Weight-Inherited Distillation for Task-Agnostic BERT Compression

Taiqiang Wu[1*], Cheng Hou[2*], Shanshan Lao[3],
Jiayi Li[3], Ngai Wong[1], Zhe Zhao[2], Yujiu Yang[3]

[1]The University of Hong Kong, [2]Tencent AI Lab, [3] Tsinghua University

Contribution Highlights

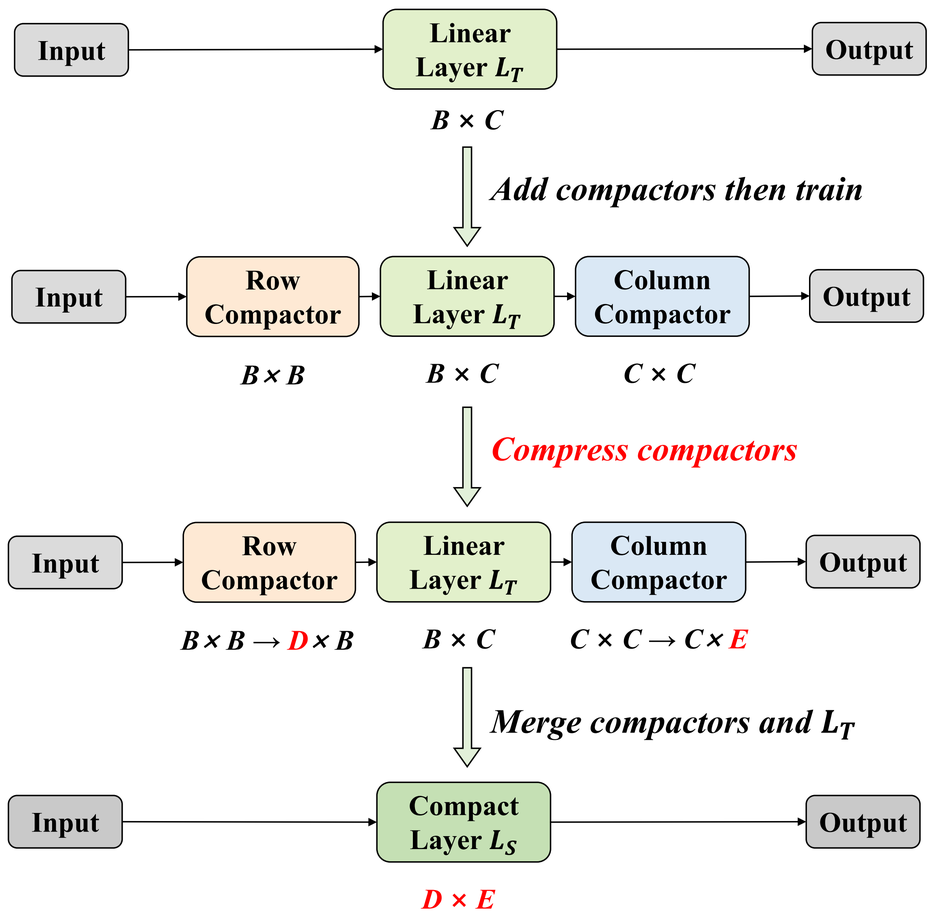

Figure: Overview of proposed Weight-Inherited Distillation (WID).

Knowledge Distillation (KD), which trains a compact student model by mimicking the behavior of a teacher model, is a predominant method for model compression. Previous KD-based methods focus mainly on designing alignment losses to minimize the distance between the teacher model and the student model. However, selecting various loss functions and balancing the weights of each loss are laborious. Meanwhile, the knowledge is embedded in the weights. This gives rise to an intuitive thought:

Can we distill the knowledge by directly inheriting the weights, rather than aligning the logit distributions or intermediate features?

In this paper, we propose Weight-Inherited Distillation (WID), which does not require any additional alignment loss and trains the student by directly inheriting the weights from the teacher.

The main contributions of our paper are as follows:

- We propose Weight-Inherited Distillation (WID), revealing a new pathway to KD by directly inheriting the weights via structural re-parameterization.
- We design the compactor alignment strategy and conduct WID for task-agnostic BERT compression. Experiments on the GLUE and SQuAD benchmark datasets demonstrate the effectiveness of WID for model compression.
- We perform further analyses on how to get better performance in BERT compression. Moreover, we find that WID is able to learn attention patterns from the teacher.

