Triple-compressed BERT for MNLI

Isaac Cheng
xianbing@stanford.edu

Abstract

Transformer based architectures have become de-facto models used for a range of Natural Language Processing tasks. In particular, the BERT based models achieved significant accuracy gain for GLUE tasks, CoNLL-03 and SQuAD. However, BERT based models have a prohibitive memory footprint and latency. To address these problems, I propose to compress BERT using a combination of three compression methods: knowledge distillation, layer pruning and quantization. I test my proposed method on the downstream task, MNLI. As a result, compared with the DistilBERT which is obtained by using only one compression method of knowledge distillation, my best model achieves comparable performance with at most 4.8% and 6.7% performance degradation on the MNLI matched task and MNLI mismatched task respectively, while being 53.5% smaller and 63% faster. I demonstrate the effectiveness of using a combination of compression methods on BERT.

1 Introduction

The NLP community has witnessed a revolution of pre-training self-supervised models. These models usually have hundreds of millions of parameters. Among these models, BERT (Devlin et al., 2018) shows substantial accuracy improvements. However, as one of the largest models ever in NLP, BERT suffers from the heavy model size and high latency, making it impractical for resource-limited mobile devices to deploy the power of BERT in mobile-based machine translation, dialogue modeling, and the like.

There have been some efforts that address this challenge by compressing BERT into a compact model. A promising method is knowledge distillation (Sanh et al., 2019; Sun et al., 2020), which trains a smaller BERT model through distillation via the supervision of a bigger BERT model. Another promising method is quantization (Shen et al., 2019), which uses low bit precision for parameter storage and enables low bit hardware operations to speed up inference. Pruning (Fan et al., 2019), such as attention head pruning and layer pruning, is also a very effective compression method.

However, to the best of my knowledge, there is not yet any work to compress BERT using a combination of compression methods. The central hypothesis of this paper is that a smaller and faster BERT can be obtained by a combination of different compression methods, compared with a BERT model which is obtained by using merely one compression method, while still maintaining good performance on a natural language inference task, MNLI (Williams et al., 2018).

At first glance, it may seem straightforward to obtain a compact BERT by simply applying a series of compression techniques. For example, one may just take a BERT model, distill and prune it, and apply quantization on it. Unfortunately, such a straightforward approach can result in significant accuracy loss. This may not be that surprising, since how different compression methods interact is not taken into account in this oversimplified process.

In this paper, I design a triple-compressed BERT which is obtained by using three compression methods: knowledge distillation, quantization and pruning.

This triple-compressed BERT is based on a pre-trained DistilBERT model which is obtained by compressing BERT using knowledge distillation and can be further fine-tuned with good performances on a wide range of tasks. I fine-tune this DistilBERT model on MNLI task. During the fine-tuning, a pruning technique, LayerDrop (Fan et al., 2019), is used. The LayerDrop technique has a regularization effect during fine-tuning and allows for efficient layer pruning at inference time. After fine-tuning, I further compress the BERT model by applying quantization only on the linear modules

of the model.

As a result of these design decisions, I am able to obtain a triple-compressed BERT model which is 53.5% smaller and 63% faster than the DistilBERT which is obtained by using only one compression method of knowledge distillation, while it can still achieve comparable performance with at most 4.8% and 6.7% performance degradation on the MNLI matched task and MNLI mismatched task respectively.

2 Related Work

2.1 Knowledge Distillation

Knowledge distillation (Bucila et al., 2006; Hinton et al., 2015) is a compression technique in which a compact model - the student - is trained to reproduce the behaviour of a larger model - the teacher - or an ensemble of models. Sanh et al., 2019 propose DistilBERT, which is a student model trained with the training objective to minimize a triple loss: masked language modeling (MLM) loss, distillation loss over the soft target probabilities of the teacher, and cosine embedding loss which will tend to align the directions of the student and teacher hidden state vectors. While DistillBERT focuses on reducing the number of layers (i.e. the depth), Sun et al., 2020 propose MobileBERT which focuses on reducing the width. In the work of MobileBERT, the student model is as deep as the teacher model, but each building block is made much smaller. The training objective is to minimize feature map transfer loss, attention transfer loss, and pre-training distillation loss.

