CPSC 677 Advanced NLP

Efficient Transformer Models

Zhiyu Liang

zhiyu.liang@yale.edu

Nov 30, 2021

Agenda

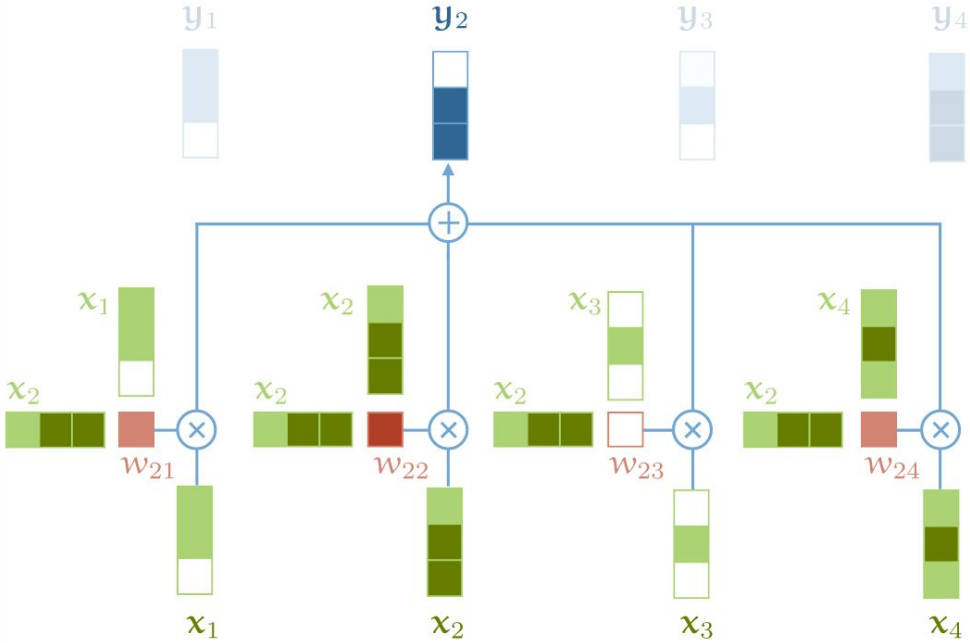
- Review of the transformer model architecture
- Motivation
- Common model compression techniques
- Relevant natural language understanding tasks
- Efficient transformer model variants
- Discussion

Transformer Model Architecture

Self-Attention

Intuition for W_{ij}:

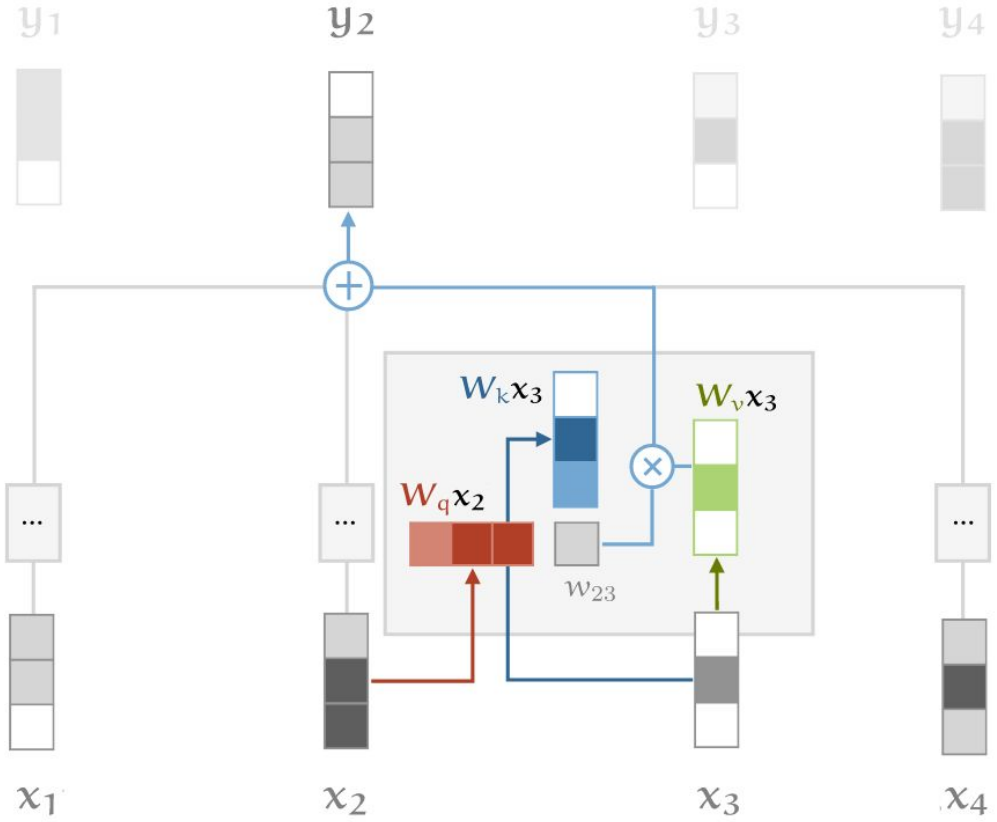
- Dot-product measures cosine of the angle between 2 vectors
- Sign (+, -) tells whether 2 words are positively or negatively related
- Magnitude tells how strong such correlation is



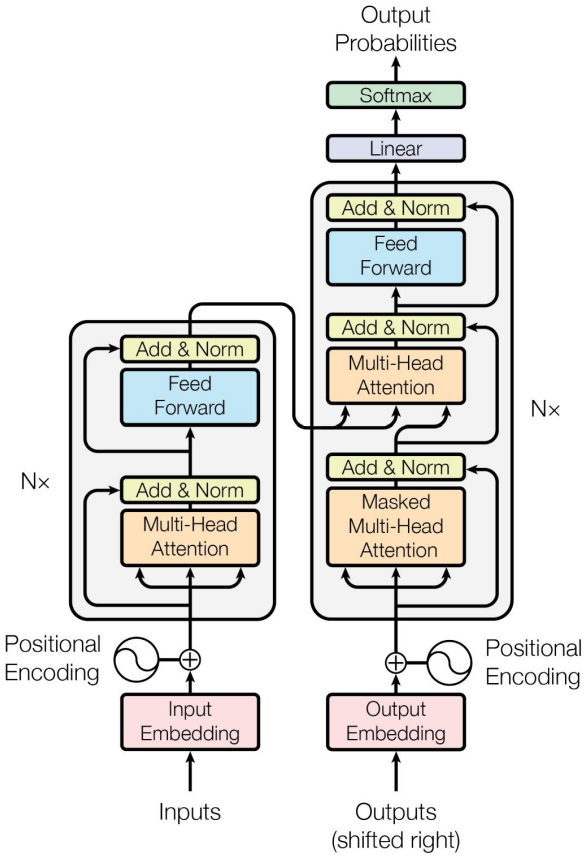
Source: Bloem

Scaled Dot-Product Attention

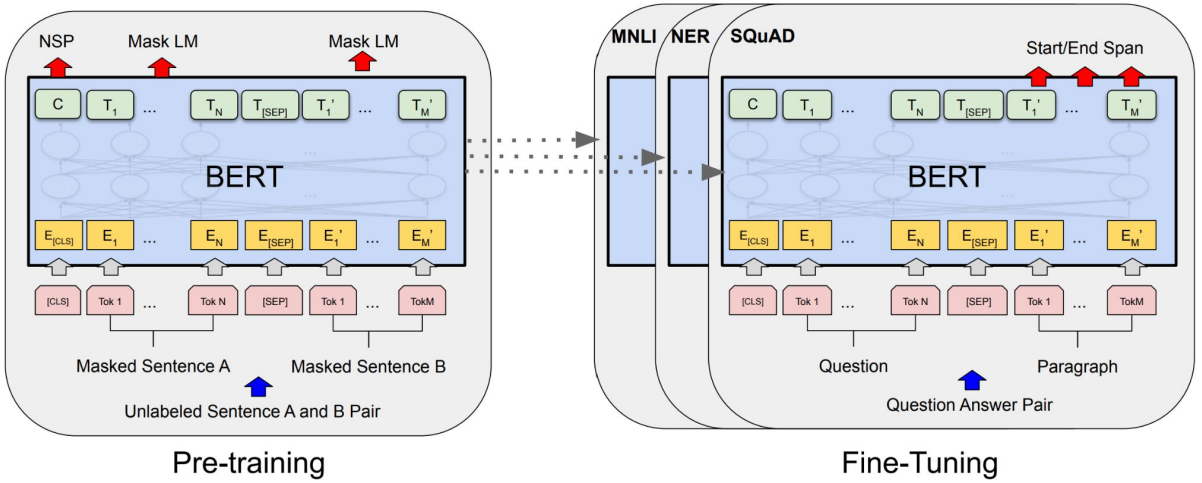
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Source: Bloem



[Transformer]



[BERT] (Transformer Encoder)

Motivation

How well do Transformer models perform?

- The General Language Understanding Evaluation (GLUE) benchmark [5]
- SuperGLUE [6]

GLUE Benchmark

- **CoLA** (Corpus of Linguistic Acceptability)
  - Determine if a sentence is grammatically correct or not.
- **MNLI** (Multi-Genre Natural Language Inference)
  - Determine if a sentence entails, contradicts or is unrelated to a given hypothesis.
- **MRPC** (Microsoft Research Paraphrase Corpus)
  - Determine if two sentences are paraphrases from one another or not.
- **QNLI** (Question-answering Natural Language Inference)
  - Determine if the answer to a question is in the second sentence or not.
- **QQP** (Quora Question Pairs2)
  - Determine if two questions are semantically equivalent or not.

GLUE Benchmark

- **RTE** (Recognizing Textual Entailment)
  - Determine if a sentence entails a given hypothesis or not.
- **SST-2** (Stanford Sentiment Treebank)
  - Determine if the sentence has a positive or negative sentiment.
- **STS-B** (Semantic Textual Similarity Benchmark)
  - Determine the similarity of two sentences with a score from 1 to 5.
- **WNLI** (Winograd Natural Language Inference)
  - Determine if a sentence with an anonymous pronoun and a sentence with this pronoun replaced are entailed or not.

