Quantization-Aware Neural Architecture Search with Hyperparameter Optimization for Industrial Predictive Maintenance Applications

Nick van de Waterlaat*, Sebastian Vogel, Hiram Rayo Torres Rodriguez, Willem Sanberg, Gerardo Daalderop
NXP Semiconductors, Eindhoven, The Netherlands
*nick.vande.waterlaat@nxp.com

***Abstract*—Optimizing the efficiency of neural networks is crucial for ubiquitous machine learning on the edge. However, it requires specialized expertise to account for the wide variety of applications, edge devices, and deployment scenarios. An attractive approach to mitigate this bottleneck is Neural Architecture Search (NAS), as it allows for optimizing networks for both efficiency and task performance. This work shows that including hyperparameter optimization for training-related parameters alongside NAS enables substantial improvements in efficiency and task performance on a predictive maintenance task. Furthermore, this work extends the combination of NAS and hyperparameter optimization with INT8 quantization to enhance efficiency further. Our combined approach, which we refer to as Quantization-Aware NAS (QA-NAS), allows for further improvements in efficiency on the predictive maintenance task. Consequently, our work shows that QA-NAS is a promising research direction for optimizing neural networks for deployment on resource-constrained edge devices in industrial applications.**

I. Introduction

The overwhelming success of neural networks (NNs) has led to an increasing demand for their deployment on resource-constrained edge devices. Consequently, optimizing their efficiency has become crucial for many deployment scenarios. A popular approach for optimizing the task performance and efficiency of a network architecture of interest, henceforth called the seed network, is Neural Architecture Search (NAS). After defining a search space with variations of the seed network, NAS aims to find architectures with optimal trade-offs between task performance and efficiency.

Various approaches exist besides NAS for optimizing the task performance and efficiency of NNs, such as hyperparameter optimization (HPO) for training-related parameters and quantization. While HPO for training-related parameters optimizes their task performance by altering their training procedure, quantization optimizes their efficiency by reducing their numerical precision. Typically, these three approaches are treated in isolation. However, they all play a relevant, intertwined, and potentially conflicting role in optimizing NNs.

This work proposes two easy-to-integrate extensions of NAS based on HPO and quantization that enhance task performance and efficiency. Specifically, the contributions of this work are

- a comparison of NAS with and without HPO for training-related parameters, demonstrating that extending NAS with HPO enables substantial improvements in both task performance and efficiency,
- a straightforward and effective approach for model compression that extends NAS and HPO with quantization, which we refer to as Quantization-Aware NAS (QA-NAS) with HPO,
- and a comparison of NAS with HPO and QA-NAS with HPO, demonstrating the added benefit of QA-NAS as it further enhances efficiency.

II. NAS with Hyperparameter Optimization

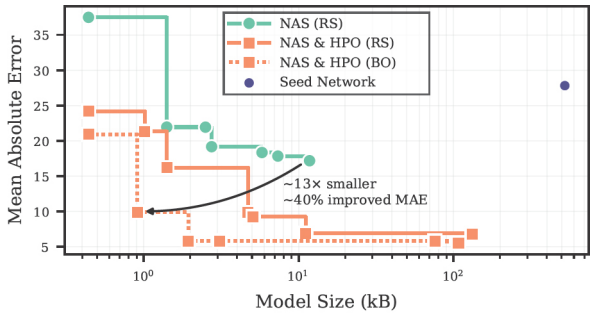
We demonstrate that HPO for training-related parameters allows for further improvement upon NAS by applying NAS with and without HPO for two different seed networks on a popular predictive maintenance task, namely the NASA Turbofan Jet Engine dataset [3]. We use a Long Short-Term Memory (LSTM) [1] and Temporal Convolutional Network (TCN) [2] as seed networks. For evaluating the task performance of each network candidate (trial), we use the Mean Absolute Error (MAE) because the dataset consists of a regression task in which the objective is to predict the number of remaining cycles until engine failure. Additionally, we use the model size as the efficiency metric. This metric acts as a secondary objective next to task performance. For HPO, we include the learning rate and its scheduler, optimizer, initialization, and regularization. Lastly, we run NAS for 100 trials with 100 epochs per trial with both random search (RS) and Bayesian optimization (BO).

Comparisons of the Pareto-optimal networks found by applying NAS with and without HPO to the LSTM and TCN seed network are shown in Fig. 1. First, the Pareto-dominance shown by NAS with HPO using RS, abbreviated "NAS & HPO (RS)", compared to NAS without HPO using RS, abbreviated "NAS (RS)", indicates the importance of the training paradigm when using NAS for model compression. Secondly, the overall Pareto-dominance shown by NAS with HPO using BO, abbreviated "NAS & HPO (BO)", indicates the significance of a sophisticated search strategy in the search space extended by HPO. For instance, NAS & HPO (BO) finds a network $\sim$13$\times$ smaller than NAS (RS) with $\sim$40% improved MAE (Fig. 1a), which can be essential for deployment on edge devices.

