log p_{\boldsymbol{\theta}}(\mathcal{D}) = \sum_{i=1}^N \log p_{\boldsymbol{\theta}}(\mathbf{x}_i).$$

- A solução de **máxima verossimilhança** pode ser obtida via **gradiente descendente estocástico** considerando minibatches \mathcal{M} de tamanho M :

$$\frac{1}{N} \nabla \log p_{\boldsymbol{\theta}}(\mathcal{D}) \approx \frac{1}{M} \sum_{\mathbf{x}_i \in \mathcal{M}} \nabla \log p_{\boldsymbol{\theta}}(\mathbf{x}_i).$$

Modelo com variáveis latentes

- Considere a existência da **variável latente** (não observada) z .
- A **verossimilhança marginal** da observação é dada por:

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

- **Importante:**
 - Para z discreto e $p_{\theta}(x|z)$ Gaussiano, teríamos um modelo de misturas de Gaussianas;
 - Para z contínuo e $p_{\theta}(x|z)$ Gaussiano, teríamos um modelo de mistura com infinitas componentes.
 - Se o modelo for linear, temos um PPCA.

Agenda

- ① Modelos generativos
- ② Deep Latent Variable Model
- ③ Variational autoencoder
 - Reparameterization trick
 - Otimização do ELBO
- ④ Tópicos adicionais
- ⑤ Referências

Deep Latent Variable Model

- Um DLVM é um modelo de variáveis latentes $p_{\theta}(x, z)$ cujas **distribuições são parametrizadas por redes neurais**.
- Caso uma variável observada condicionante y esteja disponível (e.g., uma classe), ela pode ser incluída como $p_{\theta}(x, z|y)$.
- Um DLVM pode ser escrito por:

$$\underbrace{p_{\theta}(x, z)}_{\text{conjunta}} = \underbrace{p_{\theta}(x|z)}_{\text{verossimilhança}} \underbrace{p_{\theta}(z)}_{\text{priori}}.$$

Deep Latent Variable Model

- Um DLVM é um modelo de variáveis latentes $p_{\theta}(x, z)$ cujas **distribuições são parametrizadas por redes neurais**.
- Caso uma variável observada condicionante y esteja disponível (e.g., uma classe), ela pode ser incluída como $p_{\theta}(x, z|y)$.
- Um DLVM pode ser escrito por:

$$\underbrace{p_{\theta}(x, z)}_{\text{conjunta}} = \underbrace{p_{\theta}(x|z)}_{\text{verossimilhança}} \underbrace{p_{\theta}(z)}_{\text{priori}}.$$

- Mesmo com distribuições simples para $p_{\theta}(z)$ e $p_{\theta}(x, z)$, a marginal $p_{\theta}(x)$ pode ser **arbitrariamente complexa**.
- Escolhendo z com dimensão menor que x , estamos **compactando** as observações.
- Um DLVM é uma **alternativa não linear** ao PCA probabilístico.

Deep Latent Variable Model

- Considerando observações $\mathbf{x}_i |_{i=1}^N \in [0, 1]^D$ binárias, podemos construir um DLVM com **verossimilhança de Bernoulli**:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I}), \quad \mathbf{z} \in \mathbb{R}^K,$$

$$\mathbf{p} = \text{DecoderDNN}_{\boldsymbol{\theta}}(\mathbf{z}), \quad \mathbf{p} \in \{0, 1\}^D,$$

$$\begin{aligned}\log p(\mathbf{x} | \mathbf{z}) &= \sum_{d=1}^D \log p(x_d | \mathbf{z}) = \sum_{d=1}^D \log \text{Bern}(x_d | p_d) \\ &= \sum_{d=1}^D [x_d \log p_d + (1 - x_d) \log(1 - p_d)].\end{aligned}$$

Deep Latent Variable Model

- Considerando observações $\mathbf{x}_i |_{i=1}^N \in [0, 1]^D$ binárias, podemos construir um DLVM com **verossimilhança de Bernoulli**:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I}), \quad \mathbf{z} \in \mathbb{R}^K,$$

$$\mathbf{p} = \text{DecoderDNN}_{\boldsymbol{\theta}}(\mathbf{z}), \quad \mathbf{p} \in \{0, 1\}^D,$$

$$\begin{aligned}\log p(\mathbf{x} | \mathbf{z}) &= \sum_{d=1}^D \log p(x_d | \mathbf{z}) = \sum_{d=1}^D \log \text{Bern}(x_d | p_d) \\ &= \sum_{d=1}^D [x_d \log p_d + (1 - x_d) \log(1 - p_d)].\end{aligned}$$

- **Problema:** As distribuições $p(\mathbf{x})$ e $p(\mathbf{z} | \mathbf{x})$ não podem ser obtidas analiticamente. Lembre-se que $p(\mathbf{z} | \mathbf{x}) = p(\mathbf{x}, \mathbf{z}) / p(\mathbf{x})$.

Deep Latent Variable Model

- Considerando observações $\mathbf{x}_i |_{i=1}^N \in [0, 1]^D$ binárias, podemos construir um DLVM com **verossimilhança de Bernoulli**:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I}), \quad \mathbf{z} \in \mathbb{R}^K,$$

$$\mathbf{p} = \text{DecoderDNN}_{\theta}(\mathbf{z}), \quad \mathbf{p} \in \{0, 1\}^D,$$

$$\begin{aligned}\log p(\mathbf{x} | \mathbf{z}) &= \sum_{d=1}^D \log p(x_d | \mathbf{z}) = \sum_{d=1}^D \log \text{Bern}(x_d | p_d) \\ &= \sum_{d=1}^D [x_d \log p_d + (1 - x_d) \log(1 - p_d)].\end{aligned}$$

- **Problema:** As distribuições $p(\mathbf{x})$ e $p(\mathbf{z} | \mathbf{x})$ não podem ser obtidas analiticamente. Lembre-se que $p(\mathbf{z} | \mathbf{x}) = p(\mathbf{x}, \mathbf{z}) / p(\mathbf{x})$.
- **Problema:** Sem $p(\mathbf{x})$, não podemos obter uma solução de máxima verossimilhança para θ .

Deep Latent Variable Model

- Considerando observações $\mathbf{x}_i |_{i=1}^N \in [0, 1]^D$ binárias, podemos construir um DLVM com **verossimilhança de Bernoulli**:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I}), \quad \mathbf{z} \in \mathbb{R}^K,$$

$$\mathbf{p} = \text{DecoderDNN}_{\boldsymbol{\theta}}(\mathbf{z}), \quad \mathbf{p} \in \{0, 1\}^D,$$

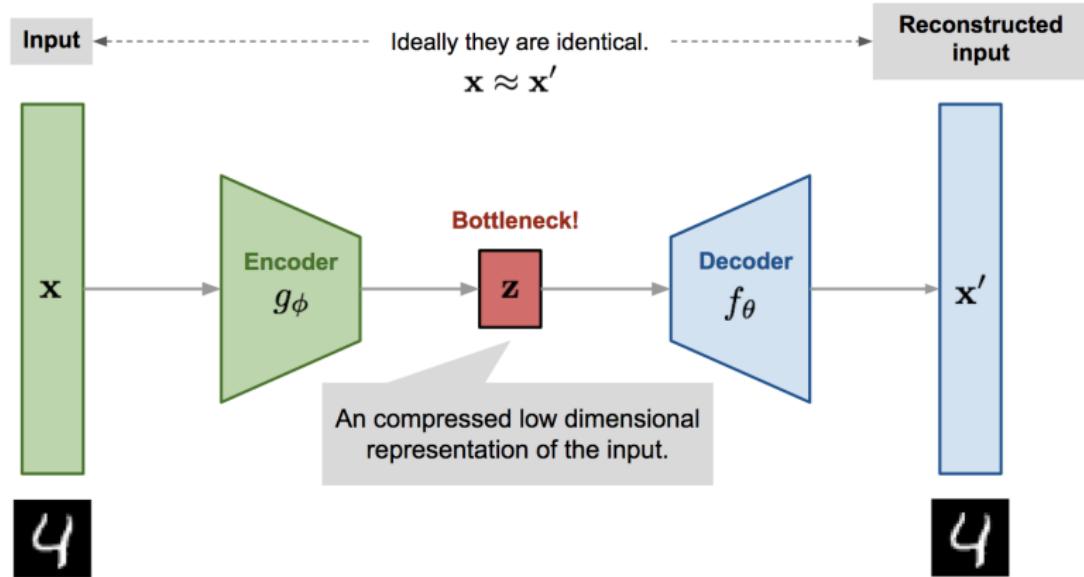
$$\begin{aligned}\log p(\mathbf{x} | \mathbf{z}) &= \sum_{d=1}^D \log p(x_d | \mathbf{z}) = \sum_{d=1}^D \log \text{Bern}(x_d | p_d) \\ &= \sum_{d=1}^D [x_d \log p_d + (1 - x_d) \log(1 - p_d)].\end{aligned}$$

- **Problema:** As distribuições $p(\mathbf{x})$ e $p(\mathbf{z} | \mathbf{x})$ não podem ser obtidas analiticamente. Lembre-se que $p(\mathbf{z} | \mathbf{x}) = p(\mathbf{x}, \mathbf{z}) / p(\mathbf{x})$.
- **Problema:** Sem $p(\mathbf{x})$, não podemos obter uma solução de máxima verossimilhança para $\boldsymbol{\theta}$.
- **Ideia:** Inferência aproximada?