GLUE Benchmark

Subset: cola | Split: train

sentence (string)	label (class label)	idx (int)
They made him president.	acceptable	38
They made him angry.	acceptable	39
They caused him to become angry by making him.	unacceptable	40

Subset: mnli | Split: train

premise (string)	hypothesis (string)	label (class label)	idx (int)
How do we fix this?'	Can we fix this?	entailment	51
but that takes too much planning	It doesn't take much planning.	contradiction	52

Subset: mrpc | Split: train

sentence1 (string)	sentence2 (string)	label (class label)	idx (int)
Amrozi accused his brother , whom he called " the witness " , of…	Referring to him as only " the witness " , Amrozi accused his…	equivalent	0
Yucaipa owned Dominick 's before selling the chain to Safeway in…	Yucaipa bought Dominick 's in 1995 for $ 693 million and sold…	not_equivalent	1
They had published an advertisement on the Internet on…	On June 10 , the ship 's owners had published an advertisement o…	equivalent	2

Subset: qnli | Split: train

question (string)	sentence (string)	label (class label)	idx (int)
How were the Portuguese expelled from Myanmar?	From the 1720s onward, the kingdom was beset with repeated…	not_entailment	7
What does the word 'customer' properly apply to?	The bill also required rotation of principal maintenance…	entailment	8

SuperGLUE Benchmark

BoolQ	**Passage:** *Barq's – Barq's is an American soft drink. Its brand of root beer is notable for having caffeine. Barq's, created by Edward Barq and bottled since the turn of the 20th century, is owned by the Barq family but bottled by the Coca-Cola Company. It was known as Barq's Famous Olde Tyme Root Beer until 2012.* **Question:** *is barq's root beer a pepsi product* **Answer:** `No`
CB	**Text:** *B: And yet, uh, I we-, I hope to see employer based, you know, helping out. You know, child, uh, care centers at the place of employment and things like that, that will help out. A: Uh-huh. B: What do you think, do you think we are, setting a trend?* **Hypothesis:** *they are setting a trend* **Entailment:** `Unknown`
COPA	**Premise:** *My body cast a shadow over the grass.* **Question:** *What's the CAUSE for this?* **Alternative 1:** *The sun was rising.* **Alternative 2:** *The grass was cut.* **Correct Alternative:** `1`
MultiRC	**Paragraph:** *Susan wanted to have a birthday party. She called all of her friends. She has five friends. Her mom said that Susan can invite them all to the party. Her first friend could not go to the party because she was sick. Her second friend was going out of town. Her third friend was not so sure if her parents would let her. The fourth friend said maybe. The fifth friend could go to the party for sure. Susan was a little sad. On the day of the party, all five friends showed up. Each friend had a present for Susan. Susan was happy and sent each friend a thank you card the next week* **Question:** *Did Susan's sick friend recover?* **Candidate answers:** *Yes, she recovered* (`T`), *No* (`F`), *Yes* (`T`), *No, she didn't recover* (`F`), *Yes, she was at Susan's party* (`T`)
ReCoRD	**Paragraph:** *(CNN) Puerto Rico on Sunday overwhelmingly voted for statehood. But Congress, the only body that can approve new states, will ultimately decide whether the status of the US commonwealth changes. Ninety-seven percent of the votes in the nonbinding referendum favored statehood, an increase over the results of a 2012 referendum, official results from the State Electorcal Commission show. It was the fifth such vote on statehood. "Today, we the people of Puerto Rico are sending a strong and clear message to the US Congress ... and to the world ... claiming our equal rights as American citizens, Puerto Rico Gov. Ricardo Rossello said in a news release. @highlight Puerto Rico voted Sunday in favor of US statehood* **Query** For one, they can truthfully say, "Don't blame me, I didn't vote for them, " when discussing the <placeholder> presidency **Correct Entities:** `US`
RTE	**Text:** *Dana Reeve, the widow of the actor Christopher Reeve, has died of lung cancer at age 44, according to the Christopher Reeve Foundation.* **Hypothesis:** *Christopher Reeve had an accident.* **Entailment:** `False`
WiC	**Context 1:** *Room and board.* **Context 2:** *He nailed boards across the windows.* **Sense match:** `False`
WSC	**Text:** *Mark told Pete many lies about himself, which Pete included in his book. He should have been more truthful.* **Coreference:** `False`

SuperGLUE Benchmark

- **BoolQ** (Natural Yes/No Questions)
  - Question Answering dataset with 15942 yes/no questions
- **CB** (Commitment Treebank)
  - Determine if the hypothesis commits to the truth
- COPA (Choice of Plausible Alternatives)
  -

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
1	ERNIE Team - Baidu	ERNIE		91.1	75.5	97.8	93.9/91.8	93.0/92.6	75.2/90.9	92.3	91.7	97.3	92.6	95.9	51.7
2	AliceMind & DIRL	StructBERT + CLEVER		91.0	75.3	97.7	93.9/91.9	93.5/93.1	75.6/90.8	91.7	91.5	97.4	92.5	95.2	49.1
3	liangzhu ge	DEBERTa + CLEVER		90.9	73.9	97.5	92.8/90.4	93.2/92.9	76.4/90.9	92.1	91.7	96.7	93.1	96.6	35.2
4	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.8	71.5	97.5	94.0/92.0	92.9/92.6	76.2/90.8	91.9	91.6	99.2	93.2	94.5	53.2
5	HFL iFLYTEK	MacALBERT + DKM		90.7	74.8	97.0	94.5/92.6	92.8/92.6	74.7/90.6	91.3	91.1	97.8	92.0	94.5	52.6
6	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6	73.5	97.2	94.0/92.0	93.0/92.4	76.1/91.0	91.6	91.3	97.5	91.7	94.5	51.2
7	T5 Team - Google	T5		90.3	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	92.2	91.9	96.9	92.8	94.5	53.1
8	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
9	Huawei Noah's Ark Lab	NEZHA-Large		89.8	71.7	97.3	93.3/91.0	92.4/91.9	75.2/90.7	91.5	91.3	96.2	90.3	94.5	47.9
10	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)		89.7	70.5	97.5	93.4/91.2	92.6/92.3	75.4/90.7	91.4	91.1	95.8	90.0	94.5	51.6
11	ELECTRA Team	ELECTRA-Large + Standard Tricks		89.4	71.7	97.1	93.1/90.7	92.9/92.5	75.6/90.8	91.3	90.8	95.8	89.8	91.8	50.7
12	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.4	68.0	96.8	93.1/90.8	92.3/92.1	74.8/90.3	91.1	90.7	95.6	88.7	89.0	50.1
13	Junjie Yang	HIRE-RoBERTa		88.3	68.6	97.1	93.0/90.7	92.4/92.0	74.3/90.2	90.7	90.4	95.5	87.9	89.0	49.3
14	Facebook AI	RoBERTa		88.1	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	95.4	88.2	89.0	48.7
15	Microsoft D365 AI & MSR AI	MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8
16	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-

Rank	Name	Model	URL	Score	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	AX-b	AX-g
1	ERNIE Team - Baidu	ERNIE 3.0		90.6	91.0	98.6/99.2	97.4	88.6/63.2	94.7/94.2	92.6	77.4	97.3	68.6	92.7/94.7
2	Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4	91.4	95.8/97.6	98.0	88.3/63.0	94.2/93.5	93.0	77.9	96.6	69.1	92.7/91.9
3	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3	90.4	95.7/97.6	98.4	88.2/63.7	94.5/94.1	93.2	77.5	95.9	66.7	93.3/93.8
4	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	76.6	99.3/99.7
5	T5 Team - Google	T5		89.3	91.2	93.9/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8	65.6	92.7/91.9
6	Huawei Noah's Ark Lab	NEZHA-Plus		86.7	87.8	94.4/96.0	93.6	84.6/55.1	90.1/89.6	89.1	74.6	93.2	58.0	87.1/74.4
7	Alibaba PAI&ICBU	PAI Albert		86.1	88.1	92.4/96.4	91.8	84.6/54.7	89.0/88.3	88.8	74.1	93.2	75.6	98.3/99.2
8	Infosys : DAWN : AI Research	RoBERTa-iCETS		86.0	88.5	93.2/95.2	91.2	86.4/58.2	89.9/89.3	89.9	72.9	89.0	61.8	88.8/81.5
9	Tencent Jarvis Lab	RoBERTa (ensemble)		85.9	88.2	92.5/95.6	90.8	84.4/53.4	91.5/91.0	87.9	74.1	91.8	57.6	89.3/75.6
10	Zhuiyi Technology	RoBERTa-mtl-adv		85.7	87.1	92.4/95.6	91.2	85.1/54.3	91.7/91.3	88.1	72.1	91.8	58.5	91.0/78.1
11	Facebook AI	RoBERTa		84.6	87.1	90.5/95.2	90.6	84.4/52.5	90.6/90.0	88.2	69.9	89.0	57.9	91.0/78.1
12	Anuar Sharafudinov	AILabs Team, Transformers		82.6	88.1	91.6/94.8	86.8	85.1/54.7	82.8/79.8	88.9	74.1	78.8	100.0	100.0/100.0
13	Rakesh Radhakrishnan Menon	ADAPET (ALBERT) - few-shot		76.0	80.0	82.3/92.0	85.4	76.2/35.7	86.1/85.5	75.0	53.5	85.6	-0.4	100.0/50.0
14	Timo Schick	iPET (ALBERT) - Few-Shot (32 Examples)		75.4	81.2	79.9/88.8	90.8	74.1/31.7	85.9/85.4	70.8	49.3	88.4	36.2	97.8/57.9
15	Adrian de Wynter	Bort (Alexa AI)		74.1	83.7	81.9/86.4	89.6	83.7/54.1	49.8/49.0	81.2	70.1	65.8	48.0	96.1/61.5
16	IBM Research AI	BERT-mtl		73.5	84.8	89.6/94.0	73.8	73.2/30.5	74.6/74.0	84.1	66.2	61.0	29.6	97.8/57.3
17	Ben Mann	GPT-3 few-shot - OpenAI		71.8	76.4	52.0/75.6	92.0	75.4/30.5	91.1/90.2	69.0	49.4	80.1	21.1	90.4/55.3
18	SuperGLUE Baselines	BERT++		71.5	79.0	84.8/90.4	73.8	70.0/24.1	72.0/71.3	79.0	69.6	64.4	38.0	99.4/51.4
		BERT		69.0	77.4	75.7/83.6	70.6	70.0/24.1	72.0/71.3	71.7	69.6	64.4	23.0	97.8/51.7

What are the costs?

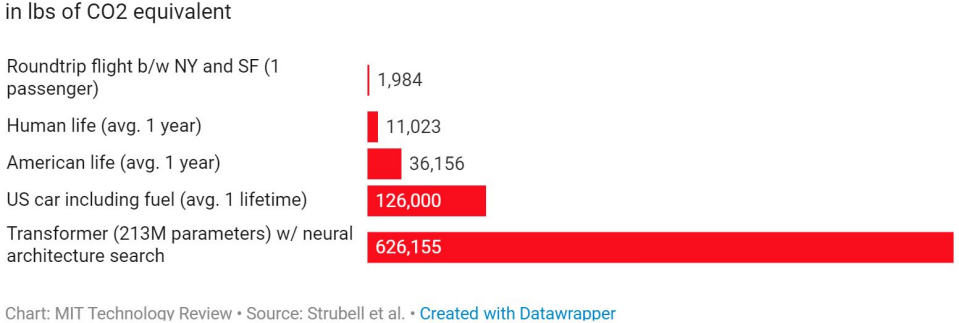
Model Size

- Using the largest version in each paper

	Transformer	BERT	RoBERTa	GPT-2	GPT-3	Megatron-Turing NLG
Model Size	213M	336M	355M	1.5B	175B	530B

	AlexNet	ResNet-101
Model Size	61M	44.5M

Estimated Training Costs



	Date of original paper	Energy consumption (kWh)	Carbon footprint (lbs of CO2e)	Cloud compute cost (USD)
Transformer (65M parameters)	Jun, 2017	27	26	$41-$140
Transformer (213M parameters)	Jun, 2017	201	192	$289-$981
ELMo	Feb, 2018	275	262	$433-$1,472
BERT (110M parameters)	Oct, 2018	1,507	1,438	$3,751-$12,571
Transformer (213M parameters) w/ neural architecture search	Jan, 2019	656,347	626,155	$942,973-$3,201,722
GPT-2	Feb, 2019	-	-	$12,902-$43,008

Note: Because of a lack of power draw data on GPT-2's training hardware, the researchers weren't able to calculate its carbon footprint.

