Attention stochastique basée patchs pour l'édition d'images

Nicolas CHEREL[1], Andrés ALMANSA[2], Yann GOUSSEAU[1], Alasdair NEWSON[1]

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris, France

[2]MAP5 & CNRS, Université de Paris, France

cherel@telecom-paris.fr, andres.almansa@parisdescartes.fr,
yann.gousseau@telecom-paris.fr, anewson@telecom-paris.fr

Résumé – Ces dernières années, les mécanismes d'attention ont pris une grande importance dans les méthodes d'apprentissage profond. Cependant, leur utilisation est souvent limitée par le fort coût algorithmique du calcul de la matrice d'attention. Nous proposons une couche d'attention efficace basée sur l'algorithme stochastique PatchMatch pour déterminer les plus proches voisins approchés entre patchs. Nous appelons cette couche "Patch-based Stochastic Attention Layer" (PSAL). Nous assurons la différentiabilité de PSAL, permettant ainsi un entraînement de bout en bout de tout réseau contenant cette couche d'attention. La couche PSAL a une faible empreinte mémoire et peut donc s'adapter à des images de haute résolution, ce que nous illustrons par une application à l'inpainting d'images.

Abstract – Attention mechanisms have become of crucial importance in deep learning in recent years. However, computing the attention matrix is an expensive step. We propose an efficient attention layer based on the stochastic algorithm PatchMatch for determining approximate nearest neighbors. We refer to our proposed layer as "Patch-based Stochastic Attention Layer" (PSAL). We ensure the differentiability of PSAL, thus allowing end-to-end training of any network containing our layer. PSAL has a small memory footprint and can therefore scale to high resolution images. We demonstrate the usefulness of PSAL on image inpainting.

1 Introduction

Les mécanismes d'attention [1] ont pris une grande importance dans de nombreuses architectures d'apprentissage profond. L'attention a permis de traiter des interactions à longue distance et ainsi de combler une limitation des convolutions qui sont des opérations *locales* uniquement. Malgré ce récent gain de popularité, la méthode standard de calcul de l'attention souffre d'une forte complexité algorithmique, quadratique selon le nombre d'éléments d'un tenseur.

Il s'avère que la couche d'attention est très étroitement liée au problème de la recherche du Plus Proche Voisin (PPV). En effet, l'opération de softmax biaise la distribution des poids d'attention vers un petit nombre de points similaires. Dans cet article, nous montrons que l'attention peut être calculée de manière efficace par une recherche de type Plus Proche Voisin Approché (PPVA). Pour cette recherche, nous nous tournons vers l'algorithme PatchMatch [2], un algorithme rapide pour la recherche PPVA qui est particulièrement efficace pour comparer des images similaires. Afin de surmonter les limites de calcul de la couche d'attention traditionnelle, nous proposons une couche d'attention qui utilise la méthode PatchMatch, spécifiquement conçue pour le cas des images, que nous nommons Patch-Based Stochastic Attention Layer (PSAL).

La couche PSAL a un faible impact sur la mémoire avec un coût linéairement proportionnel à la taille de l'image d'entrée. Par conséquent, elle peut être appliquée à des entrées bidimensionnelles de grande taille et nous permet en particulier d'appliquer le mécanisme d'attention à des images de haute résolution ou à des *feature maps* 2D à n'importe quelle profondeur d'un réseau.

Nous illustrons l'utilité de PSAL sur un problème d'inpainting et montrons que notre approche peut traiter des images de haute résolution sans dégrader la qualité des résultats.

L'article est organisé comme suit. Dans la section 2, nous détaillons les travaux précédents relatifs aux couches d'attention et à l'édition d'images basée sur les patchs. Dans la section 3, nous décrivons la couche d'attention classique, puis l'approche proposée dans cet article. Dans la section 4, nous montrons comment la couche PSAL peut être utilisée pour l'inpainting d'images, et permet notamment de traiter des images de haute résolution. Pour résumer, nous proposons une nouvelle couche d'attention pour les images qui a une complexité de mémoire considérablement réduite par rapport aux couches d'attention traditionnelles, tout en maintenant une fonctionnalité de base similaire. En particulier, cela signifie que les architectures basées sur l'attention peuvent être facilement modifiées pour traiter des images avec une résolution beaucoup plus élevée que ce qui est actuellement réalisable. À titre d'exemple concret, les besoins en mémoire de la méthode standard (*Full Attention*) augmentent de façon quadratique avec le nombre de pixels (16 Go de mémoire sont nécessaires pour une image de 256 et 256 Go pour une image de 512, ce qui est infaisable). PSAL, en revanche, passe à l'échelle linéairement et ne nécessite que 786 Ko et 3,15 Mo, respectivement, ce qui représente un gain mémoire assez considérable.

