STAT 992: Science of Large Language Models

Lecture 6: Sparsity and low-rankness

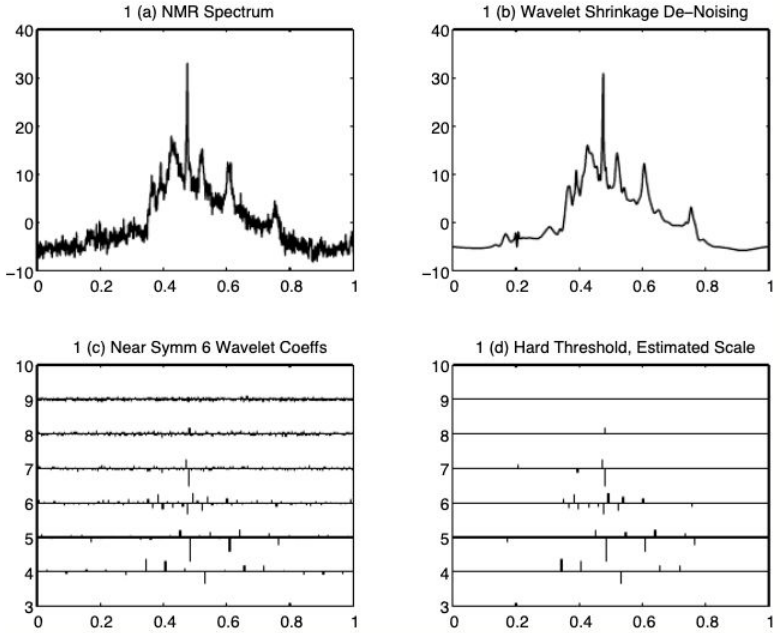
Spring 2026

Yiqiao Zhong

Sparsity and low-rankness in classical signal processing

- **Sparsity**: many entries are zero in a long vector or large matrix
- **Low-rank**: the rank of a large matrix is not large

- Both are natural mathematical properties
  - Image/video signals are *approximately sparse* under Fourier / wavelet basis
  - Equity return data, rating score matrix are *approximately low-rank*



Johnstone, Gaussian sequencing model

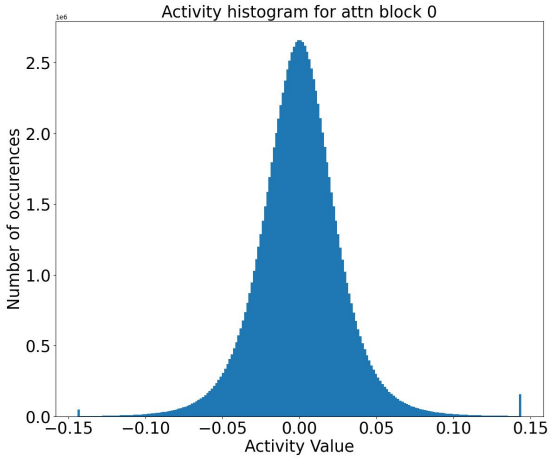
Overview of sparsity in LLMs

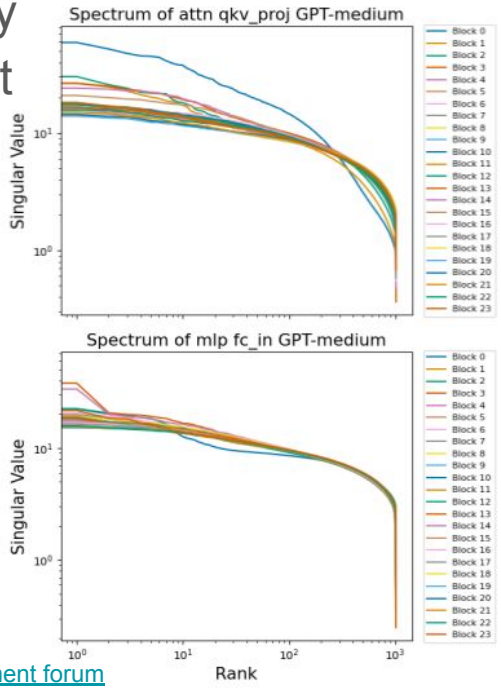
Sparsity Type	Category	Core Concept & Phenomenon
Weight	**Generally Dense**	Trained weights are usually dense matrices.
	Model Compression	**Model pruning**, "Lottery Ticket" sub-networks, reduce memory, examples: SparseGPT ~50% sparsity
Attention	**Attention Sinks & Specialized Heads**	**Attention Sinks:** Special token receives most attention. **Specialized attention heads:** focusing only on specific linguistic features (e.g., the next token or punctuation).
	Context Scaling	Full attention requires quadratic compute in context length. Context scaling: **Sliding Window Attention** (Mistral) or **Block-Sparse Attention** to 100k+ token contexts.
Activation	**Massive Activation**	**Dead neurons**: Many ReLU activations are zero in MLPs. **Outlier dimensions**: a few coordinates in hidden states have consistently large activations across different inputs.
	MoE (Mixture of Experts)	**Sparsely activated experts.** Use many MLPs (specialized experts) within one attention block, but only a few MLPs are activated. Increasing capacity while reducing FLOPs

