中国人民大學
RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院
Gaoling School of Artificial Intelligence

Enabling Lightweight Fine-tuning for Pre-trained Language Model Compression based on Matrix Product Operators

Peiyu Liu *, Ze-Feng Gao *, Wayne Xin Zhao†,
Z.Y. Xie, Zhong-Yi Lu†, Ji-Rong Wen

* equal contribution
† corresponding author

Outline

- Introduction
  - Background
  - Motivation
- Preliminary
- Approach
- Experiments
- Conclusion

Introduction-Background

Background:

- Pre-training and fine-tuning paradigm
- Huge number of parameters

Observation:

- Only a small proportion of parameters will significantly change during fine-tuning.

Model	#Total Param	#Trainable Param
BERT_base	108M	108M
BERT_large	334M	334M
BERT_xlarge	1270M	1270M

Introduction-Motivation

Matrix Product Operator (MPO)

MPO factorizes a matrix into a sequential product of local tensors.

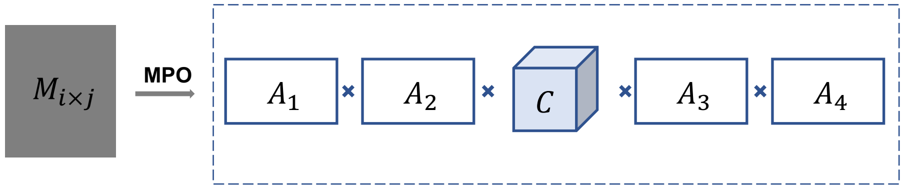
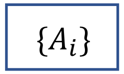
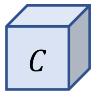


Figure 1: MPO decomposition for $M_{i\times j}$.



The auxiliary tensors with only a small proportion of parameters play a role of complementing the central tensor



The central tensor with most of parameters encode the core information of the original matrix

Motivation:

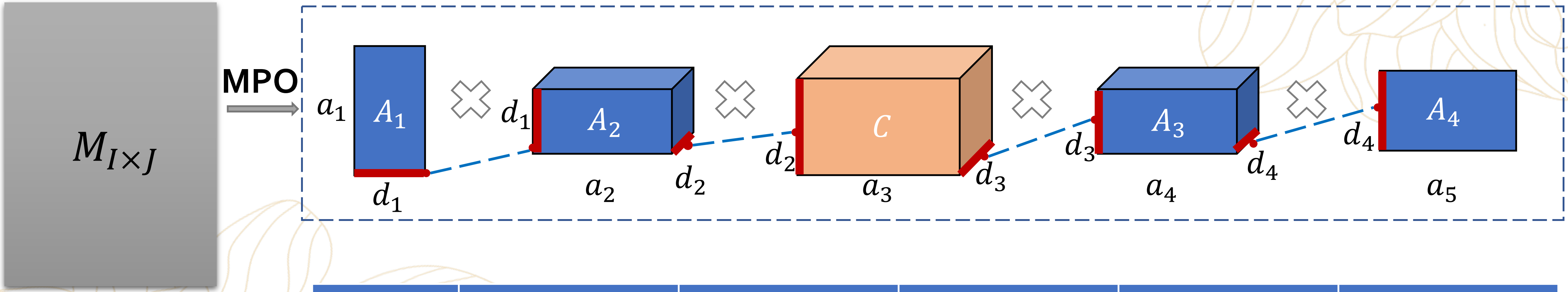
Can we compress the central tensor for parameter reduction and update auxiliary tensors for lightweight fine-tuning?

Outline

- Introduction
- Preliminary
- Approach
- Experiments
- Conclusion

Preliminary

MPO: matrix product operator technique from quantum many-body physics for compressing PLMs.



768×3072

$I = [3,4,4,4,4]$,
$J = [4,4,8,6,4]$

Tensor	A_1	A_2	A_3	A_4	A_5
shape	1×3×4×12	12×4×4×192	192×4×8×384	384×4×6×16	16×4×4×1
# ratio	0.006%	1.45%	92.74%	5.80%	0.01%

Almost all the parameters

Preliminary

MPO: matrix product operator technique from quantum many-body physics for compressing PLMs.

Matrix decomposition with MPO:

$$\mathrm{MPO}(\mathrm{M}) = \prod_{k=1}^{n} \mathcal{T}_{(k)}[d_{k-1}, i_k, j_k, d_k], (1)$$

The bond dimension d_k is defined by:

$$d_k = \min(\prod_{m=1}^{k} i_m \times j_m, \prod_{m=k+1}^{n} i_m \times j_m). (2)$$

Algorithm 1 MPO decomposition for a matrix.

