université
PARIS-SACLAY

FACULTÉ
DES SCIENCES
D'ORSAY

Mathématiques
Orsay

Variational Auto-encoding Bayes

AUTEURS

Mohamed Amine GRINI
Marine VIEILLARD

Table des matières

1 Introduction

Les autoencodeurs variationnels (VAEs) [5] permettent une inférence efficace dans les modèles probabilistes à variables latentes continues. L'algorithme Auto-Encoding Variational Bayes (AEVB) utilise l'estimateur Stochastic Gradient Variational Bayes (SGVB) pour optimiser une approximation de la distribution a posteriori via descente de gradient stochastique. Cette approche élimine le besoin de méthodes coûteuses comme MCMC. Lorsqu'un réseau de neurones est utilisé, on obtient un VAE, utile pour l'apprentissage de représentations, la réduction de bruit et la visualisation des données. Ce rapport a pour but d'expliquer les enjeux et appuis du VAE ainsi que de montrer une implémentation numérique du modèle pour évaluer sa performance sur deux jeux de données : CIFAR-10 et MNIST.

Lien du repository GitHub : https://github.com/MV-13/Master-2-Variational-Auto-Encoder

2 Résumé

Ce rapport porte sur l'étude et l'implémentation des autoencodeurs variationnels (VAE). Il s'agit d'une architecture fondée sur l'inférence variationnelle et le reparametrization trick. L'objectif est de rendre l'apprentissage des modèles probabilistes profonds tractable en approximant la distribution a posteriori par une famille paramétrée et optimisable via descente de gradient stochastique. Après une présentation théorique de l'inférence variationnelle (ELBO, SGVB, AEVB), nous avons reproduit les taches de reconstruction et génération sur les jeux de données MNIST et CIFAR-10. Sur MNIST, le VAE apprend efficacement une représentation latente continue, permettant des reconstructions lisibles et des générations plausibles. Sur CIFAR-10, les résultats confirment les limites des VAE standards : les images générées restent floues et manquent de détails, conséquence du compromis reconstruction–régularisation imposé par l'ELBO et de l'a priori gaussien. Nous avons également regardé l'impact de la dimension de l'espace latent, nous avons pu observer des résultats similaires à ceux présentés dans l'article : un espace latent réduit (2D) produit des chiffres facilement interprétables mais souvent flous, tandis qu'un espace latent plus large (20D) génère des images plus nettes mais moins structurées. Enfin, ce travail met en évidence les limites des VAE et ouvre sur des variantes modernes comme β-VAE, VQ-VAE, NVAE ou encore les modèles de diffusion, aujourd'hui références en génération d'images.

3 Recherches bibliographiques

Cet article datant de décembre 2012 est dans le développement des autoencodeur qui datent des années 1980. Avant sa publication, des approches telles que l'inférence variationnelle classique et les méthodes MCMC étaient utilisées, mais celles-ci étaient souvent trop coûteuses et instables pour être appliquées avec des grands réseaux de neurones. Cet article est considéré comme fondateur pour les autoencodeurs variationnels, en introduisant le reparameterization trick et l'algorithme SGVB. Il rend

pour la première fois compatible l'estimation du gradient de la borne inférieure variationnelle avec les méthodes d'optimisation par descente de gradient comme Adam.

Suite à la publication de cet article, des variantes ont été mises en places telles que les β-VAE en 2017 [3], qui tente d'augmenter l'interprétabilité du modèle. En effet, dans cet article, il est montré que la fonction de coût inclut un terme de divergence KL qui agit comme un régularisateur. Un hyperparamètre β est alors introduit pour contrôler ce terme, forçant ainsi le modèle à apprendre des représenttions désintriquées, c'est-à-dire où chaque neurone du code latent contrôle un aspect précis de l'image (ex : la couleur, la rotation, la taille), rendant alors le modèle plus interprétable. Un autre article publié en 2017 vient à la suite des VAE, il s'agit de l'article sur les Vector Quantized VAE [1], qui utilise des variables latentes discrètes au lieu des variables latentes continues. Cela a permis de générer des images beaucoup plus nettes et a servi de base aux premiers grands modèles génératifs d'images, comme par exemple la première version de DALL-E [7]. Plus récemment, en 2021, a été proposé un modèle hiérarchique : les NVAE [2]. Il s'agit d'une architecture VAE très profonde et hiérarchique, capable de générer des visages humains réalistes rivalisant pour la première fois avec les GANs, tout en gardant la stabilité d'entraînement des VAEs.

