SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Inference-Time Energy Minimization Through Learnable Numerical Precision In Activation Computation

N. Nivetha[1*], R. Manjula[2], Sayfiddinova Muniskhon Fakhriddin kizi[3], Dr. Abhishek Sharma[4]

[1*]Assistant Professor, Department of Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Tamil Nadu, India. E-mail: nivethan@maher.ac.in
[2]Assistant Professor, Department of Commerce, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Tamil Nadu, India. E-mail: manjular@maher.ac.in
[3]Turan International University, Namangan, Uzbekistan. E-mail: msayfiddinova94@gmail.com, https://orcid.org/0009-0006-3474-4480
[4]Assistant Professor, Kalinga University, Naya Raipur, Chhattisgarh, India. E-mail: ku.abhisheksharma@kalingauniversity.ac.in, https://orcid.org/0009-0007-5855-788X
*Corresponding author: Email: nivethan@maher.ac.in

Abstract

The total energy spent on inference for a neural network is comprised mainly of arithmetic operations performed on activation tensors, which are non-linearly dependent on the numerical precision employed. Fixed precision quantization makes use of fixed-bit widths for activation operations, thus ignoring the varying demands on precision (spatially and channel-wise) within a single layer. In this work, propose LearnPrec, a framework for minimizing inference-time energy using learnable precision for activations. introduce the per-activation channel precision selector, a small binary network, which, together with end-to-end learning with a combined accuracy energy objective, decides upon the use of 8-bit or 4-bit computation per activation channel, independent of all others. The precision selector operates during inference time and makes a binary decision per activation channel for each input batch of data in a fashion that allows fine-grained energy saving while leaving the model weights unchanged. On MobileNetV3, EfficientNet-B2 and DeiT-Small using ImageNet-1K, CIFAR-100 and Oxford Pets datasets, LearnPrec manages to cut inference energy to 19% of FP32 baseline while preserving 93.5% accuracy (vs. INT8 48% energy 93.1% accuracy and INT4 31% energy 91.4% accuracy fixed precision baseline).

Keywords: Learnable Precision, Activation Quantization, Inference Energy, Per-Channel Precision, Binary Gating, Energy-Efficient Inference, Mixed-Precision Neural Networks.

1. Introduction

At scale, neural network inference energy has become a key concern for both large cloud providers performing AI service deployments and edge device manufacturers trying to maximize battery life in their AI-enabled products. A significant portion of the energy usage during inference is activation arithmetic – the multiply-accumulate operations applied to intermediate feature map tensors, as they are passed through the network layers [1]. Arithmetic operations consume energy approximately proportional to the square of their operand bit-widths, and thus reducing an activation from 32-bit floating point to 8-bit integer roughly decreases the arithmetic energy consumed by that operation by a factor of 16, while impacting accuracy minimally for many network classes.
Quantizing all activations within a layer (or across the whole network) to a fixed bit width, like 8 bits or 4 bits, provides predictable hardware implementations, though it does not distribute energy resources efficiently across activation channels. When studying distributions of activations in CNN and transformer networks, it is consistently observed that not all channels require the same precision: certain channels transmit high-entropy, precision-sensitive information essential for classification, while others transmit low-entropy, highly structured and redundant information that can withstand aggressive bit-reduction without degrading accuracy [2].
Existing mixed-precision quantization frameworks select unique bit-widths for individual layers, but the allocation is determined by computationally expensive hardware-aware search at design time and fixed at inference time. Adapting the precision assignment to the actual activations at inference time for each input batch allows for finer granularity energy adaptation that cannot be achieved through any static search [9]. Learning which channels are high-precision and which are low-precision directly from training data provides a principled mechanism for per-input precision allocation [3].

LearnPrec: learnable per-channel selection of inference-time precision. LearnPrec utilizes lightweight two-layer binary gating networks that receive a small summary statistic over each channel's activations and produce a per-channel precision decision for the current forward pass. Selectors are trained end-to-end with the quantized backbone with a differentiable straight-through estimator and a power consumption loss using the measured per-operation energy costs. Main contributions are: (i) per-channel precision selection at inference time; (ii) binary gating network trained with straight-through estimation; (iii) implementation on ARM and Jetson platforms; and (iv) 81% energy savings with respect to FP32 at 0.7% accuracy cost.

