



Introduction

Background

At present, Pre-training model plays an key role in many neural language processing tasks. However, Transformer, and its variant model BERT, limit the effective deployment of the model to limited resource setting.

- ◆ The compression of large nature pre-training language model has been an essential problem in NLP research.
- ◆ There are some compression methods only study the compression of embedding layers and some methods can not be integrated into the model after compressing.

Research Questions

- ◆ To linearly represent a self-attention by a group of basic vectors
- ◆ To compress multi-head attention in Transformer
- ◆ After compressing, it can be directly integrated into the encoder and decoder framework of Transformer

Our Methods

Basic Ideas

- Low-rank decomposition
- Parameters sharing

- ◆ Using Tucker decomposition formulation is to construct Single-block attention
- ◆ Using Block-term decomposition + Parameters sharing formulation is to construct multi-head mechanisms(Multi-linear attention)

Tensoried Transformer

- Single-block Attention by Tucker Decomposition

$$Atten_{TD}(\mathcal{G}; Q, K, V) = \sum_{i=1}^I \sum_{j=1}^J \sum_{m=1}^M \mathcal{G}_{i,j,m} Q_i \circ K_j \circ V_m$$

- Multi-linear Attention by Block-term Decomposition

$$MultiLinear(\mathcal{G}; Q, K, V) = SplitConcat\left(\frac{1}{h} \cdot (T_1 + \dots T_h)\right) W^O$$

where $T_j = Atten_{TD}(\mathcal{G}_j; QW^q, KW^k, VW^v)$

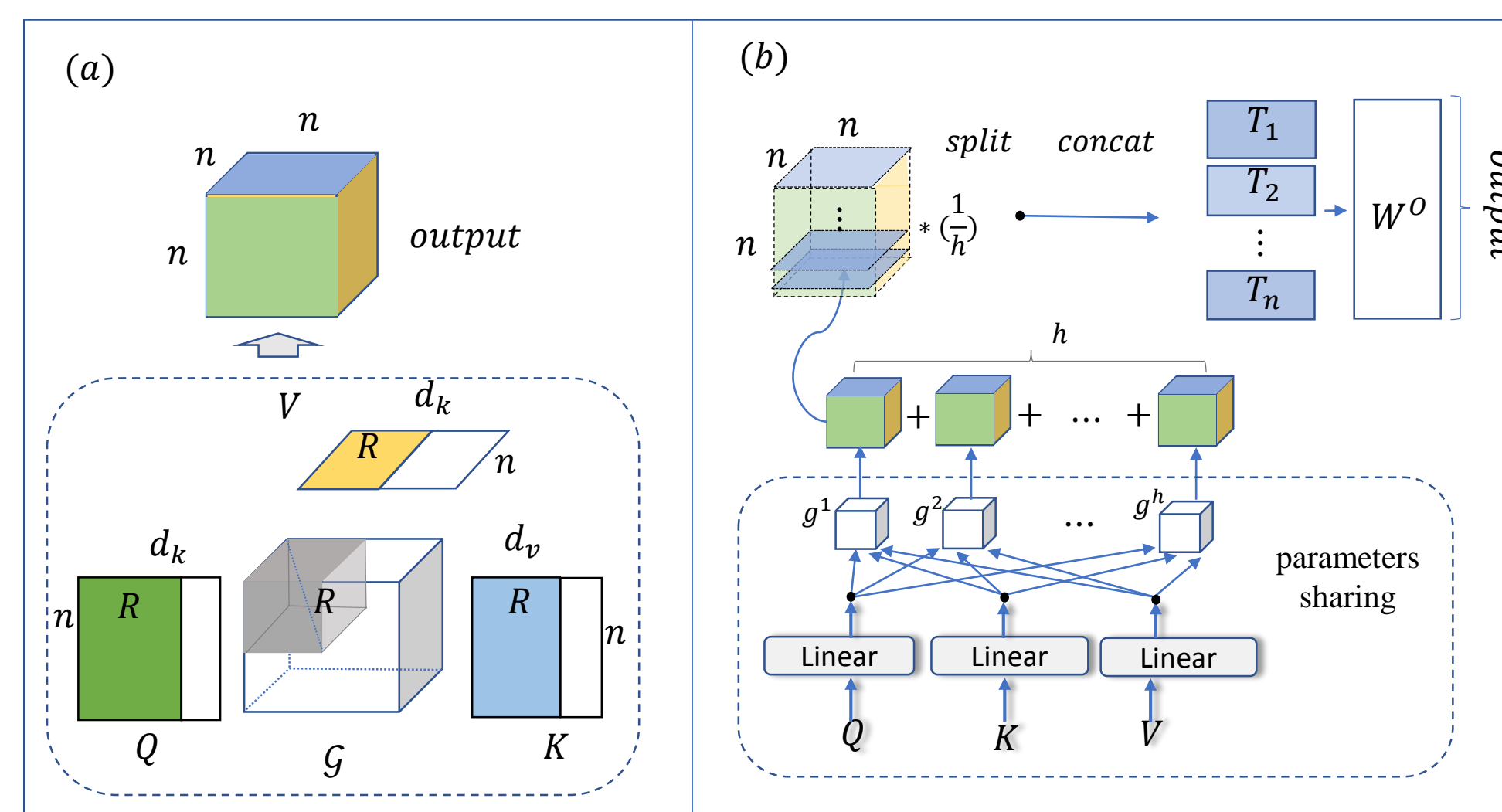


Figure: (a) is the Single-block attention using Tucker decomposition, (b) is the Multi-linear attention based on Block-term tensor decomposition.

