

RCP211 – Modèles génératifs

Auto-encodeurs variationnels

Arnaud Breloy arnaud.breloy@lecnam.net

25 septembre 2024

Conservatoire national des arts & métiers

1. Rappels sur les auto-encodeurs
2. Principe du VAE
3. Inférence variationnelle
4. Optimisation des VAE

Rappels sur les auto-encodeurs

Auto-encodeurs (récapitulatif, 1/2)

Principe de l'auto-encodeur

Apprendre une *compression* avec les pertes les + faibles possibles.

Pour une variable aléatoire $\mathbf{X} \in \mathbb{R}^n$, un réseau auto-encodeur modélise $\mathcal{H} = \mathcal{D} \circ \mathcal{E}$ telle que :

$$\|\mathcal{H}(x) - x\| \leq \varepsilon$$

- \mathcal{E} représente l'encodeur $\mathbb{R}^n \rightarrow \mathbb{R}^d$
- \mathcal{D} représente le décodeur $\mathbb{R}^d \rightarrow \mathbb{R}^n$

L'auto-encodeur construit un espace latent \mathcal{Z} qui est l'espace de dimension d contenant les codes $z = \mathcal{E}(x)$.

Auto-encodeurs (récapitulatif, 2/2)

Fonction de coût

On cherche les poids θ du réseau de neurones \mathcal{H} tels que :

$$\theta^* = \arg \min_{\theta} \mathcal{L}(x, \hat{x}) = \|\mathcal{D}(\mathcal{E}(x)) - x\|$$

La fonction de coût de reconstruction dépend de la tâche (erreur quadratique, absolue, entropie croisée...).

Optimisation

Apprentissage classique comme n'importe quel réseau de neurones : algorithme de rétropropagation et descente de gradient stochastique.

Le décodeur et l'encodeur sont appris *conjointement* (le gradient est rétropropagé du décodeur vers l'encodeur).

Auto-encodeurs (récapitulatif, 2/2)

Fonction de coût

On cherche les poids θ du réseau de neurones \mathcal{H} tels que :

$$\theta^* = \arg \min_{\theta} \mathcal{L}(x, \hat{x}) = \|\mathcal{D}(\mathcal{E}(x)) - x\|$$

La fonction de coût de reconstruction dépend de la tâche (erreur quadratique, absolue, entropie croisée...).

Optimisation

Apprentissage classique comme n'importe quel réseau de neurones : algorithme de rétropropagation et descente de gradient stochastique.

Le décodeur et l'encodeur sont appris *conjointement* (le gradient est rétropropagé du décodeur vers l'encodeur).

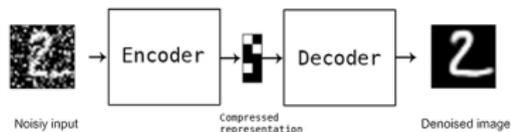
Applications des auto-encodeurs : débruitage

Débruitage/restauration

Fonction de coût : $\|\mathcal{H}(\tilde{x}) - x\|$ où $\tilde{x} = x + \varepsilon$ une version bruitée de x

Diverses applications

- Compression de signaux avec pertes
- Réduction de dimension non-linéaire



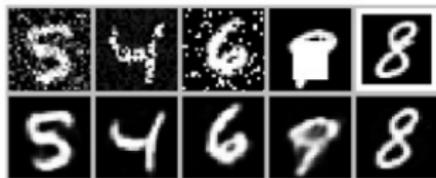
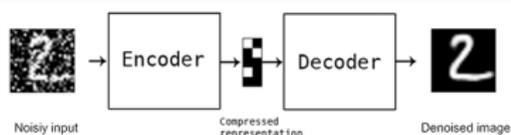
Applications des auto-encodeurs : débruitage

