Echantillonner de données complexes avec les auto-encodeurs variationnels

les bases mathématiques

Séminaire TALia du 17/06/2022

Référence

Lecture: Machine Learning for Graphs and Sequential Data

Informatik 26 - Data Analytics and Machine Learning

Prof. Dr. Stephan Günnemann

Référence 1

Référence 2

arXiv > stat > arXiv:1312.6114

Statistics > Machine Learning

[Submitted on 20 Dec 2013 (v1), last revised 1 May 2014 (this version, v10)]

Auto-Encoding Variational Bayes

Diederik P Kingma, Max Welling

arXiv > cs > arXiv:1511.06406

Computer Science > Machine Learning

[Submitted on 19 Nov 2015 (v1), last revised 4 Jan 2016 (this version, v2)]

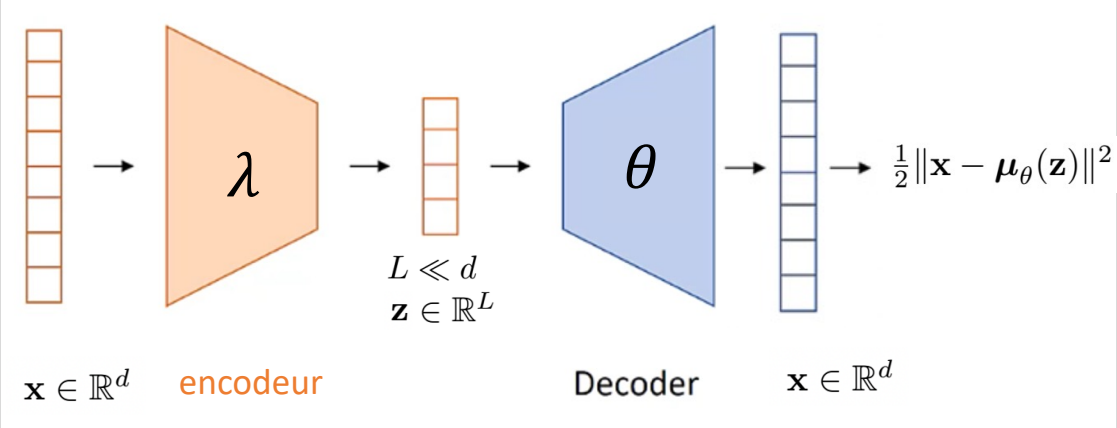
Denoising Criterion for Variational Auto-Encoding Framework

Daniel Jiwoong Im, Sungjin Ahn, Roland Memisevic, Yoshua Bengio

Plan

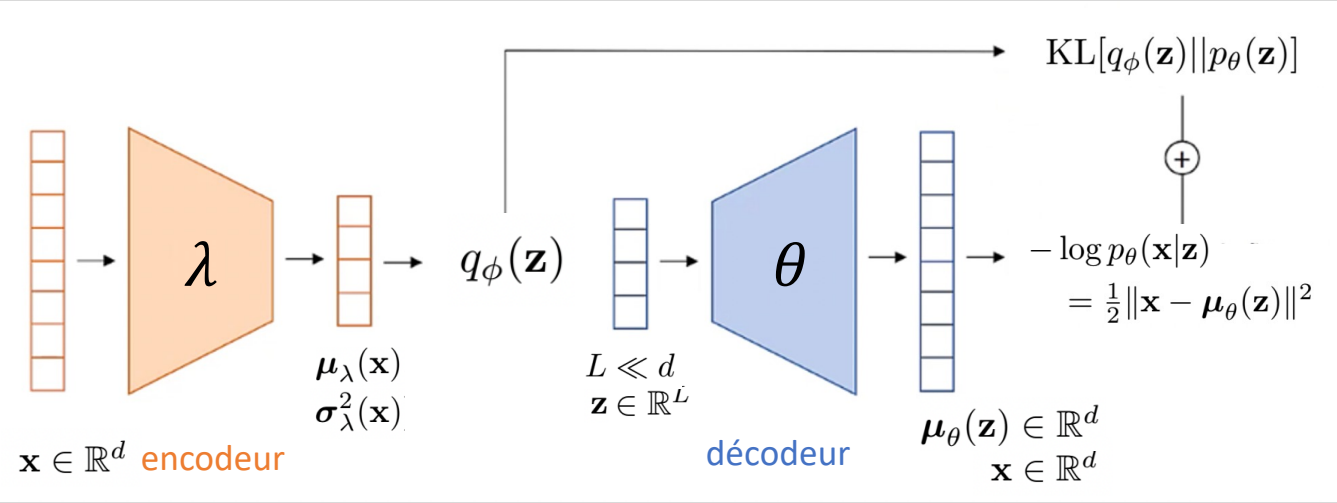
1. Modèles à variable latentes
2. Méthode variationnelle
3. Justification des intuitions
4. Applications et variantes

« Vanilla » auto-encodeur : réduction dimensionnelle (≈ PCA « non linéaire »)



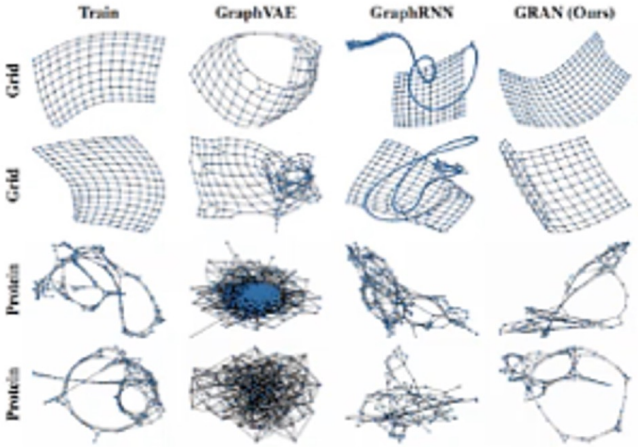
un modèle déterministe

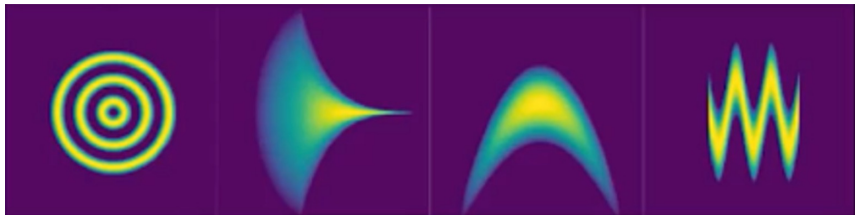
Auto-encodeur variationnel : apprentissage de représentations et modèle génératif