Main Theorem

Let e_1, \dots, e_n be basis vectors from the vector S . Assume that Q, K, V can be linearly represented by this set of basis vector. The output of the attention function can be represented by a linear combination of the set of these basis vectors.

$$Attention(Q, K, V) = (e_1, \dots, e_n)M$$

where $M \in \mathbb{R}^{N \times d}$ is a coefficient matrix, and d is a dimension of these matrices.

Conclusion

- Providing a novel self-attention method, namely Multi-linear attention.
- Combining two compression ideas, parameters sharing and low-rank decomposition.
- Achieving higher compression ratio and better experimental results in language modeling

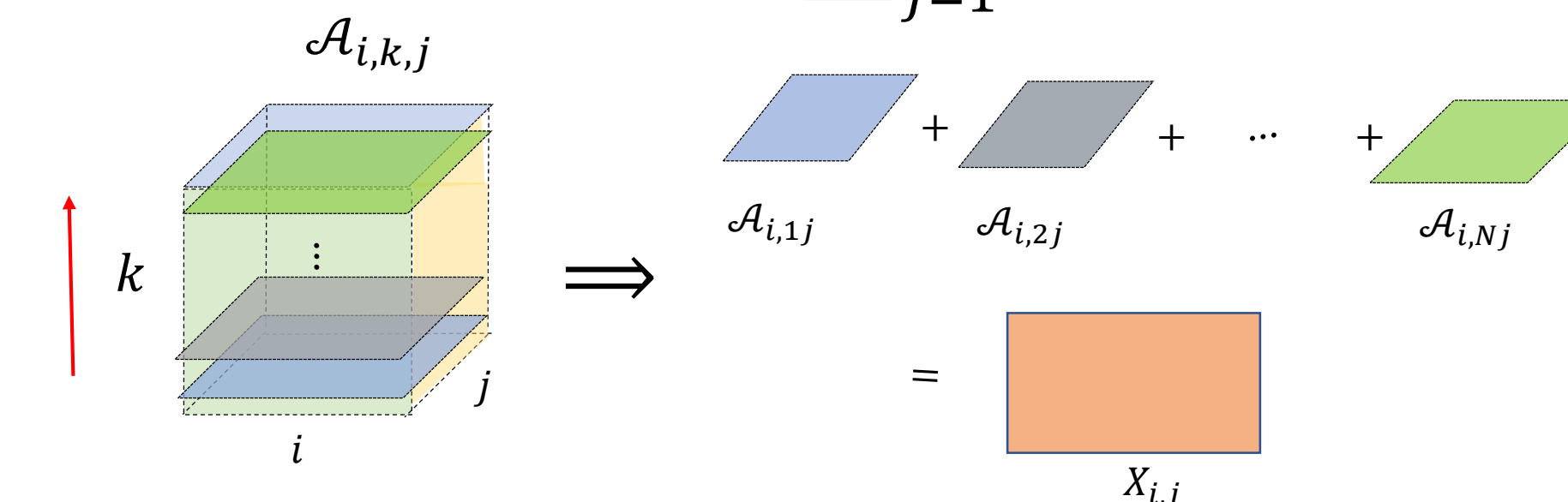
Xindian Ma, Peng Zhang*, Shuai Zhang,

Nan Duan, Yuexian Hou, Dawei Song and Ming Zhou

Related Corollary

Single-block attention can reconstruct the self attention function by the summing over the tensor according to the second index.

$$Attention(Q, K, V)_{i,m} = \sum_{j=1}^N Atten_{TD}(\mathcal{G}; Q, K, V)_{i,j,m}$$



Experimental Results

Model	Params	PTB		WikiText-103	
		Val PPL	Test PPL	Val PPL	Test PPL
LSTM+augmented loss [15]	24M	75.7	48.7	—	48.7
Variational RHN [41]	23M	67.9	65.4	—	45.2
4-layer QRNN [21]	—	—	—	151M	33.0
AWD-LSTM-MoS [37]	22M	58.08	55.97	—	29.2
Transformer+adaptive input [1]	24M	59.1	57	247M	20.5
Transformer-XL-Base [7]	24M	56.72	54.52	151M	23.1
Transformer-XL-Large [7]	—	—	—	257M	18.3
Transformer-XL+TT [18]	18 M	57.9*	55.4*	130M	23.61*
Sparse Transformer [28]	14M	74.0*	73.1*	174M	38.98*
Tensorized Transformer core-1	12M	60.5	57.9	85.3M	22.7
Tensorized Transformer core-2	12M	54.25	49.8	85.3M	19.7

One-Billion

Model	Params	Test PPL
RNN-1024+9 Gram [4]	20B	51.3
LSTM-2018-512 [17]	0.83B	43.7
GCNN-14 bottleneck [8]	—	31.9
LSTM-8192-1024+CNN Input [17]	1.04B	30.0
High-Budget MoE [32]	5B	28.0
LSTM+Mos [37]	113M	37.10
Transformer+adaptive input [1]	0.46B	23.7
Transformer-XL Base [7]	0.46B	23.5
Transformer-XL Large [7]	0.8B	21.8
Tensorized Transformer core-1	0.16B	20.5
Tensorized Transformer core-2	0.16B	19.5

Language Modeling

- Our method is mainly experimented on three language model datasets, PTB, WikiText-103, and One-Billion, respectively. The lower the PPL, the better the model is.
- Our methods achieve a more better results with fewer parameters.

Main References

- [1] Ashish et al., Attention is all you need. NeurIPS, 2017.
- [2] Zihang et al., Transformer-xl: Attentive language models beyond a fixed-length context, 2019.
- [3] Valentin et al, Tensorized embedding layers for efficient model compression, 2019.