2 Contexte

Modèles d'attention Depuis leur introduction pour le traitement du langage naturel, les modèles d'attention sont devenus un élément crucial [1]. En vision par ordinateur, l'attention a été appliquée en synthèse [3], détection d'objets ou classification vidéo.

L'attention a une complexité quadratique $\mathcal{O}(n^2)$ en mémoire et en nombre d'opérations selon le nombre d'entrées, ce qui a motivé des travaux sur des alternatives plus efficaces. Certains travaux se sont concentrés sur la réduction du nombre de distances à calculer [4, 5]. ou sur des approximations linéaires [6, 7]. Des modèles d'attention efficaces sont nécessaires pour les images et peuvent résulter de restrictions locales par exemple [3, 8].

Édition d'image L'édition d'images utilise depuis longtemps des approches basées sur les patchs pour l'inpainting ou le transfert de style, faisant un usage intensif de notions de similarité entre patchs [9, 10, 2].

Dans le domaine de l'inpainting, les modèles d'attention ont également été utilisés. Partant d'architectures d'apprentissage profond de type encodeur-décodeur et de réseaux antagonistes pour l'inpainting [11], Yu *et al*. [12] ont observé que les textures manquent souvent de détails, et ont proposé d'utiliser une couche d'attention pour réutiliser les patchs existants, combinant ainsi l'apprentissage profond et les méthodes basées sur les patchs.

3 Couche d'attention stochastique par patchs

3.1 Approche classique de l'attention

Nous donnons d'abord la définition mathématique de *l'attention complète* (en anglais *Full Attention*) (FA), introduite dans [1]. Soit $Q \in \mathbb{R}^{n\times d}$ un ensemble de n *requêtes* organisées en matrice, chaque requête étant un vecteur de $\mathbb{R}^d$. Intuitivement, ces requêtes correspondent aux différents éléments pour lesquels nous voulons un vecteur d'attention. Dans le contexte des images, il peut s'agir d'un ensemble de patchs. Les requêtes sont comparées à un ensemble de *clés*, regroupées dans la matrice $K \in \mathbb{R}^{n\times d}$. Ces clés correspondent aux éléments que l'on souhaite utiliser comme référence pour donner plus ou moins d'importance aux requêtes. Dans le cas de l'image, les clés peuvent être un ensemble de patchs (pas nécessairement les mêmes que les requêtes). Étant donné un vecteur $V \in \mathbb{R}^{n\times d'}$, la sortie de l'attention est la somme pondérée suivante :

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T)V \tag{1}$$

où le softmax de la matrice $A = QK^T$, est une matrice stochastique par ligne et $\text{softmax}_t(A)_{ij} = e^{A_{ij}/t} / \sum_j e^{A_{ij}/t}$. L'équation (1) nécessite le calcul de la matrice complète QK^T avec n^2 entrées. Il en résulte une complexité de calcul de $\mathcal{O}(n^2 d)$, et une complexité de mémoire de $\mathcal{O}(n^2)$, pour n la taille de l'entrée (longueur de la séquence ou nombre de pixels). Ce fort besoin en mémoire est la principale limitation des couches d'attention.

En pratique, nous remarquons qu'après le softmax, *seuls quelques éléments ont une influence*. Afin de limiter la complexité algorithmique, nous proposons de rendre la matrice *parcimonieuse*, ne conservant qu'une seule valeur non nulle sur chaque ligne de QK^T, correspondant au PPV. Nous proposons en outre d'utiliser PatchMatch [2], un algorithme PPVA efficace, conçu spécifiquement pour les images.