Table: MIT Technology Review • Source: Strubell et al. • Created with Datawrapper

Source: Hao

Discussion

- Should researchers be concerned with the practicality / costs when building more powerful models? Why?

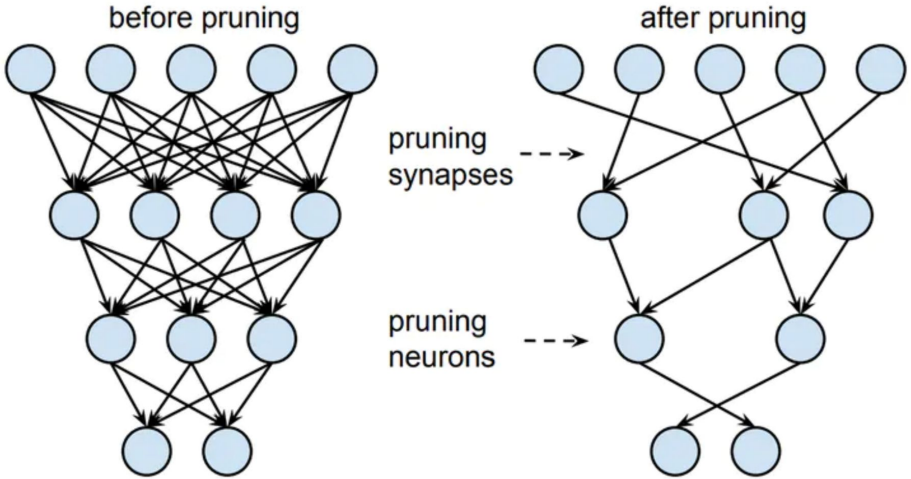
- What are some directions to mitigate the practicality issue in general?

Common Model Compression Techniques

Pruning

Basic Idea:

- Converged models usually have a large number of weights very close to 0
- Those weights do not affect model performance significantly



Source: Bilogur

The Lottery Ticket Hypothesis (Frankle, et al)

- A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations. [10]
- Usually a winning ticket is only 10-20% of the original dense network.

The Lottery Ticket Hypothesis

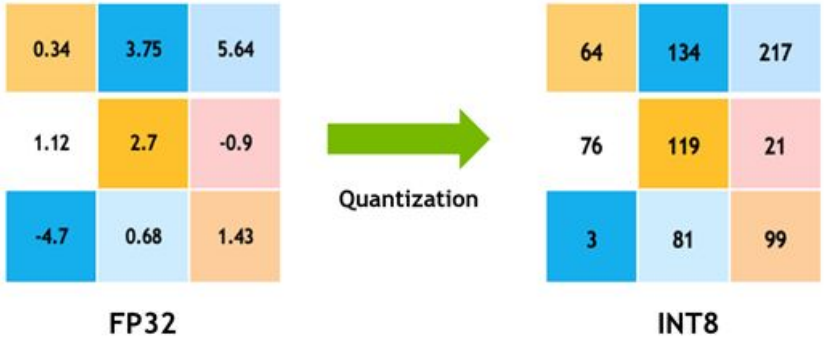
Pruning Procedures:

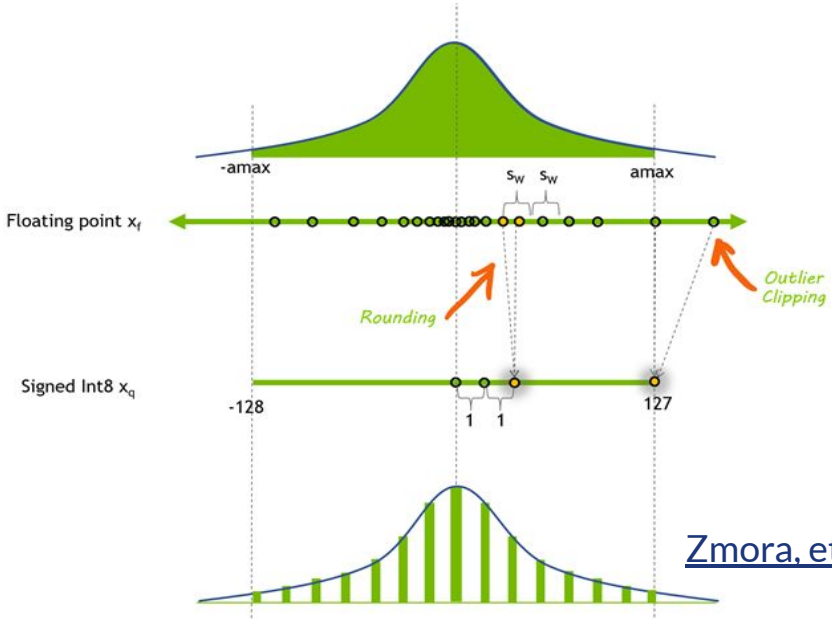
1. Randomly initialize a neural network and train for some number of iterations.
2. Prune *p*% of parameters, creating a mask *m*.
3. Reset the remaining parameters back to random while fixing all masked parameters to 0.
4. Train the winning ticket for some iterations again, repeat 2 and 3 until reaching the desired compression ratio

Quantization

Basic Idea:

- Convert model parameters and activations from floating point to lower-precision representation.
- E.g.
  - fp 32 -> int 8 reduces model size by 4x





Zmora, et al

Quantization Method

User-specified inputs:

- bit width: number of bits used to represent each number
- x_min: the minimum value to capture
- x_max: the maximum value to capture

Formula:

$$step\ size = \frac{x_{max} - x_{min}}{2^{bw-1}}$$

$$x_{offset} = \frac{x_{min}}{step\ size}$$

$$x_{quant} = round\left(\frac{x}{step\ size} - x_{offset}\right)$$

$$x_{dequant} = step\ size\ * (x_{quant} + x_{offset})$$

Quantization Schemes

Post-Training Quantization (PTQ)

- Quantize the weights and activations of a fully trained floating point model
- Collect the quantization statistics with a small portion of data
- No fine-tuning
- PTQ works well in most cases with a few exceptions

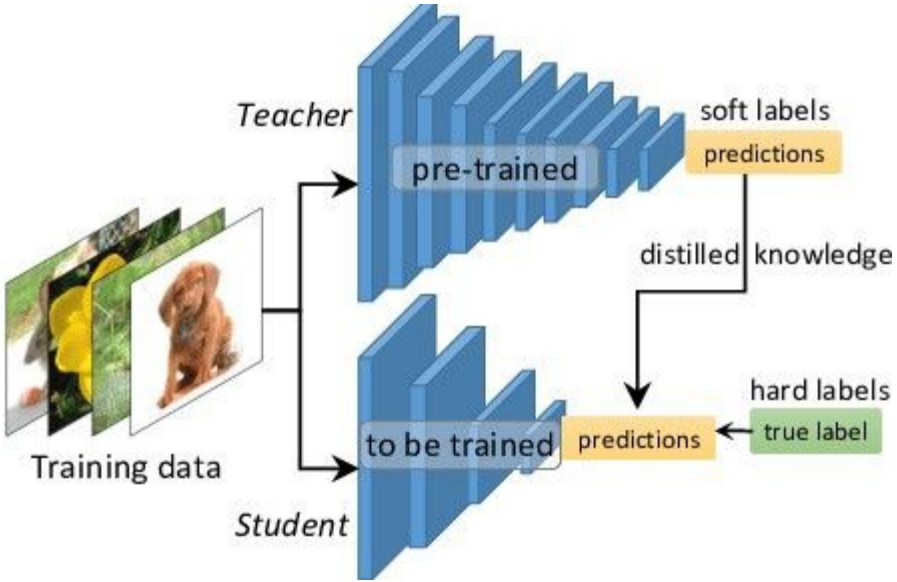
Quantization-Aware Training (QAT)

- Simulate fixed-point training by inserting fake quant nodes to the floating point model during training
- Better weight distribution since quantization loss is part of the learning objective
- Expensive

Knowledge Distillation

Basic Idea:

- Train a small (student) network to match the output distribution of a pre-trained large (teacher) network.
- The output of the teacher model contains inter-class relationships



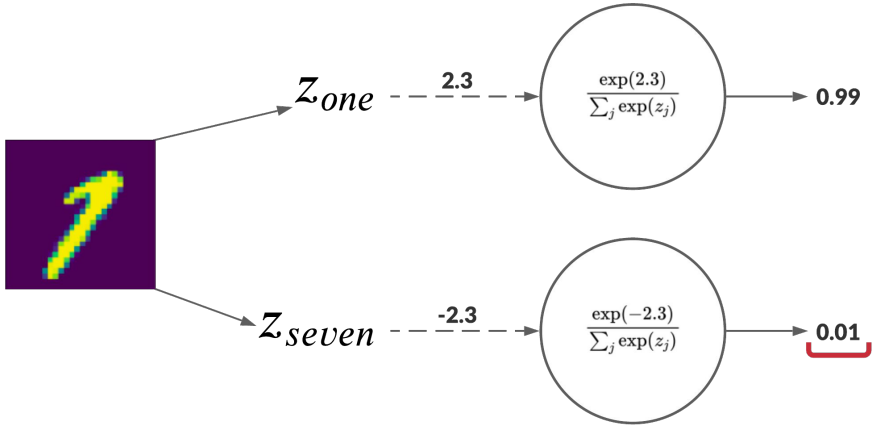
Source: Ganesh

Knowledge Distillation

Softmax Function

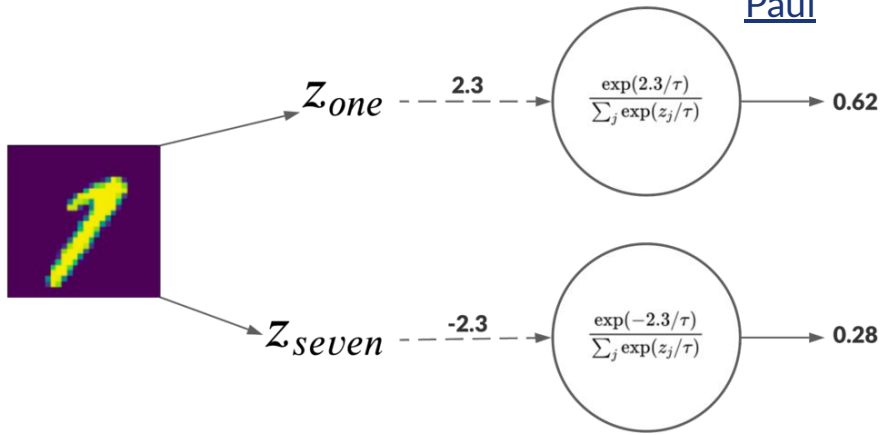
$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

- z_i is the logit for a particular class
- T is the temperature controlling the "softness" of the output distribution



$\tau = 5$

Source: Paul



Knowledge Distillation Procedures

1. Train the teacher model as usual
2. Define the student model loss function
    a. Cross Entropy Loss with the ground truth hard labels
    b. Cross Entropy Loss with the soft labels generated by the teacher model
        i. Alternatively, KL Divergence or MSE on the logits
3. Train the student model

Pruning

Quantization

Knowledge
Distillation

Source: Sučík

Discussion

- What is the ideal situation to apply each compression technique?

