Variational Autoencoders (VAE)

Jérémie Cabessa

Laboratoire DAVID, UVSQ

Introduction

- **But d'un autoencodeur (rappel):** projeter les data dans un espace de dimension plus petite – l'**espace latent (latent space)** – et être capable de les reconstruire ces data.
- On avait vu que l'**encodeur** pouvait être utilisé comme un **data compressor**...
- et le **décodeur** comme un **data generator**.
- Pour cela, il suffisait de sampler des points dans l'espace latent et de les décoder.

Introduction

- **But d'un autoencodeur (rappel):** projeter les data dans un espace de dimension plus petite – l'**espace latent (latent space)** – et être capable de les reconstruire ces data.
- On avait vu que l'**encodeur** pouvait être utilisé comme un **data compressor**...
- et le **décodeur** comme un **data generator**.
- Pour cela, il suffisait de sampler des points dans l'espace latent et de les décoder.

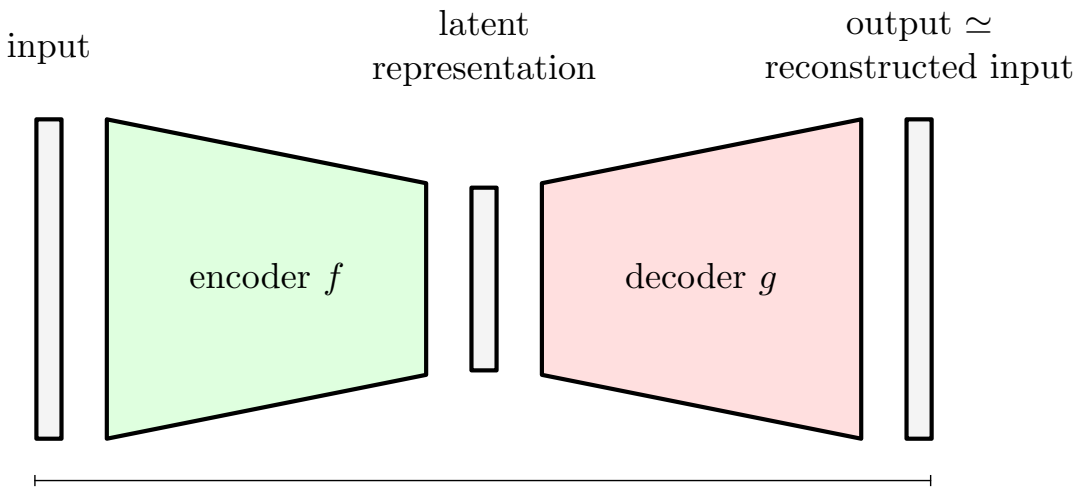
Introduction

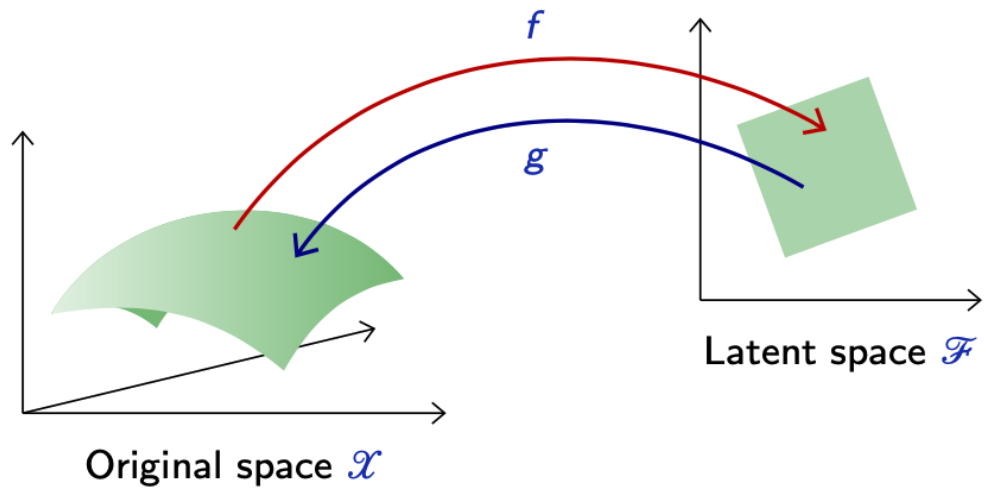
- **But d'un autoencodeur (rappel):** projeter les data dans un espace de dimension plus petite – l'**espace latent (latent space)** – et être capable de les reconstruire ces data.
- On avait vu que l'**encodeur** pouvait être utilisé comme un **data compressor**...
- et le **décodeur** comme un **data generator**.
- Pour cela, il suffisait de sampler des points dans l'espace latent et de les décoder.

Introduction

- **But d'un autoencodeur (rappel):** projeter les data dans un espace de dimension plus petite – l'**espace latent (latent space)** – et être capable de les reconstruire ces data.
- On avait vu que l'**encodeur** pouvait être utilisé comme un **data compressor**...
- et le **décodeur** comme un **data generator**.
- Pour cela, il suffisait de sampler des points dans l'espace latent et de les décoder.

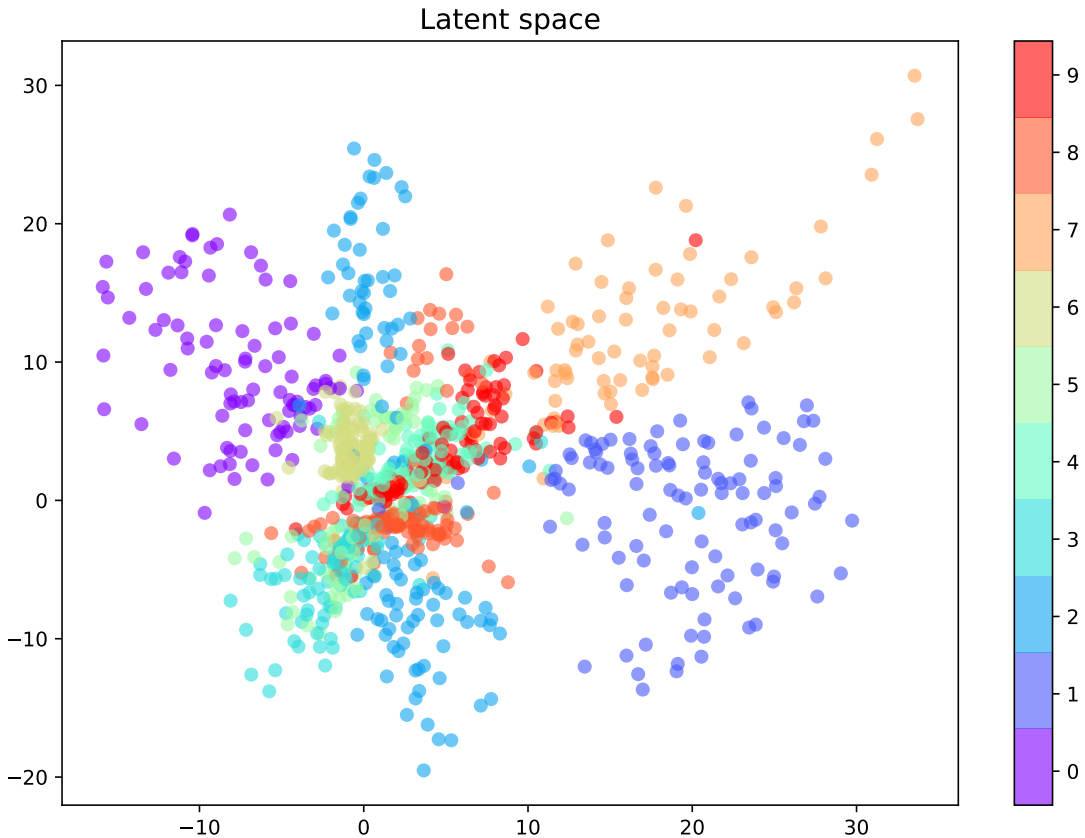
Introduction





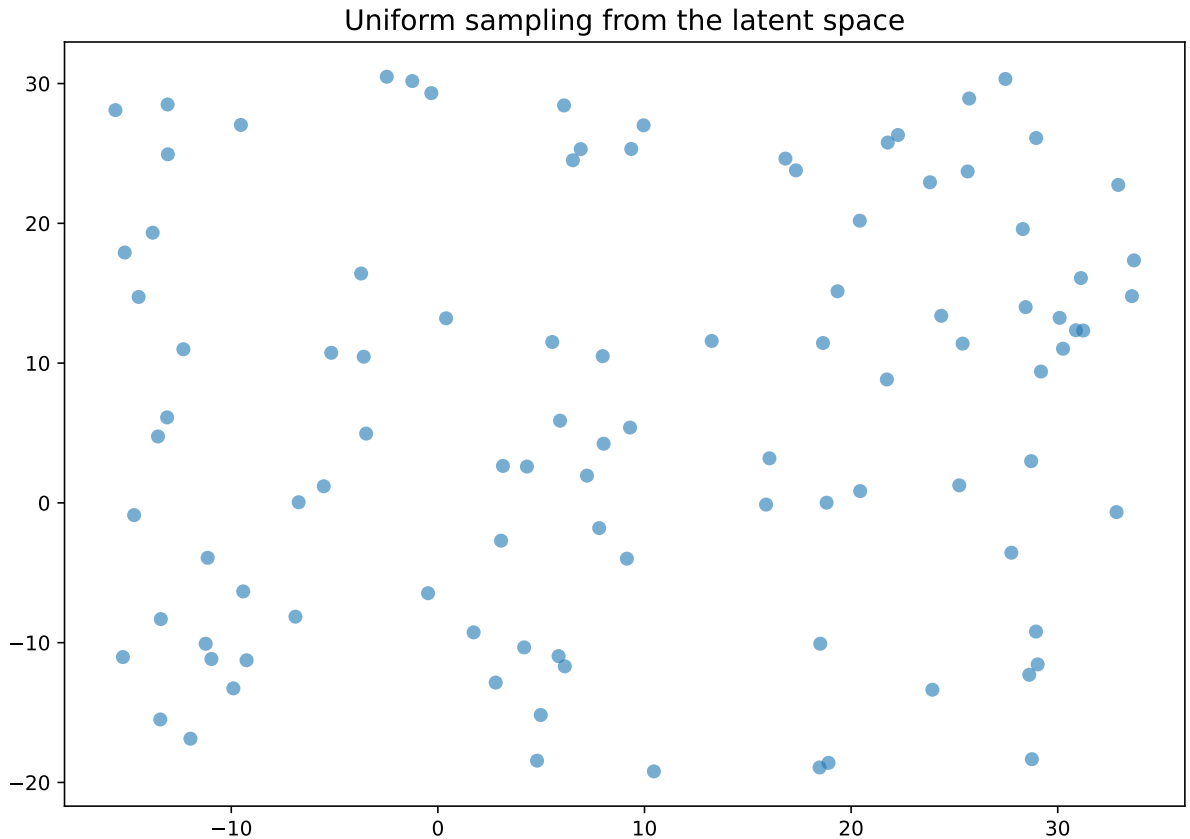
Motivation

- Dans notre cas, on avait un espace latent de dimension 2, et lorsqu'on samplait dans cet espace, les data générées étaient d'assez bonne qualité (à l'oeil).



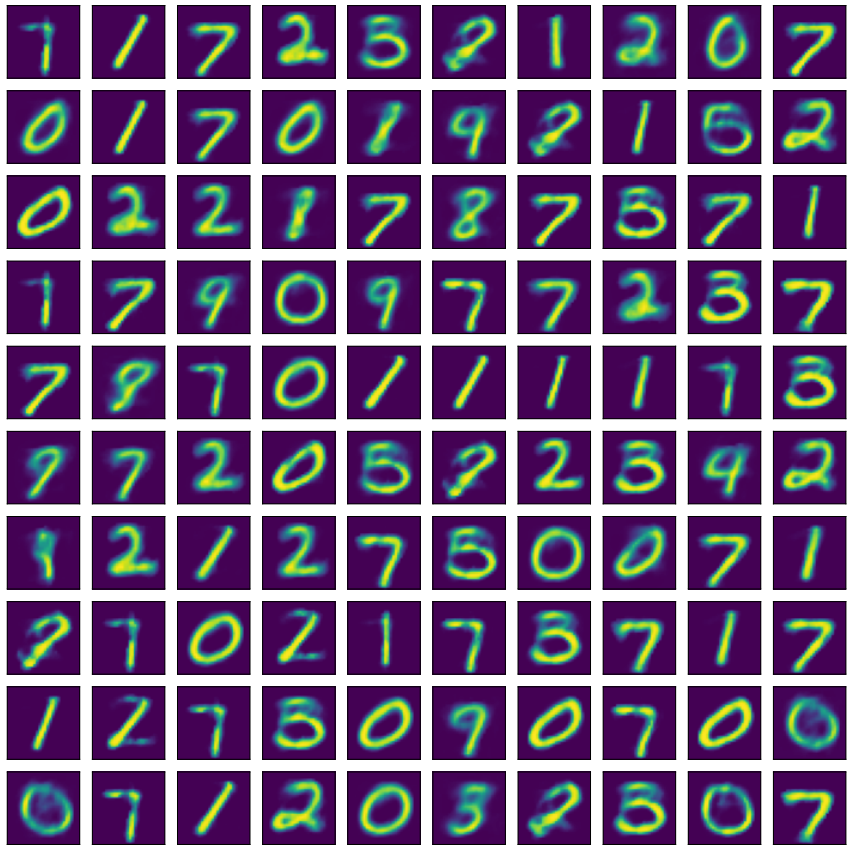
Motivation

- Dans notre cas, on avait un espace latent de dimension 2, et lorsqu'on samplait dans cet espace, les data générées étaient d'assez bonne qualité (à l'oeil).



Motivation

- Dans notre cas, on avait un espace latent de dimension 2, et lorsqu'on samplait dans cet espace, les data générées étaient d'assez bonne qualité (à l'oeil).



Motivation

- Mais cette situation n'est pas représentative...
- En pratique, on considère des espace latent de dimensions plus grandes que 2 (on arrive rarement à compresser un signal en dimension 2).
- Dans ce cas, le sampling dans l'espace latent génère des data beaucoup plus dégradées...

Motivation

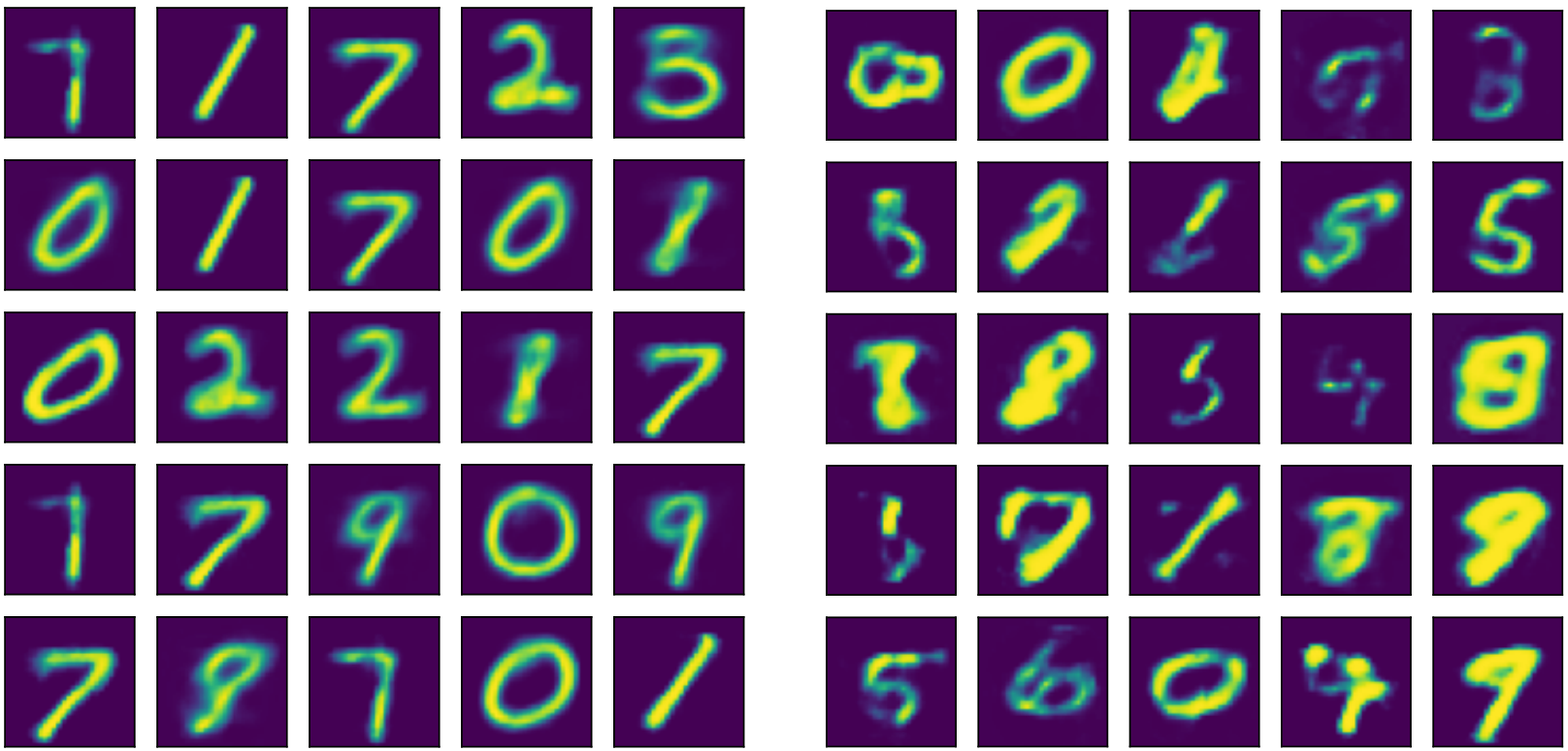
- Mais cette situation n'est pas représentative...
- En pratique, on considère des espace latent de dimensions plus grandes que 2 (on arrive rarement à compresser un signal en dimension 2).
- Dans ce cas, le sampling dans l'espace latent génère des data beaucoup plus dégradées...

Motivation

- Mais cette situation n'est pas représentative...
- En pratique, on considère des espace latent de dimensions plus grandes que 2 (on arrive rarement à compresser un signal en dimension 2).
- Dans ce cas, le sampling dans l'espace latent génère des data beaucoup plus dégradées...

