TP 8 – Espaces de représentations, Auto-Encodeurs et VAE

Étude sur MNIST

Pierre Chambet

18 novembre 2025

Introduction

Ce TP a pour objectif d'explorer la notion d'*espace latent* à travers deux familles de modèles :

— un auto-encodeur convolutionnel utilisé comme *débruiteur* (bruit ajouté en entrée) ;

— un auto-encodeur variationnel (VAE) où le bruit est déplacé dans l'espace latent.

Les expériences sont réalisées sur la base MNIST de chiffres manuscrits. Nous présentons ici les réponses aux questions du sujet liées au rôle du bruit, à la structure du VAE et au comportement des reconstructions successives.

1 Auto-encodeur débruiteur : bruit en entrée (Questions 4–5)

Question 4 – Rôle du bruit et parties de l'architecture impactées

Dans la première partie du TP, on considère un auto-encodeur convolutionnel entraîné à partir de paires

$$x_{\text{bruite}} \longmapsto x_{\text{propre}}.$$

On injecte donc un bruit additif sur les images d'entrée, alors que la cible reste l'image originale non bruitée.

Implication en termes d'apprentissage. Contrairement à un auto-encodeur classique qui apprend l'identité

$$f_\theta(x) \approx x,$$

l'auto-encodeur débruiteur doit apprendre une application beaucoup plus contrainte :

$$f_\theta(x_{\text{bruite}}) \approx x_{\text{propre}}.$$

Il ne peut pas se contenter de recopier les pixels : il doit extraire des *structures stables* des chiffres (contours, forme globale, contraste chiffre/fond) qui restent reconnaissables malgré le bruit. L'espace latent est donc forcé à capturer des informations plus sémantiques que purement pixel à pixel.

Partie de l'architecture la plus impactée. La partie la plus impactée est l'**encodeur** :

— il reçoit une entrée fortement bruitée ;

— il doit produire une représentation latente z qui soit robuste au bruit, c'est-à-dire qui varie peu quand on modifie le bruit mais pas le chiffre.

Le décodeur, lui, apprend à reconstruire une image propre à partir de cette représentation. Il est bien sûr affecté par le bruit via ce qu'il reçoit en entrée, mais la véritable "pression" d'invariance est portée par l'encodeur.

Intérêt du bruit. Le bruit joue un rôle de **régularisation** :

— il empêche le réseau de mémoriser trivialement chaque image du jeu d'apprentissage ;

— il force l'encodeur à apprendre des caractéristiques *invariantes* au bruit (la forme du chiffre) ;

— au final, l'auto-encodeur devient un *filtre débruiteur non linéaire* capable de reconstruire des chiffres propres à partir d'entrées fortement perturbées.

1.1 Résultats expérimentaux

Après entraînement (20 époques, batch size 512, jeux d'entraînement et de validation de taille 10 000), on obtient des pertes typiques autour de

$$\text{Train} \approx 0{,}021, \qquad \text{Val} \approx 0{,}021,$$

ce qui traduit une bonne convergence sans sur-apprentissage manifeste.

Les figures 1 et 2 illustrent le comportement du modèle sur les ensembles de validation et de test.