My triple-compressed BERT model is fine-tuned based on the pretrained DistilBERT model, and further compressed by two more techniques: LayerDrop and quantization.

2.2 Quantization

Quantization (Choi et al., 2018) is proven to be an effective way for model compression. Q-BERT (Shen et al., 2019) uses quantization to compress fine-tuned BERT. It achieves comparable performance on downstream tasks of SST-2, MNLI, CoNLL-03 and SQuAD. Q-BERT first applies mixed-precision quantization on BERT. Instead of assigning the same number of bits (or quantization precision) to all the layers, Q-BERT assigns more bits to more sensitive layers in order to retain performance. To be able to determine which NN layer is more sensitive to quantization, a Hessian AWare Quantization is developed, which uses both mean and variances of top eigenvalues as a sensitivity measurement.

Q-BERT also uses group-wise quantization, based on the hypothesis that directly quantizing all matrices as an entirety with the same quantization range can significant degrade the accuracy. Group-wise quantization in Q-BERT partitions each matrix into 128 groups, each with its unique quantization range and loop up table.

My triple-compressed BERT model gets much insight from Q-BERT. Similar as Q-BERT which applies different quantization precision and range into different part of the BERT model, I apply quantization only on the linear modules of the BERT model. Therefore, only the linear modules of the triple-compressed BERT model has qint8 precision, while other modules keeps float32 precision.

2.3 Pruning

My approach uses LayerDrop (Fan et al., 2019), a form of structured pruning. LayerDrop randomly drops layers at training time. It has a regularization effect on large models such as the BERT. BERT, which is trained with LayerDrop, is more robust to predicting with missing layers. During training time, pruning with a rate p means dropping the layers at a depth d such as $d\%(1/p)$ is equal to 0. In this paper, $p = 0.2$. During inference time, the strategy pruning strategy is: simply drop every other layer. RoBERTa + LayerDrop, outperforms BERT and RoBERTa trained from scratch, on MNLI-m, MRPC, QNLI, and SST-2. LayerDrop is as simple to implement as dropout.

My triple-compressed BERT model is fine-tuned based on the pretrained DistilBERT model with LayerDrop enabled.

During fine-tuning, I perform LayerDrop on the DistilBERT model with a rate $p = 0.2$. At inference time, I don't follow the standard pruning strategy: simply drop every other layer. Rather, I keep the 1st, 2nd, 3rd and 5th layer, and only drop the 4th and 6th layer. The reasoning is that the pretrained DistilBERT model has only 6 layers. The representation power of each layer, especially the first several layers, are critical.

I compare the resulting model (i.e. the pruned model by only dropping 4th and 6th layer) with the model pruned by simply dropping every other layer (i.e. 2nd, 4th, and 6th layer) and find that latter model suffers considerable accuracy loss.

3 Data

The dataset used in this work is the Multi-Genre natural Language Inference (MultiNLI) corpus, introduced by (Williams et al., 2018).

The corpus is derived from ten different genres of written and spoken English, which are collectively meant to approximate the full diversity of ways in which modern standard American English is used.

It has a collection of 433k sentence pairs annotated with textual entailment information (entailment, contradiction or neutral), in which

- 392,702 are training examples, drawn from five genres (Fiction, Government, The Slate website, Telephone and Travel);
- ∼20K are dev examples, in which
  - 9,815 are matched examples, drawn from the same five genres as the training examples;
  - 9832 are mismatched examples, drawn from additional five genres (The 9/11 report, Face-to-Face, Fundraising letters, Non-fiction from Oxford University Press, Verbatim)
- ∼20K are test examples, in which
  - 9796 are matched examples, drawn from the same five genres as the training examples;
  - 9847 are mismatched examples, drawn from additional five genres (The 9/11 report, Face-to-Face, Fundraising letters, Non-fiction from Oxford University Press, Verbatim)

In this paper, I use MNLI dataset from `https://dl.fbaipublicfiles.com/glue/data/MNLI.zip`.

