TinyBERT: Distilling BERT for Natural Language Understanding

Xiaoqi Jiao[1*†], **Yichun Yin**[2*], **Lifeng Shang**[2], **Xin Jiang**[2]
Xiao Chen[2], **Linlin Li**[3], **Fang Wang**[1] **and Qun Liu**[2]
[1]Huazhong University of Science and Technology
[2]Huawei Noah's Ark Lab
[3]Huawei Technologies Co., Ltd.

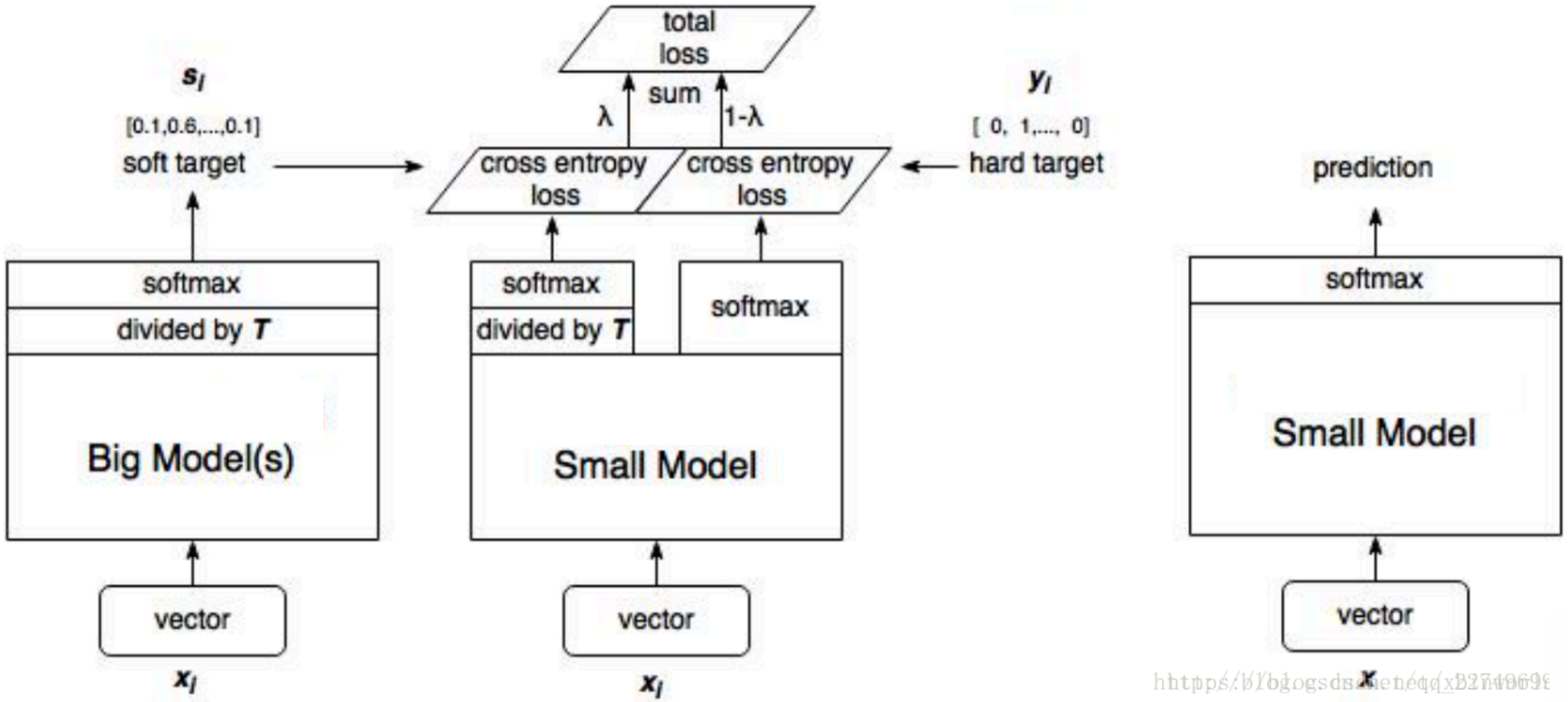
Knowledge Distilling

Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*†]
Google Inc.
Mountain View
geoffhinton@google.com