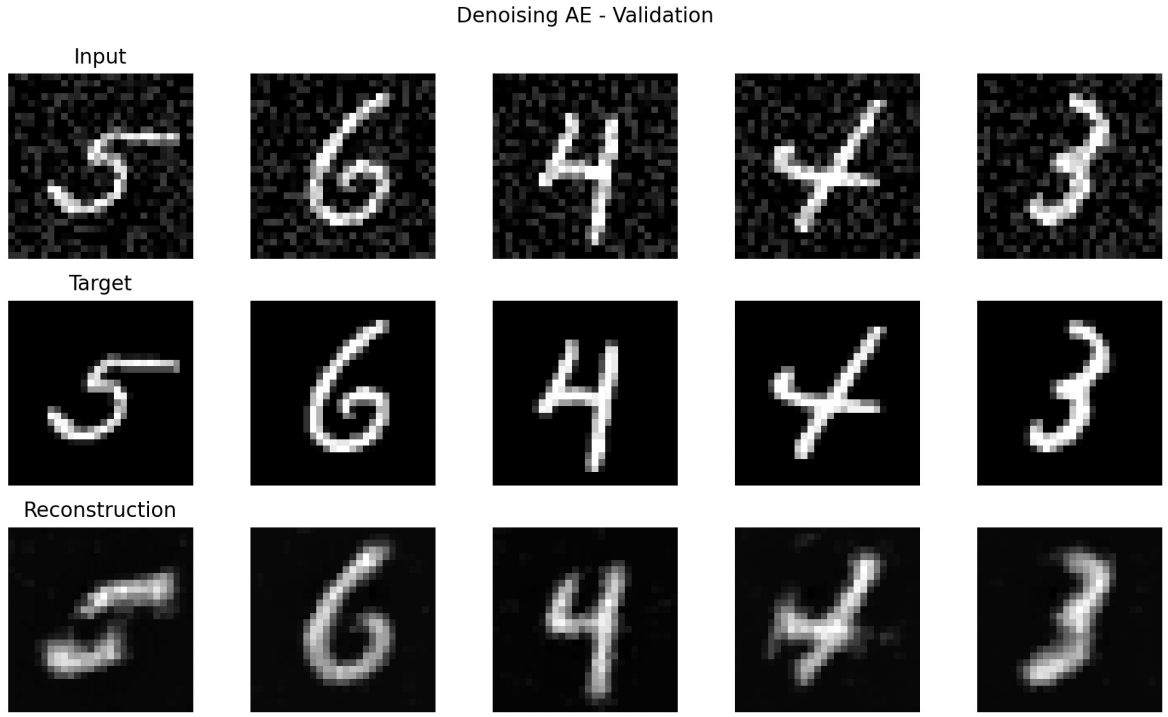

FIGURE 1 – Auto-encodeur débruiteur – Validation. Ligne du haut : entrées bruitées. Ligne du milieu : cibles propres. Ligne du bas : reconstructions.

Question 5 – Effet concret du bruit : analyse des résultats

Les figures obtenues sur la validation et le test montrent trois lignes :

— **Input** : chiffres noyés dans un bruit uniforme important ;

— **Target** : mêmes chiffres sans bruit ;

— **Reconstruction** : sorties de l'auto-encodeur.

Visuellement, les reconstructions sont nettement plus propres que les entrées : le fond est lissé, la structure du chiffre est bien restituée, même si certains détails fins sont légèrement floutés. Les courbes de coût montrent une décroissance de la MSE d'environ 0.12 à 0.02, avec des valeurs train/validation très proches, signe d'une bonne généralisation.

Concrètement, cela montre que :

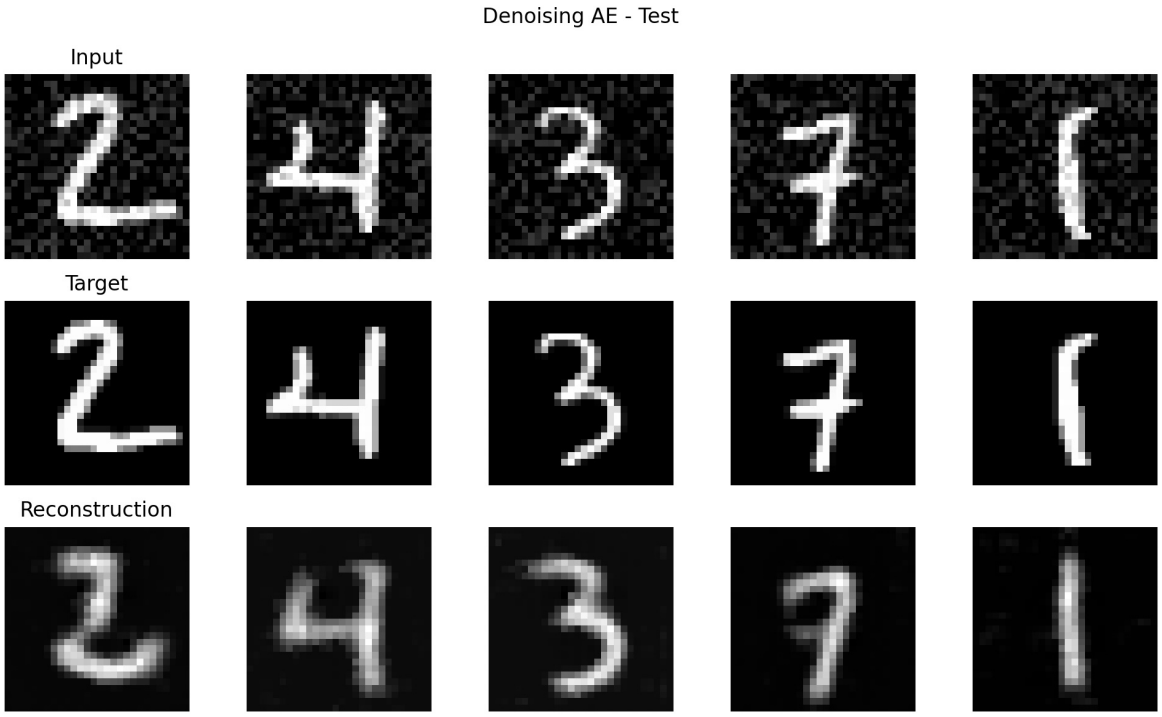

FIGURE 2 – Auto-encodeur débruiteur – Test. Organisation identique : entrées bruitées, cibles propres, reconstructions.