Input: matrix $\mathbf{M}$, the number of local tensors n
Output : MPO tensor list $\{\mathcal{T}_{(k)}\}_{k=1}^{n}$

for $k = 1 \rightarrow n-1$ **do**
    $\mathbf{M}[I, J] \longrightarrow \mathbf{M}[d_{k-1} \times i_k \times j_k, -1]$
    $\mathbf{U}\lambda\mathbf{V}^{\top} = \mathrm{SVD}(\mathbf{M})$
    $\mathbf{U}[d_{k-1} \times i_k \times j_k, d_k] \longrightarrow \mathcal{U}[d_{k-1}, i_k, j_k, d_k]$
    $\mathcal{T}^{(k)} := \mathcal{U}$
    $\mathbf{M} := \lambda\mathbf{V}^{\top}$
end for
$\mathcal{T}^{(n)} := \mathbf{M}$
Normalization
return $\{\mathcal{T}_{(k)}\}_{k=1}^{n}$

Preliminary

- MPO-based low-rank approximation
  - The truncation error induced by the k-th bond dimension d_k is denoted by ϵ_k (called local truncation error)

Truncation error: $$\epsilon_k = \sum_{i=d_k-d'_k}^{d_k} \lambda_i\,, \qquad (3)$$

Reconstruction error: $$\|\mathrm{M}-\mathrm{MPO}(\mathrm{M})\|_F \le \sqrt{\sum_{k=1}^{n-1} \epsilon_k^2}\,, \qquad (4)$$

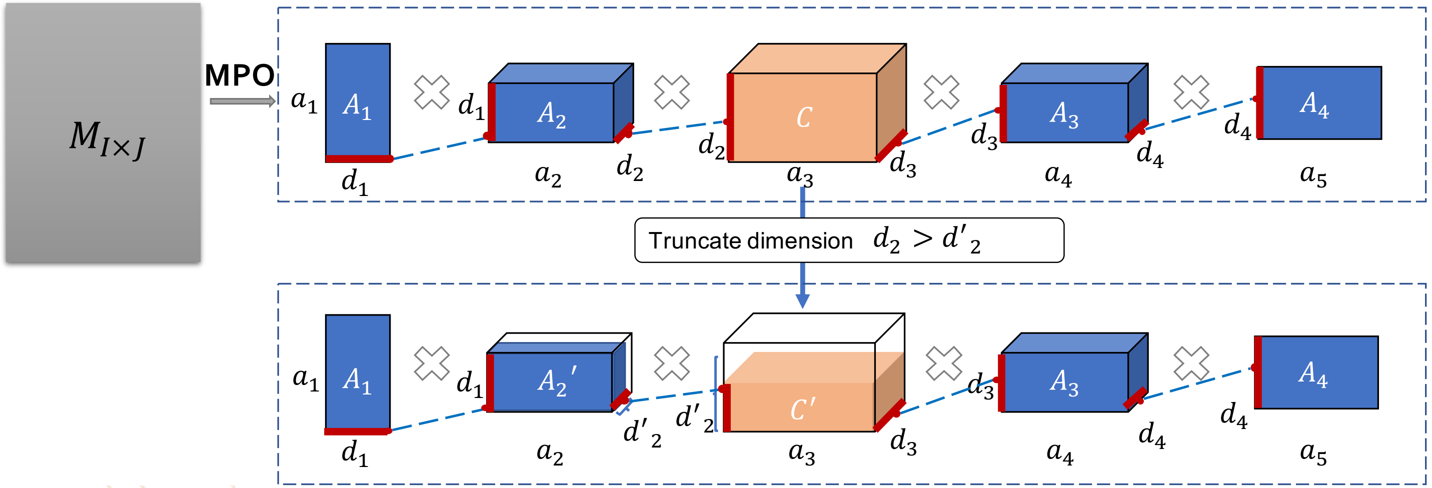
Compression ratio: $$\rho = \frac{\sum_{k=1}^{n} d'_{k-1} i_k j_k d'_k}{\prod_{k=1}^{n} i_k j_k}\,, \qquad (5)$$

Outline

- Introduction
- Preliminary
- Approach
    - Overview
    - Part 1: Lightweight fine-tuning
    - Part 2: Dimension squeezing
    - Discussion
- Experiments
- Conclusion

