LEGO: Efficiently Generate Optimized LLMs through Low-Rank Decomposition and Assembly

Pingwei Sun

2023.12

BDT Program

HKUST

CONTENTS

PART 01. Survey on model compression

Survey

Predominant Strategies

- Quantization: fp32 -> fp16、int8、int4…
- UP: sparse computing, support from framework and hardware.
- SP: simplify the model's structure (layers, heads, hidden size).
- Knowledge Distillation: transfer knowledge to smaller LLMs.
- Low-rank Approximation: decompose a large matrix into the multiplication of two small ones.

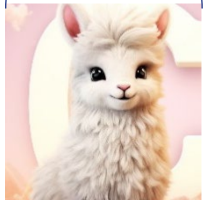
How to retain more knowledge in the weight matrix (pic): llama, "C", clouds, mountains

1024

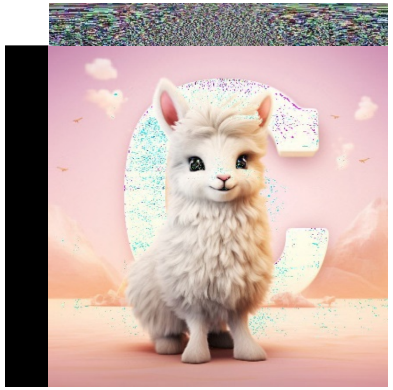
Quantization

Unstructured Pruning

512



Structured Pruning
&
Knowledge Distillation



Low-rank Approximation

Survey

Methods	Quantization (compatible with others)		Knowledge Distillation	Pruning		Low-rank Approximation
	PTQ	Data Aware Quantization	KD	Structured	Unstructured	Features are low-rank
Paper	GPTQ [1] (ICLR2023)	AWQ [2] (MIT, SJTU, THU)	MINILLM [3] (CoAI, THU)	LLMPruner [4] (NUS, Neurips2023)	SparseGPT [5] (ICML2023)	LoRD [6] (UdeM, Nolano AI)
Compress ratio	87.5% (fp32->int4)	90%+ (fp32->fp16/int3)	50%	20%	50~60%	20%
Speedup	3.2x (compared with fp16)	3x (compared with fp16)	~2x	NA	1.67x	NA
Performance	Little influence, can be ignored.	Nearly the same as fp16 on PPL	2~3 points drop of feedback from GPT4	10% drop on zero-shot	Little drop on some few-shot tasks	Less than 1% drop
Note	Widely used in industry	Work with the provided framework		Inexpertly obtain increase on some tasks		Reproducing NJU's work in AAAI 2023 on LLMs

