Compression optimisée des CNNs par élagage sous contrainte spatiale

Romane Scherrer*, Thomas Quiniou *, Nazha Selmaoui-Folcher*

*Institut de Sciences Exactes et Appliquées, Université de la Nouvelle-Calédonie
romane.scherrer@hotmail.fr; nazha.selmaoui@unc.nc

Résumé. L'article présente un nouveau critère d'élagage des filtres adapté aux modèles profonds générant des images. Contrairement aux approches traditionnelles qui se basent sur l'intensité des pixels dans les activations intermédiaires, la méthode proposée prend en compte la position des pixels dans l'image et utilise des masques binaires pour distinguer les zones essentielles (objets) et l'arrière-plan. Cette technique permet d'évaluer l'importance des filtres en tenant compte des zones significatives dans les "feature maps". L'article compare plusieurs critères d'élagage, démontrant que notre approche permet d'atteindre des taux de compression élevés tout en maintenant une faible erreur quadratique moyenne sur les images reconstruites.

1 Introduction

Les systèmes embarqués utilisés dans la vision par ordinateur sont souvent soumis à des contraintes strictes de mémoire et de calcul, ce qui rend l'utilisation des réseaux de neurones convolutionnels (CNN) difficile en raison de leurs besoins en ressources. Cette problématique est particulièrement critique dans des applications telles que la reconstruction d'images et l'analyse en temps réel, où les modèles doivent à la fois être performants et adaptés à des environnements aux capacités limitées. Réduire la taille et le temps d'inférence des réseaux de neurones est donc une étape cruciale pour déployer des solutions de vision par ordinateur dans des contextes embarqués, sans compromettre leur efficacité.

Plusieurs stratégies ont été proposées pour réduire la taille des modèles tout en maintenant des performances acceptables. L'une des approches courantes consiste à concevoir des architectures compactes, comme SqueezeNet (Iandola et al., 2016), MobileNet (Howard et al., 2017) ou ShuffleNet (Zhang et al., 2017), qui utilisent des convolutions optimisées pour minimiser les paramètres et les opérations, notamment par le biais de convolutions séparables en profondeur et parallèles. En plus des architectures compactes, la quantification (*quantization*) des paramètres est une méthode efficace pour compresser les modèles existants (Gong et al., 2014; Han et al., 2016; Hubara et al., 2016). En réduisant la précision des poids (par exemple, de 32 bits à 8 bits), la quantification diminue l'espace mémoire nécessaire, bien qu'elle puisse parfois affecter la précision du modèle. L'élagage des paramètres est une autre approche populaire qui consiste à supprimer certains paramètres d'un modèle entraîné pour réduire sa taille tout en minimisant l'impact sur les performances. Il existe deux principales approches : l'élagage non structuré, qui met à zéro certains poids sans les supprimer, et l'élagage structuré, qui

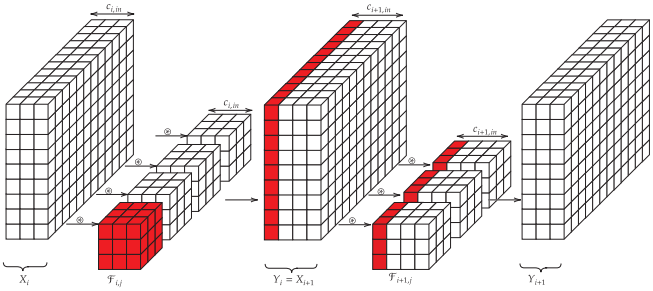

FIG. 1 – *En supprimant le filtre $\mathcal{F}i, j = 1$ en rouge, le feature map associé disparaît, entraînant également la suppression des poids de la couche suivante.*

élimine des blocs de neurones ou de filtres, réduisant ainsi directement les opérations d'inférence. Les méthodes d'élagage peuvent être globales, supprimant une fraction des paramètres du modèle (Jonathan Frankle, 2019), ou locales, agissant sur chaque couche (Han et al., 2015), et s'appuient sur divers critères comme la magnitude des poids (Li et al., 2017; Lee et al., 2018), leur influence sur la fonction de perte (LeCun et al., 1990; Yang et al., 2023) ou l'activation des neurones (Hu et al., 2016; Luo et al., 2019). Ces techniques sont souvent efficaces pour la classification, mais peuvent montrer des limites dans les tâches de régression d'images où des activations faibles peuvent être cruciales.

