

Modèles Génératifs : Autoencodeur et autoencodeur variationnel

RCP211/bloc 2

Clément Rambour

Plan

1 Auto-encoder

2 Variational Auto-Encoder

Auto-encodeur - récap

Définition

Motivation : compression apprise avec le moins de pertes possible

Soit une variable aléatoire $\mathbf{X}$ à valeurs dans $\mathbb{R}^n$. Un réseau auto-encodeur modélise une fonction $\mathcal{H}$ telle que :

$$\|\mathcal{H} - \mathbf{x}\| \leq \epsilon$$

L'auto-encodeur se décompose en deux parties :

- Un encodeur $\mathcal{E} : \mathbb{R}^n \rightarrow \mathbb{R}^d$
- Un décodeur $\mathcal{D} : \mathbb{R}^d \rightarrow \mathbb{R}^n$

Auto-encodeur - récap

L'auto-encodeur apprend à reconstruire l'entrée. On cherche donc les poids θ tels que :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathcal{E} \circ \mathcal{D}(\mathbf{x}) - \mathbf{x}\|$$

avec $\mathcal{L}$ la fonction de coût *eg.* erreur quadratique, erreur absolue, cross-entropy...

Convolutional Auto-encoder

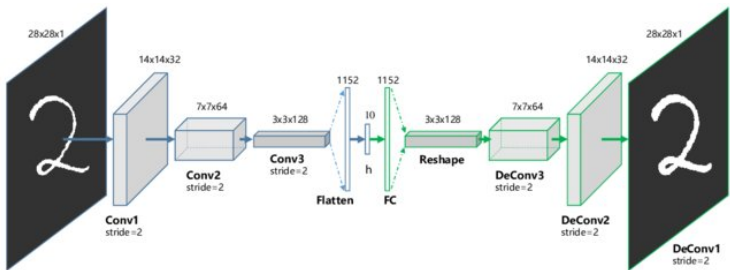


FIGURE – Guo et al. ICONIP 2017

Clustering

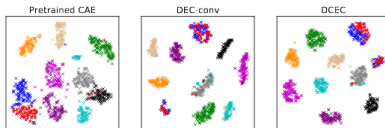


FIGURE – Guo et al. ICONIP 2017

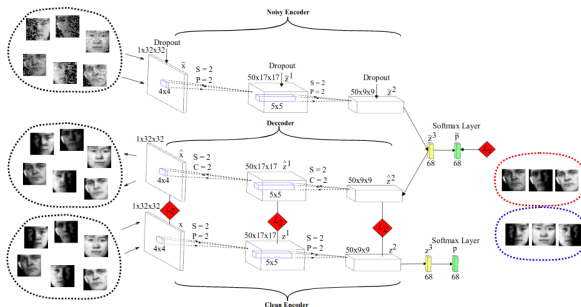


FIGURE – Dizaji et al. ICCV 2017

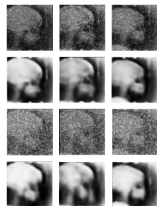
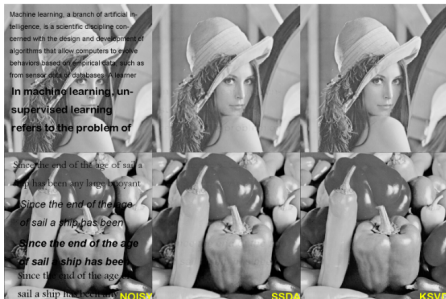


FIGURE – Gondara ICDM 2016



Estimation de pose

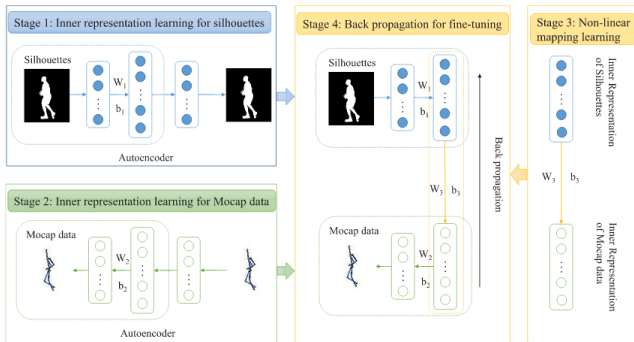


FIGURE – Hong et al. ITIP 2015

AE comme modèle génératif?