[1] GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers

[2] AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration

[3] Knowledge Distillation of Large Language Models

[4] LLM-Pruner: On the Structural Pruning of Large Language Models

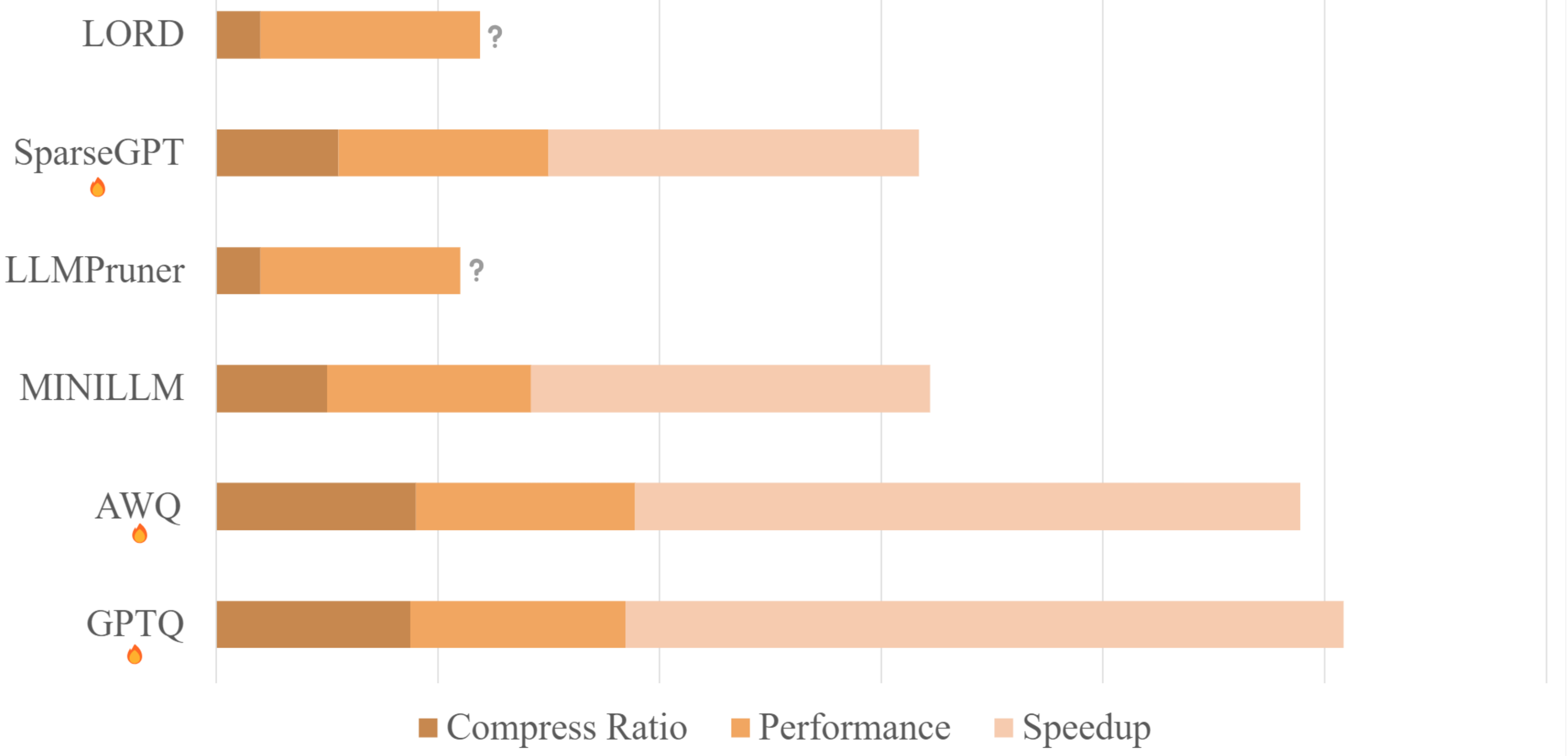
[5] SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot

[6] LORD: Low Rank Decomposition Of Monolingual Code LLMs For One-Shot Compression

Survey

Comparison

- A rough comparison of compression methods, with accurate compression ratios. The benchmarks across different methods are not entirely consistent. Some methods do not report speedup in their papers and stay closed source.



PART 02.

Preliminary study of low-rank features in LLMs

Low-rank in LLM

WHY

- Reduce the error produced during low-rank approximation
  - Decompose into matrices and singular values
  - Select first k singular values
  - Rebuild small matrices

- A SVD friendly matrix:
  - Most singular values are 0 or
  - Distribution of singular values is centralized

$$Y = XW + b \approx (XL_1)L_2 + b$$

$$W = USV^T$$

$$L_1 = (U\sqrt{S})_{[:,:k]} \quad L_2 = (\sqrt{S}V^T)_{[:k,:]}$$

Our experiments

- Weight matrix:
  - Taking more than half singular values to make up 90% of the sum of singular value.
  - Not SVD friendly