Motivation

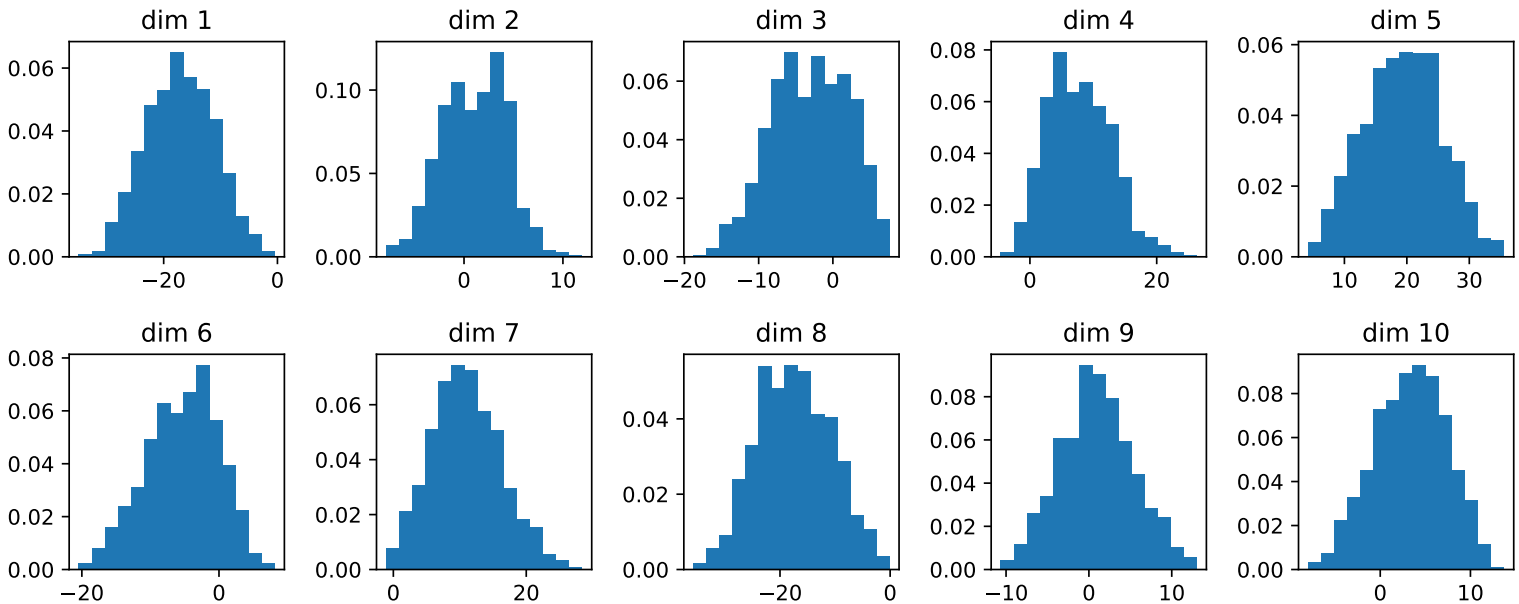
- Par exemple, si on répète l'opération avec un espace latent de dimension 10, et qu'on sample dans cet espace, les data générées sont bien moins bonnes.



Data generated from latent spaces of dim 2 and 10, respectively.

Motivation

- De plus, la distribution des data dans l'espace latent est très déséquilibrée (non-centered, skewed, etc.).



Distribution of data in the latent space for each dimension.

Motivation

- Pour un autoencodeur classique, qu'est-ce qui rend le processus de sampling de l'espace latent difficile?
  - ⋆ Distribution de l'espace latent non-continue: ne se voit pas forcément en 2D, mais en dimension supérieure, il y aura sûrement des "trous".
  - ⋆ Distribution de l'espace latent non-équilibrée et non-centrée.
- L'encodeur variationnel (variational autoencoder VAE) vise à pallier ces problèmes.
- Il force l'espace latent à suivre une certaine distribution, normale en général.

Motivation

- Pour un autoencodeur classique, qu'est-ce qui rend le processus de sampling de l'espace latent difficile?
  - ⋆ Distribution de l'espace latent non-continue: ne se voit pas forcément en 2D, mais en dimension supérieure, il y aura sûrement des "trous".
  - ⋆ Distribution de l'espace latent non-équilibrée et non-centrée.
- L'**encodeur variationnel (variational autoencoder VAE)** vise à pallier ces problèmes.
- Il force l'espace latent à suivre une certaine distribution, normale en général.

Motivation

- Pour un autoencodeur classique, qu'est-ce qui rend le processus de sampling de l'espace latent difficile?
  - ⋆ Distribution de l'espace latent non-continue: ne se voit pas forcément en 2D, mais en dimension supérieure, il y aura sûrement des "trous".
  - ⋆ Distribution de l'espace latent non-équilibrée et non-centrée.
- L'encodeur variationnel (variational autoencoder VAE) vise à pallier ces problèmes.
- Il force l'espace latent à suivre une certaine distribution, normale en général.

Motivation

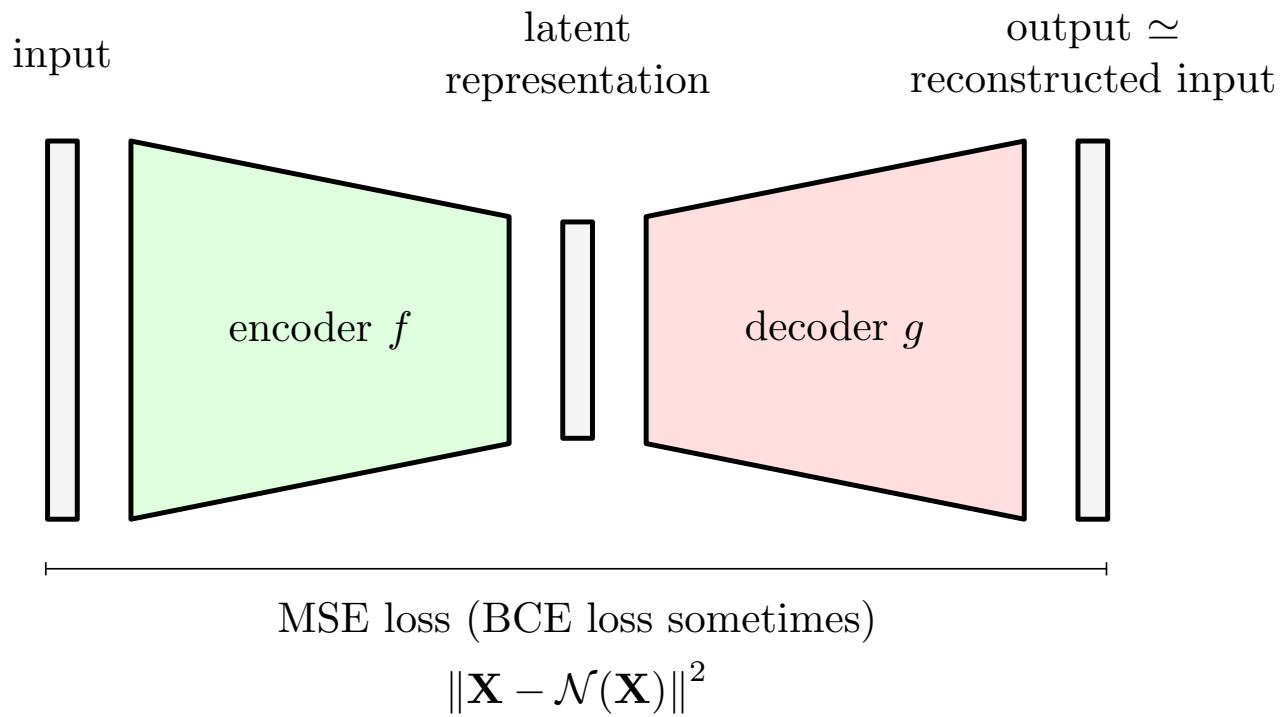
- Pour un autoencodeur classique, qu'est-ce qui rend le processus de sampling de l'espace latent difficile?
  - ⋆ Distribution de l'espace latent non-continue: ne se voit pas forcément en 2D, mais en dimension supérieure, il y aura sûrement des "trous".
  - ⋆ Distribution de l'espace latent non-équilibrée et non-centrée.
- L'**encodeur variationnel (variational autoencoder VAE)** vise à pallier ces problèmes.
- Il force l'espace latent à suivre une certaine distribution, normale en général.

Motivation

- Pour un autoencodeur classique, qu'est-ce qui rend le processus de sampling de l'espace latent difficile?
  - ⋆ Distribution de l'espace latent non-continue: ne se voit pas forcément en 2D, mais en dimension supérieure, il y aura sûrement des "trous".
  - ⋆ Distribution de l'espace latent non-équilibrée et non-centrée.
- L'**encodeur variationnel (variational autoencoder VAE)** vise à pallier ces problèmes.
- Il force l'espace latent à suivre une certaine distribution, normale en général.

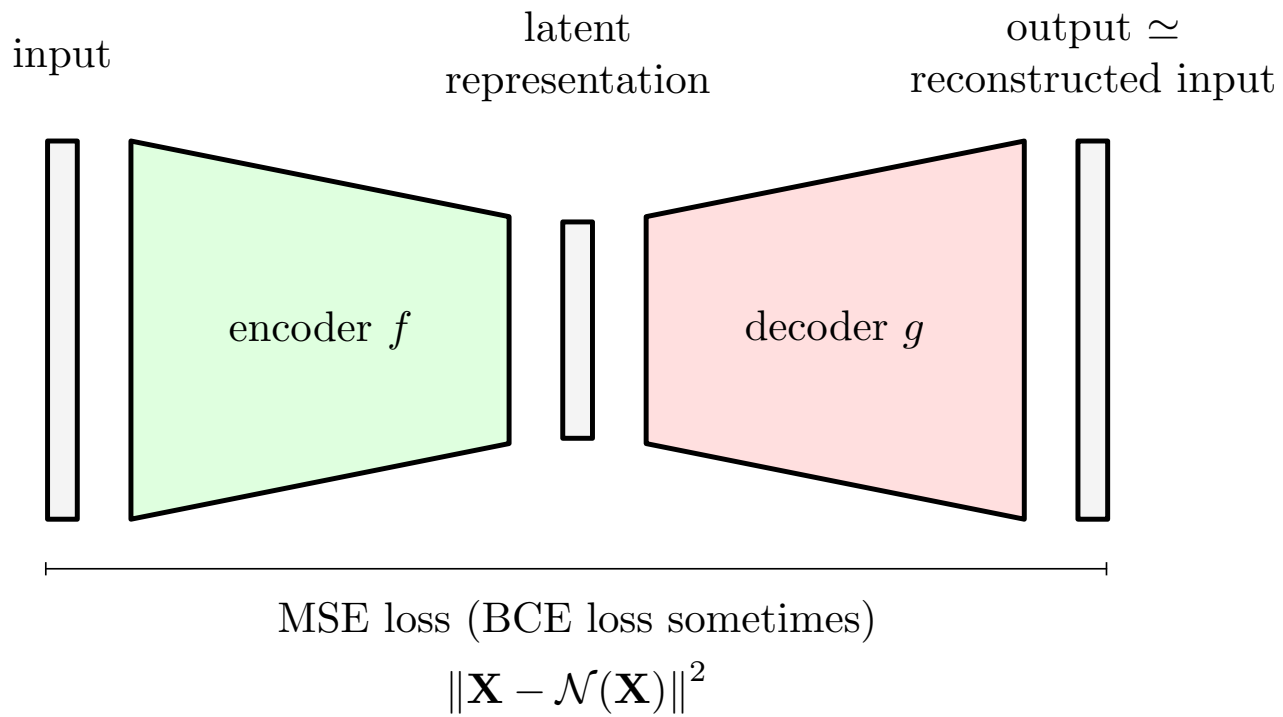
Variational Autoencoder (VAE)

- **Idée générale:** Imposer une distribution sur l'espace latent et entraîner un décodeur de manière à reconstruire les data.
- On passe d'un bottleneck *déterministe* à un bottleneck *stochastique*.



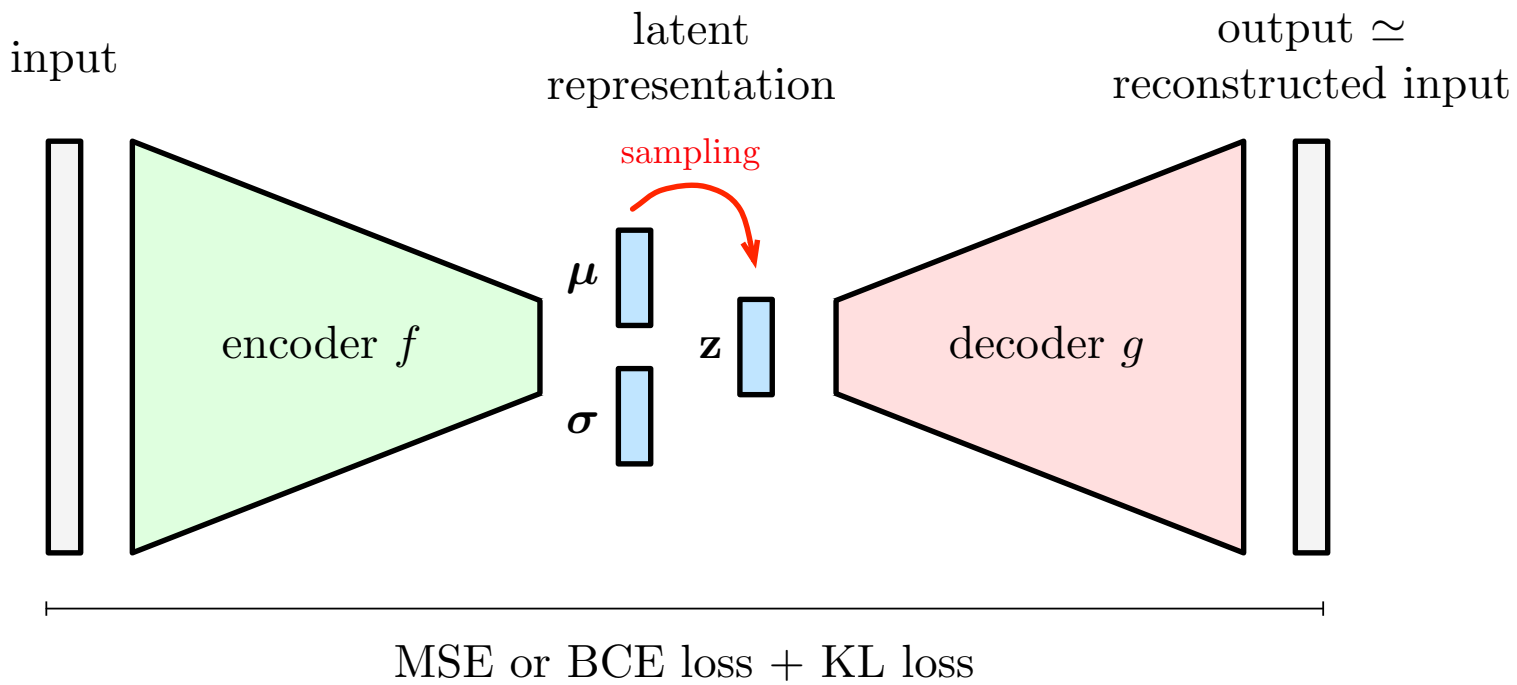
Variational Autoencoder (VAE)

- **Idée générale:** Imposer une distribution sur l'espace latent et entraîner un décodeur de manière à reconstruire les data.
- On passe d'un bottleneck *déterministe* à un bottleneck *stochastique*.



Variational Autoencoder (VAE)

- **Idée générale:** Imposer une distribution sur l'espace latent et entraîner un décodeur de manière à reconstruire les data.
- On passe d'un bottleneck *déterministe* à un bottleneck *stochastique*.



Variational Autoencoder (VAE)

- Régularisation des data: on impose des contraintes de de *continuité* et de *complétude* sur l'espace latent.
- **Contrainte de continuité:** on aimerait que des points proches de l'espace latent génère des data similaires.

→ On introduit un processus de décodage stochastique lors de l'entraînement: chaque data est samplée dans l'espace latent (voisinage) avant d'être décodée.

- **Contrainte de complétude:** on aimerait que tous les points de l'espace latent génèrent des data qui soient valables.

→ Pour "ramasser les data" (éviter les trous), on force les data de l'espace latent à suivre une loi normale centrée réduite $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Variational Autoencoder (VAE)

- Régularisation des data: on impose des contraintes de de *continuité* et de *complétude* sur l'espace latent.
- **Contrainte de continuité:** on aimerait que des points proches de l'espace latent génère des data similaires.

→ On introduit un processus de décodage stochastique lors de l'entraînement: chaque data est samplée dans l'espace latent (voisinage) avant d'être décodée.

- **Contrainte de complétude:** on aimerait que tous les points de l'espace latent génèrent des data qui soient valables.

→ Pour "ramasser les data" (éviter les trous), on force les data de l'espace latent à suivre une loi normale centrée réduite $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Variational Autoencoder (VAE)

- Régularisation des data: on impose des contraintes de de *continuité* et de *complétude* sur l'espace latent.
- **Contrainte de continuité:** on aimerait que des points proches de l'espace latent génère des data similaires.

→ On introduit un processus de décodage stochastique lors de l'entraînement: chaque data est samplée dans l'espace latent (voisinage) avant d'être décodée.

- **Contrainte de complétude:** on aimerait que tous les points de l'espace latent génèrent des data qui soient valables.

→ Pour "ramasser les data" (éviter les trous), on force les data de l'espace latent à suivre une loi normale centrée réduite $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Variational Autoencoder (VAE)

- Régularisation des data: on impose des contraintes de de *continuité* et de *complétude* sur l'espace latent.
- **Contrainte de continuité:** on aimerait que des points proches de l'espace latent génère des data similaires.

→ On introduit un processus de décodage stochastique lors de l'entraînement: chaque data est samplée dans l'espace latent (voisinage) avant d'être décodée.

- **Contrainte de complétude:** on aimerait que tous les points de l'espace latent génèrent des data qui soient valables.

→ Pour "ramasser les data" (éviter les trous), on force les data de l'espace latent à suivre une loi normale centrée réduite $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Variational Autoencoder (VAE)

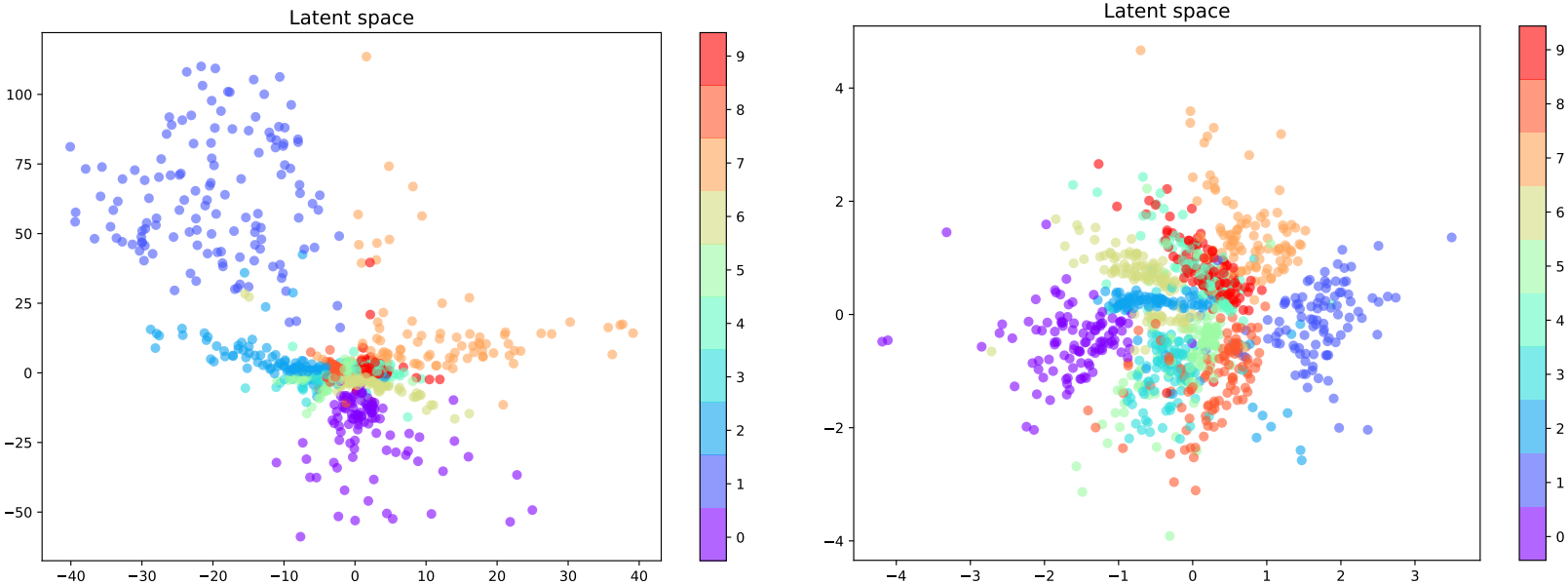
- Régularisation des data: on impose des contraintes de de *continuité* et de *complétude* sur l'espace latent.
- **Contrainte de continuité:** on aimerait que des points proches de l'espace latent génère des data similaires.