Compactor Compression

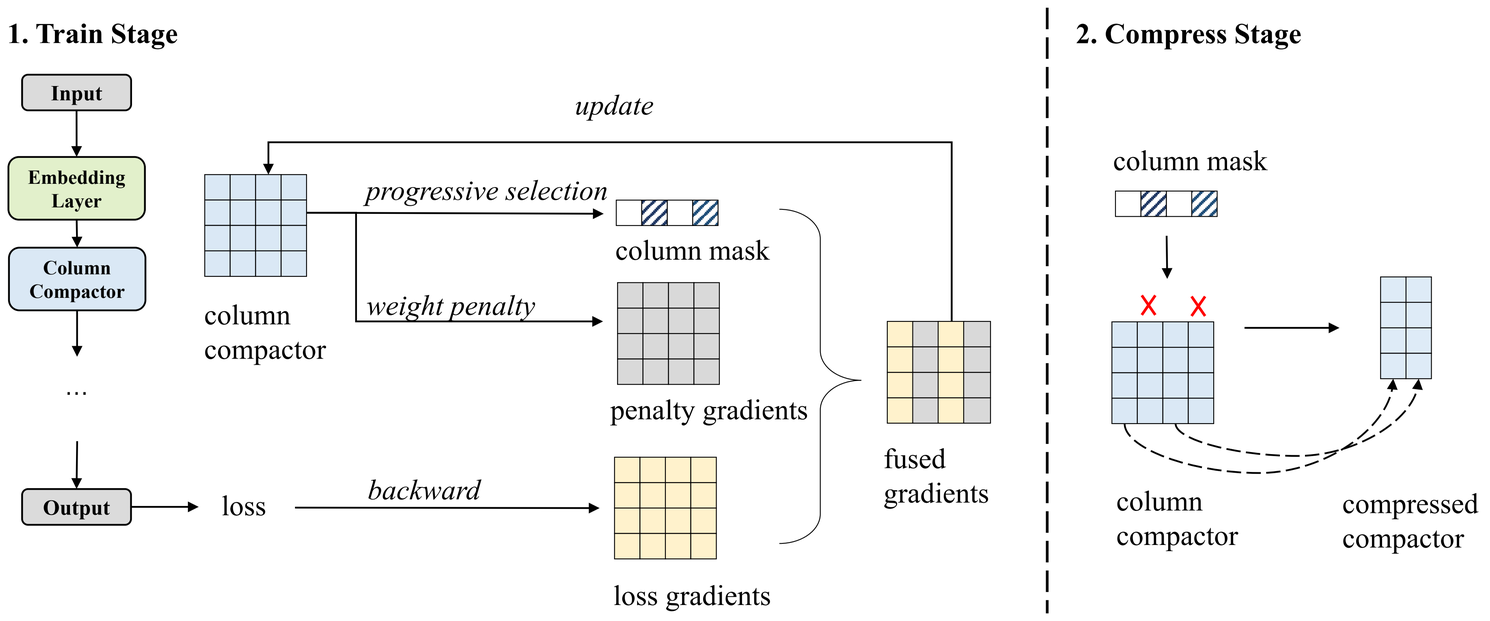

Figure: Training and compression for column compactors.

- Both row compactor and column compactor are initialized as **identity matrices**. Therefore, for an arbitrary input, the re-parameterized teacher model produces identical outputs as the original.
- During the training process, we add weight penalty gradients by columns and progressively select the mask to fuse the penalty gradients and original loss gradients. After training, we compress the column compactor following the column mask.
- Finally, we merge the compressed compactors and the original teacher layer to obtain the compact layer for the student.