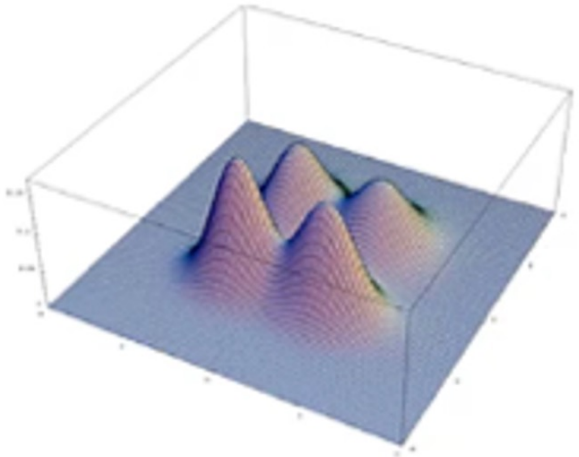


un modèle stochastique

Échantillonnage de données complexes







NLP

Applications des VAE

- Echantillonnage et complétion d'images
- Data augmentation, données synthétiques
- Résumé automatique
- Traduction automatique
- Image captioning
- Génération de dialogues
- Prédiction de liens dans des graphes avec ou sans attributs
- Clustering
- Modélisation de la parole
- Détection d'anomalies (AE)
- Concurrent des GAN et des « Normalizing Flow… »
- …

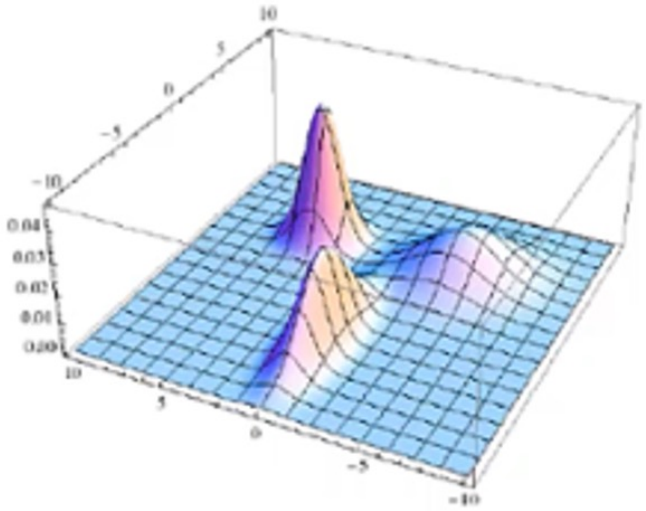
Modèles avec variables latentes

Intuition de base : une marginale $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z})\, \mathrm{d}\mathbf{z} = \int p_\theta(\mathbf{x}|\mathbf{z})\, p_\theta(\mathbf{z})\, \mathrm{d}\mathbf{z}$ peut-être « compliquée »

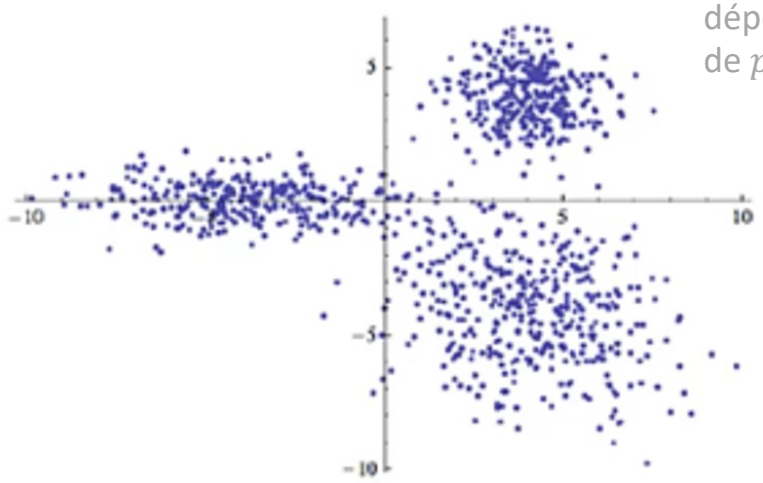
même lorsque $p_\theta(\mathbf{x}|\mathbf{z})$ est « simple ».

variables latentes non-observées

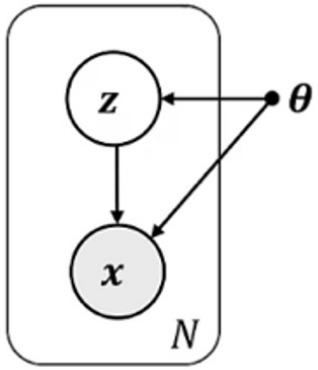
on s'intéresse aux cas où cette intégrable n'est pas calculable analytiquement à cause de la dépendance compliquée en $\boldsymbol{z}$ de $p_\theta(\mathbf{x}|\mathbf{z})$



mélange de gaussiennes



échantillons du mélange de gaussiennes



Tâches utiles pour un modèle avec variables latentes

L'inférence : étant donnée une observation $\mathbf{x}$, calculer la distribution à postériori sur les variables latentes $\mathbf{z}$
= trouver l'origine/cause la plus probable d'une observation :

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

L'apprentissage : étant donné un échantillon d'observation i.i.d. $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$ trouver le paramètre θ_{ML} qui explique le mieux les données :

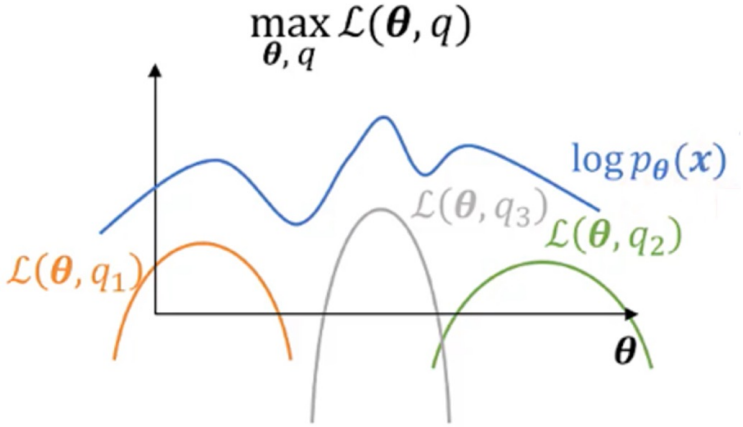
$$\theta_{\mathrm{ML}} = \arg\max_\theta \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(\mathbf{x}^{(i)})$$

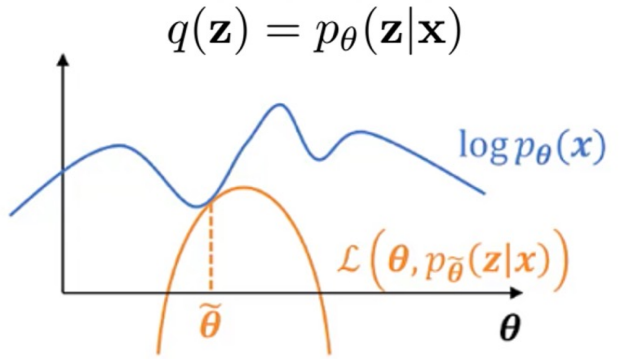
Estimation du maximum de vraisemblance avec ELBO

Le calcul direct de $\max_\theta p_\theta(\mathbf{x})$ étant impraticable on construit une borne inférieure variationnelle (ELBO) :

$$\log p_\theta(\mathbf{x}) \geq \mathscr{L}(\mathbf{x};\theta,q) = \mathbb{E}_{q(\mathbf{z})}[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z})] \quad \forall q(\mathbf{z}) \text{ avec égalité ssi } q(\mathbf{z}) = p_\theta(\mathbf{z}|\mathbf{x})$$