Dans cet article, nous proposons une nouveau critère d'élagage adapté aux modèles générant des images. Notre méthode ajoute une contrainte spatiale dans les activations des "feature maps", permettant de conserver les filtres importants pour la tâche de reconstruction. En utilisant des masques binaires pour distinguer l'arrière-plan des objets, nous rééquilibrons la contribution des pixels dans les zones pertinentes, permettant un élagage plus efficace des filtres.

2 Formalisme de l'élagage de filtres

L'élagage d'un modèle consiste à identifier et supprimer les paramètres inutiles, en minimisant leur impact sur les performances. Dans cette section, nous formalisons l'élagage des filtres dans une couche de convolution.

Une couche de convolution (i) transforme un tenseur d'entrée $X_i \in \mathbb{R}^{H \times W \times C_{in}}$ en un tenseur de sortie $Y_i \in \mathbb{R}^{H \times W \times C_{out}}$, où (H, W) sont les dimensions spatiales et (C_{in}, C_{out}) représentent le nombre de canaux d'entrée et de sortie. Cette opération est réalisée à l'aide de C_{out} filtres 3D notés $\mathcal{F}_{i,j} \in \mathbb{R}^{k \times k \times C_{in}}$ où k est la taille des noyaux. Chaque filtre génère un seul feature map. Pour élaguer une couche (i), un score d'importance est attribué à chaque filtre $\mathcal{F}_{i,j}$, basé sur les paramètres du filtre ou les activations des feature maps générés. Les filtres ayant les scores les plus faibles sont supprimés, comme illustré par la suppression d'un filtre $\mathcal{F}_{i,j}$ dans la figure 1, ce qui entraîne également la suppression des paramètres correspondants dans la couche suivante.

Deux stratégies d'élagage sont couramment utilisées : la fixation à zéro, qui met à zéro certains poids tout en conservant la structure du modèle, et la suppression complète, qui élimine les filtres inutiles et réduit directement la mémoire et le nombre d'opérations pour faire une inférence (FLOPS). Cette dernière modifie la structure du modèle et nécessite également de supprimer les poids correspondants dans les couches suivantes (Blalock et al., 2020).

3 Description de la contribution

Les approches classiques d'élagage considèrent qu'un feature map contenant de nombreux zéros ou des pixels de faible intensité est peu significatif et que le filtre le générant peut être supprimé (Hu et al., 2016). Cette méthode fonctionne bien pour les tâches de classification, mais pour les modèles générant des images, nous postulons que la position des pixels faibles est également cruciale. Par exemple, dans une image à reconstruire contenant de petits objets, la majorité des pixels appartiennent à l'arrière-plan. Les pixels correspondant à des valeurs faibles dans les feature maps mais situés dans des zones clés, comme celles des objets, peuvent être significatifs mais ne sont pas prises en compte dans les critères existants qui calculent un score d'importance global sur le feature map. Nous proposons donc une méthode qui prend en compte à la fois l'intensité et la position des pixels via des masques binaires, permettant d'évaluer séparément l'importance des zones recouvertes par l'objet de l'arrière-plan. Pour générer ces masques, les images cibles sont binarisées par seuillage, puis sous échantillonnées par des opérations d'Average Pooling pour correspondre aux dimensions des feature maps intermédiaires des CNNs. L'algorithme 1 décrit ce processus : pour chaque feature map, la moyenne des pixels dans les zones de l'objet et de l'arrière-plan est calculée. Deux seuils sont ensuite utilisés pour conserver une fraction $fract$ des feature maps ayant les scores d'importance les plus élevés dans les deux zones.

Algorithme 1 Algorithme d'élagage

Entrée: Feature Maps (FM) de dimensions $H \times W \times nbfilters$ en sortie de la couche de convolution, les images cibles (GT) et $fract$ la fraction des filtres à conserver.
Sortie: Les indices des filtres à conserver ($FiltersToKeep$).