Débruitage/restauration

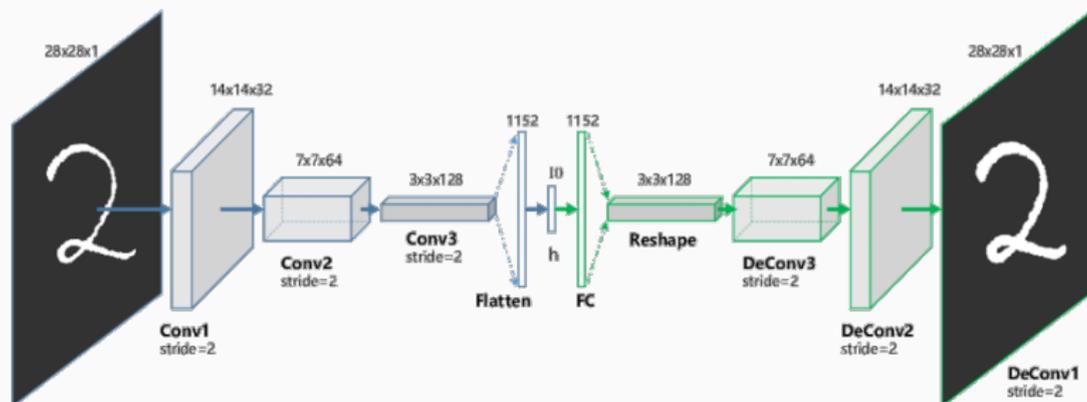
Fonction de coût : $\|\mathcal{H}(\tilde{x}) - x\|$ où $\tilde{x} = x + \varepsilon$ une version bruitée de x

Diverses applications

- Compression de signaux avec pertes
- Réduction de dimension non-linéaire



Applications des auto-encodeurs : AE convolutif



Guo et al., *Deep Clustering with Convolutional Autoencoders*, ICONIP 2017

→ cf. séance de TP

L'auto-encodeur comme modèle génératif

Décoder = générer

Le décodeur \mathcal{D} est un modèle génératif $\mathbb{P}(X|z)$.

1. Choisir au hasard un code latent z de dimension d
2. Décoder ce vecteur

Micro-quiz : à quoi peut-on s'attendre en pratique ?

1. Une observation synthétique mais plausible
2. Une reconstruction médiocre

Limites de cette approche

- Comment choisir z ? A priori on ne sait pas!
- Pas de régularité (= continuité) dans l'espace latent z
- $d \ll n$ mais d reste grand en pratique

L'auto-encodeur comme modèle génératif

Décoder = générer

Le décodeur \mathcal{D} est un modèle génératif $\mathbb{P}(X|z)$.

1. Choisir au hasard un code latent z de dimension d
2. Décoder ce vecteur

Micro-quiz : à quoi peut-on s'attendre en pratique ?

1. Une observation synthétique mais plausible
2. Une reconstruction médiocre

Limites de cette approche

- Comment choisir z ? A priori on ne sait pas!
- Pas de régularité (= continuité) dans l'espace latent z
- $d \ll n$ mais d reste grand en pratique

L'auto-encodeur comme modèle génératif

Décoder = générer

Le décodeur \mathcal{D} est un modèle génératif $\mathbb{P}(X|z)$.

1. Choisir au hasard un code latent z de dimension d
2. Décoder ce vecteur

Micro-quiz : à quoi peut-on s'attendre en pratique ?

1. Une observation synthétique mais plausible
2. Une reconstruction médiocre

Limites de cette approche

- Comment choisir z ? A priori on ne sait pas!
- Pas de régularité (= continuité) dans l'espace latent z
- $d \ll n$ mais d reste grand en pratique

Principe du VAE

De l'auto-encodeur à l'auto-encodeur variationnel

Idée générale

une observation = une distribution

- une donnée x correspond à plusieurs z possibles,
- l'encodeur ne produit pas un code mais une *distribution* $p(z|x)$.

En général, on choisira une distribution gaussienne dont les paramètres (μ_x, σ_x) sont prédits par l'encodeur.

Avantages

- L'espace latent est moins clairsemé
- Échantillonnage plus aisé pour un x donné
- $p(z)$ plus facile à estimer : $p(z) = \sum_{x_i \in \mathcal{X}} p(z|x_i)$

De l'auto-encodeur à l'auto-encodeur variationnel

Idée générale

une observation = une distribution

- une donnée x correspond à plusieurs z possibles,
- l'encodeur ne produit pas un code mais une *distribution* $p(z|x)$.

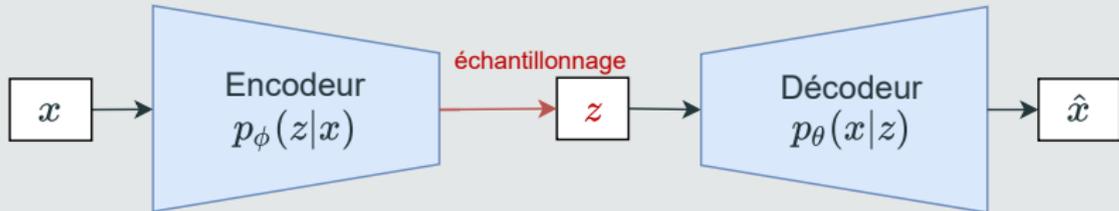
En général, on choisira une distribution gaussienne dont les paramètres (μ_x, σ_x) sont prédits par l'encodeur.