Overview

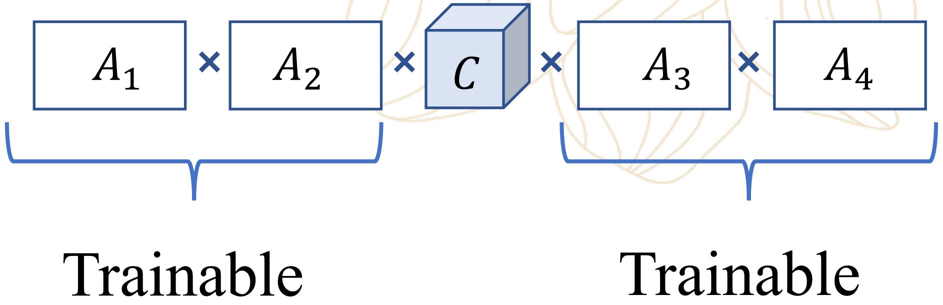
- Motivation
  - ➢ Can we compress the central tensor for parameter reduction and update auxiliary tensors for lightweight fine-tuning?
- Solution
  - ➢ Lightweight fine-tuning with auxiliary tensors
  - ➢ Dimension squeezing for stacked architecture optimization



Part 1: Lightweight Fine-tuning

Layers	(0,1e-4]	(1e-4,1e-3]	(1e-3,∞)
Word embedding	0.66	0.26	0.09
Feed-forward	0.09	0.64	0.27
Self-attention	0.09	0.64	0.27

Table 1: Distribution of parameter variations for BERT when fine-tuned on SST-2 task.

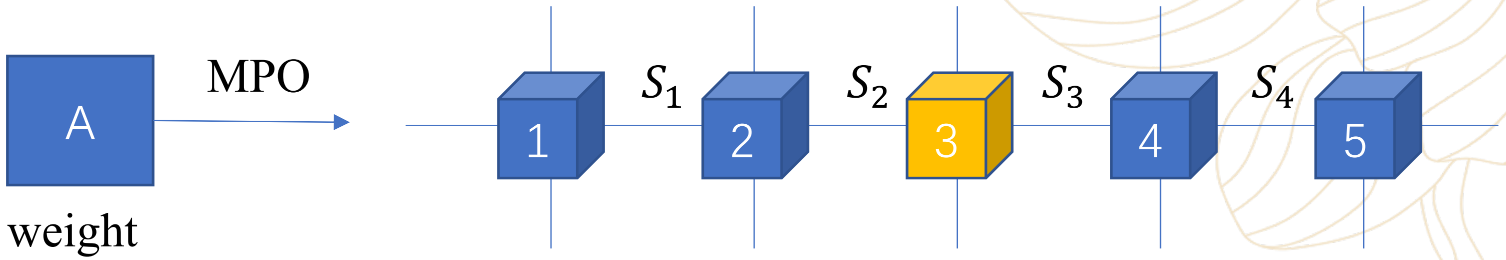


Observation: Variation degree of the parameters before and after fine-tuning.

Solution: Fix central tensor and update auxiliary tensors.

Part 1: Lightweight Fine-tuning

- Theoretical analysis



➢ Entanglement entropy: the metric to measure the information contained in MPO bonds[1], Calculation methods:

$$S_k = -\sum_{j=1}^{d_k} v_j \ln v_j, \qquad k = 1,2,\ldots,n-1, \qquad (6)$$

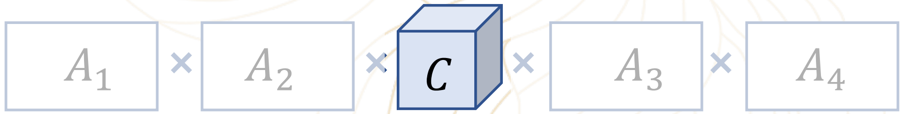
[1] Ze-Feng Gao, Song Cheng, Rong-Qiang He, ZY Xie, Hui-Hai Zhao, Zhong-Yi Lu, and Tao Xiang. 2020. Compressing deep neural networks by matrix product operators. Physical Review Research, 2(2):023300.

Part 2: Dimension Squeezing

- Motivation:
  - ➢ Low-rank approximation on C will largely reduce total parameters.

