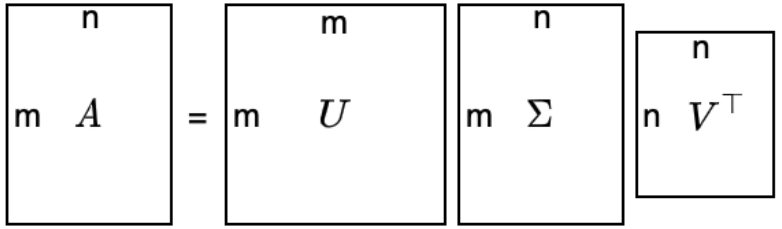
Low-Rank Prune-And-Factorize for Language Model Compression

Siyu Ren[1], Kenny Q. Zhu[2]
LREC-COLING2024 · Shanghai Jiao Tong University[1], University of Texas at Arlington[2]

Motivation

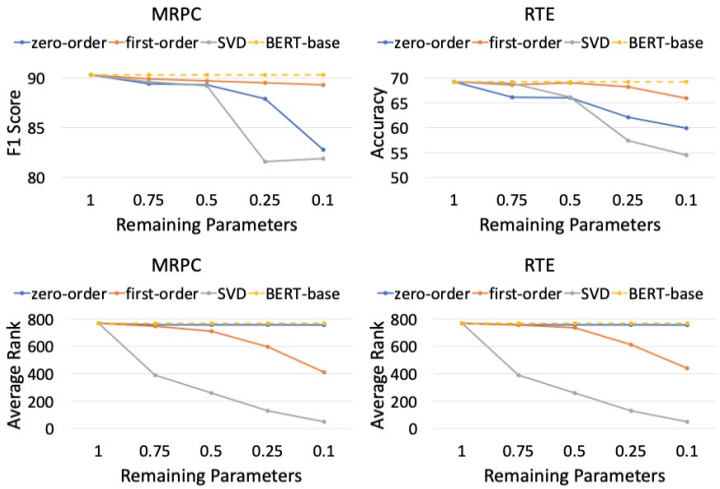
- Matrix factorization is an effective mean of reducing the size of weight matrices in large language models. It brings directly perceivable efficiency gains without the need of specialized runtime engine, e.g., DeepSparse.
- However, matrix factorization often fails to retain good task performance when compression ratio is high, e.g., >50%



Singular Value Decomposition

Motivation

- Important observations:
  - Fine-tuned LMs are high-rank(768 for BERT-base), hence direct low-rank factorization loss too much information.
  - Gradient-based pruning tends to produce low-rank sparsity pattern while still retaining decent task performance.



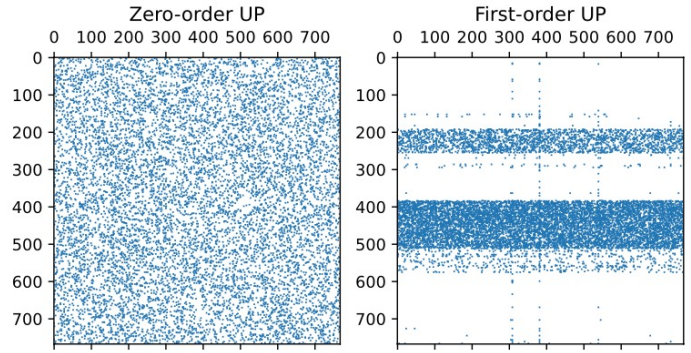
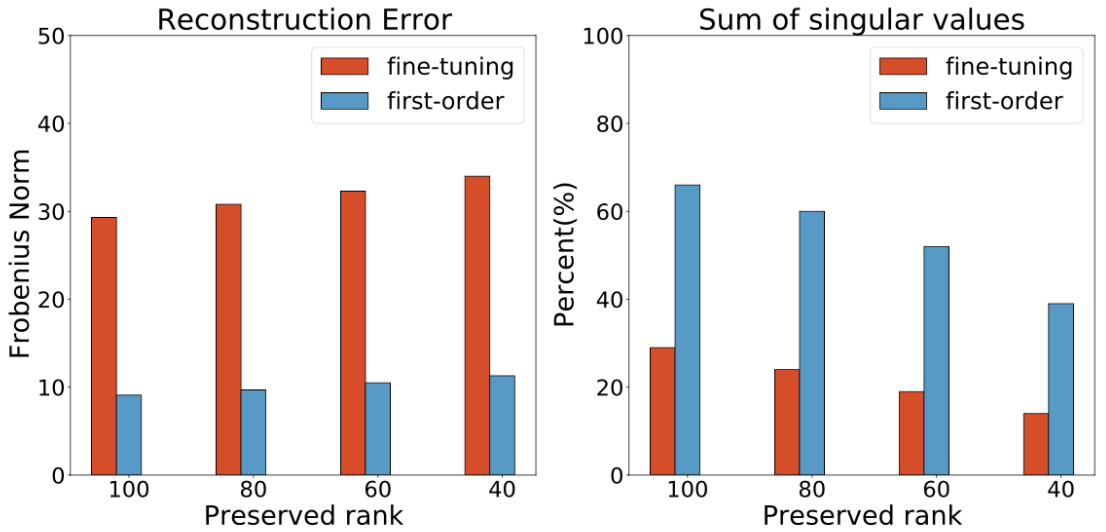


Figure 3: Sparsity patterns of the same 768x768 weight matrix pruned by UP_{zero} (left) and UP_{first} (right) on MRPC with 10% of the parameters remaining.