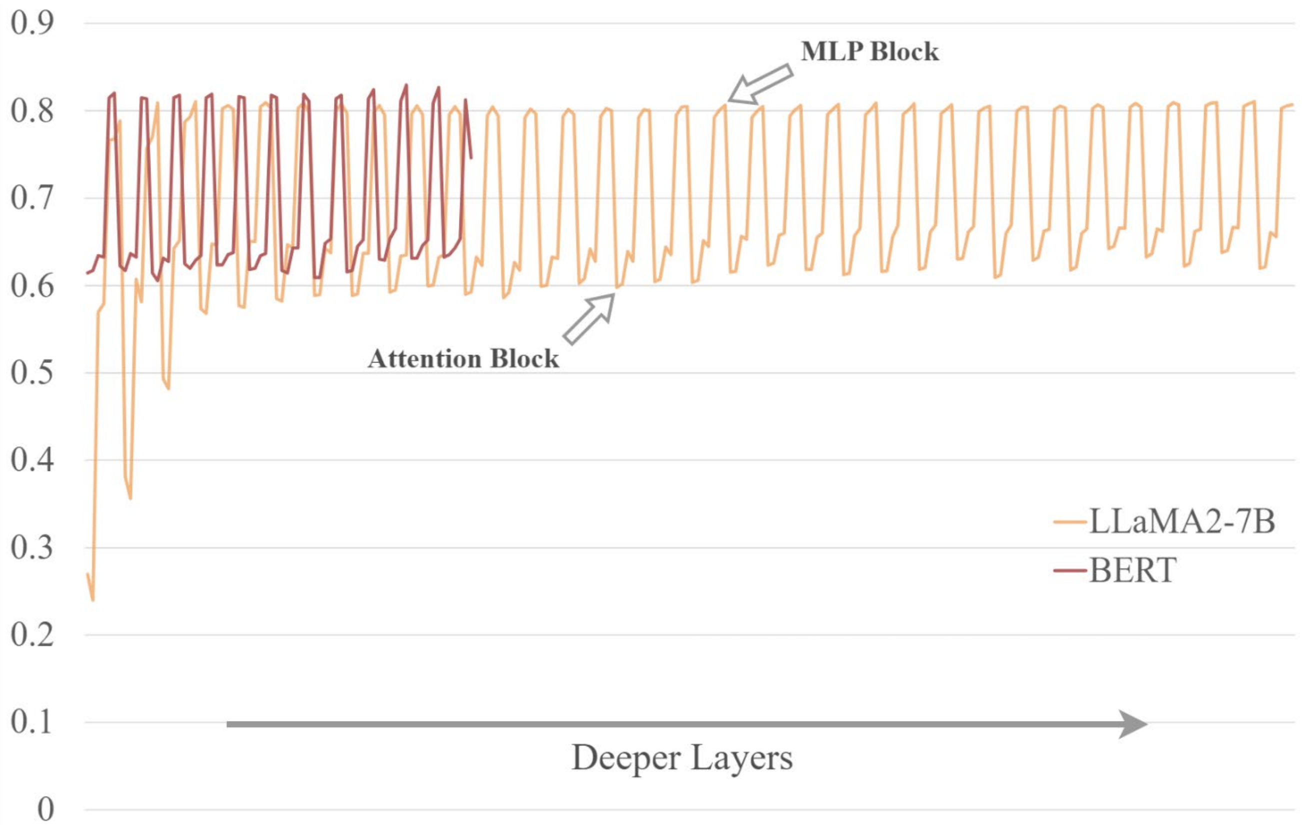
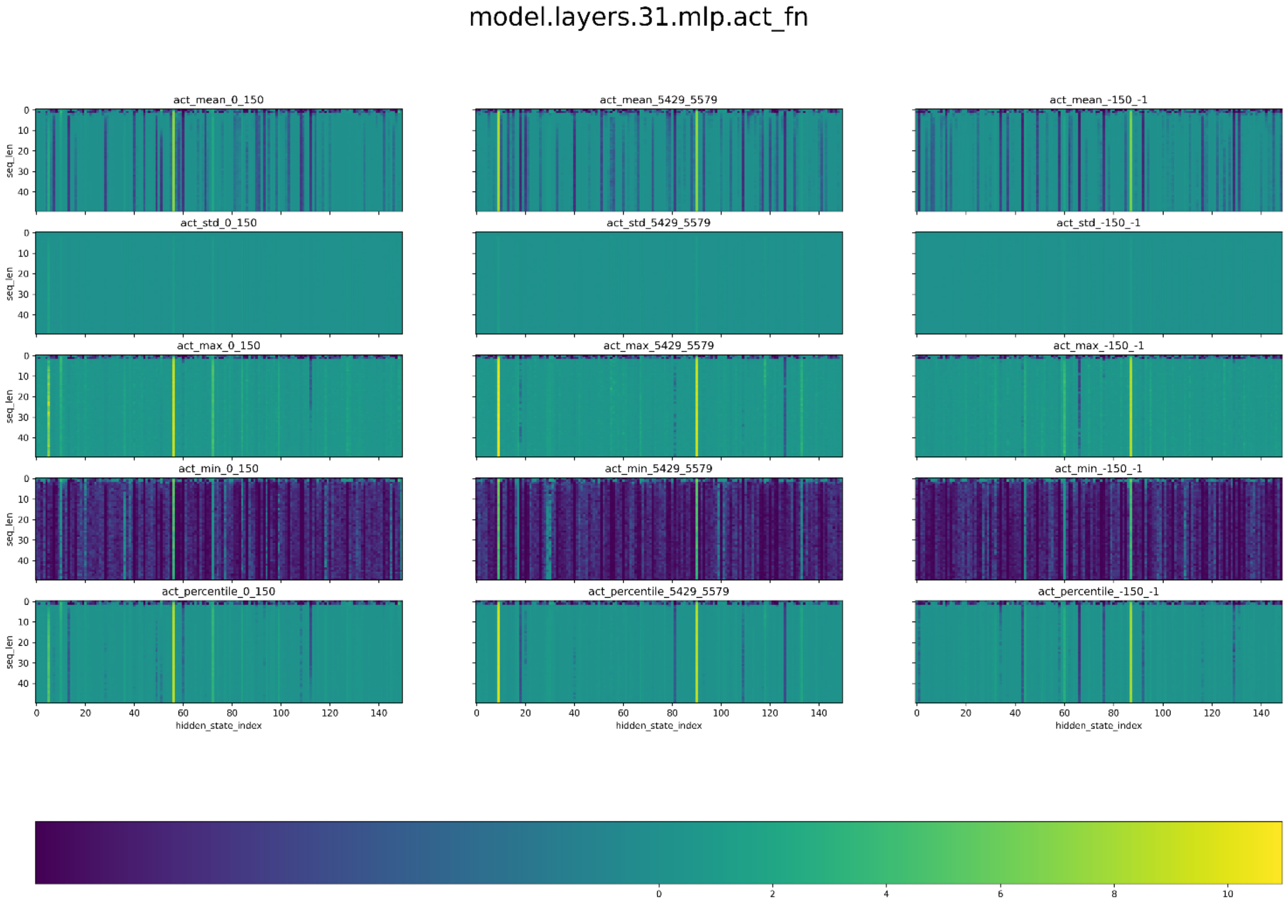