— le modèle a appris à **supprimer la majeure partie du bruit** sans perdre l'identité du chiffre ;

— l'espace latent contient des représentations de chiffres qui sont **invariantes au bruit de bas niveau** ;

— l'auto-encodeur se comporte comme un module de *prétraitement débruiteur* utilisable en amont d'autres tâches (reconnaissance, classification, etc.).

2 VAE : bruit dans l'espace latent (Questions 6–8)

Dans la seconde partie, le bruit n'est plus ajouté sur les pixels, mais dans l'espace latent. À partir d'une image x, l'encodeur produit les paramètres d'une loi gaussienne

$$(\mu(x), \log \sigma^2(x)),$$

et l'on échantillonne

$$z = \mu(x) + \sigma(x)\,\varepsilon, \quad \text{avec } \varepsilon \sim \mathcal{N}(0, I).$$

Le décodeur reconstruit ensuite $\hat{x}$ à partir de z.

Question 6 – Choix du modèle de la figure 2

La figure du sujet présente deux architectures conceptuelles :

— une où l'on sélectionnerait directement un *échantillon* z en coupant le gradient (tirage non différentiable) ;

— une où z est construit via la **reparametrization trick** : $z = \mu + \sigma\varepsilon$ avec ε gaussien indépendant.

Le modèle approprié est le **second**, celui qui utilise la reparametrization trick. En effet, on peut alors voir z comme une fonction différentiable de $(\mu, \log \sigma^2)$ et de ε. À ε fixé, la transformation $(\mu, \log \sigma^2) \mapsto z$ est différentiable, ce qui permet de rétropropager le gradient de la loss jusqu'à l'encodeur. L'autre modèle, qui manipule un tirage discret non reparamétré, empêcherait la rétropropagation classique du gradient.

Question 7 – Positionnement de l'encodage et de l'apprentissage de (μ, σ^2)

Par rapport à un auto-encodeur déterministe :

— l'encodeur ne produit plus un vecteur latent unique $z = f_\theta(x)$, mais les **paramètres** d'une distribution latente $q_\theta(z \mid x)$, en pratique une gaussienne diagonale ;

— cette opération est implémentée par deux couches linéaires (ou convolutives suivies de linéaires) :

$$\mu(x) = W_\mu h(x) + b_\mu, \qquad \log \sigma^2(x) = W_{\text{logvar}} h(x) + b_{\text{logvar}},$$

où $h(x)$ est la représentation fournie par l'encodeur convolutionnel.

L'apprentissage de μ et σ^2 est donc intégré dans la même mécanique que celle déjà vue : il s'agit simplement de couches supplémentaires, optimisées par rétropropagation. La nouveauté est que leur rôle est probabiliste : elles paramètrent une famille de distributions latentes $q(z \mid x)$ qui doit, en même temps, permettre une bonne reconstruction et rester proche d'une loi a priori $p(z)$ (souvent $\mathcal{N}(0, I)$).

Question 8 – Spécificités de cet apprentissage

L'apprentissage d'un VAE se distingue de celui d'un auto-encodeur classique sur plusieurs points clefs :

1. Fonction de coût. La loss totale est la somme de deux termes :

$$\mathcal{L}(x) = \underbrace{\mathbb{E}_{q(z|x)}\left[\|x - \hat{x}\|^2\right]}_{\text{reconstruction}} + \beta \underbrace{\mathrm{KL}\big(q(z \mid x) \,\|\, p(z)\big)}_{\text{régularisation de l'espace latent}} .$$

Le terme de reconstruction encourage $\hat{x}$ à être proche de x, tandis que le terme de Kullback–Leibler pousse $q(z \mid x)$ à rester proche d'une gaussienne standard. Le coefficient β contrôle le compromis entre fidélité de reconstruction et régularisation de l'espace latent.

