Contrôle de la cardinalité par navigation dans l'espace latent des GANs

Perla Doubinsky, Nicolas Audebert, Michel Crucianu, Hervé Le Borgne

Contrôle de la cardinalité par navigation dans l'espace latent des GANs

Perla Doubinsky[1] Nicolas Audebert[1] Michel Crucianu[1] Hervé Le Borgne[2]

[1] CEDRIC (EA4329), Conservatoire national des arts et métiers, Paris 75003, France
[2] Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

perla.doubinsky@lecnam.net

1 Introduction

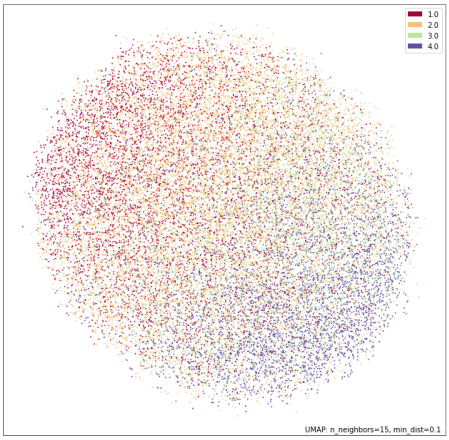
L'intégration de la notion de cardinalité se fait de manière naturelle chez les humains. Il est donc intéressant de se demander si les modèles génératifs comme les GANs intègrent également cette notion. Cela pourrait permettre de contrôler le nombre d'objets présents dans une image générée ou réelle grâce aux méthodes d'inversion. Contrôler le nombre de piétons ou de voitures sur des données représentant des scènes urbaines, par exemple, peut être particulièrement utile pour augmenter de manière ciblée des bases de données existantes. Saseendran et al. proposent une adaptation du GAN conditionnel pour générer des images conditionnées sur le nombre d'objets. L'inconvénient d'utiliser un GAN conditionnel est la nécessité de ré-entrainer le modèle. Nous souhaitons étudier si les modèles pré-entraînés encodent automatiquement la notion de cardinalité. Divers travaux [9, 12, 3, 4, 1, 8] ont étudié l'existence d'une structure reflétant la sémantique des données d'entraînement dans l'espace latent des GANs. Certains travaux ont montré que, pour des GANs entraînés sur des images représentant des visages, il est possible d'identifier des directions permettant de contrôler la présence ou l'intensité d'attributs binaires ou continus comme le genre ou l'âge. Parmi ces méthodes, certaines approches supervisées [12, 9, 3] s'appuient sur des classifieurs binaires entraînés dans l'espace latent pour guider l'apprentissage des directions. Dans ce travail, nous proposons une extension de ces méthodes à la cardinalité, dans un formalisme de régression ou de classification à plusieurs classes. Nous appliquons notre méthode dans l'espace latent $\mathcal{W}$ et la version étendue $\mathcal{W}+$ d'un StyleGAN entraîné sur le jeu de données MultiMNIST. Nos expérimentations sur ce jeu de données montrent que la formalisation multi-classe dans l'espace $\mathcal{W}$ permet de contrôler le nombre de chiffres présents dans une image.