→ On introduit un processus de décodage stochastique lors de l'entraînement: chaque data est samplée dans l'espace latent (voisinage) avant d'être décodée.

- **Contrainte de complétude:** on aimerait que tous les points de l'espace latent génèrent des data qui soient valables.

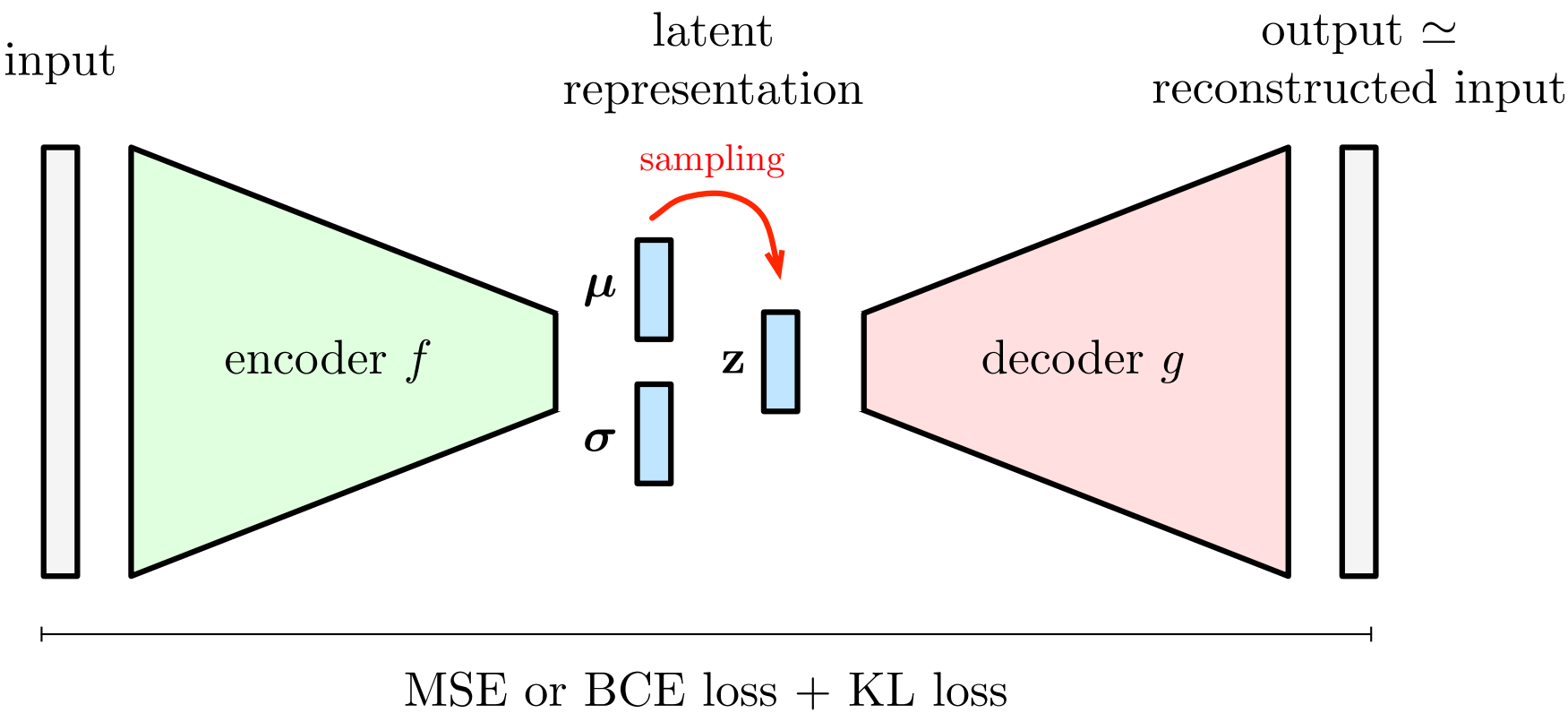
→ Pour "ramasser les data" (éviter les trous), on force les data de l'espace latent à suivre une loi normale centrée réduite $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Variational Autoencoder (VAE)



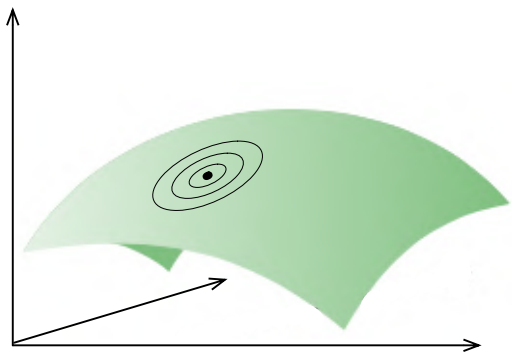
Left: less continuous and complete latent space; **Right:** more continuous and complete latent space.

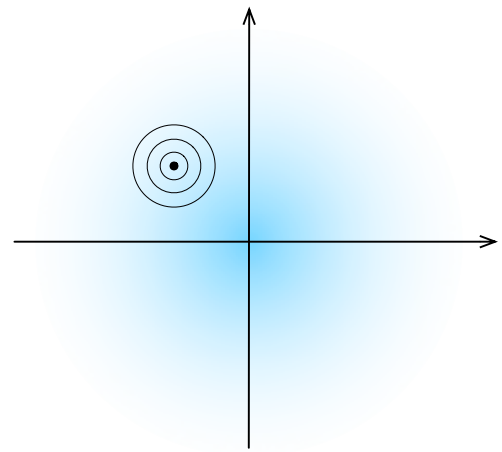
Variational Autoencoder (VAE)



Variational Autoencoder (VAE)

$q(\mathbf{X})$ —encoder f→ $q_{\mathbf{\Phi}}(\mathbf{Z} \mid \mathbf{X} = \mathbf{x}) \simeq \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$ where $\mu_{\mathbf{x}}, \mathbf{\Sigma}_{\mathbf{x}} := f(\mathbf{x}; \mathbf{\Phi})$

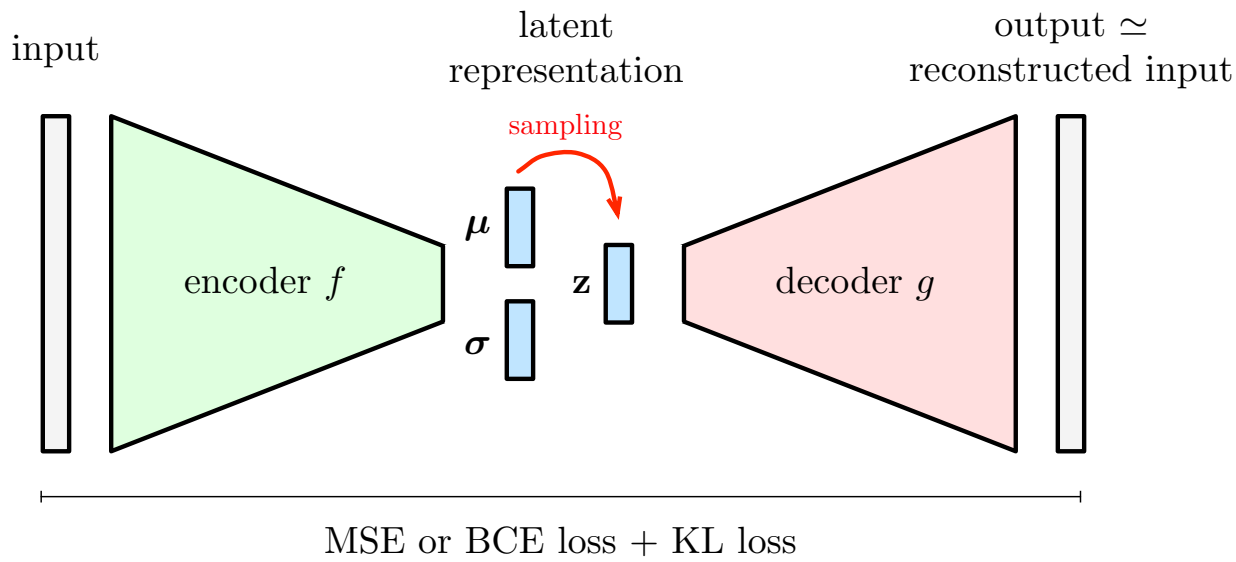




$p_{\mathbf{\Theta}}(\mathbf{X} \mid \mathbf{Z} = \mathbf{z})$ ←decoder g— $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$ where $p_{\mathbf{\Theta}}(\mathbf{Z}) \simeq \mathcal{N}(\mathbf{0}, \mathbf{I})$

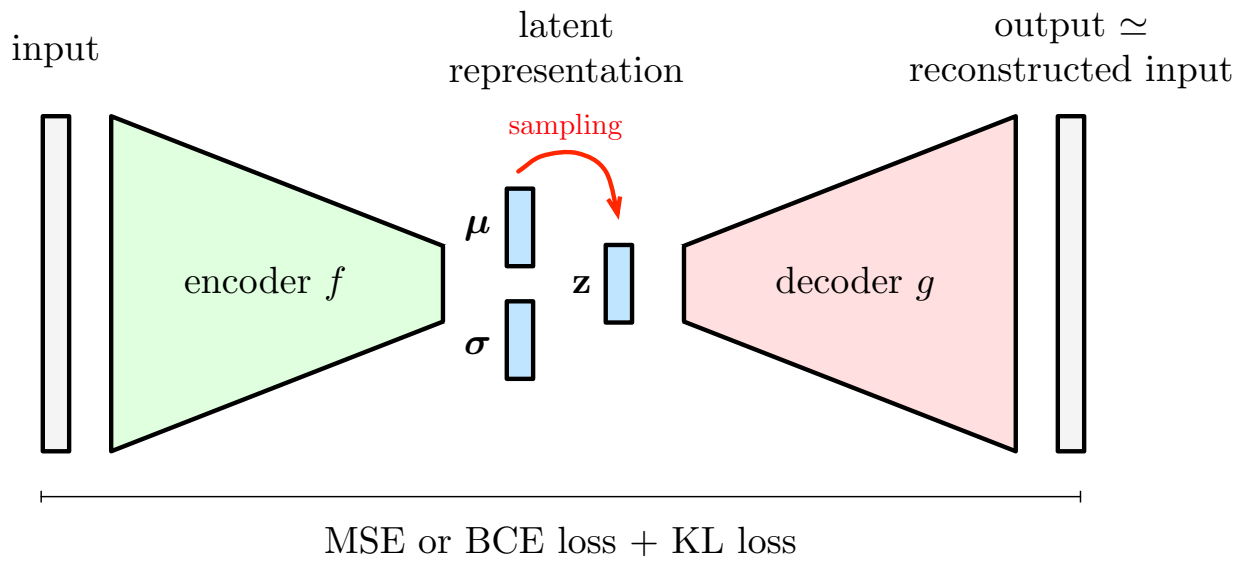
Variational Autoencoder (VAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Phi})$.
- Le **sampler** $s(\cdot; \mu, \Sigma)$ sample selon une loi normale $\mathcal{N}(\mu, \Sigma)$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta})$.
- La composition de $f(\cdot; \mathbf{\Phi})$, $s(\cdot; \mu, \Sigma)$ et $g(\cdot; \mathbf{\Theta})$ forme un **réseau de neurones stochastique** $\mathcal{N}(\cdot; \mathbf{\Phi}, \mathbf{\Theta})$.



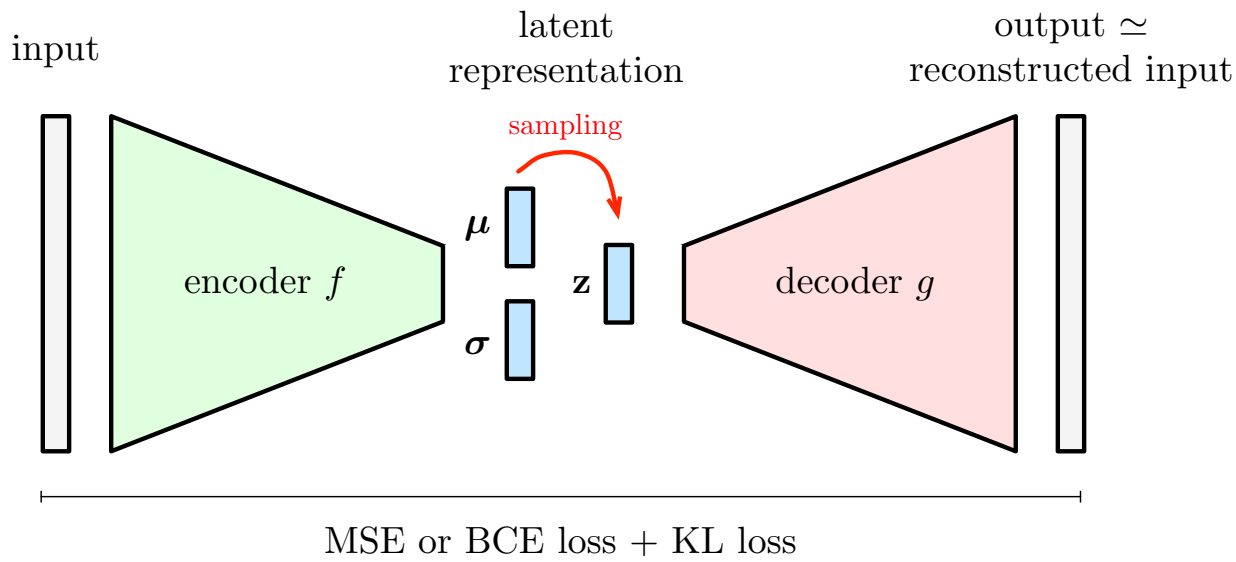
Variational Autoencoder (VAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Phi})$.
- Le **sampler** $s(\cdot; \boldsymbol{\mu}, \mathbf{\Sigma})$ sample selon une loi normale $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta})$.
- La composition de $f(\cdot; \mathbf{\Phi})$, $s(\cdot; \boldsymbol{\mu}, \mathbf{\Sigma})$ et $g(\cdot; \mathbf{\Theta})$ forme un **réseau de neurones stochastique** $\mathcal{N}(\cdot; \mathbf{\Phi}, \mathbf{\Theta})$.



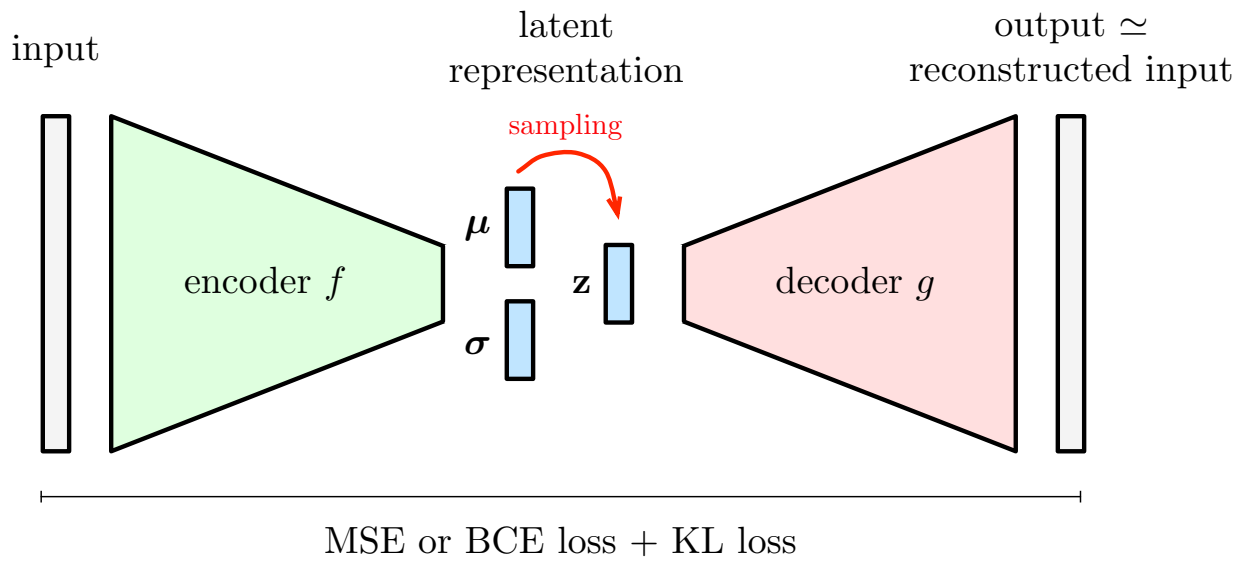
Variational Autoencoder (VAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Phi})$.
- Le **sampler** $s(\cdot; \boldsymbol{\mu}, \mathbf{\Sigma})$ sample selon une loi normale $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta})$.
- La composition de $f(\cdot; \mathbf{\Phi})$, $s(\cdot; \boldsymbol{\mu}, \mathbf{\Sigma})$ et $g(\cdot; \mathbf{\Theta})$ forme un **réseau de neurones stochastique** $\mathcal{N}(\cdot; \mathbf{\Phi}, \mathbf{\Theta})$.



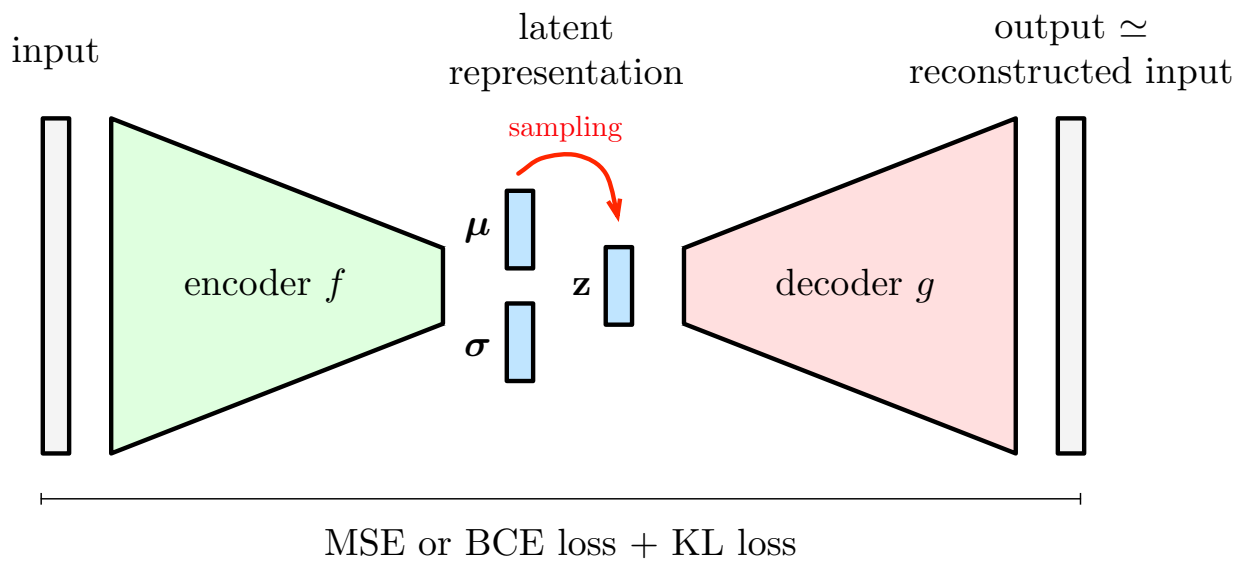
Variational Autoencoder (VAE)

- L'**encodeur** est un deep neural network $f(\cdot; \mathbf{\Phi})$.
- Le **sampler** $s(\cdot; \boldsymbol{\mu}, \mathbf{\Sigma})$ sample selon une loi normale $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$.
- Le **décodeur** est un deep neural network $g(\cdot; \mathbf{\Theta})$.
- La composition de $f(\cdot; \mathbf{\Phi})$, $s(\cdot; \boldsymbol{\mu}, \mathbf{\Sigma})$ et $g(\cdot; \mathbf{\Theta})$ forme un **réseau de neurones stochastique** $\mathcal{N}(\cdot; \mathbf{\Phi}, \mathbf{\Theta})$.



