Titre	**Green CNN : Optimisation multi-objective des architectures de réseaux de neurones convolutifs pour maximiser leurs performances et réduire leur consommation d'énergie**
Niveau du candidat	Ingénieur 5ème année, M2
Date de début	Septembre à Décembre 2024
Ville, Pays	Annecy-le-Vieux, *France*
Laboratoire	LISTIC - Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance - http://www.polytech.univ-savoie.fr/LISTIC
Encadrement	Directeur de thèse: Pr. Abdourrahmane ATTO Codirecteur de thèse : Dr. Khadija ARFAOUI
Contact:	Khadija ARFAOUI, khadija.arfaoui@univ-smb.fr

Description détaillée du sujet :

La Recherche d'Architecture de Neurones (Neural Architecture Search ou NAS) a révolutionné le domaine de l'apprentissage automatique en automatisant la conception des architectures. Déjà à ce jour, les méthodes de NAS ont surpassé les architectures conçues manuellement sur certaines tâches telles que la classification d'images [3][4], la détection d'objets [3] ou la segmentation sémantique [5]. La NAS peut être considérée comme un sous-domaine de l'AutoML et présente un chevauchement significatif avec l'optimisation des hyper-paramètres [13] et l'apprentissage méta [14].

Nous classifions les méthodes de NAS selon trois dimensions, notamment :

- **Définition de l'espace de recherche** : Pour permettre à NAS d'explorer efficacement les architectures possibles, il faut formuler les choix de conception tels que le nombre de couches, les canaux et les contraintes de ressources dans un espace de recherche. Cela permet aux humains de se concentrer sur le travail créatif tout en laissant la recherche aux algorithmes NAS [15].

- **Algorithmes de recherche** : Les méthodes de NAS ont évolué avec des algorithmes de recherche plus avancés tels que les algorithmes évolutionnaires et l'apprentissage par renforcement. Ces algorithmes permettent une exploration plus efficace de l'espace de recherche [15].

- **Évaluation et validation** : NAS nécessite des techniques d'évaluation et de validation appropriées pour comparer les performances des architectures nouvellement générées par rapport aux méthodes existantes. L'expertise humaine est cruciale à ce stade pour déterminer les exigences de conception de l'architecture et l'évaluation de la qualité de ses résultats. Les connaissances des experts guident le processus de recherche et d'optimisation [15].

La recherche d'une architecture optimale d'un réseau de neurones est une tâche fastidieuse, car les méthodes d'optimisation traditionnelles sont souvent non applicables [6]. En effet, les méthodes d'optimisation convexes classiques telles que la descente de gradient a tendance à être mal adaptées à cette problématique, car la mesure à optimiser est généralement une fonction non convexe et non différentiable [7]. De plus, les hyper-paramètres de l'architecture peuvent être discrets, catégoriques et/ou continus. Les hyper-paramètres typiques d'un réseau de

neurones convolutif (CNN), par exemple, incluent le nombre de couches de convolution et de réduction, le nombre de neurones par couche, le type d'optimiseur et la taux d'apprentissage. De plus, le temps nécessaire pour entraîner un modèle d'apprentissage automatique avec une configuration d'hyper-paramètres donnée sur un ensemble de données donné peut déjà être important, en particulier pour les ensembles de données modérés à grands.

En NAS, nous recherchons généralement une architecture de réseaux de neurones avec une configuration d'hyper-paramètres performante sur un seul ensemble de données, pour une tâche spécifique (classification, reconnaissance d'image ou autre). NAS a attiré une attention croissante ces dernières années, probablement stimulée par la popularité des algorithmes d'apprentissage profond, qui présentent des caractéristiques exigeantes (par exemple, le besoin de grandes quantités de données et de temps pour former les modèles, une complexité élevée des modèles et un mélange diversifié de types d'hyper-paramètres). Auparavant, les analystes avaient tendance à utiliser des méthodes simples pour rechercher les « meilleurs » paramètres d'hyper-paramètres. La plus basique d'entre elles est la recherche par grille [8]. Bien que cette stratégie soit simple à mettre en œuvre et à comprendre, ses performances sont influencées par le nombre d'hyper-paramètres à optimiser, et le nombre de valeurs choisies sur la grille. La recherche aléatoire [9] offre une alternative à la recherche par grille et a tendance à être populaire lorsque certains hyper-paramètres sont plus importants que d'autres. Des méthodes d'optimisation plus avancées ont également été proposées, telles que les méthodes de méta-apprentissage [10], des algorithmes d'optimisation basés sur la population [11] et les algorithmes d'apprentissage par renforcement (tels que HypRL [12]). Ces algorithmes offrent une perspective prometteuse, exploitant la nature itérative de l'évolution pour guider la recherche des architectures de neurones et l'optimisation et de leurs hyper-paramètres avec efficacité.

