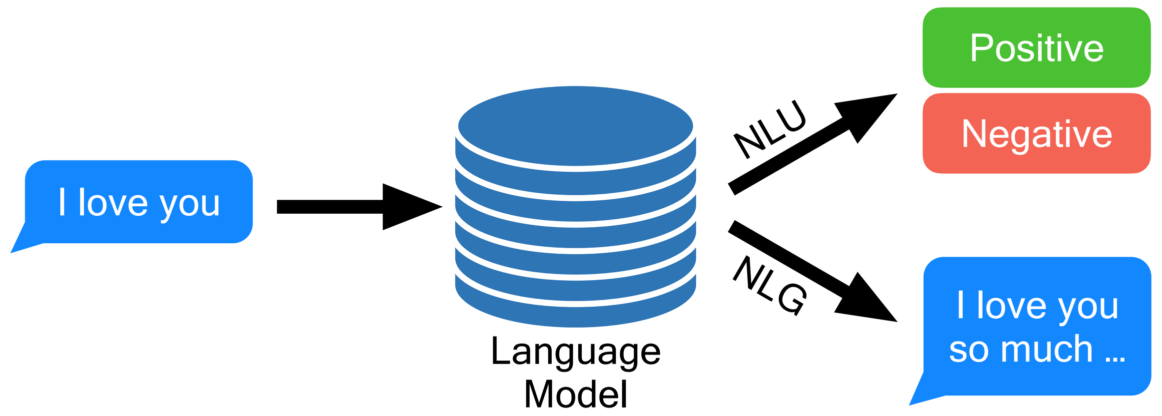
LoSparse: Structured Compression of Large Language Models based on Low-Rank and Sparse Approximation

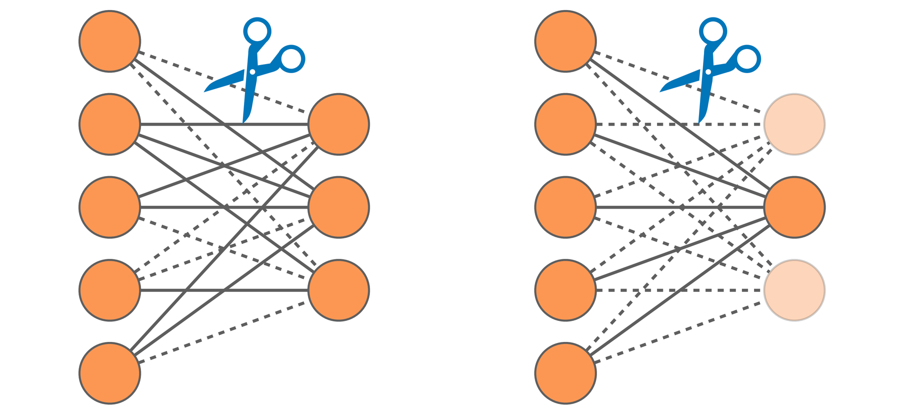
Yixiao Li*, Yifan Yu*, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, Tuo Zhao

Background

- **Large language models**: **powerful** and **successful** in NLU and NLG, but **consume huge memory** and **difficult to deploy on edge devices**.



- **Size**: T5 up to **11B**; GPT-3 up to **175B**.
- **Goal**: Reduce the size while maintain its power.
- **Previous approach**: Pruning.