$M \leftarrow BinaryMask(GT, W, H)$ ▷ *Sous-échantillonage du masque GT*
imp_{back}; $imp_{obj} \leftarrow [\ ]; [\ ]$
$indices \leftarrow [\ ]$
for $i \in [1, ..., nbfilters]$ **do**
  $I \leftarrow FM[:, :, i]$
  $I_{obj} \leftarrow I[M == 0]$ ▷ *valeurs des pixels dans la surface couverte par l'objet*
  $I_{back} \leftarrow I[M == 1]$ ▷ *valeurs des pixels dans l'arrière-plan*
  $imp_{back}[i] \leftarrow mean(I_{back})$
  $imp_{obj}[i] \leftarrow mean(I_{obj})$
  $indices[i] \leftarrow i$
end for
$thr_{back} \leftarrow quantile(imp_{back}, 1 - fract)$
$thr_{obj} \leftarrow quantile(imp_{obj}, 1 - fract)$
$FiltersToKeep_{obj} \leftarrow indices[imp_{obj} \geq thr_{obj}]$
$FiltersToKeep_{back} \leftarrow indices[imp_{back} \geq thr_{back}]$
$FiltersToKeep \leftarrow FiltersToKeep_{back} \cup FiltersToKeep_{obj}$

4 Expérience et évaluation

4.1 Jeu de données

Pour évaluer notre méthode, nous avons simulé des hologrammes. Un hologramme est une image résultant de l'interférence entre une onde de référence et une onde diffractée par un objet. Contrairement à une image classique, l'objet n'est pas directement visible sur l'hologramme et une étape de reconstruction, similaire à une re-focalisation, est nécessaire pour obtenir une représentation visuelle exploitable. La théorie complète de la formation et du traitement des hologrammes numériques est couverte dans des travaux précédents (Latychevskaia et Fink, 2015; Picart et Montresor, 2019). Ici, nous nous concentrons sur les éléments essentiels à la simulation des données. Nous avons utilisé le jeu de données MNIST, composé de 80 000 images en nuances de gris de 28x28 pixels, représentant des chiffres manuscrits de 0 à 9. Les images du jeu MNIST ont d'abord été sur-échantillonnées par un facteur de 3 puis un zéro-padding a été appliqué pour obtenir des images de 512x512 pixels. Ensuite, nous avons simulé les hologrammes en ligne en suivant le protocole décrit par Scherrer et al. (2022). Du bruit gaussien est ensuite ajouté à l'hologramme et ce dernier est recadré dans une image de 256x256 pixels. Finalement, l'hologramme est normalisé entre 0 et 1 en fonction des valeurs minimale et maximale de ses pixels.

4.2 Modèle de reconstruction

Pour évaluer notre méthode d'élagage, nous avons entraîné un modèle de reconstruction holographique, puis l'avons compressé à différents taux. Le modèle est entraîné à reconstruire l'image MNIST originale à partir de l'hologramme d'entrée. L'architecture est inspirée des travaux de Vijayanagaram (2020); Shao et al. (2020), utilisant des CNNs, en particulier des U-Nets modifiés (Ronneberger et al., 2015). Notre modèle comporte 20 couches de convolution dans deux parties : l'encodeur et le décodeur. L'encodeur contient des blocs de convolution 2D avec des noyaux 3×3, suivis de couches de MaxPooling qui réduisent la taille des feature maps. Le décodeur, symétrique, utilise des blocs similaires mais avec sur-échantillonnage pour agrandir les feature maps. Trois blocs résiduels connectent les parties encodeur et décodeur en transférant des informations entre elles. La dernière couche applique une fonction d'activation Sigmoïde, tandis que les autres utilisent ReLu. Le U-Net est entraîné pendant 100 epochs avec un batch size de 32 sur un jeu de données comprenant 5000 hologrammes d'apprentissage. L'optimiseur Adam est utilisé avec un learning rate de 10^{-4}. Pendant l'apprentissage, le modèle minimise l'entropie binaire croisée (BCE). La figure 2 présente des exemples de reconstructions holographiques réalisées par le modèle sur des données de test. L'erreur quadratique moyenne sur le jeu de test comprenant 1000 images est de $0,22 \times 10^{-3}$.

