Enhancing MobileBERT with Dynamic Data Augmentation and Fine tuning with LoRa

Final Project

Gil Biton

Paper Motivation

Why MobileBERT?

A lightweight version of BERT that balances efficiency and performance for **resource-constrained devices**.

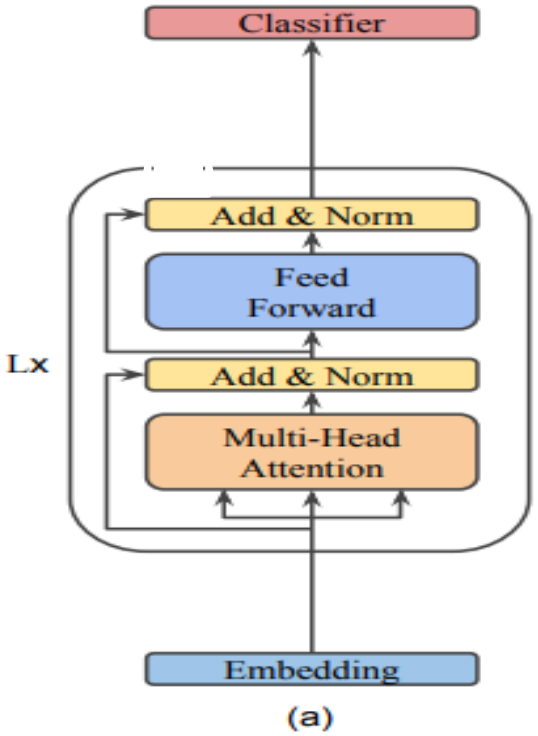
MobileBERT is 5.5 times faster than BERT and 4.3 times smaller in size.

Very close in actual results to BERT, sometimes even higher

BERT: **B**idirectional **E**ncoder **R**epresentations from **T**ransformers

ENCODER MODEL- FIND CONNECTIONS BETWEEN WORDS IN A SENTENCE, USING **ATTENTION**

COMMON TASKS : SENTIMENT ANALYSIS, SEARCH ENGINES, Q&A



Architecture :

Embedding - The → [0.2, 0.8, 0.3, …] cat → [0.5, 0.1, 0.9, …] sat → [0.7, 0.3, 0.6, …]

Multi-head attention – the words related to each word. 3 inputs :

- **Query-** For the word 'cat', who is the word 'cat' related to?
- **Key-** Each word in a sentence receives a Key vector that represents its relationship to the word
- **Values -** the final weights that are passed on

Add & Norm - Residual Connection + Normalization

Feed Forward - After the connections between the words have been processed, each vector passes through another neural network that performs additional processing to strengthen the connections.

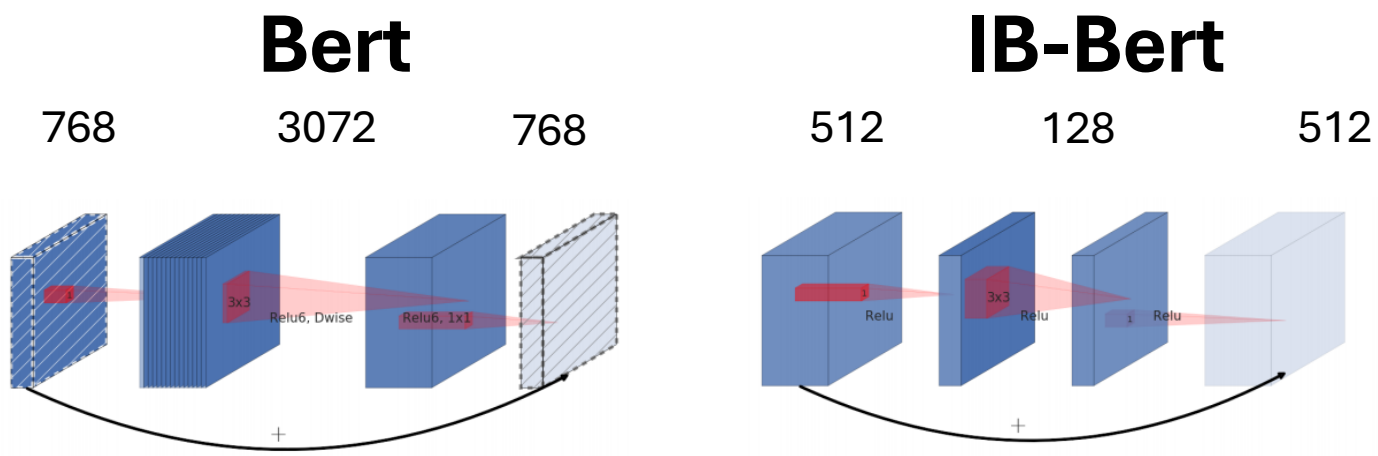
Lx - Repetitions of the Transformer block. Reinforcing sentence comprehension.

IB-Bert – “Thin” Version of BERT

The general idea - dimensionality reduction before the FFN network by linear transformation

In **BERT**, dimensionality expansion is performed which increases the complexity significantly

In contrast, in IB BERT, dimensionality reduction is performed before the network, and expansion is performed at the end so that the number of dimensions remains the same but the complexity decreases



Knowledge Distillation- Motivation

A method for reducing large models while preserving the knowledge acquired during training.

In practice, a "Student" model learns to imitate the outputs of a "Teacher" model

Imagine that we want to make cakes in the most efficient way.