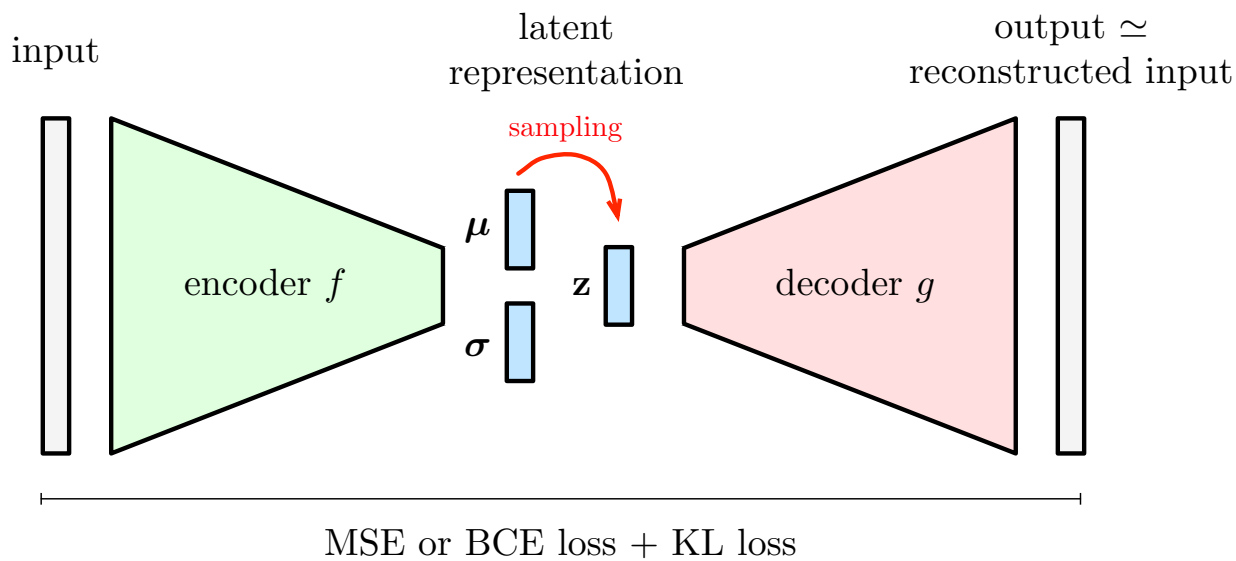
Variational Autoencoder (VAE)

1. L'**encodeur** $f(\cdot;\mathbf{\Phi})$ encode une data $\mathbf{x}$ en deux vecteurs $\boldsymbol{\mu}_\mathbf{x}$ et $\mathbf{\Sigma}_\mathbf{x}$ de plus petite dimension.
2. Le **sampler** $s(\cdot;\boldsymbol{\mu}_\mathbf{x},\mathbf{\Sigma}_\mathbf{x})$ sample un point $\mathbf{z}$ dans l'espace latent selon la loi normale $\mathcal{N}(\boldsymbol{\mu}_\mathbf{x},\mathbf{\Sigma}_\mathbf{x})$.
3. Le **décodeur** $g(\cdot;\mathbf{\Theta})$ reconstruit $\mathbf{x}$ à partir de $\mathbf{z}$.



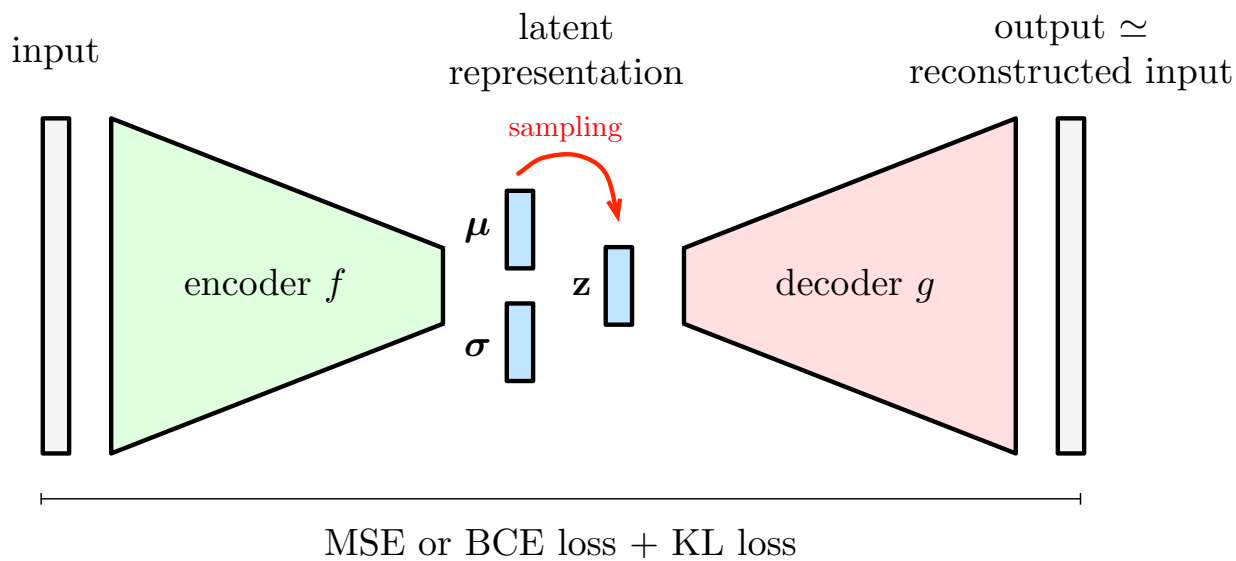
Variational Autoencoder (VAE)

1. L'**encodeur** $f(\cdot;\mathbf{\Phi})$ encode une data $\mathbf{x}$ en deux vecteurs $\boldsymbol{\mu}_\mathbf{x}$ et $\mathbf{\Sigma}_\mathbf{x}$ de plus petite dimension.
2. Le **sampler** $s(\cdot;\boldsymbol{\mu}_\mathbf{x},\mathbf{\Sigma}_\mathbf{x})$ sample un point $\mathbf{z}$ dans l'espace latent selon la loi normale $\mathcal{N}(\boldsymbol{\mu}_\mathbf{x},\mathbf{\Sigma}_\mathbf{x})$.
3. Le **décodeur** $g(\cdot;\mathbf{\Theta})$ reconstruit $\mathbf{x}$ à partir de $\mathbf{z}$.



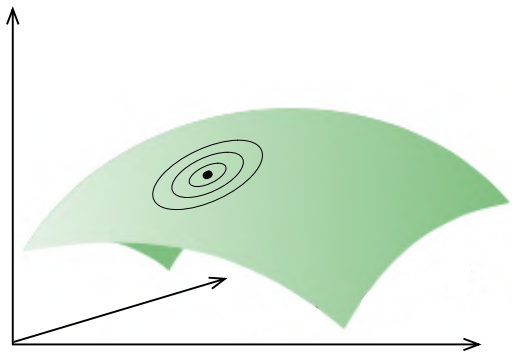
Variational Autoencoder (VAE)

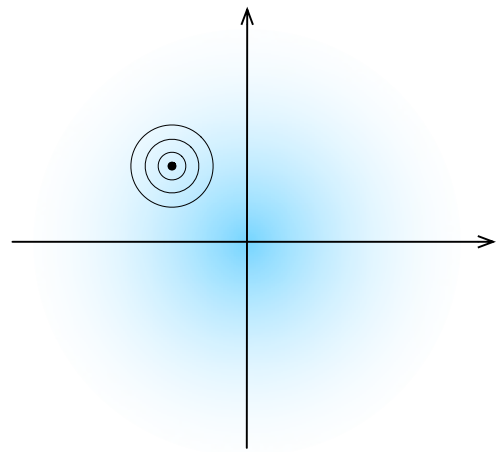
1. L'**encodeur** $f(\cdot;\mathbf{\Phi})$ encode une data $\mathbf{x}$ en deux vecteurs $\boldsymbol{\mu}_\mathbf{x}$ et $\mathbf{\Sigma}_\mathbf{x}$ de plus petite dimension.
2. Le **sampler** $s(\cdot;\boldsymbol{\mu}_\mathbf{x},\mathbf{\Sigma}_\mathbf{x})$ sample un point $\mathbf{z}$ dans l'espace latent selon la loi normale $\mathcal{N}(\boldsymbol{\mu}_\mathbf{x},\mathbf{\Sigma}_\mathbf{x})$.
3. Le **décodeur** $g(\cdot;\mathbf{\Theta})$ reconstruit $\mathbf{x}$ à partir de $\mathbf{z}$.



Variational Autoencoder (VAE)

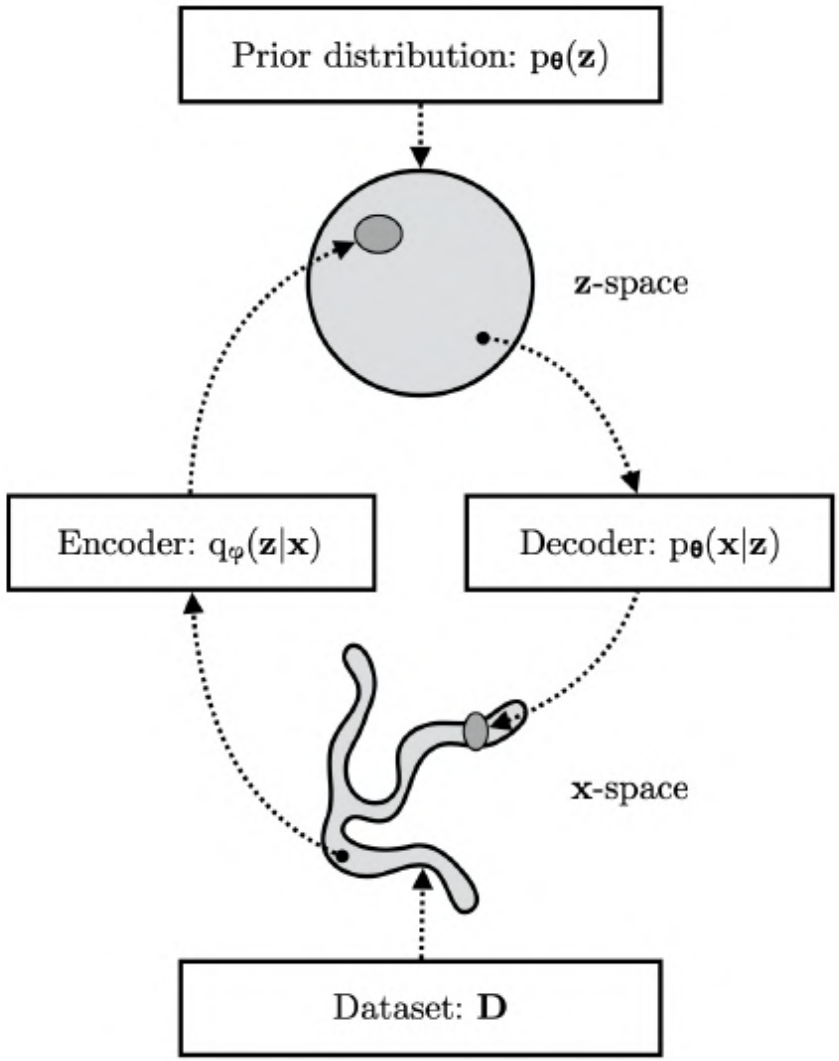
$q(\mathbf{X})$ —encoder f→ $q_{\mathbf{\Phi}}(\mathbf{Z} \mid \mathbf{X} = \mathbf{x}) \simeq \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$ where $\mu_{\mathbf{x}}, \mathbf{\Sigma}_{\mathbf{x}} := f(\mathbf{x}; \mathbf{\Phi})$





$p_{\mathbf{\Theta}}(\mathbf{X} \mid \mathbf{Z} = \mathbf{z})$ ←decoder g— $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$ where $p_{\mathbf{\Theta}}(\mathbf{Z}) \simeq \mathcal{N}(\mathbf{0}, \mathbf{I})$

Variational Autoencoder (VAE)



Variational Autoencoder (VAE)

- $q(\mathbf{x})$: Distribution empirique des data.
- $q_{\Phi}(\mathbf{z} \mid \mathbf{x})$: Distribution des data latentes étant donné la data originale $\mathbf{x}$. Fonction de $\mathbf{z}$ (variable $\mathbf{X}$ fixé à $\mathbf{x}$).

  $\rightarrow$ Dépend de l'encodeur $f(\cdot; \Phi)$.
- $p_{\Theta}(\mathbf{x} \mid \mathbf{z})$: Distribution des data reconstruites étant donné la data latente $\mathbf{z}$. Fonction de $\mathbf{x}$ (variable $\mathbf{Z}$ fixé à $\mathbf{z}$).

  $\rightarrow$ Dépend du décodeur $g(\cdot; \Theta)$.
- $p_{\Theta}(\mathbf{x})$: Distribution (marginale) des data reconstruites, induite par les distributions $p_{\Theta}(\mathbf{x} \mid \mathbf{z})$ pour tout $\mathbf{z}$.
  - La reconstruction s'effectue correctement si $q(\mathbf{x}) \simeq p_{\Theta}(\mathbf{x})$.

Variational Autoencoder (VAE)

- $q(\mathbf{x})$: Distribution empirique des data.
- $q_{\mathbf{\Phi}}(\mathbf{z} \mid \mathbf{x})$: Distribution des data latentes étant donné la data originale $\mathbf{x}$. Fonction de $\mathbf{z}$ (variable $\mathbf{X}$ fixé à $\mathbf{x}$).

  $\rightarrow$ Dépend de l'encodeur $f(\cdot; \mathbf{\Phi})$.
- $p_{\mathbf{\Theta}}(\mathbf{x} \mid \mathbf{z})$: Distribution des data reconstruites étant donné la data latente $\mathbf{z}$. Fonction de $\mathbf{x}$ (variable $\mathbf{Z}$ fixé à $\mathbf{z}$).

  $\rightarrow$ Dépend du décodeur $g(\cdot; \mathbf{\Theta})$.
- $p_{\mathbf{\Theta}}(\mathbf{x})$: Distribution (marginale) des data reconstruites, induite par les distributions $p_{\mathbf{\Theta}}(\mathbf{x} \mid \mathbf{z})$ pour tout $\mathbf{z}$.
- La reconstruction s'effectue correctement si $q(\mathbf{x}) \simeq p_{\mathbf{\Theta}}(\mathbf{x})$.

Variational Autoencoder (VAE)

- $q(\mathbf{x})$: Distribution empirique des data.
- $q_{\mathbf{\Phi}}(\mathbf{z} \mid \mathbf{x})$: Distribution des data latentes étant donné la data originale $\mathbf{x}$. Fonction de $\mathbf{z}$ (variable $\mathbf{X}$ fixé à $\mathbf{x}$).

  $\rightarrow$ Dépend de l'encodeur $f(\cdot; \mathbf{\Phi})$.
- $p_{\mathbf{\Theta}}(\mathbf{x} \mid \mathbf{z})$: Distribution des data reconstruites étant donné la data latente $\mathbf{z}$. Fonction de $\mathbf{x}$ (variable $\mathbf{Z}$ fixé à $\mathbf{z}$).

  $\rightarrow$ Dépend du décodeur $g(\cdot; \mathbf{\Theta})$.
- $p_{\mathbf{\Theta}}(\mathbf{x})$: Distribution (marginale) des data reconstruites, induite par les distributions $p_{\mathbf{\Theta}}(\mathbf{x} \mid \mathbf{z})$ pour tout $\mathbf{z}$.
- La reconstruction s'effectue correctement si $q(\mathbf{x}) \simeq p_{\mathbf{\Theta}}(\mathbf{x})$.

Variational Autoencoder (VAE)

- $q(\mathbf{x})$: Distribution empirique des data.
- $q_{\mathbf{\Phi}}(\mathbf{z} \mid \mathbf{x})$: Distribution des data latentes étant donné la data originale $\mathbf{x}$. Fonction de $\mathbf{z}$ (variable $\mathbf{X}$ fixé à $\mathbf{x}$).

  $\rightarrow$ Dépend de l'encodeur $f(\cdot; \mathbf{\Phi})$.
- $p_{\mathbf{\Theta}}(\mathbf{x} \mid \mathbf{z})$: Distribution des data reconstruites étant donné la data latente $\mathbf{z}$. Fonction de $\mathbf{x}$ (variable $\mathbf{Z}$ fixé à $\mathbf{z}$).

  $\rightarrow$ Dépend du décodeur $g(\cdot; \mathbf{\Theta})$.
- $p_{\mathbf{\Theta}}(\mathbf{x})$: Distribution (marginale) des data reconstruites, induite par les distributions $p_{\mathbf{\Theta}}(\mathbf{x} \mid \mathbf{z})$ pour tout $\mathbf{z}$.
  - La reconstruction s'effectue correctement si $q(\mathbf{x}) \simeq p_{\mathbf{\Theta}}(\mathbf{x})$.

Variational Autoencoder (VAE)

- $q(\mathbf{x})$: Distribution empirique des data.
- $q_{\mathbf{\Phi}}(\mathbf{z} \mid \mathbf{x})$: Distribution des data latentes étant donné la data originale $\mathbf{x}$. Fonction de $\mathbf{z}$ (variable $\mathbf{X}$ fixé à $\mathbf{x}$).

  $\rightarrow$ Dépend de l'encodeur $f(\cdot; \mathbf{\Phi})$.
- $p_{\mathbf{\Theta}}(\mathbf{x} \mid \mathbf{z})$: Distribution des data reconstruites étant donné la data latente $\mathbf{z}$. Fonction de $\mathbf{x}$ (variable $\mathbf{Z}$ fixé à $\mathbf{z}$).

  $\rightarrow$ Dépend du décodeur $g(\cdot; \mathbf{\Theta})$.
- $p_{\mathbf{\Theta}}(\mathbf{x})$: Distribution (marginale) des data reconstruites, induite par les distributions $p_{\mathbf{\Theta}}(\mathbf{x} \mid \mathbf{z})$ pour tout $\mathbf{z}$.

- La reconstruction s'effectue correctement si $q(\mathbf{x}) \simeq p_{\mathbf{\Theta}}(\mathbf{x})$.