- What are some tradeoffs for each method?

Neural Architecture Search

Relevant Natural Language Understanding Tasks

Language Modeling

The use of various statistical and probabilistic techniques to determine the probability of a given sequence of words occurring in a sentence. [15]

Relevant datasets:

- WikiText-103
  - Verified high quality Wikipedia articles of over 100M tokens.
- LM1B
  - WMT 2011 News Crawl data

Text Classification

- GLUE Benchmark
- SuperGLUE Benchmark

Text Classification

GLUE Benchmark [16]

- **CoLA** (Corpus of Linguistic Acceptability)
  - Determine if a sentence is grammatically correct or not.
- **MNLI** (Multi-Genre Natural Language Inference)
  - Determine if a sentence entails, contradicts or is unrelated to a given hypothesis.
- **MRPC** (Microsoft Research Paraphrase Corpus)
  - Determine if two sentences are paraphrases from one another or not.
- **QNLI** (Question-answering Natural Language Inference)
  - Determine if the answer to a question is in the second sentence or not.
- **QQP** (Quora Question Pairs2)
  - Determine if two questions are semantically equivalent or not.

Text Classification

GLUE Benchmark Cont.

- **RTE** (Recognizing Textual Entailment)
  - Determine if a sentence entails a given hypothesis or not.
- **SST-2** (Stanford Sentiment Treebank)
  - Determine if the sentence has a positive or negative sentiment.
- **STS-B** (Semantic Textual Similarity Benchmark)
  - Determine the similarity of two sentences with a score from 1 to 5.
- **WNLI** (Winograd Natural Language Inference)
  - Determine if a sentence with an anonymous pronoun and a sentence with this pronoun replaced are entailed or not.

Abstractive Summarization

Generate a short and concise summary that captures the salient ideas of the source text. The generated summaries potentially contain new phrases and sentences that may not appear in the source text. [17]

Relevant dataset:

- CNN / Daily Mail [18]
    - Given the CNN and Daily Mail news stories and human generated summaries with one of the entities hidden, answer the fill-in-the-blank question.
    - 286,817 training pairs, 13,368 validation pairs and 11,487 test pairs.

Question Answering

Answer questions (typically reading comprehension questions), but abstain when the question cannot be answered based on the provided context. [19]

Relevant Datasets:

- SQuAD 1.1 (The Stanford Question Answering Dataset) [20]
- SQuAD 2.0
    - Reading comprehension dataset with Wikipedia articles
- RACE (ReAding Comprehension dataset from Examinations) [21]
    - 27,993 passages and 97,867 questions from English exams, targeting Chinese students aged 12-18

Machine Translation

Translating a sentence in a source language to a different target language. [22]

Relevant datasets:

- WMT (English to German / French)
  - Machine translation dataset from the Ninth Workshop on Statistical Machine Translation
  - 4.5M training sentence pairs
- IWSLT (The International Workshop on Spoken Language Translation) German-English
  - 160K training sentence pairs

Paper 1

ALBERT: A Lite BERT for Self-supervised Learning of Language Representations [Lan et al, 2020]

Overview

- Parameter-reduction techniques to lower memory consumption and increase training speed of BERT.
- Achieved BERT Base performance with 6x less number of parameters.
- Achieved BERT Large performance with 5.6x less number of parameters.
- Significantly outperformed other BERT variants with similar number of parameters on NLU tasks.

Factorized Embedding Parameterization

Motivation

- Embedding Layer: learn *context-independent* representations
- Hidden Layers: learn *context-dependent* representations
- BERT and its previous variants:
  - Embedding size E = Hidden layer size H
  - Does it make sense?

Factorized Embedding Parameterization

- Untie the embedding size E and hidden size H by using E << H
- Reduce embedding parameters from O(V × H) to O(V × E + E × H)

Model	E	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base not-shared	64	87M	89.9/82.9	80.1/77.8	82.9	91.5	66.7	81.3
	128	89M	89.9/82.8	80.3/77.3	83.7	91.5	67.9	81.7
	256	93M	90.2/83.2	80.3/77.4	84.1	91.9	67.3	81.8
	768	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base all-shared	64	10M	88.7/81.4	77.5/74.8	80.8	89.4	63.5	79.0
	128	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1
	256	16M	88.8/81.5	79.1/76.3	81.5	90.3	63.4	79.6
	768	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8

Table 3: The effect of vocabulary embedding size on the performance of ALBERT-base.

Cross-layer Parameter Sharing

- Reuse the same parameters across different layers.
- Attention layers
- FFN layers

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base E=768	all-shared	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8
	shared-attention	83M	89.9/82.7	80.0/77.2	84.0	91.4	67.7	81.6
	shared-FFN	57M	89.2/82.1	78.2/75.4	81.5	90.8	62.6	79.5
	not-shared	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base E=128	all-shared	12M	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1
	shared-attention	64M	89.9/82.8	80.7/77.9	83.4	91.9	67.6	81.7
	shared-FFN	38M	88.9/81.6	78.6/75.6	82.3	91.7	64.4	80.2
	not-shared	89M	89.9/82.8	80.3/77.3	83.2	91.5	67.9	81.6

Table 4: The effect of cross-layer parameter-sharing strategies, ALBERT-base configuration.

Inter-sentence Coherence Loss

- Replace the Next Sentence Prediction (NSP) loss in BERT pretraining with the Sentence Order Prediction (SOP) loss.
  - NSP coupled *topic prediction* and *coherence prediction* in a single task, where the former is easier to learn.
  - Positive examples are the consecutive segments from the same document.
  - Negative examples are the same 2 consecutive segments with their order reversed.

	Intrinsic Tasks			Downstream Tasks					
SP tasks	MLM	NSP	SOP	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
None	54.9	52.4	53.3	88.6/81.5	78.1/75.3	81.5	89.9	61.7	79.0
NSP	54.5	90.5	52.0	88.4/81.5	77.2/74.6	81.6	**91.1**	62.3	79.2
SOP	54.0	78.9	86.5	**89.3/82.3**	**80.0/77.1**	**82.0**	90.3	**64.0**	**80.1**

Table 5: The effect of sentence-prediction loss, NSP vs. SOP, on intrinsic and downstream tasks.

Comparison

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
Single-task single models on dev										
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa-large	90.2	94.7	**92.2**	86.6	96.4	**90.9**	68.0	92.4	-	-
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7	-	-
ALBERT (1.5M)	**90.8**	**95.3**	**92.2**	**89.2**	**96.9**	**90.9**	**71.4**	**93.0**	-	-
Ensembles on test (from leaderboard as of Sept. 16, 2019)										
ALICE	88.2	95.7	**90.7**	83.5	95.2	92.6	**69.2**	91.1	80.8	87.0
MT-DNN	87.9	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5
Adv-RoBERTa	91.1	98.8	90.3	88.7	96.8	93.1	68.0	92.4	89.0	88.8
ALBERT	**91.3**	**99.2**	90.5	**89.2**	**97.1**	**93.4**	69.1	**92.5**	**91.8**	**89.4**

Table 9: State-of-the-art results on the GLUE benchmark. For single-task single-model results, we report ALBERT at 1M steps (comparable to RoBERTa) and at 1.5M steps. The ALBERT ensemble uses models trained with 1M, 1.5M, and other numbers of steps.

Comparison

Models	SQuAD1.1 dev	SQuAD2.0 dev	SQuAD2.0 test	RACE test (Middle/High)
Single model (from leaderboard as of Sept. 23, 2019)				
BERT-large	90.9/84.1	81.8/79.0	89.1/86.3	72.0 (76.6/70.1)
XLNet	94.5/89.0	88.8/86.1	89.1/86.3	81.8 (85.5/80.2)
RoBERTa	94.6/88.9	89.4/86.5	89.8/86.8	83.2 (86.5/81.3)
UPM	-	-	89.9/87.2	-
XLNet + SG-Net Verifier++	-	-	90.1/87.2	-
ALBERT (1M)	94.8/89.2	89.9/87.2	-	86.0 (88.2/85.1)
ALBERT (1.5M)	**94.8/89.3**	**90.2/87.4**	**90.9/88.1**	**86.5 (89.0/85.5)**
Ensembles (from leaderboard as of Sept. 23, 2019)				
BERT-large	92.2/86.2	-	-	-
XLNet + SG-Net Verifier	-	-	90.7/88.2	-
UPM	-	-	90.7/88.2	
XLNet + DAAF + Verifier	-	-	90.9/88.6	-
DCMN+	-	-	-	84.1 (88.5/82.3)
ALBERT	**95.5/90.1**	**91.4/88.9**	**92.2/89.7**	**89.4 (91.2/88.6)**

Table 10: State-of-the-art results on the SQuAD and RACE benchmarks.