Oriol Vinyals[†]
Google Inc.
Mountain View
vinyals@google.com

Jeff Dean
Google Inc.
Mountain View
jeff@google.com



$$q_i = \frac{exp(z_i/T)}{\Sigma_j exp(z_j/T)}$$

$$L = \alpha L^{(soft)} + (1-\alpha) L^{(hard)}$$

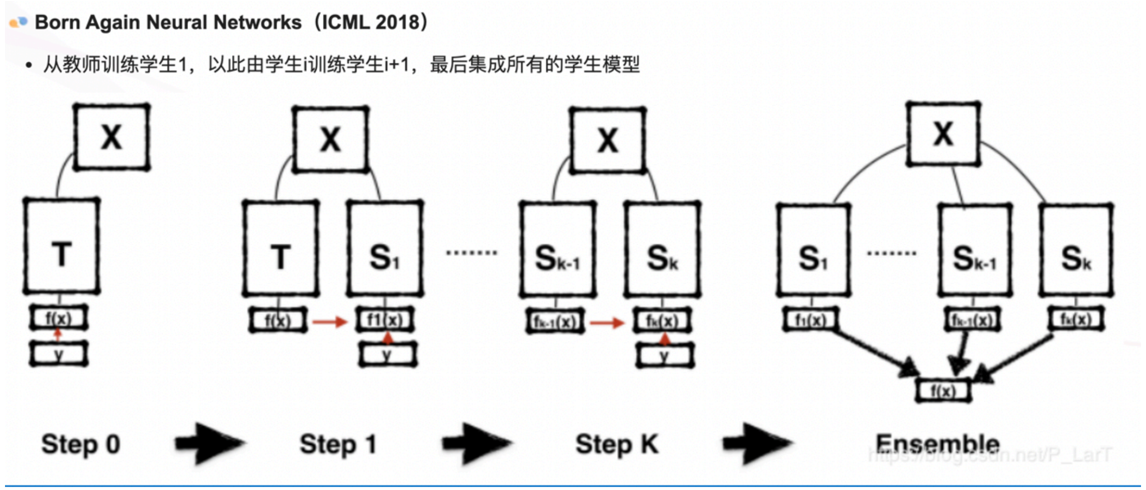
T的作用：smoothing，缩小差距

$$\mathcal{L}_{\text{distill}} = ||\boldsymbol{z}^{(B)} - \boldsymbol{z}^{(S)}||_2^2$$

Why KD ?

1. Smaller ! Faster !
   - => 量化
   - => 剪枝
     - => 对权重连接，也就是权重矩阵中的某个位置
     - => 对神经元，可以反映在权重矩阵的某一行/列
     - => 对整个权重矩阵

2. Improve performance



3. Non-autoregressive Machine Translation

Table 1: A summary of KD methods for BERT. Abbreviations: INIT(initializing student BERT with some layers of pre-trained teacher BERT), DA(conducting data augmentation for task-specific training data). Embd, Attn, Hidn, and Pred represent the knowledge from embedding layers, attention matrices, hidden states, and final prediction layers, respectively.

KD Methods	KD at Pre-training Stage					KD at Fine-tuning Stage				
	INIT	Embd	Attn	Hidn	Pred	Embd	Attn	Hidn	Pred	DA
Distilled BiLSTM$_{\text{SOFT}}$									✓	✓
BERT-PKD	✓							✓[3]	✓	
DistilBERT	✓				✓[4]				✓	
TinyBERT (our method)		✓	✓	✓		✓	✓	✓	✓	✓