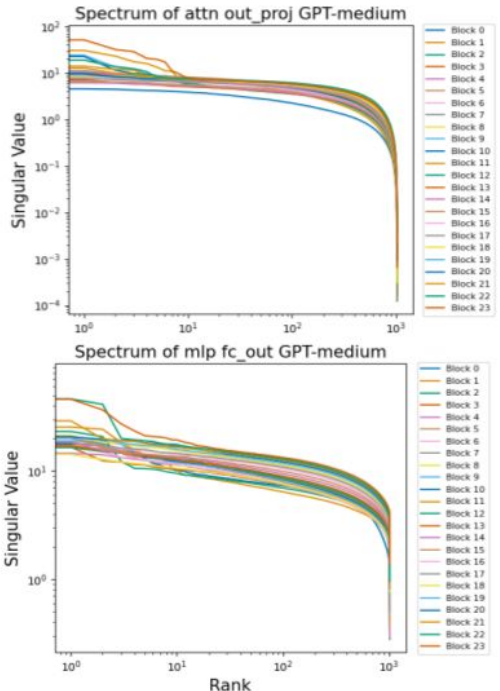
Naive phenomenon in blue, practical implementation in green. *Table edited by Gemini-3*

Model weights

- **Pretrained GPT-2**: weights are approximately Gaussian distributed
- Activations are approximately Gaussian with a few outliers
- Weight matrix spectra are mostly power law distributed; consistent with analysis of other DNNs







Soure: AI alignment forum

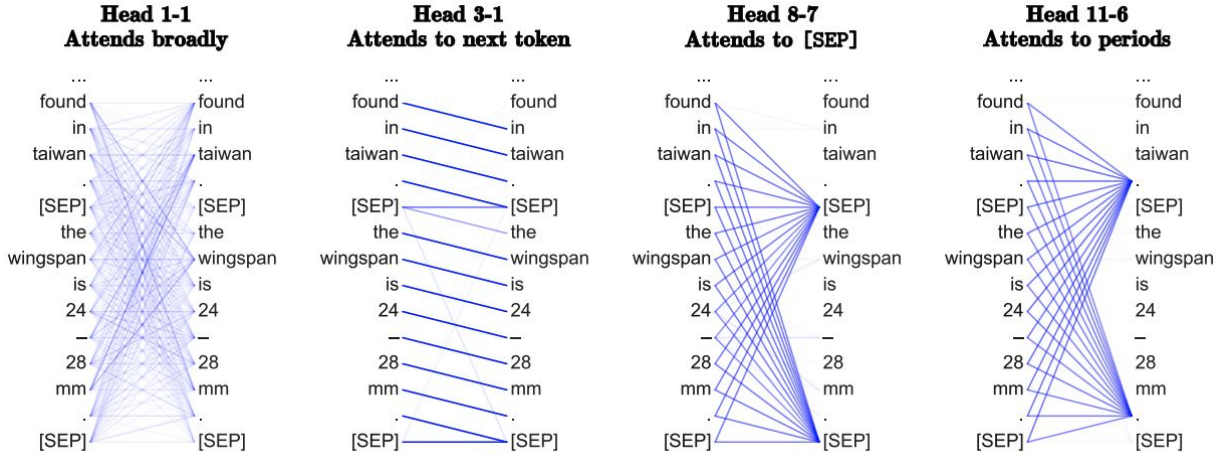
Sparse subnetworks and lottery ticket hypothesis

- **Lottery ticket hypothesis**: there exists a subnetwork in a randomly initialized network, when trained in isolation, results in no loss in accuracy.
  - Conceptual significance: optimization depends on winning lottery tickets, implication on parameter redundancy
  - Iterative pruning finds the winning subnetwork

- **Model pruning**: limited success with dropping (zeroing) weights and dropping attention heads.
  - GPU efficiency: bad for sparse matrix computation
  - Robustness issues
  - Quantization is currently better (reducing floating number precision)

