Élagage efficace des filtres basé sur les décompositions tensorielles

Van Tien Pham, Yassine Zniyed, Thanh Phuong Nguyen

Élagage efficace des filtres basé sur les décompositions tensorielles

Van Tien PHAM[1] Yassine ZNIYED[1] Thanh Phuong NGUYEN[1]

[1]Université de Toulon, Aix Marseille University, CNRS, LIS UMR 7020, Marseille, France

Résumé – Nous présentons une nouvelle méthode d'élagage des filtres pour les réseaux de neurones, appelée CORING (pour effiCient tensOr decomposition-based filteR prunING en anglais). L'approche proposée maintient l'aspect multidimensionnel des filtres grâce à l'utilisation de décompositions tensorielles. Notre approche permet de mesurer la similarité entre les filtres de manière plus efficace et plus précise que les méthodes traditionnelles qui utilisent des versions vectorisées ou matricisées des filtres. Avec cette approche, nous pouvons effectuer l'élagage des filtres plus efficacement en gardant l'essentiel de l'information en utilisant des décompositions de tenseurs sur les filtres. Les expériences menées sur différentes architectures prouvent l'efficacité de CORING.

Abstract – We present a novel filter pruning method for neural networks, named CORING, for effiCient tensOr decomposition-based filteR prunING. The proposed approach maintains the multidimensional aspect of filters through the use of tensor decompositions. Our approach leads to a more efficient and accurate way to measure the similarity, compared to traditional methods that use vectorized or matricized versions of filters. With this approach, we can perform filter pruning more efficiently by keeping most of the information using tensor decompositions on the filters. Experiments conducted on various architectures proved its effectiveness.

1 Introduction

L'élagage du réseau est une technique importante pour concevoir des modèles efficaces de réseaux de neurones convolutionnels (CNN en anglais), car elle réduit l'empreinte mémoire et les besoins de calcul tout en maintenant ou en améliorant les performances globales. Cet aspect est particulièrement important lorsque les CNNs sont déployés sur des appareils aux ressources limitées, tels que les téléphones mobiles ou les systèmes embarqués. L'hypothèse qui sous-tend l'élagage des réseaux est que de nombreux modèles sont surparamétrés, c'est-à-dire qu'ils contiennent un grand nombre de paramètres inutiles ou redondants [9]. L'élagage des paramètres redondants peut conduire à un modèle plus petit et plus efficace qui peut être déployé sur des appareils à ressources limitées, tout en améliorant la généralisation du modèle dans certains cas.

Parmi les techniques d'élagage existantes, l'élagage par filtre et l'élagage par poids sont deux approches populaires. L'élagage par le poids est une forme d'élagage non structuré [9], où les poids individuels jugés insignifiants sont élagués sans tenir compte d'une structure ou d'un modèle spécifique. D'autre part, l'élagage du filtre est un élagage structuré [17], où des filtres sont élagués sur la base de certains critères tout en conservant la structure globale du réseau. Par rapport à la première approche, l'élagage des filtres est plus facile à interpréter, moins sensible à l'initialisation, plus efficace en termes de calcul et déploiement [15].

Il ne fait aucun doute que le choix des filtres est le fondement de l'élagage des filtres. Les premiers travaux [12, 15] déterminent l'importance du filtre en mesurant uniquement l'information du filtre lui-même. Toutefois, ces approches négligent la corrélation entre les filtres, ce qui entraîne une forte redondance. Des travaux récents [17, 22] ont démontré les avantages potentiels de l'exploitation des corrélations ou des similitudes entre les filtres pour réduire la redondance. Cela repose sur l'hypothèse que des filtres similaires peuvent générer des caractéristiques en double et que l'élimination de cette redondance peut être compensée au cours du processus de réglage fin (fine-tuning en anglais). Les approches d'élagage basées sur la similarité évaluent l'importance des filtres en mesurant la distance par paire entre les filtres afin de construire la matrice de similarité de chaque couche du réseau. Sur la base de cette matrice de similarité, il est possible de déterminer l'importance ou la saillance des filtres et d'élaguer les filtres jugés moins importants.