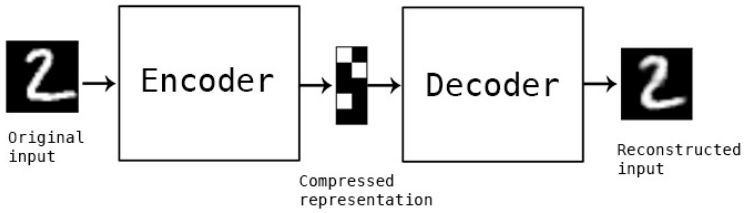
Cet article a non seulement permis des avancées en faisant des variantes plus performantes, mais a aussi permis la création d'une autre architecture qui est l'une des plus utilisées aujourd'hui pour la génération d'images. En effet, en introduisant le reparameterization trick qui permet l'optimisation de l'ELBO, cela a inspiré la création des modèles de diffusion. Ces derniers peuvent être interprétés comme des VAEs hiérarchiques avec une profondeur infinie et un encodeur fixe, dont l'optimisation repose sur la même borne variationnelle que celle introduite dans l'article sur les VAE. Les modèles de diffusion ont été une avancée majeure de ces dernières années, et sont nottament utilisés pour la génération d'image avec DALL-E 3 ou encore Midjourney.

4 Résultats et preuves

4.1 Objectifs et principaux résultats

L'objectif de l'article est de proposer une méthode pour entraîner des modèles génératifs profonds avec des variables latentes continues, dans un contexte où la vraisemblance marginale et la postérieure sont intractables. Les auteurs cherchent à résoudre trois limitations majeures : la difficulté à calculer ou différencier la vraisemblance marginale, la nécessité d'utiliser des approximations analytiques restrictives dans l'inférence variationnelle classique, et l'impossibilité de faire de l'apprentissage efficace par gradient dans des modèles probabilistes profonds. L'article introduit alors les contributions clés suivantes : le Stochastic Gradient Variational Bayes estimator (SGVB) qui est une formulation différentiable de la borne variationnelle (ELBO), ainsi que le reparameterization trick qui permet de propager des gradients à travers l'échantillonnage de variables latentes. Ces deux contributions permettent aux auteurs de proposer l'algorithme AEVB, qui entraîne conjointement un encodeur probabiliste $q_\phi(z|x)$ et un décodeur génératif $p_\theta(x|z)$, donnant ainsi le Variational Autoencoder (VAE). Les expériences qu'ils ont faites sur MNIST et Frey Faces montrent que cette approche converge rapidement, apprend

des représentations latentes continues structurées, et constitue une alternative performante au Monte Carlo EM qui ne peut pas être utilisé lorsque la densité postérieure est intractable, et serait trop lent appliqué sur des grands datasets.



4.2 Preuves

Nous introduisons d'abord la notion de divergence KL *(Kullback et Leibler, 1951)* [8], qui sert à calculer la distance entre deux distributions :

$$D_{KL}(P||Q) = \int P(x) log(\frac{P(x)}{Q(x)})\, dx = \mathbb{E}_P \left[log \left(\frac{P(x)}{Q(x)} \right) \right] \tag{1}$$

L'objectif en premier temps est de recréer les données d'entrées x, ce qui revient à maximiser la vraisemblance marginale $\log p_\theta(x^{(1)}, \ldots, x^{(N)}) = \sum_{i=1}^{N} \log p_\theta(x^{(i)})$. Pour faire cela, nous maximisons une borne inférieure à $\log p_\theta(x^{(i)})$. Nous nous retrouvons donc avec : $\log p_\theta(\mathbf{x}) \geq \mathcal{L}(\theta, \phi, x)$ où $\mathcal{L}(\theta, \phi, x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p_\theta(z))$:

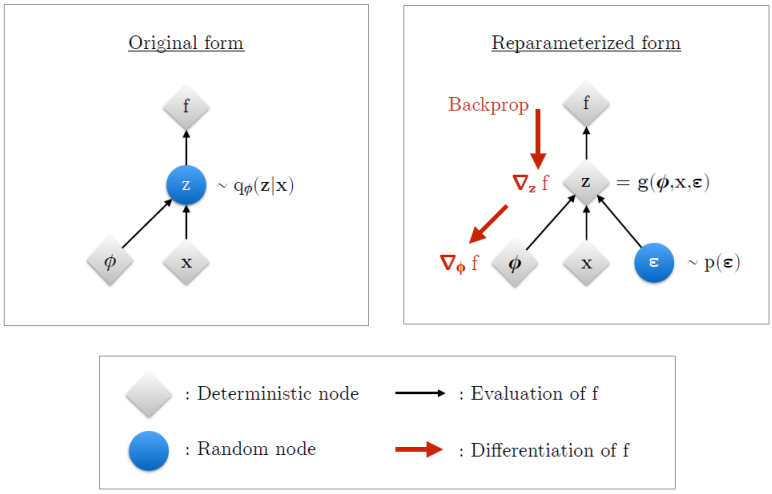
- $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ est le terme de qualité de reconstruction qu'il faut maximiser
- $D_{KL}(q_\phi(z|x)||p_\theta(z))$ est la distance entre l'a posteriori approximé et l'a posteriori réel qui doit être minimisé

Nous souhaitons différencier et optimiser la borne inférieure $\mathcal{L}(\theta, \phi, x)$ par rapport aux paramètres variationnels ϕ et aux paramètres génératifs θ. La divergence KL peut être calculée analytiquement. Soit J la dimensionnalité de z. Soient μ et σ la moyenne variationnelle et l'écart-type, et soient μ_j et σ_j le j-ième élément de ces vecteurs.

$$-KL(q_\phi(z|x)||p_\theta(z|x)) = \frac{1}{2} \sum_{j=1}^{J} \left(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j^2) \right)$$

L'erreur de reconstruction attendue $\mathbb{E}_{q_\phi(z|x)}\left[log(p_\theta(x|z))\right]$ nécessite une estimation par échantillonnage. Nous utiliserons l'échantillonnage de Monte-Carlo *(Metropolis et al., 1953)*[6] : $\mathbb{E}_{q_\phi(z|x)}\left[f(z)\right] \approx \frac{1}{L} \sum_{l=1}^{L} f(z^{(l)})$ où les $z^{(l)}$ sont des variables issues de la distribution variationnelle et L le nombre d'échantillons Monte-Carlo. En calculant le gradient de l'estimateur Monte-Carlo, nous obtenons : $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[f(\mathbf{z})] \simeq \frac{1}{L} \sum_{l=1}^{L} f(\mathbf{z}) \nabla_{q_\phi(\mathbf{z}^{(l)}|x)} \log q_\phi(\mathbf{z}^{(l)}|x)$. Cet estimateur de gradient présente une variance très élevée (*Jordan et al., 2012*)[4]. D'autre part, la rétropropagation ne pourra pas avoir lieu

pendant l'entraînement parce que cela nécessite de calculer $z \sim q_\phi(z|x)$ qui est une variable stocastique non différentiable [5]. Une reparamétrisation est alors proposée pour contourner ce problème. Cette astuce sert à faire tourner la rétropropagation en rendant les variables latentes z déterministes grâce à une transformation différentiable $z = g_\phi(\epsilon, x)$ où ϵ est un bruit auxiliaire.



Le choix de g_ϕ et ϵ dépend de $q_\phi(z|x)$ et se fait comme suit :

— Si la fonction de répartition de $q_\phi(z|x)$ est inversible, alors $\epsilon \sim \mathcal{U}(0, I)$ et $g_\phi(\epsilon, x)$ est la fonction de répartition inverse de $q_\phi(z|x)$. Exemple de loi : Exponentielle, Cauchy, Logistique, Rayleigh, Pareto.