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \mathbb{E}_{q(\mathbf{z})}[\log p_\theta(\mathbf{x})] \\
&= \int q(\mathbf{z}) \log p_\theta(\mathbf{x})\, d\mathbf{z} \\
&= \int q(\mathbf{z}) \log \frac{p_\theta(\mathbf{x},\mathbf{z})}{p_\theta(\mathbf{x}|\mathbf{z})}\, d\mathbf{z} \\
&= \int q(\mathbf{z}) \log \left(\frac{p_\theta(\mathbf{x},\mathbf{z})}{p_\theta(\mathbf{x}|\mathbf{z})} \cdot \frac{q(\mathbf{z})}{q(\mathbf{z})} \right) d\mathbf{z} \\
&= \int q(\mathbf{z}) \log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\, d\mathbf{z} + \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p_\theta(\mathbf{x}|\mathbf{z})}\, d\mathbf{z} \\
&= \underbrace{\mathbb{E}_{q(\mathbf{z})}\left[p_\theta(\mathbf{x},\mathbf{z}) - q(\mathbf{z})\right]}_{:= \mathscr{L}(\mathbf{x};\theta,\mathbf{q})} + \underbrace{\mathrm{KL}(q(\mathbf{z})||p_\theta(\mathbf{z}|\mathbf{x}))}_{\geq 0} \quad \forall q(\mathbf{z})
\end{aligned}$$





Expectation–Maximization comme application de ELBO

$$\max_{\theta} \log p_\theta(\mathbf{x}) = \max_{\theta,q} \mathscr{L}(\mathbf{x};\theta,q) \quad \text{où} \quad \mathscr{L}(\mathbf{x};\theta,q) := \mathbb{E}_{q(\mathbf{z})}[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z})] \quad (1)$$

$$= \log p_\theta(\mathbf{x}) - \mathrm{KL}[q(\mathbf{z})||p_\theta(\mathbf{z}|\mathbf{x})] \quad (2)$$

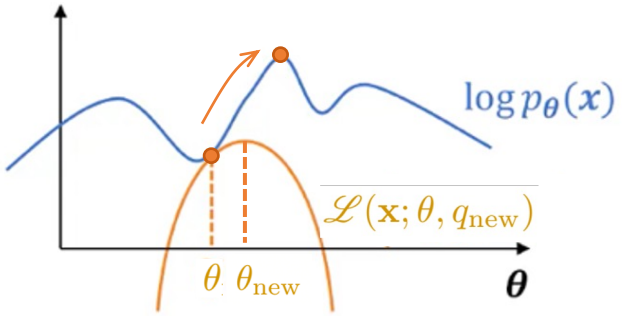
$$= \mathbb{E}_{q(\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[q(\mathbf{z})||p_\theta(\mathbf{z})] \quad (3)$$

Si la distribution à postériori $p_\theta(\mathbf{z}|\mathbf{x})$ est calculable, la maximisation alternée selon θ puis selon q correspond à la méthode dite d' « Expectation-Maximization » :

Expectation = « inférence »

$$q_{\text{new}} = \arg\max_q \mathscr{L}(\mathbf{x};\theta,q) = \arg\min_q \mathrm{KL}[q(\mathbf{z})||p_\theta(\mathbf{z}|\mathbf{x})] \overset{(2)}{=} p_\theta(\mathbf{z}|\mathbf{x})$$

$$q_{\text{new}} \to q$$



Mazimization = « apprentissage »

$$\theta_{\text{new}} = \arg\max_\theta \mathscr{L}(\mathbf{x};\theta,q) \overset{(1)}{=} \arg\max_\theta \mathbb{E}_{\mathbf{z}\sim q}[\log p_\theta(\mathbf{x},\mathbf{z})]$$

$$\theta_{\text{new}} \to \theta$$

Paramétrisation des distributions variationnelles

Problème : en général on ne sait pas calculer exactement le maximum sur les distributions q dans l'espace latent.

Solution : introduire une paramétrisation judicieuse $q_\phi(\mathbf{z})$ de l'espace des distributions avec $\phi \in \mathbb{R}^k$ et utiliser la rétropropagation et une DGS pour maximiser $\mathscr{L}$

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

Choisir un paramétrisation telle que :

1. Le paramètre optimal $\phi_{\text{opt}} = \phi_{\text{opt}}(\mathbf{x})$ qui maximise $\mathscr{L}(\mathbf{x}; \theta, q_\phi)$ puisse être approximé
2. La distribution variationnelle $q_{\phi_{\text{opt}}}(\mathbf{z})$ soit une bonne approximation de $p_\theta(\mathbf{z}|\mathbf{x})$

$\theta \in \mathbb{R}^m$: paramètres du modèle $p_\theta(\mathbf{x}, \mathbf{z})$ → $\nabla_\theta \mathscr{L}(\mathbf{x}; \theta, q_\phi)$

$\phi \in \mathbb{R}^k$: paramètres de la distribution variationnelle $q_\phi(\mathbf{z})$ → $\nabla_\phi \mathscr{L}(\mathbf{x}; \theta, q_\phi)$

Calcul des gradients de $\mathcal{L}$ par rapport à θ et à ϕ

Pour θ une simple approximation de l'intégrale par Monte Carlo fait l'affaire :

$$\begin{aligned}
\nabla_\theta \mathscr{L}(\mathbf{x};\theta,q_\phi) &= \nabla_\theta \int [\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z})]\, q_\phi(\mathbf{z})\, \mathrm{d}\mathbf{z} \\
&= \int \nabla_\theta \log p_\theta(\mathbf{x},\mathbf{z})\, q_\phi(\mathbf{z})\, \mathrm{d}\mathbf{z} \\
&\approx \frac{1}{K}\sum_{j=1}^{K} \nabla_\theta \log p_\theta(\mathbf{x},\mathbf{z}_j) \quad \text{où} \quad \mathbf{z}_j \sim q_\phi
\end{aligned}$$

Pour ϕ ça se corse un peu… il faut trouver une astuce !

$$\begin{aligned}
\nabla_\phi \mathscr{L}(\mathbf{x};\theta,q_\phi) &= \nabla_\phi \int [\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z})]\, q_\phi(\mathbf{z})\, \mathrm{d}\mathbf{z} \\
&= \int [\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z})]\, \nabla_\phi q_\phi(\mathbf{z})\, \mathrm{d}\mathbf{z} + \dots
\end{aligned}$$

on ne peut plus utiliser MC pour échantillonner !

Le « reparameterization trick » (1)

Idée : introduire une v.a. auxiliaire $\boldsymbol{\varepsilon}$ dont la loi ρ est indépendante du paramètre ϕ et construire la variable aléatoire $\mathbf{z}$ à l'aide d'une fonction déterministe appliquée à $\boldsymbol{\varepsilon}$.

