About distillation, a few words

Karol Kaczmarek

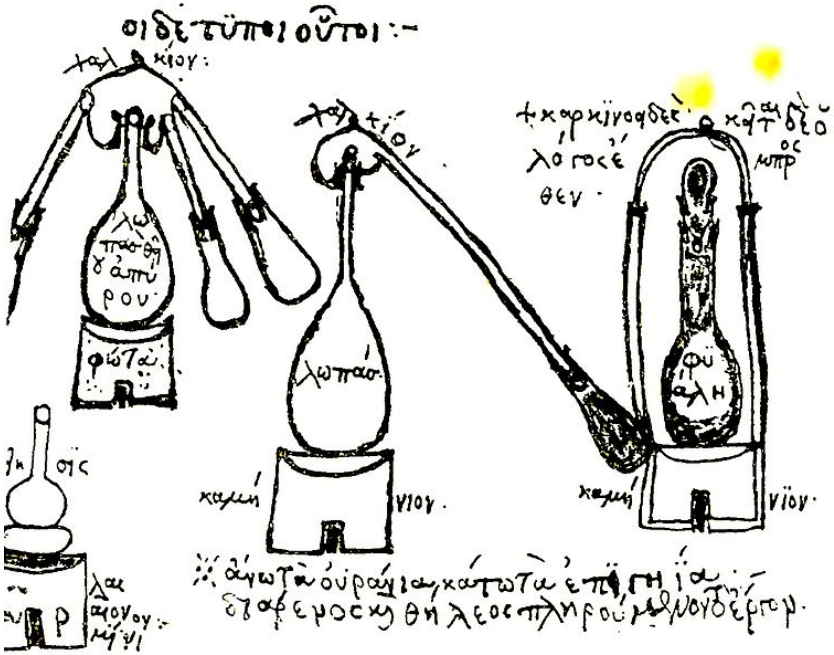
Adam Mickiewicz University
Poznań

Applica.ai
Warsaw

2021

Distillation - definition

- the action of purifying a liquid by a process of heating and cooling
- the extraction of the essential meaning or most important aspects of something



Knowledge distillation

- The **student** - is trained to reproduce the behavior of a larger model - the **teacher**
- **Knowledge distillation** (KD) is to train the small **student** model **S** on a transfer feature set with soft labels and intermediate representations provided by the large **teacher** model **T**.
- Knowledge distillation is modeled as minimizing the differences between teacher and student features:
  - $L_{\mathbf{KD}} = \sum_{e \in D} L(f^S(e), f^T(e))$
  - D - training data
  - $f^S(\cdot)$ and $f^T(\cdot)$ - features of student and teacher models respectively
  - $L(\cdot)$ - loss function, often used: **MSE** - mean squared error or **KL-divergence** (for probability distribution)

DistilBERT [1]

- August 2019, HuggingFace – code available
- DistilBERT - pre-trained version of BERT, 40% smaller, 60% faster, that retains 97% of the language understanding capabilities
- based on BERT (base)
- triple loss combining **language modeling**, **distillation** and **cosine-distance** losses
- focus on reducing the number of layers
- initialize the student from the teacher by taking one layer out of two (better than random weights initialization)
- knowledge distillation during the pre-training phase

Triple loss functions

The final training objective is a linear combination of:

- L_{ce} - the distillation loss (loss over the soft target probabilities of the teacher, use *softmax-temperature* – controls the smoothness of the output distribution)
- L_{mlm} - the supervised training loss (the masked language modeling loss)
- L_{cos} - the cosine embedding loss (align the directions of the student and teacher hidden states vectors)

Ablation	**Variation on GLUE macro-score**
$\emptyset$ - L_{cos} - L_{mlm}	-2.96
L_{ce} - $\emptyset$ - L_{mlm}	-1.46
L_{ce} - L_{cos} - $\emptyset$	-0.31
Triple loss + random weights initialization	-3.69

GLUE results (dev-set) and inference time

- GLUE results
  - **BERT**: 12 layers and 768 hidden size
  - **DistilBERT**: 6 layers and 768 hidden size

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

- Inference time (on CPU using a batch size of 1):

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

Distillation during fine-tuning

Model	IMDb (acc.)	SQuAD (EM/F1)
BERT-base	93.46	81.2/88.5
DistilBERT	92.82	77.7/85.8
DistilBERT (D)	-	79.1/86.9

TinyBERT [2]

- December 2019, Huawei – code available
- TinyBERT - 7,5x smaller, 9,4x faster on inference, that retains 96% of the language understanding capabilities
- based on BERT (base)
- design several loss functions to fit different representations from BERT layers
- propose a two-stage learning framework including the *general distillation* (distillation + pretraining) and the *task-specific distillation* (distillation + fine-tune + data augmentation)