Agenda

- ① Modelos generativos
- ② Deep Latent Variable Model
- ③ Variational autoencoder
 - Reparameterization trick
 - Otimização do ELBO
- ④ Tópicos adicionais
- ⑤ Referências

Autoencoder



Variational autoencoder

- Vamos considerar uma **distribuição variacional** que aproxime a **posteriori** $p(z|x)$:

$$q_{\phi}(z|x) \approx p(z|x),$$

em que q_{ϕ} é um **modelo de inferência** e ϕ são parâmetros variacionais.

Variational autoencoder

- Vamos considerar uma **distribuição variacional** que aproxime a **posteriori** $p(z|x)$:

$$q_\phi(z|x) \approx p(z|x),$$

em que q_ϕ é um **modelo de inferência** e ϕ são parâmetros variacionais.

- Seja uma distribuição variacional Gaussiana com momentos calculados por redes neurais profundas:

$$\begin{aligned}(\mu, \log \sigma) &= \text{EncoderDNN}_\phi(x), \\ q_\phi(z|x) &= \mathcal{N}(z|\mu, \text{diag}(\sigma^2)).\end{aligned}$$

- A mesma rede aproxima a posteriori de todos os dados $x \in \mathcal{D}$.
- Essa abordagem de compartilhamento de parâmetros variacionais é chamada de **inferência variacional amortizada**.

Variational autoencoder

- O *evidence lower bound* (ELBO), ou *variational lower bound*, será usado para **otimizar os parâmetros variacionais**.
- Assim, pela regra de Bayes:

$$p_{\theta}(z|x) = \frac{p_{\theta}(x, z)}{p_{\theta}(x)},$$

$$p_{\theta}(x) = \frac{p_{\theta}(x, z)}{p_{\theta}(z|x)}$$

$$\begin{aligned}\log p_{\theta}(x) &= \mathbb{E}_{q_{\phi}(z|x)} \left[\log \frac{p_{\theta}(x, z)}{p_{\theta}(z|x)} \right] \\ &= \mathbb{E}_{q_{\phi}(z|x)} \left[\log \frac{p_{\theta}(x, z)}{q_{\phi}(z|x)} \frac{q_{\phi}(z|x)}{p_{\theta}(z|x)} \right] \\ &= \underbrace{\mathbb{E}_{q_{\phi}(z|x)} \left[\log \frac{p_{\theta}(x, z)}{q_{\phi}(z|x)} \right]}_{\mathcal{L}_{\theta, \phi}(x) \text{ (ELBO)}} + \underbrace{\mathbb{E}_{q_{\phi}(z|x)} \left[\log \frac{q_{\phi}(z|x)}{p_{\theta}(z|x)} \right]}_{\text{KL}(q_{\phi}(z|x)||p_{\theta}(z|x))}.\end{aligned}$$

Variational autoencoder

- A **verossimilhança marginal** (evidência) dos dados passa a ser:

$$\log p_{\theta}(\mathbf{x}) = \mathcal{L}_{\theta, \phi}(\mathbf{x}) + \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z}|\mathbf{x})).$$

Variational autoencoder

- A **verossimilhança marginal** (evidência) dos dados passa a ser:

$$\log p_{\theta}(\mathbf{x}) = \mathcal{L}_{\theta, \phi}(\mathbf{x}) + \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z}|\mathbf{x})).$$

- O termo KL é sempre não negativo:

$$\text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z}|\mathbf{x})) \geq 0,$$

sendo igual a zero somente para distribuições iguais.

- O ELBO $\mathcal{L}_{\theta, \phi}(\mathbf{x})$ constitui um **limiar inferior** para a evidência:

$$\log p_{\theta}(\mathbf{x}) \geq \mathcal{L}_{\theta, \phi}(\mathbf{x})$$

- Note que ao maximizar o ELBO o termo KL torna-se cada vez menor, melhorando a aproximação de $p_{\theta}(\mathbf{z}|\mathbf{x})$ por $q_{\phi}(\mathbf{z}|\mathbf{x})$.

Variational autoencoder

- Voltamos a analisar o ELBO do VAE:

$$\begin{aligned}\mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \mathbb{E}_{q_{\phi}(z|x)} \left[\log \frac{p_{\theta}(\mathbf{x}, z)}{q_{\phi}(z|x)} \right] \\ &= \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(\mathbf{x}, z) - \log q_{\phi}(z|x)] \\ &= \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(\mathbf{x}|z) + \log p_{\theta}(z) - \log q_{\phi}(z|x)] \\ &= \underbrace{\mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(\mathbf{x}|z)]}_{\text{ajuste às observações}} - \underbrace{\text{KL}(q_{\phi}(z|x)||p_{\theta}(z))}_{\text{termo de regularização}}.\end{aligned}$$

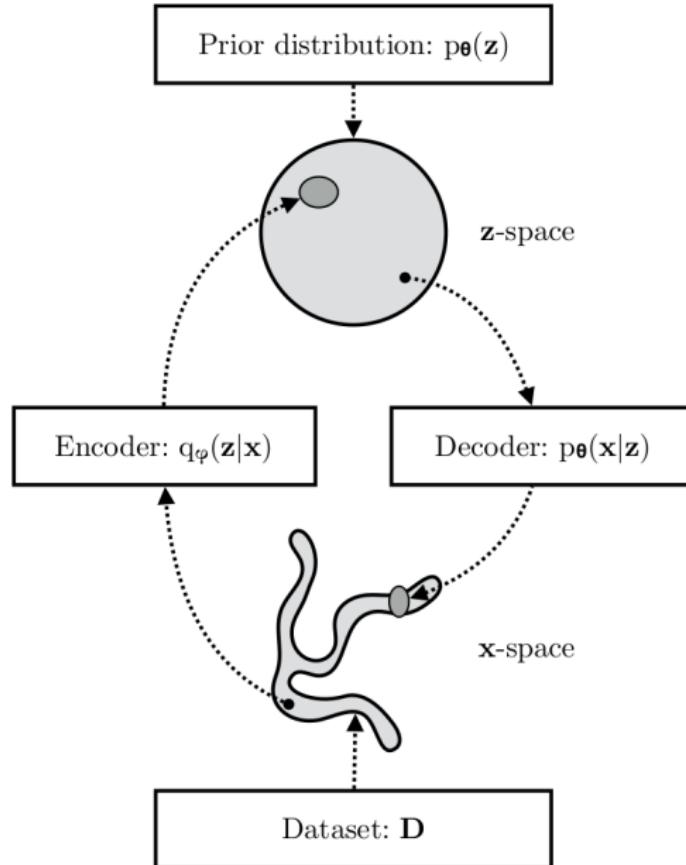
Variational autoencoder

- Voltamos a analisar o ELBO do VAE:

$$\begin{aligned}\mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \mathbb{E}_{q_{\phi}(z|x)} \left[\log \frac{p_{\theta}(\mathbf{x}, z)}{q_{\phi}(z|x)} \right] \\ &= \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(\mathbf{x}, z) - \log q_{\phi}(z|x)] \\ &= \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(\mathbf{x}|z) + \log p_{\theta}(z) - \log q_{\phi}(z|x)] \\ &= \underbrace{\mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(\mathbf{x}|z)]}_{\text{ajuste às observações}} - \underbrace{\text{KL}(q_{\phi}(z|x)||p_{\theta}(z))}_{\text{termo de regularização}}.\end{aligned}$$

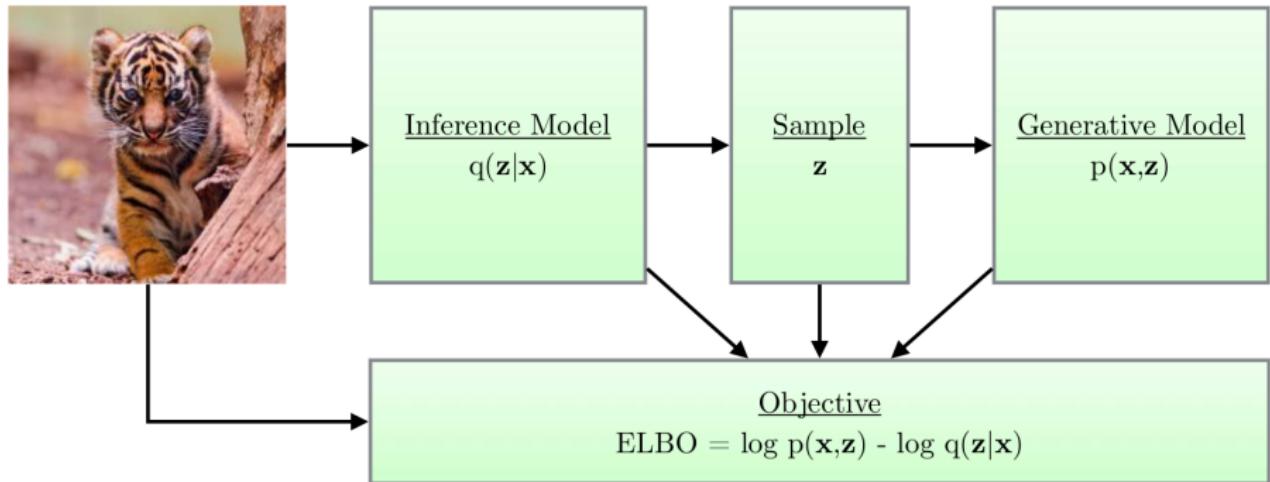
- A otimização do ELBO resulta em:
 - Um **modelo gerativo** $p_{\theta}(\mathbf{x}, z)$;
 - Formado por um *decoder* $p_{\theta}(\mathbf{x}|z) = \frac{p_{\theta}(\mathbf{x}, z)}{p_{\theta}(z)}$;
 - Um **modelo de inferência** (*encoder*) $q_{\phi}(z|x)$.
- Note que ambos os modelos são **treinados simultaneamente**.