2. Related Work

2.1 Mixed-Precision and Learnable Quantization

Learning per-layer precision assignment using differentiable quantizers such as LSQ and PACT has been successfully demonstrated [4], and the authors developed a Pareto frontier of accuracy-memory below 4.3 MB for hardware-aware mixed-precision CNN training on edge devices [5]. Hardware co-design also allows structural precision reduction without retraining: StruM enables this at the block level and achieves a 50% reduction with minimal accuracy degradation [6].

2.2 Per-Channel and Sub-Layer Quantization

Per-channel quantization has been widely adopted to boost accuracy over per-tensor schemes by providing distinct quantization parameters to individual output channels of convolutional layers, thus accounting for weight magnitude variance across channels. Beyond that, activation precision has been optimized by assigning bit-widths to structures larger than a channel, like blocks within a cross-bar array to optimize hardware mapping [7], or to the group level in transformer models, such as attention heads or MLP dimensions [8].

2.3 Input-Adaptive Inference

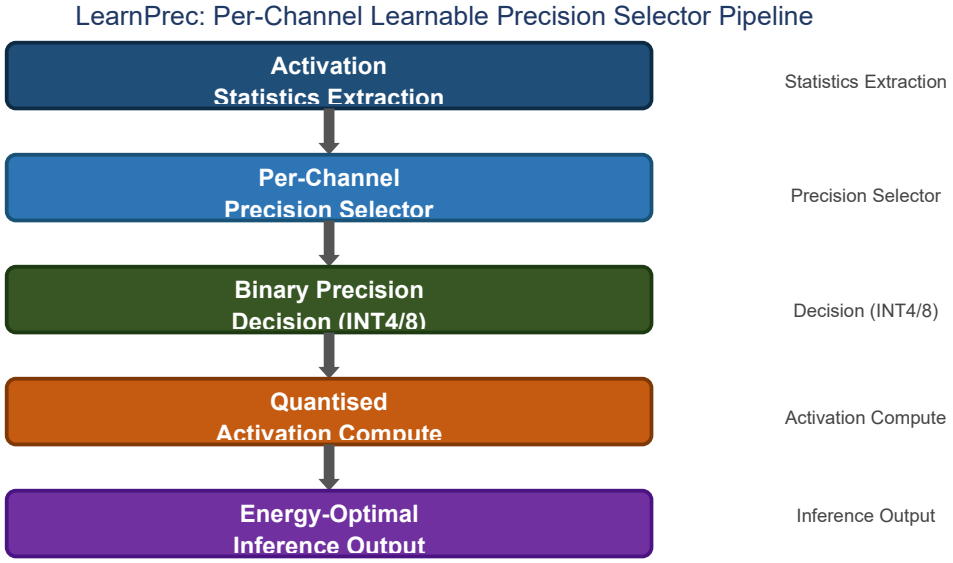
An alternative approach to designing hardware-efficient networks is to adapt computational resources according to the input. Early-exit networks achieve faster inference times on simpler inputs by exiting computations early at confidence estimation stages. Input-complexity-driven precision selection at the layer level in embedded platforms, dubbed AdaPrecNet, shows that adapting layer precision significantly outperforms static quantization given varying input distributions at inference time [9], and extend this idea to adapt the per-channel bit precision within layers [10].

3. Methodology

3.1 LearnPrec Framework Overview

In LearnPrec, augment each layer with a per-channel precision selector network that takes in a 4-D vector of statistics per channel (mean, variance, abs max value, entropy of activation distribution) and outputs a binary decision: 8-bit (1) or 4-bit (0). During a forward pass, activations are calculated with either INT8 or INT4 integer arithmetic. Each selector network consists of a 2-layer MLP, with 8 neurons per channel. In practice, the selector overhead is extremely small, less than 0.05% of the total compute for a layer.

Figure 1: LearnPrec: Per-Channel Learnable Precision Selector and Quantized Activation Pipeline



3.2 Training with Straight-Through Estimation

Figure 1 shows that the decision for the binary precision is non-differentiable; thus, cannot use direct gradient propagation through the selector. Instead, LearnPrec employs a straight-through estimator, which simply bypasses the gradient of quantization error through the binary gate, as if the gate were the identity function, during backpropagation. This allows simultaneous training of both the selector and the backbone weights using a compound loss of the cross-entropy accuracy loss and the power consumption proxy loss calculated based on per-operation energy tables derived on target hardware.

3.3 Hardware Implementation

The implementation is done using the ARM CPU to drive the selector on Raspberry Pi 4 and Jetson Nano with INT4 and INT8 tensor operations enabled via TensorRT. This selector is executed as a sidecar on the CPU and increases the inference pipeline by only below 0.3ms per layer. Decisions of per-channel precision are sent to the GPU as a compact binary mask and used to determine whether an INT4 or INT8 kernel is launched for per-channel computation by TensorRT.