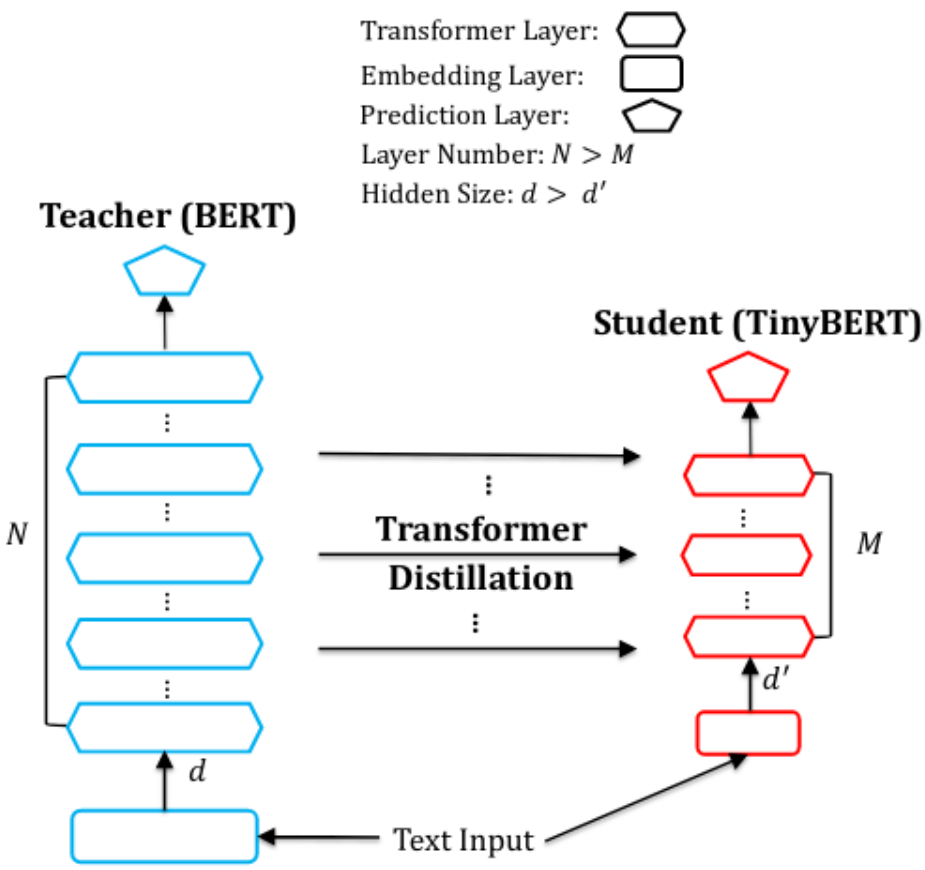
Several loss functions

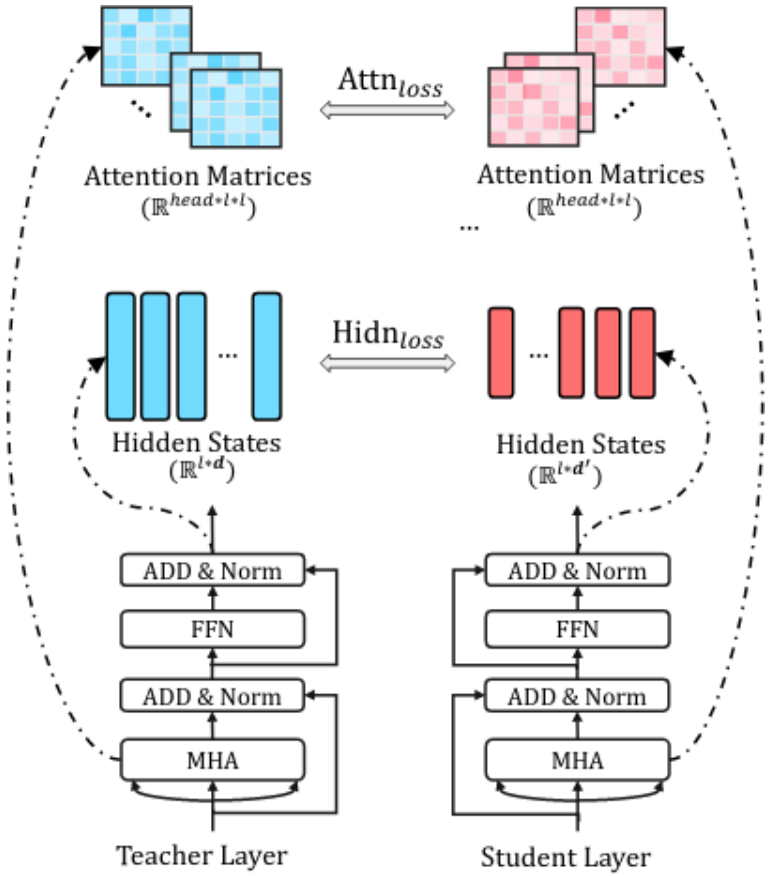
Several loss functions are designed to fit different representations from BERT layers:

- the output of the embedding layer
- the hidden states and attention matrices derived from the Transformer layer
- the logits output by the prediction layer

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

Transformer distillation



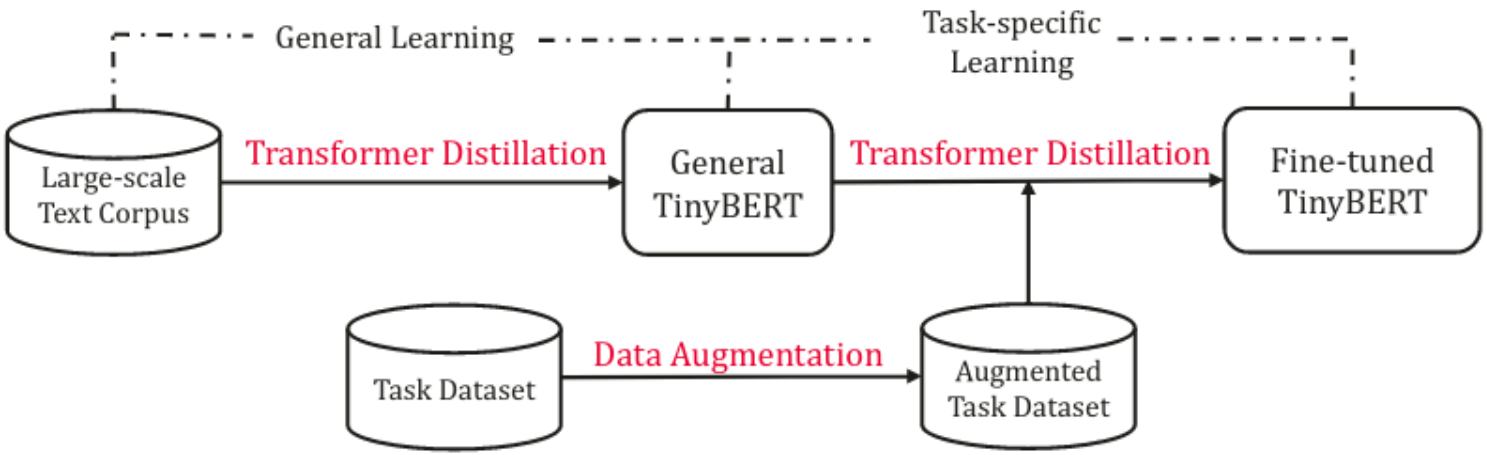


Transformer distillation

- Transformer-layer distillation include:
  - attention based distillation: $L_{attn} = \frac{1}{h}\sum_{i=1}^{h} MSE(A_i^S, A_i^T)$
  - hidden states based distillation: $L_{hidn} = MSE(H^S W_h, H^T)$
- embedding-layer distillation: $L_{embd} = MSE(E^S W_e, E^T)$
- prediction-layer distillation: soft cross-entropy loss between student and teacher **logits**

Where: h - number of attention heads, A_i - attention matrix (not *softmax* output), H^S, H^T - hidden states (FFN layer), W_h - learnable linear transformation, E^S, E^T - embeddings, W_e - learnable linear transformation

Learning illustration



GLUE results (test-set) and inference time

- GLUE results (**TinyBERT**: 4 layers and 312 hidden size):

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

- Inference time:

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)
$\text{TinyBERT/BERT}_{\text{SMALL}}$	4	312	1200	14.5M($\times$7.5)	19.9s($\times$9.4)

Ablation studies

- Two-stage learning (TD - Task-specific Distillation, GD - General Distillation, DA - Data Augmentation)