- Fast Reconstruction Error Estimation
  - ➢ Criterion
  - ➢ Efficiencies

- Fast Performance Gap Computation
  - ➢ Early stopping



Algorithm 2 Training with dimension squeezing.

Input: : L layers with corresponding central tensor $\mathcal{C}^{(l)}$ and dimension $d^{(l)}$, threshold Δ and iteration step $iter$

```
Evaluate loss p = model(Inputs)
Perform MPO decomposition for each layer
for step = 1 → iter do
    Find the layer (l*) with the least reconstruction error
    Compress MPO tensor by truncating d^(l*)
    Fine-tuning auxiliary tensors with {C^(l)}_{l=1}^L fixed
    Evaluate loss p̃ = model(Inputs)
    if || p − p̃ ||> Δ then
        break
    end if
end for
return Compressed model
```

Discussion

- Comparing with Tucker decomposition

Category	Method	Inference Time
Tucker	$\text{Tucker}_{(d=1)}$(CP)	$\mathcal{O}(nmd^2)$
	$\text{Tucker}_{(d>1)}$	$\mathcal{O}(nmd + d^n)$
MPO	$\text{MPO}_{(n=2)}$(SVD)	$\mathcal{O}(2md^3)$
	$\text{MPO}_{(n>2)}$	$\mathcal{O}(nmd^3)$

Table 2: Inference time complexities of different low-rank approximation methods. Here, n denotes the number of the tensors, m denotes $\max(\{i_k\}_{k=1}^n)$ means the largest i_k in input list, and d denotes $\max(\{d'_k\}_{k=0}^n)$ means the largest dimension d'_k in the truncated dimension list.

Outline

- Introduction
- Preliminary
- Approach
- Experiments
    - Experimental Results
    - Detailed Analysis
- Conclusion

Experimental Results

Experiments	Score	SST-2 (acc)	MNLI (m_cc)	QNLI (acc)	CoLA (mcc)	STS-B (ρ)	QQP (acc)	MRPC (acc)	RTE (acc)	WNLI (acc)	Avg. #Pr/#To(M)
$\text{ALBERT}_{\text{pub}}$	-	90.3	81.6	-	-	-	-	-	-	-	11.6/11.6
$\text{ALBERT}_{\text{rep}}$	78.9	90.6	**84.5**	89.4	53.4	88.2	89.1	88.5	71.1	54.9	11.6/11.6
MPOP	**79.7**	**90.8**	83.3	**90.5**	**54.7**	**89.2**	**89.4**	**89.2**	**73.3**	**56.3**	**1.1/9**
$\text{MPOP}_{\text{full}}$	80.3	92.2	84.4	91.4	55.7	89.2	89.6	87.3	76.9	56.3	12.7/12.7
$\text{MPOP}_{\text{full+LFA}}$	80.4	93.0	84.3	91.3	56.0	89.2	89.0	88.0	78.3	56.3	1.2/12.7
MPOP_{dir}	68.6	86.6	79.2	81.9	15.0	82.5	87.0	74.3	54.2	56.3	1.1/9

Table 3: Performance on GLUE benchmark obtained by fine-tuning ALBERT and MPOP. "$\text{ALBERT}_{\text{pub}}$" and "$\text{ALBERT}_{\text{rep}}$" denote the results from the original paper (Lan et al., 2020) and reproduced by ours, respectively. "#Pr" and "#To" denote the number (in millions) of pre-trained parameters and total parameters, respectively.

Experimental Results

- Ablation results

Experiments	Score	SST-2 (acc)	MNLI (m_cc)	QNLI (acc)	CoLA (mcc)	STS-B (ρ)	QQP (acc)	MRPC (acc)	RTE (acc)	WNLI (acc)	Avg. #Pr/#To(M)
$\text{ALBERT}_{\text{pub}}$	-	90.3	81.6	-	-	-	-	-	-	-	11.6/11.6
$\text{ALBERT}_{\text{rep}}$	78.9	90.6	**84.5**	89.4	53.4	88.2	89.1	88.5	71.1	54.9	11.6/11.6
MPOP	**79.7**	**90.8**	83.3	**90.5**	**54.7**	**89.2**	**89.4**	**89.2**	**73.3**	**56.3**	**1.1/9**
$\text{MPOP}_{\text{full}}$	80.3	92.2	84.4	91.4	55.7	89.2	89.6	87.3	76.9	56.3	12.7/12.7
$\text{MPOP}_{\text{full+LFA}}$	80.4	93.0	84.3	91.3	56.0	89.2	89.0	88.0	78.3	56.3	1.2/12.7
MPOP_{dir}	68.6	86.6	79.2	81.9	15.0	82.5	87.0	74.3	54.2	56.3	1.1/9