Malgré leurs résultats prometteurs, de nombreuses méthodes pour l'élagage des filtres souffrent de certaines limitations qui n'ont pas encore été entièrement résolues. Les travaux existants [22] transforment souvent les filtres tensoriels d'ordre 3 en matrices 2-D ou en vecteurs 1-D, ce qui peut entraîner une perte d'informations spatiales ou temporelles. La structure multidimensionnelle des filtres est importante, et la négliger peut conduire à la perte d'informations cruciales [11]. En outre, les approches inter-filtres [22] peuvent nécessiter une analyse plus complexe et plus coûteuse en termes de calcul que les méthodes qui ne prennent en compte que les informations intra-filtres. L'élagage itératif peut être coûteux en raison de la nécessité de calculer la matrice de similarité à chaque itération. Il existe donc une demande de méthodes efficaces en termes de calcul pour résoudre ce problème.

Dans cet article, nous proposons une nouvelle approche, appelée CORING [1]. Il s'agit d'une technique d'élagage des filtres qui utilise des décompositions tensorielles [11]. Plus précisément, nous décomposons les filtres de chaque couche à l'aide de la décomposition en valeurs singulières d'ordre supérieur (HOSVD en anglais) [4] et utilisons cette représentation (l'approximation de rang un) pour mesurer la similarité entre les filtres, plutôt que de considérer l'ensemble du filtre sous sa forme tensorielle, matricielle ou vectorielle. Cette méthode permet de (i) préserver la structure multidimensionnelle des filtres et leurs informations essentielles tout en fournissant une approximation de faible rang, et (ii) réduire le temps de calcul comme nous le montrerons plus loin. Cette approche est générale et peut fonctionner avec n'importe quelle métrique de

[1]Code disponible à `https://github.com/pvtien96/CORING`

similarité. Dans nos expériences, nous évaluons notre approche en utilisant les distances euclidienne, cosinus, et VBD (pour variance-based distance), une dérivée de la distance presentée dans [21].

2 Notations et préliminaires

Les notations utilisées dans le reste de ce papier sont maintenant définies. Le produit extérieur est désigné par $\circ$. La variance est désignée par $\mathrm{Var}(.)$. La fonction $\lfloor x \rfloor$ désigne le plus grand entier inférieur ou égal à x. Les tenseurs sont représentés par des lettres majuscules calligraphiées en gras $\boldsymbol{\mathcal{X}}$. La norme d'un tenseur $\boldsymbol{\mathcal{X}}$ est la racine carrée de la somme des carrés de tous ses éléments, *i.e.,* $||\boldsymbol{\mathcal{X}}|| = \sqrt{\sum_{i=1}^{N1}\sum_{j=1}^{N2}\sum_{k=1}^{N3}\mathcal{X}_{i,j,k}^2}$. $\mathrm{unfold}_q\boldsymbol{\mathcal{X}}$ désigne le déploiement du tenseur $\boldsymbol{\mathcal{X}}$ sur son q-ième mode [11]. Nous introduisons maintenant quelques définitions qui seront utiles dans la suite. Le produit du mode q est l'une des opérations les plus importantes dans le traitement des tenseurs. Il est défini comme le produit d'un tenseur $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N_1 \times \ldots \times N_Q}$ et d'une matrice $\mathbf{U} \in \mathbb{R}^{J \times N_q}$ comme :

$$(\boldsymbol{\mathcal{X}} \times_n \mathbf{U})_{i_1 \ldots i_{q-1} j i_{q+1} \ldots i_D} \triangleq \sum_{i_q=1}^{N_q} X_{i_1 i_2 \ldots i_Q} u_{j i_q}. \quad (1)$$

Définition 1 *En se basant sur la définition du produit q-mode, nous pouvons rappeler la définition d'une décomposition de Tucker (TD) [18]. Considérons un tenseur central* $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ *et 3 matrices factorielles* $\mathbf{A} \in \mathbb{R}^{N_1 \times R_1}, \mathbf{B} \in \mathbb{R}^{N_2 \times R_2}, \mathbf{C} \in \mathbb{R}^{N_3 \times R_3}$. *La décomposition de Tucker de* $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ *est donnée par :*

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \triangleq [\![\boldsymbol{\mathcal{S}}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]. \quad (2)$$