Compactor Alignment Strategy

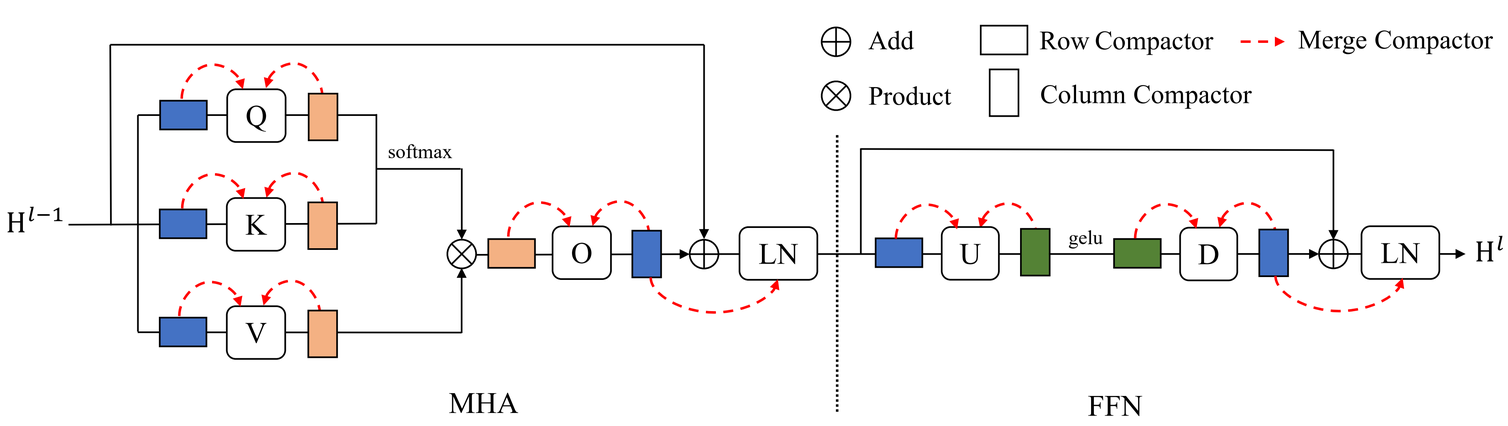

Figure: Compactor merging process for a Transformer block.

For the bias terms, we merge them with corresponding column compactors. For beta and gamma in Layer Norm (LN), we adopt the previous column compactors to update them. During training, the compactors in the **same color** are aligned. For each group of the aligned compactors, we learn one of them and duplicate (or, flip) it for the rest compactors.

BERT Compression

Table: Comparison between our WID and various task-agnostic distillation methods.

Method	FLOPs	Parameters	Average Score
$\text{BERT}_{\text{base}}$	22.7B	110.1M	84.3
DistilBERT	11.9B	67.5M	80.1
TinyBERT (GD)†	11.9B	67.5M	81.3
TinyBERT (GD)‡	10.4B	54.9M	81.2
WID_{55} (ours)	10.4B	54.9M	**83.4**
TinyBERT (GD)‡	1.6B	11.3M	75.6
WID_{11} (ours)	1.6B	11.3M	**76.7**

→ WID outperforms all the baselines.

GPT Compression

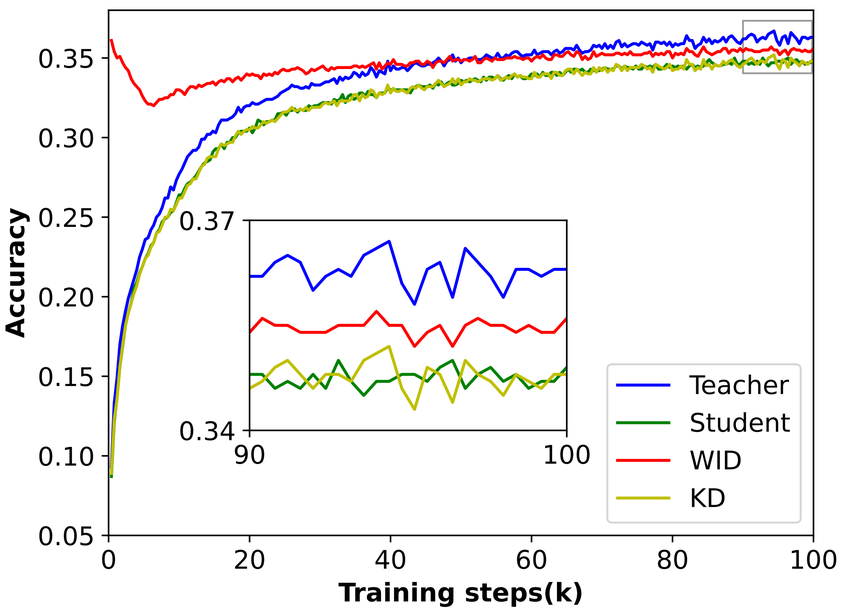

Figure: The training process for teacher GPT, vanilla student GPT, and students via KD and WID.

→ WID works for generative pre-trained language models and can get better performance than vanilla KD.

Resources

Code

Paper