µNAS: Constrained Neural Architecture Search for Microcontrollers

Edgar Liberis
el398@cam.ac.uk
University of Cambridge
Cambridge, United Kingdom

Łukasz Dudziak
Samsung AI Centre Cambridge
Cambridge, United Kingdom

Nicholas D. Lane
University of Cambridge
Samsung AI Centre Cambridge
Cambridge, United Kingdom

IoT devices are powered by microcontroller units (MCUs) which are extremely resource-scarce: a typical MCU may have an underpowered processor and around 64 KB of memory and persistent storage. Designing neural networks for such a platform requires an intricate balance between keeping high predictive performance (accuracy) while achieving low memory and storage usage and inference latency. This is extremely challenging to achieve manually, so in this work, we build a neural architecture search (NAS) system, called µNAS, to automate the design of such small-yet-powerful MCU-level networks. More specifically:

- *We propose and motivate a multiobjective constrained NAS algorithm suitable for finding MCU-level architectures, called µNAS.* It is assembled out of:
  1. a granular search space;
  2. a set of constraints that accurately capture resource scarcity of microcontroller platforms;
  3. a search algorithm capable of optimising for multiple objectives in the said search space;
  4. network pruning, to obtain small accurate models.
- *We perform ablation studies to quantitatively justify the inclusion of objectives and network pruning in µNAS.*
- *We conduct extensive experiments* over five microcontroller-friendly image classification tasks to demonstrate the superior performance of µNAS.

At its core, NAS is an optimisation problem, where we seek to find a neural network (from the search space) that maximises some objective function, such as the accuracy on the target dataset. Resource constraints can be treated as extra objectives with penalty terms. We include four objectives, three of which are resource constraints. This makes µNAS explicitly target the three primary aspects of resource scarcity of MCUs: the size of RAM, persistent storage and processor speed.

(1) Top-1 accuracy on the validation set.

(2) Peak memory usage (PMU). Surprisingly, it is not straightforward to compute the PMU of a network when it has branches. Branches permit different execution orders, which changes which tensors are present in memory at any given time, which in turn affects the peak memory usage. Thus, to most accurately capture the minimum amount of memory required to run a network, we consider topological orders of a network's computational graph to find the one with smallest peak memory usage.

(3) Model size. All static data, incl. weights of a neural network, are stored in the persistent (Flash) memory of an MCU. We reduce the storage usage by quantising parameters to 8 bits = 1 byte.

(4) Latency. For GPU-targeting NAS, proxy metrics, such as FLOPs, fail to account for scheduling, caching, parallelism and other properties of the software or hardware. However, MCUs typically lack these performance-enhancing features: the software runs on a single-core processor at a fixed frequency with no data caching. We settle on using a number of multiply-accumulate operations (MACs) as a proxy for model latency, which we verified to be an accurate estimator.

Another design element unique to MCU-level NAS is **a highly granular search space**. Our search space should consists of small models, with few restrictions on layer connectivity (to allow for powerful-yet-small feature extractors) and granular hyperparameters (a small in e.g. the number of channels can tip the model over the memory budget). The search is based on an aging evolution algorithm, which maintains a population of architectures (initially random) and proceeds by applying changes to a chosen architecture (*morphisms*) to produce a derivative (child) network. Each child network is trained from scratch with structured pruning and replaces the oldest architecture in a population.

We evaluate µNAS on five image classification problems: MNIST, CIFAR-10, Chars74K, Fashion MNIST and Speech Commands (audio classification via spectrograms). µNAS represents a significant advance in resource-efficient models, especially for "mid-tier" MCUs with memory requirements ranging from 0.5 KB to 64 KB. We show that on a variety of image classification datasets µNAS is able to (a) improve top-1 classification accuracy by up to 4.8%, or (b) reduce memory footprint by 4–13×, or (c) reduce the number of multiply-accumulate operations by at least 2×, compared to existing specialist literature and resource-efficient models.

In future work, we intend to scale µNAS to larger problems (ImageNet) and improve the search time through amortising the training across multiple models.

Acknowledgements. This work was supported by Samsung AI and by the UK's Engineering and Physical Sciences Research Council (EPSRC) with grants EPM50659X1 and EPS0015301 (the MOA project) and the European Research Council via the REDIAL project (Grant Agreement ID: 805194).