— Pour toute distribution "location-scale", nous pouvons choisir la distribution standard (avec location = 0, scale = 1) comme variable auxiliaire ϵ, et soit $g(.)$= location + scale $\cdot\epsilon$. Exemples : distributions de Laplace, elliptique, t de Student, logistique, uniforme, triangulaire et gaussienne.

— Composition : Il est souvent possible d'exprimer des variables aléatoires sous forme de différentes transformations de variables auxiliaires. Exemples : distributions log-normale (exponentiation d'une variable normalement distribuée), distribution gamma (somme de variables exponentiellement distribuées), distribution de Dirichlet (somme pondérée de variables gamma), distribution bêta, khi-deux et distribution F.

Nous obtenons donc un estimateur empirique de la borne inférieure appelé *Stochastic Gradient Variational Bayes* :

$$\tilde{\mathcal{L}}(\theta, \phi, x^{(i)}) = -KL(q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)})) + \frac{1}{L}\sum_{l=1}^{L}(\log(p_\theta(x^{(i)}|z^{(i,l)})))$$

Où $\epsilon \sim p(\epsilon)$ et $z^{(i,l)} = g_\phi(\epsilon^{(i,l)}, x^{(i)})$.

Pour minimiser le coût computationnel, nous utiliserons des minibatchs (de taille M) de données au lieu de la totalité d'un seul coup :

$$\mathcal{L}(\theta, \phi, X) \simeq \tilde{\mathcal{L}}^M(\theta, \phi, X^M) \simeq \frac{N}{M} \sum_{i=1}^{M} \tilde{\mathcal{L}}(\theta, \phi, X^M)$$

Maintenant qu'on a établi la version Mini-Batch du SGVB, nous utilisons l'algorithme *Auto-Encoding Variational Bayes* pour trouver les paramètres variationnels ϕ et les paramètres génératifs θ *(Voir Annexe)*.

5 Implémentation sur des données réelles

Les auteurs de l'article ne donnant pas accès à leur code, pour reproduire les résultats de l'article, nous nous sommes appuyés sur le repo git suivant pour faire notre implémentation :

https ://github.com/AntixK/PyTorch-VAE. Nous avons crée une interface graphique grâce à Streamlit où le VAE a été implémenté et est évalué sur deux jeux de données de référence en vision par ordinateur : MNIST et CIFAR-10. L'objectif était de démontrer la capacité du modèle à apprendre des représentations latentes structurées et à effectuer à la fois la reconstruction et la génération de données. Le modèle est évalué à l'aide de la fonction de coût MSE sur 10 epochs.