Specialized attention heads

- Some attention heads have a clear functionality
- Relative-position-based: previous-token head, or next-token head (below)
- Special-token-based: attending to delimiters (below)
- More types: previous-k-token head, induction heads, reverse induction heads, etc.
- Caveat: a lot of attention heads have mixed functionalities (superposition)



What Does BERT Look At? An Analysis of BERT's Attention, 2019

Note: unlike BERT, GPT-type transformers have causal masks—attention can only go to previous tokens

Attention sinks

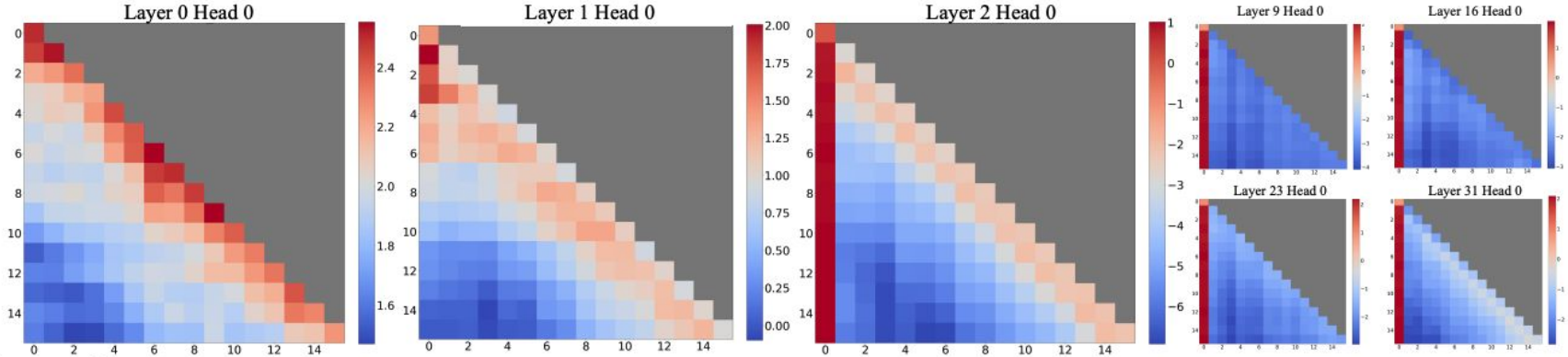


Figure 2: Visualization of the *average* attention logits in Llama-2-7B over 256 sentences, each with a length of 16. Observations include: (1) The attention maps in the first two layers (layers 0 and 1) exhibit the "local" pattern, with recent tokens receiving more attention. (2) Beyond the bottom two layers, the model heavily attends to the initial token across all layers and heads.

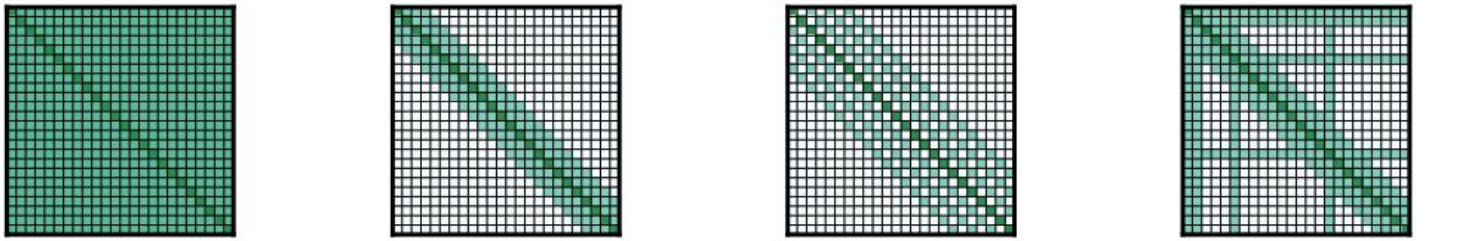
Efficient Streaming Language Models With Attention Sinks, 2023

- Why: softmax defaults to the first (sink) token if attention is not needed:

$$\text{SoftMax}(x)_i = \frac{e^{x_i}}{e^{x_1} + \sum_{j=2}^{N} e^{x_j}}, \quad x_1 \gg x_j, j \in 2, \ldots, N$$

Long context and structured attention

- Attention calculation requires compute quadratic in context length
- Practical fix: design structured attention utilizing sparsity



(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Structured attention utilizing sparsity: Longformer: The Long-Document Transformer 2020