Figure 2: The percentage of singular values contributing to ninety percent of the total sum of singular values.

Low-rank in LLM

Our experiments

- Weight matrix:
  - Taking more than half singular values to make up 90% of the sum of singular value.
  - Not SVD friendly
- Output features:
  - Exists low-rank features
  - LoRD a recent work have used it on Code LLM



Domain Specific Parameters

- Updating of parameters is low-rank during finetuning
  - LoRA
  - QLoRA, LQ-LoRA (quantization + LoRA)

Domain Specific Parameters

- Updating of parameters is low-rank during finetuning
  - LoRA
  - QLoRA, LQ-LoRA (quantization + LoRA)

Core Region Parameters

- There exists core linguistic region
  - After finetuning on several languages some parameters remain stable.
  - Model's performance is sensitive to them.
  - Their distribution lies in rows or columns.

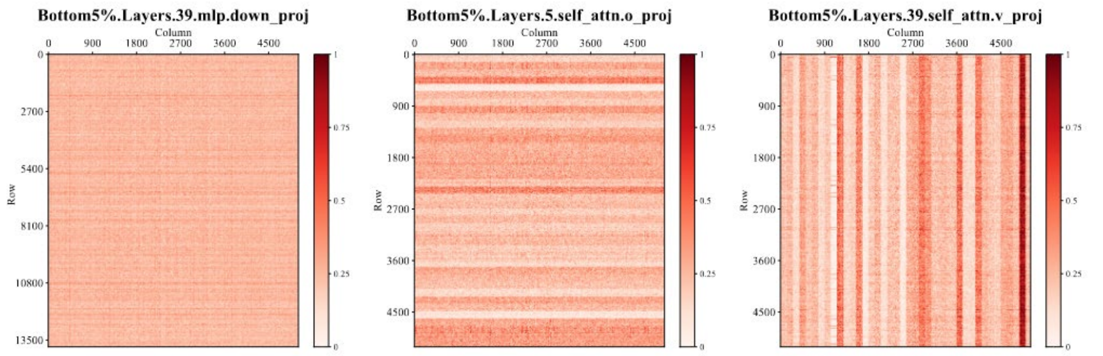


Figure 3: Visualization of the linguistic competence region (the 'Bottom' region). The scale from 0 to 1 (after normalization) represent the proportion of parameters within a 3×3 vicinity that belong to the Bottom region.