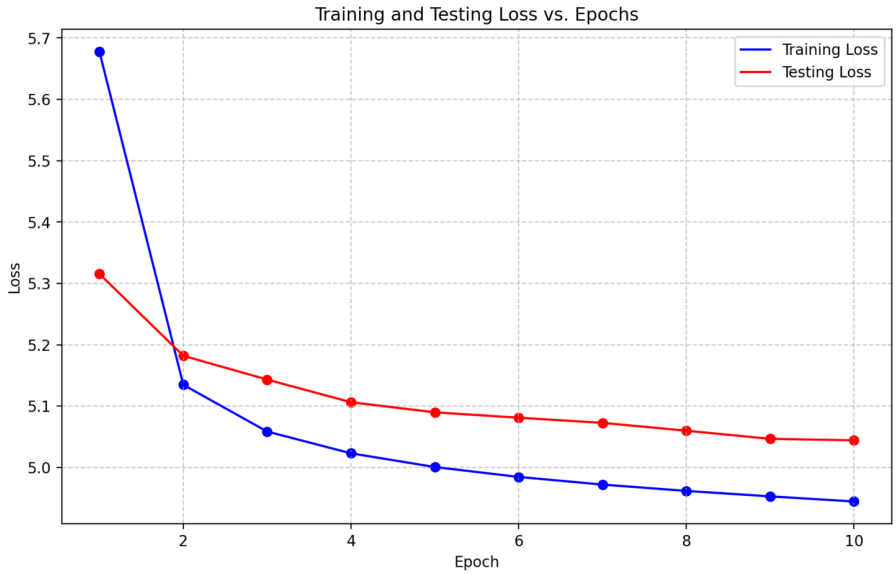


FIGURE 1 – Graphe d'évolution de la loss d'entraînement et de test sur MNIST

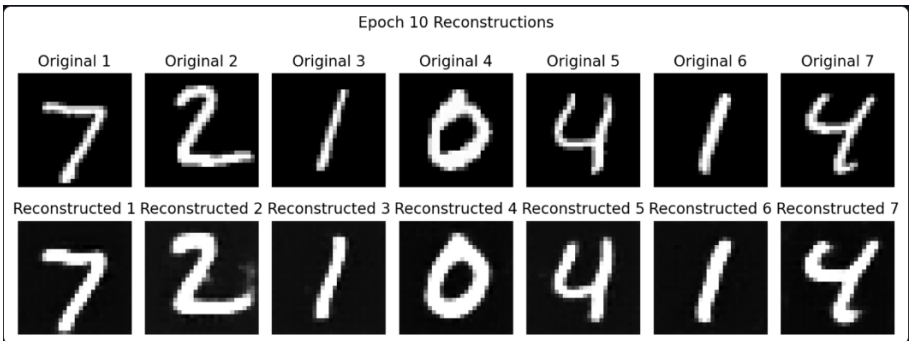


FIGURE 2 – Reconstuction au bout de l'epoch 10

L'implémentation sur MNIST confirme la validité de l'approche VAE pour l'apprentissage de représentations et la reconstruction. L'utilisation de CIFAR-10 permet d'explorer l'efficacité du modèle face à des données visuelles plus complexes, un domaine où les variantes du VAE ont apporté des améliorations significatives.

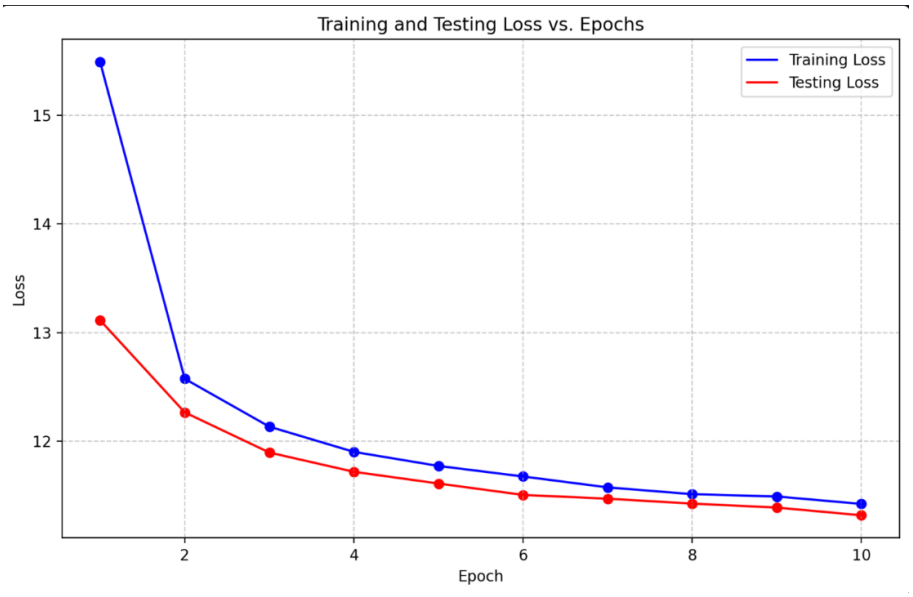
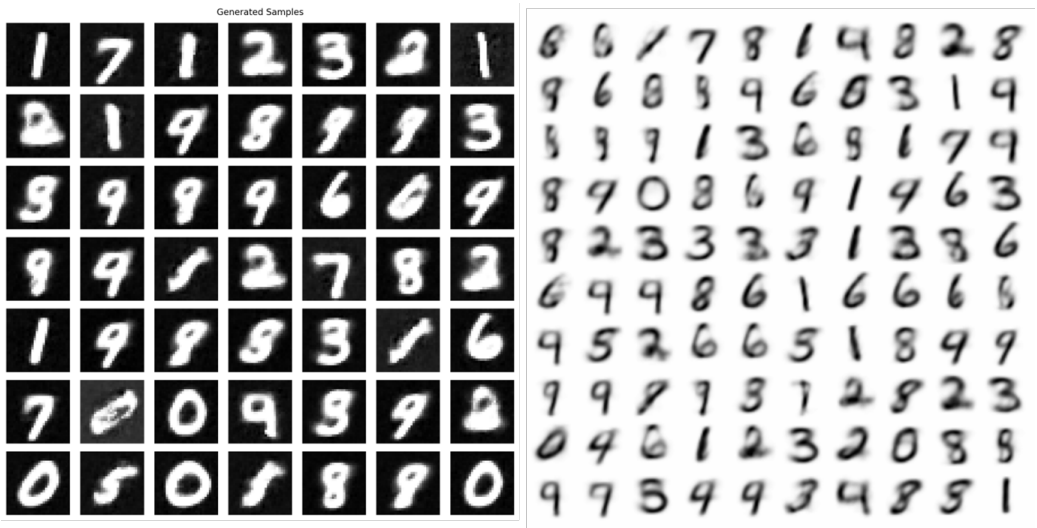