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

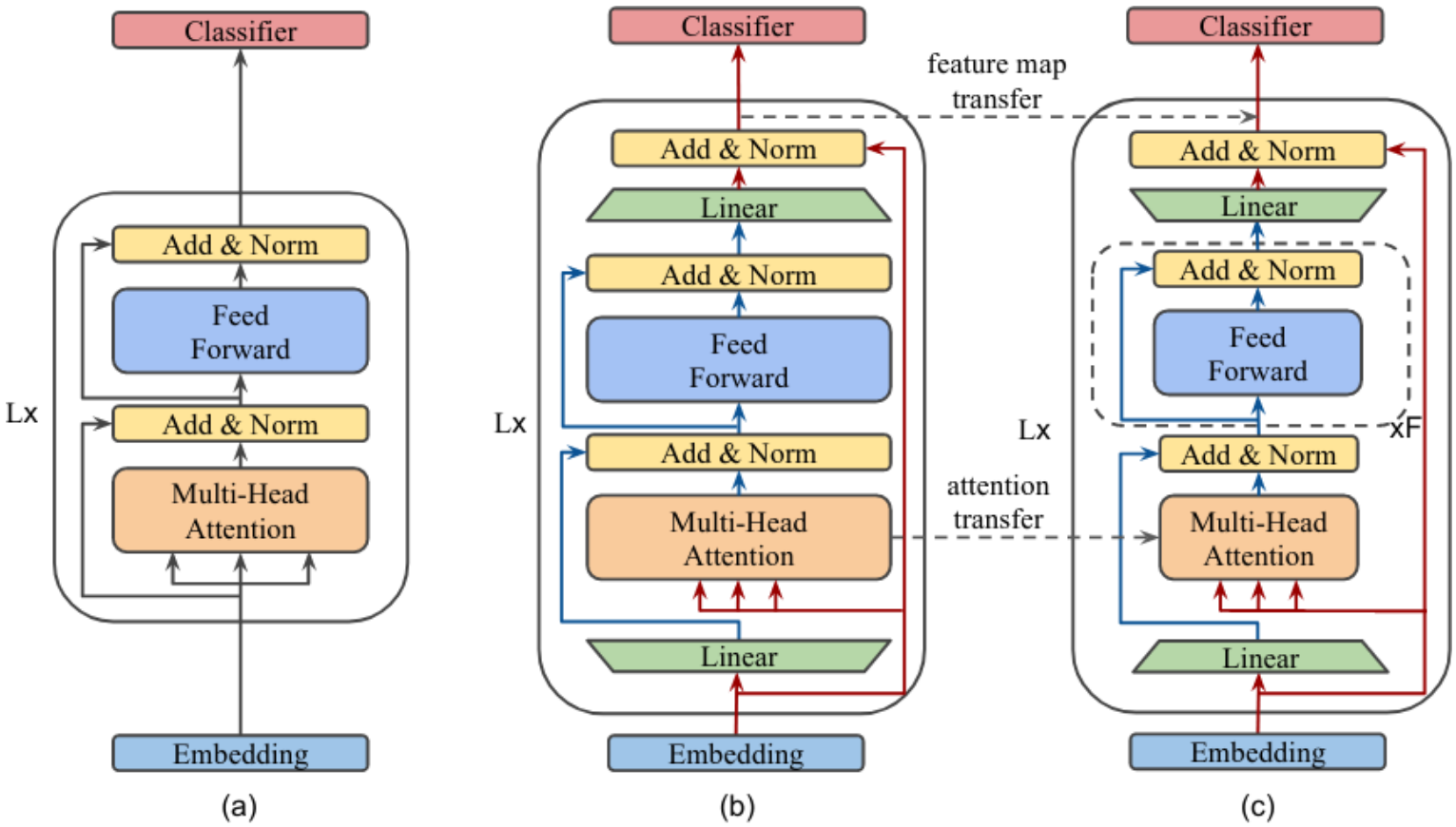
- Different distillation objectives

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

MobileBERT [3]

- April 2020, Google – code available
- MobileBERT - 4,3x smaller, 5,5x faster than BERT (base)
- based on BERT (large)
- train a specially designed teacher model - an inverted-bottleneck incorporated BERT (large) model
- progressive knowledge transfer
- task-agnostic lightweight pre-trained model (knowledge transfer in the pre-training stage)

Inverted-Bottleneck BERT (IB-BERT)



(a) BERT; (b) IB-BERT; (c) MobileBERT

Inverted-Bottleneck BERT (IB-BERT)