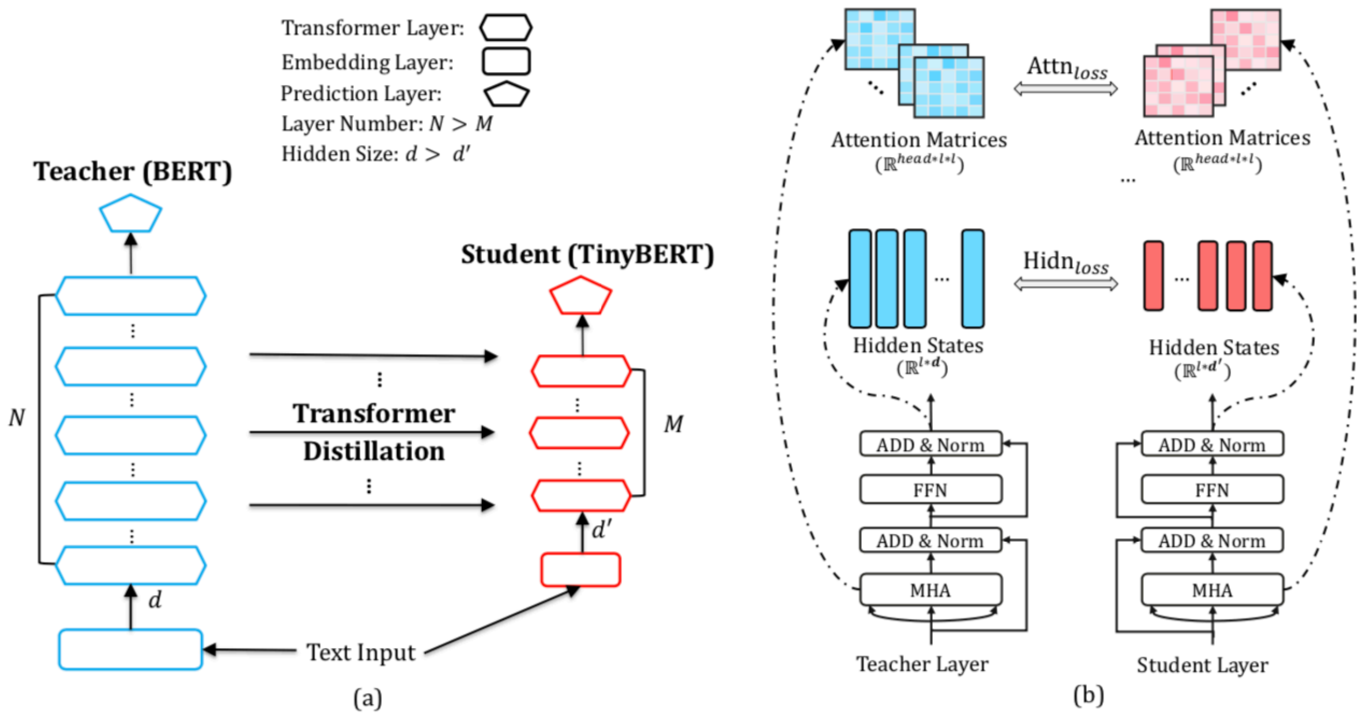


Figure 1: An overview of Transformer distillation: (a) the framework of Transformer distillation, (b) the details of Transformer-layer distillation consisting of Attn$_{loss}$(attention based distillation) and Hidn$_{loss}$(hidden states based distillation).

$$\mathcal{L}_{\text{model}} = \sum_{m=0}^{M+1} \lambda_m \mathcal{L}_{\text{layer}}(S_m, T_{g(m)}),$$

$$\mathcal{L}_{\text{layer}}(S_m, T_{g(m)}) = \begin{cases} \mathcal{L}_{\text{embd}}(S_0, T_0), & m = 0 \\ \mathcal{L}_{\text{hidn}}(S_m, T_{g(m)}) + \mathcal{L}_{\text{attn}}(S_m, T_{g(m)}), & M \geq m > 0 \\ \mathcal{L}_{\text{pred}}(S_{M+1}, T_{N+1}), & m = M + 1 \end{cases}$$

$$\mathcal{L}_{\text{embd}} = \texttt{MSE}(\boldsymbol{E}^S \boldsymbol{W}_e, \boldsymbol{E}^T),$$

$$\mathcal{L}_{\text{hidn}} = \texttt{MSE}(\boldsymbol{H}^S \boldsymbol{W}_h, \boldsymbol{H}^T),$$

$$\mathcal{L}_{\text{attn}} = \frac{1}{h} \sum_{i=1}^{h} \texttt{MSE}(\boldsymbol{A}_i^S, \boldsymbol{A}_i^T),$$

$$\mathcal{L}_{\text{pred}} = -\texttt{softmax}(\boldsymbol{z}^T) \cdot \texttt{log\_softmax}(\boldsymbol{z}^S / t),$$ find that $t = 1$ performs well.