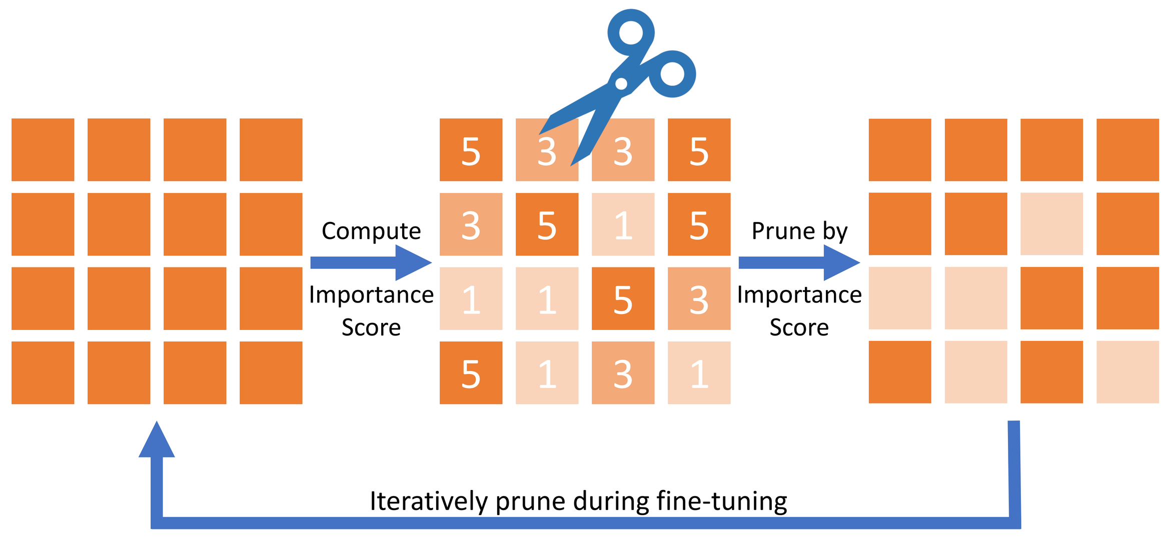


Unstructured pruning   Structured pruning

Question: Is pruning the optimal solution for reducing model sizes?

Pruning

Unstructured Iterative Pruning



- **Importance Score**: Given a weight w_{ij} and its gradient from the loss $\mathcal{L}$, the importance score of w_{ij} is its sensitivity [1], defined as

$$I(w_{ij}) = |w_{ij} \cdot \nabla_{w_{ij}}\mathcal{L}|.$$

- **Iterative Pruning**: Pruning happens at the same time as fine-tuning.

Challenges

- **Difficulty in Storing Unstructured Matrices.** Storing unstructured matrices requires high sparsity (e.g., 99%), which often hurts the performance.

Pruning (Cont'd)

- **Unwanted Importance Distribution.** Ideally, a few weights have high importance scores so that the rest can be pruned. In practice, however, many weights are important.

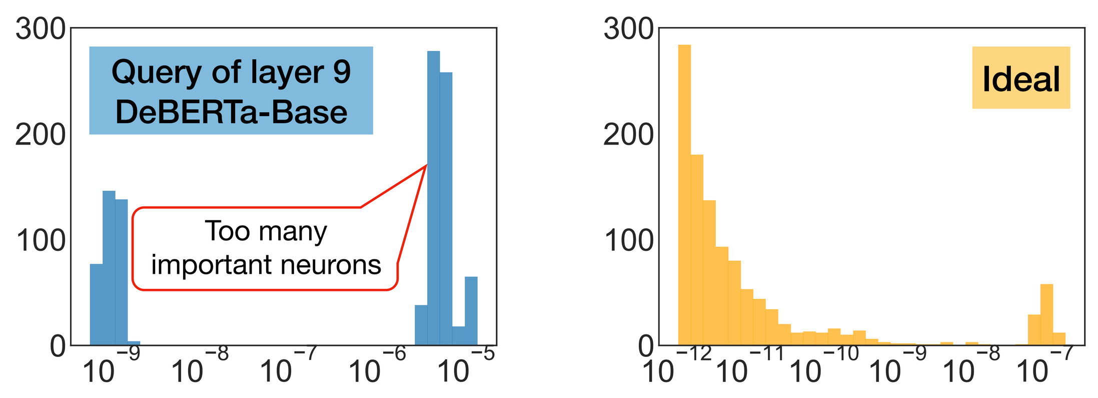


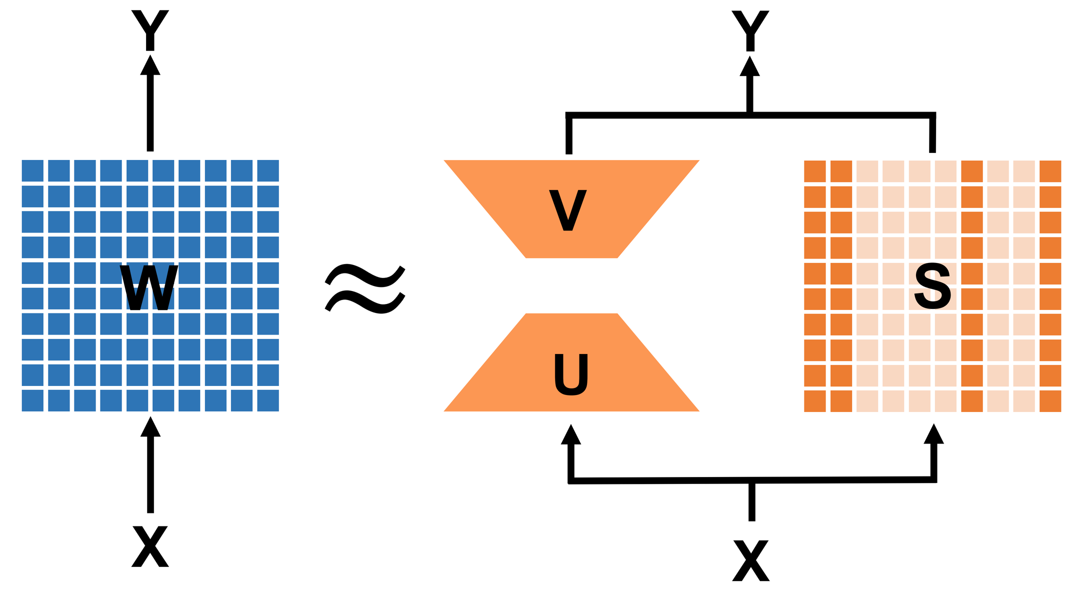
Figure 1: Histogram of neuron importance scores