Table 1 shows randomly chosen development set examples from the MNLI dataset. For each example, premise, hypothesis and label used for classification (entailment, contradiction or neutral) are specified.

4 Triple-compressed BERT

4.1 Knowledge Distillation

In my work, the model is a DistilBERT base model with a sequence 3-classes (entailment, contradiction, and neutral) classification head. It has the same general architecture as DistilBERT base model. The number of layers (i.e. Transformer blocks) is 6, the hidden size is 3072, and the number of self-attention heads is 12.

The model is first initiated using a pre-trained DistilBERT base model which is obtained by compressing the BERT base model using knowledge distillation. This model is then fine-tuned on the MNLI dataset.

4.2 LayerDrop

During the fine-tuning process, LayerDrop technique is used in this way:

- before a batch of training examples is forwarded into a layer, generate a random number between 0 to 1, using normal distribution;
- if this random number is smaller than $p = 0.2$, simply ignore this layer. That's to say, we do not forward this batch of training examples to this layer;
- if this random number is larger or equal to $p = 0.2$, forward this batch of training example to this layer.
- continue to next layer, and do the same LayerDrop procedure again.

LayerDrop randomly drops layers at fine-tuning time. After fune-tuning is completed, I further compress the model by pruning its the 4th and 6th layer. Therefore, the resulting model has 4 (i.e. $6 - 2 = 4$) layers.

4.3 Quantization

After fine-tuning is completed and pruning is applied, I perform quantization on the model. Specifically, I specify that I want the torch.nn.Linear modules in the model (i.e. the multi-head self-attention module and the feed-forward network module) to be quantized. Also, I specify that I want weights to be converted from float32 to quantized int8 values.

After quantization, I obtain a triple-compressed BERT model, which is compressed by 3 different compression techniques: knowledge distillation, pruning and quantization.

5 Experiments

In this section, I first present the baseline model, then the metrics used to compare the baseline model and the triple-compressed BERT model, and present the results.

Premise	Genre	Gold Label	Hypothesis
Met my first girlfriend that way	FACE-TO-FACE	contradiction	I didn't meet my first girlfriend until later.
8 million in relief in the form of emergency housing.	GOVERNMENT	neutral	The 8 million dollars for emergency housing was still not enough to solve the problem.
Now, as children tend their gardens, they have a new appreciation of their relationship to the land, their cultural heritage, and their community.	LETTERS	neutral	All of the children love working in their gardens.
At 8:34, the Boston Center controller received a third transmission from American 11	9/11	entailment	The Boston Center controller got a third transmission from American 11.
I am a lacto-vegetarian.	SLATE	neutral	I enjoy eating cheese too much to abstain from dairy.
someone else noticed it and i said well i guess that's true and it was somewhat melodious in other words it wasn't just you know it was really funny	TELEPHONE	contradiction	No one noticed and it wasn't funny at all.

Table 1: Randomly chosen examples from the development set of MNLI dataset, shown with their genre labels, and their selected gold labels.

5.1 Baseline Model

The baseline model is a DistilBERT base model with a sequence 3-classes (entailment, contradiction, and neutral) classification head. It has the same general architecture as DistilBERT base model. It is initiated using a pre-trained DistilBERT base model which is obtained by compressing the BERT base model using knowledge distillation, then it is fine-tuned on the MNLI dataset.

There are only two differences between the baseline DistilBERT model and the triple-compressed BERT model:

- there is no LayerDrop applied on fine-tuning process of the baseline DistilBERT model;
- after fine-tuning completes, there is no quantization applied on the fine-tuned baseline DistilBERT model.

