UNIVERSIDADE FEDERAL DO CEARÁ

DEPARTAMENTO DE COMPUTAÇÃO

Aprendizagem de Máquina Probabilística

César Lincoln Cavalcante Mattos

2024

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.
  - $\rightarrow$ Intenção de simular o processo em que os dados são gerados.
  - $\rightarrow$ 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 $\boldsymbol{x}$ cuja distribuição é aproximada por um modelo com parâmetros $\boldsymbol{\theta}$ (e.g. uma rede neural profunda):

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

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

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

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

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

Modelo com variáveis latentes

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

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}) \mathrm{d}\boldsymbol{z}.$$

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

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

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

Deep Latent Variable Model

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

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

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

Deep Latent Variable Model

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

$$
\begin{aligned}
p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I}), \quad \boldsymbol{z} \in \mathbb{R}^K, \\
\boldsymbol{p} &= \mathrm{DecoderDNN}_{\boldsymbol{\theta}}(\boldsymbol{z}), \quad \boldsymbol{p} \in \{0,1\}^D, \\
\log p(\boldsymbol{x}|\boldsymbol{z}) &= \sum_{d=1}^{D} \log p(x_d|\boldsymbol{z}) = \sum_{d=1}^{D} \log \mathrm{Bern}(x_d|p_d) \\
&= \sum_{d=1}^{D} \left[x_d \log p_d + (1 - x_d) \log(1 - p_d) \right].
\end{aligned}
$$

Deep Latent Variable Model

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

$$
\begin{aligned}
p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I}), \quad \boldsymbol{z} \in \mathbb{R}^K, \\
\boldsymbol{p} &= \mathrm{DecoderDNN}_{\boldsymbol{\theta}}(\boldsymbol{z}), \quad \boldsymbol{p} \in \{0,1\}^D, \\
\log p(\boldsymbol{x}|\boldsymbol{z}) &= \sum_{d=1}^{D} \log p(x_d|\boldsymbol{z}) = \sum_{d=1}^{D} \log \mathrm{Bern}(x_d|p_d) \\
&= \sum_{d=1}^{D} \left[x_d \log p_d + (1 - x_d) \log(1 - p_d)\right].
\end{aligned}
$$

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

Deep Latent Variable Model

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

$$
\begin{aligned}
p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I}), \quad \boldsymbol{z} \in \mathbb{R}^K, \\
\boldsymbol{p} &= \mathrm{DecoderDNN}_{\boldsymbol{\theta}}(\boldsymbol{z}), \quad \boldsymbol{p} \in \{0,1\}^D, \\
\log p(\boldsymbol{x}|\boldsymbol{z}) &= \sum_{d=1}^{D} \log p(x_d|\boldsymbol{z}) = \sum_{d=1}^{D} \log \mathrm{Bern}(x_d|p_d) \\
&= \sum_{d=1}^{D} \left[x_d \log p_d + (1 - x_d) \log(1 - p_d)\right].
\end{aligned}
$$

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

Deep Latent Variable Model

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

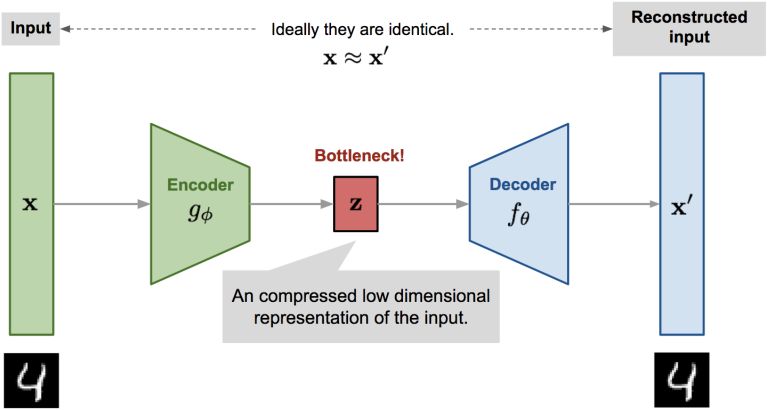
$$
\begin{aligned}
p(\boldsymbol{z}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I}), \quad \boldsymbol{z} \in \mathbb{R}^K, \\
\boldsymbol{p} &= \text{DecoderDNN}_{\boldsymbol{\theta}}(\boldsymbol{z}), \quad \boldsymbol{p} \in \{0,1\}^D, \\
\log p(\boldsymbol{x}|\boldsymbol{z}) &= \sum_{d=1}^{D} \log p(x_d|\boldsymbol{z}) = \sum_{d=1}^{D} \log \text{Bern}(x_d|p_d) \\
&= \sum_{d=1}^{D} \left[x_d \log p_d + (1 - x_d) \log(1 - p_d)\right].
\end{aligned}
$$

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

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

em que $q_{\boldsymbol{\phi}}$ é um **modelo de inferência** e $\boldsymbol{\phi}$ são parâmetros variacionais.