Paper 2

MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices [Sun et al, 2020]

Overview

- MobileBERT is 4.3x smaller and 5.5x faster than BERT_base while achieving similar performance
- Model architecture optimizations
- Operational optimizations
- Knowledge transfer

Model Architecture Optimizations

- **Embedding Factorization**
  - Reduced the embedding dimension to 128
  - Applied 1D convolution to produce 512 dimensional output
- **Bottleneck and Inverted-Bottleneck**
  - Added linear transformations for each block to adjust its input and output dimension to 512
- **Stacked Feed-Forward Networks**
  - To achieve similar parameter count ratio between MHA and FFN as in BERT, stack multiple FFN after each MHA

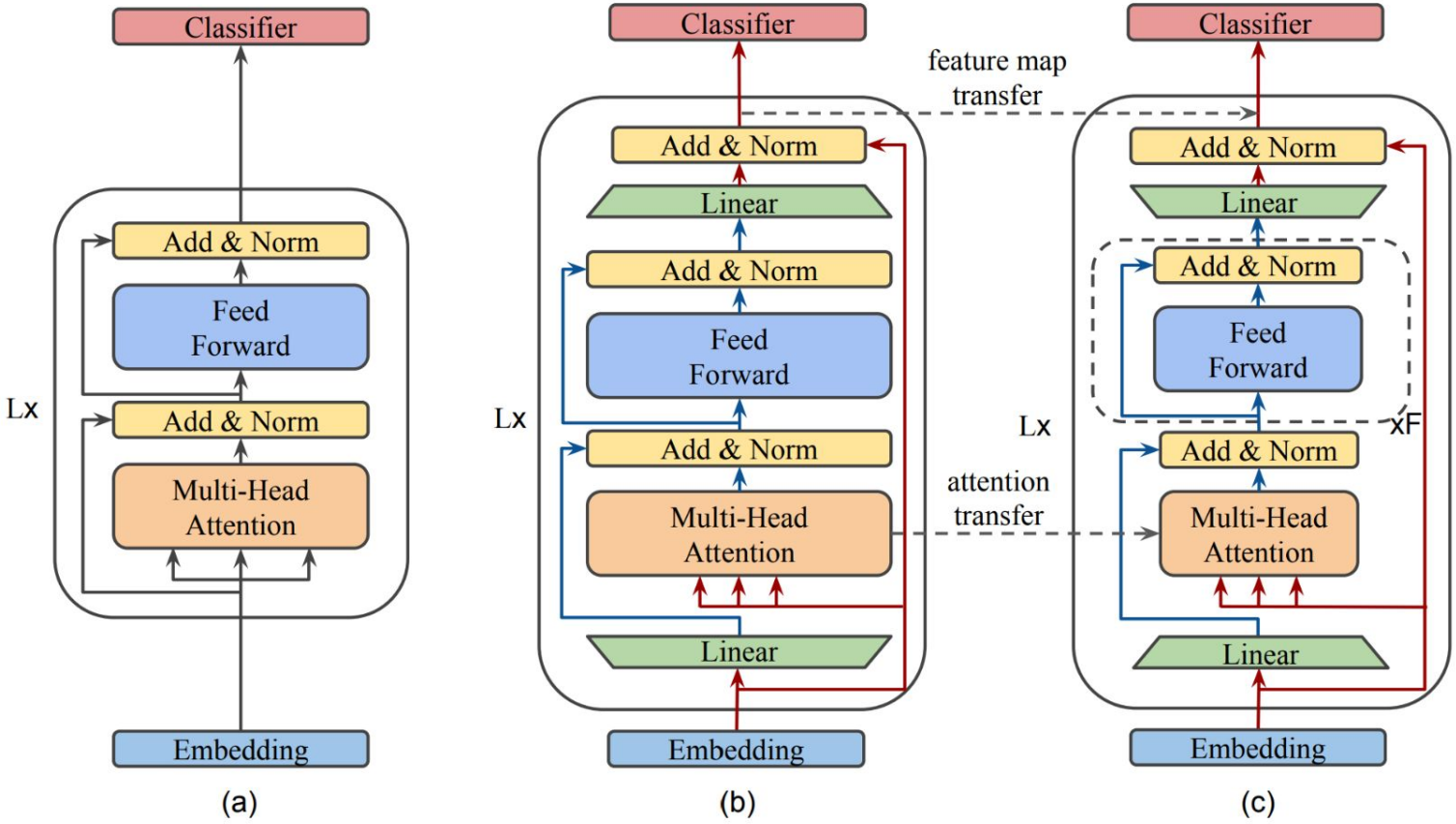


Figure 1: Illustration of three models: (a) BERT; (b) Inverted-Bottleneck BERT (IB-BERT); and (c) MobileBERT. In (b) and (c), red lines denote inter-block flows while blue lines intra-block flows. MobileBERT is trained by layer-to-layer imitating IB-BERT.

			$\text{BERT}_{\text{LARGE}}$	$\text{BERT}_{\text{BASE}}$	$\text{IB-BERT}_{\text{LARGE}}$	MobileBERT	$\text{MobileBERT}_{\text{TINY}}$
embedding		$h_{\text{embedding}}$	1024	768	128		
			no-op	no-op	3-convolution		
		h_{inter}	1024	768	512		
body	Linear	h_{input} h_{output}	$\left[\begin{pmatrix}1024\\16\\1024\end{pmatrix}\right.$ $\left.\begin{pmatrix}1024\\4096\\1024\end{pmatrix}\right]\times 24$	$\left[\begin{pmatrix}768\\12\\768\end{pmatrix}\right.$ $\left.\begin{pmatrix}768\\3072\\768\end{pmatrix}\right]\times 12$	$\left[\begin{pmatrix}512\\1024\end{pmatrix}\right.$ $\left.\begin{pmatrix}1024\\512\end{pmatrix}\right]\times 24$	$\left[\begin{pmatrix}512\\128\end{pmatrix}\right.$ $\left.\begin{pmatrix}128\\512\end{pmatrix}\times 4\right]\times 24$	$\left[\begin{pmatrix}512\\128\end{pmatrix}\right.$ $\left.\begin{pmatrix}128\\512\end{pmatrix}\times 2\right]\times 24$
	MHA	h_{input} #Head h_{output}					
	FFN	h_{input} h_{FFN} h_{output}					
	Linear	h_{input} h_{output}					
#Params			334M	109M	293M	25.3M	15.1M

Table 1: The detailed model settings of a few models. h_{inter}, h_{FFN}, $h_{\text{embedding}}$, #Head and #Params denote the inter-block hidden size (feature map size), FFN intermediate size, embedding table size, the number of heads in multi-head attention, and the number of parameters, respectively.

Operational Optimizations

- **Remove Layer Normalization**
  - Replace with an element-wise linear transformation
- **Replace gelu with relu activation**

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

Feature Map Transfer (FMT)

Goal:

- Match the layer-wise feature maps between teacher and student model
- Use the Mean Squared Error loss
- T is the sequence length, N is the feature map size, l is the layer index

$$\mathcal{L}_{FMT}^{\ell} = \frac{1}{TN} \sum_{t=1}^{T} \sum_{n=1}^{N} (H_{t,\ell,n}^{tr} - H_{t,\ell,n}^{st})^2$$

Attention Transfer (AT)

Goal:

- Match the attention learned by the student to the teacher
- Use KL-divergence between the per-head self-attention distributions of the student and teacher
- A is the number of attention heads

$$\mathcal{L}_{AT}^{\ell} = \frac{1}{TA} \sum_{t=1}^{T} \sum_{a=1}^{A} D_{KL}(a_{t,\ell,a}^{tr} || a_{t,\ell,a}^{st})$$

Pre-training Distillation (PD)

Goal:

- Use the teacher to guide the student on Masked Language Modeling (MLM) loss during pretraining
- The learning objective of the student becomes

$$\mathcal{L}_{PD} = \alpha \mathcal{L}_{MLM} + (1 - \alpha)\mathcal{L}_{KD} + \mathcal{L}_{NSP}$$

Ablation Study

	MNLI-m	QNLI	MRPC	SST-2
$\text{BERT}_{\text{LARGE}}$	86.6	92.1†	**87.8**	93.7
$\text{IB-BERT}_{\text{LARGE}}$	**87.0**	**93.2**	87.3	**94.1**
$\text{BERT}_{\text{BASE}}$	**84.4**	91.1†	86.7	**92.9**
MobileBERT (bare)	80.8	88.2	84.3	90.1
+ PD	81.1	88.9	85.5	91.7
+ PD + FMT	83.8	91.1	**87.0**	92.2
+ PD + FMT + AT	**84.4**	**91.5**	**87.0**	92.5

Result

	#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**

Table 4: The test results on the GLUE benchmark (except WNLI). The number below each task denotes the number of training examples. The metrics for these tasks can be found in the GLUE paper (Wang et al., 2018). "OPT" denotes the operational optimizations introduced in Section 3.3. †denotes that the results are taken from (Jiao et al., 2019). *denotes that it can be unfair to directly compare MobileBERT with these models since MobileBERT is task-agnosticly compressed while these models use the teacher model in the fine-tuning stage.

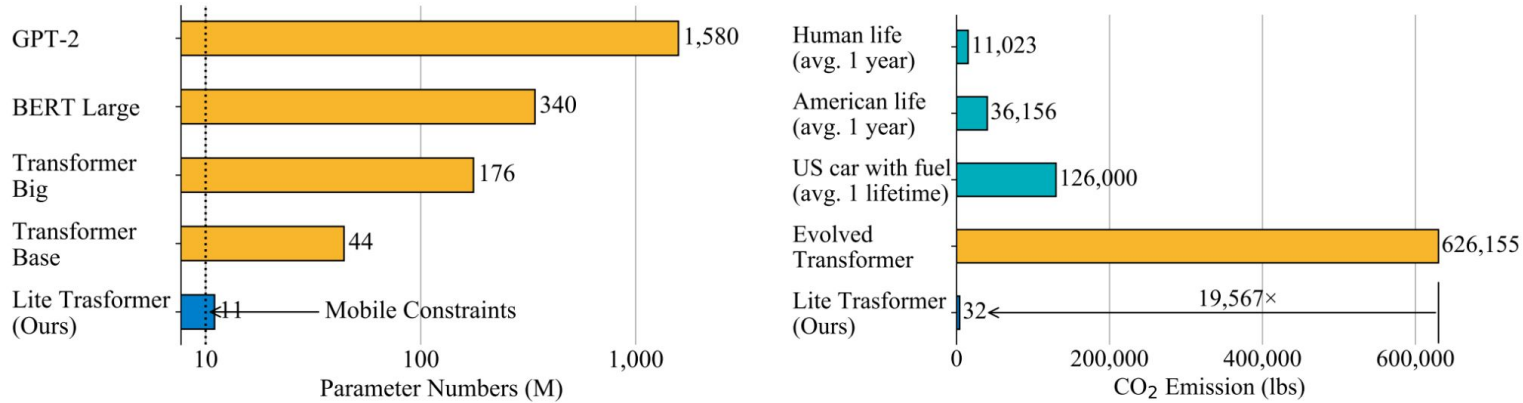
Paper 3

Lite Transformer with Long-Short Range Attention [Wu et al, 2020]

Overview

- Long-Short Range Attention (LSRA) modules to decouple the learning of local context and long-distance relationship
- Reducing the computation of Transformer base by up to 18.2x with little performance degradation
- Outperforming Transformer and Evolved Transformer under similar computation overhead

Motivation

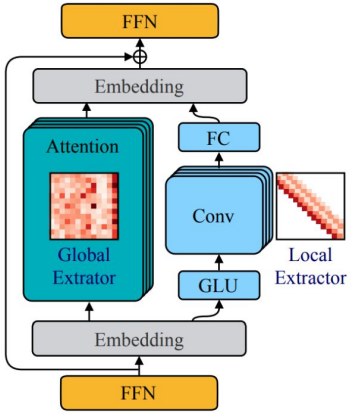


(a) Parameter numbers of modern NLP models.

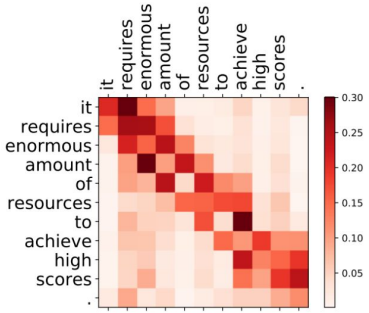
(b) The design cost measured in CO_2 emission (lbs).