[Unveiling A Core Linguistic Region in Large Language Models (FDU, 23 Oct)](#)

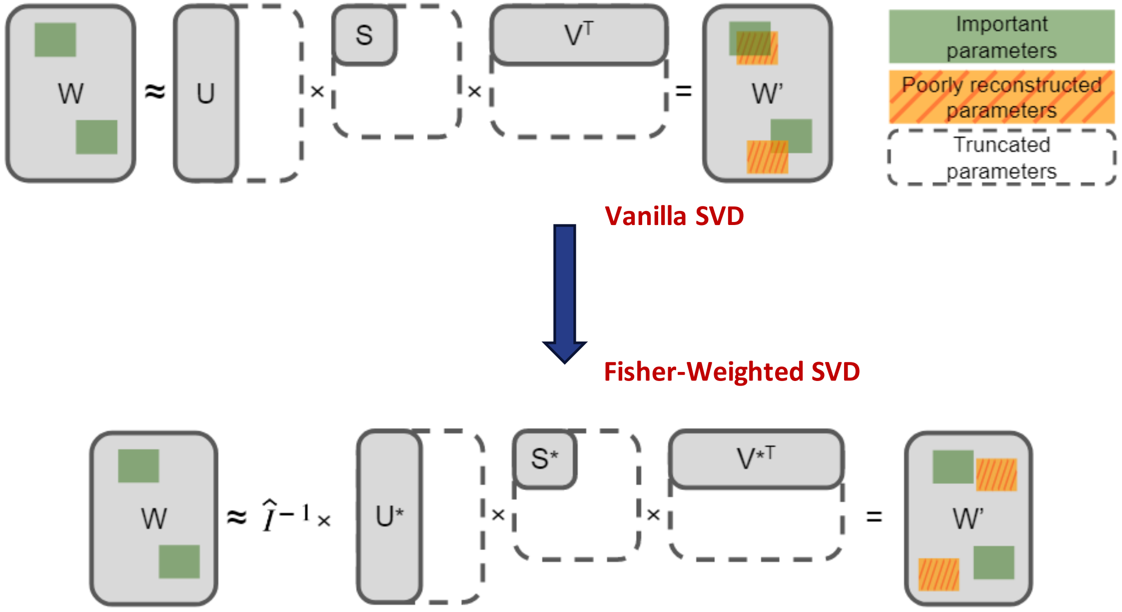
PART 03.

LEGO: A compression method based on LoRA weighted SVD

Weighted SVD

Information from Gradient

- Improve the accurate by fisher information in FWSVD.
- Align the objective of compression with the model's task, instead of minimizing reconstruction error (F- norm).
- Obtain SVD friendly matrix by assigning weights.



$$I_w = E\left[\left(\frac{\partial}{\partial w}\log p(D|w)\right)^2\right] \approx \frac{1}{|D|}\sum_{i=1}^{|D|}\left(\frac{\partial}{\partial w}\mathcal{L}(d_i;w)\right)^2 = \hat{I}_w.$$

[Language model compression with weighted low-rank factorization (ICLR 2022)](#)
[Numerical Optimizations for Weighted Low-rank Estimation on Language Model (ACL 2022)](#)

