

Energy-Aware Scheduling for Edge Inference with Latency Guarantees

1st Given Name Surname
Dept. of Electrical Engineering
Example University
City, Country
first.author@example.edu

2nd Given Name Surname
School of Computing
Another University
City, Country
second.author@example.org

3rd Given Name Surname
Research Division
Example Labs Inc.
City, Country
third.author@example.com

Abstract—Edge devices increasingly run deep neural network inference under strict latency budgets while being limited by battery capacity. This paper proposes an energy-aware scheduler that selects a per-request model variant and processor frequency so that the end-to-end deadline is met with the lowest expected energy. We formulate the choice as a constrained Markov decision process, derive a lightweight threshold policy, and evaluate it on three commodity edge boards. Compared with a performance governor, the scheduler reduces energy per inference by 31 % on average while missing fewer than 0.5 % of deadlines.

Index Terms—edge computing, energy efficiency, DVFS, deep learning inference, real-time scheduling

I. INTRODUCTION

DEEP learning inference has moved from data centres to phones, cameras and sensors, where every joule matters. Dynamic voltage and frequency scaling (DVFS) and model selection are the two most effective knobs for trading latency against energy, but they are usually tuned independently and offline [1], [2].

This paper makes three contributions. First, we model the joint selection of a model variant and a core frequency as a constrained Markov decision process (CMDP) whose constraint is the deadline-miss rate (Section III). Second, we show that the optimal policy has a threshold structure that can be stored as a small table and evaluated in microseconds. Third, we implement the scheduler on three edge boards and report energy, latency and accuracy (Section IV).

II. RELATED WORK

Prior work on efficient inference falls into two groups. Model-side techniques such as pruning, quantisation and early exit reduce the work per request [1], [3]. System-side techniques such as DVFS governors and offloading change how the work is executed [2], [4]. Our scheduler sits between the two: it treats the available model variants as actions and chooses among them online.

III. METHOD

A. System Model

Requests arrive as a renewal process with mean inter-arrival time $1/\lambda$. For each request the scheduler picks a model variant

$m \in \mathcal{M}$ and a frequency $f \in \mathcal{F}$. The latency $L(m, f)$ and energy $E(m, f)$ are measured offline; the accuracy $A(m)$ depends on the variant only.

B. Problem Formulation

We minimise the long-run average energy subject to a bound on the deadline-miss probability and a minimum average accuracy:

$$\begin{aligned} \min_{\pi} \quad & \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{\pi} [E(m_t, f_t)] \\ \text{s.t.} \quad & \Pr [L(m_t, f_t) > D] \leq \delta, \quad \bar{A} \geq A_{\min}. \end{aligned} \quad (1)$$

Using a Lagrangian relaxation of (1), the problem reduces to an unconstrained MDP whose per-step cost is $E(m, f) + \mu \mathbb{I}[L(m, f) > D] - \nu A(m)$, and the resulting policy is a threshold on the remaining slack time [5].

C. Algorithm

- 1: Offline: profile $L(m, f)$, $E(m, f)$ for all (m, f)
- 2: Solve the relaxed MDP by value iteration; store thresholds
- 3: **for** each request with deadline D **do**
- 4: $s \leftarrow D$ - queueing delay
- 5: $(m, f) \leftarrow \text{lookup}(s)$ {threshold table}
- 6: set frequency f ; run model m
- 7: **end for**

IV. EXPERIMENTAL RESULTS

A. Setup

We evaluate on three boards (Table I) running image classification with four variants of a MobileNet-style network. The deadline is 50 ms and the target miss rate $\delta = 0.5\%$.

TABLE I
EVALUATION PLATFORMS

Board	CPU cores	Freq. range (GHz)	Idle power (W)
Board A	4	0.6–1.8	1.9
Board B	6	0.4–2.2	2.6
Board C	8	0.5–2.0	3.1

B. Energy and Latency

Fig. 1 shows energy per inference. The threshold policy tracks the offline optimum within 4 % and beats the performance governor by 31 % on average, because it slows the processor whenever the slack time allows and switches to a smaller variant only when slowing down would miss the deadline.

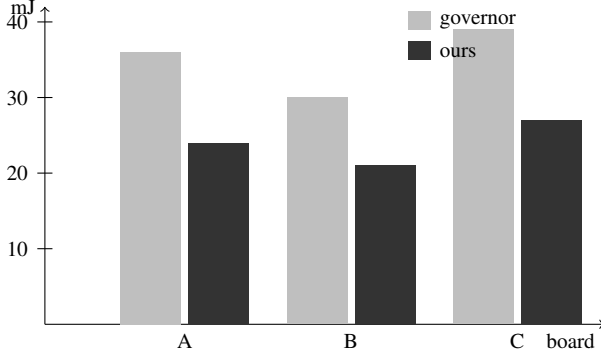


Fig. 1. Energy per inference for the performance governor and the proposed scheduler on the three boards.

C. Discussion

The accuracy constraint was active on Board A only, where the scheduler chose the smallest variant for 12 % of requests. Raising $A_{\min}$ by one point increased energy by 6 %, which quantifies the accuracy–energy trade-off the operator controls through ν in (1).

V. CONCLUSION

We presented a scheduler that jointly chooses the model variant and processor frequency for edge inference, proved that the optimal policy is a threshold rule, and showed energy savings of 31 % with negligible deadline misses. Future work includes multi-tenant devices and learning the profiles online.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their comments and the Example University systems group for access to the test boards.

REFERENCES

- [1] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.
- [2] N. D. Lane *et al.*, “DeepX: A software accelerator for low-power deep learning inference on mobile devices,” in *Proc. 15th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, 2016, pp. 1–12.
- [3] S. Teerapittayanon, B. McDanel, and H. T. Kung, “BranchyNet: Fast inference via early exiting from deep neural networks,” in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, 2016, pp. 2464–2469.
- [4] Y. Kang *et al.*, “Neurosurgeon: Collaborative intelligence between the cloud and mobile edge,” in *Proc. 22nd Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLOS)*, 2017, pp. 615–629.
- [5] E. Altman, *Constrained Markov Decision Processes*. Boca Raton, FL, USA: Chapman & Hall/CRC, 1999.