Figure 1: Left: the size of recent NLP models grows rapidly and exceeds the mobile constraints to a large extent. Right: the search cost of AutoML-based NLP model is prohibitive, which emits carbon dioxide nearly 5× the average lifetime emissions of the car.

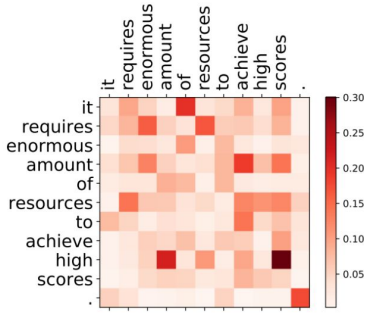
Long-Short Range Attention



(a) Lite Transformer block



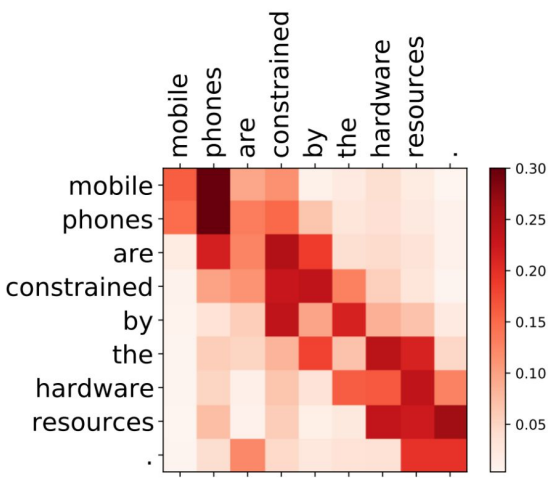
(b) Conventional Attention. It captures local information on the diagonal and global context as sparse points. (Redundant)



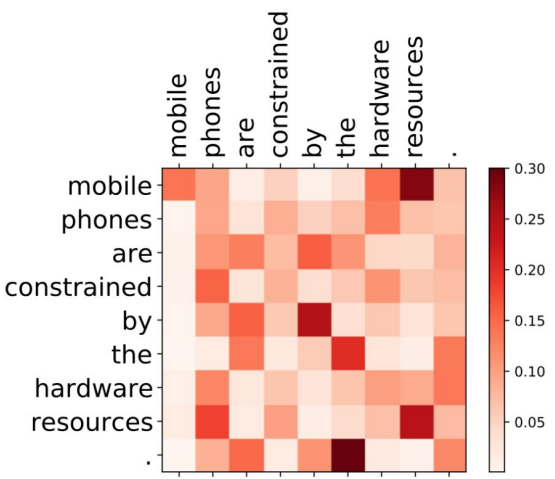
(c) Attention in LSRA. It is specialized for long-term relationships, indicated as points away from the diagonal. (Efficient)

Figure 3: Lite Transformer architecture (a) and the visualization of attention weights. Conventional attention (b) puts too much emphasis on local relationship modeling (see the diagonal structure). We specialize the local feature extraction by a convolutional branch which efficiently models the locality so that the attention branch can specialize in global feature extraction (c). More visualizations are available in Figure A1.

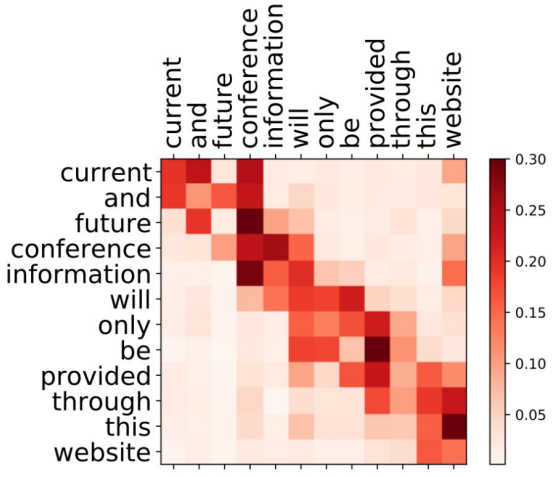
Attention Visualization



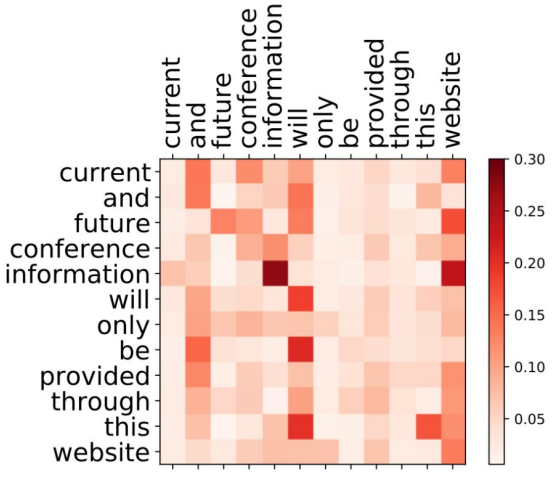
(a) Conventional Attention.



(b) Attention in LSRA.



(c) Conventional Attention.



(d) Attention in LSRA.

Flattened FFN Layers

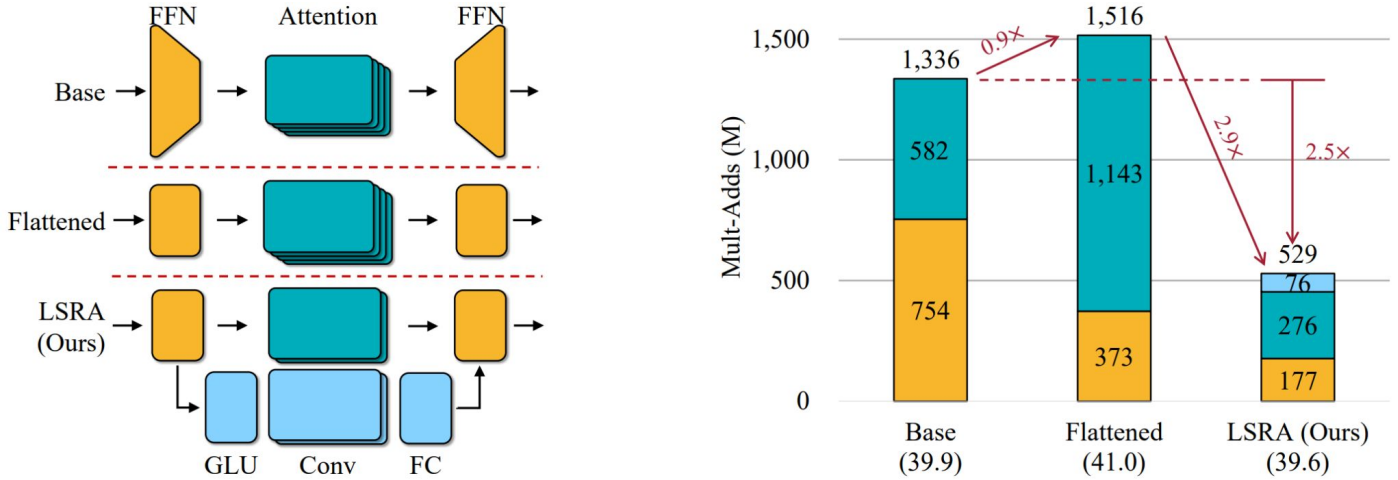


Figure 2: Flattening the bottleneck of transformer blocks increases the proportion of the attention versus the FFN, which is good for further optimization for attention in our LSRA.

Machine Translation

	#Parameters	#Mult-Adds	WMT'14 En-De		WMT'14 En-Fr	
			BLEU	ΔBLEU	BLEU	ΔBLEU
Transformer (Vaswani et al., 2017)	2.8M	87M	21.3	–	33.6	–
Lite Transformer (Ours)	2.9M	90M	**22.5**	**+1.2**	**35.3**	**+1.7**
Transformer (Vaswani et al., 2017)	11.1M	338M	25.1	–	37.6	–
Lite Transformer (Ours)	11.7M	360M	**25.6**	**+0.5**	**39.1**	**+1.5**
Transformer (Vaswani et al., 2017)	17.3M	527M	26.1	–	38.4	–
Lite Transformer (Ours)	17.3M	527M	**26.5**	**+0.4**	**39.6**	**+1.2**

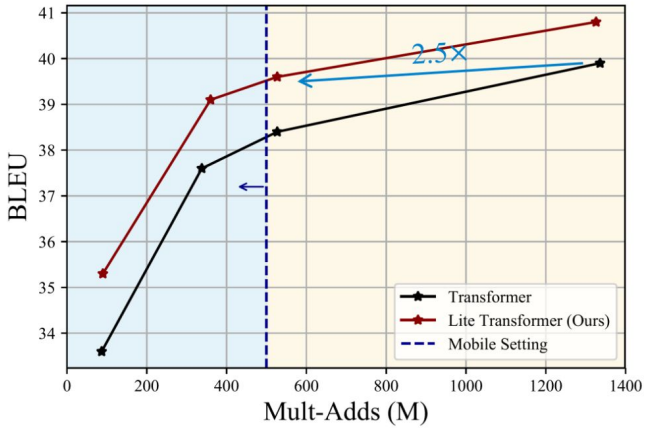
Table 2: Results on WMT'14 En-De and WMT'14 En-Fr. Our Lite Transformer improves the BLEU score over the transformer under similar Mult-Adds constraints.

Machine Translation

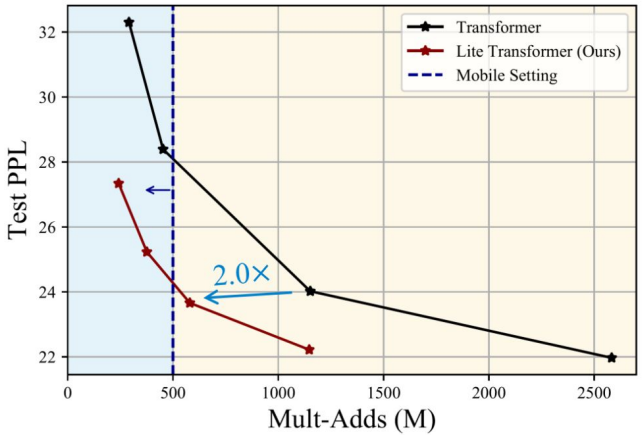
	#Params	#Mult-Adds	BLEU	GPU Hours	CO_2e (lbs)	Cloud Computation Cost
Transformer (Vaswani et al., 2017)	2.8M	87M	21.3	8×12	26	$68 - $227
Evolved Transformer (So et al., 2019)	3.0M	94M	22.0	8×274K	626K	$1.6M - $5.5M
Lite Transformer (Ours)	2.9M	90M	**22.5**	8×14	32	$83 - $278
Transformer (Vaswani et al., 2017)	11.1M	338M	25.1	8×16	36	$93.9 - $315
Evolved Transformer (So et al., 2019)	11.8M	364M	25.4	8×274K	626K	$1.6M - $5.5M
Lite Transformer (Ours)	11.7M	360M	**25.6**	8×19	43	$112 - $376

Table 3: Performance and training cost of an NMT model in terms of CO_2 emissions (lbs) and cloud compute cost (USD). The training cost estimation is adapted from Strubell et al. (2019). The training time for transformer and our Lite Transformer is measured on NVIDIA V100 GPU. The cloud computing cost is priced by AWS (lower price: spot instance; higher price: on-demand instance).