Avantages

- L'espace latent est moins clairsemé
- Échantillonnage plus aisé pour un x donné
- $p(z)$ plus facile à estimer : $p(z) = \sum_{x_i \in \mathcal{X}} p(z|x_i)$

Schéma du VAE

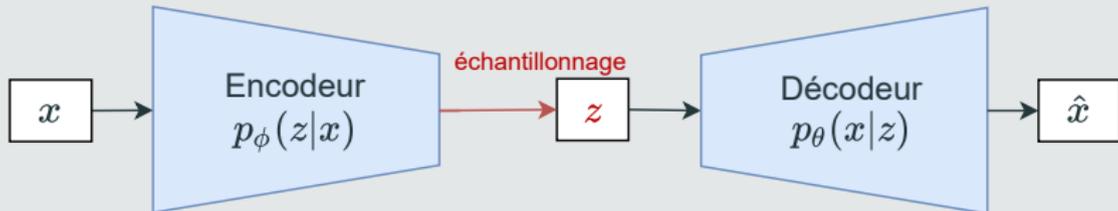
Schéma général



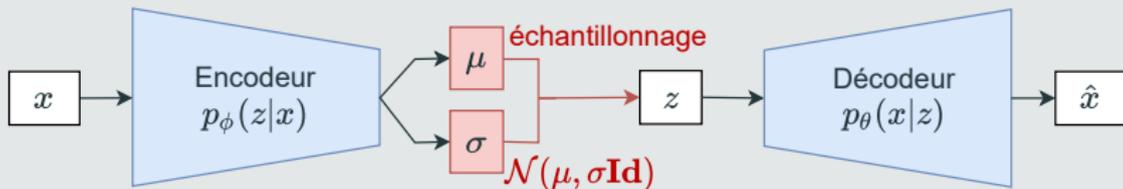
VAE gaussien

Schéma du VAE

Schéma général



VAE gaussien



Récapitulons...

- L'encodeur produit une distribution $p_\phi(z|x)$
 - Typiquement, $p_\phi(z|x) = \mathcal{N}(\mu_x, \sigma_x)$
- On échantillonne $p_\phi(z|x)$ pour trouver un (ou plusieurs) code z
- Le décodeur reconstruit x à partir de z

Échantillonnage lors de l'inférence

À l'inférence, comment choisir z ?

- Option 1 : échantillonner sur la loi a posteriori
 - $p(z) = \frac{1}{n} \sum_{i=1}^n p_\phi(z|x_i) = \frac{1}{n} \sum_{i=1}^n \mathcal{N}(\mu_{x_i}, \sigma_{x_i})$
 - approximation mauvaise si peu de données, dimension de z grande, etc.
 - cher à calculer si beaucoup de données (n grand)
- Option 2 : forcer $p(z)$ à suivre une loi a priori

Récapitulons...

- L'encodeur produit une distribution $p_\phi(z|x)$
 - Typiquement, $p_\phi(z|x) = \mathcal{N}(\mu_x, \sigma_x)$
- On échantillonne $p_\phi(z|x)$ pour trouver un (ou plusieurs) code z
- Le décodeur reconstruit x à partir de z

Échantillonnage lors de l'inférence

À l'inférence, comment choisir z ?

- Option 1 : échantillonner sur la loi a posteriori
 - $p(z) = \frac{1}{n} \sum_{i=1}^n p_\phi(z|x) = \frac{1}{n} \sum_{i=1}^n \mathcal{N}(\mu_{x_i}, \sigma_{x_i})$
 - approximation mauvaise si peu de données, dimension de z grande, etc.
 - cher à calculer si beaucoup de données (n grand)
- Option 2 : forcer $p(z)$ à suivre une loi a priori

Récapitulons...

- L'encodeur produit une distribution $p_\phi(z|x)$
 - Typiquement, $p_\phi(z|x) = \mathcal{N}(\mu_x, \sigma_x)$
- On échantillonne $p_\phi(z|x)$ pour trouver un (ou plusieurs) code z
- Le décodeur reconstruit x à partir de z

Échantillonnage lors de l'inférence

À l'inférence, comment choisir z ?