3.2 Couche d'attention stochastique par patchs (PSAL)

Comme nous l'avons noté plus haut, les couches d'attention sont étroitement liées à la recherche de PPVs. Nous commençons par définir la fonction PPV ψ, entre les vecteurs de requête et les vecteurs clés :

$$\psi(i) = \underset{j\in\{1,\dots,n\}}{\arg\max} \langle Q_i, K_j \rangle , \tag{2}$$

où Q_i est la $i$ème ligne de la matrice Q, correspondant au vecteur i, et de même pour K. Le produit scalaire est couramment utilisé, mais pour plus de généralité, nous introduisons une fonction de similarité des patchs $s(Q_i, K_j)$.

Enfin, nous définissons la matrice parcimonieuse associée $A \in \mathbb{R}^{n\times n}$:

$$A_{i,j} = \begin{cases} 1 & \text{si} \quad \psi(i) = j \\ 0 & \text{sinon} \end{cases} . \tag{3}$$

Notre définition de l'attention est alors simplement :

$$\text{Attention}(Q, K, V) = AV. \tag{4}$$

L'étape suivante consiste à approximer ψ de façon efficace. Dans le cas général, cela peut être fait à l'aide d'algorithmes PPVA, par KD-tree ou par hachage (Locality Sensitive Hashing) comme dans Kitaev *et al*. [5]. Pour les images et les *feature maps*, où les vecteurs sont des patchs, PatchMatch [2] est une alternative efficace. PatchMatch est en effet un algorithme stochastique efficace pour rechercher des PPVAs de patchs dans des images et des vidéos, entre une image requête et une image clé. Un avantage significatif de PatchMatch est qu'il ne conserve que le PPV actuel, ce qui réduit considérablement les besoins en mémoire.

La Table 1 compare la complexité de PSAL à d'autres méthodes efficaces de calcul de l'attention. En particulier, la complexité de la mémoire est linéaire par rapport au nombre de pixels, alors que celle de *Full Attention* est quadratique. Ceci a des conséquences importantes, notamment sur la résolution maximale des images qui peuvent être traitées par les réseaux profonds qui utilisent des couches d'attention.

3.3 Différentiabilité

Une limitation de l'approche est que PatchMatch, en utilisant un unique voisin, n'est pas différentiable par rapport à Q et K, en raison de l'opérateur argmax dans l'équation (2).

Cependant, nous pouvons approcher la matrice A dans l'équation (4), par une relaxation continue $\tilde{A}_t(Q, K)$. Cette relaxation est liée à une version tronquée de la matrice d'attention complète $\mathrm{softmax}_t(QK^T)$ dans l'équation (1) et admet A comme cas limite (lorsque $t \to 0$). De plus, la relaxation $\tilde{A}$ est différentiable par rapport à Q et K tant que $k > 1$, et nous proposons deux façons de construire $\tilde{A}$ de manière efficace en termes de calcul et de mémoire : l'une basée sur l'utilisation de plusieurs voisins, et l'autre basée sur l'agrégation de patchs.

3.3.1 Différentiabilité avec k voisins

Nous proposons d'enrichir la matrice A en utilisant plusieurs voisins pour chaque patch, ce qui peut être fait avec une version modifiée de PatchMatch [2]. Chaque $\psi(i)$ est maintenant un *ensemble* de k correspondances. Nous redéfinissons la matrice A :

$$A_{i,j} = \begin{cases} s(Q_i, K_j) & \text{si } j \in \psi(i) \\ 0 & \text{sinon.} \end{cases} \tag{5}$$

Ensuite, nous appliquons une opération de softmax le long des lignes. On constate que l'implémentation Full Attention correspond au cas $k = n$. Nous désignons par PSAL k, la version avec k voisins.

3.3.2 Différentiabilité par agrégation

La deuxième approche que nous proposons consiste à effectuer une agrégation spatiale. Intuitivement, nous enrichissons la liste des PPVs pour un patch donné en utilisant les PPVs des voisins *spatiaux* de ce patch. Pour le dire plus familièrement, le voisin de mon voisin spatial est mon voisin. Dans ce cas, nous redéfinissons A en fonction d'un voisin spatial i' et de son voisin j' dans l'espace des patchs, comme suit :

$$A_{i,j} = \begin{cases} s(Q_{i'}, K_{j'}) & \text{si } \begin{cases} i' \in \mathcal{N}_i \text{ et } j' \in \psi(i') \\ \text{et } i' - i = j' - j \end{cases} \\ 0 & \text{sinon} \end{cases}, \tag{6}$$

où $\mathcal{N}_i$ est le voisinage spatial du patch i. La condition de l'équation (6) dit essentiellement que, pour un patch i, nous analysons son voisin *spatial* i' et le PPV de i', $\psi(i')$. Nous vérifions ensuite que le décalage spatial entre le patch i et i' est le même que celui des patchs PPV j et j'. La dernière condition est nécessaire pour relier j à j'. En pratique, le voisinage spatial coïncide avec la taille du patch. Cette agrégation peut être utile à d'autres couches d'attention parcimonieuses.