Variational autoencoder

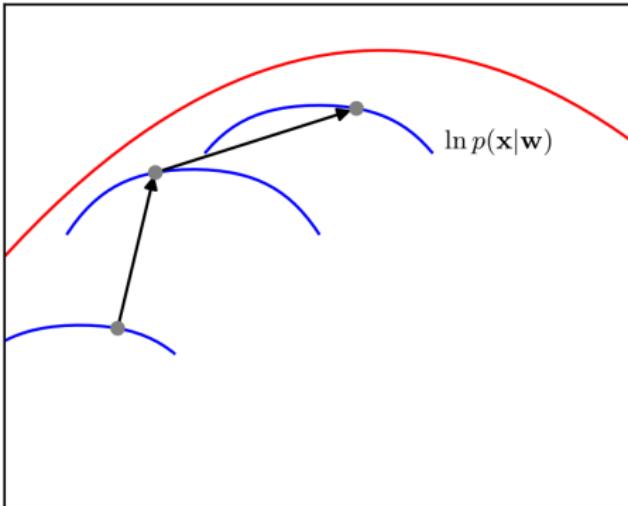
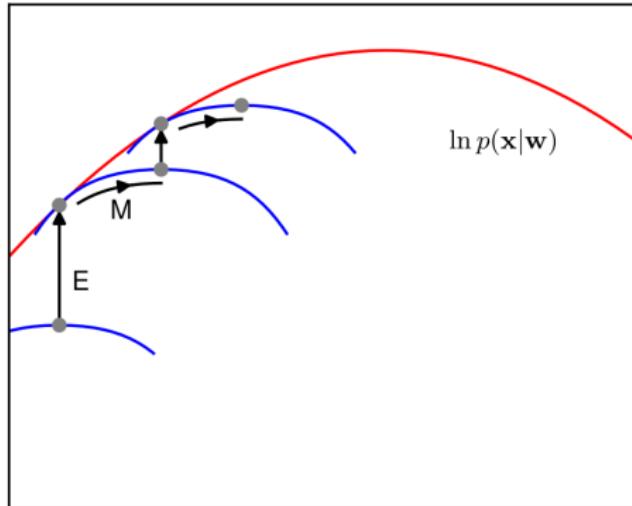


Variational autoencoder

Datapoint



Variational autoencoder



- No EM (à esquerda), os passos E (atualização da posteriori) e M (maximização dos parâmetros) são alternados. Quando o passo E é exato, o limiar encosta na log-verossimilhança verdadeira.
- No VAE (à direita), a posteriori (codificador) e os parâmetros do decodificador são conjuntamente otimizados, mas a atualização da posteriori não é exata.

Variational autoencoder

- Considerando um dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, o ELBO torna-se:

$$\mathcal{L}_{\theta, \phi}(\mathcal{D}) = \sum_{\mathbf{x} \in \mathcal{D}} \mathcal{L}_{\theta, \phi}(\mathbf{x}) = \sum_{\mathbf{x} \in \mathcal{D}} \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right].$$

- Os gradientes em relação a θ serão dados por:

$$\begin{aligned}\nabla_{\theta} \mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \nabla_{\theta} \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}, \mathbf{z}) - \log q_{\phi}(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\nabla_{\theta} (\log p_{\theta}(\mathbf{x}, \mathbf{z}) - \log q_{\phi}(\mathbf{z}|\mathbf{x}))] \\ &\simeq \frac{1}{S} \sum_{s=1}^S \nabla_{\theta} \log p_{\theta}(\mathbf{x}, \tilde{\mathbf{z}}^{(s)}),\end{aligned}$$

em que $\tilde{\mathbf{z}}^{(s)} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$ é uma amostra que resulta em uma **aproximação de Monte Carlo** não enviesada do gradiente.

- Posteriormente, usaremos $S = 1$ amostra de Monte Carlo.

Variational autoencoder

- Os gradientes em relação a ϕ não podem ser estimados da mesma maneira:

$$\begin{aligned}\nabla_{\phi} \mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \nabla_{\phi} \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(\mathbf{x}, z) - \log q_{\phi}(z|x)] \\ &\neq \mathbb{E}_{q_{\phi}(z|x)} [\nabla_{\phi} (\log p_{\theta}(\mathbf{x}, z) - \log q_{\phi}(z|x))],\end{aligned}$$

Variational autoencoder

- Os gradientes em relação a ϕ não podem ser estimados da mesma maneira:

$$\begin{aligned}\nabla_{\phi} \mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \nabla_{\phi} \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(\mathbf{x}, z) - \log q_{\phi}(z|x)] \\ &\neq \mathbb{E}_{q_{\phi}(z|x)} [\nabla_{\phi} (\log p_{\theta}(\mathbf{x}, z) - \log q_{\phi}(z|x))],\end{aligned}$$

pois a esperança em relação a $q_{\phi}(z|x)$ depende de ϕ .

- No caso de variáveis z contínuas, podemos usar o chamado **truque da reparametrização** (*reparameterization trick*) (Kingma e Welling, 2014; Rezende *et al.*, 2014).

Reparameterization trick

- Escrevemos a variável $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ como uma transformação determinística de outra variável aleatória ϵ :

$$\mathbf{z} = g(\epsilon, \phi, \mathbf{x}).$$

- A esperança de uma função $f(\cdot)$ qualquer em relação a $q_\phi(\mathbf{z}|\mathbf{x})$ pode ser reescrita como sendo em relação a $p(\epsilon)$:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})] = \mathbb{E}_{p(\epsilon)}[f(g(\epsilon, \phi, \mathbf{x}))].$$

Reparameterization trick

- Escrevemos a variável $\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$ como uma transformação determinística de outra variável aleatória ϵ :

$$\mathbf{z} = g(\epsilon, \phi, \mathbf{x}).$$

- A esperança de uma função $f(\cdot)$ qualquer em relação a $q_{\phi}(\mathbf{z}|\mathbf{x})$ pode ser reescrita como sendo em relação a $p(\epsilon)$:

$$\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})] = \mathbb{E}_{p(\epsilon)}[f(g(\epsilon, \phi, \mathbf{x}))].$$

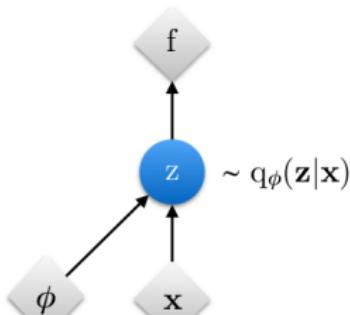
- Os gradientes em relação a ϕ passam a ser:

$$\begin{aligned}\nabla_{\phi} \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})] &= \nabla_{\phi} \mathbb{E}_{p(\epsilon)}[f(g(\epsilon, \phi, \mathbf{x}))] \\ &= \mathbb{E}_{p(\epsilon)}[\nabla_{\phi} f(g(\epsilon, \phi, \mathbf{x}))] \\ &\simeq \frac{1}{S} \sum_{s=1}^S \nabla_{\phi} f(g(\tilde{\epsilon}^{(s)}, \phi, \mathbf{x})),\end{aligned}$$

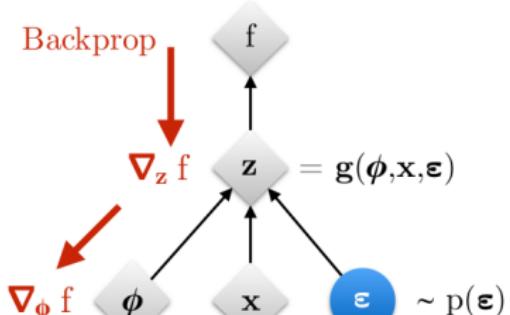
em que $\tilde{\epsilon}^{(s)} \sim p(\epsilon)$ é uma amostra estocástica e o estimador resultante é não enviesado.

Reparameterization trick

Original form



Reparameterized form



: Deterministic node

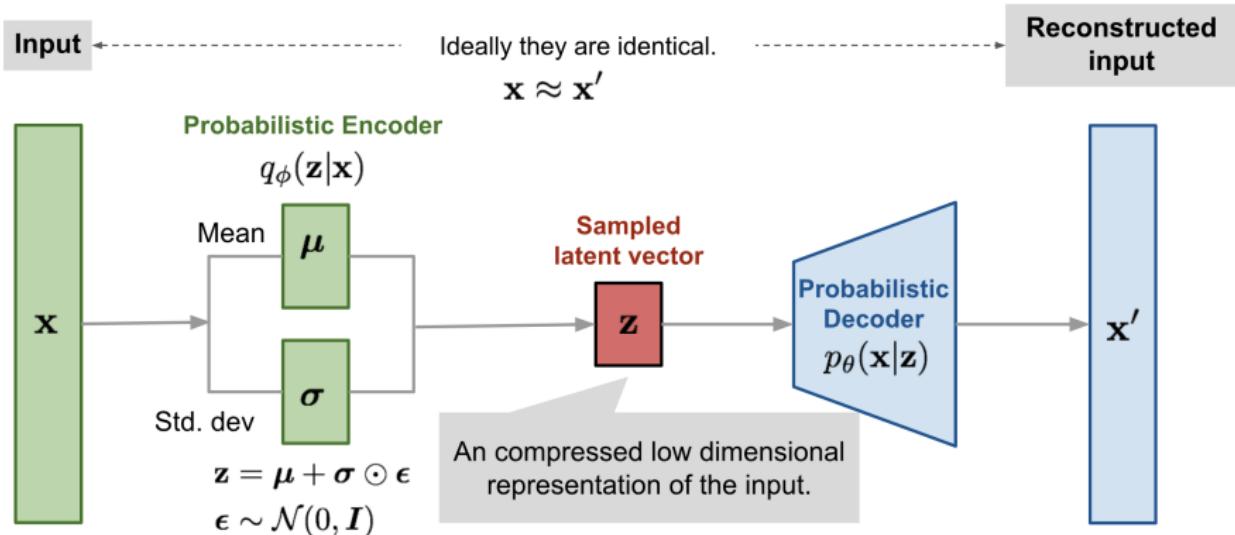
→ : Evaluation of f



: Random node

→ : Differentiation of f

Variational autoencoder



Variational autoencoder

- A partir do truque da reparametrização, reescrevemos o ELBO:

$$\begin{aligned}\mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(\mathbf{x}|z)] - \text{KL}(q_{\phi}(z|x) || p_{\theta}(z)) \\ &= \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(\mathbf{x}, z) - \log q_{\phi}(z|x)] \\ &= \mathbb{E}_{p(\epsilon)} [\log p_{\theta}(\mathbf{x}, z) - \log q_{\phi}(z|x)],\end{aligned}$$

em que $z = g(\epsilon, \phi, \mathbf{x})$.

Variational autoencoder

- A partir do truque da reparametrização, reescrevemos o ELBO:

$$\begin{aligned}\mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] - \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\theta}(\mathbf{z})) \\ &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}, \mathbf{z}) - \log q_{\phi}(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{p(\epsilon)} [\log p_{\theta}(\mathbf{x}, \mathbf{z}) - \log q_{\phi}(\mathbf{z}|\mathbf{x})],\end{aligned}$$

em que $\mathbf{z} = g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \mathbf{x})$.

- Caso o termo KL seja analítico, o ELBO pode ser estimado por:

$$\begin{aligned}\tilde{\boldsymbol{\epsilon}}^{(s)} &\sim p(\boldsymbol{\epsilon}), \quad \tilde{\mathbf{z}}^{(s)} = g(\boldsymbol{\epsilon}^{(s)}, \boldsymbol{\phi}, \mathbf{x}), \\ \tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) &= \frac{1}{S} \sum_{s=1}^S [\log p_{\theta}(\mathbf{x}|\tilde{\mathbf{z}}^{(s)})] - \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\theta}(\mathbf{z})).\end{aligned}$$

Variational autoencoder

- A partir do truque da reparametrização, reescrevemos o ELBO:

$$\begin{aligned}\mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] - \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\theta}(\mathbf{z})) \\ &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}, \mathbf{z}) - \log q_{\phi}(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{p(\epsilon)} [\log p_{\theta}(\mathbf{x}, \mathbf{z}) - \log q_{\phi}(\mathbf{z}|\mathbf{x})],\end{aligned}$$

em que $\mathbf{z} = g(\epsilon, \phi, \mathbf{x})$.

- Caso o termo KL seja analítico, o ELBO pode ser estimado por:

$$\begin{aligned}\tilde{\epsilon}^{(s)} &\sim p(\epsilon), \quad \tilde{\mathbf{z}}^{(s)} = g(\epsilon^{(s)}, \phi, \mathbf{x}), \\ \tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) &= \frac{1}{S} \sum_{s=1}^S [\log p_{\theta}(\mathbf{x}|\tilde{\mathbf{z}}^{(s)})] - \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\theta}(\mathbf{z})).\end{aligned}$$

- Se o KL for custoso ou não analítico, usamos a estimação:

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) = \frac{1}{S} \sum_{s=1}^S [\log p_{\theta}(\mathbf{x}, \tilde{\mathbf{z}}^{(s)}) - \log q_{\phi}(\tilde{\mathbf{z}}^{(s)}|\mathbf{x})].$$

Variational autoencoder

- Considerando um minibatch \mathcal{M} com M amostras de \mathcal{D} , o ELBO do conjunto de dados pode ser aproximado por:

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathcal{D}) = \frac{N}{M} \sum_{x \in \mathcal{M}} \tilde{\mathcal{L}}_{\theta, \phi}(x).$$

Variational autoencoder

- Considerando um minibatch \mathcal{M} com M amostras de \mathcal{D} , o ELBO do conjunto de dados pode ser aproximado por:

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathcal{D}) = \frac{N}{M} \sum_{x \in \mathcal{M}} \tilde{\mathcal{L}}_{\theta, \phi}(x).$$

- A estimativa do gradiente $\nabla_{\theta, \phi} \tilde{\mathcal{L}}_{\theta, \phi}(x)$ pode ser usado para otimizar o modelo via SGD com minibatches, sendo chamado de **Auto-Encoding Variational Bayes (AEVB)**.

Variational autoencoder

- Considerando um minibatch \mathcal{M} com M amostras de \mathcal{D} , o ELBO do conjunto de dados pode ser aproximado por:

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathcal{D}) = \frac{N}{M} \sum_{x \in \mathcal{M}} \tilde{\mathcal{L}}_{\theta, \phi}(x).$$

- A estimação do gradiente $\nabla_{\theta, \phi} \tilde{\mathcal{L}}_{\theta, \phi}(x)$ pode ser usado para otimizar o modelo via SGD com minibatches, sendo chamado de **Auto-Encoding Variational Bayes (AEVB)**.
- O estimador dos gradientes do ELBO via truque de reparametrização é chamado de **Stochastic Gradient Variational Bayes (SGVB)**.

Variational autoencoder

- Voltamos a considerar uma distribuição variacional fatorada:

$$q_{\phi}(z|x) = \mathcal{N}(z|\mu, \text{diag}(\sigma^2)).$$

Variational autoencoder

- Voltamos a considerar uma distribuição variacional fatorada:

$$q_{\phi}(z|x) = \mathcal{N}(z|\mu, \text{diag}(\sigma^2)).$$

- O truque da reparametrização nos permite reescrever:

$$\begin{aligned}\epsilon &\sim \mathcal{N}(\mathbf{0}, I), \\ (\mu, \log \sigma) &= \text{EncoderDNN}_{\phi}(x), \\ z &= g(\epsilon, \phi, x) = \mu + \sigma \odot \epsilon,\end{aligned}$$

- \odot denota o produto de Hadamard (ponto a ponto).
- Como $z|x \sim \mathcal{N}(\mu, \text{diag}(\sigma))$, temos $g(\epsilon, \phi, x) = \mu + \sigma \odot \epsilon$.

Variational autoencoder

- Dessa maneira, o ELBO pode ser estimado por:

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}) = \text{EncoderDNN}_{\phi}(\mathbf{x}),$$

$$\tilde{\boldsymbol{\epsilon}}^{(s)} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \mathbf{I}), \quad \tilde{\mathbf{z}}^{(s)} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \tilde{\boldsymbol{\epsilon}}^{(s)},$$

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) = \frac{1}{S} \sum_{s=1}^S \log p_{\theta}(\mathbf{x} | \tilde{\mathbf{z}}^{(s)}) - \text{KL}(\mathcal{N}(\mathbf{z} | \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2)) || \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I})).$$

Variational autoencoder

- Dessa maneira, o ELBO pode ser estimado por:

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}) = \text{EncoderDNN}_{\phi}(\mathbf{x}),$$

$$\tilde{\boldsymbol{\epsilon}}^{(s)} \sim \mathcal{N}(\boldsymbol{\epsilon} | \mathbf{0}, \mathbf{I}), \quad \tilde{\mathbf{z}}^{(s)} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \tilde{\boldsymbol{\epsilon}}^{(s)},$$

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) = \frac{1}{S} \sum_{s=1}^S \log p_{\theta}(\mathbf{x} | \tilde{\mathbf{z}}^{(s)}) - \text{KL}(\mathcal{N}(\mathbf{z} | \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2)) || \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I})).$$

- A divergência KL entre Gaussianas é dada por:

$$\text{KL}(\mathcal{N}_0(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) || \mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)) =$$

$$\frac{1}{2} \left[\text{Tr}(\boldsymbol{\Sigma}_1^{-1} \boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\top} \boldsymbol{\Sigma}_1 (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - K + \log \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} \right].$$

Variational autoencoder

- Dessa maneira, o ELBO pode ser estimado por:

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}) = \text{EncoderDNN}_{\phi}(\mathbf{x}),$$

$$\tilde{\boldsymbol{\epsilon}}^{(s)} \sim \mathcal{N}(\boldsymbol{\epsilon} | \mathbf{0}, \mathbf{I}), \quad \tilde{\mathbf{z}}^{(s)} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \tilde{\boldsymbol{\epsilon}}^{(s)},$$

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) = \frac{1}{S} \sum_{s=1}^S \log p_{\theta}(\mathbf{x} | \tilde{\mathbf{z}}^{(s)}) - \text{KL}(\mathcal{N}(\mathbf{z} | \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2)) || \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I})).$$

- A divergência KL entre Gaussianas é dada por:

$$\text{KL}(\mathcal{N}_0(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) || \mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)) =$$

$$\frac{1}{2} \left[\text{Tr}(\boldsymbol{\Sigma}_1^{-1} \boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\top} \boldsymbol{\Sigma}_1 (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - K + \log \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} \right].$$

- Assim:

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) = \frac{1}{S} \sum_{s=1}^S \log p_{\theta}(\mathbf{x} | \tilde{\mathbf{z}}^{(s)}) - \frac{1}{2} \sum_k [\mu_k^2 + \sigma_k^2 - 1 - 2 \log \sigma_k].$$

Variational autoencoder

- Poderíamos ter escolhido um formato não fatorado para a distribuição variacional:

$$q_{\phi}(z|x) = \mathcal{N}(z|\mu, \Sigma).$$

Variational autoencoder

- Poderíamos ter escolhido um formato não fatorado para a distribuição variacional:

$$q_{\phi}(z|x) = \mathcal{N}(z|\mu, \Sigma).$$

- Nesse caso, o truque de reparametrização poderia ser:

$$\epsilon \sim \mathcal{N}(\mathbf{0}, I),$$

$$z = \mu + L\epsilon,$$

em que L é uma matriz triangular inferior e $\Sigma = LL^T$, ou seja, L corresponde à decomposição de Cholesky de Σ .