4.3 Élagage du modèle et évaluation des modèles compressés

Pour évaluer et comparer notre méthode d'élagage, nous avons implémenté plusieurs critères couramment utilisés. Pour calculer le score d'importance d'un filtre $\mathcal{F}_{i,j}$ dans une couche i (notés $s_{i,j}$), certaines de ces méthodes nécessitent un sous-ensemble du jeu de données, tandis que d'autres calculent directement des statistiques sur les paramètres des filtres. Nous avons

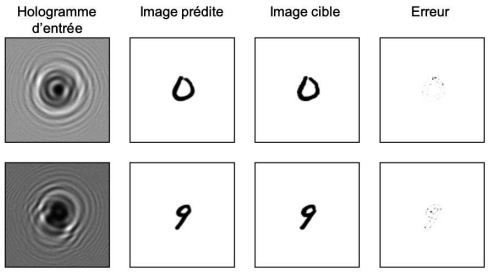

FIG. 2 – *Exemples de reconstruction d'hologrammes de test avec le modèle initial.*

utilisé $N = 200$ hologrammes de validation pour calculer les scores d'importance. Ces critères sont donnés sur le tableau 1 :

TAB. 1 – *Résumé des méthodes d'élagage des filtres*

Méthode	**Description**
Random	Les filtres sont supprimés de manière aléatoire dans chaque couche.
Weight Sum (Li et al., 2017)	Score basé sur la somme des valeurs absolues des noyaux du filtre : $s_{i,j} = \sum_{k=1}^{C_{in}} \lvert\mathcal{F}_{i,j}[:,:,k]\rvert$, où C_{in} est le nombre de canaux d'entrée.
APoZ (Hu et al., 2016)	Score basé sur la proportion de pixels nuls dans le feature map de sortie : $s_{i,j} = \frac{1}{N}\sum_{k=1}^{N} f(Y_i[:,:,j])$, où N est le nombre d'images de validation, $f(\cdot)$ mesure la proportion de pixels nuls, et Y_i est le tenseur en sortie.
Mean-mean	Score calculé comme la moyenne des valeurs des pixels dans le feature map de sortie : $s_{i,j} = \frac{1}{N}\sum_{k=1}^{N} \text{mean}(Y_i[:,:,j])$, où k est l'indice des images.
ThiNet (Luo et al., 2019)	Sélection des filtres en minimisant l'erreur de reconstruction dans la couche suivante, évaluant leur importance selon leur impact sur la qualité des feature maps produites.

Tous les critères sont appliqués dans une configuration *layerwise*, c'est-à-dire qu'une fraction fixe des filtres est conservée dans chaque couche du modèle initial. Le modèle initial est élagué avec des fractions allant de 0,1 à 0,9 avec un pas de 0,1. Notre stratégie d'élagage, implémentée sur Tensorflow, peut être qualifiée de "Prune Once and retrain" : le modèle est élagué une fois puis réentrainé sur quelques epochs pour compenser la diminution de performance induite par la suppression des paramètres. Une fois que les indices des filtres à conserver dans chaque couche sont connus, nous créons un nouveau modèle de petite taille et nous initialisons les valeurs des paramètres avec celles des filtres survivants du modèle initial. Pendant cette étape, nous modifions également les couches Add des blocs résiduels par l'implémentation de la Connectivité Mixte de Lemaire et al. (2019). Les modèles élagués sont ensuite réentraînés pendant 10 epochs avec les mêmes hyperparamètres que le modèle initial mais avec un learning rate de 10^{-5}.

Pour évaluer la qualité des images reconstruites par les modèles élagués, nous utilisons l'erreur quadratique moyenne (MSE) sur un ensemble de 1000 images de test, après avoir identifié l'epoch où la fonction de coût atteint son minimum sur le jeu de validation.