Par exemple : $\boldsymbol{\epsilon} \sim \rho = \mathcal{N}(\mathbf{0}, \mathbb{I})$

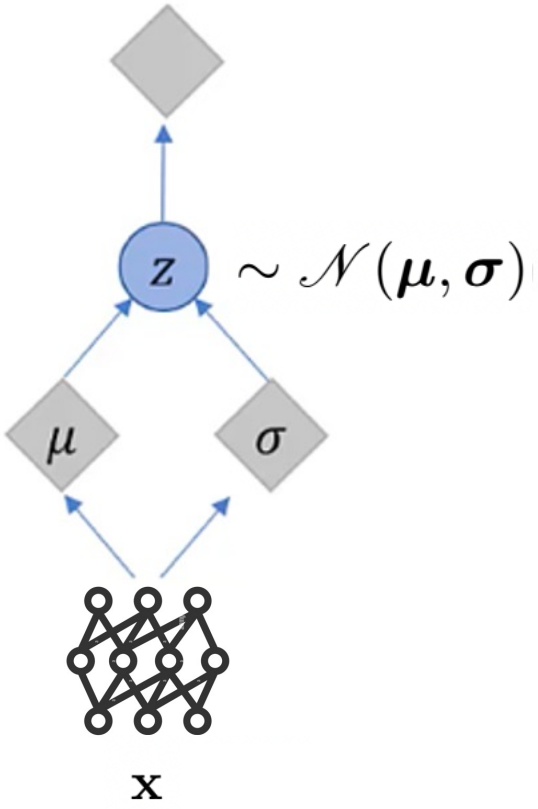
$\mathbf{z} = g_\phi(\boldsymbol{\epsilon}) := \mathbf{A}\boldsymbol{\epsilon} + \mathbf{b}$ où $\phi := (\mathbf{A}, \mathbf{b})$

$\mathbf{z} \sim q_\phi = \mathcal{N}\left(\mathbf{b}, \mathbf{A}\mathbf{A}^\top\right)$

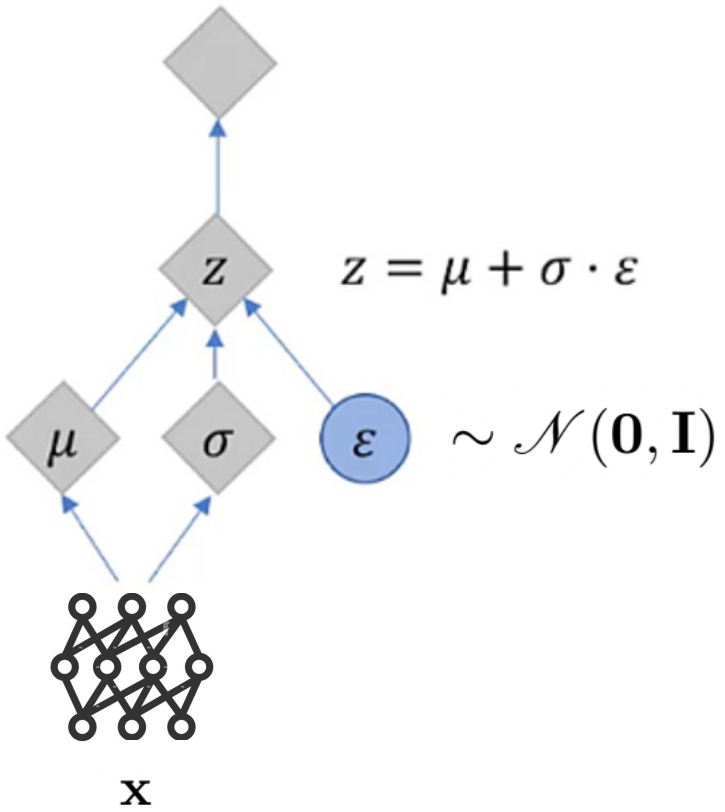
Ce qui permet de calculer le gradient de $\mathcal{L}$ par rapport à ϕ avec MC :

$$\begin{aligned}
\nabla_\phi \mathscr{L}(\mathbf{x}; \theta, q_\phi) &= \nabla_\phi \int [\log p_\theta(\mathbf{x}, g_\phi(\boldsymbol{\epsilon})) - \log q_\phi(g_\phi(\boldsymbol{\epsilon}))] \ \rho(\boldsymbol{\epsilon}) \mathrm{d}\boldsymbol{\epsilon} \\
&= \int \nabla_\phi [\log p_\theta(\mathbf{x}, g_\phi(\boldsymbol{\epsilon})) - \log q_\phi(g_\phi(\boldsymbol{\epsilon}))] \ \rho(\boldsymbol{\epsilon}) \mathrm{d}\boldsymbol{\epsilon} \\
&\approx \frac{1}{K} \sum_{j=1}^{K} \nabla_\phi [\log p_\theta(\mathbf{x}, g_\phi(\boldsymbol{\epsilon}_j)) - \log q_\phi(g_\phi(\boldsymbol{\epsilon}_j))] \quad \text{où} \quad \boldsymbol{\epsilon}_j \sim \mathcal{N}(\mathbf{0}, \mathbb{I})
\end{aligned}$$

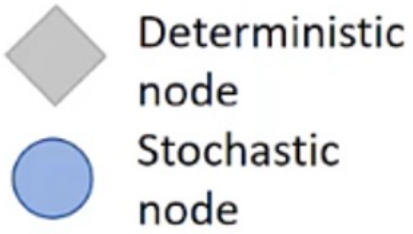
Le « reparameterization trick » (2)



Graphe de calcul original avec un nœud **stochastique** dont les paramètres sont calculés par un RN



Graphe de calcul déterministe avec « reparameterization trick ». Le nœud stochastique est externalisé à l'aide de données fictives échantillonnées selon une loi statique



Maximum de la vraisemblance pour un échantillon

$$\theta_{\mathrm{ML}} = \arg\max_{\theta} \frac{1}{N} \sum_{j=1}^{N} \log p_\theta(\mathbf{x}^{(j)})$$

$$\log p_\theta(\mathbf{x}^{(j)}) > \mathscr{L}(\mathbf{x}^{(j)}; \theta, q_{\phi^{(j)}})$$

où $\phi^{(j)} := \phi_{\mathrm{opt}}(\mathbf{x}^{(j)}) := \arg\max_{\phi} \mathscr{L}(\mathbf{x}^{(j)}; \theta, q_\phi)$

En principe il faudrait résoudre un problème d'optimisation pour calculer $\phi^{(j)} := \phi_{\mathrm{opt}}(\mathbf{x}^{(j)})$ pour chaque nouvelle observation $\mathbf{x}^{(j)}$. Beaucoup trop coûteux !