			$\text{BERT}_{\text{LARGE}}$	$\text{BERT}_{\text{BASE}}$	$\text{IB-BERT}_{\text{LARGE}}$	MobileBERT
embedding		$h_{\text{embedding}}$	1024	768	128	
			no-op	no-op	3-convolution	
		h_{inter}	1024	768	512	
body	Linear	h_{input} h_{output}	$\left[\begin{pmatrix}1024\\16\\1024\end{pmatrix}\right.$ $\left.\begin{pmatrix}1024\\4096\\1024\end{pmatrix}\right]\times 24$	$\left[\begin{pmatrix}768\\12\\768\end{pmatrix}\right.$ $\left.\begin{pmatrix}768\\3072\\768\end{pmatrix}\right]\times 12$	$\left[\begin{pmatrix}512\\1024\end{pmatrix}\right.$ $\begin{pmatrix}512\\4\\1024\end{pmatrix}$ $\begin{pmatrix}1024\\4096\\1024\end{pmatrix}$ $\left.\begin{pmatrix}1024\\512\end{pmatrix}\right]\times 24$	$\left[\begin{pmatrix}512\\128\end{pmatrix}\right.$ $\begin{pmatrix}512\\4\\128\end{pmatrix}$ $\begin{pmatrix}128\\512\\128\end{pmatrix}\times 4$ $\left.\begin{pmatrix}128\\512\end{pmatrix}\right]\times 24$
	MHA	h_{input} #Head h_{output}				
	FFN	h_{input} h_{FFN} h_{output}				
	Linear	h_{input} h_{output}				
#Params			334M	109M	293M	25.3M

Bottleneck and Inverted-Bottleneck

- deep as BERT (large), but each building block is made much smaller
- the hidden dimension of each building block is only 128 (for MobileBERT)
- two linear transformations for each building block to adjust its input and output dimensions to 512
- the teacher network (IB-BERT) is just BERT (large) with inverted-bottleneck structures to adjust its feature map size to 512
- IB-BERT and MobileBERT have the same feature map size which is 512 (which is used to knowledge transfer)

Other optimizations

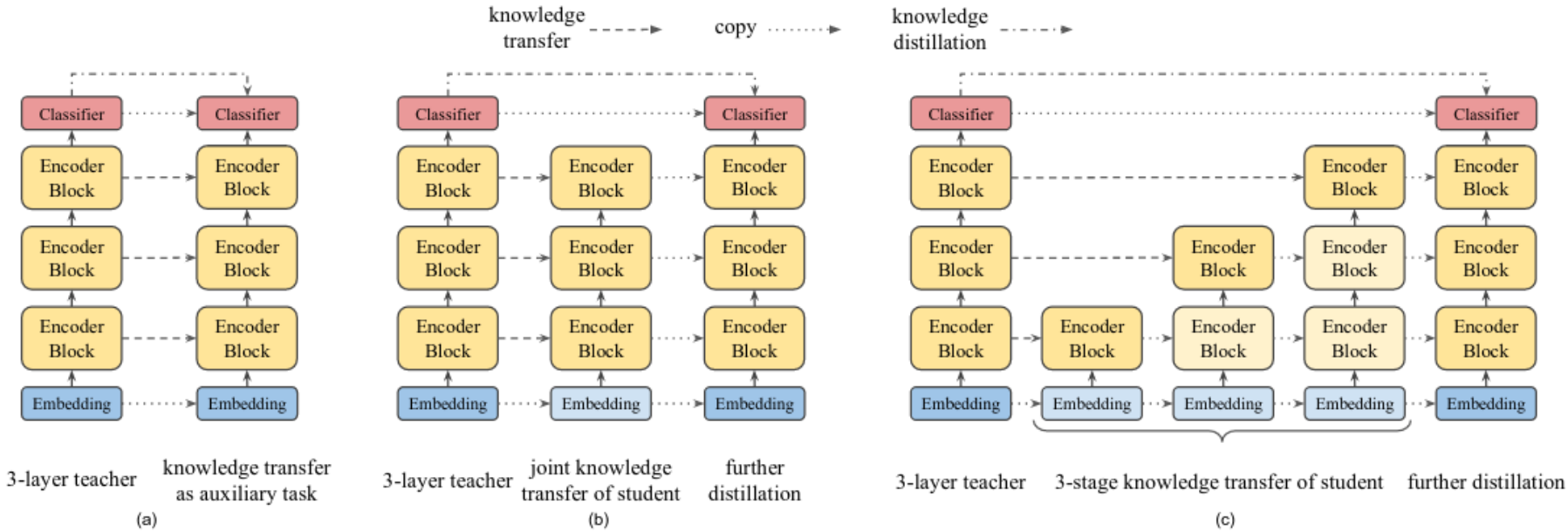
- stacked Feed-Forward Networks (to avoid more parameters)
- remove layer normalization (replace the layer normalization of a n-channel hidden state with an element-wise linear transformation)
- use ReLU activation (instead of GELU activation)
- reduce the embedding dimension to 128 and apply 1D convolution with kernel size 3 on the raw token embedding to produce a 512 dimensional output