2 Méthode

Étant donné un GAN pré-entraîné que l'on note $\mathbf{G}(\cdot)$ et l'espace latent noté $\mathcal{Z}$ tel que $\mathbf{I} = \mathbf{G}(\mathbf{z})$ avec $\mathbf{z} \in \mathcal{Z}$, nous cherchons à trouver une direction dans cet espace selon laquelle un déplacement fait varier le nombre d'objets présents dans l'image $\mathbf{I}$. Nous adoptons le formalisme de Yao et al. pour la recherche de directions locales : $\mathbf{z}' = \mathbf{z} + \alpha \cdot \mathbf{H}(\mathbf{z})$ où $\mathbf{H}$ est une transformation linéaire et $\alpha \in \mathbb{N}$. L'image modifiée est alors $\mathbf{I}' = \mathbf{G}(\mathbf{z}')$ et doit contenir α objets de plus que l'image $\mathbf{I}$. Pour contrôler la cardinalité, nous proposons d'utiliser un classifieur noté $\mathbf{C}(\cdot)$ entraîné à prédire le nombre d'objets étant donné un code latent $\mathbf{z}$. Ce classifieur permet de guider l'apprentissage des directions grâce à la fonction de coût suivante : $\mathcal{L}_{count} = \text{loss}(\mathbf{C}(\mathbf{z}'), \mathbf{y}+\alpha))$ où $\mathbf{y} = \mathbf{C}(\mathbf{z})$ est le nombre d'objets avant déplacement. Pour l'apprentissage du classifieur $\mathbf{C}(\cdot)$ puis de la matrice $\mathbf{H}$, nous pouvons modéliser le problème de deux façons : régression (MSE) ou classification multi-classe (entropie croisée) où chaque classe correspond à un nombre d'objets. La fonction de coût $\mathcal{L}_{\text{count}}$ permet de s'assurer que la direction nous emmène dans une région de l'espace latent où les codes latents correspondent au nombre d'objets souhaité du point de vue du classifieur mais il est possible que ces régions soient hors-distribution du point de vue du générateur. Afin de limiter cette éventualité, nous cherchons à contraindre le déplacement grâce à la fonction de coût suivante : $\mathcal{L}_{\text{rec}} = \|\mathbf{z}' - \mathbf{z}\|_2$. La fonction de coût finale est donc $\mathcal{L} = \mathcal{L}_{\text{count}} + \lambda \mathcal{L}_{rec}$. Enfin, afin de faciliter l'apprentissage, nous nous limitons à l'ajout ou au retrait d'un objet pendant l'entraînement, soit $\alpha = 1$ ou $\alpha = -1$ choisi de manière aléatoire.

3 Résultats préliminaires

Pour tester la faisabilité de notre approche, nous avons généré une version de MultiMNIST [10], constituée de $40K$ images de taille 128×128 comportant de 1 à 4 chiffres par image. Nous avons choisi d'utiliser StyleGAN2 [6] pour générer les données (FID : 7,14). La particularité de StyleGAN est, premièrement, de disposer d'un espace latent intermédiaire $\mathcal{W}$ avec des propriétés de démêlement, et deuxièmement d'injecter un code latent $\mathbf{w} \in \mathcal{W}$ au niveau des différentes couches de convolution du générateur. L'ensemble des codes latents des différentes couches constitue l'espace latent $\mathcal{W}$ étendu noté $\mathcal{W}+$. Pour les images générées, ces deux espaces sont équivalents car le code latent $\mathbf{w} \in \mathcal{W}$ est simplement répliqué. En revanche,



(a) Projection UMAP des codes latents colorés en fonction du nombre d'objets.

space	architecture	loss	train	test
$\mathcal{W}+$	[6144,2048,512]	MSE	100%	100%
$\mathcal{W}+$	[6144,2048,512]	CE	100%	100%
$\mathcal{W}$	[512,256]	MSE	89%	69%
$\mathcal{W}$	[512,256]	CE	91%	83%

(b) Performances des classifieurs (précision en %) entraînés dans l'espace latent.

FIGURE 1 – Projections des codes latents et performances des classifieurs dans $\mathcal{W}+$ et $\mathcal{W}$.

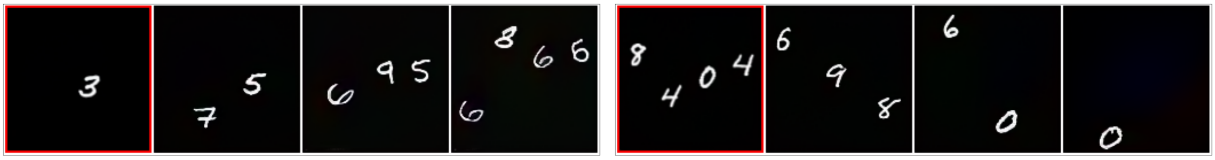


FIGURE 2 – Exemples de contrôle du nombre de chiffres avec notre approche appliqué dans $\mathcal{W}$ pour un classifieur entraîné avec une entropie croisée. L'image encadrée en rouge correspond à une image générée à partir d'un code latent échantillonné. Les images suivantes correspondent aux images obtenues par déplacements successifs d'amplitude $\alpha = \pm 1$.