2. Bruit dans l'espace latent. Le bruit n'est plus "physique" sur l'image, mais *structurel* dans la représentation. À chaque passage, on échantillonne un z différent pour la même image x. Cela a plusieurs conséquences :

— les reconstructions sont intrinsèquement **floues** : le modèle reconstruit une version moyenne plausible plutôt qu'un duplicata pixel à pixel ;

— l'espace latent devient **continu et régulier** : des points proches dans z produisent des chiffres similaires.

3. Aspect génératif. Une fois le VAE entraîné, on peut générer de nouvelles images en tirant directement $z \sim p(z) = \mathcal{N}(0, I)$ et en l'envoyant dans le décodeur. Dans nos expériences, les échantillons générés ressemblent à un "chiffre moyen" flou, ce qui illustre deux points :

— le modèle a bien appris à associer des gaussiennes centrées autour de la zone de l'espace latent qui code pour un "digit typique" ;

— mais avec un latent de dimension modérée et un nombre d'époques limité, la diversité des chiffres n'est pas encore bien séparée : on observe un *mode collapse* partiel.

Les valeurs de coût observées (par exemple une loss de validation totale autour de 0.07, avec une composante de reconstruction ≈ 0.068 et un KL ≈ 0.004) montrent que le terme de KL est présent mais relativement petit : l'espace latent est régularisé, mais reste dominé par la reconstruction.

3 Reconstructions successives (Questions 12–13)

Question 12 – Images reconstruites de reconstructions

On peut considérer l'opération "reconstruction" comme une application R qui prend une image en entrée et produit sa reconstruction $R(x)$. La question revient à regarder $R(R(x))$, $R(R(R(x)))$, etc.

Pour l'auto-encodeur débruiteur, appliquer plusieurs fois R tend à :

— lisser progressivement l'image ;

— supprimer les derniers résidus de bruit présents dans la première reconstruction ;

— stabiliser la forme du chiffre autour d'une version "idéale" (celle que le réseau reconstruit naturellement).

Autrement dit, R se comporte comme une application contractante qui attire les images vers un sous-espace de chiffres propres "typés" par le modèle.

Pour le VAE, la situation est encore plus marquée : à chaque passage, on rééchantillonne (au moins en principe) un latent $z = \mu + \sigma\varepsilon$. Si l'on effectue plusieurs reconstructions successives en réinjectant la sortie comme nouvelle entrée, l'image est progressivement projetée vers ce que le modèle considère comme un *échantillon typique* de sa distribution. Dans nos images, cela se traduit par une convergence vers des chiffres de plus en plus flous et moyens.

3.1 Reconstructions et génération

Les reconstructions sur le jeu de validation sont illustrées figure 3. On constate que les chiffres reconstruits sont reconnaissables mais sensiblement plus flous que dans le cas de l'auto-encodeur débruiteur.

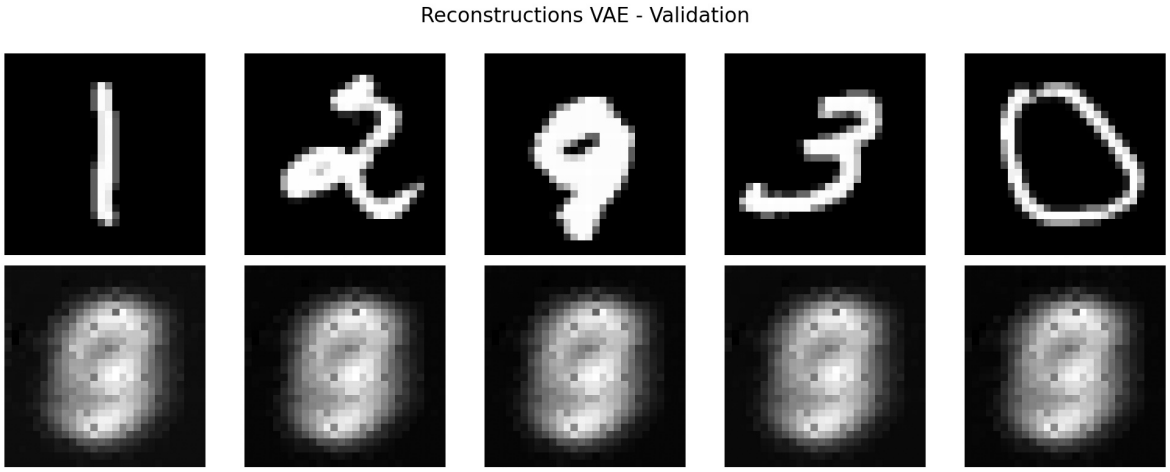

FIGURE 3 – Reconstructions du VAE sur MNIST. Ligne du haut : images originales. Ligne du bas : reconstructions $\hat{x}$.