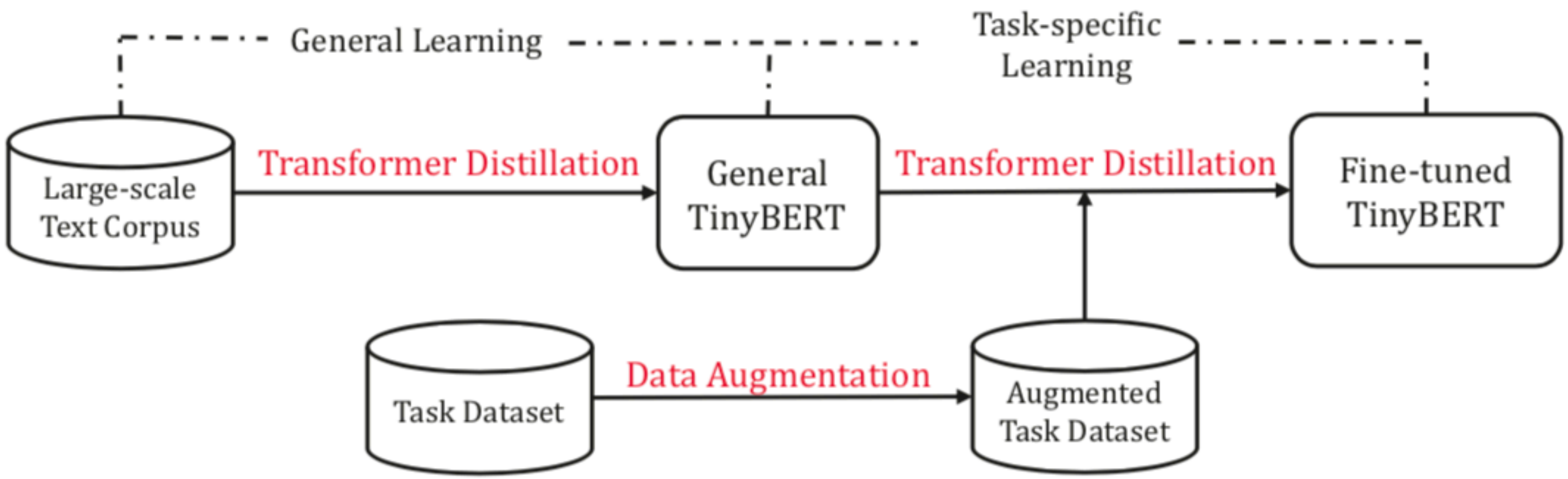


Figure 2: The illustration of TinyBERT learning

Table 2: Results are evaluated on the test set of GLUE official benchmark. All models are learned in a single-task manner. "-" means the result is not reported.

System	MNLI-m	MNLI-mm	QQP	SST-2	QNLI	MRPC	RTE	CoLA	STS-B	Average
$\text{BERT}_{\text{BASE}}$ (Google)	84.6	83.4	71.2	93.5	90.5	88.9	66.4	52.1	85.8	79.6
$\text{BERT}_{\text{BASE}}$ (Teacher)	83.9	83.4	71.1	93.4	90.9	87.5	67.0	52.8	85.2	79.5
$\text{BERT}_{\text{SMALL}}$	75.4	74.9	66.5	87.6	84.8	83.2	62.6	19.5	77.1	70.2
Distilled $\text{BiLSTM}_{\text{SOFT}}$	73.0	72.6	68.2	90.7	-	-	-	-	-	-
BERT-PKD	79.9	79.3	70.2	89.4	85.1	82.6	62.3	24.8	79.8	72.6
DistilBERT	78.9	78.0	68.5	91.4	85.2	82.4	54.1	32.8	76.1	71.9
TinyBERT	82.5	81.8	71.3	92.6	87.7	86.4	62.9	43.3	79.9	76.5

Table 3: The model sizes and inference time for baselines and TinyBERT. The number of layers does not include the embedding and prediction layers.

System	Layers	Hidden Size	Feed-forward Size	Model Size	Inference Time
$\text{BERT}_{\text{BASE}}$ (Teacher)	12	768	3072	109M($\times$1.0)	188s($\times$1.0)
Distilled $\text{BiLSTM}_{\text{SOFT}}$	1	300	400	10.1M($\times$10.8)	24.8s($\times$7.6)
BERT-PKD/DistilBERT	4	768	3072	52.2M($\times$2.1)	63.7s($\times$3.0)
TinyBERT/$\text{BERT}_{\text{SMALL}}$	4	312	1200	14.5M($\times$7.5)	19.9s($\times$9.4)

Table 5: Ablation studies of different procedures (i.e., TD, GD, and DA) of the two-stage learning framework. The variants are validated on the dev set.

System	MNLI-m	MNLI-mm	MRPC	CoLA	Average
TinyBERT	82.8	82.9	85.8	49.7	75.3
No GD	82.5	82.6	84.1	40.8	72.5
No TD	80.6	81.2	83.8	28.5	68.5
No DA	80.5	81.0	82.4	29.8	68.4

Table 6: Ablation studies of different distillation objectives in the TinyBERT learning. The variants are validated on the dev set.

System	MNLI-m	MNLI-mm	MRPC	CoLA	Average
TinyBERT	82.8	82.9	85.8	49.7	75.3
No Embd	82.3	82.3	85.0	46.7	74.1
No Pred	80.5	81.0	84.3	48.2	73.5
No Trm	71.7	72.3	70.1	11.2	56.3
No Attn	79.9	80.7	82.3	41.1	71.0
No Hidn	81.7	82.1	84.1	43.7	72.9

Table 7: Results (dev) of different mapping strategies.

System	MNLI-m	MNLI-mm	MRPC	CoLA	Average
TinyBERT (Uniform-strategy)	82.8	82.9	85.8	49.7	75.3
TinyBERT (Top-strategy)	81.7	82.3	83.6	35.9	70.9
TinyBERT (Bottom-strategy)	80.6	81.3	84.6	38.5	71.3

这里其实也可以总结一下一些KD的套路：

- soft label（+hard label）用 交叉熵/MSE
- temperature
- 大模型初始化小模型
- 利用各个layer的中间状态给loss学习
- 小模型各个层对应大模型哪个layer（uniform/top/bottom）