Dans ce domaine, le réseau neuronal convolutif (CNN) émerge comme une architecture prédominante. Ce dernier, structurellement complexe, se distingue par sa capacité à extraire des caractéristiques hiérarchiques à partir de données, offrant ainsi une puissante capacité de compréhension des structures complexes [1].

Cependant, la performance d'un CNN est étroitement liée à la configuration de ses paramètres, et la sélection judicieuse de ces derniers demeure une tâche cruciale [2]. Le choix approprié des paramètres ne se révèle pas seulement comme un exercice d'ajustement technique, mais comme un levier essentiel déterminant la capacité du CNN à apprendre et généraliser à partir des données. L'exploration systématique de cet espace paramétrique touche aux paramètres d'architecture incluent des éléments tels que la profondeur du réseau (couches de convolution, de pooling, de réduction et de mise en commun), la taille des filtres de convolution, le nombre de neurones dans chaque couche, ainsi que l'utilisation de techniques comme la régularisation (dropout, L1/L2) ou encore les méthodes d'initialisation des poids (Glorot, He, etc.)

En plus de la complexité des objectifs et méthodes de recherche NAS, de nombreux défis matériels peuvent être relevés dans ce contexte, notamment :

- **Complexité computationnelle** : La recherche d'architectures optimales nécessite l'évaluation de nombreuses configurations. Cela peut être extrêmement coûteux en termes de temps de calcul. Les ressources informatiques nécessaires pour explorer efficacement l'espace de recherche sont souvent considérables, en particulier lorsque l'on utilise des approches évolutives ou des algorithmes d'apprentissage par renforcement [16].
- **Taille des modèles** : Les architectures neuronales synthétisées par NAS peuvent être complexes et comporter un grand nombre de couches et de paramètres. Cela peut entraîner des modèles trop volumineux pour être déployés sur des plates-

formes à ressources limitées, telles que les appareils IoT, les téléphones mobiles ou les systèmes embarqués [17].

- **Hétérogénéité matérielle** : Différentes plates-formes matérielles ont des caractéristiques différentes (par exemple, GPU, CPU, FPGA). Concevoir des architectures qui fonctionnent bien sur diverses plates-formes tout en exploitant leurs avantages spécifiques est un défi complexe. Il faut tenir compte des spécificités matérielles telles que la mémoire, la bande passante et la puissance de calcul [18].
- **Consommation d'Énergie** : Les modèles d'IA, en particulier les réseaux de neurones profonds, nécessitent des ressources de calcul massives. Les centres de données et les serveurs utilisés pour l'entraînement des modèles consomment une quantité considérable d'électricité. Ainsi, trouver un équilibre entre la précision du modèle et l'énergie consommée par les ressources allouées est un défi majeur. Les architectures NAS doivent être adaptées à ces contraintes tout en maintenant des performances acceptables [19].

Ce dernier défi représente l'objectif de nombreuses recherches récentes visant à trouver un compromis entre la recherche d'une architecture de réseaux de neurones optimale et la réduction de l'énergie consommée par les ressources allouées. A cet effet, certaines architectures compactes sont proposées par conception manuelle afin de déployer des applications d'apprentissage profond sur des appareils à faible puissance de calcul et économiser l'énergie consommée, par exemple SqueezeNet [20], MobileNet [21] [22], GhostNet [23], ShuffleNet [24], Xception [25]. Cependant, la conception manuelle d'architectures plus efficaces repose fortement sur l'expérience des experts humains, sans parler de la conception du réseau dans des conditions de ressources limitées. Contrairement à la proposition de modèles à faible coût par conception, certaines approches de compression de modèle, telles que l'élagage du réseau, la quantification et la distillation des connaissances, sont largement utilisées pour obtenir des modèles compacts [26].

Malgré tous ces efforts à la recherche d'une architecture neuronale économe et compacte, la plupart des approches proposées dans la littérature restent coûteuses en termes de temps de calcul, en particulier lors de l'exploration d'un grand espace de recherche [26]. De plus, mesurer précisément la consommation d'énergie des modèles d'apprentissage profond sur des appareils réels est difficile et chronophage [27].

