KENNESAW STATE UNIVERSITY

IPCCC.ORG

Activation Sparsity Opportunities for Compressing General Large Language Models

Nobel Dhar, Bobin Deng, Md Romyull Islam,
Kazi Fahim Ahmad Nasif, Liang Zhao, Kun Suo

Presented by

Nobel Dhar

Outline

- Introduction
- Related Works
- Main Contributions
- Model Descriptions
- Weight & Activation Sparsity analysis
- Sparsity Enforcing
- Performance Evaluation and Trade-offs
- Sparsity Prediction Analysis
- Conclusion

Introduction

Edge Computing

- Performed at or near the data source
- Limited computational resources available

Compression Techniques

- Pruning, quantization, and knowledge distillation
- Compression possibilities

Activation Functions Determining Activation Sparsity

- ReLU vs Non-ReLU (SwiGLU, GeLU)
- 50% neurons can be dynamically pruned

Predict, Prefetch and Optimize

- Predict the sparsity and activation states
- Prefetch the neurons
- Optimize up to 50%

Related Works

- LLM Compression for Edge Systems:
  - Quantization
  - Pruning
  - Model distillation
- Activation Sparsity Utilization:
  - Intrinsic activation sparsity
  - Inactive neurons can be ignored
- ReLUfication:
  - Research are limited to ReLU-based Transformer architectures
  - Sparse ReLU-based activations into non-ReLU LLMs

Recent LLMs do not show intrinsic activation sparsity that can be utilized to optimize.

Our work aim is to fill this gap.

Main Contributions

- We explore the weight and activation sparsity of state-of-the-art LLMs:
  - Activation magnitude distributions
  - The importance levels of FFN neurons

- Based on activation magnitude distributions, we enforced activation sparsity to the state-of-the-art LLMs for the first time ever.

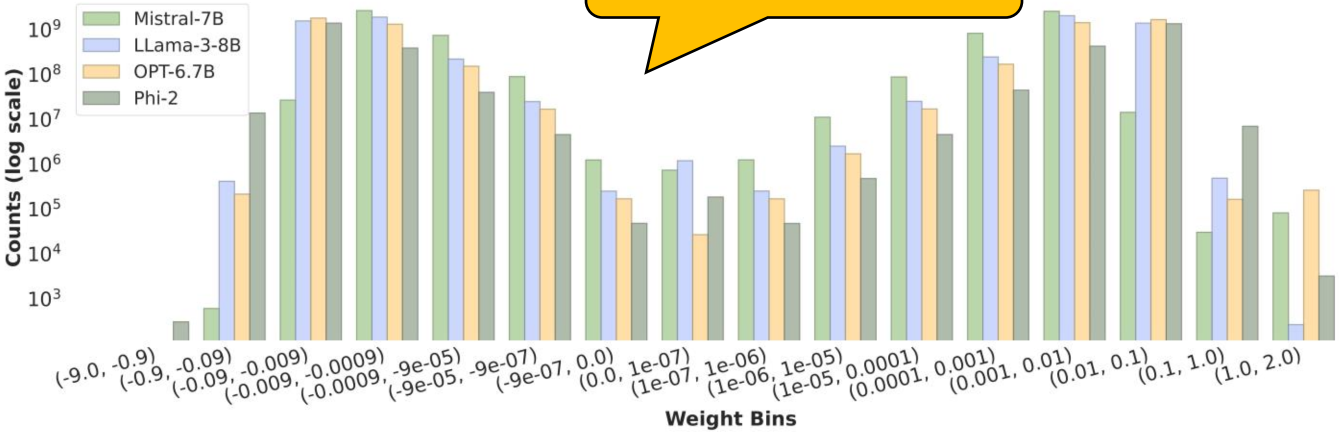
- To convert the extra activation sparsity to compression benefits, we further investigate the predictability of activation patterns.