il est également possible de faire du contrôle sur des images réelles. Cela nécessite d'utiliser une méthode d'inversion, autrement dit une méthode permettant de retrouver un code latent qui génère une image proche de l'image réelle. Différents travaux comme e4e [11] ont montré que les résultats d'inversion des images réelles sont meilleurs dans $\mathcal{W}+$ que dans $\mathcal{W}$. Nous avons testé notre méthode à la fois dans $\mathcal{W}$ de dimension 512 et dans $\mathcal{W}+$ de dimension $12 \times 512 = 6144$ en inversant les images avec e4e. Dans le premier cas, les exemples d'apprentissage sont les codes latents échantillonnés puis labellisés grâce à des classifieurs image pré-entraînés appliqués sur les images générés correspondantes. Dans le deuxième cas, les exemples d'apprentissage sont les images réelles inversées. La table 1b présente les performances des différents classifieurs entraînés dans l'espace latent, qui sont bonnes dans l'ensemble. On remarque cependant que les performances du classifieur entraîné dans $\mathcal{W}$ avec la MSE sont nettement en dessous de celles du classifieur entraîné avec l'entropie croisée. Ces bonnes performances semblent indiquer une structure reflétant la cardinalité. Ajouté à cela, la figure 1a montre une projection des codes latents où on observe un regroupement en fonction du nombre de chiffres en particulier dans $\mathcal{W}+$. Malgré les bonnes performances des classifieurs dans $\mathcal{W}+$, nous ne sommes pas parvenus à apprendre des directions contrôlant la cardinalité. Notre hypothèse est que les différents clusters vivent sur des variétés trop éloignées les unes des autres, les manipulations aboutissant donc à des codes « hors distribution ». Ceci peut être dû au fait que MultiMNIST est un jeu de données trop simple et que la dimension de $\mathcal{W}+$ est démesurée pour ce jeu de données. En revanche, avec le modèle de classification multi-classe (entropie croisée) appliqué dans $\mathcal{W}$, nous parvenons à obtenir des directions qui contrôlent le nombre de chiffres présents dans une image comme observé dans la figure 2.

4 Conclusion et perspectives

Nous proposons une méthode permettant de contrôler un nouveau type d'attribut, la cardinalité, dans des modèles GANs pré-entraînés. Nous adaptons les formalismes existants, se concentrant sur des attributs binaires ou continus, au cas d'un attribut discret multi-classe. Nos expérimentations sur MultiMNIST montrent qu'il y a une structure propice au contrôle de la cardinalité et qu'il est possible d'identifier des directions dans l'espace latent $\mathcal{W}$ de StyleGAN avec un classifieur entraîné avec l'entropie croisée. Cependant, le jeu de données MultiMNIST est un jeu de données relativement simple où il y a peu de sémantique pouvant interférer avec la cardinalité. Nous prévoyons de tester sur des jeux de données plus complexes comme CLEVR [5] ou Cityscapes [2] pour confirmer ces résultats. Nous souhaitons également étendre la méthode pour pouvoir contrôler l'ajout d'un objet d'une catégorie en particulier (par exemple, les différents chiffres pour MultiMNIST) et assurer

la préservation des objets initialement présents. Nous souhaitons également continuer nos expérimentations dans $\mathcal{W}+$ sur d'autres jeux de données et adapter les méthodes d'inversion existantes pour pouvoir mieux contrôler la cardinalité dans les images réelles.

Références

[1] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka. StyleFlow : Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3), May 2021. ISSN 0730-0301. doi : 10.1145/3447648. URL `https://doi.org/10.1145/3447648`.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] X. Hou, X. Zhang, H. Liang, L. Shen, Z. Lai, and J. Wan. Guidedstyle : Attribute knowledge guided style manipulation for semantic face editing. *Neural Networks*, 145 :209–220, 2022.

[4] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. GANSpace : Discovering interpretable GAN controls. In *Proc. NeurIPS*, 2020.

[5] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr : A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[6] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

[7] A. Saseendran, K. Skubch, and M. Keuper. Multi-class multi-instance count conditioned adversarial image generation. *arXiv preprint arXiv :2103.16795*, 2021.

[8] Y. Shen and B. Zhou. Closed-form factorization of latent semantics in GANs. In *CVPR*, 2021.

[9] Y. Shen, C. Yang, X. Tang, and B. Zhou. InterFaceGAN : Interpreting the disentangled face representation learned by GANs. *TPAMI*, 2020.

[10] S.-H. Sun. Multi-digit MNIST for few-shot learning, 2019. URL `https://github.com/shaohua0116/MultiDigitMNIST`.

[11] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or. Designing an encoder for StyleGAN image manipulation. *arXiv preprint arXiv :2102.02766*, 2021.

[12] X. Yao, A. Newson, Y. Gousseau, and P. Hellier. A latent transformer for disentangled face editing in images and videos. *2021 International Conference on Computer Vision*, 2021.