Our Method

High-level idea: A dense weight matrix W is approximated by a **low-rank matrix** and a structured **sparse matrix**:

$$W = UV^\top + S.$$

- $U, V \in \mathbb{R}^{d\times r}$, $r \ll d$. A typical d is 768 and 1024; a typical r is 8, 16, and 32.
- Columns in $S \in \mathbb{R}^{d\times d}$ are zeroed out. A typical remaining ratio ranges from 10% to 50%.



Why Low-rank Matrices?

Decoupling coherent and incoherent parts of neurons.

- Coherent parts: common knowledge, brewed by low-rank approximation. [2]
- Incoherent parts: neuron-specific information, learned by sparse approximation.

Large Singular Values. A few large singular values in language models, indicating the approximation error is small with low-rank approximation.

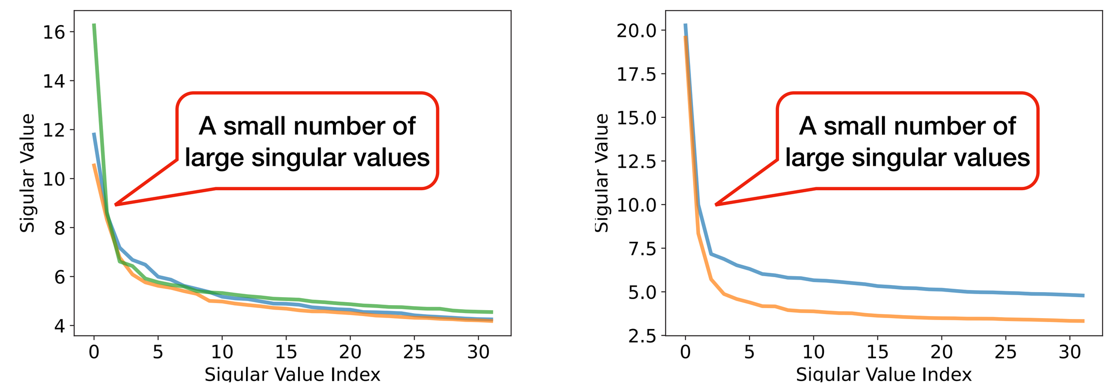


Figure 2: Singular value spectrum. Left: Layer 10 of BART-large; Right: Layer 14 of DeBERTaV3-large.

Why Low-rank Matrices? (Cont'd)

Importance Score Shift. Successfully shift the importance score distribution to the ideal one, helping achieve the high sparsity level without hurting performance drastically.

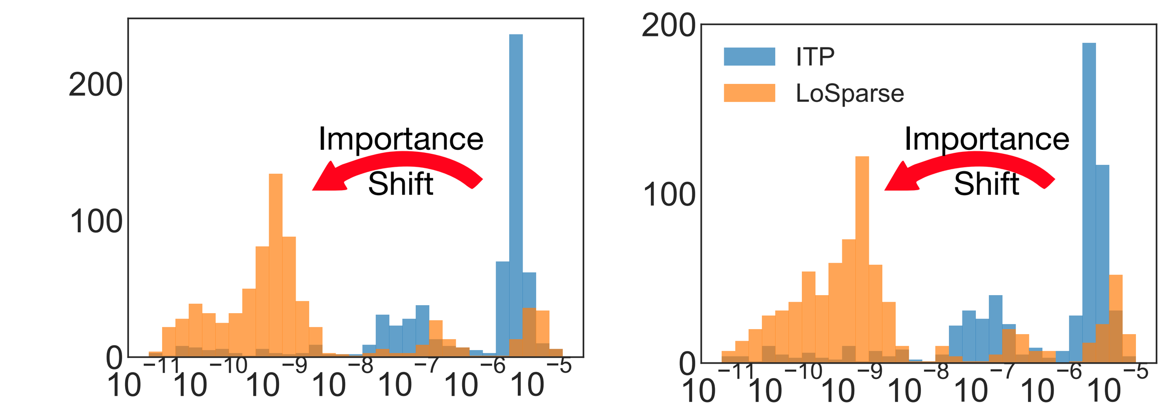


Figure 3: Histogram of neuron importance score. W_q (left) and W_k (right) of Layer 3 in DeBERTaV3-base.