Model Description

Model Name	Version	Activation Function	Number of Parameters
Llama	3	SwiGLU	8B
Mistral	0.1	SwiGLU	7B
Phi	2	NewGELU	2.7B
Phi	3-mini-128k	SwiGLU	3.8B
OPT	1	ReLU	6.7B

Let's Explore the Inherent Sparsity of the LLMs

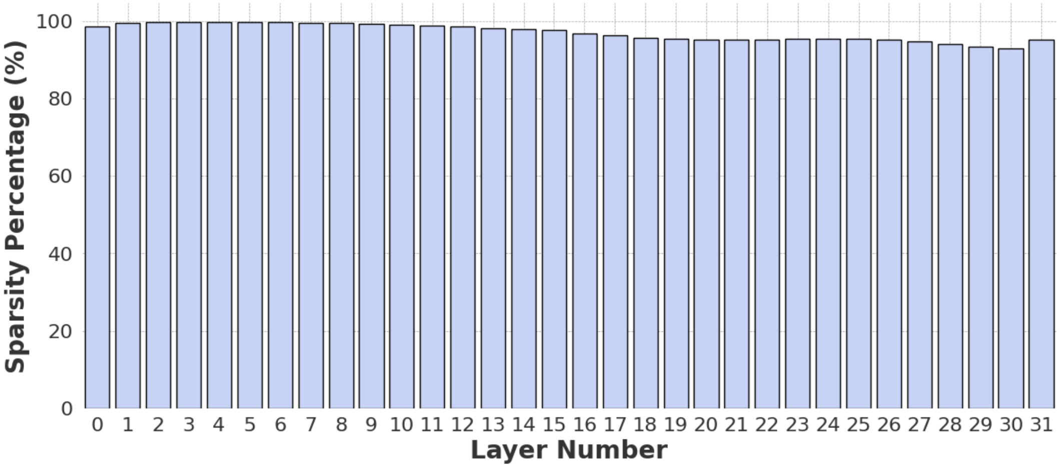
Weight Magnitude Distributions



We do not obverse substantial weight sparsity in four recent LLMs

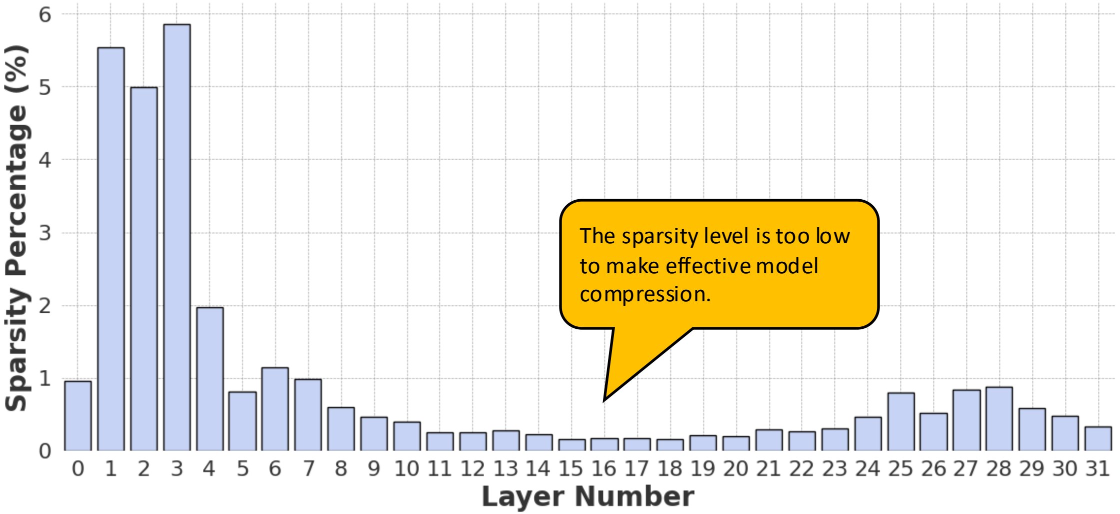
Insight: To compress LLM via the sparsity feature, we must explore another sparsity type: activation sparsity.

Activation Sparsity Analysis (OPT-6.7B)



Insight: The benefits of natural activation sparsity only exist in ReLU-based LLMs (e.g., OPT-6.7B).