Le triplet des valeurs minimales de $\{R_1, R_2, R_3\}$ *forme le* rang multilinéaire *de* $\boldsymbol{\mathcal{T}}$. *Une manière simple d'obtenir la TD d'un tenseur est d'utiliser l'algorithme HOSVD [4]. Dans le cas du HOSVD, les matrices factorielles de* (2) *sont contraintes d'être orthonormées, ce qui peut simplifier le calcul et l'interprétation des matrices factorielles.*

Considérons un modèle CNN pré-entraîné avec L couches, désigné par $\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_L$. Les paramètres de $\mathscr{C}_l$ peuvent être représentés comme un ensemble de filtres d'ordre 3 $\boldsymbol{\mathcal{F}}^{l_1}, \boldsymbol{\mathcal{F}}^{l_2}, \ldots, \boldsymbol{\mathcal{F}}^{l_{c_l}}$ contenant c_l filtres $\boldsymbol{\mathcal{F}}^{l_i} \in \mathbb{R}^{c_{l-1} \times h_l \times w_l}$ où c_l, c_{l-1}, h_l et w_l désignent respectivement le nombre de canaux de sortie, le nombre de canaux d'entrée, la hauteur du noyau et la largeur du noyau. En général, si nous définissons un tenseur d'ordre 4 t.q. $\boldsymbol{\mathcal{W}}^l_{:,:,:,i} = \boldsymbol{\mathcal{F}}^{l_i}$, alors l'objectif de l'élagage du filtre est d'optimiser la fonction de coût suivante :

$$\min_{\{\boldsymbol{\mathcal{W}}^l\}_{l=1}^L} \mathcal{L}(\boldsymbol{\mathcal{Y}}, f(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}}^l)), \quad \text{s.t.} \quad g(\boldsymbol{\mathcal{W}}^l) \leq \kappa_l, \quad (3)$$

où $\mathcal{L}(\cdot,\cdot)$ est la fonction de coût, $\boldsymbol{\mathcal{Y}}$ sont les étiquettes de vérité terrain, $\boldsymbol{\mathcal{X}}$ sont les données d'entrée, $f(\cdot,\cdot)$ est la sortie du CNN, κ_l est le nombre de filtres à conserver dans la l^{th} couche, et $g(.)$ est une fonction qui renvoit le nombre de filtres non nuls de son argument.

3 L'approche CORING

L'approche CORING, illustrée dans la figure 1, peut être divisée en trois étapes distinctes, chacune d'entre elles est abordée dans les sous-sections suivantes.

3.1 Décomposition de filtres

Dans cette section, par souci de clarté et pour simplifier la notation, nous considérerons un filtre sans son indice comme $\boldsymbol{\mathcal{F}}$ de taille $c_{l-1} \times h_l \times w_l$. Maintenant, si nous appliquons la TD dans (2) à $\boldsymbol{\mathcal{F}}$ en considérant que $R_1 = R_2 = R_3 = 1$, le modèle se réduit à

$$\boldsymbol{\mathcal{F}} \approx s \times_1 \mathbf{a} \times_2 \mathbf{b} \times_3 \mathbf{c} = [\![s; \mathbf{a}, \mathbf{b}, \mathbf{c}]\!], \quad (4)$$

où $\mathbf{a} \in \mathbb{R}^{c_{l-1}}$, $\mathbf{b} \in \mathbb{R}^{h_l}$, $\mathbf{c} \in \mathbb{R}^{w_l}$, et s est un scalaire qui peut aussi être vu comme un tenseur d'ordre 3 $s \in \mathbb{R}^{1\times1\times1}$. Sans perte de généralité, nous pouvons maintenant dénoter $\boldsymbol{\mathcal{F}}$ simplement comme

$$\boldsymbol{\mathcal{F}} \approx [\![\mathbf{a}, \mathbf{b}, \mathbf{c}]\!]. \quad (5)$$

Pour décomposer $\boldsymbol{\mathcal{F}}$ comme dans (5), il existe plusieurs méthodes de décomposition des tenseurs. Dans ce travail, nous avons choisi d'utiliser l'algorithme HOSVD [4]. Ce choix est justifié par plusieurs facteurs. Tout d'abord, le HOSVD est un algorithme non itératif, contrairement aux techniques basées sur l'ALS [10], qui peuvent être coûteuses en termes de calcul. Deuxièmement, il est basé sur la SVD, qui garantit que l'approximation par rapport aux matrices de déploiement décomposées est optimale [8]. Enfin, le HOSVD est relativement facile à mettre en œuvre, ce qui en fait un choix pratique pour de nombreuses applications.