Variational autoencoder

- Assim, o ELBO pode ser estimado por:

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}, \mathbf{L}') = \text{EncoderDNN}_{\phi}(\mathbf{x}),$$

$$\mathbf{L} = \mathbf{L}' + \text{diag}(\boldsymbol{\sigma}), \quad \tilde{\boldsymbol{\epsilon}}^{(s)} \sim \mathcal{N}(\boldsymbol{\epsilon} | \mathbf{0}, \mathbf{I}), \quad \tilde{\mathbf{z}}^{(s)} = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{\epsilon}^{(s)},$$

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) = \frac{1}{S} \sum_{s=1}^S \log p_{\theta}(\mathbf{x} | \tilde{\mathbf{z}}^{(s)}) - \text{KL}(\mathcal{N}(z | \boldsymbol{\mu}, \boldsymbol{\Sigma}) || \mathcal{N}(z | \mathbf{0}, \mathbf{I})),$$

em que \mathbf{L}' é uma matriz triangular com zeros na diagonal.

Variational autoencoder

- Assim, o ELBO pode ser estimado por:

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}, \mathbf{L}') = \text{EncoderDNN}_{\phi}(\mathbf{x}),$$

$$\mathbf{L} = \mathbf{L}' + \text{diag}(\boldsymbol{\sigma}), \quad \tilde{\boldsymbol{\epsilon}}^{(s)} \sim \mathcal{N}(\boldsymbol{\epsilon} | \mathbf{0}, \mathbf{I}), \quad \tilde{\mathbf{z}}^{(s)} = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{\epsilon}^{(s)},$$

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) = \frac{1}{S} \sum_{s=1}^S \log p_{\theta}(\mathbf{x} | \tilde{\mathbf{z}}^{(s)}) - \text{KL}(\mathcal{N}(z | \boldsymbol{\mu}, \boldsymbol{\Sigma}) || \mathcal{N}(z | \mathbf{0}, \mathbf{I})),$$

em que \mathbf{L}' é uma matriz triangular com zeros na diagonal.

- Como o KL entre Gaussianas é analítico, temos:

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) = \frac{1}{S} \sum_{s=1}^S \log p_{\theta}(\mathbf{x} | \tilde{\mathbf{z}}^{(s)}) - \frac{1}{2} \sum_k [\mu_k^2 + \sigma_k^2 - 1 - 2 \log \sigma_k].$$

- Note que o último termo ficou idêntico ao caso da posterior fatorada.

Variational autoencoder

- Se o KL for custoso ou não analítico, usamos a estimação:

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) = \frac{1}{S} \sum_{s=1}^S \left[\underbrace{\log p_{\theta}(\mathbf{x}, \tilde{\mathbf{z}}^{(s)})}_{\log p_{\theta}(\mathbf{x}|\tilde{\mathbf{z}}^{(s)}) + \log p_{\theta}(\tilde{\mathbf{z}}^{(s)})} - \log q_{\phi}(\tilde{\mathbf{z}}^{(s)}|\mathbf{x}) \right].$$

- Para calcular o termo $\log q_{\phi}(\mathbf{z}|\mathbf{x})$ do estimador do ELBO, aplicamos a regra de transformação de uma variável aleatória:

$$q_{\phi}(\mathbf{z}|\mathbf{x}) = \left| \det \left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\epsilon}} \right) \right|^{-1} p(\boldsymbol{\epsilon}),$$

$$\log q_{\phi}(\mathbf{z}|\mathbf{x}) = \log p(\boldsymbol{\epsilon}) - \log \left| \det \left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\epsilon}} \right) \right|, \text{ em que } \frac{\partial \mathbf{z}}{\partial \boldsymbol{\epsilon}} = \frac{\partial g(\boldsymbol{\epsilon}, \phi, \mathbf{x})}{\partial \boldsymbol{\epsilon}}.$$

- A matriz Jacobiana é definida por:

$$\frac{\partial \mathbf{z}}{\partial \boldsymbol{\epsilon}} = \frac{\partial(z_1, \dots, z_K)}{\partial(\epsilon_1, \dots, \epsilon_K)} = \begin{bmatrix} \frac{\partial z_1}{\partial \epsilon_1} & \cdots & \frac{\partial z_1}{\partial \epsilon_K} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_K}{\partial \epsilon_1} & \cdots & \frac{\partial z_K}{\partial \epsilon_K} \end{bmatrix}.$$

Variational autoencoder

- Quando $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}))$, temos:

$$\mathbf{z} = g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \mathbf{x}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}.$$

→ A Jacobiana é simples:

$$\frac{\partial \mathbf{z}}{\partial \boldsymbol{\epsilon}} = \text{diag}(\boldsymbol{\sigma}) \text{ e } \log |\det\left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\epsilon}}\right)| = \sum_k \log \sigma_k.$$

Variational autoencoder

- Quando $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}))$, temos:

$$\mathbf{z} = g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \mathbf{x}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}.$$

→ A Jacobiana é simples:

$$\frac{\partial \mathbf{z}}{\partial \boldsymbol{\epsilon}} = \text{diag}(\boldsymbol{\sigma}) \text{ e } \log |\det\left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\epsilon}}\right)| = \sum_k \log \sigma_k.$$

- Quando $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, temos

$$\mathbf{z} = g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \mathbf{x}) = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{\epsilon},$$

em \mathbf{L} é a decomposição de Cholesky de $\boldsymbol{\Sigma}$, i.e., $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top$.

→ A Jacobiana continua simples:

$$\frac{\partial \mathbf{z}}{\partial \boldsymbol{\epsilon}} = \mathbf{L} \text{ e } \log |\det\left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\epsilon}}\right)| = \sum_k \log L_{kk}.$$

Variational autoencoder

- Quando $q_\phi(z|x) = \mathcal{N}(z|\mu, \text{diag}(\sigma))$, temos:

$$z = g(\epsilon, \phi, x) = \mu + \sigma \odot \epsilon.$$

→ A Jacobiana é simples:

$$\frac{\partial z}{\partial \epsilon} = \text{diag}(\sigma) \text{ e } \log |\det\left(\frac{\partial z}{\partial \epsilon}\right)| = \sum_k \log \sigma_k.$$

- Quando $q_\phi(z|x) = \mathcal{N}(z|\mu, \Sigma)$, temos

$$z = g(\epsilon, \phi, x) = \mu + L\epsilon,$$

em L é a decomposição de Cholesky de Σ , i.e., $\Sigma = LL^\top$.

→ A Jacobiana continua simples:

$$\frac{\partial z}{\partial \epsilon} = L \text{ e } \log |\det\left(\frac{\partial z}{\partial \epsilon}\right)| = \sum_k \log L_{kk}.$$

- Uma sequência de transformações $z = \mu + L\epsilon$ tornaria $q_\phi(z|x)$ mais flexível.
- Se as Jacobianas forem triangulares, o termo log-determinante continuará fácil de calcular.
- **Normalizing Flows** explora essa ideia nas próximas aulas!

Variational autoencoder

Resumo do algoritmo

- ① Colete $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ e escolha uma verossimilhança $p_{\theta}(\mathbf{x}|\mathbf{z})$;
- ② Inicialize $\boldsymbol{\theta}$ e $\boldsymbol{\phi}$;
- ③ Repita até convergir ou por um número máximo de iterações:
 - ① Crie um minibatch \mathcal{M} a partir de M amostras de \mathcal{D} ;
 - ② Calcule os termos do ELBO para $\mathbf{x} \in \mathcal{M}$ (considerando $S = 1$):

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}, \mathbf{L}') = \text{EncoderDNN}_{\phi}(\mathbf{x}),$$

$$\mathbf{L} = \mathbf{L}' + \text{diag}(\boldsymbol{\sigma}), \quad \tilde{\boldsymbol{\epsilon}} \sim \mathcal{N}(\boldsymbol{\epsilon} | \mathbf{0}, \mathbf{I}), \quad \tilde{\mathbf{z}} = \boldsymbol{\mu} + \mathbf{L}\tilde{\boldsymbol{\epsilon}},$$

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}) = \frac{N}{M} \sum_{\mathbf{x} \in \mathcal{M}} \left[\log p_{\theta}(\mathbf{x}|\tilde{\mathbf{z}}) - \frac{1}{2} \sum_k [\mu_k^2 + \sigma_k^2 - 1 - 2 \log \sigma_k] \right].$$

- ③ Atualize $\boldsymbol{\theta}$ e $\boldsymbol{\phi}$ via SGD com os gradientes $\nabla_{\theta, \phi} \tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x})$;
- ④ Retorne os parâmetros otimizados $\hat{\boldsymbol{\theta}}$ e $\hat{\boldsymbol{\phi}}$.