- Option 1 : échantillonner sur la loi a posteriori
 - $p(z) = \frac{1}{n} \sum_{i=1}^n p_\phi(z|x) = \frac{1}{n} \sum_{i=1}^n \mathcal{N}(\mu_{x_i}, \sigma_{x_i})$
 - approximation mauvaise si peu de données, dimension de z grande, etc.
 - cher à calculer si beaucoup de données (n grand)
- **Option 2 : forcer $p(z)$ à suivre une loi a priori**

La divergence de Kullback-Leibler

Mesure de dissimilarité entre deux distributions $P(x)$ et $Q(x)$

- $D_{\text{KL}}(P|Q) = \sum_i P(x_i) \log \frac{P(x_i)}{Q(x_i)}$

A priori gaussien dans les VAE

Objectif : on exige que $p(z)$ suive approximativement la loi normale $\in \mathbb{R}^d$

- on impose :

$$D_{\text{KL}}\left(\underbrace{q_\phi(z|x)}_{\text{approximation du postérieur par l'encodeur}} \mid \underbrace{p(z)}_{\text{la loi a priori}}\right) \leq \varepsilon$$

- avec $q_\phi(z|x) = \mathcal{N}(\mu_x, \sigma_x)$
- on choisit comme prior : $p(z) = \mathcal{N}(0, \mathbf{I})$

La divergence de Kullback-Leibler

Mesure de dissimilarité entre deux distributions $P(x)$ et $Q(x)$

- $D_{\text{KL}}(P|Q) = \sum_i P(x_i) \log \frac{P(x_i)}{Q(x_i)}$

A priori gaussien dans les VAE

Objectif : on exige que $p(z)$ suive approximativement la loi normale $\in \mathbb{R}^d$

- on impose :

$$D_{\text{KL}}(\underbrace{q_\phi(z|x)}_{\text{approximation du postérieur par l'encodeur}} \mid \underbrace{p(z)}_{\text{la loi a priori}}) \leq \varepsilon$$

- avec $q_\phi(z|x) = \mathcal{N}(\mu_x, \sigma_x)$
- on choisit comme *prior* : $p(z) = \mathcal{N}(0, \mathbf{I})$

Fonction de coût du VAE

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)} [\|\hat{x} - x\|] + D_{\text{KL}}(q_\phi(z|x)|p(z))$$

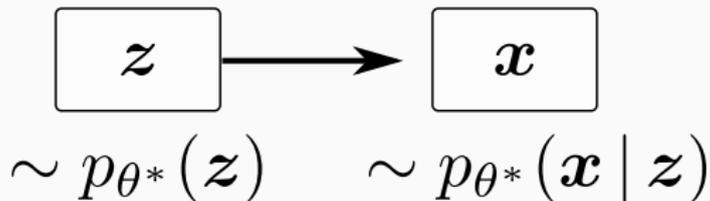
- **Reconstruction** : « similarité » entre l'entrée et la sortie
 - dépend du problème (MSE, MAE, BCE...)
- **Régularisation** : divergence-KL (contrainte de proximité au *prior*)
 - Expression analytique dans le cas où tout est gaussien :

$$D_{\text{KL}}(q_\phi(z|x)|p(z)) = \frac{1}{2}(\text{tr}(\sigma_x) + \mu_x^t \mu_x - d - \log \det(\sigma_x))$$

Inférence variationnelle

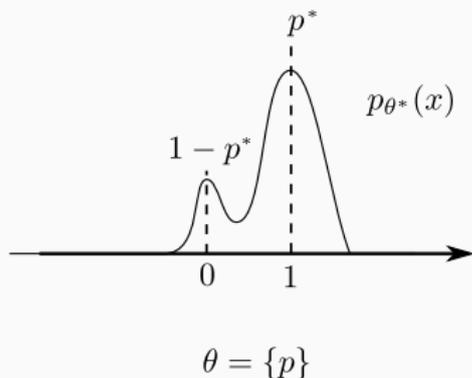
Cadre formel

- $\mathcal{D} = \{x_1, \dots, x_n\}$ un jeu de données i.i.d. issues d'un processus génératif $p_{\mathcal{D}}$.
- On cherche le modèle génératif qui permet d'approcher $p_{\mathcal{D}}$.
- Soit $z \in \mathcal{Z}$ une variable latente qui suit une distribution *a priori* $p_{\theta^*}(z)$, paramétrée par θ^* .
 - on suppose que x_i s'obtient par la réalisation de $p_{\theta^*}(x_i | z_i)$
 - en pratique, on ne connaît ni θ^* , ni les variables latentes z_i



Trouver la distribution latente

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



Modèle génératif

On cherche θ qui maximise la (log-)vraisemblance sur \mathcal{D} :

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

Problèmes

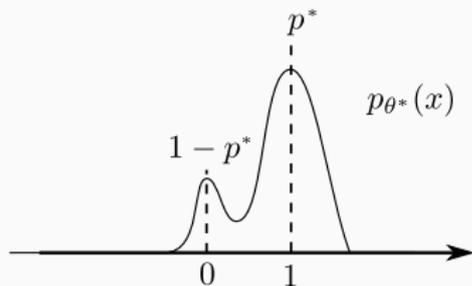
- $p_{\theta}(x)$ est difficile à calculer...