En effet, la HOSVD de $\boldsymbol{\mathcal{F}}$ implique le calcul de la SVD des trois matrices de déploiement [11] $\mathrm{unfold}_1\boldsymbol{\mathcal{F}}$, $\mathrm{unfold}_2\boldsymbol{\mathcal{F}}$ and $\mathrm{unfold}_3\boldsymbol{\mathcal{F}}$, de taille, respectivement, $c_{l-1} \times (h_l \cdot w_l)$, $h_l \times (w_l \cdot c_{l-1})$ et $w_l \times (h_l \cdot c_{l-1})$. Les trois vecteurs $\mathbf{a}$, $\mathbf{b}$, et $\mathbf{c}$ dans (5) représentent, respectivement, les premiers vecteurs singuliers gauches dominants des matrices mentionnées précédemment. En appliquant la SVD rang-1 aux matrices $\mathrm{unfold}_q\boldsymbol{\mathcal{F}}$ (pour $1 \leq q \leq 3$), il est garanti, par le théorème d'Eckart-Young [6], que l'approximation obtenue est la meilleure approximation rang-1 dans la norme de Frobenius. Cependant, l'approximation globale du tenseur de faible rang n'est généralement pas optimale. Néanmoins, il a été montré que la décomposition obtenue est une bonne approximation de $\boldsymbol{\mathcal{F}}$, et qu'elle est bornée par une certaine limite [11]. Du point de vue de la complexité de calcul, le choix de $R_1 = R_2 = R_3 = 1$ fournit la méthode d'approximation la plus efficace en termes de complexité pour un tenseur donné. S'il est vrai que l'augmentation du rang multilinéaire peut potentiellement conduire à une meilleure approximation du tenseur original, nous avons décidé d'utiliser des rangs multilinéaires tous égaux à 1. Ce choix a été motivé par des considérations de complexité ainsi que par les résultats que nous avons obtenus au cours de nos expériences. Plus précisément, nous avons constaté que l'utilisation de cette approximation offrait un bon compromis entre la précision de l'approximation et l'efficacité du calcul, ce qui en fait un choix pratique pour notre méthode. De cette manière, non seulement l'information multidimensionnelle est préservée, mais l'efficacité du calcul est également atteinte. Il convient de noter que pour obtenir la matrice de similarité d'une couche avec N filtres, la complexité est de $\mathcal{O}(Nchw) + \mathcal{O}(N^2chw)$ alors que notre approche nécessite $\mathcal{O}(Nchw) + \mathcal{O}(N^2max(c,h,w))$.

3.2 Mesure de similarité

Considérons $d(.,.)$ comme une fonction de distance générale. Pour calculer la distance entre une paire de filtres $\boldsymbol{\mathcal{F}}^i$, $\boldsymbol{\mathcal{F}}^j$,