Variational autoencoder

Resumo do algoritmo

- ① A verossimilhança marginal de um padrão \mathbf{x} é estimada por:

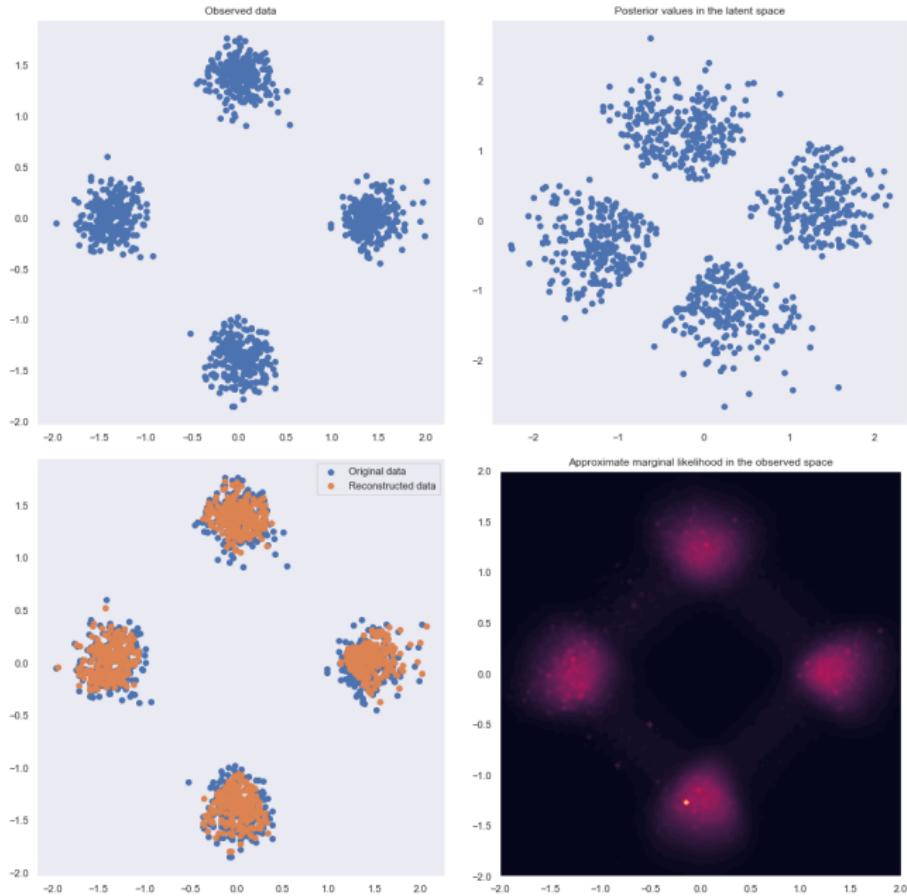
$$p_{\theta}(\mathbf{x}) = \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{p_{\theta}(\mathbf{z}|\mathbf{x})}$$

$$\begin{aligned}\log p_{\theta}(\mathbf{x}) &= \log \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[\frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{p_{\theta}(\mathbf{z}|\mathbf{x})} \right] \\ &\approx \log \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[\frac{p_{\theta}(\mathbf{x}|\mathbf{z})p_{\theta}(\mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] \\ &\approx \log \frac{1}{S} \sum_{s=1}^S \frac{p_{\theta}(\mathbf{x}|\tilde{\mathbf{z}}^{(s)})p_{\theta}(\tilde{\mathbf{z}}^{(s)})}{q_{\phi}(\tilde{\mathbf{z}}^{(s)}|\mathbf{x})},\end{aligned}$$

em que $\tilde{\mathbf{z}}^{(s)} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$ é uma amostra da posteriori variacional.

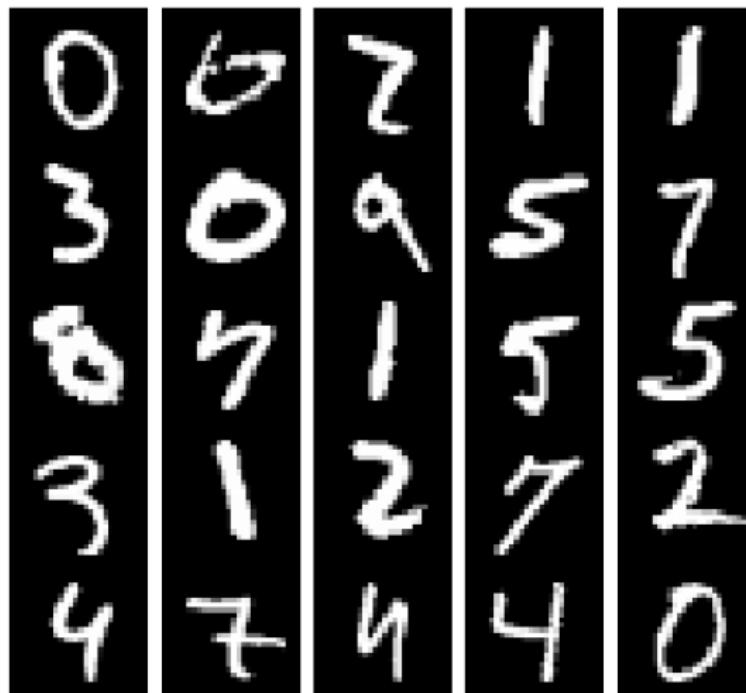
- **Observação:** O estimador acima é mais apropriado para espaços latentes de dimensão baixo ($K \leq 5$) e valores grandes de S .