Nous calculons la MSE pour comparer les images cibles et reconstruites, en précisant que cette métrique ne prend en compte que les valeurs des pixels, et non leur agencement. Pour

évaluer la similarité structurelle des images reconstruites par rapport aux images cibles, un CNN est utilisé pour classifier les images MNIST en 10 catégories. Si les modèles élagués sont performants après l'étape de fine-tuning, le CNN devrait être capable de classer correctement les images qu'ils génèrent. Ce CNN, composé de deux couches de convolution (32 et 64 filtres), suivies de MaxPooling 2x2 et de deux couches fully connected (100 et 10 neurones), a été entraîné pendant 100 epochs en minimisant l'entropie croisée catégorielle sur un jeu de 10 000 images. Nous mesurons et reportons l'écart d'accuracy ($\Delta(acc)$) entre les images générées par les modèles élagués et les images cibles du jeu de test. Pour quantifier le taux d'élagage, nous indiquons les scores de MSE et $\Delta(acc)$ en fonction de deux métriques : (1) **Accélération théorique (*speedup*)**, définie comme le rapport des FLOPS du modèle initial à ceux du modèle élagué, et (2) **Compression**, définie comme le rapport du nombre de paramètres du modèle initial à celui du modèle élagué.

5 Résultats

La figure 3 présente les performances des modèles élagués après la phase d'apprentissage supervisé. Pour des taux de compression allant jusqu'à 4, tous les critères, à l'exception de *Random*, permettent d'obtenir des modèles élagués avec des scores MSE plus faibles que ceux du modèle initial. Cela indique que la phase d'apprentissage a réussi à compenser la perte de performance due à la suppression des paramètres, ce qui conduit à une amélioration globale des performances des modèles.

Pour des taux de compression proches de 10, les modèles élagués avec notre critère, *APOZ* et *weight sum* présentent des performances très similaires à celles du modèle initial. Cependant, à des taux de compression plus élevés, les scores MSE augmentent rapidement et la précision du CNN diminue. Cela suggère que certaines caractéristiques essentielles des objets reconstruits, nécessaires à leur classification, disparaissent des images. En d'autres termes, la phase d'apprentissage devient de moins en moins efficace pour compenser la perte d'information liée à l'élagage, ce qui affecte la capacité du modèle élagué à reconstruire correctement l'image.

Les modèles élagués selon notre critère affichent les scores MSE les plus bas sur l'ensemble de la plage de compression. Par exemple, pour un taux de compression de 20, la MSE est plus de deux fois inférieure à celle obtenue avec les autres critères. De plus, avec notre critère, la perte de performance en classification du CNN est seulement de 0.4% pour une compression de 20, ce qui démontre l'efficacité de notre approche pour préserver la qualité de reconstruction des objets tout en minimisant la perte de performance du modèle.

6 Conclusion

Dans cet article, nous avons introduit un nouveau critère d'élagage de filtres, intégrant l'information spatiale des feature maps pour mieux préserver les caractéristiques des objets dans les tâches de reconstruction. Contrairement aux approches traditionnelles, qui ont tendance à minimiser l'erreur de reconstruction de l'arrière-plan, notre méthode privilégie les filtres essentiels à l'extraction des caractéristiques des objets, ce qui se traduit par une amélioration notable des performances.

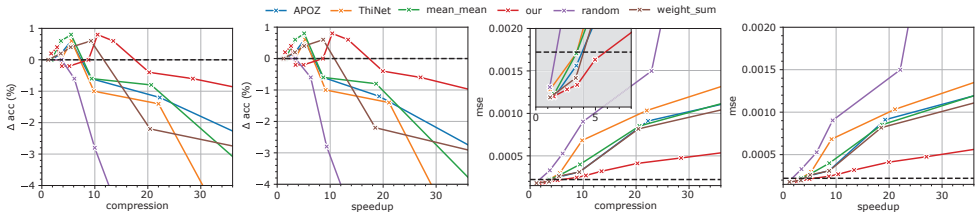

FIG. 3 – *Performances des modèles élagués après l'apprentissage. Les lignes pointillées correspondent aux scores du modèle initial. Les graphiques dans les zones grisées présentent les performances pour des taux de compression inférieurs à 8.*

Cependant, notre méthode repose sur l'utilisation d'un masque binaire de segmentation pour isoler les objets de l'arrière-plan. Si cette segmentation est relativement simple dans des cas où les objets sont bien définis, elle peut s'avérer plus complexe dans des contextes où la distinction entre l'objet et l'arrière-plan est moins évidente. Une solution pourrait être d'utiliser un modèle externe de segmentation, introduisant un surcoût limité, car la segmentation ne serait effectuée qu'une seule fois sur un sous-ensemble du jeu de validation. De plus, notre critère peut également être utilisé dans une configuration *globalwise*, permettant une suppression proportionnelle des paramètres dans l'ensemble du modèle, plutôt que couche par couche. Dans le futur, nous prévoyons d'élargir l'évaluation de notre méthode à d'autres jeux de données et types d'images, en particulier dans des contextes où la segmentation est plus complexe.