La figure 4 présente des échantillons générés à partir de $z \sim \mathcal{N}(0, I)$. Tous les chiffres générés se ressemblent fortement, ce qui est cohérent avec un VAE simple ayant une dimension latente modérée et une régularisation KL importante.

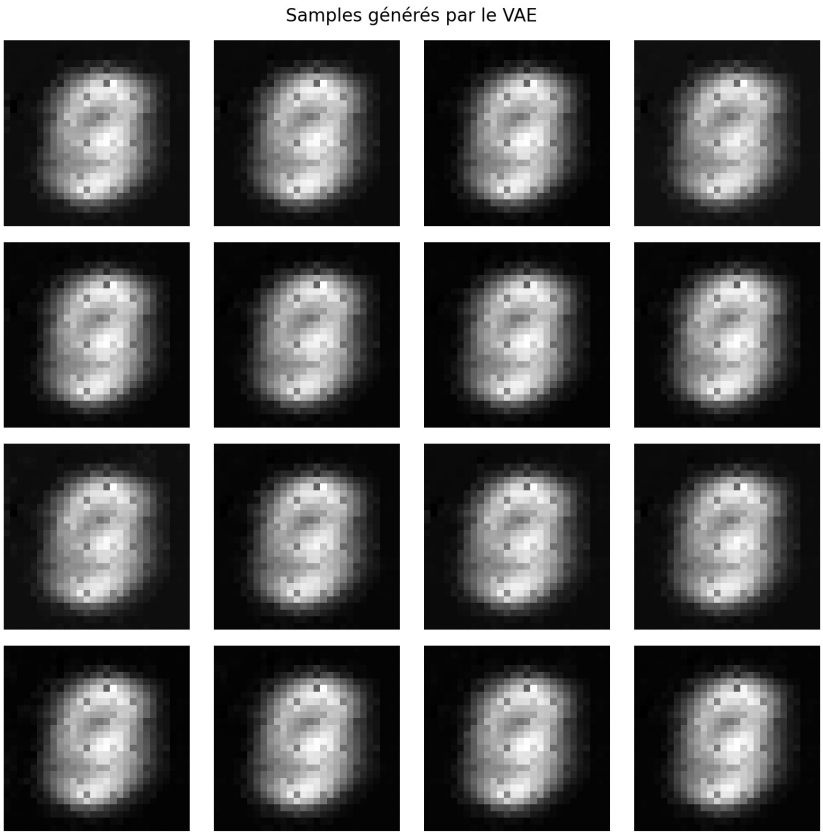

FIGURE 4 – Échantillons générés par le VAE en tirant $z \sim \mathcal{N}(0, I)$. On observe un "chiffre moyen" flou, typique d'un VAE fortement régularisé.

Question 13 – Effet de l'enchaînement des reconstructions

En enchaînant les reconstructions (AE ou VAE), on observe numériquement que :

— les détails idiosyncratiques de l'image initiale (traits particuliers d'un chiffre manuscrit) disparaissent peu à peu ;

— l'image converge vers une sorte de *prototype* du chiffre pour ce modèle, c'est-à-dire une forme moyenne qu'il sait reconstruire le plus facilement ;

— dans le cas du VAE, le flou augmente : le modèle projette l'entrée vers une région de forte probabilité dans $p(x)$, au prix d'une perte d'information fine.

D'un point de vue plus théorique, on peut voir l'opérateur de reconstruction comme une dynamique itérée $x_{k+1} = R(x_k)$ qui possède des attracteurs (images stables ou quasi-stables). Le TP illustre expérimentalement ce phénomène : les reconstructions successives révèlent à quoi ressemblent les "points fixes" du modèle dans l'espace des images.

Conclusion

Ce TP montre concrètement comment :

— un auto-encodeur débruiteur apprend des représentations latentes robustes au bruit de bas niveau et peut être utilisé comme filtre de prétraitement ;

— un VAE introduit un bruit contrôlé dans l'espace latent, ce qui permet un modèle génératif probabiliste au prix de reconstructions plus floues ;

— les reconstructions successives révèlent la structure interne du modèle : prototypes de chiffres, zones de forte probabilité et attracteurs de la dynamique de reconstruction.

Ces expériences donnent une première intuition de la façon dont les espaces latents sont structurés et utilisés dans les modèles génératifs modernes.