Solution : on va plutôt apprendre le mapping qui associe les paramètres optimaux à chaque nouvelle observation :

$$\mathbf{x}^{(j)} \rightarrow \phi^{(j)} := \phi_{\mathrm{opt}}(\mathbf{x}^{(j)})$$

Une instanciation d'un modèle à variables latentes : les VAE

$$\max_{\theta} \log p_\theta(\mathbf{x}) \geq \max_{\theta,\phi} \mathscr{L}(\mathbf{x};\theta,q_\phi), \quad \mathscr{L}(\mathbf{x};\theta,q_\phi) = \mathbb{E}_{q_\phi(\mathbf{z})}[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z})] \quad (1)$$

$$= \log p_\theta(\mathbf{x}) - \mathrm{KL}[q_\phi(\mathbf{z})||p_\theta(\mathbf{z}|\mathbf{x})] \quad (2)$$

$$= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{– erreur de reconstruction avec injection de bruit}} - \underbrace{\mathrm{KL}[q_\phi(\mathbf{z})||p_\theta(\mathbf{z})]}_{\text{terme de régularisation qui favorise la proximité de la distribution à postériori avec la distribution à priori choisie.}} \quad (3)$$

Ce qu'il faut choisir pour définir un modèle :

1. Quelles sont les données $\mathbf{x}$?
2. Quelles sont les variables latentes $\mathbf{z}$?
3. Quelle est la **distribution à priori** $p_\theta(\mathbf{z})$ sur les variables latentes $\mathbf{z}$?
4. Quelle est le **modèle d'inférence** = **distribution variationnelle** (encodeur) $q_\phi(\mathbf{z})$ qui approxime $p_\theta(\mathbf{z}|\mathbf{x})$
5. Quelle est le **modèle génératif** (décodeur) $p_\theta(\mathbf{x}|\mathbf{z})$?

Chaque choix conduit à un modèle différent.

Choix de la distribution à priori $p_\theta(\mathbf{z})$ dans l'espace latent

Les données $\mathbf{x}$ dépendent naturellement des applications

Images couleurs : $\mathbf{x} \in \mathbb{R}^d$

Images noir et blanc : $\mathbf{x} \in \{0, 1\}^d$

Une distribution normale $p_\theta(\mathbf{z}) \equiv p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbb{I})$, indépendante des paramètres θ offre les avantages suivants :

1. Des variables latentes continues $\mathbf{z} \in \mathbb{R}^L$ seront plus faciles à échantillonner par reparamétrisation à partir d'une loi continue pour la distribution variationnelle $q_\phi(\mathbf{z})$.
2. Le choix d'une loi normale pour $p(\mathbf{z})$ facilitera le calcul de $\mathrm{KL}[q_\phi(\mathbf{z})||p_\theta(\mathbf{z})]$
3. Le choix d'une loi normale pour $p(\mathbf{z})$ facilitera l'échantillonnage des $\mathbf{x} \sim p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})\, p(\mathbf{z})\, \mathrm{d}\mathbf{z}$

Choix de la distribution variationnelle $q_\phi(\mathbf{z})$: encodeur

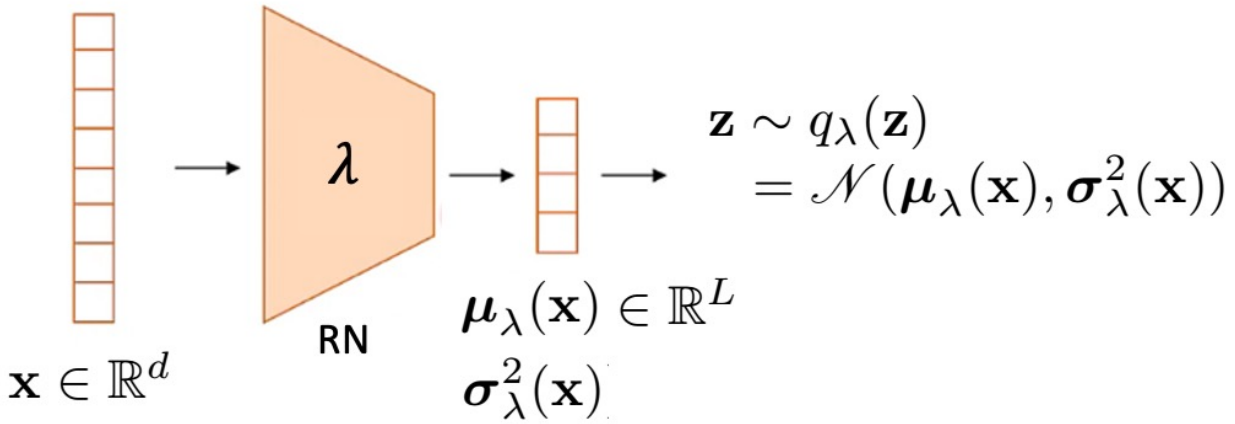
Pour faciliter le calcul du terme $\mathrm{KL}[q_\phi(\mathbf{z})||p_\theta(\mathbf{z})]$ de la fonction de coût (ELBO)

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

rappel : $\mathbf{x} \to \phi_{\mathrm{opt}}(\mathbf{x}) := \left(\boldsymbol{\mu}_\lambda(\mathbf{x}), \boldsymbol{\sigma}^2_\lambda(\mathbf{x})\right)$

on choisit là encore une gaussienne

$$q_{\phi_{\mathrm{opt}}(\mathbf{x})}(\mathbf{z}) := q_\lambda(\mathbf{z}) = \mathscr{N}(\boldsymbol{\mu}_\lambda(\mathbf{x}), \boldsymbol{\sigma}^2_\lambda(\mathbf{x}))$$



Si on choisit $\boldsymbol{\sigma} = \mathrm{diag}[\sigma_1, ..., \sigma_L]$, on vérifie que

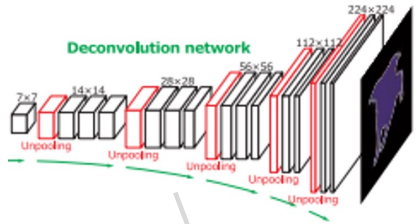
$$\mathrm{KL}[q_\lambda(\mathbf{z})||p(\mathbf{z})] = \frac{1}{2}\sum_{j=1}^{L}[\sigma_{\lambda,j}(\mathbf{x})^2 + \mu_{\lambda,j}(\mathbf{x})^2 - \log\sigma_{\lambda,j}(\mathbf{x})^2 - 1]$$

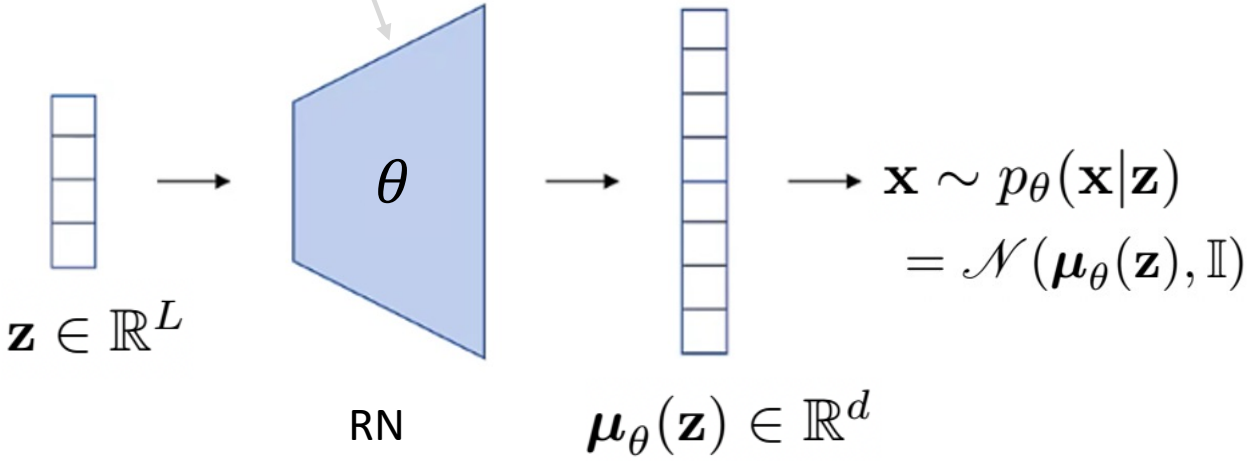
Choix de la distribution conditionnelle $p_\theta(\mathbf{x}|\mathbf{z})$: décodeur

