# NEURAL NETWORK ARCHITECTURES GUIDED BY REAL DATA OR SIMPLIFIED MODELS

Clara Oliveira, Jérémy Morlier
*IMT Atlantique — Department of Mathematical and Electrical Engineering (MEE)*

**Abstract**

Advances in deep neural networks have created highly accurate models, but also challenges in efficiency and hardware adaptation. Performance is often evaluated using simplified metrics such as FLOPs, which ignore real device behaviour, or using latency and energy measurements, which incur computational cost. This raises an open question: do these different strategies lead NAS to the same architectures, or do they diverge? To explore this, we implement a hardware-aware NAS framework based on ResNet18 and compare three optimisation strategies: (i) FLOPs-only, (ii) latency–energy, and (iii) a combined multi-objective score. Our hypothesis is that simplified metrics may diverge from real measurements depending on the search space. Through controlled experiments on a GTX 1060, we analyse how each metric influences architecture ranking and selection.

**Keywords:** Neural Architecture Search (NAS), Hardware-aware learning, Model-based optimization, Embedded systems

## INTRODUCTION

Before the emergence of Neural Architecture Search (NAS), neural networks were designed manually through trial and error, experience, and human intuition. NAS was introduced in [1] as a technique that automates the design of neural networks in order to identify architectures tailored to specific tasks. To optimise this search process, several strategies have been proposed (RL-based, Bayesian, probabilistic and differentiable methods) [2, 3, 4, 5, 6].

However, evaluating hardware-related metrics for each candidate architecture is expensive and time-consuming [2]. For this reason, many NAS methods rely on analytical predictors to estimate latency or energy, avoiding the cost of real measurements. Despite their limitations, such simplified metrics remain popular because they are fast to compute, hardware-agnostic, and suitable for large-scale NAS. An alternative is to rely on real hardware measurements such as latency and energy, but these incur additional computational overhead and are more sensitive to variability. This raises a fundamental question: can analytical metrics such as FLOPs, which ignore hardware behaviour, reliably guide architecture selection compared to real measurements?

The objective of this work is to evaluate different strategies for hardware-aware NAS, comparing analytical guidance with real hardware feedback. Our working hypothesis is that these different metrics may lead to different architecture rankings and selections.

## MATERIALS AND METHODS

The objective of this work is to evaluate strategies for hardware-aware NAS by comparing analytical guidance (FLOPs-based predictors) and real hardware measurements (latency and energy). Our hypothesis is that these metrics may lead to different architecture rankings depending on the search space. To evaluate this, we implement a hardware-aware NAS pipeline as shown in 1, based on ResNet18 and integrating analytical and real GPU metrics.

Our method includes three steps: defining the search space, running the NAS controller, and integrating hardware metrics. We identified the parameters forming the NAS search space (Table 1) on CIFAR-10 [7], while training hyperparameters are described below. We implemented a NAS code integrating the selected metrics, and all hardware metrics (latency, FLOPs and energy) were computed using batch size 1 and 32×32 input images.

The first modification applied to ResNet18 consists of changing the conv1 layer, adjusting its kernel size and stride for each sampled architecture. All convolutional channels are then scaled by width_mult to adjust network width and computational cost. Latency is measured using CUDA Events, and energy via NVML power readings. FLOPs represent the theoretical computational cost and can be computed before training.

The training hyperparameters are fixed using a learning rate of 0.01, momentum 0.9, batch size 128, five epochs for low-fidelity accuracy (E_quick) and 100 epochs for final training (E_full). Each architecture is trained for 5 epochs to obtain low-fidelity accuracy [2] and scored using weights a, b and c. We compute three scores to compare hardware-aware optimisation strategies: a FLOPs-only score (Equation 2), a latency–energy score (Equation 3), and the combined multi-objective score (Equation 1). These scores show how guidance metrics influence architecture selection, and the best architecture is then trained for 100 epochs to obtain its final accuracy.

$$\text{score}_{\text{full}} = \text{acc} - a\,\text{lat} - b\,\text{FLOPs} - c\,\text{energy} \quad (1)$$

$$\text{score}_{\text{FLOPs}} = \text{acc} - b\,\text{FLOPs} \quad (2)$$

* Corresponding author. E-mail: clara.afonso-oliveira@imt-atlantique.net

$$\text{score}_{\text{lat-energy}} = \text{acc} - a\,\text{lat} - c\,\text{energy} \quad (3)$$