Both baseline DistilBERT and triple-compressed BERT are trained using the 392,702 training examples and evaluated on the ∼20K dev examples. Each training example is a token sequence which concatenates the premise-hypothesis pair with a [SEP] token, with [CLS] as the 1st and last token.

5.2 Metrics

Here are the 4 metrics used in this paper:

- MNLI-matched Accuracy: the sum of the correct predictions divided by the sum of all predictions for ∼10K matched dev examples.
- MNLI-mismatched Accuracy: the sum of the correct predictions divided by the sum of all predictions for ∼10K unmatched dev examples.
- Inference Time on CPU: the inference time on CPU needed for a full pass on the ∼10K MNLI dev examples.
- Model Size: model size in MB.

5.3 Results on MNLI

The baseline DistilBERT model is fine-tuned separately on 2 sub-tasks: MNLI-matched task and MNLI-mismatched task. Therefore, two different baseline DistilBERT models are obtained after fine-tuning. One is fine-tuned by using the 392,702 training examples and the ∼10K matched dev examples for the MNLI-matched task, and another

one is fine-tuned by using the same 392,702 training examples and the ∼10K mismatched dev examples for the MNLI-mismatched task.

Same is true for the triple-compressed BERT model. Two different triple-compressed BERT model are obtained. One is fine-tuned for the MNLI-matched task, and another is fine-tuned for the MNLI-mismatched task.

At inference time, I run each model to predict the labels for these ∼10k matched/unmatched dev examples on CPU, compute the MNLI-matched accuracy and MNLI-mismatched accuracy, and keep record of the inference time it takes for labelling these dev examples.

As shown in Table 2, the triple-compressed BERT is 4.8% point behind the baseline DistilBERT in accuracy on the MNLI-matched task, while being 53.5% smaller and 62.9% faster.

As shown in Table 3, the triple-compressed BERT is 6.7% point behind the baseline DistilBERT in accuracy on the MNLI-mismatched task, while being 53.5% smaller and 63.8% faster.

5.4 Ablation Studies

In this section, I evaluate the effect of applying pruning and quantization on the DistilBERT model.

Table 4 shows that after applying quantization but without LayerDrop, the accuracy loss on MNIL-matched task is only 1.3%, the inference time is 48.5% faster and the model is 48.2% smaller. In addition, after applying LayerDrop but without quantization, the accuracy loss is 5.6%, the inference time is 34.2% faster and the model is 21.2% smaller.

Table 5 that after applying quantization but without LayerDrop, the accuracy loss on MNIL-mismatched task is only 2.4%, the inference time is 46% faster and the model is 48.2% smaller. In addition, after applying LayerDrop but without quantization, the accuracy loss is 3.6%, the inference time is 30.1% faster and the model is 21.2% smaller.

I also evaluate the effect of pruning different layers on the performance of the model. Table 6 shows that the triple-compressed BERT which is obtained by dropping only the 4th and 6th layers, is 7% more accurate than triple-compressed BERT which is obtained by dropping the 2nd, 4th and 6th layer, on the MNLI-matched task. Table 7 shows that the triple-compressed BERT which is obtained by dropping only the 4th and 6th layers, is 10.5% more accurate than triple-compressed BERT which is obtained by dropping the 2nd, 4th and 6th layer, on the MNLI-mismatched task.

6 Analysis

Section 5.4 ablation study shows that applying only quantization on the fine-tuned DistilBERT does not cause much accuracy loss at all. And it greatly improve inference time and results in much smaller model. However, applying LayerDrop causes larger accuracy loss.

This indicates that quantization on the linear modules of the DistilBERT model is a very effective compression approach. It proves that the knowledge distillation and quantization can interact very well regarding compressing the BERT model.

The interaction of knowledge distillation and LayerDrop is less effective, compared with the interaction of knowledge distillation and quantization. But its result (accuracy, inference time and model size) is still very promising: there is no much accuracy loss and the model is much smaller and faster.