FIGURE 3 – Graphe d'évolution de la loss d'entraînement et de test sur CIFAR-10



a) Our implementation b) Article result

FIGURE 5 – Résultat génération pour un espace latent à 2 dimensions

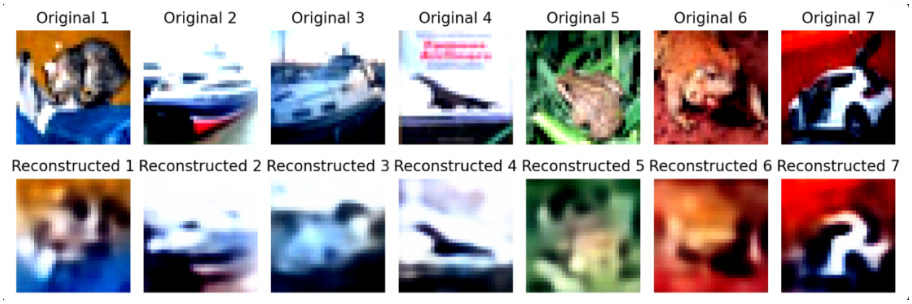
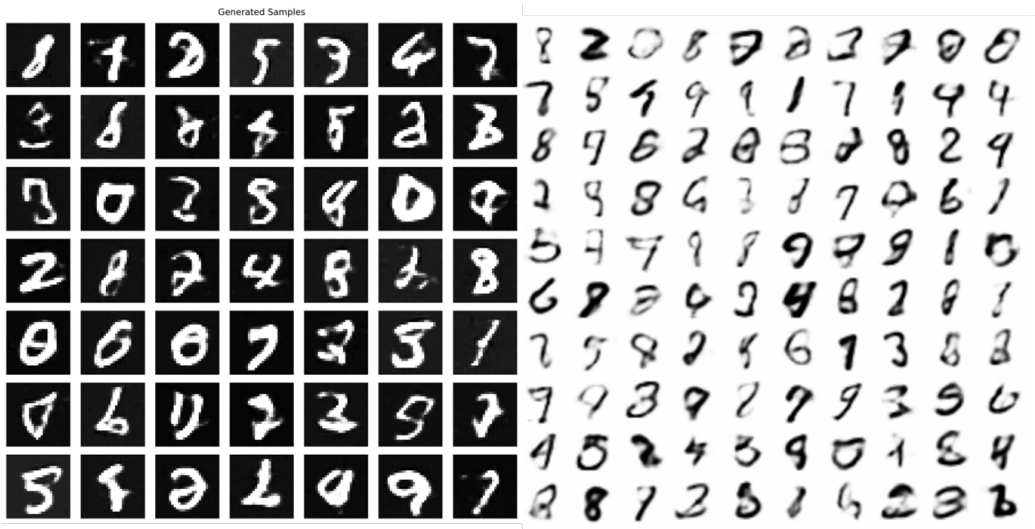


FIGURE 4 – Reconstuction au bout de l'epoch 10

Ces résultats montrent qu'après 10 epochs d'entraînement, le VAE parvient à capturer les formes globales et les couleurs dominantes, mais les reconstructions restent très floues.