Algorithm

SVD Initialization. Given a pre-trained weight W_0, obtain the initial low-rank part U_0 and V_0 by truncated SVD of W_0:

$$U_0 = [\sqrt{\sigma_1}u_1; \sqrt{\sigma_2}u_2; ...; \sqrt{\sigma_r}u_r],$$
$$V_0 = [\sqrt{\sigma_1}v_1; \sqrt{\sigma_2}v_2; ...; \sqrt{\sigma_r}v_r].$$

Obtain the initial structured sparse matrix S_0 by

$$S_0 = W_0 - U_0V_0^\top.$$

Iterative Pruning. We update U_t and V_t by SGD-type optimization at each iteration. A column s_t in a structured sparse matrix S_t at the next iteration is

$$s_{t+1} = \mathcal{T}(\widetilde{s}_t, I(s_t)),$$

where $\widetilde{s}_t = s_t - \alpha\nabla_{s_t}\mathcal{L}$ comes from the SGD-type optimation and

$$\mathcal{T}(\widetilde{s}_t, I(s_t))_{*i} = \begin{cases} \widetilde{s}_t & \text{if } I(\widetilde{s}_t) \text{ in top } p_t\%, \\ 0 & \text{o.w.} \end{cases}$$

The remaining ratio p_t is gradually decreased to the target sparsity as iteration goes on.

Main Results

Compressing BART-large on NLG tasks, summarization task XSum for example.

Ratio	Method	XSum
-	Lead-3	16.30/1.60/11.95
100%	$\text{BART}_{\text{large}}$	45.14/22.27/37.25
50%	ITP	38.42/16.32/31.43
	LoSparse	**39.18/16.91/31.62**
40%	ITP	36.71/14.96/29.86
	LoSparse	**38.30/16.02/30.72**
30%	ITP	34.42/13.15/27.99
	LoSparse	**37.41/15.42/30.02**

Main Results (Cont'd)

Compressing DeBERTaV3-base on NLU tasks, GLUE dataset for instance.

Ratio	10%		
Method	Movement	ITP	LoSparse
MNLI	N.A.	79.7	**81.7**
RTE	N.A.	N.A.	**66.0**
QNLI	N.A.	82.3	**86.1**
MRPC	77.0	78.5	**82.3**
QQP	N.A.	88.3	**89.5**
SST-2	88.0	88.3	**89.2**
CoLA	N.A.	38.0	**40.0**
STS-B	N.A.	86.3	**87.2**

N.A. means the method doesn't converge.

How about CNN?

No More Fast Decay. Due to small kernel sizes and channel numbers, convolution weights (reorganized) don't have fast singular value decay.

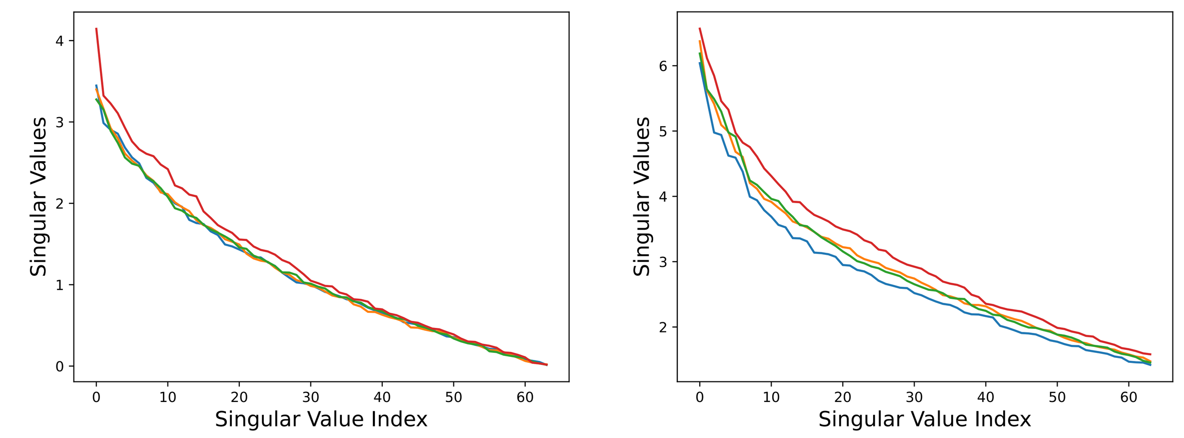


Figure 4: Singular value spectrum. Two convolution weights of ResNet-50, pre-trained on ImageNet-1k.

References

[1] Molchanov, P., Mallya, A., Tyree, S., Frosio, I., and Kautz, J. Importance estimation for neural network pruning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11264–11272, 2019.
[2] Candès, Emmanuel J., et al. "Robust principal component analysis?." Journal of the ACM (JACM) 58.3 (2011): 1-37.
[3] Yu, Xiyu, et al. "On compressing deep models by low rank and sparse decomposition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.