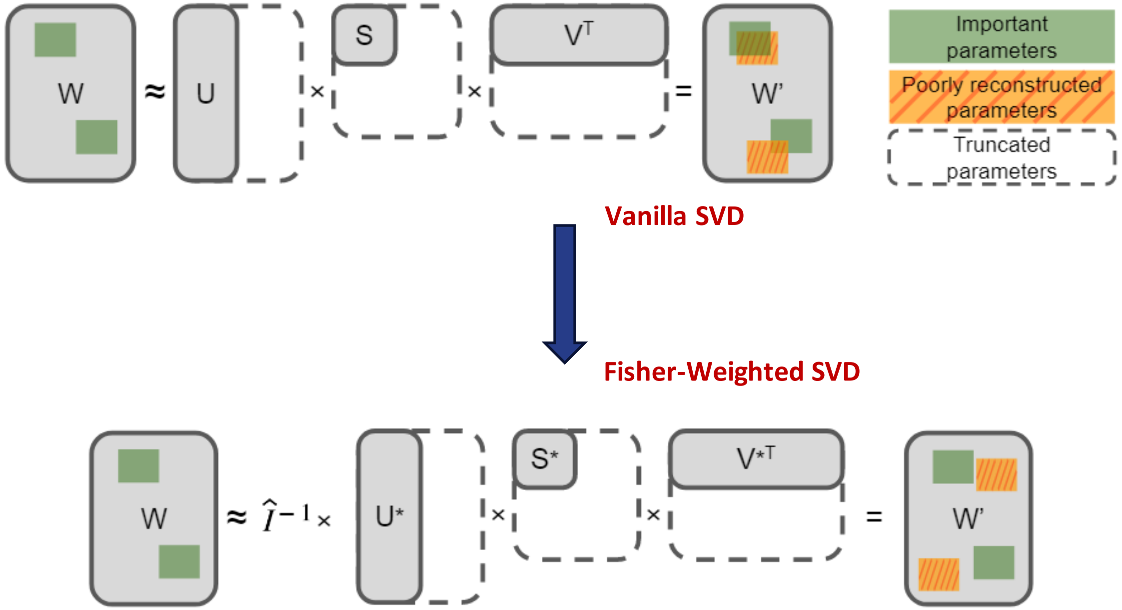
Weighted SVD

Information from Gradient

- Improve the accurate by fisher information in FWSVD.
- Align the objective of compression with the model's task, instead of minimizing reconstruction error (F- norm).
- Obtain SVD friendly matrix by assigning weights.

Limitations

- Computationally expansive
  - Forward + backward: 7B model->60GB+ memory footprint
  - Fp32 is necessary for gradient storage
- Important parameters may be miss-weighted for their relatively stable gradients



$$I_w = E\left[\left(\frac{\partial}{\partial w}\log p(D|w)\right)^2\right] \approx \frac{1}{|D|}\sum_{i=1}^{|D|}\left(\frac{\partial}{\partial w}\mathcal{L}(d_i;w)\right)^2 = \hat{I}_w.$$

Language model compression with weighted low-rank factorization (2022 ICLR)
Numerical Optimizations for Weighted Low-rank Estimation on Language Model (ACL 2022)

Fine-grained Importance Evaluation

- FWSVD: **larger gradient** on downstream dataset -> to be remained
  - But we need a general base model
  - Directly multiplying importance with weights leads to re-finetuning
- CLR: elements with **less variants** during multi-domain finetuning are important
  - Core language region -> general ability of language modeling

L E G O

Fine-grained Importance Evaluation

- FWSVD: **larger gradient** on downstream dataset -> to be remained
  - But we need a general base model
  - Directly multiplying importance with weights leads to re-finetuning
- CLR: elements with **less variants** during multi-domain finetuning are important
  - Core language region -> general ability of language modeling