Ces deux solutions pour la différentiabilité peuvent être combinées. Sans ces approches, les réseaux ont de grandes difficultés à apprendre et produisent de mauvais résultats.

4 Résultats pour l'inpainting

Les modèles d'attention sont très utilisés pour l'inpainting d'images, processus consistant à remplir automatiquement les régions inconnues ou endommagées d'une image. Yu *et al.* [12] ont introduit avec succès une couche d'attention dans un réseau d'inpainting, appelée Contextual Attention (CA), pour une meilleure reconstruction des textures et des détails fins. Après un premier inpainting grossier, l'image est raffinée dans deux branches différentes : un réseau entièrement convolutif, et un réseau basé sur l'attention. Les résultats sont ensuite fusionnés.

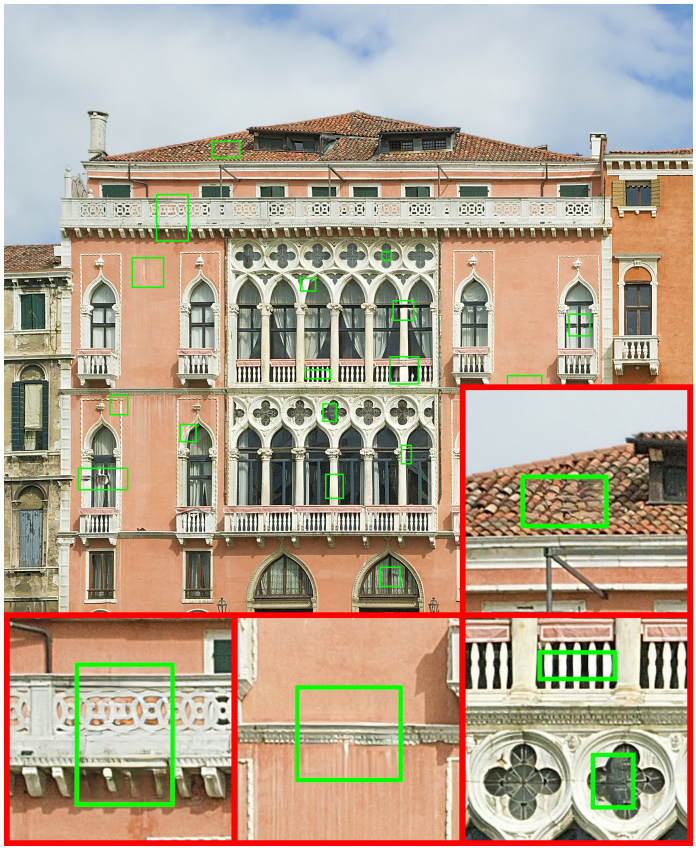

FIGURE 1 – Une image de taille 2700x3300 complétée avec PSAL. Les occultations sont indiquées en vert. Photo originale par Didier Descouens - Licence CC BY-SA 4.0

Malheureusement, la taille spatiale est très rapidement trop importante pour une occupation mémoire raisonnable, notamment pendant l'apprentissage. Yu *et al.* limitent le nombre de patchs à calculer grâce à un sous-échantillonnage. Une fois de plus, ceci illustre le besoin pratique d'une couche d'attention qui s'adapte aux grandes images.

Nous remplaçons directement la couche d'attention par PSAL 3. Nous utilisons une taille de patch de 7 pour PSAL, ce qui est équivalent à une taille de patch de 3 plus un sous-échantillonnage avec un facteur de 2 comme utilisé par CA. Les résultats quantitatifs ne montrent pas de différence significative entre PSAL et CA (Table 2). Cela confirme que PSAL peut effectivement remplacer la couche CA, sans perte de qualité, mais avec une très importante réduction des besoins en mémoire.