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

- $p_{\theta}(z|x)$ est inconnu...
- \mathcal{Z} est trop grand pour échantillonner efficacement

Trouver la distribution latente

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



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

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

→ **sampling**

Modèle génératif

On cherche θ qui maximise la (log-)vraisemblance sur \mathcal{D} :

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

Problèmes

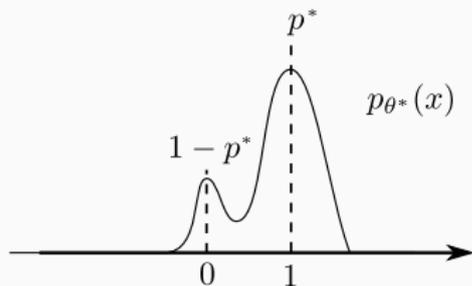
- $p_{\theta}(x)$ est difficile à calculer...

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

- $p_{\theta}(z|x)$ est inconnu...
- \mathcal{Z} est trop grand pour échantillonner efficacement

Trouver la distribution latente

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



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

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

→ sampling

Modèle génératif

On cherche θ qui maximise la (log-)vraisemblance sur \mathcal{D} :

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

Problèmes

- $p_{\theta}(x)$ est difficile à calculer...

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

- $p_{\theta}(z|x)$ est inconnu...
- \mathcal{Z} est trop grand pour échantillonner efficacement

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

On cherche à être proche du véritable postérieur $p_\theta(z|x)$:

$$\hat{\phi} = \arg \min_{\phi} \text{KL}(q_\phi(z|x) | p_\theta(z|x))$$

On peut montrer que :

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

Minimiser la divergence KL revient à **maximiser** \mathcal{L} .

Définition

ELBO (*evidence lower-bound*) est définie par :

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

Minimiser la divergence KL revient à **maximiser** l'ELBO.

Fonction objectif de l'inférence variationnelle

- On cherche θ qui vérifie $\hat{\theta} = \arg \max_{\theta} \mathbb{E}_{p_{\mathcal{D}}} [\log p_{\theta}(x)]$
- Or, $\log p_{\theta}(x) = \text{KL} (q_{\phi}(z|x) | p_{\theta}(z|x)) + \mathcal{L}(\theta, \phi; x)$
- Comme la divergence KL est positive, il vient :

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

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

Définition

ELBO (*evidence lower-bound*) est définie par :

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

Minimiser la divergence KL revient à **maximiser** l'ELBO.

Fonction objectif de l'inférence variationnelle

- On cherche θ qui vérifie $\hat{\theta} = \arg \max_{\theta} \mathbb{E}_{p_{\mathcal{D}}} [\log p_{\theta}(x)]$
- Or, $\log p_{\theta}(x) = \text{KL}(q_{\phi}(z|x) || p_{\theta}(z|x)) + \mathcal{L}(\theta, \phi; x)$
- Comme la divergence KL est positive, il vient :

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

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

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

Quelles fonctions choisir comme postérieur et *prior* ?

- Un choix naturel pour $q_\phi(z|x)$: une gaussienne $\mathcal{N}(\mu, \sigma)$
 - ses paramètres dépendent de x
- Un choix naturel pour $p_\theta(z)$: la loi normale $\mathcal{N}(0, \mathbf{I})$
 - structure simple et facile à échantillonner

Vraisemblance

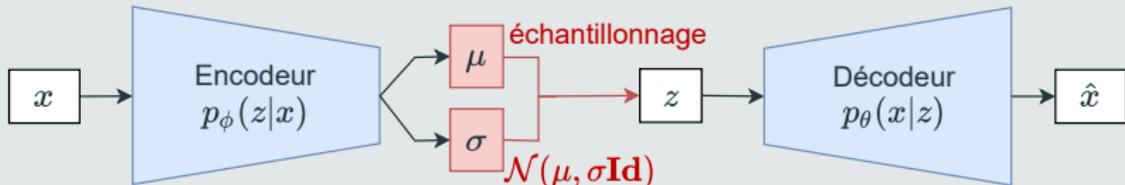
Dans notre cas, en notant f le décodeur :

$$\arg \max \mathbb{E}_{q_\phi(z|x)} [\log p(x|z)] = \arg \min \mathbb{E}_{q_\phi(z|x)} \|x - f(z)\|$$