Motivation

- Important observations:
  - Compared to the fine-tuned parameter matrix, applying low-rank matrix decomposition to the low-rank parameter matrix obtained from first-order pruning search results in smaller reconstruction error and preserves more task-relevant knowledge.



Reconstruction error(Frobenius Norm)

Cumulative sum of singular values

Method

- Low-Rank Prune-And-Factorize for Language Model Compression

  - Step-1: obtaining the low-rank sparse model $T_{\text{sparse}} = \text{UP}_{\text{first}}(T, D, v)$. v is the percent of remained parameters after pruning.
  - Step-2: performing matrix factorization on each weight matrix (excluding the embedding layer) in T_{sparse} and obtain its low-rank factorized form $T_{\text{factorized}}$.
  - Step-3: re-training $T_{\text{factorized}}$ on D using task-specific loss function until convergence.

Method

- Optimization-1: Sparsity-aware SVD

$$\min_{\boldsymbol{A},\boldsymbol{B}} ||\boldsymbol{W}-\boldsymbol{AB}||_F = \min_{\boldsymbol{A},\boldsymbol{B}} \sum_{i,j} (\boldsymbol{W}_{i,j} - (\boldsymbol{AB})_{i,j})^2 \quad \text{s.t.} \quad \text{rank}(\boldsymbol{AB}) = k \tag{6}$$

$$\Rightarrow \quad \min_{\boldsymbol{A},\boldsymbol{B}} \sum_{i,j} \boldsymbol{S}_{i,j} (\boldsymbol{W}_{i,j} - (\boldsymbol{AB})_{i,j})^2 \quad \text{s.t.} \quad \text{rank}(\boldsymbol{AB}) = k$$

$$\Rightarrow \quad \min_{\boldsymbol{A},\boldsymbol{B}} ||\hat{\boldsymbol{I}}\boldsymbol{W} - \hat{\boldsymbol{I}}\boldsymbol{AB}||_F \quad \text{s.t.} \quad \text{rank}(\boldsymbol{AB}) = k$$

$$\boldsymbol{A} = \hat{\boldsymbol{I}}^{-1}\hat{\boldsymbol{U}}_{[:,:k]}\hat{\boldsymbol{\Sigma}}_{[:k,:k]}, \boldsymbol{B} = \hat{\boldsymbol{V}}^{\mathrm{T}}_{[:,:k]}$$

Method

- Optimization-2: Mixed-rank fine-tuning

Random replacement: $\boldsymbol{x}_{out} = (1 - z_i) * (\boldsymbol{AB})_i \boldsymbol{x}_{in} + z_i * \boldsymbol{W}_i \boldsymbol{x}_{in}$

Probability scheduler: $p = \max(0, p_{\text{init}} - d * t)$

Consistency loss: $\mathcal{L}_c = \mathcal{D}(y_{\boldsymbol{z}^1}, y_{\boldsymbol{z}^2})$

Experimental Results

- Results of various methods on GLUE benchmark.