Références

Blalock, D. W., J. J. G. Ortiz, J. Frankle, et J. V. Guttag (2020). What is the state of neural network pruning ?

Gong, Y., L. Liu, M. Yang, et L. Bourdev (2014). Compressing deep convolutional networks using vector quantization.

Han, S., H. Mao, et W. J. Dally (2016). Deep compression : Compressing deep neural networks with pruning, trained quantization and huffman coding.

Han, S., J. Pool, J. Tran, et W. J. Dally (2015). Learning both weights and connections for efficient neural networks. *Advances in Neural Information Processing Systems 2015-Janua*, 1135–1143.

Howard, A. G., M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, et H. Adam (2017). Mobilenets : Efficient convolutional neural networks for mobile vision applications.

Hu, H., R. Peng, Y.-W. Tai, et C.-K. Tang (2016). Network Trimming : A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures.

Hubara, I., M. Courbariaux, D. Soudry, R. El-Yaniv, et Y. Bengio (2016). Quantized neural networks : Training neural networks with low precision weights and activations.

Iandola, F. N., S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, et K. Keutzer (2016). Squeezenet : Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size.

Jonathan Frankle, M. C. (2019). The Lottery Ticket hypothesis. *Iclr 2*(10991), 2–6.

Latychevskaia, T. et H.-W. Fink (2015). Practical algorithms for simulation and reconstruction of digital in-line holograms. *Applied Optics 54*(9), 2424.

LeCun, Y., J. S. Denker, et S. A. Solla (1990). Optimal Brain Damage (Pruning).

Lee, N., T. Ajanthan, et P. H. S. Torr (2018). Snip : Single-shot network pruning based on connection sensitivity.

Lemaire, C., A. Achkar, et P. M. Jodoin (2019). Structured pruning of neural networks with budget-aware regularization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2019-June*, 9100–9108.

Li, H., A. Kadav, I. Durdanovic, H. Samet, et H. P. Graf (2017). Pruning filters for efficient convnets.

Luo, J. H., H. Zhang, H. Y. Zhou, C. W. Xie, J. Wu, et W. Lin (2019). ThiNet : Pruning CNN Filters for a Thinner Net. *IEEE Transactions on Pattern Analysis and Machine Intelligence 41*(10), 2525–2538.

Picart, P. et S. Montresor (2019). *Digital holography*. Elsevier Inc.

Ronneberger, O., P. Fischer, et T. Brox (2015). U-net : Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science 9351*, 234–241.

Scherrer, R., R. Govan, T. Quiniou, T. Jauffrais, H. Lemonnier, S. Bonnet, et N. Selmaoui-Folcher (2022). Real-time automatic plankton detection, tracking and classification on raw hologram. In *Computational Intelligence Methods for Bioinformatics and Biostatistics*, Cham, pp. 25–39. Springer International Publishing.

Shao, S., K. Mallery, et J. Hong (2020). Machine learning holography for measuring 3D particle distribution. *Chemical Engineering Science 225*(3), 2987–2999.

Vijayanagaram, R. (2020). Application of deep learning techniques to digital holographic microscopy for numerical reconstruction. *CEUR Workshop Proceedings 2535*, 1–15.

Yang, Z., Y. Cui, X. Yao, et S. Wang (2023). Gradient-based intra-attention pruning on pre-trained language models.

Zhang, X., X. Zhou, M. Lin, et J. Sun (2017). Shufflenet : An extremely efficient convolutional neural network for mobile devices.

Summary

The article presents a new filter pruning method tailored for deep models that generate images. Unlike traditional approaches that rely on pixel intensity in intermediate activations, the proposed method considers the spatial position of pixels in the image and employs binary masks to distinguish essential areas (objects) from the background. This technique allows for assessing filter importance by focusing on significant regions within the feature maps. The article compares several pruning criteria, demonstrating that our approach achieves high compression rates while maintaining a low mean squared error on reconstructed images.