- **BERTLARGE** - An expert baker who learns from scratch how to bake cakes.
  It takes him years to learn and train himself. This process is very expensive and slow.
- **MobileBERT** - A student with a simplified recipe
  Instead of learning from scratch, he **receives a simplified version of the recipe** that keeps only the essential steps.

What did we gain?
Instead of each student learning on their own from scratch (like training MobileBERT from scratch),
everyone uses the teacher's knowledge and learns much faster.

Knowledge Distillation- Steps

Step 1 – **Input** - same text to both IB-BERT and MobileBERT.

For example, the sentence "The cat sat on the mat".

Step 2 – **Attention Transfer**

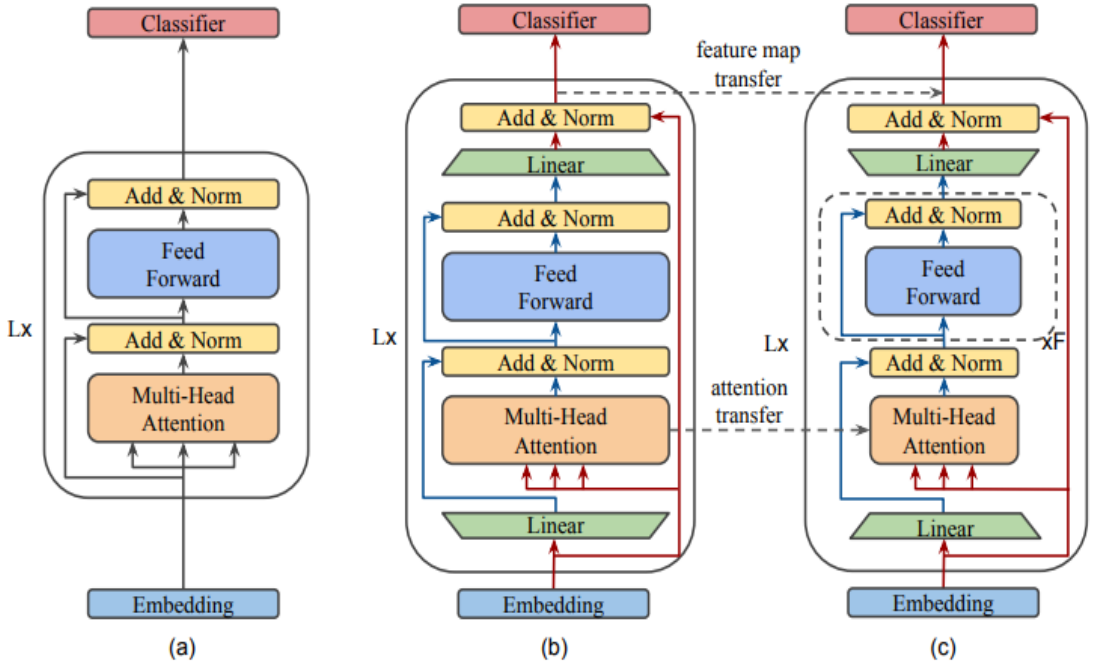
MobileBERT learns to which words IB-BERT pays more attention to. For example - the word "sat" is closely related to "cat“

Step 3 –**Feature Map Transfer**

MobileBERT learns to mimic the vectors that come out of each layer of IB-BERT.

Step 4 – Final Fine-Tuning

MobileBERT Main Changes from BERT :



Bert **IB-Bert** **MobileBERT**

	BERT (Large)	IB-BERT	MobileBERT
Parameters	334M	293M	**25.3M**
Number of Attention Heads	16	4	**4**
Embedding Size	768	512	**128** (using convolution)
Normalization	LayerNorm	LayerNorm	**NoNorm**
Activation Function	GELU	Gelu	**ReLU**
FFN Size	4096	128	**128**
Training (Pretraining)	Standard training	**Knowledge Distillation (from BERT)**	**Knowledge Distillation (from IB-BERT)**

Our Project

To improve the mini-model and its robustness, we will make key changes:

Dynamic Augmentation- modifying training data to increase its diversity and improve the generalizability of the model, By:

Change words order :

"The cat sat on the mat." ⟷ "On the mat, the cat sat."

Adding Stop Words:

"The cat sat on the mat." ⟷ "The cat is sat on the mat."

Replacing by synonyms and phrases with the same meaning :

"The cat sat on the mat." ⟷ "The feline rested on the rug."

Fine-Tuning with LORA - Use original weights (without augmentation) and perform FINE TUNING after retraining with augmentation using LORA**.**

Project Pipeline

Dynamic Data Augmentation:

- Introduce real-time variations in input data during fine-tuning to improve model generalization and robustness.

Efficient Fine-Tuning with LoRA:

- Use Low-Rank Adaptation (LoRA) to adapt MobileBERT's pre-trained weights efficiently, reducing computational overhead.

Evaluate on various Tasks:

- Assess improvements in generalization and robustness across multiple NLP benchmarks.

Expected Results

Improved Generalization:

Higher accuracy and F1 scores on augmented datasets and unseen data.

Robustness to Variations:

Better handling of noisy or syntactically altered input.

Efficient Fine-Tuning:

Reduced training time and memory usage due to LoRa.

Deployment-Ready Model:

Enhanced MobileBERT remains efficient and suitable for mobile applications.

Future Directions

1. **Domain-Specific Applications**:
   Fine-tuning MobileBERT for specific domains like healthcare or legal NLP tasks.

2. **Advanced Augmentation**:
   Use backtranslation or context-aware replacements for better augmentations.

3. **Integration with Edge AI**:
   Implement the enhanced MobileBERT on edge devices for real-time NLP.