Motivation : Utiliser le decoder pour générer des données

- Tirer un vecteur de taille d aléatoire
- Reconstruire l'image correspondant à ce vecteur

AE comme modèle génératif?

Motivation : Utiliser le decoder pour générer des données

- Tirer un vecteur de taille d aléatoire
- Reconstruire l'image correspondant à ce vecteur

À quoi peut-on s'attendre ?

- 1 Une données synthétique crédible
- 2 Généralement une mauvaise reconstruction

AE comme modèle génératif?

Motivation : Utiliser le decoder pour générer des données

- Tirer un vecteur de taille d aléatoire
- Reconstruire l'image correspondant à ce vecteur

À quoi peut-on s'attendre ?

- 1 Une données synthétique crédible
- 2 Généralement une mauvaise reconstruction

Pourquoi ?

- 1 on ne sait pas où tirer
- 2 on a pas de garantie que la distribution de l'espace latent soit continue
- 3 la dimension de l'espace latente est trop grande
- 4 le modèle n'est pas assez complexe

Plan

1 Auto-encoder

2 Variational Auto-Encoder

Scénario

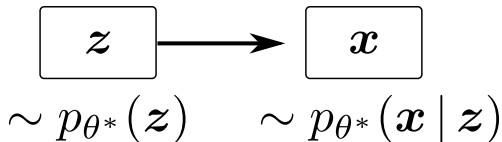
- Soit un jeu de données $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ de N variables indépendantes et identiquement distribuées (*iid*) correspondant aux réalisations d'une variable aléatoire $\mathbf{x} \in \mathcal{X}$.
- Soit $\mathbf{x} \sim p_{\mathcal{D}}$, le processus génératif

Scénario

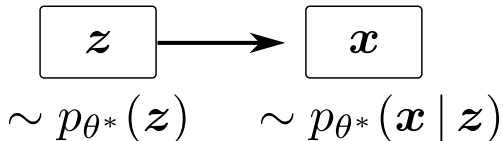
- Soit un jeu de données $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ de N variables indépendantes et identiquement distribuées (*iid*) correspondant aux réalisations d'une variable aléatoire $\mathbf{x} \in \mathcal{X}$.
- Soit $\mathbf{x} \sim p_{\mathcal{D}}$, le processus génératif
- On cherche à représenter le processus génératif $p_{\mathcal{D}}$ par un modèle appris
- $\mathbf{x}$ est générée par un processus aléatoire impliquant une variable aléatoire continue $\mathbf{z} \in \mathcal{Z}$.

Scénario

- Soit un jeu de données $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ de N variables indépendantes et identiquement distribuées (*iid*) correspondant aux réalisations d'une variable aléatoire $\mathbf{x} \in \mathcal{X}$.
 - Soit $\mathbf{x} \sim p_{\mathcal{D}}$, le processus génératif
 - On cherche à représenter le processus génératif $p_{\mathcal{D}}$ par un modèle appris
 - $\mathbf{x}$ est générée par un processus aléatoire impliquant une variable aléatoire continue $\mathbf{z} \in \mathcal{Z}$.
- 1 $\mathbf{z}_i$ est généré d'après une distribution *a priori* $p_{\theta^*}(\mathbf{z})$
 - 2 $\mathbf{x}_i$ est généré suivant la probabilité conditionnelle $p_{\theta^*}(\mathbf{x}|\mathbf{z})$