Optimisation des VAE

Entraînement en pratique par descente de gradient

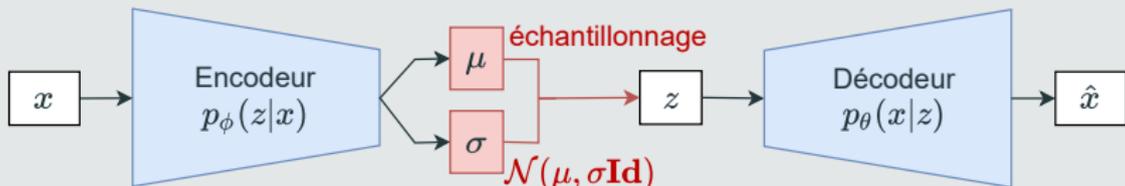
Schéma du VAE



1. Création d'un *batch* $\{x_i\}$
2. Passage dans l'encodeur
 - Calcul de μ_{x_i}, σ_{x_i} pour chaque i
3. Échantillonnage de $z_i \sim \mathcal{N}(\mu_{x_i}, \sigma_{x_i})$ ← non dérivable!
4. Passage dans le décodeur
 - Calcul de \hat{x}_i
5. Calcul de la fonction de coût
6. Calcul du gradient, rétropropagation

Entraînement en pratique par descente de gradient

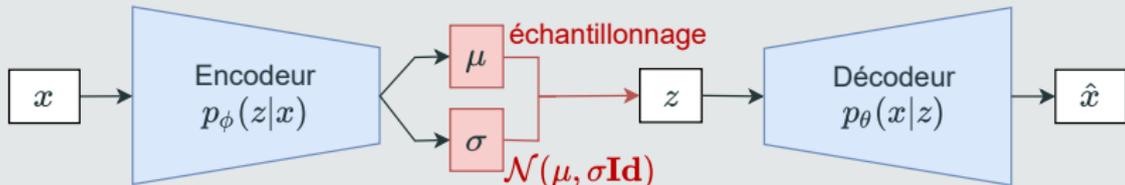
Schéma du VAE



1. Création d'un *batch* $\{x_i\}$
2. Passage dans l'encodeur
 - Calcul de μ_{x_i}, σ_{x_i} pour chaque i
3. Échantillonnage de $z_i \sim \mathcal{N}(\mu_{x_i}, \sigma_{x_i})$ ← non dérivable!
4. Passage dans le décodeur
 - Calcul de \hat{x}_i
5. Calcul de la fonction de coût
6. Calcul du gradient, rétropropagation

Entraînement en pratique par descente de gradient

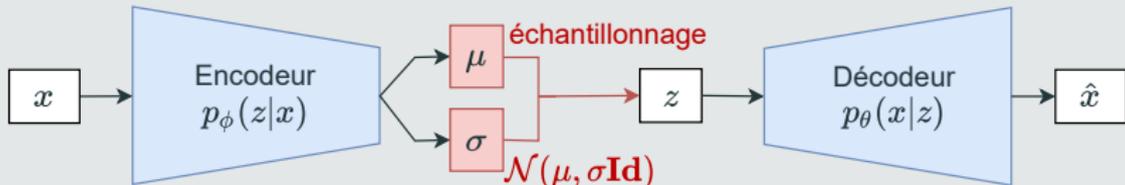
Schéma du VAE



1. Création d'un *batch* $\{x_i\}$
2. Passage dans l'encodeur
 - Calcul de μ_{x_i}, σ_{x_i} pour chaque i
3. Échantillonnage de $z_i \sim \mathcal{N}(\mu_{x_i}, \sigma_{x_i})$ ← non dérivable!
4. Passage dans le décodeur
 - Calcul de \hat{x}_i
5. Calcul de la fonction de coût
6. Calcul du gradient, rétropropagation