VAE - 4 Gaussianas $N = 1000, D = 2, K = 2$

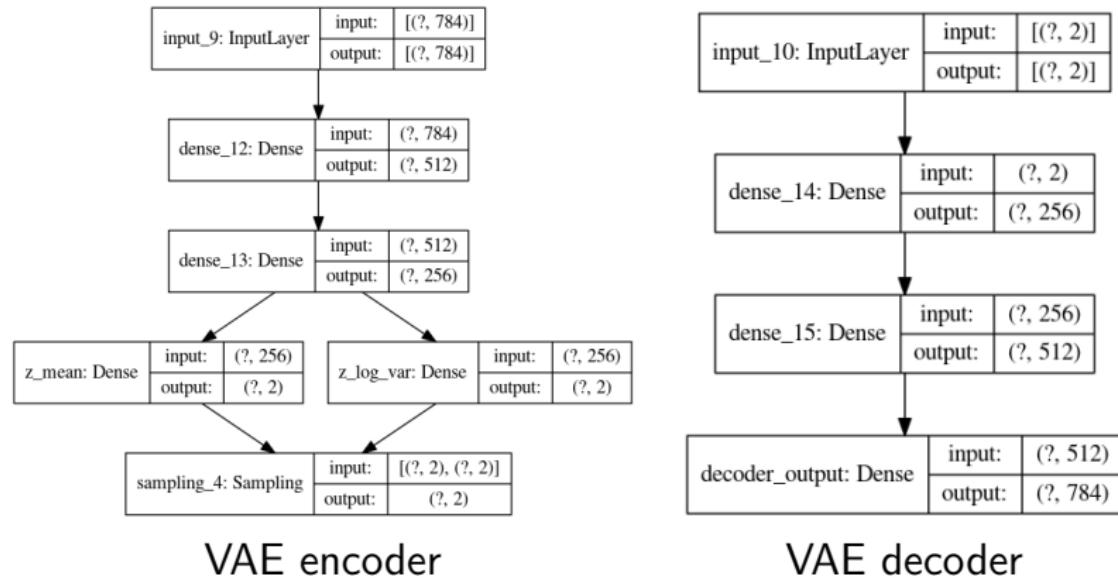


Dígitos MNIST $N = 70000, D = 784$

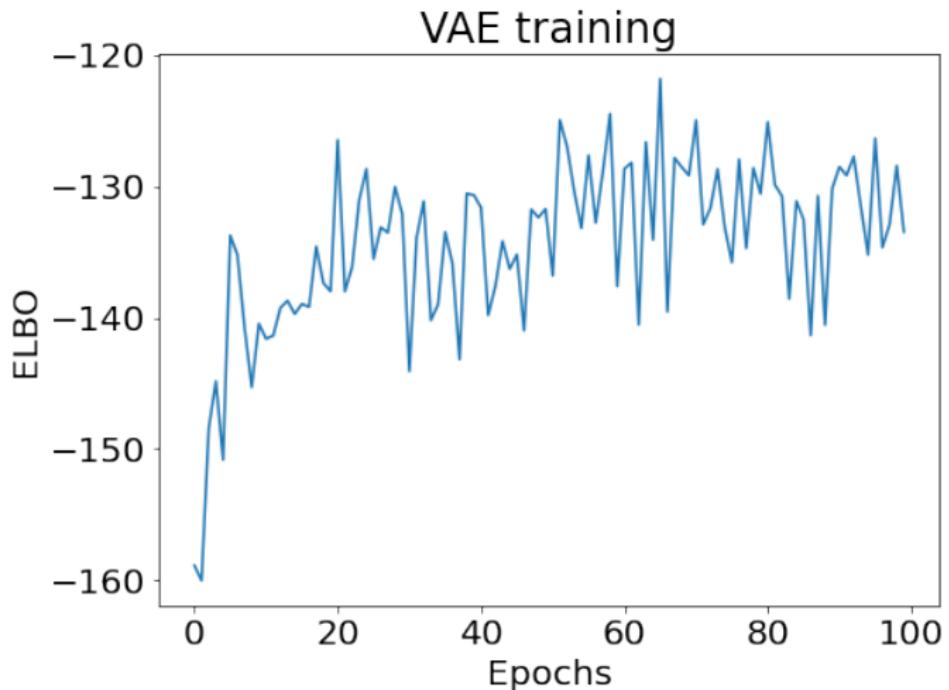
Original samples from all classes



VAE com RNAs - MNIST $N = 70000, D = 784, K = 2$

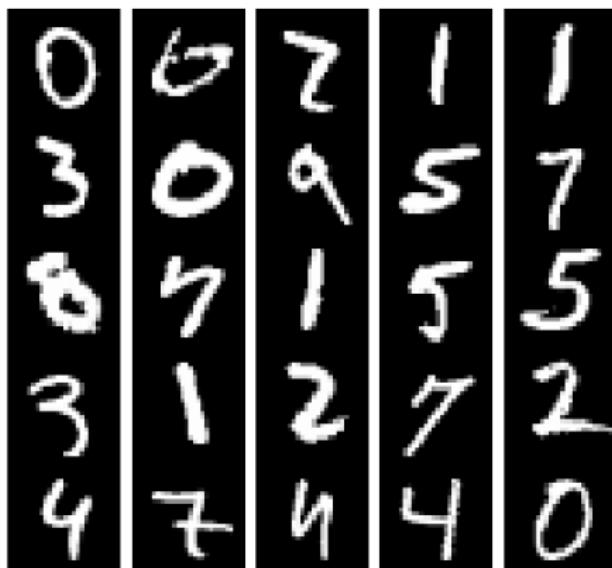


VAE com RNAs - MNIST $N = 70000, D = 784, K = 2$

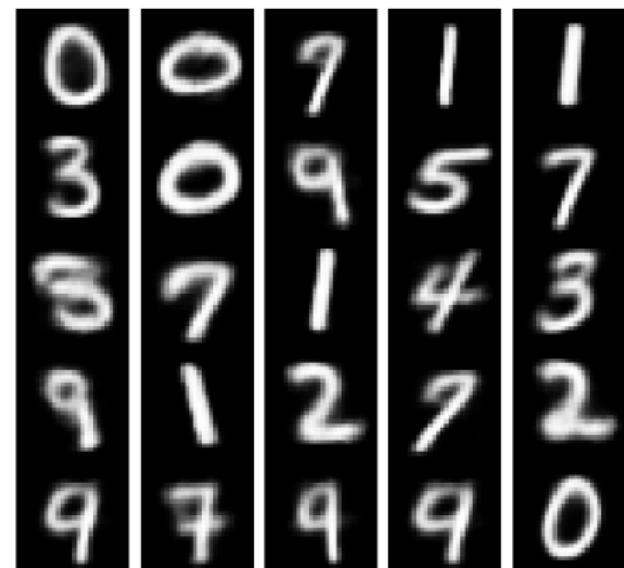


VAE - Reconstrução MNIST $N = 70000, D = 784, K = 2$

Original samples from all classes

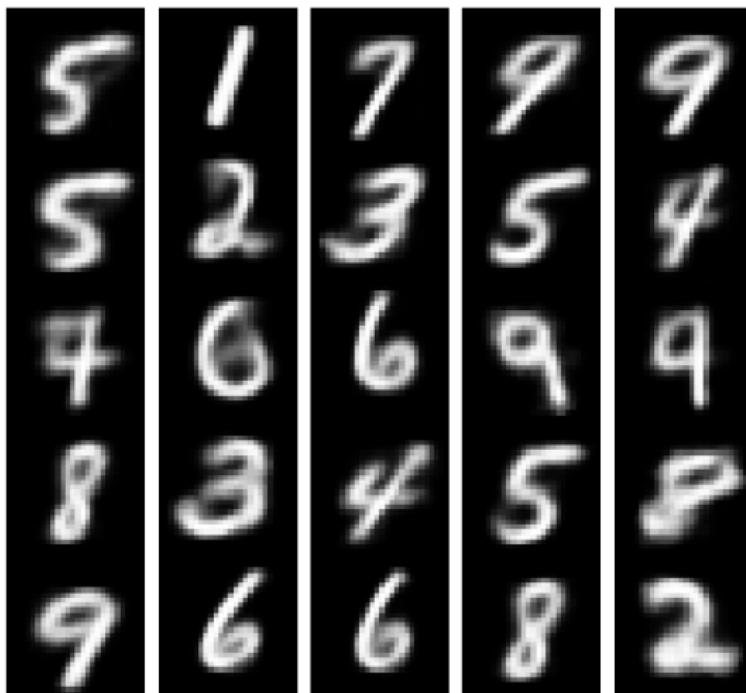


VAE - Reconstructions

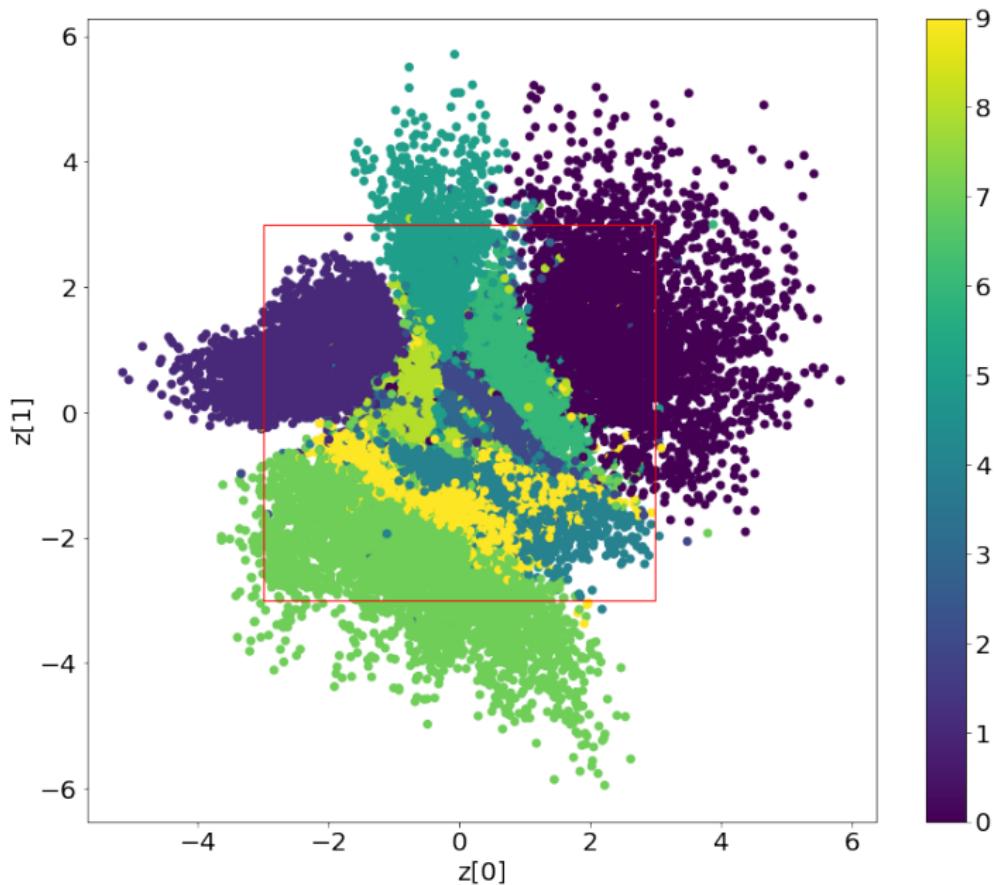


VAE - Geração MNIST $N = 70000, D = 784, K = 2$

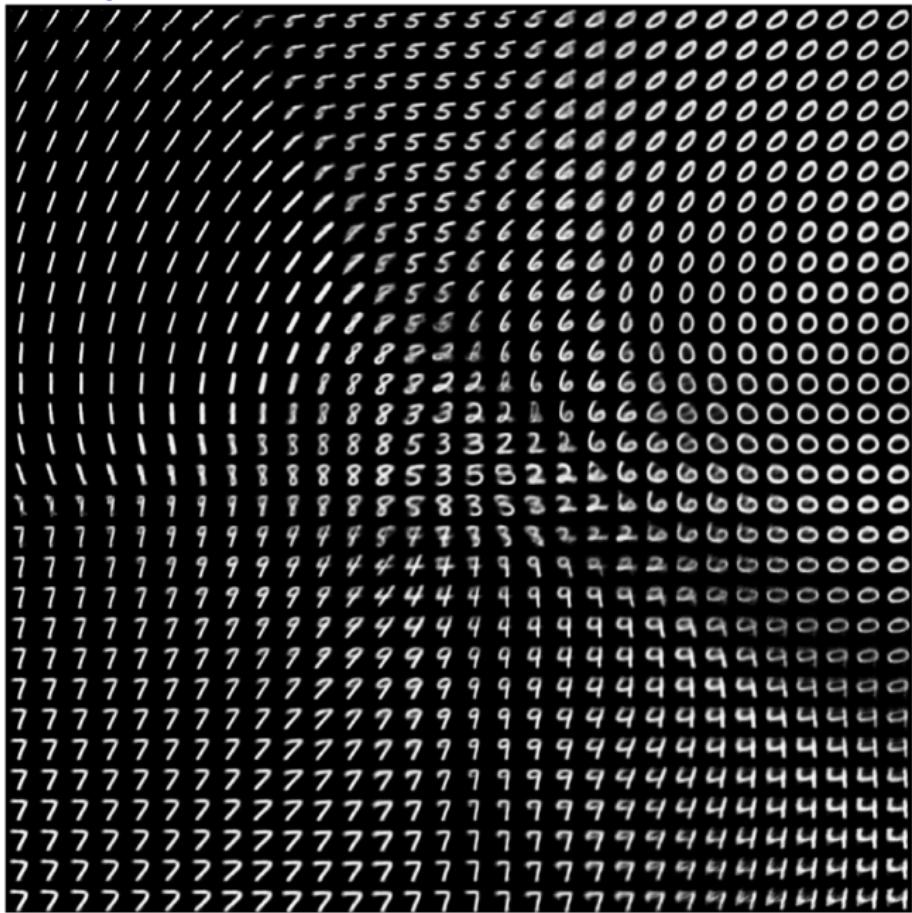
VAE - generated samples



VAE - Espaço latente MNIST $D = 784, K = 2$

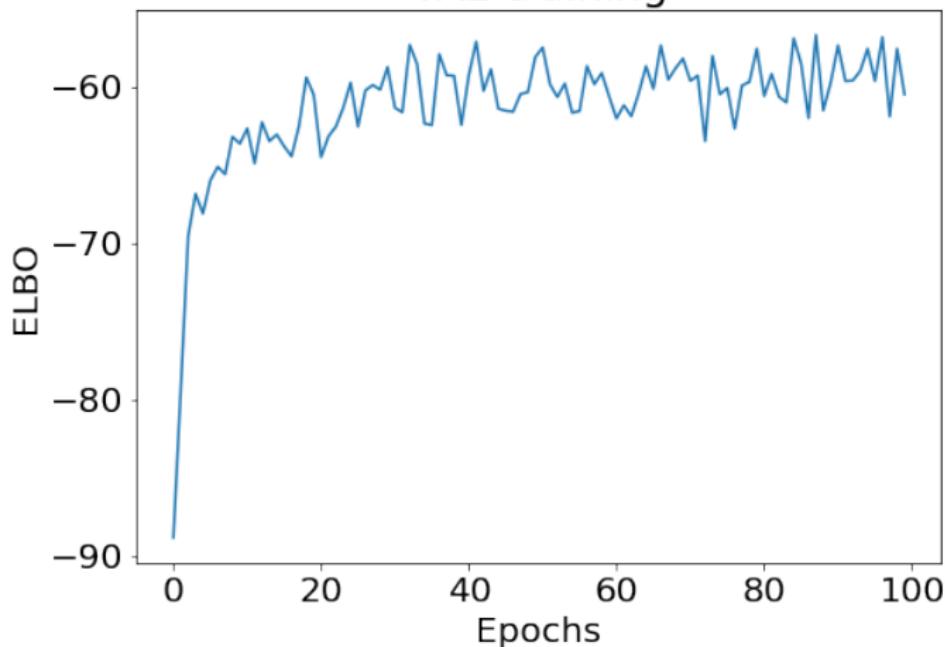


VAE - Geração MNIST $D = 784, K = 2$



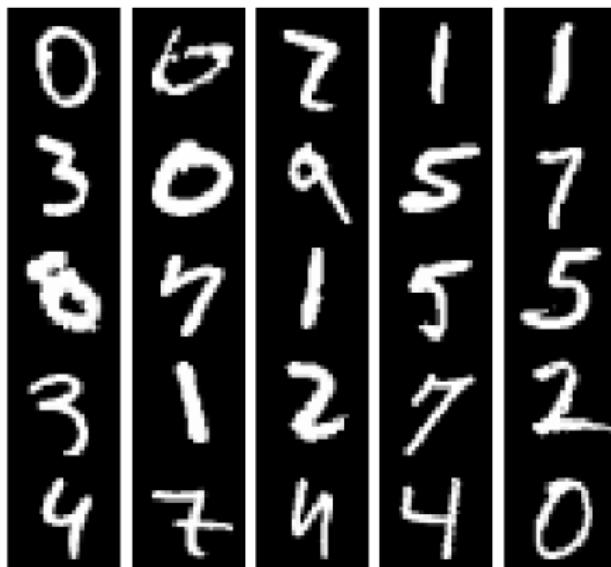
VAE com RNAs - MNIST $N = 70000, D = 784, K = 32$

VAE training

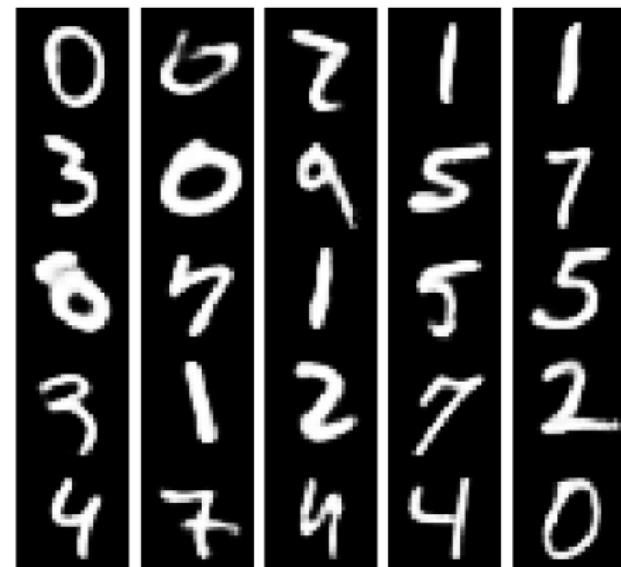


VAE - Reconstrução MNIST $N = 70000, D = 784, K = 32$

Original samples from all classes



VAE - Reconstructions



VAE - Interpolação MNIST $N = 70000, D = 784, K = 32$

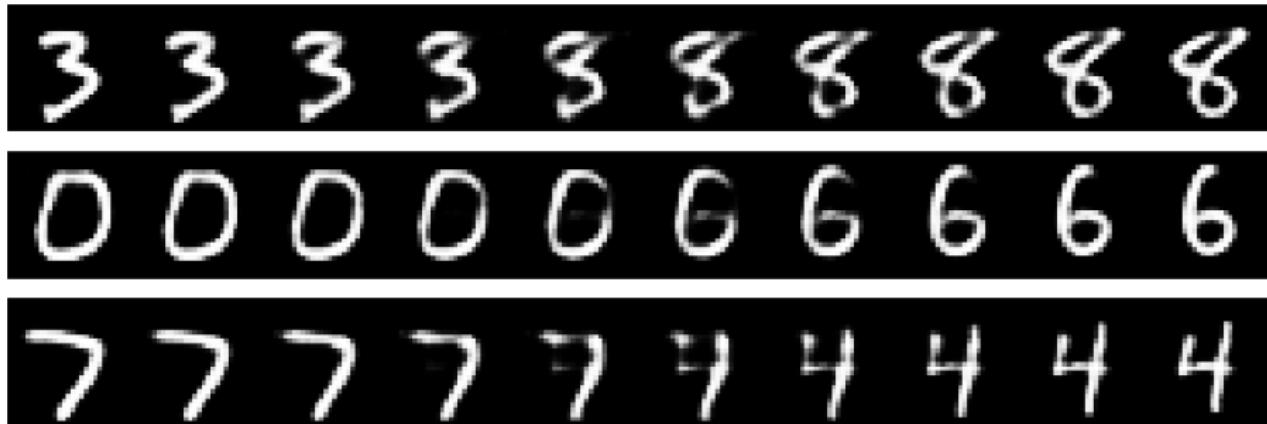
- Dados dois padrões x_1 e x_2 , podemos interpolar novas amostras:

$$z_1 = \text{encoder}(x_1), \quad z_2 = \text{encoder}(x_2),$$

$$z = (1 - \lambda)z_1 + \lambda z_2, \quad 0 \leq \lambda \leq 1,$$

$$x = \text{decoder}(z).$$

- Essa interpolação linear é válida por causa da baixa curvatura do *manifold* aprendido pelo VAE.