Par exemple pour $\mathbf{x} \in \mathbb{R}^d$

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{z}), \mathbb{I}) = \prod_{j=1}^{d} \mathcal{N}(\mu_{\theta,j}(\mathbf{z}), 1)$$

- Les sont x_j conditionnellement indépendants
- Il faut trouver un compromis entre l'expressivité et le coût du calcul





Par exemple pour $\mathbf{x} \in \{0,1\}^d$

$$p_\theta(\mathbf{x}|\mathbf{z}) = \prod_{j=1}^{d} \text{Bernouilli}(\sigma_{\theta,j}(\mathbf{z}))$$

Récapitulatif : la fonction de coût complète

$$\frac{1}{N}\sum_{j=1}^{N}\mathscr{L}\left(\mathbf{x}^{(j)};\theta,q_{\phi^{(j)}}\right)=\frac{1}{N}\sum_{j=1}^{N}\left\{\mathbb{E}_{q_{\phi^{(j)}}(\mathbf{z}^{(j)})}\left[\log p_\theta\left(\mathbf{x}^{(j)}|\mathbf{z}^{(j)}\right)\right]-\mathrm{KL}\left[q_{\phi^{(j)}}\left(\mathbf{z}^{(j)}\right)\middle\|p\left(\mathbf{z}^{(j)}\right)\right]\right\}$$

Erreur de reconstruction.
Estimée avec MC, souvent
un seul échantillon.

Pénalisation des écarts à la
distribution à priori
dans l'espace latent.
Calculable analytiquement.

où :

à priori : $p(\mathbf{z})=\mathcal{N}(\mathbf{0},\mathbb{I})$

encodeur : $q_\phi(\mathbf{z})=\mathcal{N}(\boldsymbol{\mu}_\lambda(\mathbf{x}),\boldsymbol{\sigma}^2_\lambda(\mathbf{x}))$ paramétrée par $\mathbf{z}=\boldsymbol{\sigma}_\lambda(\mathbf{x})\,\boldsymbol{\epsilon}+\boldsymbol{\mu}_\lambda(\mathbf{x})$ où $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbb{I})$

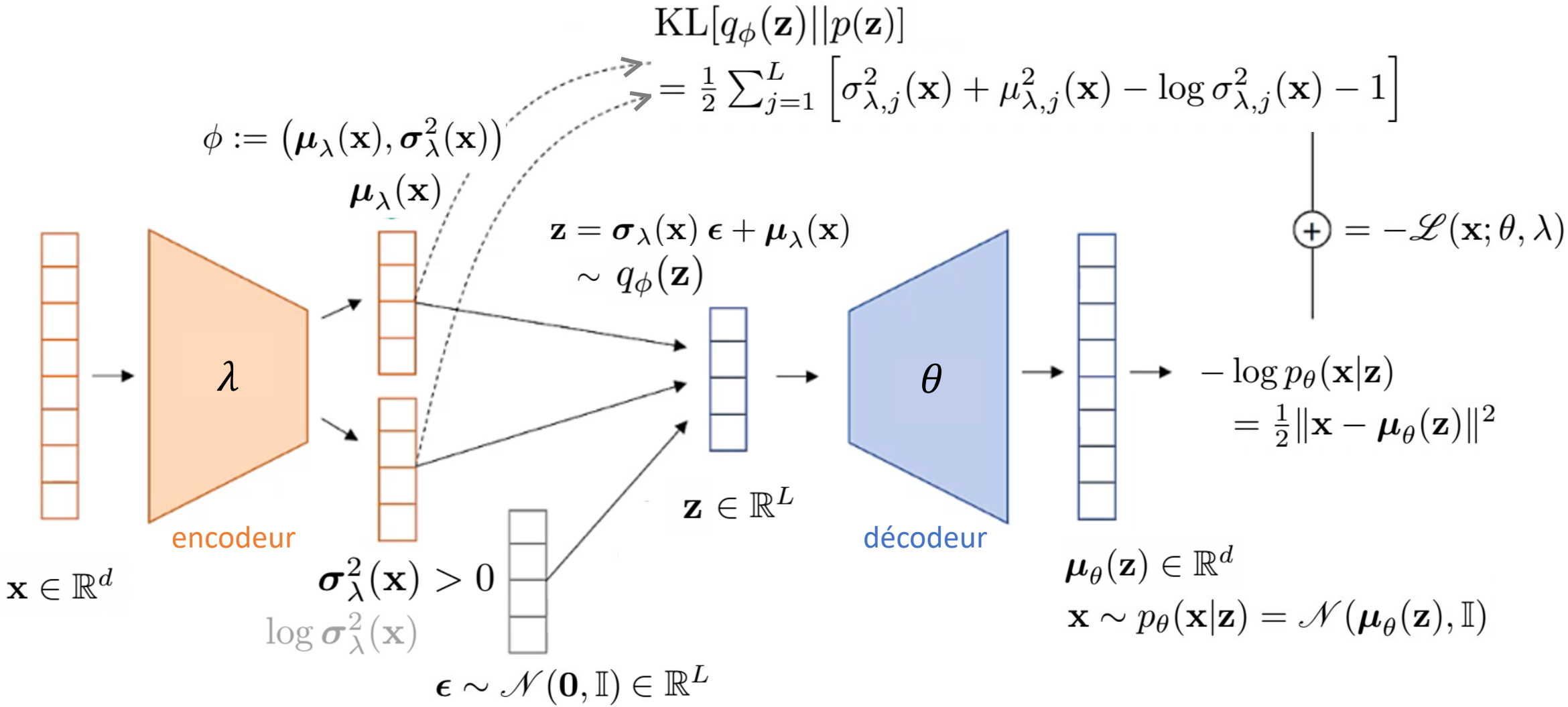
décodeur : $p_\theta(\mathbf{x}|\mathbf{z})=\mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{z}),\mathbb{I})$

Les étapes de l'entraînement d'un VAE (1)

Pour chaque donnée du jeu d'entraînement : $\left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right)$