Entraînement en pratique par descente de gradient

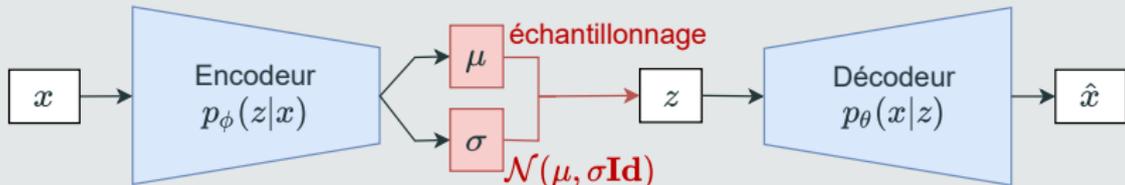
Schéma du VAE



1. Création d'un *batch* $\{x_i\}$
2. Passage dans l'encodeur
 - Calcul de μ_{x_i}, σ_{x_i} pour chaque i
3. Échantillonnage de $z_i \sim \mathcal{N}(\mu_{x_i}, \sigma_{x_i}) \leftarrow$ non dérivable!
4. Passage dans le décodeur
 - Calcul de \hat{x}_i
5. Calcul de la fonction de coût
6. Calcul du gradient, rétropropagation

Entraînement en pratique par descente de gradient

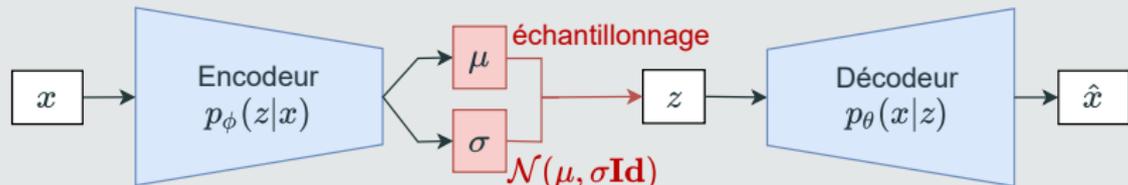
Schéma du VAE



1. Création d'un *batch* $\{x_i\}$
2. Passage dans l'encodeur
 - Calcul de μ_{x_i}, σ_{x_i} pour chaque i
3. Échantillonnage de $z_i \sim \mathcal{N}(\mu_{x_i}, \sigma_{x_i}) \leftarrow$ non dérivable!
4. Passage dans le décodeur
 - Calcul de \hat{x}_i
5. Calcul de la fonction de coût
6. Calcul du gradient, rétropropagation

Entraînement en pratique par descente de gradient

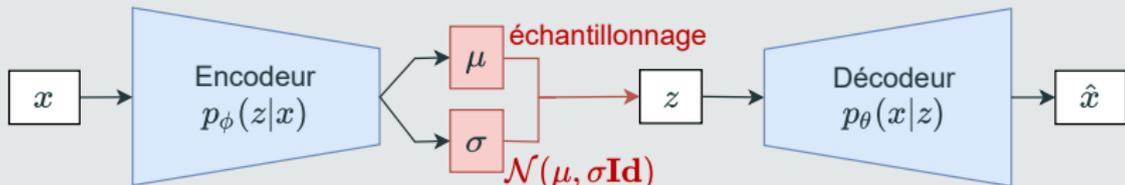
Schéma du VAE



1. Création d'un *batch* $\{x_i\}$
2. Passage dans l'encodeur
 - Calcul de μ_{x_i}, σ_{x_i} pour chaque i
3. Échantillonnage de $z_i \sim \mathcal{N}(\mu_{x_i}, \sigma_{x_i}) \leftarrow$ non dérivable!
4. Passage dans le décodeur
 - Calcul de \hat{x}_i
5. Calcul de la fonction de coût
6. Calcul du gradient, rétropropagation

Entraînement en pratique par descente de gradient

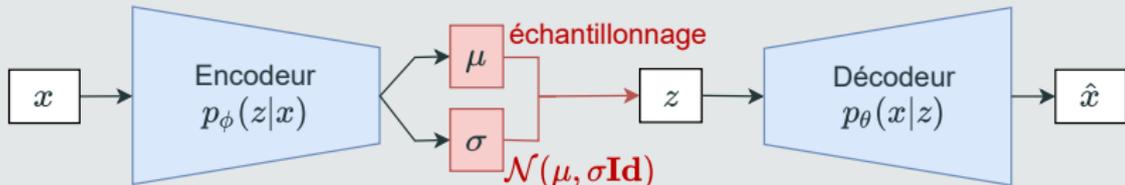
Schéma du VAE



1. Création d'un *batch* $\{x_i\}$
2. Passage dans l'encodeur
 - Calcul de μ_{x_i}, σ_{x_i} pour chaque i
3. Échantillonnage de $z_i \sim \mathcal{N}(\mu_{x_i}, \sigma_{x_i}) \leftarrow$ non dérivable!
4. Passage dans le décodeur
 - Calcul de \hat{x}_i
5. Calcul de la fonction de coût
6. Calcul du gradient, rétropropagation