KL-divergence et maximum likelihood

- **Rappel:** étant donné la distribution empirique $q(\mathbf{x})$, on cherche à *apprendre* la distribution $q(\mathbf{x})$, c'est à dire à trouver $p_{\mathbf{\Theta}}(\mathbf{x})$ telle que

$$q(\mathbf{x}) \simeq p_{\mathbf{\Theta}}(\mathbf{x})$$

- Pour cela, on minimiser la *divergence de Kullback–Leibler* entre $q(\mathbf{x})$ et $p_{\mathbf{\Theta}}(\mathbf{x})$ définie par:

$$\begin{aligned} D_{\mathrm{KL}}(q \,||\, p_{\mathbf{\Theta}}) &= \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p_{\mathbf{\Theta}}(\mathbf{x})} \, d\mathbf{x} \\ &= \int q(\mathbf{x}) \left(\log q(\mathbf{x}) - \log p_{\mathbf{\Theta}}(\mathbf{x}) \right) \, d\mathbf{x} \\ &= \mathbb{E}_{q(\mathbf{x})} \left[\log q(\mathbf{x}) \right] - \mathbb{E}_{q(\mathbf{x})} \left[\log p_{\mathbf{\Theta}}(\mathbf{x}) \right] \end{aligned}$$

KL-divergence et maximum likelihood

- **Rappel:** étant donné la distribution empirique $q(\mathbf{x})$, on cherche à *apprendre* la distribution $q(\mathbf{x})$, c'est à dire à trouver $p_{\mathbf{\Theta}}(\mathbf{x})$ telle que

$$q(\mathbf{x}) \simeq p_{\mathbf{\Theta}}(\mathbf{x})$$

- Pour cela, on minimiser la *divergence de Kullback–Leibler* entre $q(\mathbf{x})$ et $p_{\mathbf{\Theta}}(\mathbf{x})$ définie par:

$$\begin{aligned} D_{\mathrm{KL}}(q \,||\, p_{\mathbf{\Theta}}) &= \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p_{\mathbf{\Theta}}(\mathbf{x})} \, d\mathbf{x} \\ &= \int q(\mathbf{x}) \left(\log q(\mathbf{x}) - \log p_{\mathbf{\Theta}}(\mathbf{x}) \right) \, d\mathbf{x} \\ &= \mathbb{E}_{q(\mathbf{x})} \left[\log q(\mathbf{x}) \right] - \mathbb{E}_{q(\mathbf{x})} \left[\log p_{\mathbf{\Theta}}(\mathbf{x}) \right] \end{aligned}$$

VAE: loss function

- Ceci revient à maximiser la *log-likelihood* de $p_{\boldsymbol{\Theta}}(\mathbf{x})$ définie par:

$$\mathbb{E}_{q(\mathbf{x})}\left[\log p_{\boldsymbol{\Theta}}(\mathbf{x})\right] = \int q(\mathbf{x}) \log p_{\boldsymbol{\Theta}}(\mathbf{x})\, d\mathbf{x}$$

- Dans cette expression, le calcul de $p_{\boldsymbol{\Theta}}(\mathbf{x})$ s'effectue en marginalisant sur la variable latente, i.e.,

$$p_{\Theta}(\mathbf{x}) = \int p_{\Theta}(\mathbf{x}, \mathbf{z}) \mathbf{dz} = \int p_{\Theta}(\mathbf{x} \mid \mathbf{z})\, p_{\Theta}(\mathbf{z})\, d\mathbf{z}$$

- Mais cette expression n'est pas possible à optimiser efficacement (intractable), car son évaluation requiert un parcours de tout l'espace latent pour chaque $\mathbf{x}$.

VAE: loss function

- Ceci revient à maximiser la *log-likelihood* de $p_{\boldsymbol{\Theta}}(\mathbf{x})$ définie par:

$$\mathbb{E}_{q(\mathbf{x})}\left[\log p_{\boldsymbol{\Theta}}(\mathbf{x})\right] = \int q(\mathbf{x}) \log p_{\boldsymbol{\Theta}}(\mathbf{x})\, d\mathbf{x}$$

- Dans cette expression, le calcul de $p_{\boldsymbol{\Theta}}(\mathbf{x})$ s'effectue en marginalisant sur la variable latente, i.e.,

$$p_{\boldsymbol{\Theta}}(\mathbf{x}) = \int p_{\boldsymbol{\Theta}}(\mathbf{x}, \mathbf{z}) \mathbf{dz} = \int p_{\boldsymbol{\Theta}}(\mathbf{x} \mid \mathbf{z})\, p_{\boldsymbol{\Theta}}(\mathbf{z})\, d\mathbf{z}$$

- Mais cette expression n'est pas possible à optimiser efficacement (intractable), car son évaluation requiert un parcours de tout l'espace latent pour chaque $\mathbf{x}$.

VAE: loss function

- Ceci revient à maximiser la *log-likelihood* de $p_{\boldsymbol{\Theta}}(\mathbf{x})$ définie par:

$$\mathbb{E}_{q(\mathbf{x})}\left[\log p_{\boldsymbol{\Theta}}(\mathbf{x})\right] = \int q(\mathbf{x}) \log p_{\boldsymbol{\Theta}}(\mathbf{x})\, d\mathbf{x}$$

- Dans cette expression, le calcul de $p_{\boldsymbol{\Theta}}(\mathbf{x})$ s'effectue en marginalisant sur la variable latente, i.e.,

$$p_{\boldsymbol{\Theta}}(\mathbf{x}) = \int p_{\boldsymbol{\Theta}}(\mathbf{x}, \mathbf{z}) \mathbf{dz} = \int p_{\boldsymbol{\Theta}}(\mathbf{x} \mid \mathbf{z})\, p_{\boldsymbol{\Theta}}(\mathbf{z})\, d\mathbf{z}$$

- Mais cette expression n'est pas possible à optimiser efficacement (intractable), car son évaluation requiert un parcours de tout l'espace latent pour chaque $\mathbf{x}$.

VAE: loss function

- Par contre, on peut montrer que (cf. notes manuscrites):

$$\begin{aligned} \log p_{\boldsymbol{\Theta}}(\mathbf{x}) \quad &\geq \quad \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{z}|\mathbf{x})}\Big[\log p_{\boldsymbol{\Theta}}\big(\mathbf{x} \mid \mathbf{z}\big)\Big] \\ &- \quad \mathrm{D}_{\mathrm{KL}}\Big(\underbrace{q_{\boldsymbol{\Phi}}\big(\mathbf{z} \mid \mathbf{x}\big)}_{\sim\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}},\boldsymbol{\Sigma}_{\mathbf{x}})} \;\|\; \underbrace{p_{\boldsymbol{\Theta}}(\mathbf{z})}_{\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \Big) \end{aligned} \tag{1}$$

Le terme de droite se nomme *evidence lower bound (ELBO)*.

- En prenant l'espérance sur $q_{\boldsymbol{\Phi}}(\mathbf{x})$ à gauche et à droite, on a:

$$\begin{aligned} \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{x})}\left[\log p_{\boldsymbol{\Theta}}(\mathbf{x})\right] \quad &\geq \quad \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{z},\mathbf{x})}\Big[\log p_{\boldsymbol{\Theta}}\big(\mathbf{x} \mid \mathbf{z}\big)\Big] \\ &- \quad \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{x})}\Big[\mathrm{D}_{\mathrm{KL}}\Big(q_{\boldsymbol{\Phi}}\big(\mathbf{z} \mid \mathbf{x}\big) \,\|\, p_{\boldsymbol{\Theta}}(\mathbf{z})\Big)\Big] \end{aligned} \tag{2}$$

- Ainsi, maximiser les termes de droite dans (1) et (2) – ou alors minimiser leurs opposés – permet de maximiser les log-probabilité et log-likelihood de gauche, respectivement.

VAE: loss function

- Par contre, on peut montrer que (cf. notes manuscrites):

$$\begin{aligned} \log p_{\boldsymbol{\Theta}}(\mathbf{x}) \quad &\geq \quad \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{z}|\mathbf{x})}\Big[\log p_{\boldsymbol{\Theta}}\big(\mathbf{x} \mid \mathbf{z}\big)\Big] \\ &- \quad \mathrm{D}_{\mathrm{KL}}\Big(\underbrace{q_{\boldsymbol{\Phi}}\big(\mathbf{z} \mid \mathbf{x}\big)}_{\sim\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}},\boldsymbol{\Sigma}_{\mathbf{x}})} \;||\; \underbrace{p_{\boldsymbol{\Theta}}(\mathbf{z})}_{\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \Big) \end{aligned} \tag{1}$$

  Le terme de droite se nomme *evidence lower bound (ELBO)*.

- En prenant l'espérance sur $q_{\boldsymbol{\Phi}}(\mathbf{x})$ à gauche et à droite, on a:

$$\begin{aligned} \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{x})}\left[\log p_{\boldsymbol{\Theta}}(\mathbf{x})\right] \quad &\geq \quad \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{z},\mathbf{x})}\Big[\log p_{\boldsymbol{\Theta}}\big(\mathbf{x} \mid \mathbf{z}\big)\Big] \\ &- \quad \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{x})}\Big[\mathrm{D}_{\mathrm{KL}}\Big(q_{\boldsymbol{\Phi}}\big(\mathbf{z} \mid \mathbf{x}\big) \;||\; p_{\boldsymbol{\Theta}}(\mathbf{z})\Big)\Big] \end{aligned} \tag{2}$$

- Ainsi, maximiser les termes de droite dans (1) et (2) – ou alors minimiser leurs opposés – permet de maximiser les log-probabilité et log-likelihood de gauche, respectivement.

VAE: loss function

- Par contre, on peut montrer que (cf. notes manuscrites):

$$\begin{aligned} \log p_{\boldsymbol{\Theta}}(\mathbf{x}) \quad &\geq \quad \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{z}|\mathbf{x})}\Big[\log p_{\boldsymbol{\Theta}}\big(\mathbf{x} \mid \mathbf{z}\big)\Big] \\ &- \quad \mathrm{D}_{\mathrm{KL}}\Big(\underbrace{q_{\boldsymbol{\Phi}}\big(\mathbf{z} \mid \mathbf{x}\big)}_{\sim\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}},\boldsymbol{\Sigma}_{\mathbf{x}})} \;||\; \underbrace{p_{\boldsymbol{\Theta}}(\mathbf{z})}_{\sim\mathcal{N}(\mathbf{0},\mathbf{I})} \Big) \end{aligned} \tag{1}$$

  Le terme de droite se nomme *evidence lower bound (ELBO)*.

- En prenant l'espérance sur $q_{\boldsymbol{\Phi}}(\mathbf{x})$ à gauche et à droite, on a:

$$\begin{aligned} \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{x})}\left[\log p_{\boldsymbol{\Theta}}(\mathbf{x})\right] \quad &\geq \quad \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{z},\mathbf{x})}\Big[\log p_{\boldsymbol{\Theta}}\big(\mathbf{x} \mid \mathbf{z}\big)\Big] \\ &- \quad \mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{x})}\Big[\mathrm{D}_{\mathrm{KL}}\Big(q_{\boldsymbol{\Phi}}\big(\mathbf{z} \mid \mathbf{x}\big) \;||\; p_{\boldsymbol{\Theta}}(\mathbf{z})\Big)\Big] \end{aligned} \tag{2}$$

- Ainsi, maximiser les termes de droite dans (1) et (2) – ou alors minimiser leurs opposés – permet de maximiser les log-probabilité et log-likelihood de gauche, respectivement.

VAE: loss function

- On prend alors comme fonction de loss individuelle (cf. Eq. (1)):

$$\ell_i(\mathbf{\Phi}, \mathbf{\Theta}) = \underbrace{-\mathbb{E}_{q_{\mathbf{\Phi}}(\mathbf{z}|\mathbf{x})}\Big[\log p_{\mathbf{\Theta}}\big(\mathbf{x} \mid \mathbf{z}\big)\Big]}_{\text{reconstruction term}} + \underbrace{\mathrm{D}_{\mathrm{KL}}\Big(q_{\mathbf{\Phi}}\big(\mathbf{z} \mid \mathbf{x}\big) \,||\, p_{\mathbf{\Theta}}(\mathbf{z})\Big)}_{\text{regularization term}}$$

- Et comme fonction de loss collective (cf. Eq. (2)):

$$\mathcal{L}(\mathbf{\Phi}, \mathbf{\Theta}) = \underbrace{-\mathbb{E}_{q_{\mathbf{\Phi}}(\mathbf{z},\mathbf{x})}\Big[\log p_{\mathbf{\Theta}}\big(\mathbf{x} \mid \mathbf{z}\big)\Big]}_{\text{reconstruction term}} + \underbrace{\mathbb{E}_{q_{\mathbf{\Phi}}(\mathbf{x})}\Big[\mathrm{D}_{\mathrm{KL}}\Big(q_{\mathbf{\Phi}}\big(\mathbf{z} \mid \mathbf{x}\big) \,||\, p_{\mathbf{\Theta}}(\mathbf{z})\Big)\Big]}_{\text{regularization term}}$$

VAE: loss function

- On prend alors comme fonction de loss individuelle (cf. Eq. (1)):

$$\ell_i(\boldsymbol{\Phi}, \boldsymbol{\Theta}) = \underbrace{-\mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{z}|\mathbf{x})}\Big[\log p_{\boldsymbol{\Theta}}\big(\mathbf{x} \mid \mathbf{z}\big)\Big]}_{\text{reconstruction term}} + \underbrace{\mathrm{D}_{\mathrm{KL}}\Big(q_{\boldsymbol{\Phi}}\big(\mathbf{z} \mid \mathbf{x}\big) \,||\, p_{\boldsymbol{\Theta}}(\mathbf{z})\Big)}_{\text{regularization term}}$$

- Et comme fonction de loss collective (cf. Eq. (2)):

$$\mathcal{L}(\boldsymbol{\Phi}, \boldsymbol{\Theta}) = \underbrace{-\mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{z},\mathbf{x})}\Big[\log p_{\boldsymbol{\Theta}}\big(\mathbf{x} \mid \mathbf{z}\big)\Big]}_{\text{reconstruction term}} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\Phi}}(\mathbf{x})}\Big[\mathrm{D}_{\mathrm{KL}}\Big(q_{\boldsymbol{\Phi}}\big(\mathbf{z} \mid \mathbf{x}\big) \,||\, p_{\boldsymbol{\Theta}}(\mathbf{z})\Big)\Big]}_{\text{regularization term}}$$

VAE: loss function

- Pour un batch de data $\{\mathbf{x_i} : i = 1, \dots, B\}$, on a les *estimateurs* suivants de ces fonctions de loss:

$$\hat{\ell}_i(\mathbf{\Phi}, \mathbf{\Theta}) = -\log p_{\mathbf{\Theta}}\left(\mathbf{x_i} \mid \mathbf{z_i}\right) + \hat{\mathrm{D}}_{\mathrm{KL}}\left(q_{\mathbf{\Phi}}\left(\mathbf{z_i} \mid \mathbf{x_i}\right) \,||\, p_{\mathbf{\Theta}}(\mathbf{z_i})\right)$$

$$\hat{\mathcal{L}}(\mathbf{\Phi}, \mathbf{\Theta}) = \sum_{i=1}^{B}\left(-\log p_{\mathbf{\Theta}}\left(\mathbf{x_i} \mid \mathbf{z_i}\right) + \hat{\mathrm{D}}_{\mathrm{KL}}\left(q_{\mathbf{\Phi}}\left(\mathbf{z_i} \mid \mathbf{x_i}\right) \,||\, p_{\mathbf{\Theta}}(\mathbf{z_i})\right)\right)$$

où $\mathbf{z_i} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x_i}}, \boldsymbol{\Sigma}_{\mathbf{x_i}})$ avec $\boldsymbol{\mu}_{\mathbf{x_i}}, \boldsymbol{\Sigma}_{\mathbf{x_i}} := f(\mathbf{x_i}; \mathbf{\Phi})$

- Cette loss (collective) est minimisée par backpropagation.

Variational Autoencoder (VAE): training

- Mais le processus de sampling n'est pas différentiable, ce qui pose problème pour l'algorithme de backpropagation.
- Pour pallier cela, on recourt au "reparametrization trick" et au "log-var trick".
- Au lieu de sampler de $\mathbf{z}$ comme suit:

$$\begin{cases} \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}} := f(\mathbf{x}; \boldsymbol{\Phi}) \\ \mathbf{z} \sim q_{\boldsymbol{\Phi}}(\mathbf{Z} \mid \mathbf{X} = \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}) \end{cases}$$

on procède de la manière suivante:

$$\begin{cases} \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}} := f(\mathbf{x}; \boldsymbol{\Phi}) \\ \mathbf{z} = \boldsymbol{\mu}_{\mathbf{x}} + \boldsymbol{\sigma}_{\mathbf{x}} \odot \boldsymbol{\epsilon} \quad \text{où} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \text{et} \quad \boldsymbol{\sigma}_{\mathbf{x}} = \operatorname{diag}(\boldsymbol{\Sigma}_{\mathbf{x}})^{\frac{1}{2}} \end{cases}$$

Variational Autoencoder (VAE): training

- Mais le processus de sampling n'est pas différentiable, ce qui pose problème pour l'algorithme de backpropagation.
- Pour pallier cela, on recourt au "reparametrization trick" et au "log-var trick".
- Au lieu de sampler de $\mathbf{z}$ comme suit:

$$\begin{cases} \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}} := f(\mathbf{x}; \boldsymbol{\Phi}) \\ \mathbf{z} \sim q_{\boldsymbol{\Phi}}(\mathbf{Z} \mid \mathbf{X} = \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}) \end{cases}$$

on procède de la manière suivante:

$$\begin{cases} \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}} := f(\mathbf{x}; \boldsymbol{\Phi}) \\ \mathbf{z} = \boldsymbol{\mu}_{\mathbf{x}} + \boldsymbol{\sigma}_{\mathbf{x}} \odot \boldsymbol{\epsilon} \quad \text{où} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \text{et} \quad \boldsymbol{\sigma}_{\mathbf{x}} = \operatorname{diag}(\boldsymbol{\Sigma}_{\mathbf{x}})^{\frac{1}{2}} \end{cases}$$

Variational Autoencoder (VAE): training

- Mais le processus de sampling n'est pas différentiable, ce qui pose problème pour l'algorithme de backpropagation.
- Pour pallier cela, on recourt au "reparametrization trick" et au "log-var trick".
- Au lieu de sampler de $\mathbf{z}$ comme suit:

$$\begin{cases} \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}} := f(\mathbf{x}; \boldsymbol{\Phi}) \\ \mathbf{z} \sim q_{\boldsymbol{\Phi}}(\mathbf{Z} \mid \mathbf{X} = \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}) \end{cases}$$

on procède de la manière suivante:

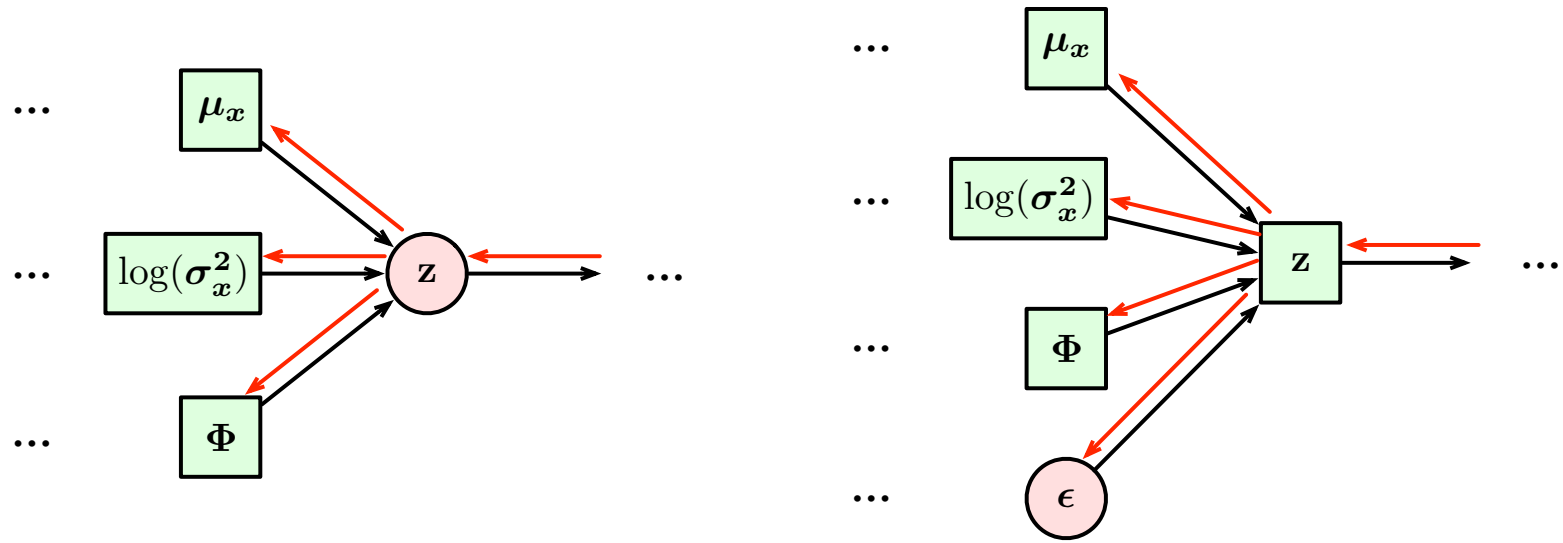
$$\begin{cases} \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}} := f(\mathbf{x}; \boldsymbol{\Phi}) \\ \mathbf{z} = \boldsymbol{\mu}_{\mathbf{x}} + \boldsymbol{\sigma}_{\mathbf{x}} \odot \boldsymbol{\epsilon} \quad \text{où} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \text{et} \quad \boldsymbol{\sigma}_{\mathbf{x}} = \operatorname{diag}(\boldsymbol{\Sigma}_{\mathbf{x}})^{\frac{1}{2}} \end{cases}$$