1. Calculer $\boldsymbol{\mu}_\lambda(\mathbf{x}^{(j)})$ et $\boldsymbol{\sigma}^2_\lambda(\mathbf{x}^{(j)})$ (notés $\phi^{(j)}$ ci-dessous)
2. Estimer $\mathscr{L}$ avec MC :
   a. Échantillonner $\mathbf{z}^{(j)} \sim \mathcal{N}(\boldsymbol{\mu}_\lambda(\mathbf{x}^{(j)}), \boldsymbol{\sigma}_\lambda(\mathbf{x}^{(j)}))$ (avec la reparamétrisation)
   b. Calculer $\boldsymbol{\mu}_\theta(\mathbf{z}^{(j)})$ qui définit $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{z}), \mathbb{I})$
   c. Estimer $\mathscr{L}\left(\mathbf{x}^{(j)}; \theta, q_{\phi^{(j)}}\right) \approx \left[\log p_\theta\left(\mathbf{x}^{(j)}|\mathbf{z}^{(j)}\right)\right] - \mathrm{KL}\left[q_{\phi^{(j)}}\left(\mathbf{z}^{(j)}\right) \middle\| p\left(\mathbf{z}^{(j)}\right)\right]$
3. Calculer $\nabla_\theta \mathscr{L}$ et $\nabla_\lambda \mathscr{L}$ par rétropropagation
4. Mettre à jour les deux paramètres θ et λ

Les étapes de l'entraînement d'un VAE (2)

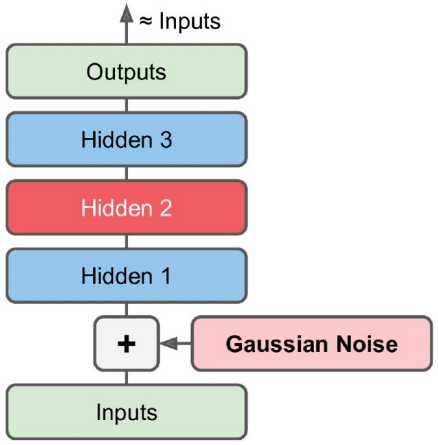


Exemples pour des images générées par un VAE

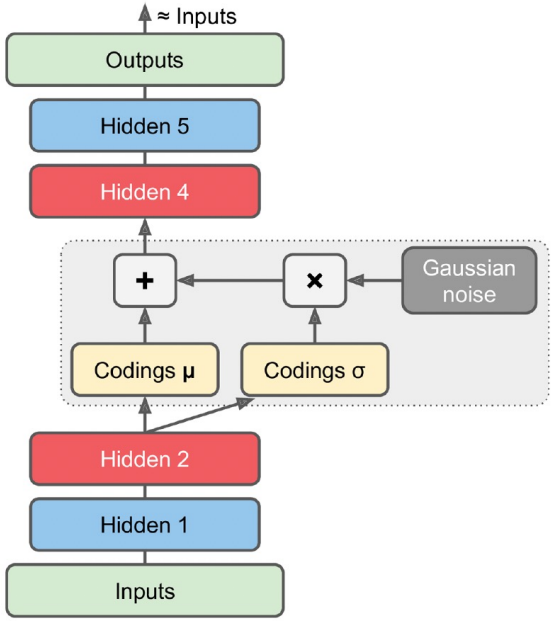
(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

DVAE : Denoising Variational Autoencoder

Idée : ajouter du bruit à l'entrée plutôt que dans l'espace latent. Ou même les deux à la fois ⇒ **DVAE**



DAE



VAE

arXiv > cs > arXiv:1511.06406

Computer Science > Machine Learning

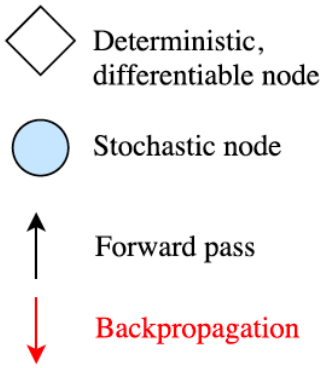
[Submitted on 19 Nov 2015 (v1), last revised 4 Jan 2016 (this version, v2)]

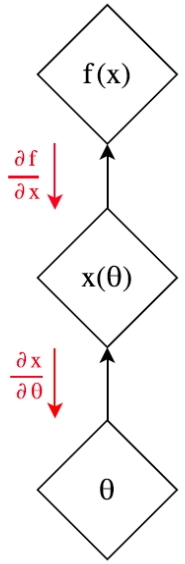
Denoising Criterion for Variational Auto-Encoding Framework

Daniel Jiwoong Im, Sungjin Ahn, Roland Memisevic, Yoshua Bengio

- En bruitant le signal d'entrée et en demandant au DVAE de reconstruire le signal non bruité on améliore les performance du système.
- Revient à choisir une classe de distribution variationnelle $q_\phi(\mathbf{z})$ plus riche que la version gaussienne à un mode des VAE.
- Demande de bien choisir le modèle de bruit.
- Fait partie d'un ensemble de recherche qui vise à augmenter l'expressivité de la distribution variationnelle $q_\phi(\mathbf{z})$

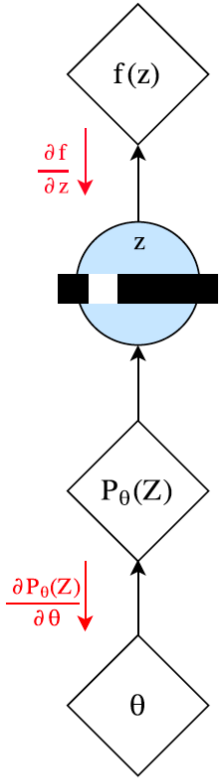
Variables latentes discrètes : le « Gumbel trick »





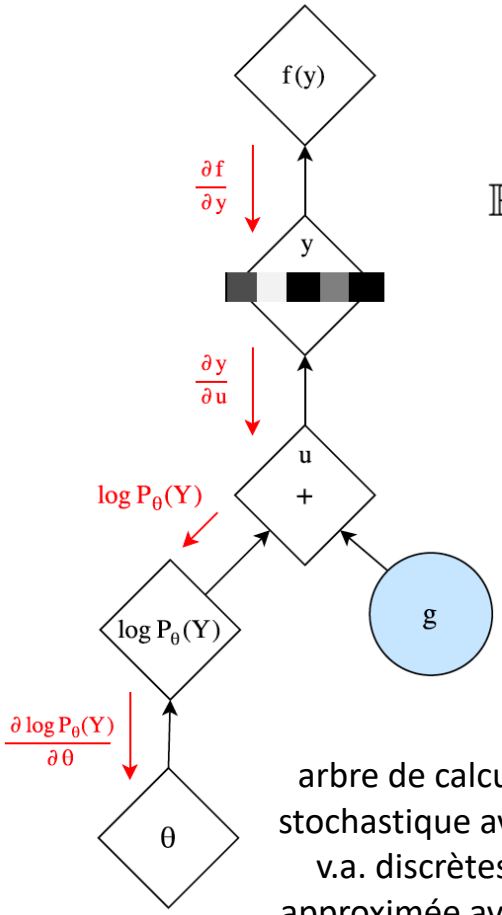
arbre de calcul déterministe

backprop **OK**



arbre de calcul stochastique avec v.a. discrètes

backprop **KO**



arbre de calcul stochastique avec v.a. discrètes approximée avec le Gumbel trick

backprop **OK**

variable catégorielle

$$\mathbb{E}_p[\hat{z}] = [\pi_1, ..., \pi_k]$$

$$\mathbb{R}^k \ni z = \texttt{one\_hot}\left(\arg\max_{i=1,...,k}[g_i + \log \pi_i]\right)$$