Variational autoencoder

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

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

  em que $q_{\boldsymbol{\phi}}$ é um **modelo de inferência** e $\boldsymbol{\phi}$ são parâmetros variacionais.
- Seja uma distribuição variacional Gaussiana com momentos calculados por redes neurais profundas:

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

- A mesma rede aproxima a posteriori de todos os dados $\boldsymbol{x} \in \mathcal{D}$.
- 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:

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

Variational autoencoder

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

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

Variational autoencoder

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

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

- O termo KL é sempre não negativo:

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

  sendo igual a zero somente para distribuições iguais.
- O ELBO $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x})$ constitui um **limiar inferior** para a evidência:

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

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

Variational autoencoder

- Voltamos a analisar o ELBO do VAE:

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

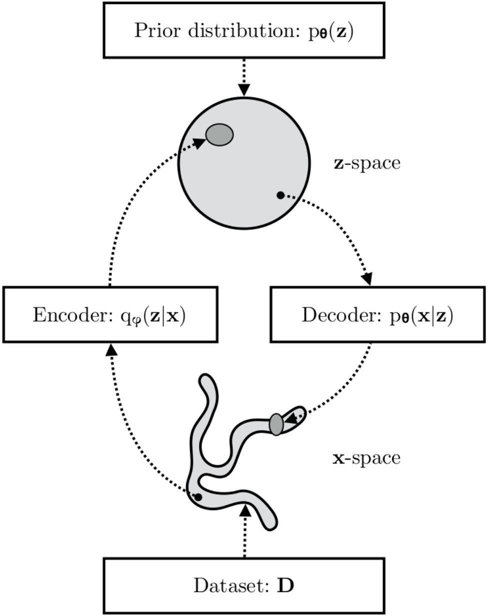
Variational autoencoder

- Voltamos a analisar o ELBO do VAE:

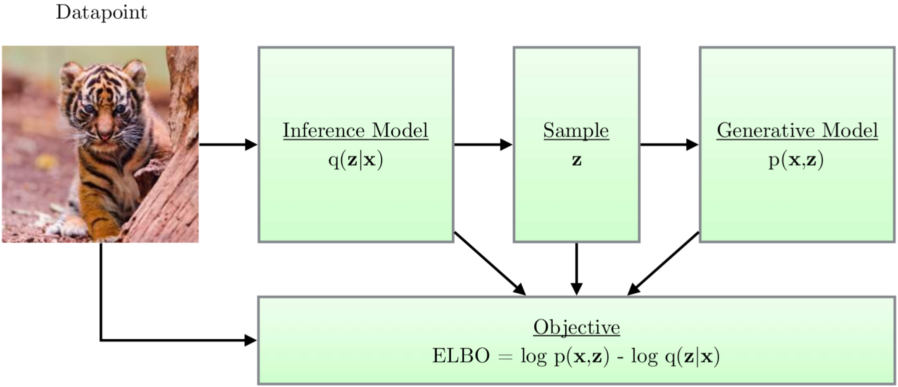
$$
\begin{aligned}
\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\lef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

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

Variational autoencoder



Variational autoencoder



Variational autoencoder

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

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

- Os gradientes em relação a $\boldsymbol{\theta}$ serão dados por:

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

  em que $\tilde{\boldsymbol{z}}^{(s)} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ 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:

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

Variational autoencoder

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

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

  pois a esperança em relação a $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ depende de $\boldsymbol{\phi}$.
- No caso de variáveis $\boldsymbol{z}$ contínuas, podemos usar o chamado **truque da reparametrização** (*reparameterization trick*) (Kingma e Welling, 2014; Rezende *et al.*, 2014).