Training Objectives

- Feature Map Transfer (FMT) - mean squared error between the feature maps: $L^l_{FMT} = \frac{1}{TN}\sum_{t=1}^{T}\sum_{n=1}^{N}(H^{tr}_{t,l,n} - H^{st}_{t,l,n})^2$
- Attention Transfer (AT) - KL-divergence between the per-head self-attention distributions : $L_{AT} = \frac{1}{TN}\sum_{t=1}^{T}\sum_{a=1}^{N} D_{KL}(attn^{tr}_{t,l,a} - attn^{st}_{t,l,a})^2$
- Pre-training Distillation (PD) - linear combination of the original masked language modeling (MLM) loss, next sentence prediction (NSP) loss, and the new MLM Knowledge Distillation (KD) loss: $L_{PD} = \alpha L_{MLM} + (1-\alpha)L_{KD} + L_{NSP}$

Where: l - index of layers, T - sequence length, N - feature map size, H - feature map, A - number of attention heads

Training Strategies



(a) Auxiliary Knowledge Transfer; (b) Joint Knowledge Transfer (first train with all layer-wise knowledge transfer losses jointly, and then further train it by pre-training distillation); (c) Progressive Knowledge Transfer (the errors from the lower layers may affect the knowledge transfer in the higher layers)

GLUE results (test-set)

- **MobileBERT**: 24 layers and 128 hidden size

	#Params	#FLOPS	Latency	CoLA 8.5k	SST-2 67k	MRPC 3.7k	STS-B 5.7k	QQP 364k	MNLI-m/mm 393k	QNLI 108k	RTE 2.5k	GLUE
ELMo-BiLSTM-Attn	-	-	-	33.6	90.4	84.4	72.3	63.1	74.1/74.5	79.8	58.9	70.0
OpenAI GPT	109M	-	-	47.2	93.1	87.7	84.8	70.1	80.7/80.6	87.2	69.1	76.9
$\text{BERT}_{\text{BASE}}$	109M	22.5B	342 ms	**52.1**	**93.5**	**88.9**	**85.8**	71.2	**84.6/83.4**	90.5	66.4	78.3
$\text{BERT}_{\text{BASE}}$-6L-PKD*	66.5M	11.3B	-	-	92.0	85.0	-	70.7	81.5/81.0	89.0	65.5	-
$\text{BERT}_{\text{BASE}}$-4L-PKD†*	52.2M	7.6B	-	24.8	89.4	82.6	79.8	70.2	79.9/79.3	85.1	62.3	-
$\text{BERT}_{\text{BASE}}$-3L-PKD*	45.3M	5.7B	-	-	87.5	80.7	-	68.1	76.7/76.3	84.7	58.2	-
$\text{DistilBERT}_{\text{BASE}}$-6L†	62.2M	11.3B	-	-	92.0	85.0		70.7	81.5/81.0	89.0	65.5	-
$\text{DistilBERT}_{\text{BASE}}$-4L†	52.2M	7.6B	-	32.8	91.4	82.4	76.1	68.5	78.9/78.0	85.2	54.1	-
TinyBERT*	14.5M	1.2B	-	43.3	92.6	86.4	79.9	**71.3**	82.5/81.8	87.7	62.9	75.4
$\text{MobileBERT}_{\text{TINY}}$	15.1M	3.1B	40 ms	46.7	91.7	87.9	80.1	68.9	81.5/81.6	89.5	65.1	75.8
MobileBERT	25.3M	5.7B	62 ms	50.5	92.8	88.8	84.4	70.2	83.3/82.6	90.6	66.2	77.7
MobileBERT w/o OPT	25.3M	5.7B	192 ms	51.1	92.6	88.8	84.8	70.5	84.3/**83.4**	**91.6**	**70.4**	**78.5**

