Modeling Structured vs Unstructured and Dual vs Single Sided Sparsity in DNNs

Team 12, Yicheng Huang and Nate Mustafa

Motivation:

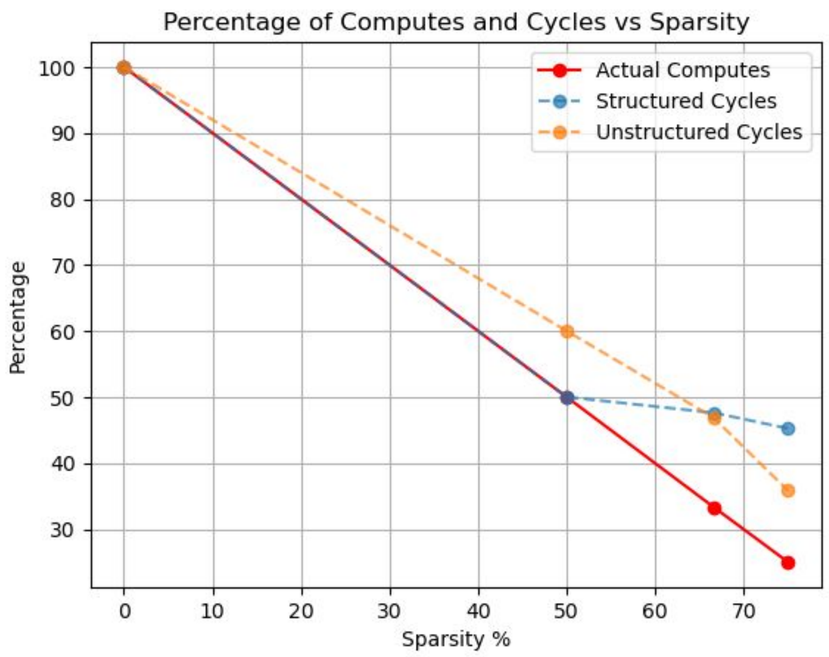
- Many approaches to exploit sparsity to save energy and reduce computation:

	Unstructured	Structured
Single-Sided	- Skip weights only - Random pattern - Flexible but higher overhead - Example: Deep Compression (Han et al.)	- Skip weights only - Structured pattern (e.g., 2:4) - Hardware-friendly, low overhead - Example: Nvidia STC
Dual-Sided	- Skip weights + activations - Random pattern in both - Maximum flexibility, high overhead - Example: Dual-Side Sparse Tensor Core	- Skip weights + activations - Regular pattern in both - High savings, more complex - Example: Structured DSTC extension

Goal: Identify which type yields best performance under various workflows

Structured v.s. Unstructured:

- Structured outperforms unstructured in energy across most cases.
- Structured ~2× energy savings vs. unstructured up to ~70% sparsity.
- Unstructured needs ≥75% sparsity to beat baseline (single-sided).
- Dual-sided unstructured requires ~40% sparsity to be effective.
- At extreme sparsity, both converge in performance.
- Both skip the same number of computations.
- Unstructured faster in cycles at sparsity ≥66%.