Agenda

- ① Modelos generativos
- ② Deep Latent Variable Model
- ③ Variational autoencoder
 - Reparameterization trick
 - Otimização do ELBO
- ④ Tópicos adicionais
- ⑤ Referências

Tópicos adicionais

- “Truques” na otimização do ELBO (e.g. *annealing* do termo KL).
- Escolha de distribuições a posteriori mais expressivas, como normalizing flows.
- Conditional VAE (CVAE):

$$\tilde{\mathcal{L}}_{\theta, \phi}(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y})} [\log p_{\theta}(\mathbf{y}|\mathbf{x}, \mathbf{z})] - \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}) || p_{\theta}(\mathbf{z})).$$

- β -VAE.
- VQ-VAE (Vector Quantised-Variational AutoEncoder).
- Abordagens com estrutura recorrente: Deep Recurrent Attentive Writer (DRAW), PixelVAE, etc.

Agenda

- ① Modelos generativos
- ② Deep Latent Variable Model
- ③ Variational autoencoder
 - Reparameterization trick
 - Otimização do ELBO
- ④ Tópicos adicionais
- ⑤ Referências

Referências bibliográficas

- KIGMA, Diederik P.; WELLING, Max. **An Introduction to Variational Autoencoders**, Foundations and Trends in Machine Learning, 2019.
- KIGMA, Diederik P.; WELLING, Max. **Auto-encoding variational Bayes**, ICLR, 2014.
- Cap. 20 - MURPHY, K. **Probabilistic Machine Learning: An Introduction**, 2020. Disponível em <https://github.com/probml/pml-book/releases/latest/download/book1.pdf>.
- WENG, Lilian. **From Autoencoder to Beta-VAE**. Disponível em <https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html>.