ETH zürich

HANNA: Hardware-Aware Neural Network Analysis

Mark Vero, Thorir Mar Ingolfsson, Xiaying Wang, Lorenzo Lamberti, Matteo Spallanzani, Luca Benini

Neural Architecture Search (NAS) is the deep-learning-specific variant of model selection. The goal of NAS is **discovering those network topologies that have good task accuracy, i.e., that are most effective**.

NAS spaces are vast:
- many degrees of freedom (layers, connectivity, …);
- many options for each degree of freedom.

Several **stochastic and probabilistic algorithms** to explore NAS spaces have been proposed:
- Evolutionary Algorithms;
- Reinforcement Learning;
- Gradient-Based Learning;
- Bayesian Methods;
- Random Network Generation.

They **require training candidate networks** for several epochs: this is **time-consuming and computationally expensive**.

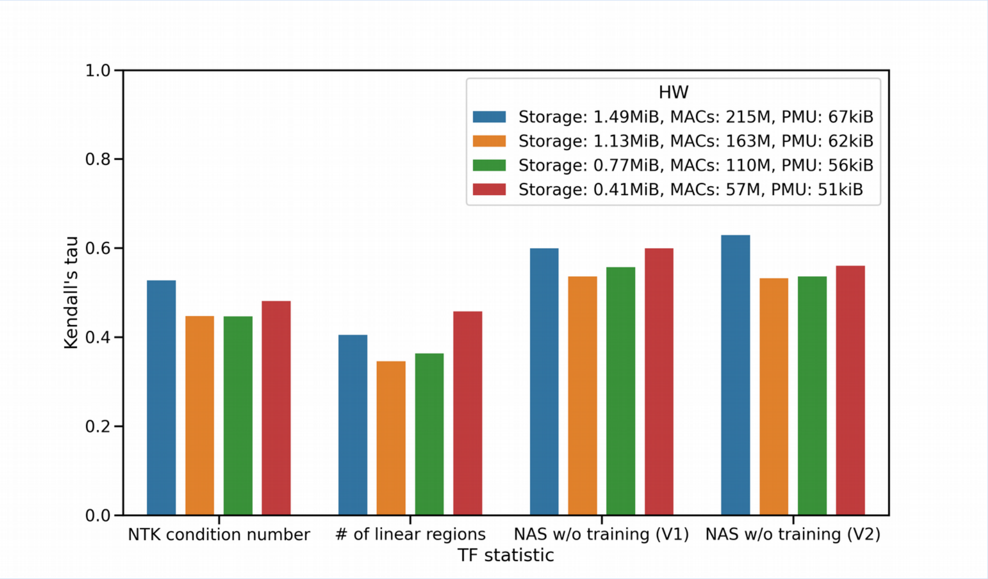
When thinking to a **DNN f_λ as a computer program**, we model it as a **computational graph G_λ**. To design better HW for DNNs, we need to answer the following question: "**What are the features shared by good programs?**" Similarly, to design better DNNs for constrained HW, we need to answer the following question: **"What are the features shared by good HW-constrained programs?"** We analyse data sets of network topologies (e.g., **NAS-Bench-201**) whose task accuracy is known. We quantify the similarity between programs using **distances between graphs**:

$$k_{\lambda_1,\lambda_2} \coloneqq k\left(G_{\lambda_1}, G_{\lambda_2}\right).$$

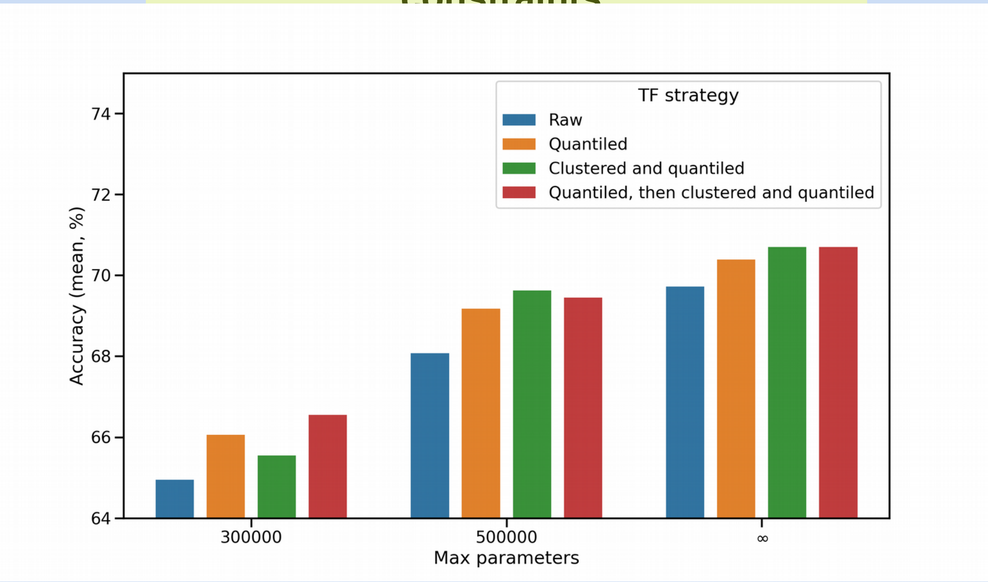
"How can we discriminate clusters of bad programs from clusters of good programs when accuracy is not known?" **Training-free (TF) statistics** measure properties of DNNs before any epoch of training is run. The hypothesis is that **good values of TF statistics correlate with good task accuracy**:

$$s(f_\lambda) \propto a(f_\lambda).$$

The predictive power of TF statistics appears to be independent of HW constraints.



The predictive power of TF statistics is enhanced by the cluster latent variable over different HW constraints



Effective HW-constrained programs use non-destructive operators (e.g., no pooling).