Objectif



θ^* et les variables latentes $\{z_1, \dots, z_N\}$ sont **inconnues**

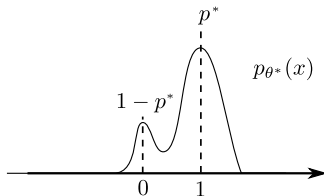
On cherche une paramétrisation $\theta^* \in \Theta$ permettant de générer x à partir de z . Une fonction objectif possible : maximiser la *log*-vraisemblance sur $\mathcal{D}$:

$$\hat{\theta} = \operatorname{argmax}_{\theta} \mathbb{E}_{p_{\mathcal{D}}} [\log p_{\theta}(x)]$$

ie. maximiser la vraisemblance marginal d'une donnée $x \sim p_{\theta}$ sur l'ensemble des observation.

Exemple simpliste

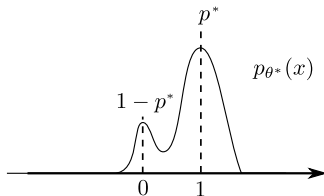
$$p_{\theta^*}(z) = \text{Bern}(p) \quad p_{\theta^*}(x | z) = \mathcal{N}(z, \sigma)$$



$$\theta = \{p\}$$

Exemple simpliste

$$p_{\theta^*}(z) = \text{Bern}(p) \quad p_{\theta^*}(x | z) = \mathcal{N}(z, \sigma)$$



$$\theta = \{p\}$$

$$p_{\theta}(x) = (1 - p)p_{\theta}(x|0, \sigma) + p p_{\theta}(x|1, \sigma)$$

→ sampling

Cas général

- $p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{z})p_{\theta}(\mathbf{x}|\mathbf{z})d\mathbf{z}$: pas d'expression analytique, pas non plus pour son gradient
- *posterior* $p_{\theta}(\mathbf{z}|\mathbf{x})$: pas d'expression analytique
- Le jeu de données est trop large pour échantillonner dans $\mathcal{X}$
- *Généralement* l'espace latent $\mathcal{Z}$ est trop grand pour être échantillonné efficacement

Cas général

- $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})d\mathbf{z}$: pas d'expression analytique, pas non plus pour son gradient
- *posterior* $p_\theta(\mathbf{z}|\mathbf{x})$: pas d'expression analytique
- Le jeu de données est trop large pour échantillonner dans $\mathcal{X}$
- *Généralement* l'espace latent $\mathcal{Z}$ est trop grand pour être échantillonné efficacement

Intuition : On approche le *posterior* par une distribution $q_\phi(\mathbf{z}|\mathbf{x})$ paramétrée par ϕ . Sous réserve que $q_\phi(\mathbf{z}|\mathbf{x})$ soit bien construite, cette distribution (*connue*) donne accès à des valeurs de $\mathbf{z}$ étant *vraisemblablement* à l'origine d'un $\mathbf{x}$.

Kullback-Leibler Divergence

Comment choisir $q_\phi(\mathbf{z}|\mathbf{x})$?

→ on souhaite avoir une distribution la plus proche possible du vrai
posterior $p_\theta(\mathbf{z}|\mathbf{x})$

Kullback-Leibler Divergence

Comment choisir $q_\phi(\mathbf{z}|\mathbf{x})$?

→ on souhaite avoir une distribution la plus proche possible du vrai *posterior* $p_\theta(\mathbf{z}|\mathbf{x})$

$$\rightarrow \hat{\phi} = \underset{\phi}{\operatorname{argmin}} \operatorname{KL} (q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x}))$$

Kullback-Leibler Divergence

Définition

La divergence de Kullback-Leigler noté KL est donnée par :

$$\text{KL} (q(\mathbf{x})||p(\mathbf{x})) = \int \log \frac{q(\mathbf{x})}{p(\mathbf{x})} q(\mathbf{x}) d\mathbf{x}$$

$$\text{KL} (q(\mathbf{x})||p(\mathbf{x})) = \mathbb{E}_{q(\mathbf{x})}[\log q(\mathbf{x}) - \log p(\mathbf{x})]$$