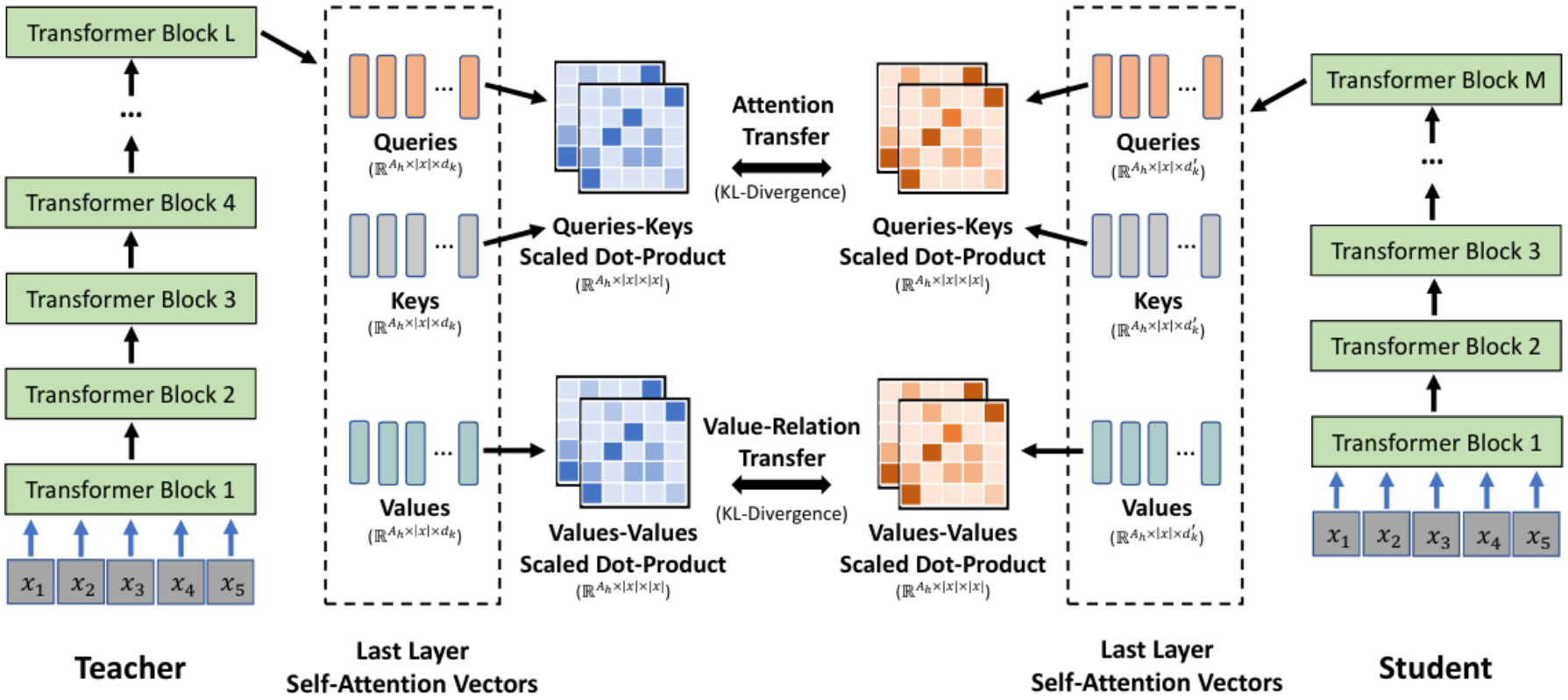
The real-world inference latency and the theoretical computation overhead (FLOPS)

Setting	#FLOPS	Latency
`LayerNorm & gelu`	5.7B	192 ms
`LayerNorm & relu`	5.7B	167 ms
`NoNorm & gelu`	5.7B	92 ms
`NoNorm & relu`	5.7B	62 ms

MiniLM [4]

- April/September 2020, Microsoft – ~~code available~~
- based on BERT (base)
- Deep Self-Attention Distillation with Teacher Assistant (TA)
  - Attention Transfer (Queries-Keys Scaled Dot-Product)
  - Value-Relation Transfer (Values-Values Scaled Dot-Product)
- distillation on the last layer (better than layer-to-layer)
- task-agnostic knowledge distillation (pre-training)

Overview of Deep Self-Attention Distillation



Deep Self-Attention Distillation

Self-Attention Distribution Transfer	Self-Attention Value-Relation Transfer
	$\mathbf{VR}^T_{L,a} = \text{softmax}(\frac{\mathbf{V}^T_{L,a}\mathbf{V}^{T\intercal}_{L,a}}{\sqrt{d_k}})$ $\mathbf{VR}^S_{M,a} = \text{softmax}(\frac{\mathbf{V}^S_{M,a}\mathbf{V}^{S\intercal}_{M,a}}{\sqrt{d'_k}})$
$\mathcal{L}_{\text{AT}} = \frac{1}{A_h\|x\|}\sum_{a=1}^{A_h}\sum_{t=1}^{\|x\|} D_{KL}(\mathbf{A}^T_{L,a,t} \parallel \mathbf{A}^S_{M,a,t})$	$\mathcal{L}_{\text{VR}} = \frac{1}{A_h\|x\|}\sum_{a=1}^{A_h}\sum_{t=1}^{\|x\|} D_{KL}(\mathbf{VR}^T_{L,a,t} \parallel \mathbf{VR}^S_{M,a,t})$