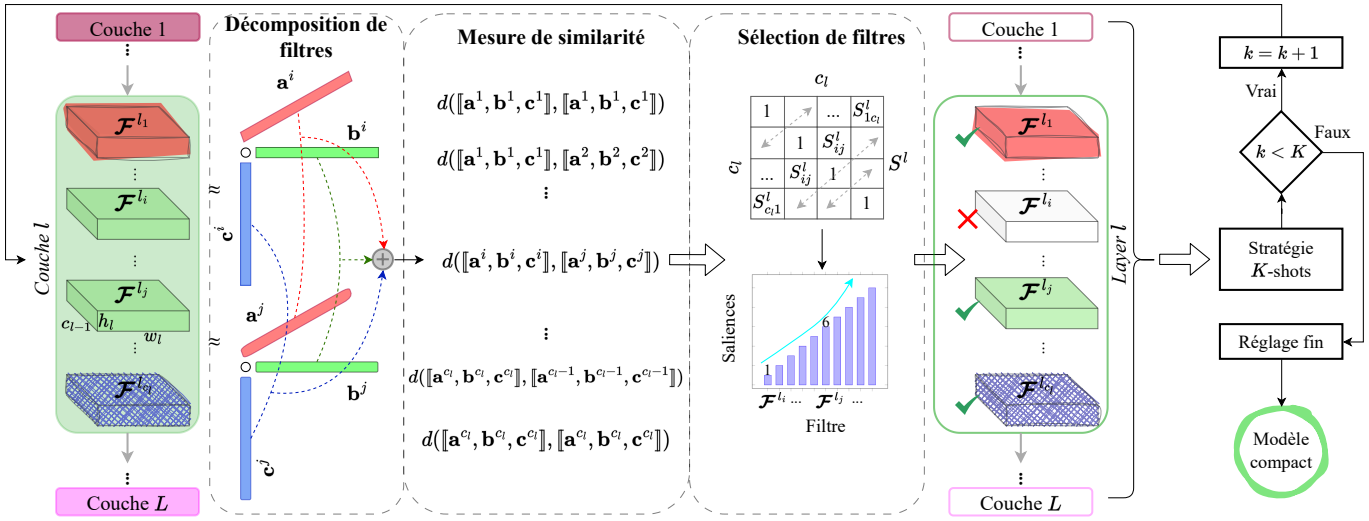

FIGURE 1 : L'approche CORING pour l'élagage des filtres pour une couche, résumée en trois étapes.

nous utiliserons leur HOSVD comme dans (5). Supposons que $\mathcal{F}^i = [\![\mathbf{a}^i, \mathbf{b}^i, \mathbf{c}^i]\!]$ et $\mathcal{F}^j = [\![\mathbf{a}^j, \mathbf{b}^j, \mathbf{c}^j]\!]$. Dans ce cas, nous pouvons calculer la distance entre $\mathcal{F}^i$ et $\mathcal{F}^j$ comme suit.

$$d(\mathcal{F}^i, \mathcal{F}^j) = d\big([\![\mathbf{a}^i, \mathbf{b}^i, \mathbf{c}^i]\!], [\![\mathbf{a}^j, \mathbf{b}^j, \mathbf{c}^j]\!]\big). \tag{6}$$

Une manière simple de calculer cette distance est de prendre la moyenne des distances entre les facteurs correspondants des deux filtres comme suit.

$$d(\mathcal{F}^i, \mathcal{F}^j) = \frac{d(\mathbf{a}^i, \mathbf{a}^j) + d(\mathbf{b}^i, \mathbf{b}^j) + d(\mathbf{c}^i, \mathbf{c}^j)}{3}. \tag{7}$$

En fonction de la métrique de distance choisie, une matrice de similarité $\mathbf{S}$ de taille $c\ foisc$ peut être construite de telle sorte que $S_{ij} = d(\mathcal{F}^i, \mathcal{F}^j)$. S_{ij} représente la similarité entre le i-ième et le j-ième filtre, et $d(\cdot, \cdot)$ est la fonction de distance.

Il convient de noter que les auteurs de [21] ont proposé une métrique basée sur le SNR pour mesurer la similarité des paires d'images pour l'apprentissage métrique profond. Bien que cette *quasi*-métrie se soit avérée efficace, elle présente une mise en garde importante : elle ne satisfait pas à la propriété de symétrie, qui est importante pour les fonctions de distance. Pour y remédier, nous définissons par la suite la VBD comme $d_{VBD}(\mathcal{F}^i, \mathcal{F}^j) = \frac{\mathrm{Var}(\mathcal{F}^i - \mathcal{F}^j)}{\mathrm{Var}(\mathcal{F}^i) + \mathrm{Var}(\mathcal{F}^j)}$.