Reparameterization trick

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

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

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

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

Reparameterization trick

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

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

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

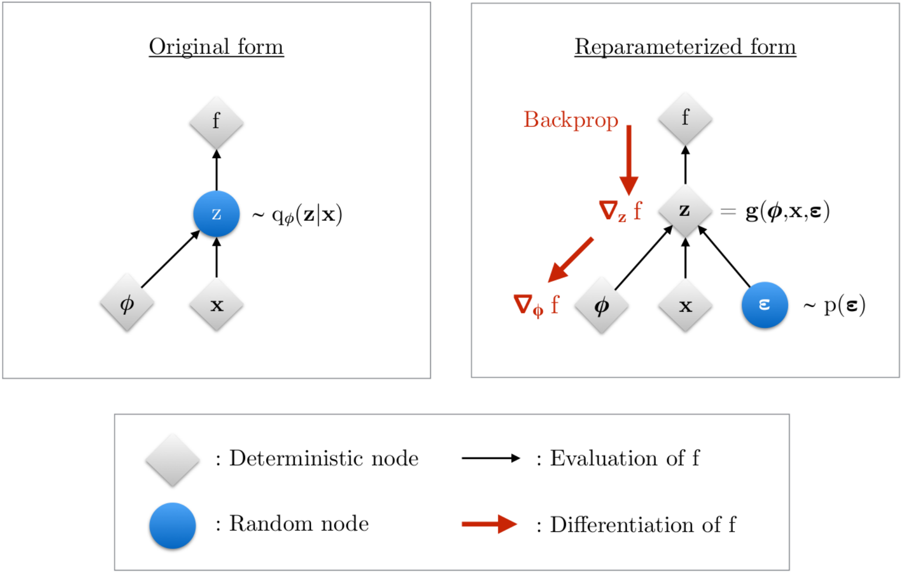
$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[f(\boldsymbol{z})] = \mathbb{E}_{p_{(\boldsymbol{\epsilon})}}[f(g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \boldsymbol{x}))].$$

- Os gradientes em relação a $\boldsymbol{\phi}$ passam a ser:

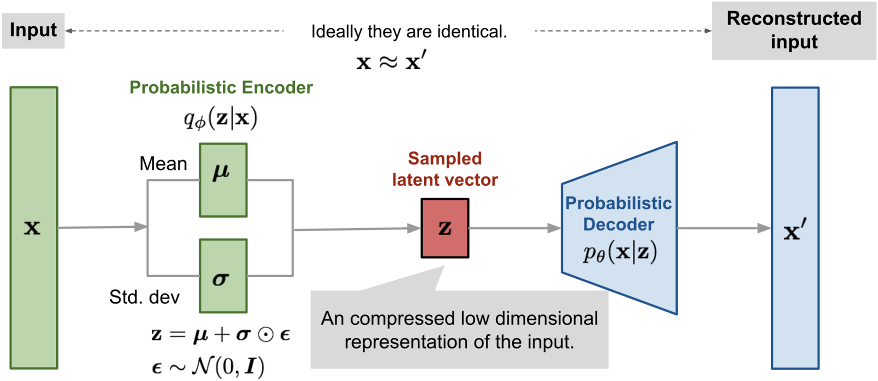
$$\begin{aligned}
\nabla_{\boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[f(\boldsymbol{z})] &= \nabla_{\boldsymbol{\phi}}\mathbb{E}_{p_{(\boldsymbol{\epsilon})}}[f(g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \boldsymbol{x}))] \\
&= \mathbb{E}_{p_{(\boldsymbol{\epsilon})}}[\nabla_{\boldsymbol{\phi}}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

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

Reparameterization trick



Variational autoencoder



Variational autoencoder

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

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

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

Variational autoencoder

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

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

  em que $\boldsymbol{z} = g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \boldsymbol{x})$.
- Caso o termo KL seja analítico, o ELBO pode ser estimado por:

$$\tilde{\boldsymbol{\epsilon}}^{(s)} \sim p(\boldsymbol{\epsilon}), \quad \tilde{\boldsymbol{z}}^{(s)} = g(\boldsymbol{\epsilon}^{(s)}, \boldsymbol{\phi}, \boldsymbol{x}),$$

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

Variational autoencoder

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

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

  em que $\boldsymbol{z} = g(\boldsymbol{\epsilon}, \boldsymbol{\phi}, \boldsymbol{x})$.
- Caso o termo KL seja analítico, o ELBO pode ser estimado por:

$$\tilde{\boldsymbol{\epsilon}}^{(s)} \sim p(\boldsymbol{\epsilon}), \quad \tilde{\boldsymbol{z}}^{(s)} = g(\boldsymbol{\epsilon}^{(s)}, \boldsymbol{\phi}, \boldsymbol{x}),$$

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

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

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

Variational autoencoder

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

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

Variational autoencoder

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

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

- A estimação do gradiente $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \tilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x})$ 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:

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

- A estimação do gradiente $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \tilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x})$ 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:

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

Variational autoencoder

- Voltamos a considerar uma distribuição variacional fatorada:

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

- O truque da reparametrização nos permite reescrever:

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

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

Variational autoencoder

- Dessa maneira, o ELBO pode ser estimado por:

$$
\begin{aligned}
(\boldsymbol{\mu}, \log \boldsymbol{\sigma}) &= \mathrm{EncoderDNN}_{\boldsymbol{\phi}}(\boldsymbol{x}), \\
\tilde{\boldsymbol{\epsilon}}^{(s)} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \boldsymbol{I}), &\quad \tilde{\boldsymbol{z}}^{(s)} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \tilde{\boldsymbol{\epsilon}}^{(s)}, \\
\tilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) &= \frac{1}{S} \sum_{s=1}^{S} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{z}}^{(s)}) - \mathrm{KL}(\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))||\mathcal{N}(\boldsymbol{z}|\mathbf{0}, \boldsymbol{I})).
\end{aligned}
$$

Variational autoencoder

- Dessa maneira, o ELBO pode ser estimado por:

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

- A divergência KL entre Gaussianas é dada por:

$$\mathrm{KL}(\mathcal{N}_0(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)||\mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)) =$$
$$\frac{1}{2}\left[\mathrm{Tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_1(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - K + \log\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|}\right].$$

Variational autoencoder

- Dessa maneira, o ELBO pode ser estimado por:

$$
\begin{aligned}
(\boldsymbol{\mu}, \log \boldsymbol{\sigma}) &= \text{EncoderDNN}_{\boldsymbol{\phi}}(\boldsymbol{x}),\\
\tilde{\boldsymbol{\epsilon}}^{(s)} &\sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \boldsymbol{I}), \quad \tilde{\boldsymbol{z}}^{(s)} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \tilde{\boldsymbol{\epsilon}}^{(s)},\\
\tilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) &= \frac{1}{S}\sum_{s=1}^{S} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{z}}^{(s)}) - \text{KL}(\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))||\mathcal{N}(\boldsymbol{z}|\mathbf{0}, \boldsymbol{I})).
\end{aligned}
$$

- A divergência KL entre Gaussianas é dada por:

$$
\begin{aligned}
&\text{KL}(\mathcal{N}_0(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)||\mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)) =\\
&\frac{1}{2}\left[\text{Tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_1(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - K + \log\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|}\right].
\end{aligned}
$$

- Assim:

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

Variational autoencoder

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

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

Variational autoencoder

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

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

- Nesse caso, o truque de reparametrização poderia ser:

$$\begin{aligned} \boldsymbol{\epsilon} &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \\ \boldsymbol{z} &= \boldsymbol{\mu} + \boldsymbol{L}\boldsymbol{\epsilon}, \end{aligned}$$

  em que $\boldsymbol{L}$ é uma matriz triangular inferior e $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^{\top}$, ou seja, $\boldsymbol{L}$ corresponde à decomposição de Cholesky de $\boldsymbol{\Sigma}$.

Variational autoencoder

- Assim, o ELBO pode ser estimado por:

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}, \boldsymbol{L}') = \mathrm{EncoderDNN}_{\boldsymbol{\phi}}(\boldsymbol{x}),$$
$$\boldsymbol{L} = \boldsymbol{L}' + \mathrm{diag}(\boldsymbol{\sigma}), \quad \tilde{\boldsymbol{\epsilon}}^{(s)} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \boldsymbol{I}), \quad \tilde{\boldsymbol{z}}^{(s)} = \boldsymbol{\mu} + \boldsymbol{L}\boldsymbol{\epsilon}^{(s)},$$
$$\tilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) = \frac{1}{S}\sum_{s=1}^{S} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{z}}^{(s)}) - \mathrm{KL}(\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma})||\mathcal{N}(\boldsymbol{z}|\mathbf{0}, \boldsymbol{I})),$$

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

Variational autoencoder

- Assim, o ELBO pode ser estimado por:

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}, \boldsymbol{L}') = \text{EncoderDNN}_{\boldsymbol{\phi}}(\boldsymbol{x}),$$
$$\boldsymbol{L} = \boldsymbol{L}' + \text{diag}(\boldsymbol{\sigma}), \quad \tilde{\boldsymbol{\epsilon}}^{(s)} \sim \mathcal{N}(\boldsymbol{\epsilon}|\boldsymbol{0}, \boldsymbol{I}), \quad \tilde{\boldsymbol{z}}^{(s)} = \boldsymbol{\mu} + \boldsymbol{L}\boldsymbol{\epsilon}^{(s)},$$
$$\tilde{\mathcal{L}}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) = \frac{1}{S}\sum_{s=1}^{S} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\tilde{\boldsymbol{z}}^{(s)}) - \text{KL}(\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma})||\mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})),$$

  em que $\boldsymbol{L}'$ é uma matriz triangular com zeros na diagonal.
- Como o KL entre Gaussianas é analítico, temos:

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

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

Variational autoencoder

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

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

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

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

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

- A matriz Jacobiana é definida por:

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

Variational autoencoder

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

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

  → A Jacobiana é simples:
  $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\epsilon}} = \mathrm{diag}(\boldsymbol{\sigma})$ e $\log \left|\det \left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\epsilon}}\right)\right| = \sum_k \log \sigma_k$.

Variational autoencoder

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

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

  $\rightarrow$ A Jacobiana é simples:
  $\frac{\partial z}{\partial \boldsymbol{\epsilon}} = \mathrm{diag}(\boldsymbol{\sigma})$ e $\log \left|\det \left(\frac{\partial z}{\partial \boldsymbol{\epsilon}}\right)\right| = \sum_k \log \sigma_k$.

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

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

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

  $\rightarrow$ A Jacobiana continua simples:
  $\frac{\partial z}{\partial \boldsymbol{\epsilon}} = \boldsymbol{L}$ e $\log \left|\det \left(\frac{\partial z}{\partial \boldsymbol{\epsilon}}\right)\right| = \sum_k \log L_{kk}$.

Variational autoencoder

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

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

  $\rightarrow$ A Jacobiana é simples:
  $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\epsilon}} = \mathrm{diag}(\boldsymbol{\sigma})$ e $\log \left|\det\left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\epsilon}}\right)\right| = \sum_k \log \sigma_k$.

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

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

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

  $\rightarrow$ A Jacobiana continua simples:
  $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\epsilon}} = \boldsymbol{L}$ e $\log \left|\det\left(\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\epsilon}}\right)\right| = \sum_k \log L_{kk}$.

- Uma sequência de transformações $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{L}\boldsymbol{\epsilon}$ tornaria $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ 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 $\mathcal{D} = \{\boldsymbol{x}_i\}|_{i=1}^N$ e escolha uma verossimilhança $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$;
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 $\boldsymbol{x} \in \mathcal{M}$ (considerando $S = 1$):

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

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

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

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

Variational autoencoder

Resumo do algoritmo

1. A verossimilhança marginal de um padrão $\boldsymbol{x}$ é estimada por:

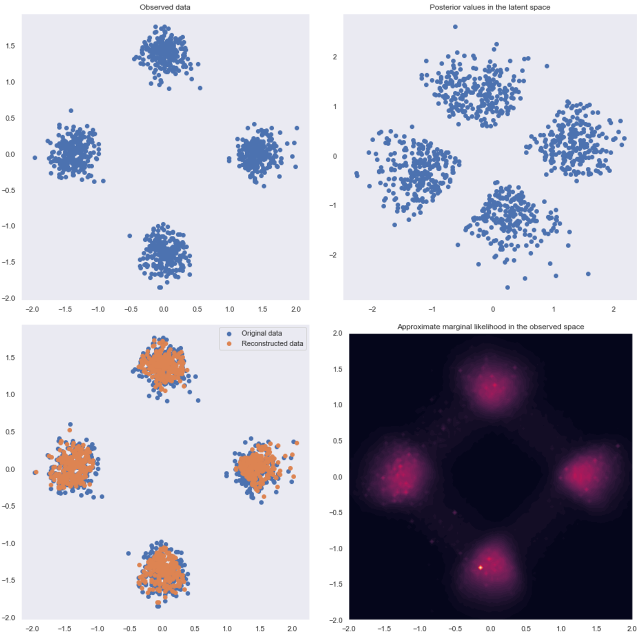
$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}$$

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

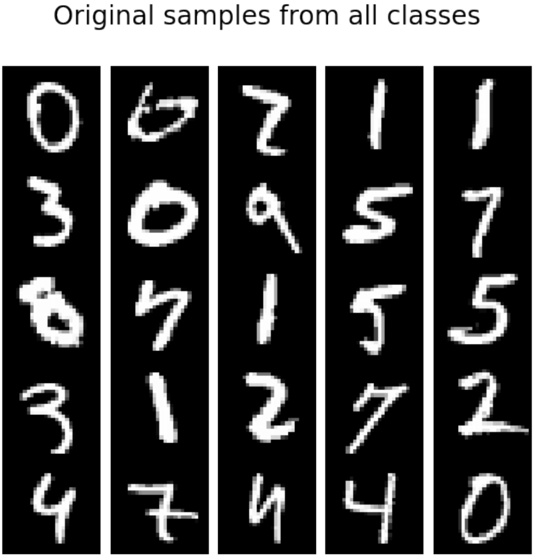
em que $\tilde{\boldsymbol{z}}^{(s)} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ é uma amostra da posteriori variacional.