Machine Translation & Language Modeling



(a) BLEU score *vs*. Mult-Adds (on WMT En-Fr)



(b) PPL *vs*. Mult-Adds (on WIKITEXT-103)

Figure 4: Trade-off curve for machine learning on WMT En-Fr and language modeling on WIKITEXT-103 dataset. Both curves illustrate that our Lite Transformer outperform the basic transformer under the mobile settings (blue region).

Further Compression

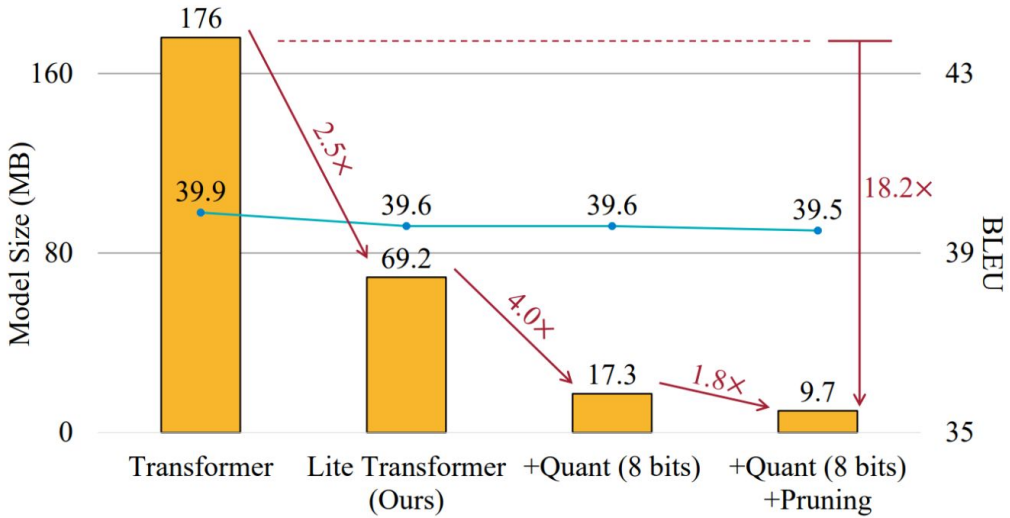


Figure 5: The model size and BLEU score on WMT En-Fr dataset with model compression. Our Lite Transformer can be combined with general compression techniques and achieves 18.2× model size compression. * 'Quant' indicates 'Quantization'.

Abstractive Summarization

	#Params	#MAdds (30)	#MAdds (100)	#MAdds (1000)	R-1	R-2	R-L
Transformer	44.1M	2.0G	3.6G	29.9G	41.4	18.9	38.3
Lite Transformer	**17.3M**	**0.8G**	**1.5G**	**12.5G**	41.3	18.8	38.3

Table 4: Results on CNN-DailyMail dataset for abstractive summarization. Our Lite Transformer achieves similar F1-Rouge (R-1, R-2 and R-L) to the transformer (Vaswani et al., 2017) with more than 2.4× less computation and 2.5× less model size. "#MAdds (x)" indicates the #Mult-Adds required by the model with the input length of x.

Language Modeling

	#Params	#MAdds (100)	#MAdds (1000)	Speed (tokens/s)	Valid ppl.	Test ppl.
Adaptive Inputs	37.8M	3.9G	50.3G	7.6K	23.2	24.0
Lite Transformer	37.2M	3.9G	48.7G	10.2K	**21.4**	**22.2**

Table 5: Results on WIKITEXT-103 dataset for language modeling. We apply our Lite Transformer architecture on transformer base model with adaptive inputs (Baevski & Auli, 2019) and achieve 1.8 lower test perplexity under similar resource constraint.

Paper 4

Synthesizer: Rethinking Self-Attention in Transformer Models [Tay et al, 2020]

Overview

Is dot-product self-attention really needed?

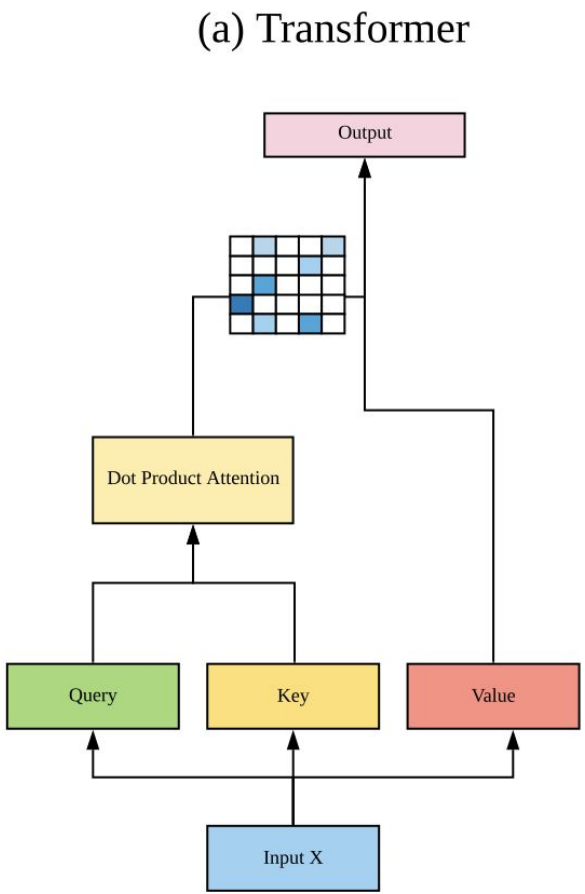
- Random alignment matrices perform surprisingly competitively
- Attention weights from token-token (query-key) interactions is useful but not that important

Rethinking Dot-Product Self-Attention

Observations:

- To learn self-alignment
  - i.e. To determine the relative importance of a single token with respect to all other tokens in the sequence
- Attention weights are learned at the instance / sample level pairwise interactions
  - Such interactions often fluctuate freely across different instances
  - Lack of consistent global context

Rethinking Dot-Product Self-Attention



The Proposed Alternatives

- Dense Synthesizer
- Random Synthesizer
- Factorized Dense Synthesizer
- Factorized Random Synthesizer

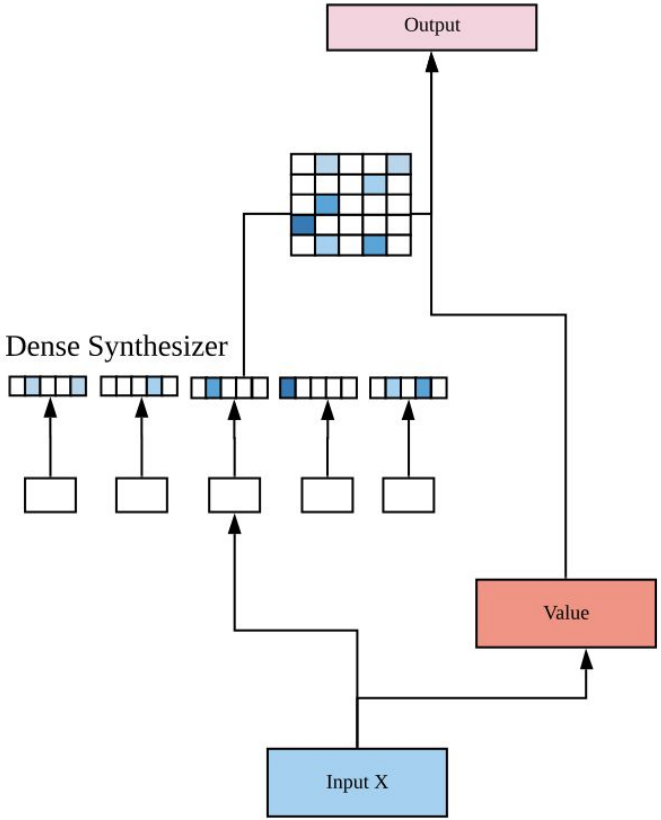
Dense Synthesizer

Use a 2-layer fully connected module with ReLU activations to replace the self-attention

$$F_{h,\ell}(X_{i,h,\ell}) = W_{2,h,\ell}(\sigma_R(W_{1,h,\ell}(X_{i,h,\ell}))$$

$$Y_{h,\ell} = \text{softmax}(B_{h,\ell})G_{h,\ell}(X_{h,\ell})$$

(b) Synthesizer (Dense)

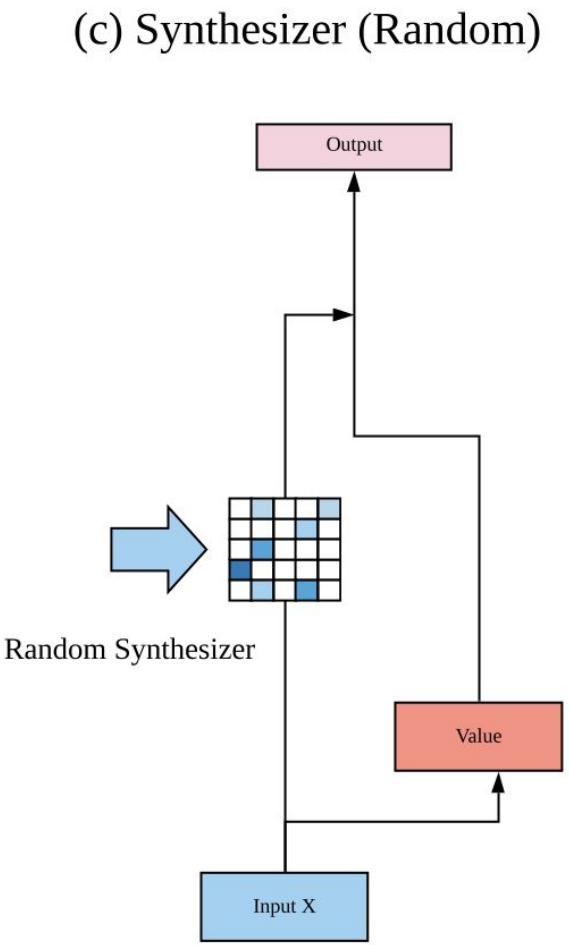


Random Synthesizer

Use a randomly initialized square matrix as the attention matrix

The random matrix can either be fixed or trainiable

$$Y_{h,\ell} = \text{softmax}(R_{h,\ell})G_{h,\ell}(X_{h,\ell})$$



Factorized Synthesizer

For Dense Synthesizer

- Use a bottleneck layer to decompose the Linear Layer from N×N to N×k, k×N

For Random Synthesizer

- Use 2 low rank matrices of N×k to reconstruct the original N×N matrices

Machine Translation & Language Modeling

Model	NMT (BLEU) $\lvert\theta\rvert$	EnDe	EnFr	LM (PPL) $\lvert\theta\rvert$	LM
Transformer†	67M	27.30	38.10	-	-
Transformer	67M	27.67	41.57	70M	38.21
Synthesizer (Fixed Random)	61M	23.89	38.31	53M	50.52
Synthesizer (Random)	67M	27.27	41.12	58M	40.60
Synthesizer (Factorized Random)	61M	27.30	41.12	53M	42.40
Synthesizer (Dense)	62M	27.43	41.39	53M	40.88
Synthesizer (Factorized Dense)	61M	27.32	41.57	53M	41.20
Synthesizer (Random + Dense)	67M	27.68	41.21	58M	42.35
Synthesizer (Dense + Vanilla)	74M	27.57	41.38	70M	**37.27**
Synthesizer (Random + Vanilla)	73M	**28.47**	**41.85**	70M	40.05

Table 2. Experimental Results on WMT'14 English-German, WMT'14 English-French Machine Translation tasks and Language Modeling One Billion (LM1B). † denotes original reported results in (Vaswani et al., 2017).