3.3 Sélection de filtres

L'algorithme 1 présente la procédure de sélection des filtres utilisée dans CORING. L'algorithme prend en entrée une matrice de similarité entre toutes les paires de filtres, l'ensemble des filtres, et un ratio de parcimonie. La sortie de l'algorithme est un ensemble de κ filtres sélectionnés. La procédure fonctionne en supprimant itérativement les filtres qui sont les plus similaires aux autres filtres. L'algorithme commence par trouver la paire de filtres ayant la plus grande similarité et supprime l'un des filtres. Le choix du filtre à supprimer est basé sur la somme de ses similarités avec les autres filtres de la couche. L'algorithme met ensuite à jour la matrice de similarité en supprimant la ligne et la colonne du filtre supprimé, et continue à supprimer des filtres jusqu'à ce que l'objectif de parcimonie souhaité soit atteint. Par souci de simplicité, nous avons omis l'indice l dans le pseudo-code, mais l'élagage est effectué pour chaque couche convolutive en considérant les κ_l filtres à préserver à la couche l. Les κ_l filtres les moins similaires, appelés

Algorithme 1 : Sélection de filtres

Entrées : Matrice de similarité $\boldsymbol{S} \in \mathbb{R}^{c \times c}$, filtres $\mathcal{F}^1, \mathcal{F}^2, \ldots, \mathcal{F}^c$, ratio de parcimonie κ.
Sorties : Filtres sélectionnés $\mathcal{F}^{p_1}, \mathcal{F}^{p_2}, \ldots, \mathcal{F}^{p_\kappa}$.
pour $t = 1$ *à* $c - \kappa$ **faire**
  Trouver la plus grande similarité :

$$(i, j) = \underset{\substack{(x,y) \\ x \neq y}}{\operatorname{argmax}} \ S_{x,y}$$

  `Décider quel filtre supprimer entre` $\mathcal{F}^i$ `et` $\mathcal{F}^j$ `:`
  si $\sum_{k=1}^{c} S_{i,k} \geq \sum_{k=1}^{c} S_{j,k}$ **alors**
    Supprimer $\mathcal{F}^i$.
  sinon
    Supprimer $\mathcal{F}^j$.
  fin
fin

$\mathcal{F}^{p_1}, \mathcal{F}^{p_2}, \ldots, \mathcal{F}^{p_{\kappa_l}}$, sont conservés et le reste est élagué. Ce processus est exécuté en parallèle sur toutes les couches.

4 Simulations numériques

Bases de données et modèles de référence. CORING est évalué sur les bases de données CIFAR-10 et ImageNet, avec respectivement les modèles VGG-16 et ResNet-50.

Protocoles d'évaluation. La performance du modèle élagué est évaluée sur trois critères : précision, FLOPs et paramètres. CORING-X-K représente la combinaison de la distance X et de la stratégie K-shots [7], où X $\in$ {C, E, V} (correspondant à la distance cosinus, euclidienne et VBD) et $K \in \{5, 10, 15\}$. Sans K dans le suffixe, c'est un élagage 1-shot.

Résultats pour CIFAR-10. Le tableau 1 présente les résultats de l'élagage de VGG sur CIFAR-10. Dans les trois niveaux, en comparaison avec les autres méthodes, CORING obtient systématiquement la plus grande précision tout en maintenant le même niveau d'élagage. Les résultats des deux stratégies (1 et K- shots) d'élagage sont supérieurs à ceux de l'état de l'art.