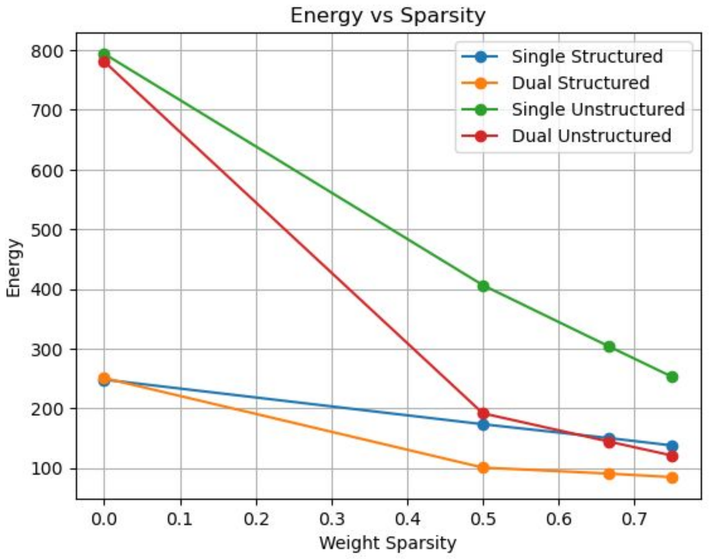


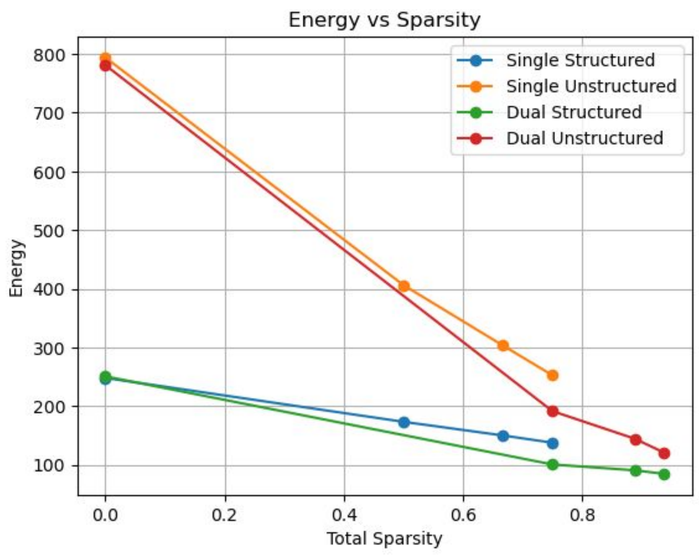
References:

1. Song Han et al., "Deep Compression," arXiv, 2016.
2. Yang Wang et al., "Dual-Side Sparse Tensor Core," arXiv, 2021.
3. Jeff Pool, "Accelerating Sparsity in the Nvidia Ampere Architecture," Nvidia, 2021.

Experimental Setup:

- Simulated ResNet 50 on:
  - DRAM, local DRAM buffer (4 cores), SRAM, MAC units.
  - 8-bit precision.
- Sparsities tested:
  - [0%, 50%, 66.66%, 75%] sparse weights and activations
- Measured energy, cycles, and computations



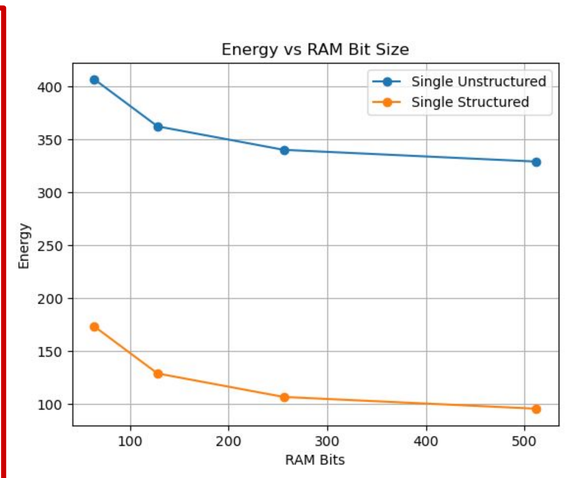


Dual-sided v.s. Single-Sided:

- Dual-sided generally outperforms single-sided.
- Dual-sided overhead dominates at low sparsity, especially for unstructured.
- After adjusting for combined input + weight sparsity, dual-sided still leads but advantage shrinks.

Memory Impact:

- Increasing RAM reduces energy for both structured and unstructured equally.
- Increasing only metadata RAM has no energy/cycle effect.
- Increasing register file meta-data storage capacity decreases energy usage by a nearly insignificant amount.



Limitations:

- Structured pruning (e.g., 2:4, 2:6) requires heavy preprocessing.
- Weight pruning is a one-time cost; input pruning incurs per-inference cost.
- Structured pruning can hurt accuracy.
- In practice, unstructured sparsity's flexibility may outweigh its hardware overhead.