Where: D_{KL} - KL-divergence, $|x|$ - sequence length, A_h - number of attention heads, L, M - number of layers for teacher and student

Total loss
$\mathcal{L} = \mathcal{L}_{\text{AT}} + \mathcal{L}_{\text{VR}}$

Teacher Assistant (TA)

- the teacher model consists of L-layer Transformer with d_h hidden size, the student model has M-layer Transformer with d'_h hidden size (where $M \leq \frac{1}{2}L$ and $d'_h \leq \frac{1}{2}d_h$)
- Teacher Assistant procedure:
  1. distill the **teacher** into a **teacher assistant** with L-layer Transformer and d'_h hidden size
  2. distill the **assistant teacher** into the **student** with M-layer Transformer and d'_h hidden size
- brings improvements for smaller student models

Teacher Assistant (TA)

Architecture	#Param	Model	SQuAD 2.0	MNLI-m	SST-2	Average
M=6;d'_h=384	22M	MLM-KD (Soft-Label Distillation)	67.9	79.6	89.8	79.1
		TinyBERT	71.6	81.4	90.2	81.1
		MINILM	72.4	82.2	91.0	81.9
		MINILM (w/ TA)	**72.7**	**82.4**	**91.2**	**82.1**
M=4;d'_h=384	19M	MLM-KD (Soft-Label Distillation)	65.3	77.7	88.8	77.3
		TinyBERT	66.7	79.2	88.5	78.1
		MINILM	69.4	80.3	90.2	80.0
		MINILM (w/ TA)	**69.7**	**80.6**	**90.6**	**80.3**
M=3;d'_h=384	17M	MLM-KD (Soft-Label Distillation)	59.9	75.2	88.0	74.4
		TinyBERT	63.6	77.4	88.4	76.5
		MINILM	66.2	78.8	89.3	78.1
		MINILM (w/ TA)	**66.9**	**79.1**	**89.7**	**78.6**

GLUE results (dev-set) and inference time

- GLUE results (**MiniLM**: 6 layers and 768 hidden size):

Model	#Param	SQuAD2	MNLI-m	SST-2	QNLI	CoLA	RTE	MRPC	QQP	Average
$\text{BERT}_{\text{BASE}}$	109M	76.8	84.5	93.2	91.7	58.9	68.6	87.3	91.3	81.5
DistillBERT	66M	70.7	79.0	90.7	85.3	43.6	59.9	87.5	84.9	75.2
TinyBERT	66M	73.1	83.5	91.6	90.5	42.8	**72.2**	**88.4**	90.6	79.1
MINILM	66M	**76.4**	**84.0**	**92.0**	**91.0**	**49.2**	71.5	**88.4**	**91.0**	**80.4**

- Inference time (Emd - number of parameters Embedding, Trm - Transformer parameter):

#Layers	Hidden Size	#Param (Emd)	#Param (Trm)	Inference Time
12	768	23.4M	85.1M	93.1s (1.0×)
6	768	23.4M	42.5M	46.9s (2.0×)
12	384	11.7M	21.3M	34.8s (2.7×)
6	384	11.7M	10.6M	17.7s (5.3×)
4	384	11.7M	7.1M	12.0s (7.8×)
3	384	11.7M	5.3M	9.2s (10.1×)

Distillation approaches

Approach	Teacher Model	Distilled Knowledge	Layer-to-Layer Distillation	Requirements on the number of layers of students	Requirements on the hidden size of students
DistillBERT	$\text{BERT}_{\text{BASE}}$	Soft target probabilities Embedding outputs			✓
TinyBERT	$\text{BERT}_{\text{BASE}}$	Embedding outputs Hidden states Self-Attention distributions	✓		
MOBILEBERT	$\text{IB-BERT}_{\text{LARGE}}$	Soft target probabilities Hidden states Self-Attention distributions	✓	✓	✓
MINILM	$\text{BERT}_{\text{BASE}}$	Self-Attention distributions Self-Attention value relation			

References I

[1] V. Sanh, L. Debut, J. Chaumond, and T. Wold, “Distilbert, a distilled version of bert: smaller,faster, cheaper and lighter,” 2019.

[2] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, “Tinybert: Distilling bert for natural language understanding,” 2019.

[3] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, “Mobilebert: a compact task-agnostic bert for resource-limited devices,” 2020.

[4] W. Wan, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, “Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers,” 2020.