Nous avons ensuite observé les capacités de génération d'image de notre VAE. Pour ceux-là, nous avons entrainé le modèle sur 25 epochs, pour un espace latent de dimension 2 (figure 5) et pour un espace latent de dimension 20 (figure 6) de sorte à pouvoir comparer nos résultats avec ceux présentés dans l'article.

Ce résultat pour un espace latent à 2 dimensions montre que le VAE a bien appris à représenter les structures des chiffres de MNIST. Nous parvenons à reconnaitre la classe de la plupart des images générées, et quelques unes de ces images présentent tout de même des problèmes de structures ou de déformation. Nous pouvons voir que les images générées par notre implémation sont presque aussi qualitatives que celles présentées par l'article. Elles présentent une structure et un flou similaire, bien



a) Our implementation b) Article result

FIGURE 6 – Résultat génération pour un espace latent à 20 dimensions

que nous ayons plus d'images présentant une structure moins bonne. Concernant les résultats pour un espace latent à 20 dimensions, nous pouvons voir tant dans les résultats de l'article que dans les notre, une dégradation des performances de génération. Le flou est moins fort mais les structures sont moins claires et les images plus difficiles à classer. Nous ne pouvons pas comparer autrement que visuellement ces résultats étant donné que nous ne connaissons pas en détail l'architecture et les paramètres d'entraînement qu'ils ont utilisé.

6 Commentaires

Nos tests, notamment sur CIFAR-10, attestent des contraintes inhérentes aux VAEs traditionnels : les images reconstituées présentent un flou distinctif et sont moins détaillées que les originales, ce qui est le résultat de l'optimisation de la limite inférieure ELBO visant à « lisser » les résultats pour réduire l'erreur moyenne. L'obstacle à représenter la complexité des images naturelles en se basant uniquement sur un a priori gaussien a stimulé l'élaboration d'architectures plus sophistiquées mentionnées dans notre revue de littérature, comme les VQ-VAE qui exploitent des variables discrètes pour améliorer la netteté, ou les NVAE qui offrent une structure hiérarchique approfondie. De plus, la gestion délicate du compromis entre reconstruction et régularisation, parfois adressée par des méthodes comme le β-VAE pour favoriser le désintricage, reste un défi majeur qui a finalement pavé la voie aux modèles de diffusion actuels.

Reproductibilité des résultats : Concernant la reproductibilité des résultats, les auteurs n'ont pas rendu accessible leur code. Nous avons acccès au jeu de données MNIST et à la dimension de

l'espace latent, mais n'avons que peu de détail sur l'architecture entrainée (nombre et dimensions des couches cachées, nombre d'epochs, optimiseur, loss).

7 Conclusion

Ce projet nous a offert l'opportunité de nous familiariser plus en détail avec le fonctionnement et la mise en place de l'algorithme Auto-Encoding Variational Bayes (AEVB). Nous avons pu observer que l'intégration de l'estimateur SGVB et du reparametrisation trick représente une solution raffinée aux enjeux d'inférence dans les modèles profonds à variables latentes continues.

Cette méthode constitue une divergence notable par rapport aux techniques conventionnelles comme l'algorithme EM vu en cours ou les procédures MCMC. Alors que l'EM standard exige une détermination précise de la distribution a posteriori (souvent impossible pour les réseaux de neurones sophistiqués) et se révèle coûteux sur des ensembles de données volumineux, le VAE offre une estimation efficace grâce à l'inférence variationnelle. Le VAE, en substituant l'étape onéreuse d'espérance par une optimisation conjointe via la descente de gradient stochastique, offre une possibilité d'échelle inédite comparée aux méthodes précédentes.

Néanmoins, nos conclusions concernant CIFAR-10 ont souligné les contraintes de cette structure, notamment le flou des images produites attribuable à la nature de la fonction de coût et à la simplicité de l'a priori gaussien. Ces contraintes permettent d'explorer des architectures plus sophistiquées mises en évidence lors de notre revue de littérature. Pour des générations plus précises et réalistes, il est judicieux d'explorer les VAE quantifiés par vecteurs (VQ-VAE) qui se servent d'espaces latents discrets, les NVAE hiérarchisés, ou encore les modèles de diffusion. Ces derniers, considérés comme une extension des principes variationnels du VAE, sont actuellement à la pointe de la technologie en matière de génération d'images.