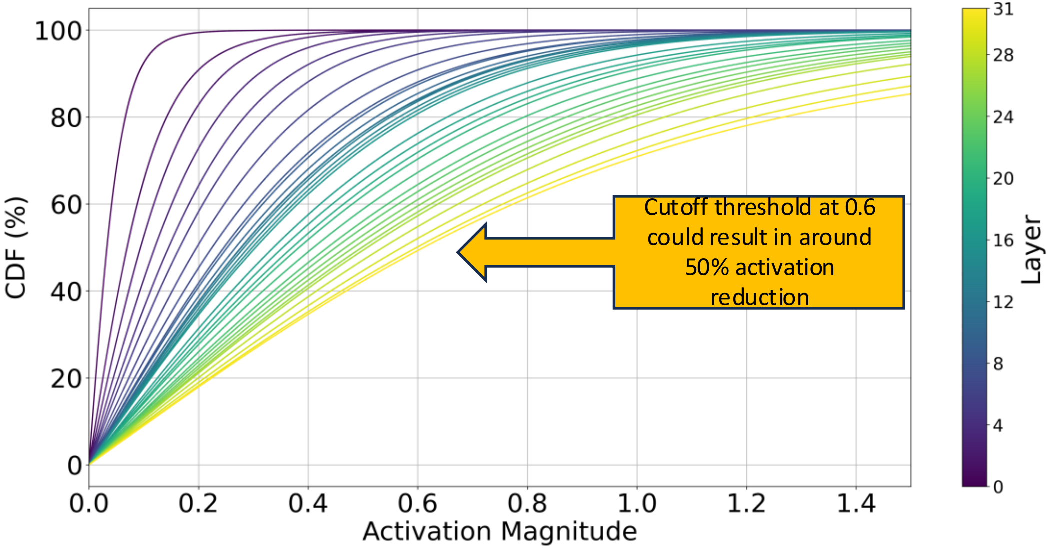
Activation Sparsity Analysis (Phi-2-2.7B)



The sparsity level is too low to make effective model compression.

Insight: Even though we can also observe natural activation sparsity in NewGELU-based LLMs (e.g., Phi-2-2.7B)
These were the only two models that showed natural activation sparsity.

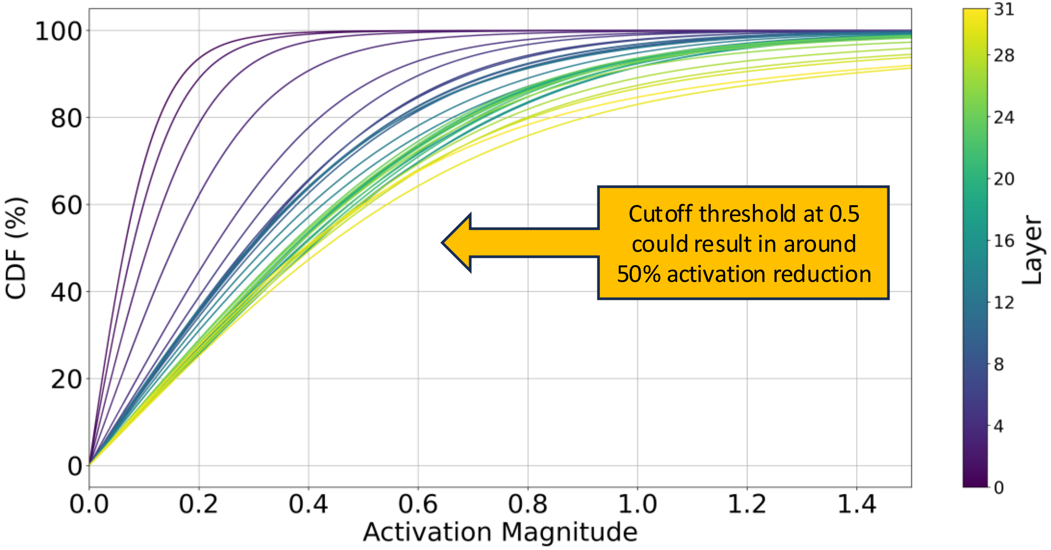
Activation Distribution Analysis (Phi-3-3.8B)



Cutoff threshold at 0.6 could result in around 50% activation reduction

Insight: The CDF suggests that setting a cutoff threshold has potential for extra activation sparsity.