Dans ce projet de thèse, nous nous intéressons particulièrement aux approches d'optimisation multi-objectifs des architectures de réseaux de neurones permettant à la fois d'optimiser le choix des paramètres et composants de l'architecture en fonction des données à traiter ainsi que réduire la consommation d'énergie des ressources utilisées. Cette thèse se propose d'explorer l'optimisation multi-objective des architectures de réseaux de neurones convolutifs (CNN). L'objectif est d'améliorer les performances du CNN (en termes de précision, de Recall et de réduction du modèle) tout en minimisant l'énergie consommée par les ressources matérielles utilisées.

Nous envisageons, en premier lieu, d'explorer systématiquement l'espace des paramètres, y compris les paramètres d'architecture et les hyper-paramètres, en utilisant des approches évolutives. Nous visons à évaluer à l'impact de différentes configurations de ces paramètres sur les performances du CNN. Nous nous focaliserons, dans un second lieu, sur les architectures compactes et l'évaluation de leur consommation d'énergie afin de proposer une approche d'optimisation multi-objective permettant de trouver le compromis entre efficacité du CNN et consommation énergie réduite.

Pour valider nos travaux, nous prévoyons d'implémenter des cas d'utilisation concrets avec des ensembles de données réels. Cette phase de validation permettra de démontrer la pertinence, la capacité de généralisation et l'application tangible de nos recherches démontrant ainsi la pertinence, la capacité de généralisation et l'application tangible des avancées obtenues dans le domaine de l'optimisation des hyper-paramètres pour les CNN.

Références:

[1] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
[2] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.
[3] Zhao Zhong, Zichen Yang, Boyang Deng, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Blockqnn: Efficient block-wise neural network architecture generation. arXiv preprint, 2018b.
[4] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Aging Evolution for Image Classifier Architecture Search. In AAAI, 2019.
[5] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 8713–8724. Curran Associates, Inc., 2018.
[6] Luo G (2016) A review of automatic selection methods for machine learning algorithms and hyper-parameter values. Netw Model Anal Health Inform Bioinform 5(1):18.
[7] Stamoulis D, Chin T-W, Prakash AK, Fang H, Sajja S, Bognar M, Marculescu D (2018) Designing adaptive neural networks for energy-constrained image classification. Proceedings of the international conference on computer-aided design (pp 1-8).
[8] Montgomery DC (2017) Design and analysis of experiments. Wiley, Hoboken.
[9] Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(1):281–305
[10] Bui K-HN, Yi H (2020) Optimal hyperparameter tuning using meta-learning for big traffic datasets. In: Lee W et al. (ed) 2020 IEEE international conference on big data and smart computing (bigcomp 2020) pp 48–54. IEEE.
[11] J aderberg M, Dalibard V, Osindero S, Czarnecki WM, Donahue J, Razavi A, et al (2017) Population based training of neural networks.
[12] Jomaa HS, Grabocka J, Schmidt-Thieme L (2019) Hyp-rl: Hyper-parameter optimization by reinforcement learning.
[13] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. Automatic Machine Learning: Methods, Systems, Challenges. Springer, 2019.
[14] Joaquin Vanschoren. Meta-learning. In Hutter et al. (2019), pages 39–68.
[15] Alonso-García, M., & Corchado, J. M. (2023). Neural Architecture Search: Practical Key Considerations. In Distributed Computing and Artificial Intelligence (pp. 314). Springer.
[16] Alonso-García, M., & Corchado, J. M. (2023). Neural Architecture Search: Practical Key Considerations. In Distributed Computing and Artificial Intelligence (pp. 314). Springer.
[17] Zoph, B., & Le, Q. V. (2017). Neural architecture search with reinforcement learning. arXiv preprint:1611.01578.
[18] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning (pp. 6105-6114).
[19] Pham, H., Guan, M. Y., Zoph, B., Le, Q. V., & Dean, J. (2018). Efficient neural architecture search via parameter sharing. In International Conference on Machine Learning (pp. 4095-4104).
[20] Iandola, F., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2017). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size. arXiv: Computer Vision and Pattern Recognition.
[21] Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., et al. (2019). Searching for mobilenetv3. In Proceedings of the IEEE international conference on computer vision (pp. 1314–1324).

[22] Howard, G. A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv: Computer Vision and Pattern Recognition.
[23] Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020). Ghostnet: More features from cheap operations. In 2020 IEEE/CVF conference on computer vision and pattern recognition (pp. 1580–1589).
[24] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Computer vision and pattern recognition.
[25] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In CVPR.
[26] Lu, L., & Lyu, B. (2021). Reducing energy consumption of Neural Architecture Search: An inference latency prediction framework. Sustainable Cities and Society, 67, 102747.
[27] Gebauer, H., Klimach, H., & Porrmann, M. (2020). Energy Efficiency of Deep Learning Inference on Mobile Devices: A Survey. IEEE Access, 8, 154001-154020.