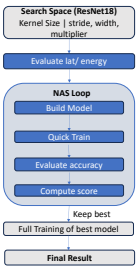


Figure 1. Pipeline of the NAS process.

| Hyperparameter | Values | Meaning |
|---|---|---|
| kernel | {3, 5} | Size of the first convolution filter; affects receptive field and FLOPs. |
| stride | {1, 2} | Downsampling factor; reduces resolution and latency. |
| width_mult | {0.5, 1.0, 1.5} | Scales channels; controls width and computational cost. |
| a | 0.1 | Latency penalty weight in the score. |
| b | 0.001 | FLOPs penalty weight in the score. |
| c | 0.01 | Energy penalty weight in the score. |

Table 1. NAS hyperparameters.

## RESULTS

Using the three scoring strategies (FLOPs-only, latency–energy and the full multi-objective score), we observed that they did not all select the same architecture. Both the FLOPs-only score and the full score selected Model 5 (kernel = 3, stride = 2, width_mult = 1.0), due to its low computational cost and balanced trade-off, as illustrated in Fig. 2. In contrast, the latency–energy score favoured Model 6 (3, 1, 1.5), which achieved the highest quick-training accuracy while maintaining moderate latency and energy, as shown in Fig.3. After full training (100 epochs), the architecture selected by the full score (Model 5) reached an accuracy of 0.83, with 3.16 ms latency, 34.41 MFLOPs, and 23.37 J of energy consumption.

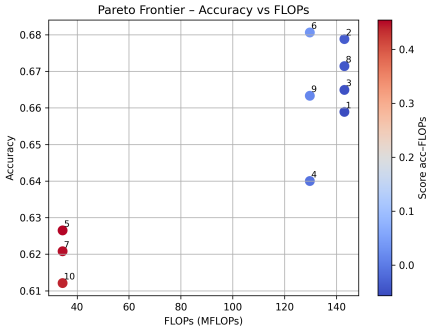


Figure 2. Accuracy vs FLOPs for the 10 architectures, highlighting Model 5 as the best FLOPs-based trade-off.

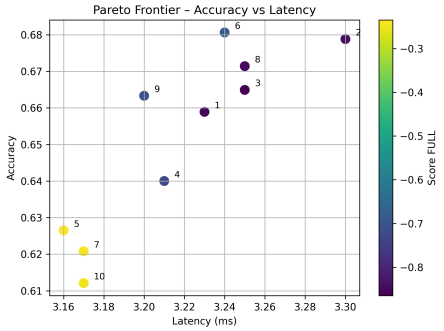


Figure 3. Accuracy vs latency coloured by the full score, with Model 5 emerging as the best global trade-off.

## CONCLUSION

These results confirm that hardware-aware evaluation significantly impacts neural architecture selection. FLOPs provide a limited approximation of real execution cost, while latency and energy reveal differences that theoretical metrics cannot capture. Multi-objective optimisation offers a more reliable trade-off between accuracy and efficiency, although its behaviour depends on the search space. Future work should explore broader architectures, alternative hardware platforms and faster surrogate metrics to scale hardware-aware NAS more effectively.

## References


[1] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2017.

[2] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," 2019.

[3] L. Yang, Z. Yan, M. Li, H. Kwon, L. Lai, T. Krishna, V. Chandra, W. Jiang, and Y. Shi, "Co-exploration of neural architectures and heterogeneous asic accelerator designs targeting multiple tasks," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2020.

[4] H. Shi, R. Pi, H. Xu, Z. Li, J. T. Kwok, and T. Zhang, "Bridging the gap between sample-based and one-shot neural architecture search with bonas," 2020.

[5] Z. Yan, X. Dai, P. Zhang, Y. Tian, B. Wu, and M. Feiszli, "Fp-nas: Fast probabilistic neural architecture search," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15134–15143, 2021.

[6] L. Yuan, Z. Huang, and N. Wang, "Prednas: A universal and sample efficient neural architecture search framework," 2022.

[7] P. Radiuk, "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets," *Information Technology and Management Science*, vol. 20, pp. 20–24, 12 2017.