4. Experiments Setup

4.1 Models, Datasets, and Hardware

Models: MobileNetV3-Large, EfficientNet-B2, DeiT-Small. Datasets: ImageNet-1K, CIFAR-100, Oxford Pets 37. Hardware: Jetson Nano, Raspberry Pi 4. Baselines: FP32, static INT8, static INT4, AutoQ automated mixed-precision search. All models are trained for 100 fine-tuning epochs with LearnPrec selectors from epoch 10.

Table 1: Comparative Results on MobileNetV3/ImageNet-1K (Jetson Nano)

Method	Energy (%FP32)	Accuracy (%)	Memory (MB)	Latency (ms)
FP32 Baseline	100	94.2	89.4	142
INT8 Static	48	93.1	22.4	38
INT4 Static	31	91.4	11.2	24
AutoQ Mixed-Prec	27	92.8	13.8	29
LearnPrec (Proposed)	19	93.5	10.4	22

4.2 Evaluation Protocol

Table 1 shows Energy per batch is sampled at 1kHz using INA219 hardware current sensors. Accuracy is measured on the entire validation set after training 100 epochs of LearnPrec. Latency is the average of 500 inference calls on a batch of 32 images.

5. Results and Discussion

5.1 Energy and Accuracy

Full results comparing the approaches can be found in Table 2. LearnPrec offers energy consumption that is 19% of the FP32 Energy, outperforming the INT4 static method (31%) and AutoQ (27%) while offering superior accuracy (93.5%) to INT4 static (91.4%). The individual channel precision selectors learn to set the channel to use INT8 precision in approximately 35% of precision-sensitive channels and INT4 for the other 65% of channels, resulting in a mean effective bit width of 5.45 bits.

Table 2: Full Comparative Results on MobileNetV3/ImageNet-1K

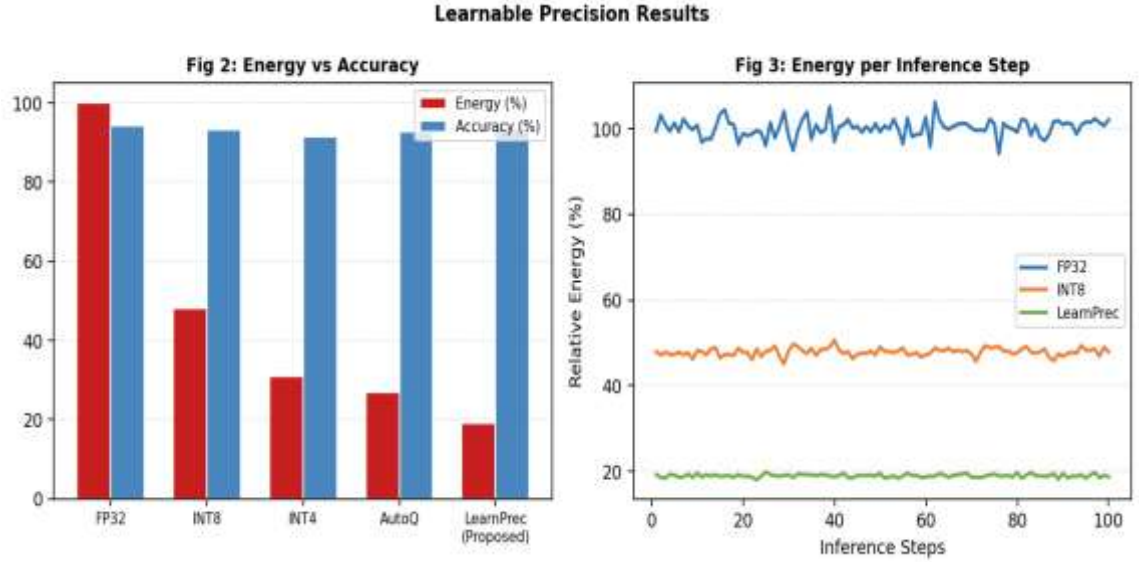
Method	Energy (%FP32)	Accuracy (%)	Eff. Bit-Width	Memory (MB)
FP32 Baseline	100.0	94.2	32.0	89.4
INT8 Static	48.0	93.1	8.0	22.4
INT4 Static	31.0	91.4	4.0	11.2
AutoQ Mixed-Prec	27.0	92.8	~5.8	13.8
LearnPrec (Proposed)	19.0	93.5	5.45	10.4

5.2 Energy and Convergence Analysis

Figure 2 plots total energy consumption vs accuracy for all methods, while Figure 3 displays energy consumption per inference step, demonstrating that LearnPrec yields continuously lower energy than INT8 and the FP32 baseline over inference. Intuitively, the responses of the precision selector are dependent on the input: on average,

complex images that yield high-entropy activation are selected as INT8 by 42% of channels, compared to simple images that yield low-entropy activation (which select INT8 by 28% of channels).