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

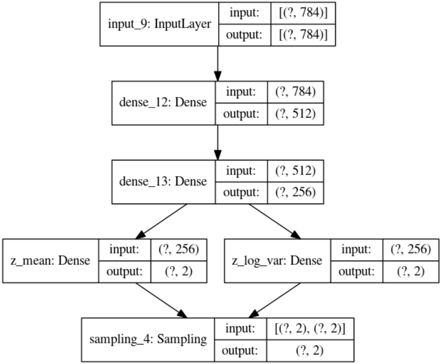
VAE - 4 Gaussianas $N = 1000, D = 2, K = 2$



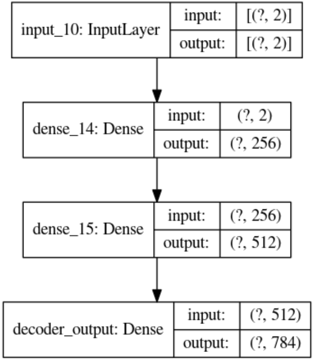
Dígitos MNIST $N = 70000, D = 784$



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

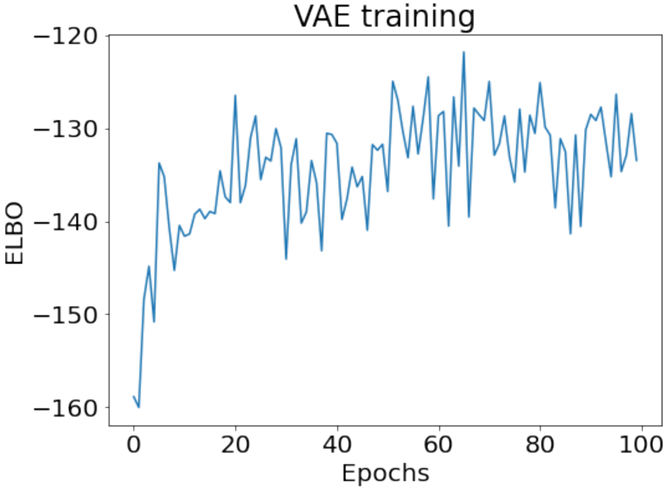


VAE encoder



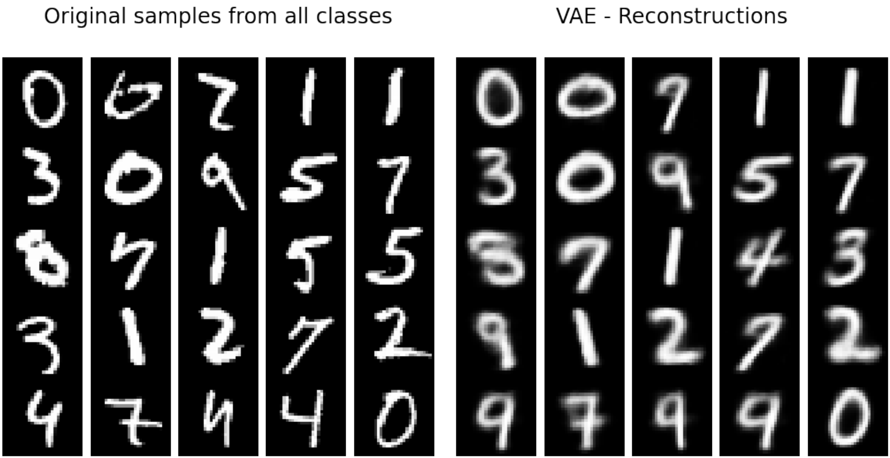
VAE decoder

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



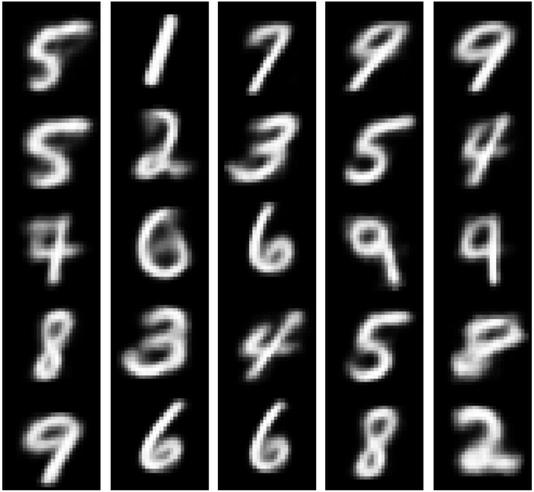
$$\underbrace{784}_{\text{dado}} \underbrace{\xrightarrow{\text{ReLU}} 512 \xrightarrow{\text{ReLU}} 256 \xrightarrow{\text{Linear}}}_{\text{encoder}} \underbrace{2}_{\boldsymbol{\mu}, \log \boldsymbol{\sigma} \rightarrow \text{sample}} \underbrace{\xrightarrow{\text{ReLU}} 256 \xrightarrow{\text{ReLU}} 512 \xrightarrow{\text{Sigmoid}}}_{\text{decoder}} \underbrace{784}_{\text{reconstrução}}$$