Enfin, dans la Figure 1, nous montrons qu'avec PSAL nous pouvons traiter des images de haute résolution (ici 3300×3300), ce qui est notre motivation initiale. En comparaison, le traitement d'une telle image en utilisant la couche d'origine néces-

TABLE 1 – Mémoire (mem.) (Go) et nombre d'opérations flottantes (GFLOPs) requis par la couche d'attention en fonction de la taille de l'entrée, le nombre de pixels n. La taille de patch est $p = 7$ et nous utilisons $d = 16$ canaux, ainsi un patch représente $D = p^2 d$ valeurs. Pour l'attention locale, la taille de fenêtre w est fixée à 50. Pour l'approche *Performer*, nous utilisons le paramètre recommandé $M = D \log D$. PSAL 3 et PSAL Agreg. ont une empreinte mémoire suffisemment faible pour autoriser des tailles de batches supérieures à 1. Le nombre d'opérations est aussi très inférieur.

Méthode d'attention	Complexité mémoire	Mem. pour 256x256	Complexité calcul	GFLOPs pour 256x256
Full Attention [1]	$\mathcal{O}(n^2)$	15.26	$\mathcal{O}(n^2 D)$	1113
Local Attention [3]	$\mathcal{O}(w^2 n)$	3.232	$\mathcal{O}(w^2 n D)$	385
Performer [7]	$\mathcal{O}(nD \log D)$	11.55	$\mathcal{O}(nD^2 \log^2(D))$	1789
PSAL 3	$\mathcal{O}(3n)$	0.075	$\mathcal{O}(10n \log(n) D)$	36
PSAL Agreg.	$\mathcal{O}(p^2 n)$	0.738	$\mathcal{O}(10n \log(n) D)$	38

TABLE 2 – Métriques d'inpainting sur l'ensemble de validation de la base Places2. PSAL permet d'obtenir des scores comparables en utilisant considérablement moins de mémoire. * réentrainé.

Méthode d'attention	Erreur $\ell_1 \downarrow$	Erreur $\ell_2 \downarrow$	PSNR $\uparrow$	Variation Totale $\downarrow$	SSIM $\uparrow$
ContextualAttention* [12]	11.8%	3.6%	16.4	**6.6%**	53.7
PSAL 3	**11.6%**	**3.6%**	**16.6**	6.9%	**54.1**

siterait plus de 1000 Go de mémoire.

5 Conclusion

Dans ce travail, nous avons présenté PSAL, une couche d'attention stochastique efficace basée sur les patchs qui n'est pas limitée par la mémoire du GPU. Cela rend possible le traitement d'images de haute résolution avec des réseaux profonds utilisant de l'attention, sans avoir recours aux approximations habituelles (sous-échantillonnage, etc.). De plus, ceci pourait permettre de nouvelles architectures de réseaux utilisant des mécanismes d'attention sur des *feature maps* de bas niveau.

Nous prévoyons de poursuivre ce travail en appliquant PSAL à d'autres tâches d'édition d'images, ainsi qu'à des traitements vidéos, ce qui n'est pas réalisable à l'heure actuelle avec les architectures d'attention classiques, en raison des contraintes de mémoire abordées dans le présent travail.

Une version étendue de ce travail est disponible en ligne : https://arxiv.org/abs/2202.03163, présentant davantage de résultats et d'autres applications.

Références

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and Dan B. Goldman, "PatchMatch : a randomized correspondence algorithm for structural image editing," in *SIGGRAPH 2009*, 2009.

[3] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam M. Shazeer, Alexander Ku, and Dustin Tran, "Image Transformer," *ICML*, 2018.

[4] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating Long Sequences with Sparse Transformers," *ArXiv*, 2019.

[5] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya, "Reformer : The Efficient Transformer," *ICLR*, 2020.

[6] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns : Fast autoregressive transformers with linear attention," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[7] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlós, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. J. Colwell, and A. Weller, "Rethinking Attention with Performers," *ArXiv*, 2020.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer : Hierarchical Vision Transformer using Shifted Windows," in *(ICCV) International Conference on Computer Vision*, mar 2021, p. 11.

[9] A. Criminisi, P. Perez, and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, Sept. 2004.

[10] Yonatan Wexler, Eli Shechtman, and Michal Irani, "Space-Time Completion of Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[11] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 107 :1–107 :14, July 2017.

[12] Jiahui Yu, Zhe L. Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang, "Generative Image Inpainting with Contextual Attention," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.