Entraînement en pratique par descente de gradient

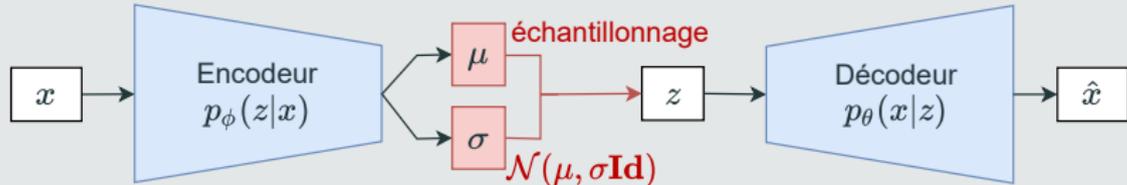
Schéma du VAE



1. Création d'un *batch* $\{x_i\}$
2. Passage dans l'encodeur
 - Calcul de μ_{x_i}, σ_{x_i} pour chaque i
3. Échantillonnage de $z_i \sim \mathcal{N}(\mu_{x_i}, \sigma_{x_i}) \leftarrow$ non dérivable!
4. Passage dans le décodeur
 - Calcul de \hat{x}_i
5. Calcul de la fonction de coût
6. Calcul du gradient, rétropropagation

Entraînement en pratique par descente de gradient

Schéma du VAE



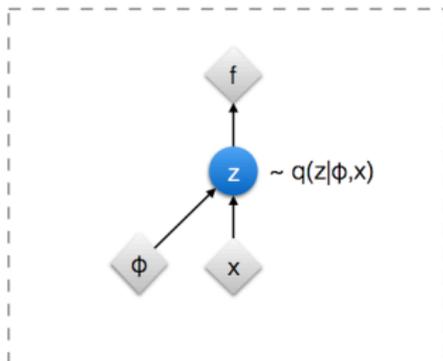
1. Création d'un *batch* $\{x_i\}$
2. Passage dans l'encodeur
 - Calcul de μ_{x_i}, σ_{x_i} pour chaque i
3. Échantillonnage de $z_i \sim \mathcal{N}(\mu_{x_i}, \sigma_{x_i}) \leftarrow$ **non dérivable!**
4. Passage dans le décodeur
 - Calcul de \hat{x}_i
5. Calcul de la fonction de coût
6. Calcul du gradient, rétropropagation

Reparameterization trick

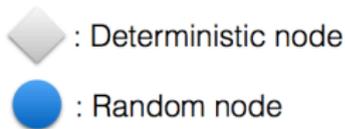
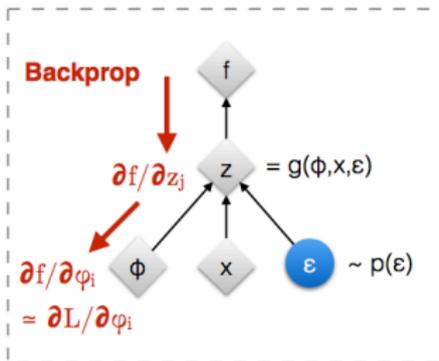
Astuce

Ne pas échantillonner z directement en réécrivant : $z = \mu + \sigma \odot \varepsilon$ avec ε un bruit gaussien aléatoire $\sim \mathcal{N}(0, \mathbf{I})$.

Original form



Reparameterised form



[Kingma, 2013]
[Bengio, 2013]
[Kingma and Welling 2014]
[Rezende et al 2014]

$$\mathcal{L}(\theta, \phi; x) = \underbrace{\mathbb{E}_{q_\phi(z|x)} \|\hat{x} - x\|}_{\text{Erreur de reconstruction}} + \beta \cdot \underbrace{\text{KL}(q_\phi(z|x) \parallel p_\theta(z))}_{\text{Écart au prior}}$$

β contrôle le dilemme reconstruction/structure de l'espace latent :

- $\beta = 1$: VAE classique,
- $\beta > 1$: encodage plus efficace et plus proche du *prior* mais reconstruction moins bonne,
- $\beta < 1$: meilleure reconstruction mais espace latent moins bien structuré.