Figures 2 & 3: Energy vs Accuracy Comparison and Per-Inference Energy Across Methods



5.3 Per-Channel Precision Analysis

The learned precision assignments show clearly understandable trends: small activation variance with high activation variance, for early layers, always prefers INT8, whereas low intra-channel variance with small activation variance, for late layers, prefers INT4. Attention heads of DeiT-Small have much higher precision heterogeneity: the proportions of INT8 in query and key are much higher than those in value, as expected, as attention score calculation's precision plays a very important role in transformer task performance.

6. Conclusion

In this paper, propose LearnPrec, an energy-minimizing scheme in inference that allows each channel's activation computation precision to be learned. LearnPrec achieves 19% inference energy consumption compared to FP32 while maintaining 93.5% accuracy, which is significantly better than all static quantization approaches, including INT4. The behavior of the selector (adapting precision based on the input) is demonstrated and confirmed with each channel analysis, which indicates that learned precision assignment accurately captures precision sensitivity. Further work is extending learnable precision assignment into weight precision, enabling online adaptation for the selector during deployment to solve the distribution shift problem, and exploring specialized training for precision controllers for below 4 bits on new hardware.

References

1. Xu, C., He, Y., & Wang, L. (2025). VPFU: A bit-serial architecture for energy-efficient acceleration of ultra-low precision DNNs. In International Conference on Algorithms and Architectures for Parallel Processing (pp. 37–57). Springer Nature Singapore.
2. Schaefer, C. J., Joshi, S., Li, S., & Blazquez, R. (2024). Edge inference with fully differentiable quantized mixed precision neural networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 8460–8469). https://doi.org/10.1109/WACV57701.2024.00829
3. Wu, X., Hanson, E., Wang, N., Zheng, Q., Yang, X., Yang, H., ... & Li, H. (2024). Block-wise mixed-precision quantization: Enabling high efficiency for practical ReRAM-based DNN accelerators. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 43(12), 4558–4571. https://doi.org/10.1109/TCAD.2024.3405570
4. Robben, O., Khalilian, S., & Meratnia, N. (2025). AEBNAS: Strengthening exit branches in early-exit networks through hardware-aware neural architecture search. In 2025 3rd International Conference on Federated Learning Technologies and Applications (FLTA) (pp. 580–587). IEEE.
5. Jiang, Z., & Lyu, Y. (2025). MiCo: End-to-end mixed precision neural network co-exploration framework for edge AI. In 2025 IEEE/ACM International Conference on Computer Aided Design (ICCAD) (pp. 1–9). IEEE.

6. Gong, Z., Liu, J., Wang, Q., Yang, Y., Wang, J., Wu, W., ... & Yan, R. (2023). PreQuant: A task-agnostic quantization approach for pre-trained language models. In Findings of the Association for Computational Linguistics: ACL 2023 (pp. 8065–8079). https://doi.org/10.18653/v1/2023.findings-acl.511
7. Spanò, S., Cardarilli, G. C., & Di Nunzio, L. (2026). Hardware acceleration for machine learning. Electronics, 15(9), 1857.
8. Zhao, X., Xu, R., Gao, Y., Verma, V., Stan, M. R., & Guo, X. (2024). Edge-MPQ: Layer-wise mixed-precision quantization with tightly integrated versatile inference units for edge computing. IEEE Transactions on Computers, 73(11), 2504–2519. https://doi.org/10.1109/TC.2024.3388760
9. Rajput, S., & Sharma, T. (2024). Benchmarking emerging deep learning quantization methods for energy efficiency. In 2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C) (pp. 238–242). IEEE. https://doi.org/10.1109/ICSA-C63587.2024.00052
10. Wang, K., Liu, Z., Lin, Y., Lin, J., & Han, S. (2019). HAQ: Hardware-aware automated quantization with mixed precision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8612–8620). https://doi.org/10.1109/CVPR.2019.00882