Task	RTE (2.5K)			MRPC (3.7K)			CoLA (8.5K)			SST-2 (67K)			QQP (364K)			QNLI (105K)			MNLI (393K)		
% Params.	50%	25%	16%	50%	25%	16%	50%	25%	16%	50%	25%	16%	50%	25%	16%	50%	25%	16%	50%	25%	16%
DistilBERT	65.0	61.0	56.3	85.8	77.0	72.5	51.3	32.1	21.1	90.0	88.9	86.4	90.8	89.4	88.0	86.0	83.8	81.6	81.7	76.4	71.3
TinyBERT	67.7	67.2	64.6	86.3	85.3	78.2	53.8	33.3	21.3	92.3	89.8	88.0	90.5	90.0	88.7	89.9	87.7	84.5	83.1	80.6	77.4
PKD	65.5	59.2	53.8	81.9	76.2	71.3	45.5	22.0	19.1	91.3	88.1	87.2	88.4	88.5	87.5	88.4	82.7	78.0	81.3	75.7	72.7
Theseus	65.6	62.1	58.8	86.2	77.2	72.8	51.1	17.9	17.6	91.5	88.5	86.1	89.6	89.0	86.0	89.5	85.0	80.3	82.3	76.4	73.5
CKD	67.3	66.5	60.8	86.0	81.1	76.6	55.1	40.1	32.9	**93.0**	89.8	88.7	91.2	90.1	88.9	90.5	87.0	84.9	83.6	79.0	76.8
MetaDistil	69.0	66.7	61.0	86.8	81.8	77.3	**56.3**	33.6	24.3	92.3	88.9	87.0	91.0	88.9	86.9	90.4	86.8	84.9	83.5	79.5	76.8
ISP	66.4	65.0	63.9	86.1	83.6	82.8	55.3	45.6	31.0	90.6	90.4	89.4	90.8	90.1	89.3	90.5	88.7	87.2	83.2	81.9	80.8
FLOP	66.1	58.5	56.0	82.1	80.1	78.4	49.1	35.3	28.6	91.4	89.7	89.4	91.1	90.1	89.1	90.5	88.5	87.1	82.6	79.9	79.4
BP_{hybrid}	66.4	64.3	63.9	84.1	83.8	81.1	50.0	37.3	35.4	90.8	89.8	89.2	90.8	90.1	89.8	90.2	88.7	88.1	83.2	80.6	80.1
CoFi	**69.3**	66.4	66.4	84.6	84.3	83.6	51.8	44.1	30.3	91.6	89.7	89.2	91.0	90.2	89.9	90.8	88.8	87.6	83.5	80.8	80.5
SVD_{Ft}	62.1	60.3	55.6	79.9	70.1	70.0	44.9	26.6	18.0	90.8	88.9	85.3	91.3	90.0	87.9	91.0	86.1	83.8	83.0	79.9	76.6
LPAF (ours)	68.2	**68.0**	**67.9**	**86.8**	**86.5**	**86.0**	55.5	**48.5**	**42.8**	92.4	**90.7**	**89.7**	**91.5**	**90.4**	**90.1**	**91.3**	**89.3**	**88.6**	**84.6**	**82.6**	**81.7**
-w/o Step-1	64.2	32.1	21.1	82.1	32.1	21.1	49.0	32.9	18.2	91.2	89.9	88.4	91.3	90.3	89.7	91.2	87.8	84.8	83.3	82.0	79.6
-w/o Step-2	65.3	32.1	21.1	86.0	32.1	21.1	52.0	48.0	41.0	91.2	89.2	88.8	91.2	90.2	90.0	90.9	89.0	87.9	83.4	82.4	81.5
-w/o Step-3	65.0	32.1	21.1	84.8	32.1	21.1	52.9	48.2	42.2	91.4	89.5	88.8	91.1	90.3	89.9	91.1	88.9	88.1	83.0	81.3	81.0
BERT-base	69.2			86.4			57.8			92.7			91.5			91.4			84.6		

Experimental Results

- Results of various methods on SQuAD v1.1 and SQuAD v2.0 for extractive question-answering.

Task	SQuAD v1.1 (88K)			SQuAD v2.0 (131K)		
% Params.	50%	25%	16%	50%	25%	16%
DistilBERT	85.8	78.0	66.5	68.2	62.5	56.2
TinyBERT	82.5	58.0	38.1	72.2	85.3	78.2
Theseus	84.2	72.7	63.2	71.2	77.2	72.8
ISP	86.0	84.9	81.9	76.9	74.1	71.8
FLOP	88.1	85.7	81.5	77.7	75.3	71.3
CoFi	87.7	86.8	84.9	77.3	73.9	72.4
SVD_{Ft}	87.8	85.5	81.1	77.4	70.1	70.0
LPAF (ours)	**89.1**	**87.2**	**85.7**	**79.1**	**77.2**	**75.1**
BERT-base	88.2			77.9		

Experimental Results

- Compressing an already compact language model MiniLM with 2x compression ratio.

Task	SST-2	QNLI	MNLI-m/mm
CKD	**91.2**	89.3	83.0/83.7
SVD_{Ft}	90.0	89.6	82.8/83.0
LPAF	91.1	**90.5**	**84.4/84.5**
MiniLM	92.4	91.2	85.0/85.2

Ablation Study

- Ablation study on our proposed optimization strategies:
  - Sparsity-aware SVD
  - Mixed-rank fine-tuning

	Before→After Step-3		
Strategy/k	260	130	80
w/ S	**81.4→92.4**	**79.9→90.7**	**77.5→89.7**
w/ M	81.0→92.1	79.7→90.4	77.2→89.3
Vanilla	79.1→91.4	77.9→89.2	75.9→88.8

Different SVD objectives

Fine-tuning Method	k=260	k=130	k=80
mixed-rank	**92.4**	**90.7**	**89.7**
- w/o $\mathcal{L}_c$	91.9	89.8	89.1
vanilla fine-tuning	91.4	89.5	88.8

Different fine-tuning objectives

Thanks!