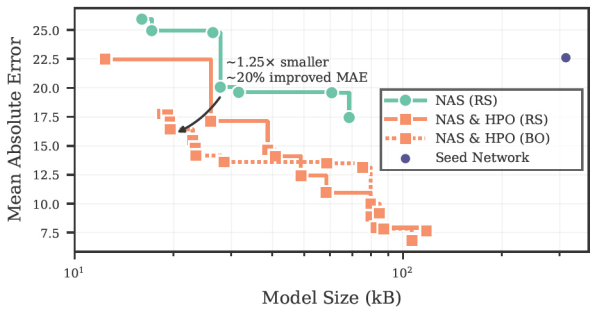
III. Quantization-Aware NAS

Although NAS with HPO has shown strong results in Section II, edge devices can further benefit from INT8 format for deployment. To that end, we extend the combination of NAS and HPO with INT8 quantization, which we refer to

978-3-9819263-7-8/DATE23/© 2023 EDAA

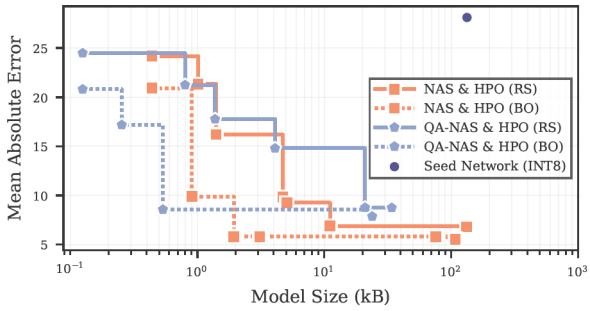


(a) LSTM

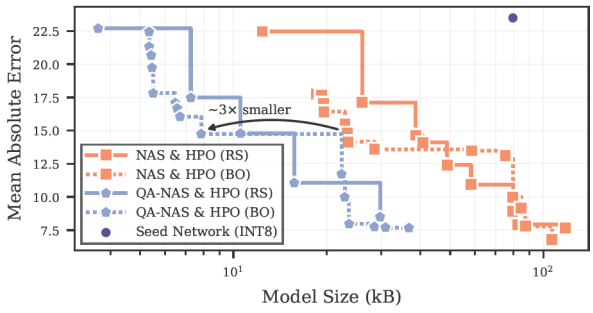


(b) TCN

Fig. 1: The Pareto curves found by NAS with and without HPO for training-related parameters for the LSTM and TCN seed network on the NASA Turbofan Jet engine dataset.



(a) LSTM



(b) TCN

Fig. 2: The Pareto curves found by NAS with HPO and QA-NAS with HPO through random search (RS) and Bayesian optimization (BO) for the LSTM and TCN seed network on the NASA Turbofan Jet Engine dataset.

as QA-NAS with HPO. We demonstrate the added benefit of QA-NAS with HPO over NAS with HPO by applying both approaches to the LSTM and TCN seed networks. For QA-NAS, we perform INT8 quantization for the weights and activations with per-channel min-max quantization, which is the standard quantization strategy in TFLite [4].

Comparisons of the Pareto-optimal networks found by applying NAS with HPO and QA-NAS with HPO for both RS and BO are shown in Fig. 2. First, these comparisons highlight the added benefit of QA-NAS over NAS as significantly smaller networks can be found with equivalent MAE across all sampled regions for both the LSTM and TCN. For instance, QA-NAS & HPO (BO) finds a network $\sim$3$\times$ smaller than NAS & HPO (BO). However, this finding does not hold for LSTMs larger than $\sim$1.25 kB. These larger FP32 networks may outperform quantized networks in both size and MAE. Despite quantization being more challenging in this case, QA-NAS & HPO (BO) still outperforms NAS & HPO (BO) for networks smaller than $\sim$1.25 kB. Noteworthy, QA-NAS & HPO (RS) does not showcase similar behavior. As suggested by our results, a more sophisticated search strategy that considers performance (e.g., BO) may help avoid network configurations that are challenging to quantize, thereby likely leading to improved performance.

IV. Conclusion

Optimizing the efficiency of NNs is crucial for ubiquitous machine learning on the edge. A popular approach to optimize efficiency is to search for more efficient network variants through NAS. This work has shown that extending NAS with HPO for training-related parameters can enable substantial improvements in efficiency and task performance, as validated on an industrial predictive maintenance task.

Furthermore, we have extended NAS and HPO with INT8 quantization, which we refer to as QA-NAS with HPO. Our results on the predictive maintenance task demonstrate that this extension can enable further improvements in efficiency. Consequently, this work has laid out a straightforward and effective approach for further enhancing task performance and efficiency. Despite the surprisingly little attention this approach has received from the research community, our results demonstrate that it can lead to vast improvements in task performance and efficiency. Therefore, we strongly encourage future work in this direction to further boost industry-wide adoption of efficient NNs for resource-constrained edge devices.

References

[1] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation,* 9(8), pp.1735-1780.

[2] Bai, S., Kolter, J.Z. and Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271.*

[3] Saxena, A. and Goebel, K., 2008. Turbofan engine degradation simulation data set. *NASA Ames Prognostics Data Repository,* pp.1551-3203.

[4] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. and Kalenichenko, D., 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2704-2713).