<table>
<tr><th>MPO representation</th><th>Fine-tuning</th><th>Experiment</th></tr>
<tr><td rowspan="2">Full-rank</td><td>Regular fine-tuning</td><td>$\text{MPOP}_{\text{full}}$</td></tr>
<tr><td>Lightweight fine-tuning</td><td>$\text{MPOP}_{\text{full+LFA}}$</td></tr>
<tr><td>Truncate rank directly</td><td rowspan="2">Lightweight fine-tuning</td><td>MPOP_{dir}</td></tr>
<tr><td>Dimension squeezing</td><td>MPOP</td></tr>
</table>

Experimental Results

- Ablation results

Models	WNLI (acc)	MRPC (acc)	RTE (acc)	Avg. #Pr/#To(M)
BERT	56.3	**85.5**	70.0	110/110
$MPOP_B$	56.3	84.3	**70.8**	**7.7/70.4**
DistilBERT	56.3	84.1	61.4	66/66
$MPOP_D$	56.3	**84.3**	**61.7**	**4.0/43.4**
MobileBERT	56.2	**86.0**	63.5	25.3/25.3
$MPOP_M$	56.2	85.3	**65.7**	**4.4/15.4**

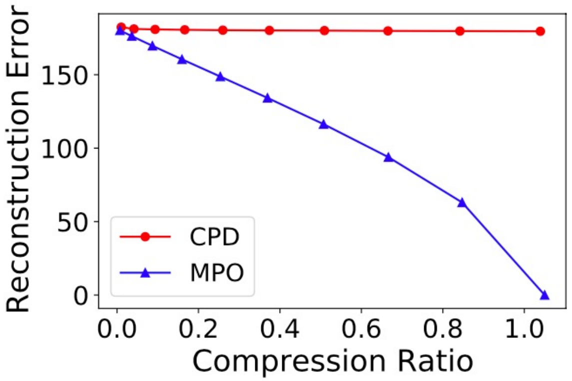
Table 4: Evaluation with different BERT variants.

Models	SST-2	MRPC	RTE	Avg. #Pr(M)
$BERT_{10-12}$	91.9	76.5	67.2	45.7
$BERT_{11-12}$	91.7	75.3	62.8	38.6
$BERT_{12}$	91.4	72.1	61.4	31.5
$MPOP_B$	**92.6**	**84.3**	**70.8**	**10.1**

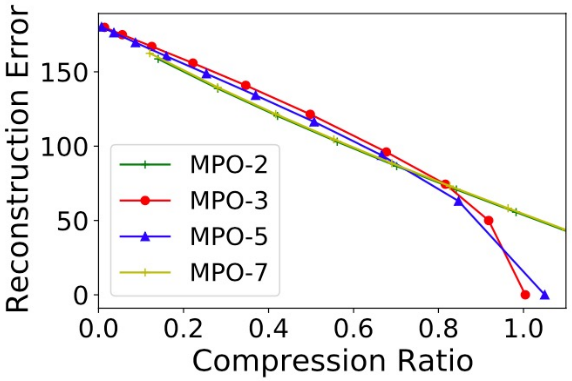
Table 5: Comparison of different fine-tuning strategies on three GLUE tasks. The subscript number in $BERT_{(\cdot)}$ denotes the index of the layers to be fine-tuned.

Experimental Results

- Ablation results



(a) CPD *v.s.* MPO.



(b) # of local tensors.

Figure 2: Comparison of different low-rank approximation variants. x-axis denotes the compression ratio (ρ in Eq. (5)) and y-axis denotes the reconstruction error, measured in the Frobenius norm.

Outline

- Introduction
- Preliminary
- Approach
- Experiments
- Conclusion

Conclusion

We proposed an MPO-based PLM compression method. With MPO decomposition, we were able to reorganize and aggregate information in central tensors effectively. Inspired by this, we make following contributions:

- **Lightweight fine-tuning strategy:** we largely reduced the parameters to be fine-tuned by only updating the auxiliary tensors.
- **Dimension squeezing algorithm:** we could optimize low-rank approximation over stacked network architectures.

Q&A

Source code

https://github.com/RUCAIBox/MPOP

Thank you