$$g = -\log(-\log(\mathrm{u}))$$

$$u \sim \mathrm{Uniform}(0, 1)$$

Loi de Gumbel

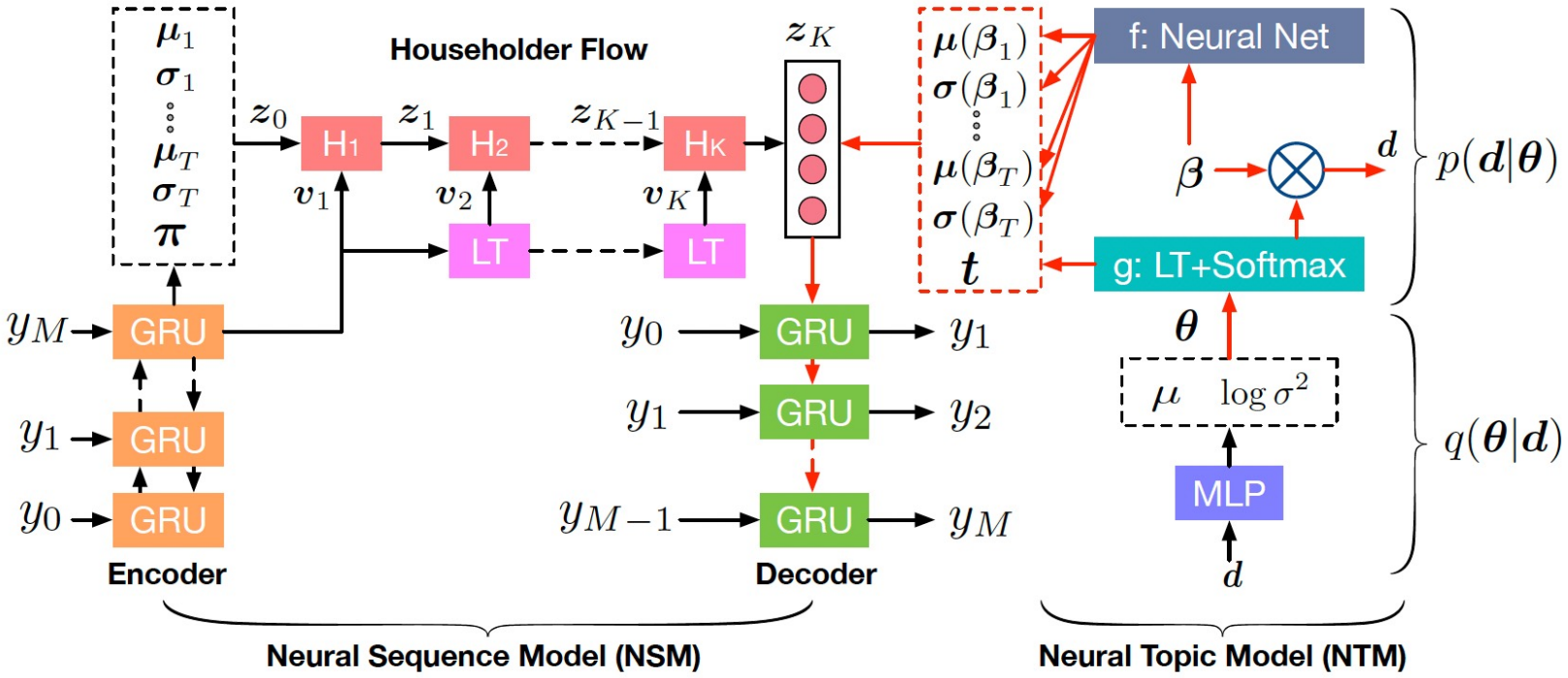
Approximation continue des v.a. one hot

$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\pi_j) + g_j)/\tau)}$$

$$i = 1, ..., k$$

température

TGVAE : une application à la génération de texte



Reconstruit une phrase à partir étant donné un topic model qui correspond à une distribution à priori dans l'espace latent.

Apprend un topic model en reconstruisant les documents conçus comme des BOW.

Idées :

1. Guider la génération de phrases par une collection de thèmes représentés par une distribution à priori qui est un mélange de plusieurs gaussiennes plutôt qu'une seule.
2. VAE : cohérence à long terme, RNN : cohérence à court terme
3. VAE + un « neural topic model » NTM sont entraînés simultanément.
4. Une distribution variationnelle suffisamment flexible est construite à l'aide d'un « Normalizing Flow » spécifique (Householder transfo).

Et encore…

- β-VAE : vise l'interprétabilité des dimensions latentes

$$\mathcal{L}_{\text{BETA}}(\mathbf{x};\theta,\phi,\beta) = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + \beta D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))$$

- VGAE : Variational Graph Auto-Encoders permet un apprentissage non supervisé de représentations pour des graphes avec attributs (c.-à-d. des observations reliées entre elles et donc non i.i.d.)

encodeur

$$\begin{cases} q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} q(\mathbf{z}_i \mid \mathbf{X}, \mathbf{A}) \quad \text{où} \quad q(\mathbf{z}_i \mid \mathbf{X}, \mathbf{A}) = \mathcal{N}(\mathbf{z}_i \mid \boldsymbol{\mu}_i, \operatorname{diag}(\boldsymbol{\sigma}_i^2)) \\ \boldsymbol{\mu} = \operatorname{GCN}_{\boldsymbol{\mu}}(\mathbf{X}, \mathbf{A}) \quad \text{et} \quad \log \boldsymbol{\sigma} = \operatorname{GCN}_{\boldsymbol{\sigma}}(\mathbf{X}, \mathbf{A}) \end{cases}$$

décodeur

$$p(\mathbf{A} \mid \mathbf{Z}) = \prod_{i=1}^{N} \prod_{j=1}^{N} p(A_{ij} \mid \mathbf{z}_i, \mathbf{z}_j) \quad \text{où} \quad p(A_{ij} = 1 \mid \mathbf{z}_i, \mathbf{z}_j) = \sigma(\mathbf{z}_i^\top \mathbf{z}_j)$$

ELBO

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[\log p(\mathbf{A} \mid \mathbf{Z})\right] - \operatorname{KL}\left[q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}) \,||\, p(\mathbf{Z})\right] \quad \text{où} \quad p(\mathbf{Z}) = \prod_i p(\mathbf{z_i}) = \prod_i \mathcal{N}(\mathbf{z}_i \mid 0, \mathbf{I})$$

- Augmenter l'expressivité de la distribution variationnelle $q_\phi(\mathbf{z})$
- Augmenter l'expressivité de la distribution à priori $p_\theta(\mathbf{z})$
- …