TABLE 1 : Résultats de l'élagage de VGG-16 sur CIFAR-10

Model	Top-1 (%)	# Params. (↓%)	FLOPs (↓%)
VGG-16-BN	93.96	14.98M(00.0)	313.73M(00.0)
L1 [12]	93.40	5.40M(64.0)	206.00M(34.3)
CHIP [17]	93.86	2.76M(81.6)	131.17M(58.1)
EZCrop [16]	93.01	2.76M(81.6)	131.17M(58.1)
DECORE-500 [1]	94.02	5.54M(63.0)	203.08M(35.3)
FPAC [20]	94.03	2.76M(81.6)	131.17M(58.1)
CORING-C	**94.16**		
CORING-E	94.10		
CORING-V	94.11		
CORING-C-5	94.25	**2.76M(81.6)**	**131.17M(58.1)**
CORING-E-5	**94.42**		
CORING-V-10	94.36		
HRank-2 [15]	92.34	2.64M(82.1)	108.61M(65.3)
DECORE-200 [1]	93.56	1.66M(89.0)	110.51M(64.8)
EZCrop [16]	93.70	2.50M(83.3)	104.78M(66.6)
CHIP [17]	93.72	2.50M(83.3)	104.78M(66.6)
FSM [5]	93.73	N/A(86.3)	N/A(66.0)
FPAC [20]	93.86	2.50M(83.3)	104.78M(66.6)
AutoBot [2]	94.01	6.44M(57.0)	108.71M(65.3)
CORING-C	93.79		
CORING-E	**94.20**		
CORING-V	94.19		
CORING-C-15	**94.07**	2.50M(83.3)	**104.78M(66.6)**
CORING-E-15	94.03		
CORING-V-10	94.04		
HRank-3 [15]	91.23	1.78M(92.0)	73.70M(76.5)
DECORE-50 [1]	91.68	0.26M(98.3)	36.85M(88.3)
DECORE-100 [1]	92.44	0.51M(96.6)	51.20M(81.5)
FSM [5]	92.86	N/A(90.6)	N/A(81.0)
CHIP [17]	93.18	1.90M(87.3)	66.95M(78.6)
CORING-C	93.56		
CORING-E	93.54		
CORING-V	**93.63**		
CORING-C-10	93.68	1.90M(87.3)	66.95M(78.6)
CORING-E-15	**93.83**		
CORING-V-5	93.71		

Plus précisément, notre méthode améliore la généralisation du modèle en augmentant de 0.46 % le score de précision avec CORING-E-5 en diminuant plus de 81 % des paramètres.

Résultats pour ImageNet. Pour évaluer l'évolutivité de CORING, nous menons des expériences sur l'ensemble de données ImageNet avec ResNet-50, comme indiqué dans le tableau 2. Notre approche montre une plus grande précision dans toutes les régions, en particulier dans les zones à hauts ratios de compression, tout en bénéficiant de moins de FLOPs.

5 Conclusion

Nous avons proposé une nouvelle méthode d'élagage des filtres d'un réseau de neurones convolutionnel, appelée CORING, impliquant une approche de décomposition tensorielle pour extraire l'information du filtre qui peut être appliquée à n'importe quelle métrique ou stratégie d'élagage. Notre méthode a prouvé sa capacité à généraliser le modèle via l'élagage, sur différents jeux de données et diverses architectures. Des résultats de simulation ont démontré l'efficacité de cette approche.

TABLE 2 : Résultats de l'élagage de ResNet-50 sur ImageNet

Model	Top-1(%)	Top-5(%)	# Params (↓%)	FLOPs (↓%)
ResNet-50	76.15	92.87	25.50M(00.0)	4.09B(00.0)
CLR-RNF-0.2 [14]	74.85	92.31	16.92M(33.6)	2.45B(40.1)
LeGR [3]	76.20	93.00	N/A	N/A(27.0)
DECORE-8 [1]	76.31	93.02	22.69M(11.0)	3.54B(13.4)
CHIP [17]	76.30	93.02	15.10M(40.8)	2.26B(44.8)
TPP [19]	76.44	N/A	N/A	N/A(32.9)
CORING-V	**76.77**	**93.26**	**15.10M(40.8)**	**2.26B(44.8)**
HRank-1 [15]	74.98	92.33	16.15M(36.7)	2.30B(43.8)
DECORE-6 [1]	74.58	92.18	14.10M(44.7)	2.36B(42.3)
FPAC [20]	75.62	92.63	15.09M(40.9)	2.26B(45.0)
EZCrop [16]	75.68	92.70	15.09M(40.9)	2.26B(45.0)
LeGR [3]	75.70	92.70	N/A	N/A(42.0)
CHIP [17]	76.15	92.91	14.23M(44.2)	2.10B(48.7)
CORING-C	**76.34**	**93.06**	**14.23M(44.2)**	**2.10B(48.7)**
HRank-2 [15]	71.98	91.01	13.77M(46.0)	1.55B(62.1)
FPAC [20]	74.17	91.84	11.05M(56.7)	1.52B(62.8)
EZCrop [16]	74.33	92.00	11.05M(56.7)	1.52B(62.8)
CC-0.6 [13]	74.54	92.25	10.58M(58.5)	1.53B(62.6)
TPP [19]	75.12	N/A	N/A	N/A(60.9)
CHIP [17]	75.26	92.53	11.04M(56.7)	1.52B(62.8)
LeGR [3]	75.30	92.40	N/A	N/A(53.0)
CORING-V	**75.55**	**92.61**	11.04M(56.7)	**1.52B(62.8)**
DECORE-5 [1]	72.06	90.82	8.87M(65.2)	1.60B(60.9)
FPAC [20]	72.30	90.74	8.02M(68.6)	0.95B(76.7)
CHIP [17]	72.30	90.74	8.01M(68.6)	0.95B(76.7)
CLR-RNF-0.44 [14]	72.67	91.09	9.00M(64.7)	1.23B(69.9)
CORING-V	**73.98**	**91.77**	**8.01M(68.6)**	**0.95B(76.7)**