Kullback-Leibler Divergence

Objectif

$$\phi = \underset{\phi}{\operatorname{argmin}} \operatorname{KL} (q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z}|\mathbf{x}))$$

$$\begin{aligned} \operatorname{KL} (q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log q_{\phi}(\mathbf{z}|\mathbf{x}) - \log p_{\theta}(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log q_{\phi}(\mathbf{z}|\mathbf{x}) - \log p_{\theta}(\mathbf{x}|\mathbf{z}) - \log p_{\theta}(\mathbf{x})] + \log p_{\theta}(\mathbf{x}) \\ &= -\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] + \operatorname{KL} (q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z})) + \log p_{\theta}(\mathbf{x}) \end{aligned}$$

Evidence Lower Bound

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

Définition

L'*evidence lower bound* (ELBO) est définie par

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

Minimiser la divergence de Kullback-Leibler $\text{KL} (q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x}))$ revient à **maximiser** l'ELBO $\mathcal{L}(\theta, \phi; \mathbf{x})$.

ELBO et fonction objectif

On cherche θ tel que :

$$\hat{\theta} = \operatorname{argmax}_{\theta} \mathbb{E}_{p_{\mathcal{D}}} [\log p_{\theta}(\mathbf{x})]$$

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

De plus la KL-divergence est positive donc

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

Maximiser l'ELBO revient donc à **maximiser** la vraisemblance

Evidence Lower Bound

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{Espérance de la vraisemblance}} - \underbrace{\text{KL} (q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}))}_{\text{écart au } \textit{prior}}$$

Evidence Lower Bound

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{Espérance de la vraisemblance}} - \underbrace{\text{KL} (q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}))}_{\text{écart au prior}}$$

Quel fonction pour $q_\phi(\mathbf{z}|\mathbf{x})$?

→ Un choix naturel : loi normale

→ $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma} \mathbf{I})$

Evidence Lower Bound

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})]}_{\text{Espérance de la vraisemblance}} - \underbrace{\text{KL} (q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z}))}_{\text{écart au prior}}$$

Quel fonction pour $q_{\phi}(\mathbf{z}|\mathbf{x})$?

→ Un choix naturel : loi normale

→ $q_{\phi}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma} \mathbf{I})$

Quel *prior* $p_{\theta}(\mathbf{z})$?

→ On cherche un espace latent structuré

→ Encore une fois : $p_{\theta}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$

Evidence Lower Bound

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})]}_{\text{Espérance de la vraisemblance}} - \underbrace{\text{KL} (q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z}))}_{\text{écart au prior}}$$

Quel fonction pour $q_{\phi}(\mathbf{z}|\mathbf{x})$?

→ Un choix naturel : loi normale

→ $q_{\phi}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma} \mathbf{I})$

Quel *prior* $p_{\theta}(\mathbf{z})$?

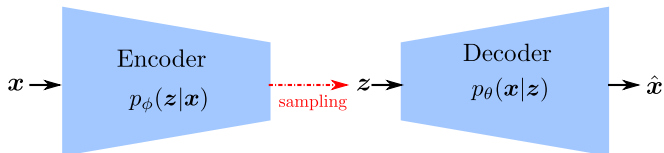
→ On cherche un espace latent structuré

→ Encore une fois : $p_{\theta}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$

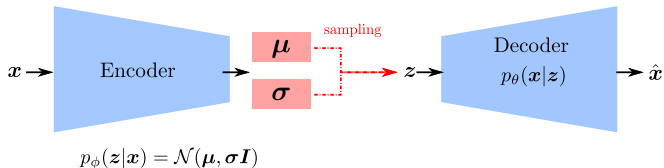
On alors

$$\text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z})) = \frac{1}{2} (\text{tr}(\boldsymbol{\sigma} \mathbf{I}) + \boldsymbol{\mu}^T \boldsymbol{\mu} - k - \log \det(\boldsymbol{\sigma} \mathbf{I}))$$