---

Algorithm 1: LoRA Weighted SVD

Input: Weight matrix $A \in R^{mn}$ in LLM

$Finetune\ on\ domains\ for\ LoRA\ l_i$

$Init\ zero\ matrices\ W$

for $l = l_1, \ldots, l_i$ **do**

    $extract\ W_l^1\ and\ W_l^2\ from\ l$

    $W{+}= |W_l^1 * W_l^2|$

end

$W_{col} = \sqrt{mean(W, dim = 1)}$

$W_{row} = \sqrt{mean(W, dim = 0)}$

$USV^T = Algorithm_{svd}(W_{col} * A * W_{row})$

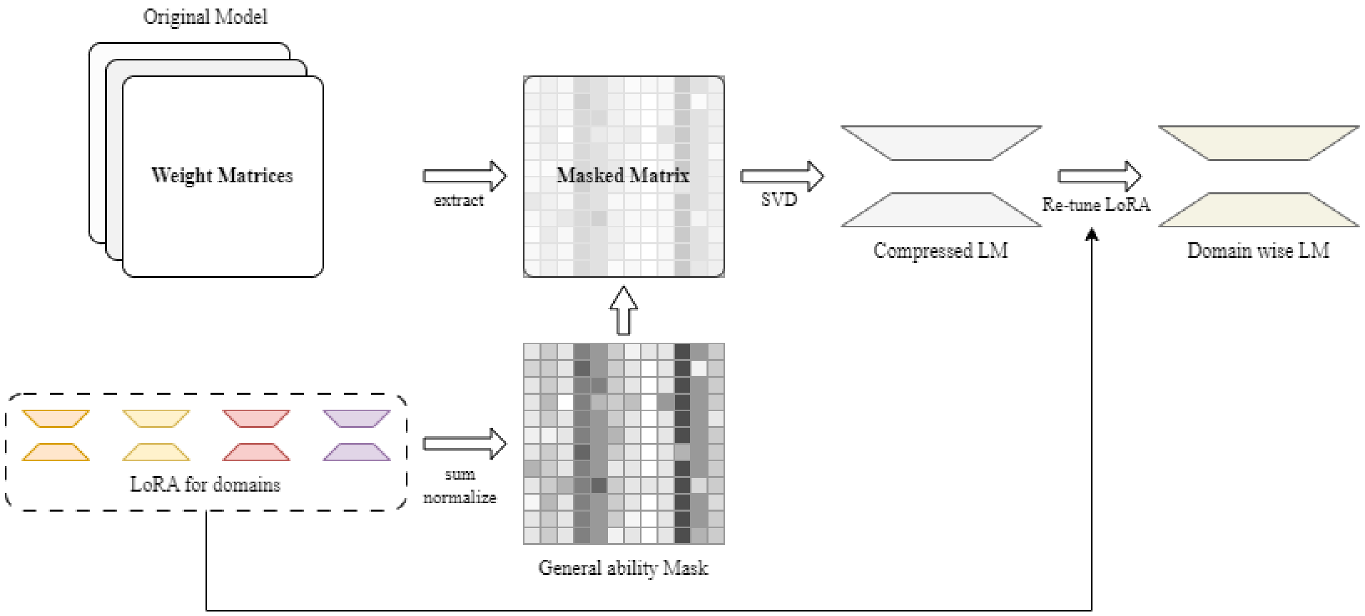
$L_1 = (U\sqrt{S})_{[:,:k]} \quad L_2 = (\sqrt{S}V^T)_{[:k,:]}$

return $L_1/W_{col},\ L_2/W_{row}$

L E G O

Fine-grained Importance Evaluation

- FWSVD: **larger gradient** on downstream dataset -> to be remained
  - But we need a general base model
  - Directly multiplying importance with weights leads to re-finetuning
- CLR: elements with **less variants** during multi-domain finetuning are important
  - Core language region -> general ability of language modeling



Algorithm 1: LoRA Weighted SVD

Input: Weight matrix $A \in R^{mn}$ in LLM

$Finetune\ on\ domains\ for\ LoRA\ l_i$

$Init\ zero\ matrices\ W$

for $l = l_1, \ldots, l_i$ **do**

  $extract\ W_l^1\ and\ W_l^2\ from\ l$

  $W+ = |W_l^1 * W_l^2|$

end

$W_{col} = \sqrt{mean(W, dim = 1)}$

$W_{row} = \sqrt{mean(W, dim = 0)}$

$USV^T = Algorithm_{svd}(W_{col} * A * W_{row})$

$L_1 = (U\sqrt{S})_{[:,:k]} \quad L_2 = (\sqrt{S}V^T)_{[:k,:]}$

return $L_1 / W_{col},\ L_2 / W_{row}$

PART 04. Experiments and future work

Baseline

LLaMA2

- Wikitext: PPL
    - Sequence length = 2048
    - torch.svd_lowrank
    - Fp16 inference

Comp ratio	LLaMA2-1.3B				LLaMA2-7B	
	SVD		FWSVD		SVD	
Dense	8.13		-		5.47	
Full rank	8.12		-		5.47	
	w/o ft	w/ ft	w/o ft	w/ ft	w/o ft	w/ ft
50%	342357.34	4500.30	640.38	$\boxed{61.02}$	49486.63	1203.74
80%	69208.00	2519.09	57971.49	770.69	28181.95	1433.54
90%	69378.82	20029.31	95004.77	1504.05	130991.09	1527.52

Table 1: SVD and FWSVD baseline: values of PPL on Wikitext-2 with sequence length of 2048.

Future Work

Experiments

- Performance of LEGO
  - Core functions have been finished, working on the test and benchmark now.
  - Evaluate both base model and domain specific models with LoRA.
- Comparisons between other methods
  - GPTQ, AWQ, QLoRA, LQ-LoRA …
  - Compress ratio, Performance, Speedup
- Ablation experiment to verify the contribution of LoRA weighted SVD.

Further Improvement

- Efficient integration of LoRA module and the compressed model.
- Combination with quantization methods.

Q & A

Thanks for your time