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

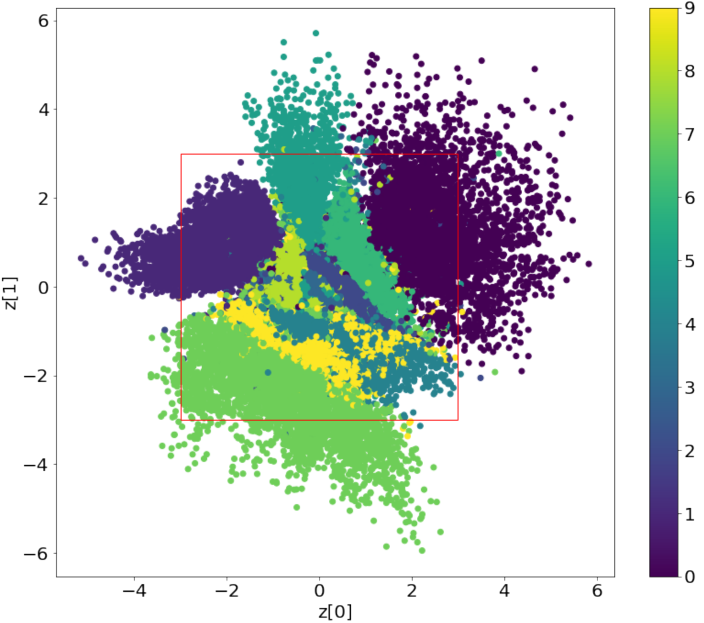


VAE - Geração MNIST $N = 70000, D = 784, K = 2$

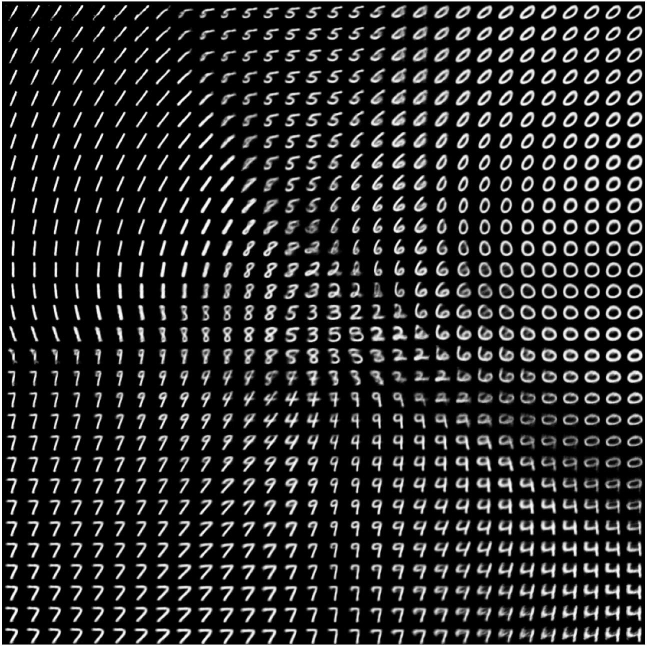
VAE - generated samples



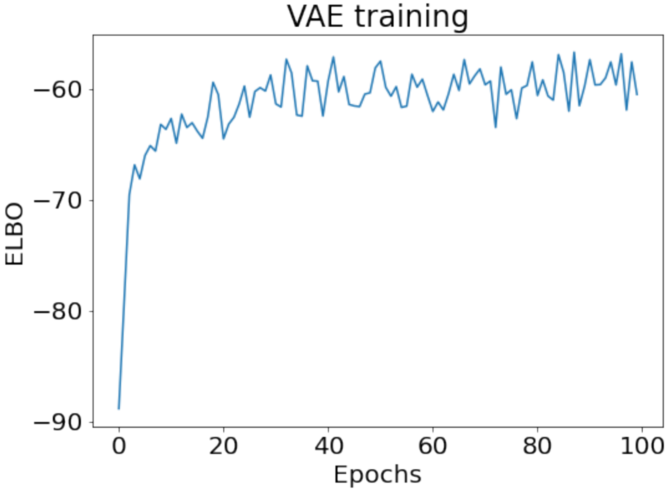
VAE - Espaço latente MNIST $D = 784, K = 2$



VAE - Geração MNIST $D = 784, K = 2$

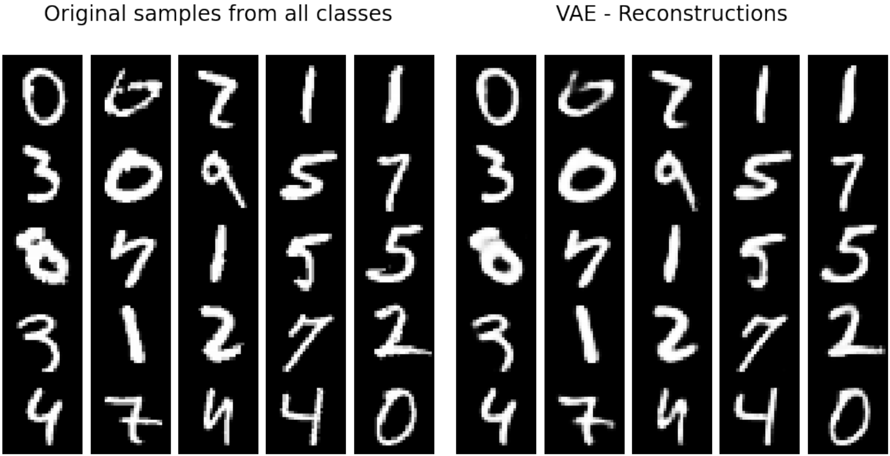


VAE com RNAs - MNIST $N = 70000, D = 784, K = 32$



$$\underbrace{784}_{\text{dado}} \underbrace{\xrightarrow{\text{ReLU}} 512 \xrightarrow{\text{ReLU}} 256 \xrightarrow{\text{Linear}}}_{\text{encoder}} \underbrace{32}_{\boldsymbol{\mu}, \log \boldsymbol{\sigma} \to \text{sample}} \underbrace{\xrightarrow{\text{ReLU}} 256 \xrightarrow{\text{ReLU}} 512 \xrightarrow{\text{Sigmoid}}}_{\text{decoder}} \underbrace{784}_{\text{reconstrução}}$$