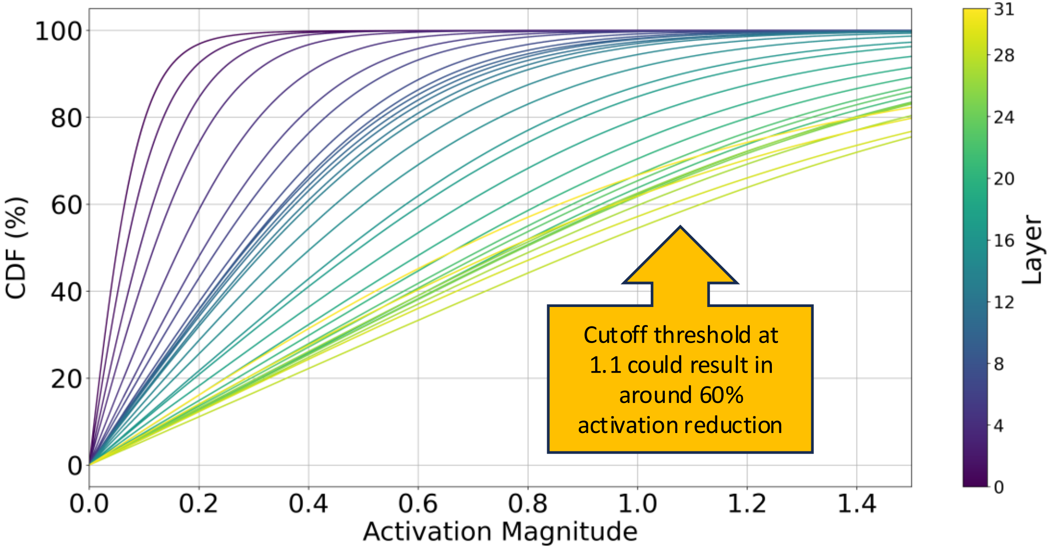
Activation Distribution Analysis (LLaMA-3-8B)



Cutoff threshold at 0.5 could result in around 50% activation reduction

Insight: The graph suggests that setting a threshold implies potential for 50% activation sparsity.

Activation Distribution Analysis (Mistral-3-7B)

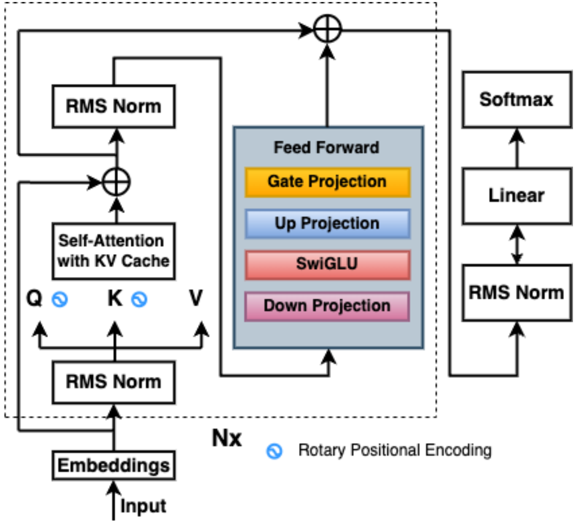


Cutoff threshold at 1.1 could result in around 60% activation reduction

Insight: The magnitudes of majority activation values fall into very small data ranges. This observation allows us to set small thresholds to omit less contributing weights and easily obtain high sparsity levels.

Let's Enforce Sparsity into State-of-the-art LLMs

Targeting Components of LLMs for Compression



Architecture of state-of-the-art decoder-only LLMs

Thresholds Determining Methodology

$$\mathbf{A}_{i,l} = f_l(\mathbf{x}_i)$$

$$\mathbf{A}_l = \{\mathbf{A}_{i,l} \mid i \in \{1, 2, \ldots, N\}\}$$

$$\mathbf{A}_l^{\text{flat}} = \text{flatten}(\mathbf{A}_l)$$

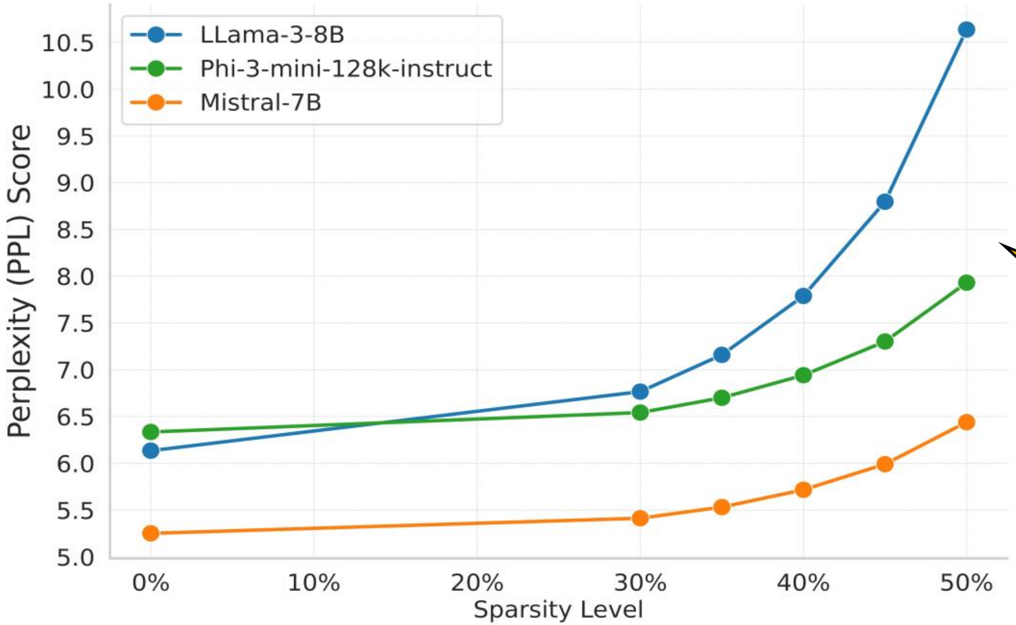
$$\mathbf{A}_l^{\text{abs}} = \left|\mathbf{A}_l^{\text{flat}}\right|$$

