SparseLoom: Multi-DNN Inference of Sparse Models on Edge SoCs

Jiawei Luo, Di Wu, Simon Dobson, Blesson Varghese
School of Computer Science, University of St Andrews, Scotland, UK
{jl425, dw217, simon.dobson, bv6}@st-andrews.ac.uk

Modern edge applications, such as augmented reality and autonomous driving, comprise multiple deep neural network (DNN) inference tasks that execute concurrently to support diverse functionalities. These tasks may have distinct Service Level Objectives (SLOs), depending on application requirements. Such requirements stem from varying user expectations, application criticality, such as real-time responsiveness, or resource constraints, such as latency budgets and energy limits.

Given the range of diverse SLOs, a single task may be serviced by multiple DNN model variants. Each variant is a sparse model that is compressed from a common base model using techniques, such as pruning or quantization, and offers different accuracy-latency trade-offs. Consequently, each task maintains a sparse model zoo - a collection of sparse DNN variants. The model zoo allows for the selection of the most suitable variant that satisfies the task-specific SLO.

In edge deployments, DNN variants are executed on heterogeneous processors within a single System-on-Chip (SoC), comprising Central Processing Units (CPUs), Graphics Processing Units (GPUs), and Neural Processing Units (NPUs). These processors offer different sparse computational capabilities and therefore exhibit different performance characteristics for the same sparse variant. As a result, variant selection and processor placement become tightly coupled decisions in deployments with heterogeneous processors.

However, existing systems typically support only a single model or a small number of sparse variants per task, which often leads to a high SLO violation rate. Moreover, they do not account for processors' sparse computation capabilities during task placement, resulting in suboptimal processor placement and throughput degradation.

To address these limitations, we introduce the idea of model stitching. It constructs stitched variants by combining subgraphs (i.e., consecutive layer blocks) from different sparse models without re-training, thereby expanding the latency–accuracy trade-off space and reducing SLO violation rate. To efficiently support model stitching, we propose SparseLoom, a multi-DNN inference system that addresses three key challenges: (i) a performance estimator that predicts the accuracy and latency of stitched variants to reduce the profiling cost induced by the exponential number of possible stitched variants, (ii) a sparsity-aware optimizer that jointly selects variants and processor placement, and (iii) a hot-subgraph preloader that pre-loads subgraphs based on hotness scores under a global memory budget while reducing memory overheads incurred in loading subgraphs.

We show experimentally on a range of Intel and NVIDIA edge SoCs that SparseLoom reduces SLO violations by up to 74%, improves throughput by up to 2.31×, and reduces memory overhead by an average of 28% compared to state-of-the-art multi-DNN inference systems.