Variational Autoencoder (VAE): training

- En pratique, pour des raisons de stabilité, on apprend les vecteurs $\boldsymbol{\mu}_{\mathbf{x}}$ et $\log(\boldsymbol{\sigma}^{\mathbf{2}}_{\mathbf{x}})$ au lieu de $\boldsymbol{\mu}_{\mathbf{x}}$ et $\boldsymbol{\sigma}^{\mathbf{2}}_{\mathbf{x}}$. On a alors

$$\begin{cases} \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}} := f(\mathbf{x}; \boldsymbol{\Phi}) \\ \mathbf{z} = \boldsymbol{\mu}_{\mathbf{x}} + \boldsymbol{\sigma}_{\mathbf{x}} \odot \boldsymbol{\epsilon} = \boldsymbol{\mu}_{\mathbf{x}} + e^{\frac{1}{2}\log(\boldsymbol{\sigma}^{\mathbf{2}}_{\mathbf{x}})} \odot \boldsymbol{\epsilon} \quad \text{où} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \end{cases}$$

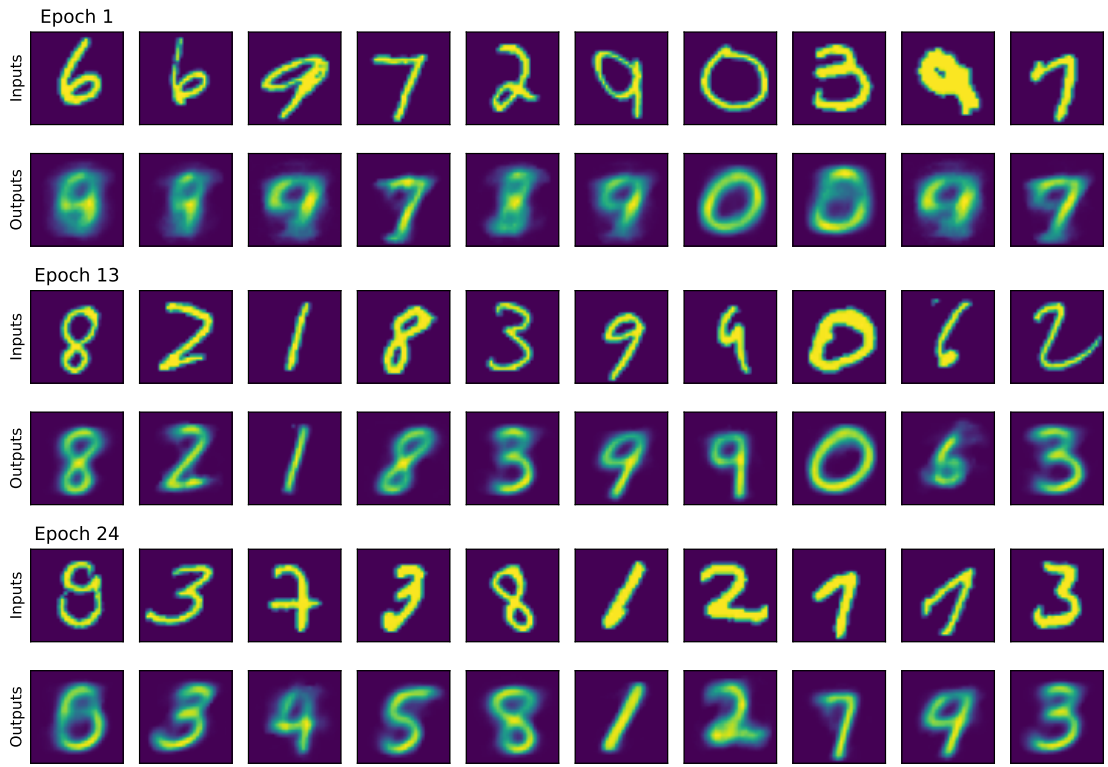
Variational Autoencoder (VAE): training



Computational graphs with and without the reparametrization trick. Green and red nodes are deterministic and stochastic nodes, respectively. Red arrows illustrate backpropagation.

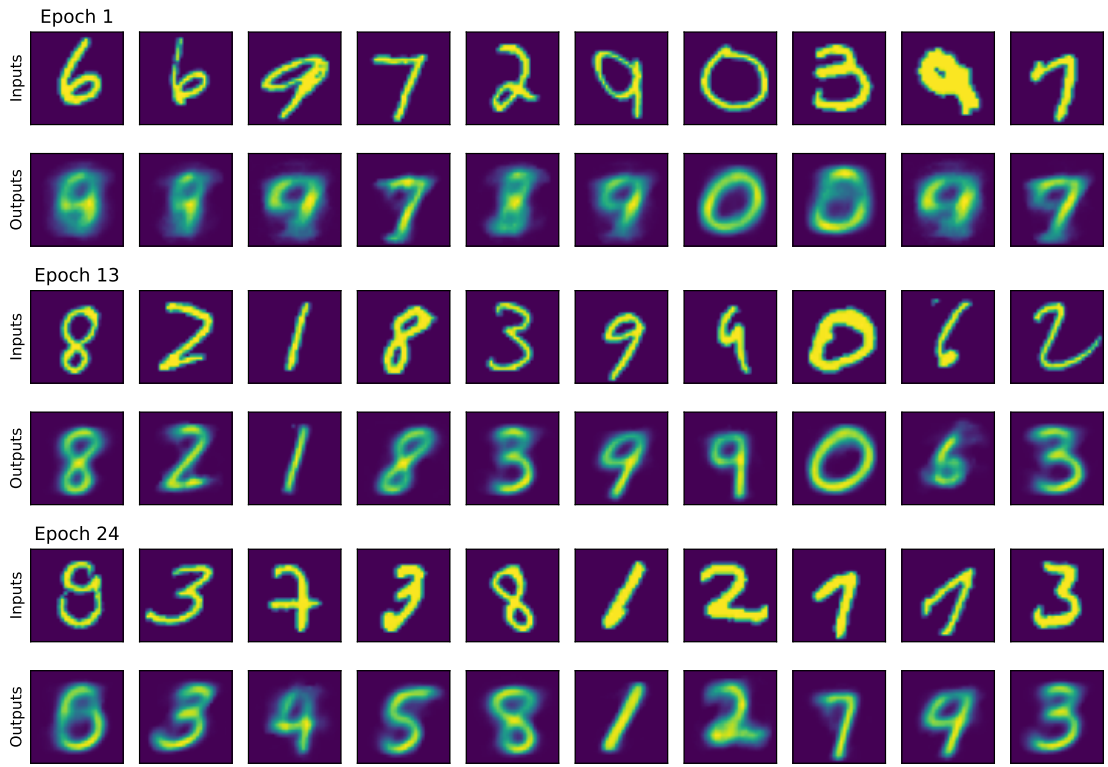
VAE linéaire: espace latent de dimension 2

- L'espace latent est réparti de manière bien plus gaussienne et les capacité de génération sont censées être améliorées (poids des termes de reconstruction et régularisation: 10 et 0.001).
- Pas si clair sur un espace latent de dim 2 (cf. slides suivants)...



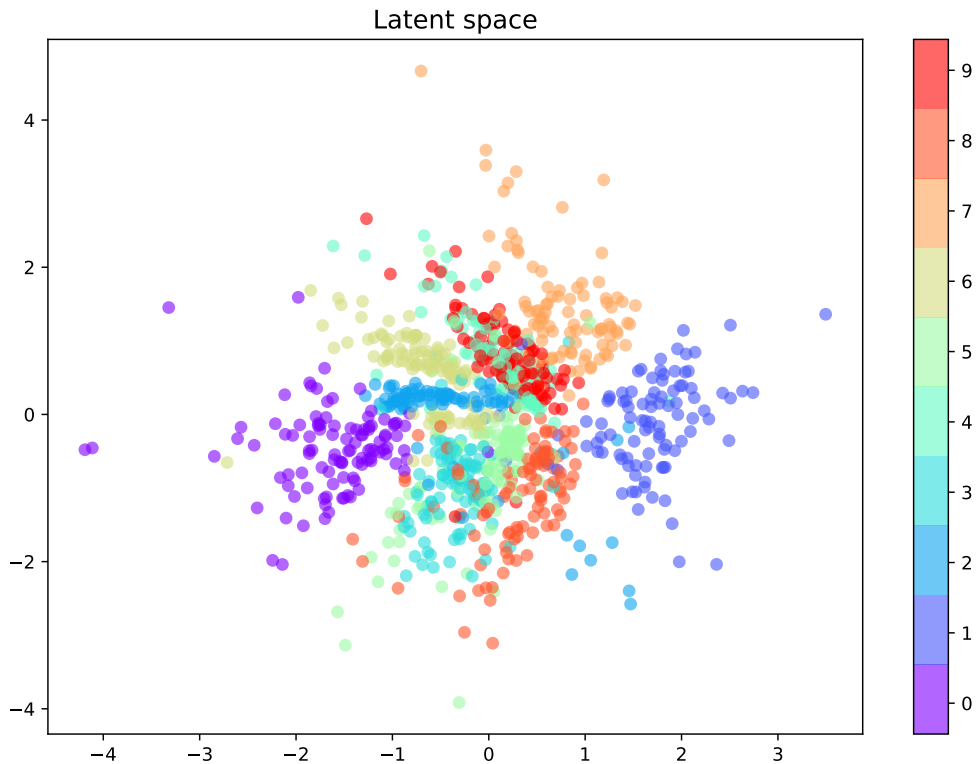
VAE linéaire: espace latent de dimension 2

- L'espace latent est réparti de manière bien plus gaussienne et les capacité de génération sont censées être améliorées (poids des termes de reconstruction et régularisation: 10 et 0.001).
- Pas si clair sur un espace latent de dim 2 (cf. slides suivants)...



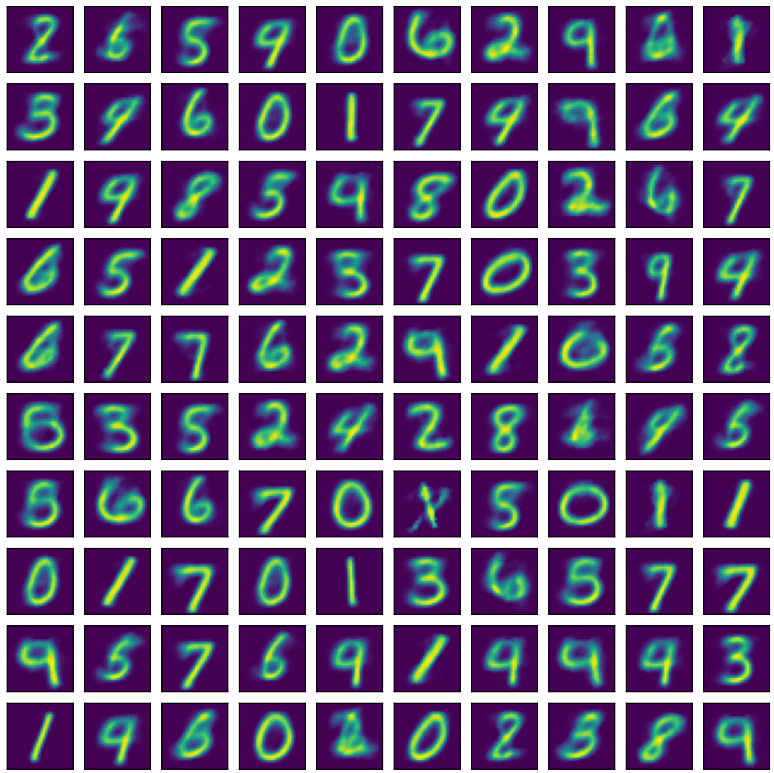
VAE linéaire: espace latent de dimension 2

- L'espace latent est réparti de manière bien plus gaussienne et les capacité de génération sont censées être améliorées (poids des termes de reconstruction et régularisation: 10 et 0.001).
- Pas si clair sur un espace latent de dim 2 (cf. slides suivants)...



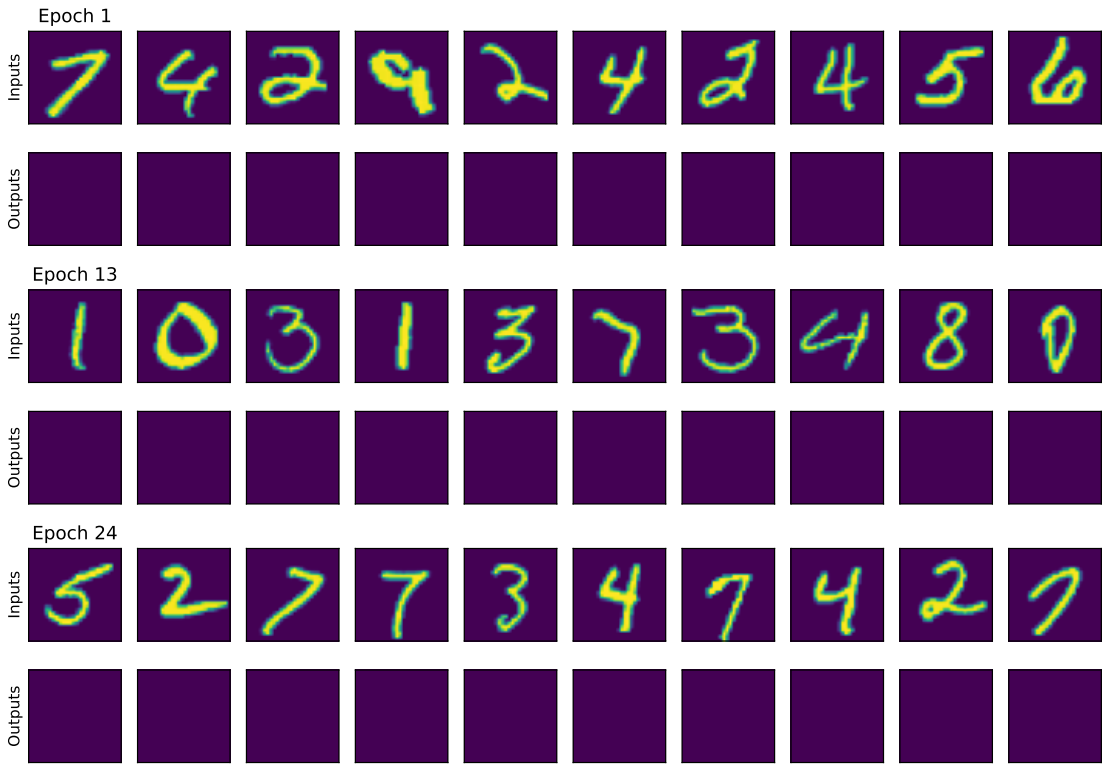
VAE linéaire: espace latent de dimension 2

- L'espace latent est réparti de manière bien plus gaussienne et les capacité de génération sont censées être améliorées (poids des termes de reconstruction et régularisation: 10 et 0.001).
- Pas si clair sur un espace latent de dim 2 (cf. slides suivants)...



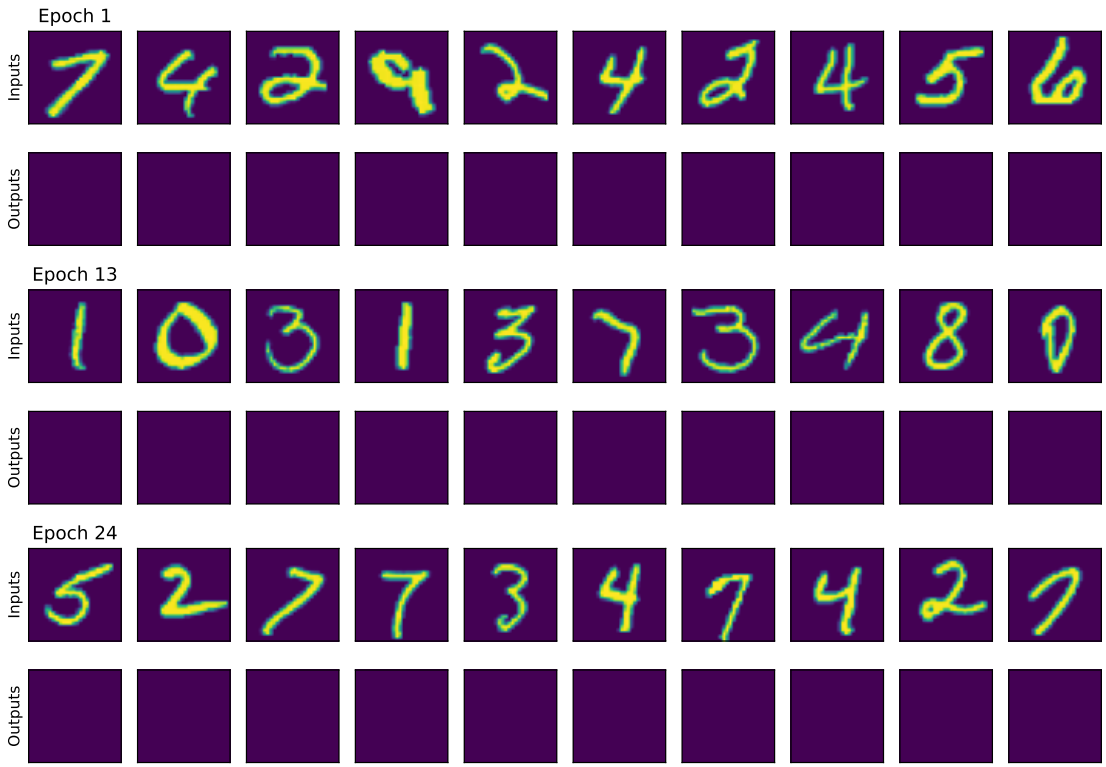
VAE linéaire: espace latent de dimension 2

- Si on annule le terme de reconstruction, les points de l'espace latent sont distribués de manière parfaitement normale...
- mais le réseau n'a rien appris à reconstruire.