$$\mathbf{A}_l^{\text{sorted}} = \text{sort}\left(\mathbf{A}_l^{\text{abs}}\right)$$

$$T_{l,\alpha} = P_\alpha\left(\mathbf{A}_l^{\text{sorted}}\right)$$

$$\mathbf{A}_{i,l}^{\text{thresholded}} = \begin{cases} 0 & \text{if } |\mathbf{A}_{i,l}| < T_{l,\alpha} \\ \mathbf{A}_{i,l} & \text{otherwise} \end{cases}$$

Performance Evaluation & Trade-offs



We can obtain an extra 50% activation sparsity for the state-of-the-art LLMs by enforcing the threshold setting while maintaining acceptable PPL/accuracy.

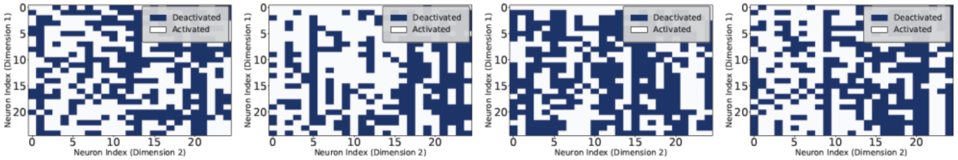
Let's Analyze the Predictability of LLM Activation Patterns

Activation Pattern Matching Rates for Similar Inputs

Sample	Similarity Percentages of Input Variants					
	95%	90%	85%	80%	75%	70%
1	100%	100%	100%	100%	100%	100%
2	57.73%	57.73%	57.73%	57.73%	57.73%	57.73%
3	100%	100%	100%	100%	100%	100%
4	100%	53.83%	53.83%	53.83%	53.93%	53.83%
5	100%	100%	100%	100%	100%	100%
6	100%	100%	100%	100%	100%	100%
7	100%	100%	100%	100%	100%	100%
8	57.73%	57.73%	57.73%	57.73%	57.73%	57.73%
9	100%	100%	100%	100%	100%	100%
10	100%	100%	100%	100%	100%	100%
11	100%	100%	100%	100%	100%	100%
12	100%	100%	100%	100%	100%	100%

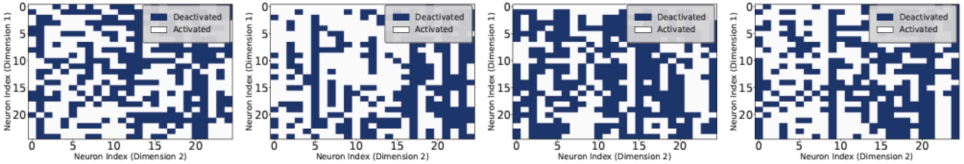
Insight: Most input samples have 100% same activation patterns for all layers with their 70% similarity variants.

Pattern Visualization & Comparison



(a) Layer 1 (b) Layer 10 (c) Layer 20 (d) Layer 32

Activation heatmap pattern of LLaMa-3-8b with default Sample 1 input



(a) Layer 1 (b) Layer 10 (c) Layer 20 (d) Layer 32

Activation heatmap pattern of LLaMa-3-8b with 70% similarity

Insight: The activation patterns are predictable and can effectively compress new large language models from the main memory's perspective.

Conclusion

- The insufficient computing and memory resources on edge devices
- Safely secure 50% extra sparsity in FFN layers with a negligible accuracy loss
- Compress the LLMs from memory’s perspective via prediction and prefetching
- A guidelines leading to less LLM execution latency, lower power cost, and improved user experience

Q
&
A