Références

[1] AARON VAN DEN OORD, ORIOL VINYALS, K. K. Neural discrete representation learning. *31st Conference on Neural Information Processing Systems* (2018).

[2] ARASH VAHDAT, J. K. Nvae : Adeephierarchical variational autoencoder. *34st Conference on Neural Information Processing Systems* (2021).

[3] IRINA HIGGINS, LOIC MATTHEY, A. P. C. B. X. G. M. B. S. M. A. L. beta-vae : Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations (ICLR)* (2017).

[4] JOHN PAISLEY, DAVID BLEI, M. J. Variational bayesian inference with stochastic search. *ICML 2012* (2012).

[5] KINGMA, D. P., AND WELLING, M. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)* (2014).

[6] METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., AND TELLER, E. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*, 6 (1953), 1087–1092.

[7] RAMESH, A., PAVLOV, M., GOH, G., GRAY, S., VOSS, C., RADFORD, A., CHEN, M., AND SUTSKEVER, I. Zero-shot text-to-image generation, 2021.

[8] S. KULLBACK, R. A. L. On information and sufficiency. *Annals of Mathematical Statistics 22*.

Annexe

A Calcul de la Borne Inférieure Variationnelle (ELBO)

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z})\, d\mathbf{z} \qquad \text{(Vraisemblance marginale)} \tag{2}$$

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z})\, d\mathbf{z} \qquad \text{(Prendre le logarithme)} \tag{3}$$

$$= \log \int q_\phi(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\, d\mathbf{z} \qquad \text{(Multiplication par } q_\phi(\mathbf{z}|\mathbf{x})\text{)} \tag{4}$$

$$\geq \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\, d\mathbf{z} \qquad \text{(Inégalité de Jensen)} \tag{5}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \qquad \text{(Notation avec l'espérance)} \tag{6}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \qquad \text{(Décomposition } p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})\text{)} \tag{7}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})) \qquad \text{(Définition de l'ELBO)} \tag{8}$$

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z})) \qquad \text{(Borne inférieure finale)} \tag{9}$$

B Algorithme AEVB

Initialisation : θ, ϕ

Répéter :

— $X^M \leftarrow$ Mini-lot aléatoire de M points de données (tiré du jeu de données complet)

— $\epsilon \leftarrow$ Échantillons aléatoires de la distribution de bruit $p(\epsilon)$

— $g \leftarrow \nabla_{\theta,\phi}\hat{\mathcal{L}}^M(\theta, \phi; X^M, \epsilon)$

— $\theta, \phi \leftarrow$ Mettre à jour les paramètres à l'aide des gradients g (par exemple, SGD ou Adagrad)

Jusqu'à : Convergence des paramètres (θ, ϕ)

Retourner : θ, ϕ

C Modèles génératifs

Nous avons utilisé Gemini 3 pour la recherche bibliographique concernant les avancées permises à la suite de la publication de l'article sur les VAE. Prompt : "Pour comprendre l'impact de cet article, donne moi les articles majeurs publiés suite à cet article et en lien avec celui-ci". Nous avons vérifié la véracité de la réponse en lisant les articles qu'il nous a proposé pour s'assurer que les articles étaient bien en lien avec les VAE et qu'il s'agissait bien d'une avancée majeure. Nous avons détecté quelques petites erreurs notamment sur les dates de publication des articles pour certaines desquelles il se trompait d'un an.

Nous avons utilisé Claude Sonnet 4.5 pour générer le `README` de notre code, ainsi que pour créer l'application Streamlit qui permet de visualiser proprement nos résultats. Nous avons vérifié la qualité de sa réponse en executant le code qu'il nous a proposé. Le code était de qualité et nous a permis d'avoir une bonne structure pour ensuite faire nos modifications pour l'adapter au mieux à ce que nous voulions faire.