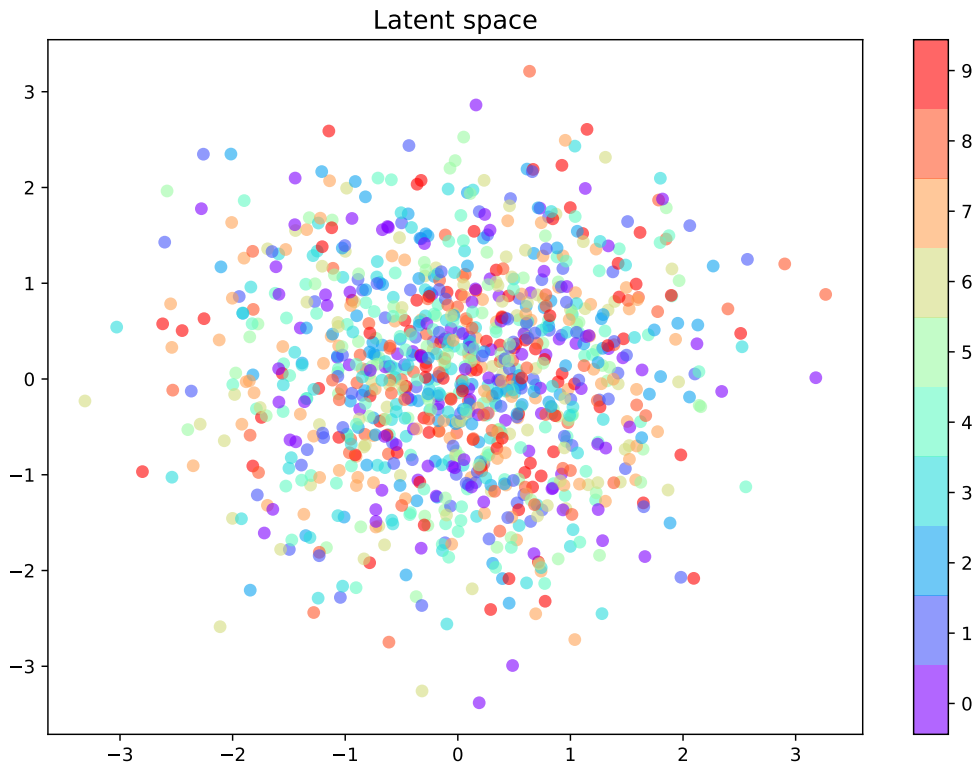
VAE linéaire: espace latent de dimension 2

- Si on annule le terme de reconstruction, les points de l'espace latent sont distribués de manière parfaitement normale...
- mais le réseau n'a rien appris à reconstruire.



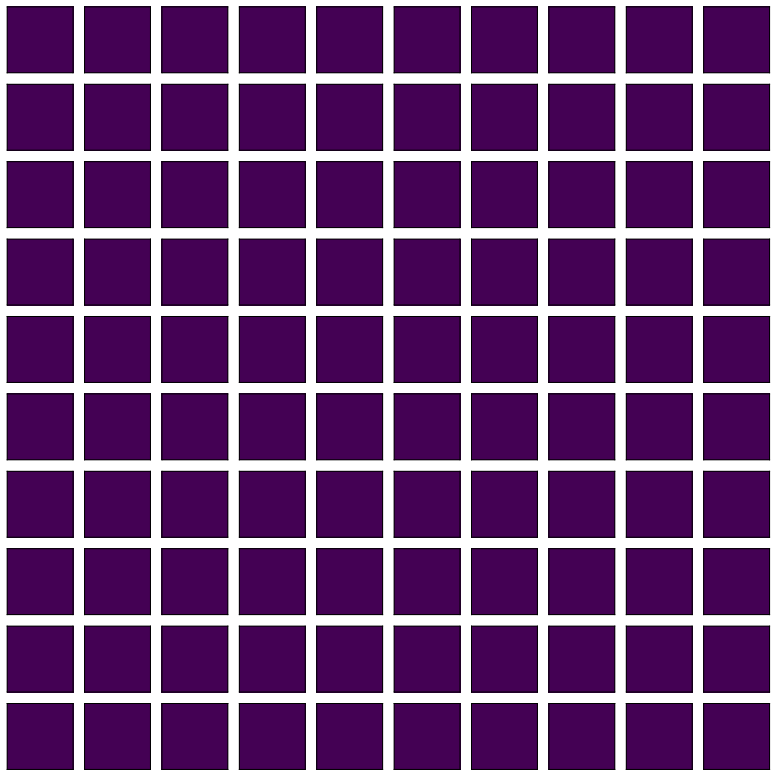
VAE linéaire: espace latent de dimension 2

- Si on annule le terme de reconstruction, les points de l'espace latent sont distribués de manière parfaitement normale...
- mais le réseau n'a rien appris à reconstruire.



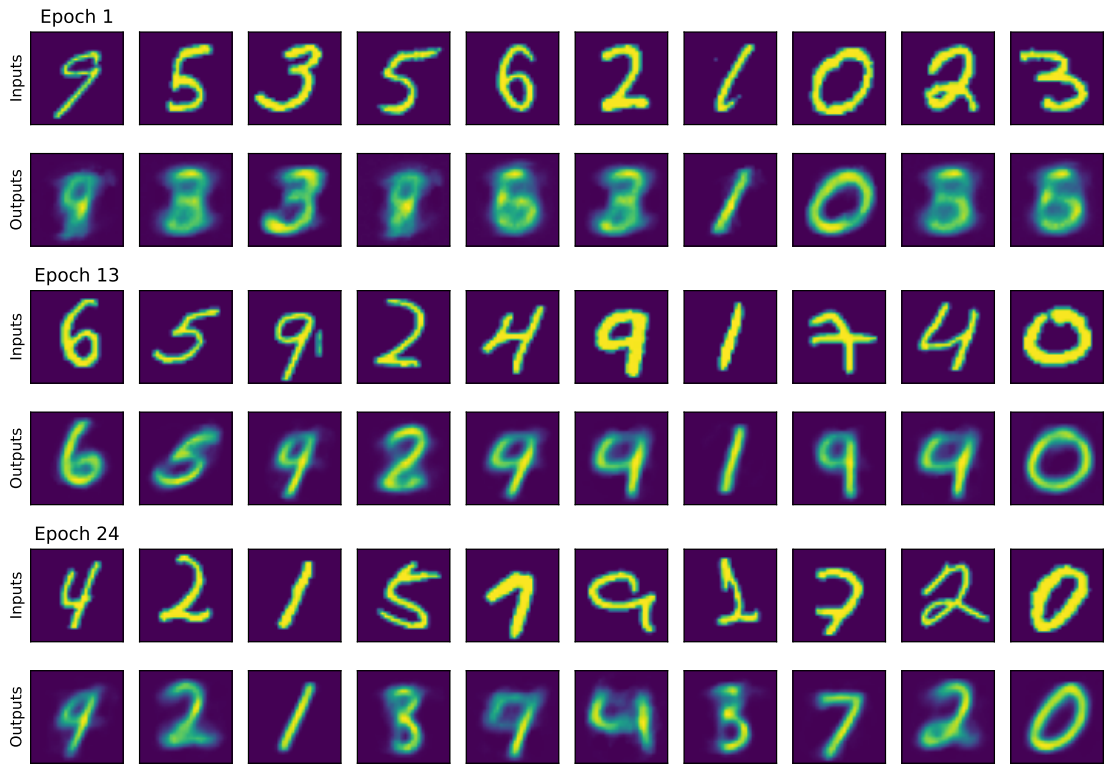
VAE linéaire: espace latent de dimension 2

- Si on annule le terme de reconstruction, les points de l'espace latent sont distribués de manière parfaitement normale...
- mais le réseau n'a rien appris à reconstruire.



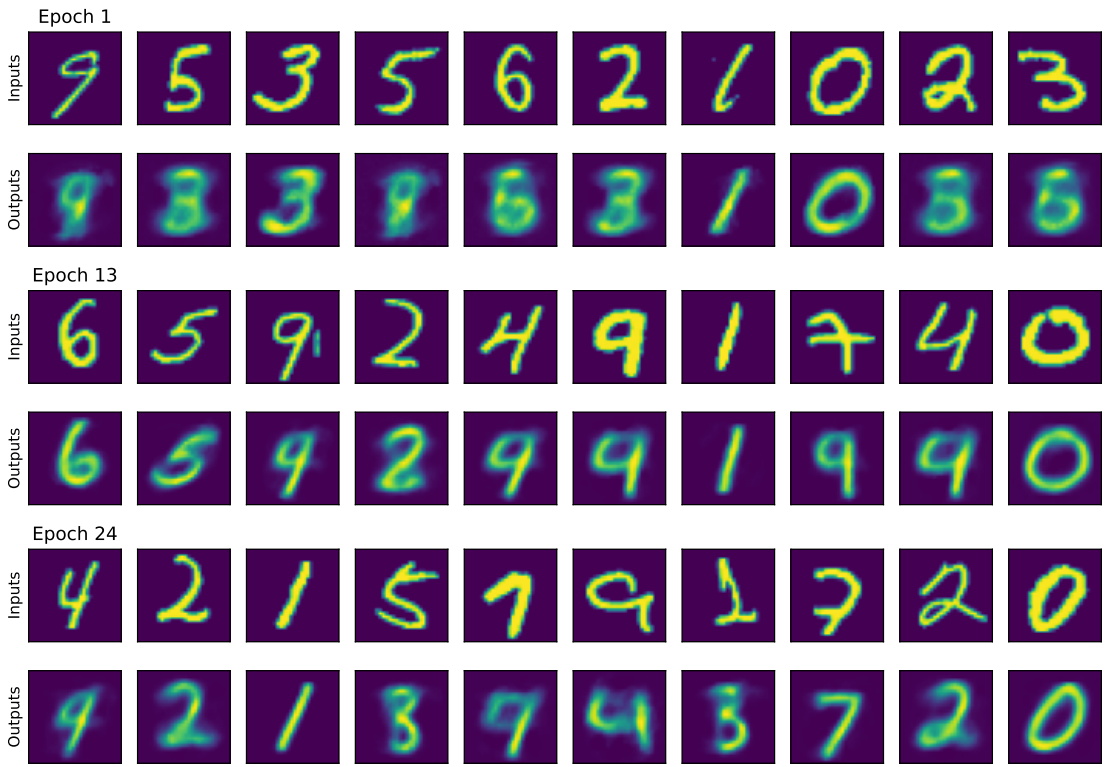
VAE linéaire: espace latent de dimension 2

- Si, au contraire, on annule le terme de régularisation, les points de l'espace latent ne sont plus distribués de manière normale...
- le réseau a appris à reconstruire des data, mais en samplant dans $\mathcal{N}(0,1)$, on génère des data assez similaires.



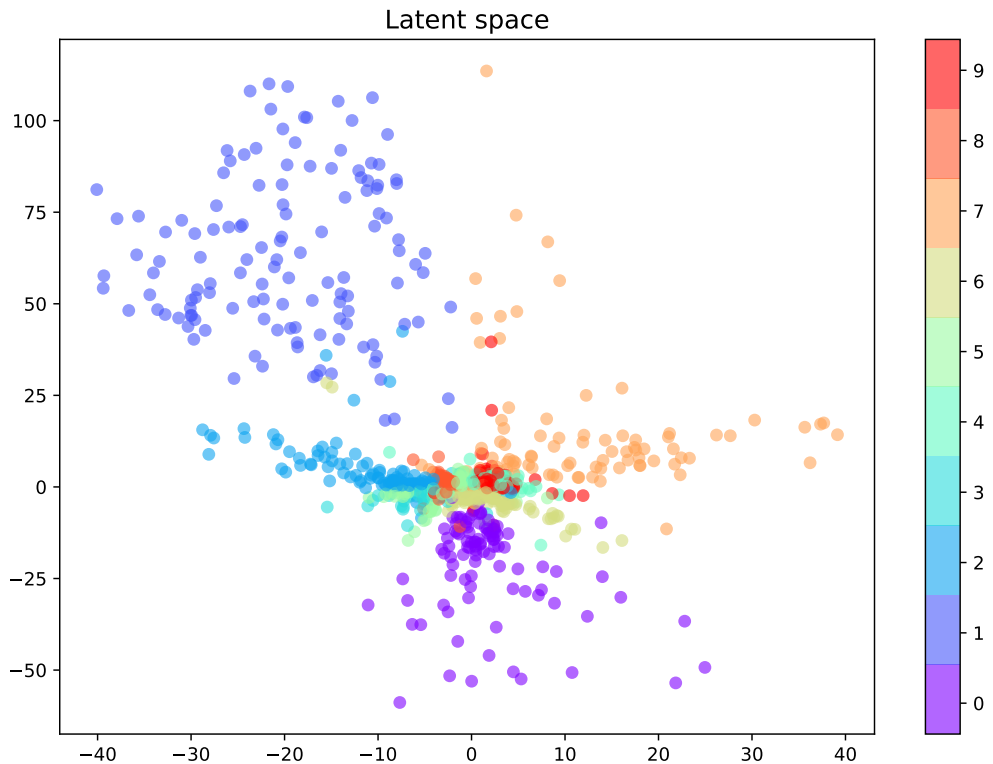
VAE linéaire: espace latent de dimension 2

- Si, au contraire, on annule le terme de régularisation, les points de l'espace latent ne sont plus distribués de manière normale...
- le réseau a appris à reconstruire des data, mais en samplant dans $\mathcal{N}(0,1)$, on génère des data assez similaires.



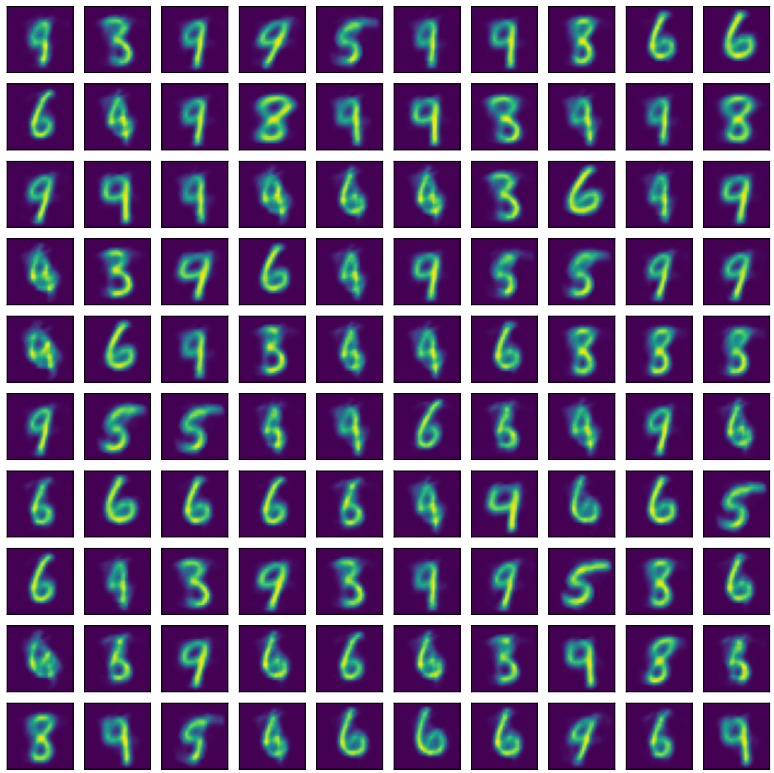
VAE linéaire: espace latent de dimension 2

- Si, au contraire, on annule le terme de régularisation, les points de l'espace latent ne sont plus distribués de manière normale...
- le réseau a appris à reconstruire des data, mais en samplant dans $\mathcal{N}(0,1)$, on génère des data assez similaires.



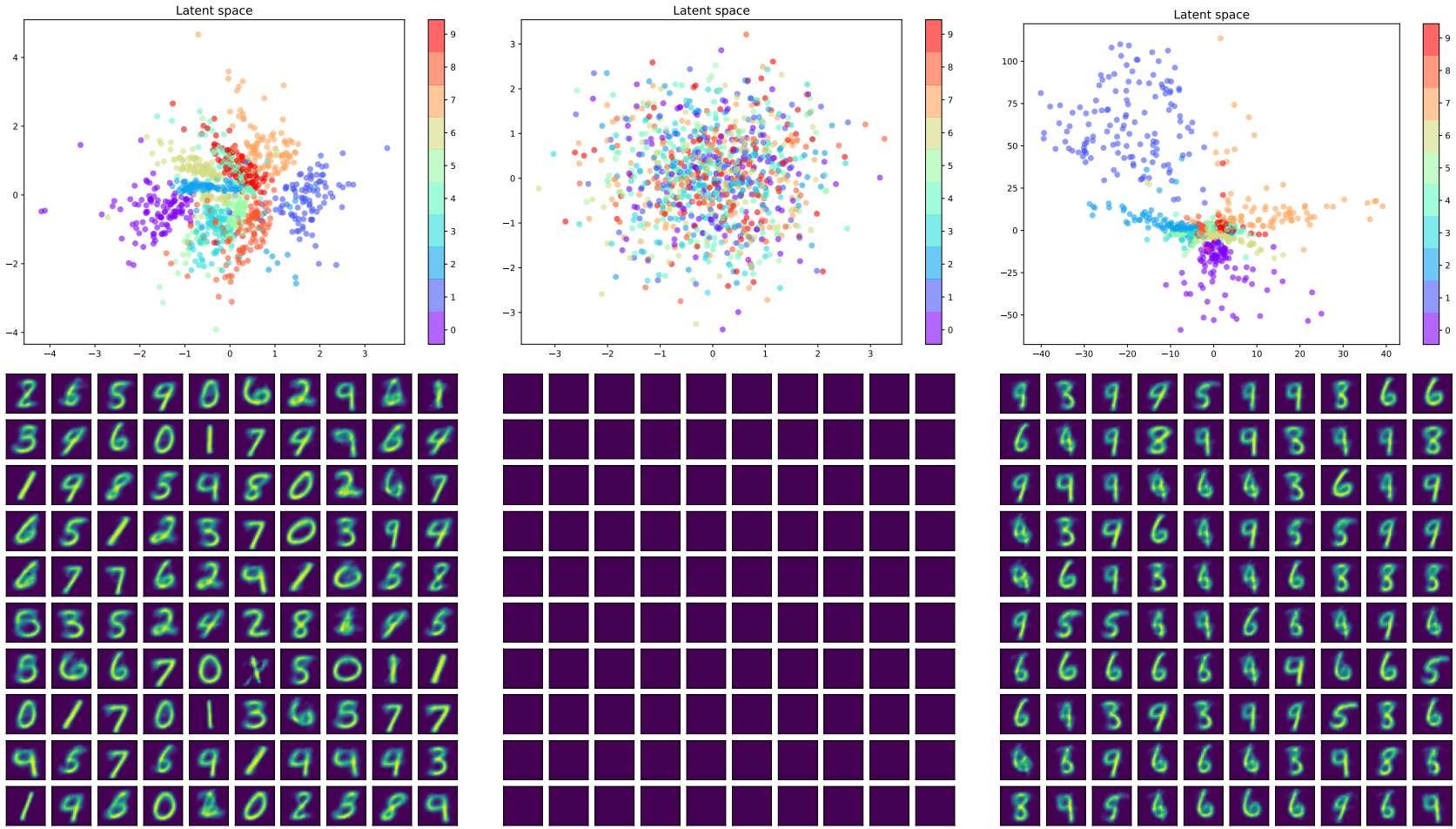
VAE linéaire: espace latent de dimension 2

- Si, au contraire, on annule le terme de régularisation, les points de l'espace latent ne sont plus distribués de manière normale...
- le réseau a appris à reconstruire des data, mais en samplant dans $\mathcal{N}(0, 1)$, on génère des data assez similaires.