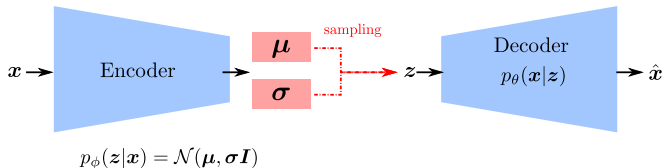
Variational Auto-Encoder



Variational Auto-Encoder



Variational Auto-Encoder



Problème : on ne peut pas dériver une opération d'échantillonnage (*sampling*).

Reparameterization trick

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{gradient par rapport à } \phi?} - \underbrace{\text{KL} (q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}))}_{\text{expression analytique ici}}$$

Reparameterization trick

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{gradient par rapport à } \phi?} - \underbrace{\text{KL} (q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}))}_{\text{expression analytique ici}}$$

Astuce : $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ avec $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Reparameterization trick

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{gradient par rapport à } \phi?} - \underbrace{\text{KL} (q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}))}_{\text{expression analytique ici}}$$

Astuce : $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ avec $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

On peut donc calculer le gradient par rapport à une donnée (i) de $\mathcal{D}$:

$$\begin{aligned} \mathcal{L}^{(i)}(\theta, \phi; \mathbf{x}^{(i)}) &\simeq \frac{1}{2} \sum_j^d (1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2) \\ &\quad + \frac{1}{L} \sum_{l=1}^L \log p_\theta(\mathbf{x}^{(i)} | \mathbf{z}^{(i,l)}) \end{aligned}$$

avec $\mathbf{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)}$

Reparameterization trick

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{gradient par rapport à } \phi?} - \underbrace{\text{KL} (q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}))}_{\text{expression analytique ici}}$$

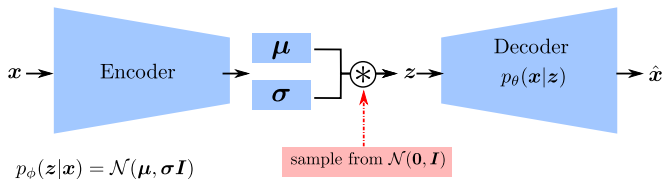
Astuce : $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ avec $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Si les batch sont assez larges, on peut ne garder qu'un tirage comme estimation de l'espérance :

$$\begin{aligned} \mathcal{L}^{(i)}(\theta, \phi; \mathbf{x}^{(i)}) &\simeq \frac{1}{2} \sum_j^d (1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2) \\ &\quad + \log p_\theta(\mathbf{x}^{(i)} | \mathbf{z}^{(i)}) \end{aligned}$$

avec $\mathbf{z}^{(i)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}$

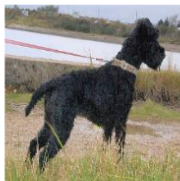
Variational Auto-Encoder : final form



VAE - générations de données



a man with a snowboard
next to a man with glasses



a big black dog standing on
the grass



a player is holding a
hockey stick

FIGURE – Pu et al. 2016

VAE - générations de données



FIGURE – Razavi et al. 2019

VAE - interpolation



FIGURE – Sainburg et al. 2018

VAE - interpolation

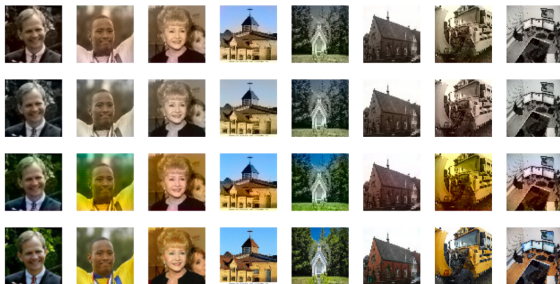


FIGURE – Deshpande et al. CVPR 2016