Massive activation

- Outlier features (certain coordinates of hidden states) have large values
- It is likely a training artifact

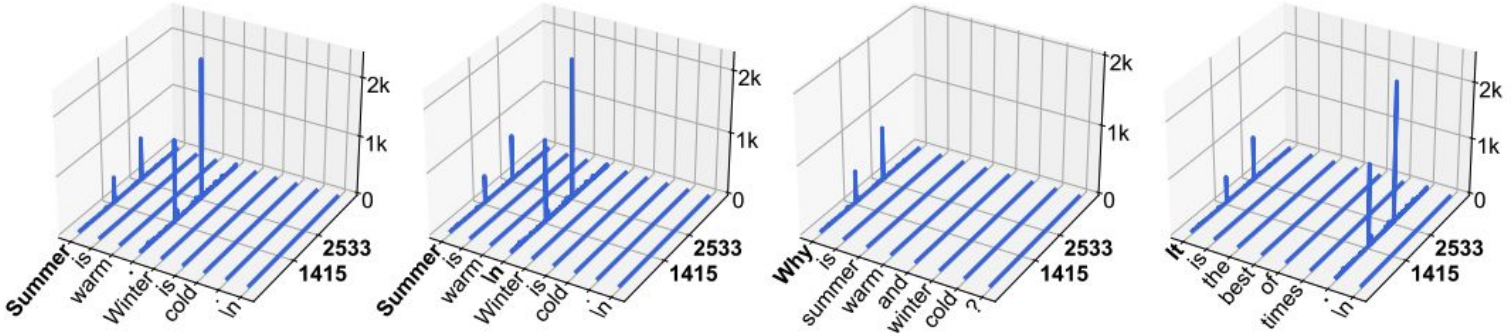


Figure 1: **Activation Magnitudes (z-axis) in LLaMA2-7B.** **x** and **y** axes are sequence and feature dimensions. For this specific model, we observe that activations with massive magnitudes appear in two fixed feature dimensions (1415, 2533), and two types of tokens—the starting token, and the first period (.) or newline token (\n).

Massive Activations in Large Language Models 2024

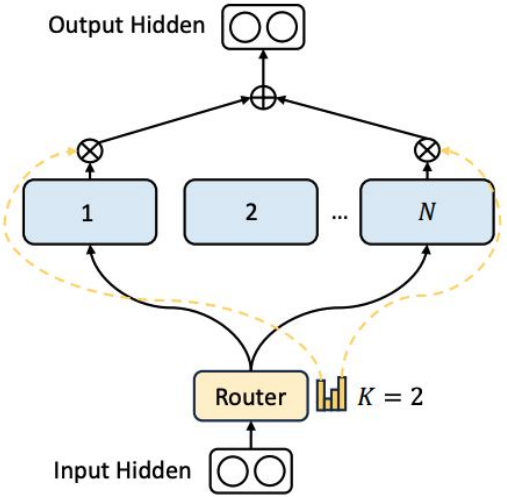
Mixture-of-experts (MoEs)

- Many parallel subnetworks, but only a few are activated for a given input
- Interpretation: a large panel of specializing experts, but a few are needs for a given question
- Large parameter count while low FLOPs

$$y = \sum_{i-1}^{n} G(x)_i E_i(x)$$

$$G_\sigma(x) = Softmax(x \cdot W_g)$$

Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer, 2019



DeepSeekMoE, 2019

Low-rankness

- From Lecture 5, the task-relevant concept vectors may span a low-dim subspace

$$\text{“apple”} = 0.09\ \text{“dessert”} + 0.11\ \text{“organism”} + 0.16\ \text{“fruit”} + 0.22\ \text{“mobile\&IT”} + 0.42\ \text{“others”}.$$

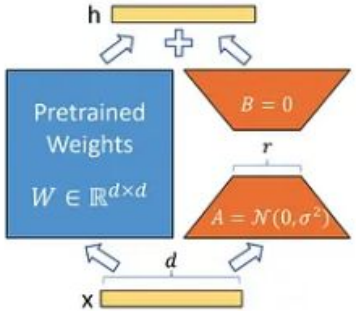
$$\boldsymbol{x} = \sum_{j=1}^{K} a_j \boldsymbol{\varphi}_j$$

LoRA as efficient finetuning

- Full model finetuning is too expensive (too many parameters)

$$W = W_0 + BA$$

- Motivation of LoRA: use gradient descent on much fewer parameters



LoRA: Low-Rank Adaptation of Large Language Models, 2019