VAE linéaire: espace latent de dimension 2

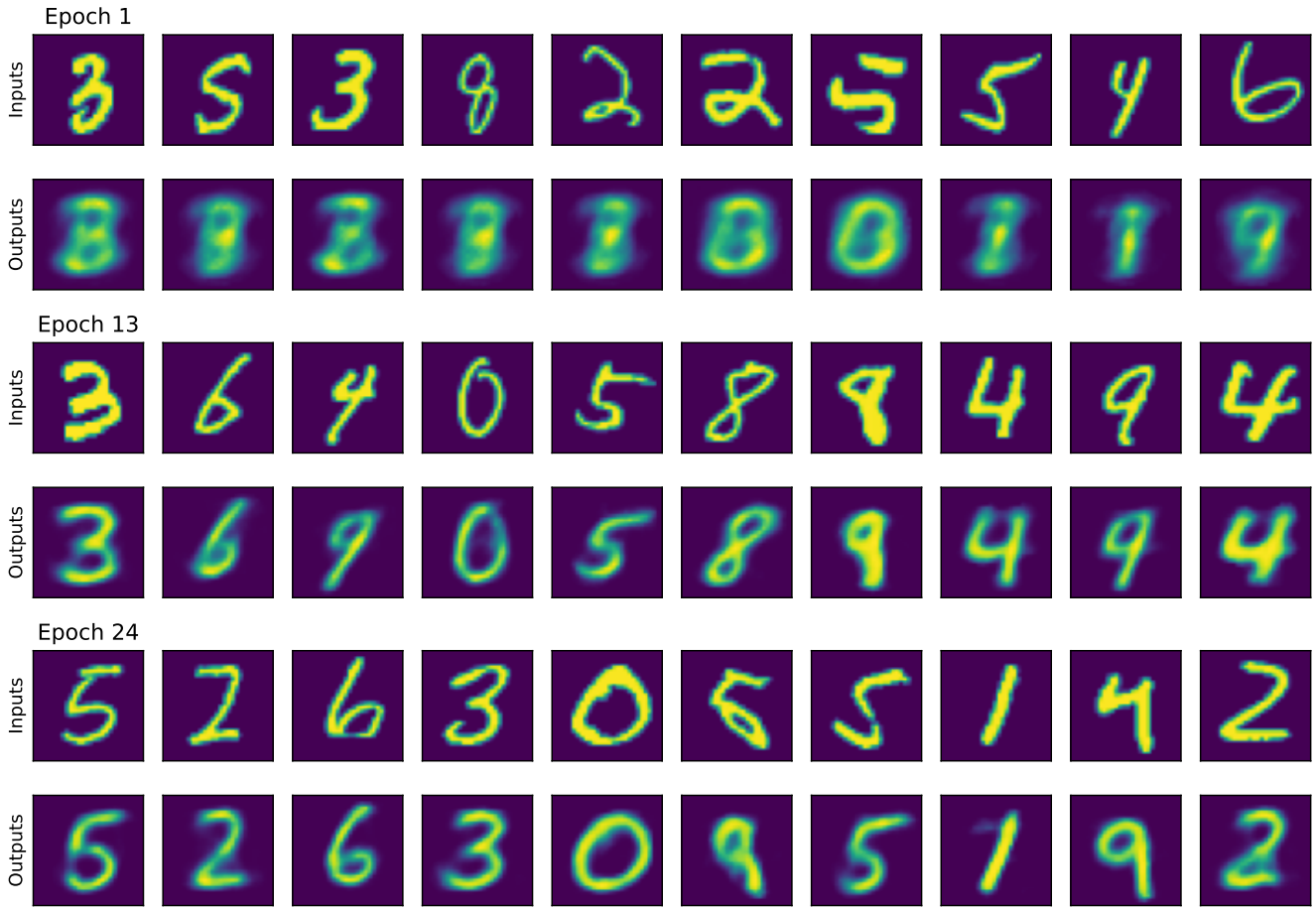
- Comparaison entre les trois situations.



Left: combination of MSE and KL losses; **Center:** only KL loss; **Right:** only MSE loss.

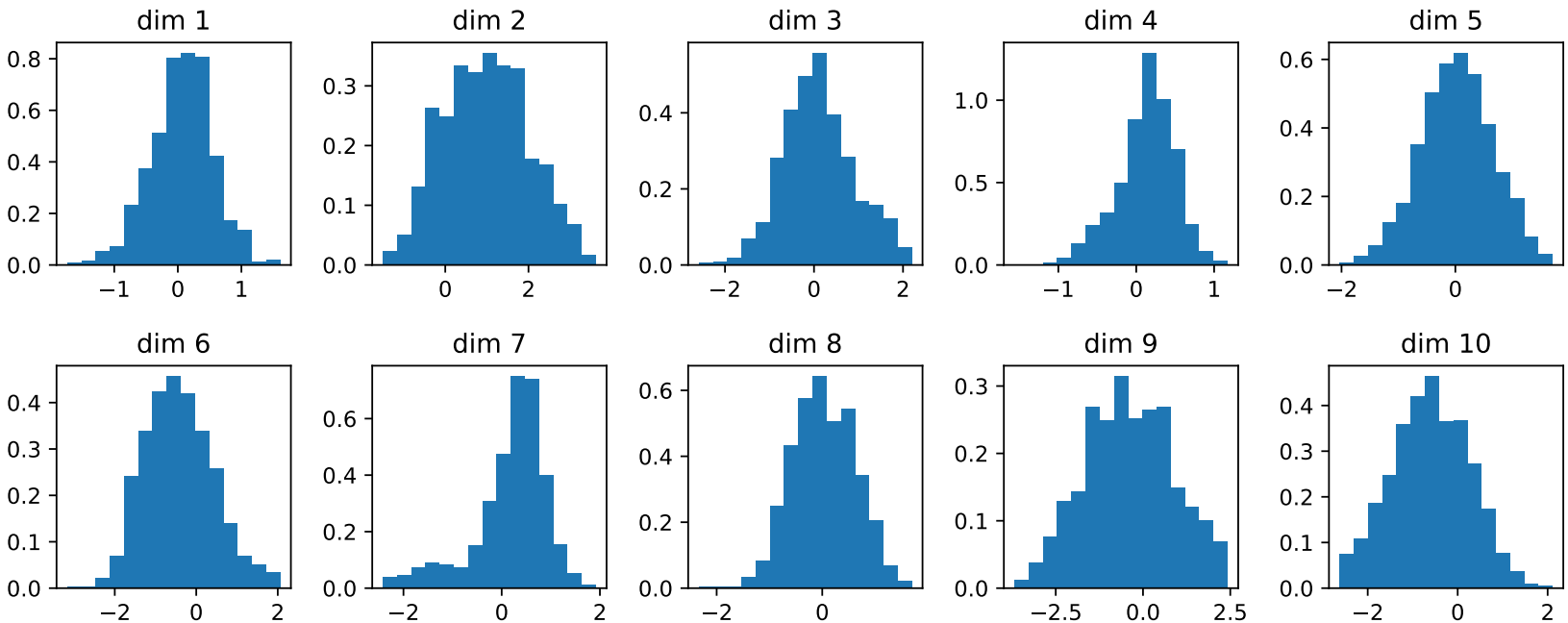
VAE linéaire: espace latent de dimension 10

- En augmentant la dimension de l'espace latent, la génération de data est de meilleure qualité.



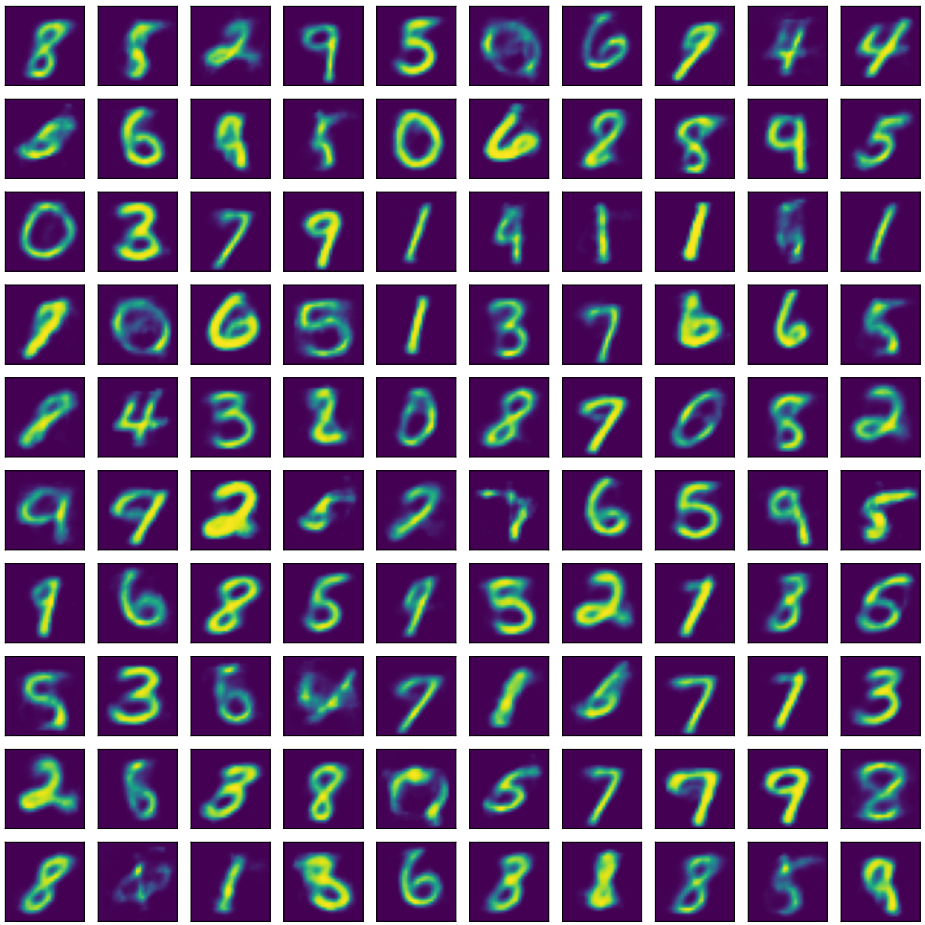
VAE linéaire: espace latent de dimension 10

- En augmentant la dimension de l'espace latent, la génération de data est de meilleure qualité.

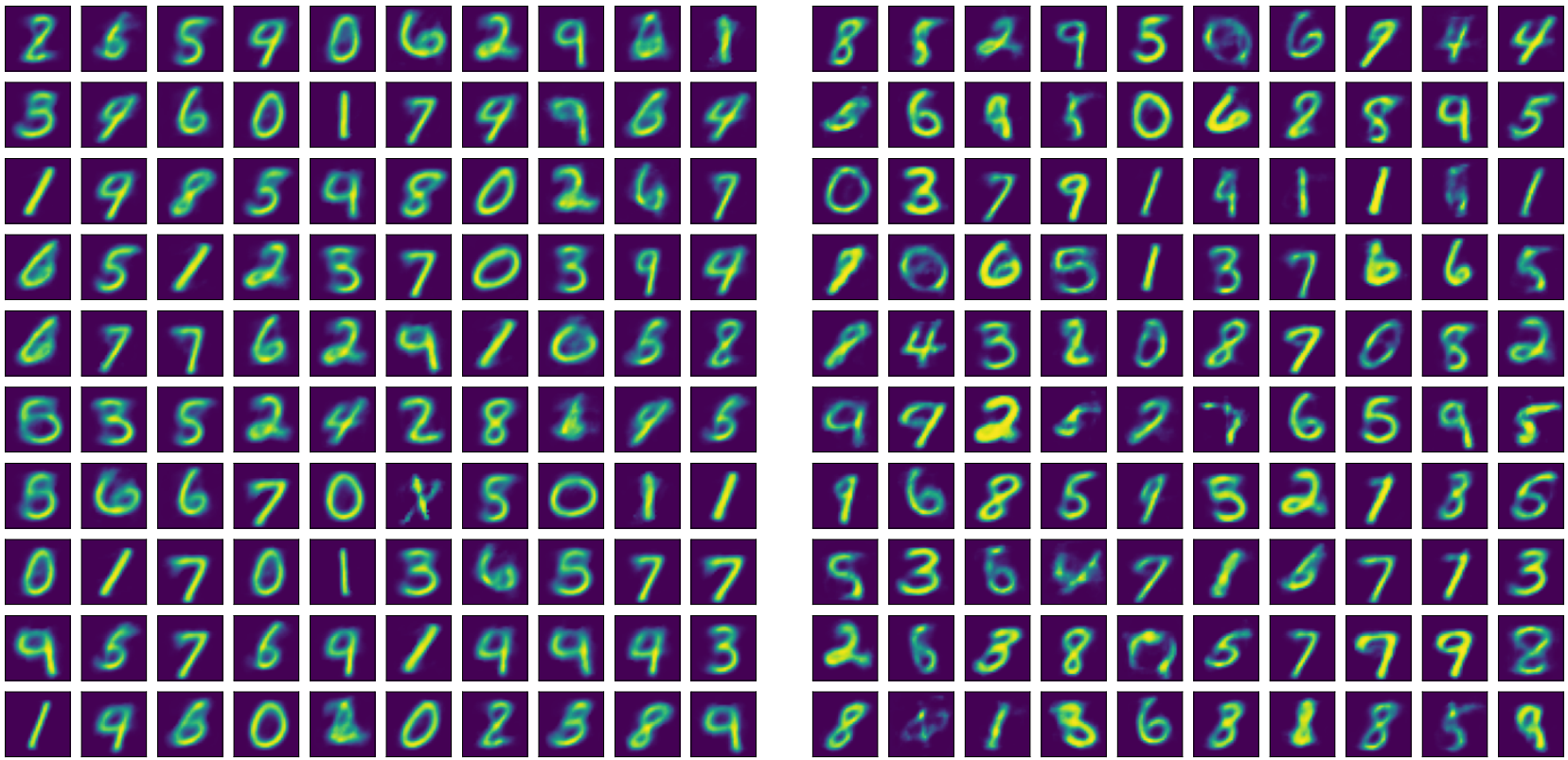


VAE linéaire: espace latent de dimension 10

- En augmentant la dimension de l'espace latent, la génération de data est de meilleure qualité.



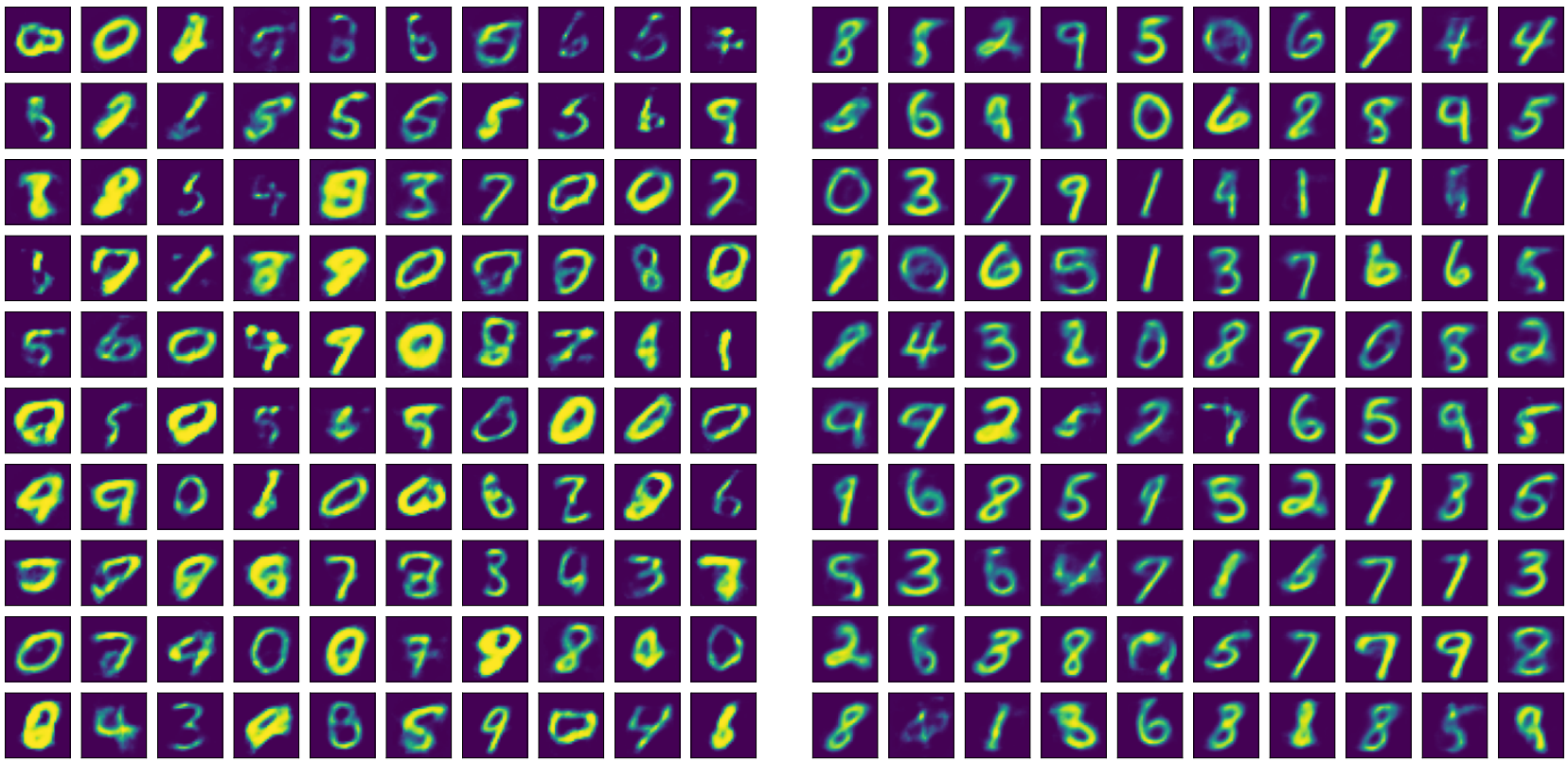
VAE linéaire: espaces latents de dimensions 2 et 10



Data generation of two VAEs with latent spaces of dimensions 2 (left) and 10 (right), respectively.

AE vs VAE: espace latent de dimension 10

- Génération de data: AE vs VAE.



Data generation of an AE (left) and a VAE (right) with latent space of dimension 10.

Conclusion

- Les autoencodeurs variationnels (VAEs) améliorent les capacité de génération de data par rapports aux autoencodeurs (AEs).
- L'ajout de stochasticité et l'imposition d'une distribution sur l'espace latent répond à des contraintes de continuité et de complétude sur ce espace.
- La continuité de l'espace latent rend les variables latentes interprétables: on comprend bien l'effet induit par la modification d'une variable alors que les autres restent fixes.

Conclusion

- Les autoencodeurs variationnels (VAEs) améliorent les capacité de génération de data par rapports aux autoencodeurs (AEs).
- L'ajout de stochasticité et l'imposition d'une distribution sur l'espace latent répond à des contraintes de continuité et de complétude sur ce espace.
- La continuité de l'espace latent rend les variables latentes interprétables: on comprend bien l'effet induit par la modification d'une variable alors que les autres restent fixes.

Conclusion

- Les autoencodeurs variationnels (VAEs) améliorent les capacité de génération de data par rapports aux autoencodeurs (AEs).
- L'ajout de stochasticité et l'imposition d'une distribution sur l'espace latent répond à des contraintes de continuité et de complétude sur ce espace.
- La continuité de l'espace latent rend les variables latentes interprétables: on comprend bien l'effet induit par la modification d'une variable alors que les autres restent fixes.

Bibliographie


Alexander Amini (2022).
Alexander amini: Youtube channel.


CNRS - Formation FIDLE (2022).
Cnrs - formation fidle: Youtube channel.


Fleuret, F. (2022).
Deep Learning Course.


Kingma, D. P. and Welling, M. (2014).
Auto-encoding variational bayes.
In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.


Kingma, D. P. and Welling, M. (2019).
An introduction to variational autoencoders.
Found. Trends Mach. Learn., 12(4):307–392.


Sebastian Raschka (2022).
Sebastian raschka: Youtube channel.


Stephen Odaibo (2020).
Stephen odaibo : Meduim post.