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

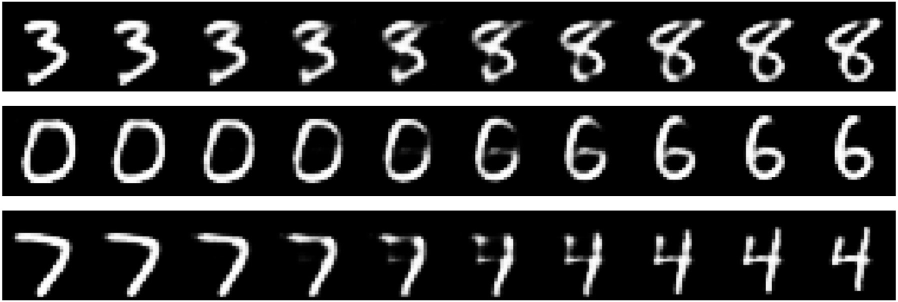


VAE - Interpolação MNIST $N = 70000, D = 784, K = 32$

- Dados dois padrões $\boldsymbol{x}_1$ e $\boldsymbol{x}_2$, podemos interpolar novas amostras:

$$
\begin{aligned}
\boldsymbol{z}_1 &= \text{encoder}(\boldsymbol{x}_1), \quad \boldsymbol{z}_2 = \text{encoder}(\boldsymbol{x}_2),\\
\boldsymbol{z} &= (1-\lambda)\boldsymbol{z}_1 + \lambda \boldsymbol{z}_2, \quad 0 \le \lambda \le 1,\\
\boldsymbol{x} &= \text{decoder}(\boldsymbol{z}).
\end{aligned}
$$

- 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):

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

- 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`.