Section 5.4 ablation study also shows that if I prune the model by every other layer (i.e. 2nd, 4th and 6th layer), the model would suffer much more accuracy degradation than the model which is pruned by dropping only the 4th and 6th layer. This may be due to the fact: the original DistilBERT has only 6 layers, and its representation power would suffer much more loss if more of its layers are pruned.

Section 5.3 shows that the triple-compressed BERT is a very effective technique to compress the BERT model. It can make BERT much smaller and faster, while still maintain comparable accuracy on MNLI task, especially MNLI-matched task.

7 Conclusion

I have presented a triple-compressed BERT which is obtained by using a combination of three compres- sion methods: knowledge distillation, layer pruning and quantization. The results on MNLI task show that the triple-compressed BERT is comparable with the DistilBERT which is obtained by using only one compression method of knowledge distillation, while being much smaller and faster.

In this paper, I show that it is a very promising approach to combine different compression techniques to compress BERT.

Model	MNLI-matched Accuracy	Inference Time on CPU(s)	Model Size(MB)
Baseline: DistilBERT	0.817	830.8	267.86
Triple-compressed BERT	0.769	307.7	124.45

Table 2: Triple-compressed BERT yields comparable performance on MNLI-matched task, and it is much faster and smaller

Model	MNLI-mismatched Accuracy	Inference Time on CPU(s)	Model Size(MB)
Baseline: DistilBERT	0.819	879.4	267.86
Triple-compressed BERT	0.752	317.7	124.45

Table 3: Triple-compressed BERT yields comparable performance on MNLI-mismatched task, and it is much faster and smaller

Model	MNLI-matched Accuracy	Inference Time on CPU(s)	Model Size(MB)
Baseline: DistilBERT	0.817	830.8	267.86
DistilBERT + quantization	0.793	448.4	138.70
DistilBERT + LayerDrop	0.781	558.0	211.14
Triple-compressed BERT (DistilBERT + LayerDrop + quantization)	0.769	307.7	124.45

Table 4: The effect of applying pruning and quantization on the DistilBERT model, for MNLI-matched task.

Model	MNLI-mismatched Accuracy	Inference Time on CPU(s)	Model Size(MB)
Baseline: DistilBERT	0.819	879.4	267.86
DistilBERT + quantization	0.806	454.3	138.70
DistilBERT + LayerDrop	0.763	577.8	211.14
Triple-compressed BERT (DistilBERT + LayerDrop + quantization)	0.752	317.7	124.45

Table 5: The effect of applying pruning and quantization on the DistilBERT model, for MNLI-mismatched task.

Model	MNLI-matched Accuracy
A triple-compressed BERT model after dropping 2nd, 4th and 6th layers	0.699
A triple-compressed BERT model after dropping 4th and 6th layers	0.769

Table 6: The effect of pruning different layers on the performance of the model, for the MNLI-matched task.

Model	MNLI-mismatched Accuracy
A triple-compressed BERT model after dropping 2nd, 4th and 6th layers	0.647
A triple-compressed BERT model after dropping 4th and 6th layers	0.752

Table 7: The effect of pruning different layers on the performance of the model, for the MNLI-mismatched task.

Acknowledgements

The work presented in this paper is conducted by Isaac Cheng, myself. I would like to give thanks to God for loving and guiding me along the way. Also, I appreciate the help of all the XCS224U course staff for the support along the way.

References

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. *KDD*.

J. Choi, Z. Wang, S. Venkataramani, V. Srinivasan P. I.-J. Chuang, and K. Gopalakrishnan. 2018. Pact: Parameterized clipping activation for quantized neural networks. *arXiv:1805.06085*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *arXiv:1909.11556*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv,abs/1503.02531*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2019. Q-BERT: Hessian based ultra low precision quantization of BERT. *arXiv:1909.05840*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. *arXiv:2004.02984*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. *In Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.