Références

[1] Manoj ALWANI, Vashisht MADHAVAN et Yang WANG : Decore : Deep compression with reinforcement learning. *CVPR*, 2022.

[2] Thibault CASTELLS et Seul-Ki YEOM : Automatic neural network pruning that efficiently preserves the model accuracy. *AAAI*, 2023.

[3] Ting CHIN, RUIZHOU, Cha ZHANG et Diana MARCULESCU : Towards efficient model compression via learned global ranking. *CVPR*, 2020.

[4] Lieven DE LATHAUWER, Bart DE MOOR et Joos VANDEWALLE : A multilinear singular value decomposition. *SIAM SIMAX*, 21(4), 2000.

[5] Yuanzhi DUAN, Yue ZHOU, Peng HE, Q LIU, Shukai DUAN et Xiaofang HU : Network pruning via feature shift minimization. *ACCV*, 2022.

[6] Carl ECKART et G. Marion YOUNG : The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

[7] Jonathan FRANKLE et Michael CARBIN : The lottery ticket hypothesis : Training pruned neural networks. *ICLR*, 2019.

[8] Gene H. GOLUB et Charles F. VAN LOAN : *Matrix Computations*. The Johns Hopkins University Press, third édition, 1996.

[9] Song HAN, Jeff POOL, John TRAN et William J. DALLY : Learning both weights and connections for efficient neural networks. *NIPS*, 2015.

[10] Richard A. HARSHMAN : Foundations of the parafac procedure : Models and conditions for an "explanatory" multi-model factor analysis. 1970.

[11] Tamara G. KOLDA et Brett W. BADER : Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[12] Hao LI, Asim KADAV, Igor DURDANOVIC, Hanan SAMET et Hans Peter GRAF : Pruning filters for efficient convnets. *ICLR*, 2017.

[13] Yuchao LI, Shaohui LIN, Fan YANG, JINCHENG, Qi TIAN et Rongrong JI : Towards compact cnns via collaborative compression. *CVPR*, 2021.

[14] Mingbao LIN, L CAO, Y ZHANG, L SHAO et R JI : Pruning networks with cross-layer ranking & k-reciprocal nearest filters. *TNNLS*, 2022.

[15] Mingbao LIN, R JI, Yan W, Y ZHANG, B ZHANG, Y TIAN et L SHAO : Hrank : Filter pruning using high-rank feature map. *CVPR*, 2020.

[16] R. LIN, J. RAN, D. WANG, K. CHIU et N. WONG : Ezcrop : Energy-zoned channels for robust output pruning. *WACV*, 2022.

[17] Yang SUI, M. YIN, Yi XIE, H. PHAN et Bo YUAN : Chip : Channel independence-based pruning for compact neural networks. *NIPS*, 2021.

[18] L. R. TUCKER : Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[19] H WANG et Y FU : Trainability preserving neural pruning. *ICLR*, 2023.

[20] Huoxiang YANG, Y LIANG, Wei LIU et Fanyang MENG : Filter pruning via attention consistency on feature maps. *Applied Sciences*, 2023.

[21] Tong YUAN, W DENG, J TANG, Y TANG et B CHEN : Signal-to-noise ratio : A robust distance metric for deep metric learning. *CVPR*, 2019.

[22] Wei ZHANG et Zhiming WANG : Fpfs : Filter-level pruning via distance weight measuring filter similarity. *Neurocomputing*, 512:40–51, 2022.