GLUE & SuperGLUE Benchmark

Model	Glue	CoLA	SST	MRPC	STSB	QQP	MNLI	QNLI	RTE
T5 (Base)	83.5	53.1	**92.2**	**92.0/88.7**	89.1/88.9	88.2/91.2	84.7/**85.0**	91.7	76.9
T5 (Base+)	82.8	54.3	92.9	88.0/83.8	85.2/85.4	88.3/91.2	84.2/84.3	91.4	79.1
DyConv	69.4	33.9	90.6	82.6/72.5	60.7/63.1	84.2/88.2	73.8/75.1	84.4	58.1
Syn (R)	75.1	41.2	91.2	85.9/79.4	74.0/74.3	85.5/89.0	77.6/78.1	87.6	59.2
Syn (D)	72.0	18.9	89.9	86.4/79.4	75.3/75.5	85.2/88.3	77.4/78.1	86.9	57.4
Syn (D+V)	82.6	48.6	92.4	91.2/87.7	88.9/89.0	88.6/91.5	84.3/84.8	91.7	75.1
Syn (R+V)	**84.1**	**53.3**	**92.2**	91.2/87.7	**89.3/88.9**	**88.6/91.4**	**85.0**/84.6	**92.3**	**81.2**

Table 5. Experimental results (dev scores) on multi-task language understanding (GLUE benchmark) for *small* model and `en-mix` mixture. Note: This task has been co-trained with SuperGLUE.

Model	SGlue	BoolQ	CB	CoPA	MultiRC	ReCoRD	RTE	WiC	WSC
T5 (Base)	70.3	78.2	72.1/83.9	59.0	73.1/32.1	**71.1/70.3**	77.3	**65.8**	**80.8**
T5 (Base+)	70.7	79.3	81.1/87.5	60.0	75.1/34.4	71.7/70.7	80.5	64.6	71.2
DyConv	57.8	66.7	65.9/73.2	58.0	57.9/8.71	58.4/57.4	69.0	58.6	73.1
Syn (R)	61.1	69.5	54.6/73.2	60.0	63.0/15.7	58.4/57.4	67.5	64.4	66.3
Syn (D)	58.5	69.5	51.7/71.4	51.0	66.0/15.8	54.1/53.0	67.5	65.2	58.7
Syn (D+V)	69.7	79.3	74.3/85.7	64.0	73.8/33.7	69.9/69.2	78.7	64.3	68.3
Syn (R+V)	**72.2**	**79.3**	**82.7/91.1**	**64.0**	**74.3/34.9**	70.8/69.9	**82.7**	64.6	75.0

Table 6. Experimental results (dev scores) on multi-task language understanding (SuperGLUE benchmark) for *small* model and `en-mix` mixture. Note: This task has been co-trained with GLUE.

Abstractive Summarization

Model	Sum. RL	B_4	Dialogue RL	Met.	CIDr
Trans.	35.77	3.20	13.38	5.89	18.94
Synthesizer Models					
R	33.10	2.25	15.00	6.42	19.57
D	33.70	**4.02**	**15.22**	**6.61**	**20.54**
D+V	**36.02**	3.57	14.22	6.32	18.87
R+V	35.95	2.28	14.79	6.39	19.09

Table 3. Experimental results on Abstractive Summarization (CNN/Dailymail) and Dialogue Generation (PersonaChat). We report on RL (Rouge-L), B4 (Bleu-4), Met. (Meteor) and CIDr.

Discussion

- Why does the Synthesizer work? What accounts for the success of Transformer models?

- Should Transformer be the answer to every NLP (or even CV) task?

References

[1] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, et al. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. ICLR 2020

[2] Zhiqing Sun, Hongkun Yu, Xiaodan Song, et al. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. ACL 2020

[3] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, Song Han. Lite Transformer with Long-Short Range Attention. ICLR 2020

[4] Yi Tay, Dara Bahri, Donald Metzler, et al. Synthesizer: Rethinking Self-Attention in Transformer Models. ICML 2020

[5] Alex Wang, Amanpreet Singh, Julian Michael, et al. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. ICLR 2019

[6] Alex Wang, Yada Pruksachatkun, Nikita Nangia, et al. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. NeurIPS 2019

[7] Karen Hao. Training a single AI model can emit as much carbon as five cars in their lifetimes.

[8] Kelvin. Model Compression via Pruning

[9] Aleksey Bilogur. A developer-friendly guide to model pruning in PyTorch

[10] Jonathan Frankle, Michael Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. ICLR 2019

References

[11] Neta Zmora, Hao Wu, Jay Rodge. Achieving FP32 Accuracy for INT8 Inference Using Quantization Aware Training with NVIDIA TensorRT

[12] Geoffrey Hinton, Oriol Vinyals, Jeff Dean. Distilling the Knowledge in a Neural Network. NeurIPS 2014 Deep Learning Workshop

[13] Sayak Paul. Distilling Knowledge in Neural Networks

[14] Samuel Sučík. Compressing BERT for faster prediction

[15] Ben Lutkevich. Language Modeling

[16] Huggingface text classification notebook

[17] Abstractive Text Summarization

[18] Ramesh Nallapati, Bowen Zhou, et al. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond

[19] Question Answering

[20] SQuAD

References

[21] RACE (ReAding Comprehension dataset from Examinations)

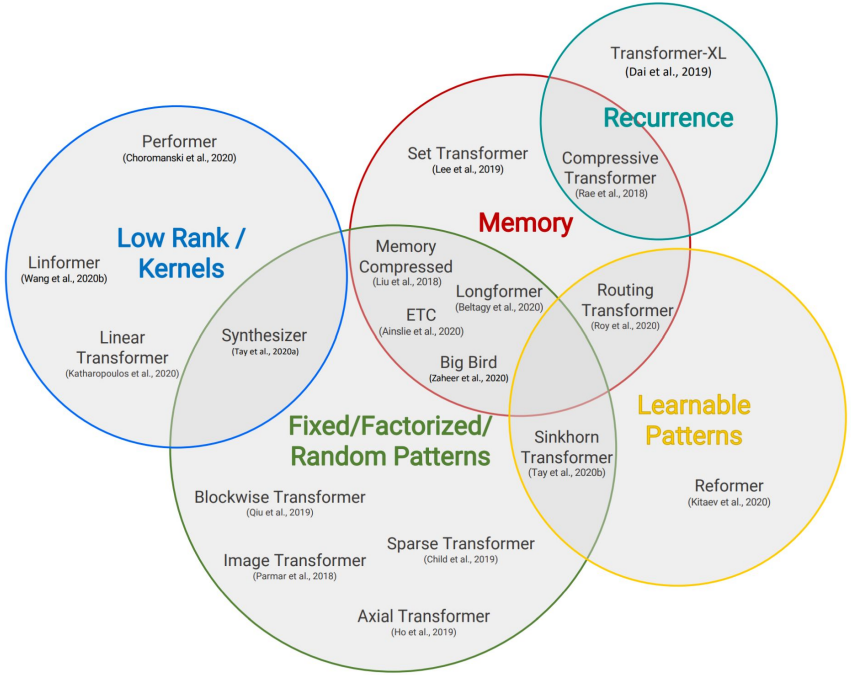
[22] Machine Translation

[23] Yi Tay, et al. Efficient Transformers: A Survey

Compression Tools

- TensorFlow Model Optimization Toolkit
- TensorFlow Lite Examples
- PyTorch Quantization
- PyTorch Pruning
- HuggingFace Transformers Android Demo Apps

More Efficient Transformers...



Source: Tay

More Efficient Transformers...

Model / Paper	Complexity	Decode	Class
Memory Compressed† (Liu et al., 2018)	$\mathcal{O}(n_c^2)$	✓	FP+M
Image Transformer† (Parmar et al., 2018)	$\mathcal{O}(n.m)$	✓	FP
Set Transformer† (Lee et al., 2019)	$\mathcal{O}(nk)$	✗	M
Transformer-XL† (Dai et al., 2019)	$\mathcal{O}(n^2)$	✓	RC
Sparse Transformer (Child et al., 2019)	$\mathcal{O}(n\sqrt{n})$	✓	FP
Reformer† (Kitaev et al., 2020)	$\mathcal{O}(n \log n)$	✓	LP
Routing Transformer (Roy et al., 2020)	$\mathcal{O}(n \log n)$	✓	LP
Axial Transformer (Ho et al., 2019)	$\mathcal{O}(n\sqrt{n})$	✓	FP
Compressive Transformer† (Rae et al., 2020)	$\mathcal{O}(n^2)$	✓	RC
Sinkhorn Transformer† (Tay et al., 2020b)	$\mathcal{O}(b^2)$	✓	LP
Longformer (Beltagy et al., 2020)	$\mathcal{O}(n(k+m))$	✓	FP+M
ETC (Ainslie et al., 2020)	$\mathcal{O}(n_g^2 + nn_g)$	✗	FP+M
Synthesizer (Tay et al., 2020a)	$\mathcal{O}(n^2)$	✓	LR+LP
Performer (Choromanski et al., 2020)	$\mathcal{O}(n)$	✓	KR
Linformer (Wang et al., 2020b)	$\mathcal{O}(n)$	✗	LR
Linear Transformers† (Katharopoulos et al., 2020)	$\mathcal{O}(n)$	✓	KR
Big Bird (Zaheer et al., 2020)	$\mathcal{O}(n)$	✗	FP+M

Table 1: Summary of Efficient Transformer Models presented in chronological order of their first public disclosure. Some papers presented sequentially may first appear at the same time, e.g., as an ICLR submission. Papers annotated with a superscript † are peer-reviewed papers. Class abbreviations include: FP = Fixed Patterns or Combinations of Fixed Patterns, M = Memory, LP = Learnable Pattern, LR = Low Rank, KR = Kernel and RC = Recurrence. Furthermore, n generally refers to the sequence length and b is the local window (or block) size. We use subscript g on n to denote global memory length and n_c to denote convolutionally compressed sequence lengths.

Source: Tay

Thank you!