Université de Montréal
Faculté des arts et des sciences
Département d'**informatique et de recherche opérationnelle**

Applying further model compression to TinyBERT

Gao Yinghan, Wei Wei, Frederic Boileau

February 6, 2021

Abstract

Attention based pretrained-language models (PLM) now dominate the state of the art (SOA) in NLP for language modelling tasks such as GLUE and more downstream ones like question answering (Squad) or question generation. The models however tend to be huge, the pretraining itself it not possible to experiment with unless one has access to an array of costly hardware accelerators. PLMs on the other hand can be fine tuned after pre-training for more specific tasks such as question answering. The problem still stands as the models are huge, while BERT has *only* 340M, others are much larger, such as Amazon's MegatronLM have 8.3B and for true exageration one can look at GShard, a swooping 600B. This poses challenges for deployment as even once trained inference and simply storing these models are costly. We thus tackle *model compression*. One very promising model issued from such an idea is TinyBERT which used knowledge distillation (KD). We will investigate how such a model can be further compressed with techniques such as *pruning* and *quantization*

Introcuction

Pretrained Language Models Bidirectional Encoder Representation from (BERT)[3] is an architecture based on the encoder module of the transformer. Its training is done in two steps, first it learns a Language Model (LM) which results in a Pretrained Language Model (PLM) through some unsupervised learning tasks. The training procedure is done over two tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). The MLM task simply trains the model to learn to predict randomly masked words from a sentence. In NSP, BERT learns to predict, given a pair of sentences, whether one follows the other. When trained on those tasks the BERT yields a PLM. The resulting model can then be fine-tuned in a supervised setting for a specific tasks such as question answering. This ability to train once over a huge corpus of data to yield a LM and then fine tune it downstream adresses one core issue of deep learning: how to transfer learning or knowledge? BERT is just one of many examples of large PLM deployed today. The adjoined table lists the models and their associated number of parameters.

Knowledge Distillation Knowledge Distillation (KD) addresse the following issue, how can we leverage the state of the art (SOA) results given by large PLMs to do inference in a context where memory and computing power are limited. One avenue would be to use one of the large PLMs to "teach" a smaller model (the student) which we can deploy in more ressource limited environments. Hinton, though not specifically with respect to PLMs, argued in 2015[4] that one conceptual roadblock to knowledge transfer or "distillation" had been the rigid identification of a model with the learned parameter values instead of the more abstract view of a "learned mapping from input vectors to output vectors"[4]. The way to do this according to Hinton et al is to make the student learn through an objective function which reflects the generalization ability learned in the teacher model. To achieve this he proposes using an objective function which averages over the soft target (the output probabilities of the teacher where the logits are divided by temperature factor to adjust the smoothness of those targets) and the ground truth.

Related Work

TinyBERT The idea of KD is a fertile one and has been applied to BERT to train a model called TinyBERT[5]. In this paper the authors "introduce a new two-stage learning framework for TinyBERT, which performs Transformer distillation at both the pre-training and task-specific learning stages." [5] This enables the model to learn both general LM features and more downstream tasks. They also propose three types of loss functions which learn from different parameters of the teacher, namely the output of the embedding layer, the hidden states and attention matrices and the logits output by the prediction layer. In choosing to learn direclty from the attention matrices the authors are inspired by the work of Clark et al. (2019)[6] which shows that the former can "substantial linguistic knowledge" [5]. With those aforementionned methods tinyBERT " achieves more than 96.8% the performance of its teacher BERT BASE on GLUE benchmark, while being 7.5x smaller and 9.4x faster on inference." [5]

Task and Dataset

The two framworks for evaluating performance we are considering are Squad[9] and GLUE[10]

Methods

General approach In light of the previous discussion we propose to experiment with different paradigms of model compression on top of the KD based one implemented by TinyBERT. The strategy outlined below is mainly inspired by Gupta et al [7]. We are considering augmenting the compression of TinyBERT through a mix of quantization and pruning. Gupta et al suggest that "Mixed-precision quantization combined with pruning is highly effective for Transformer based models." [7]

Pruning In the case of pruning it is recommended to do the process iteratively, over epochs during traing, this is called iterative or gradual pruning. Different patterns of controlling this process exist but they are independent of the categories of pruning we now describe and we privilege gradual over static pruning. Unstructured weight pruning (e.g. eliminating low magnitude weights) leads to difficulties in manipulating the resulting sparse data structures. Neuron pruning doesn't yield the same difficulty but is limited since we need to eliminate whole columns or rows of weight matrices. A more promising approach is to prune blocks which are contiguously stored in memory. The blocks to be pruned can be guided through group Lasso regularization. The latter has been already experimented, however it was targetting RNN based models and not transformers[8] We plan on first experimenting with simple structured block pruning and more intricate schemes describe in Gupta given we have enough time.

Quantization With regards to quantization while binary quantization does not work effectively for text-based Neural networks, ternary and higher-bit quantization "lead to significant model size reduction without loss in accuracy across tasks" [7]. Moreover more fancy, non-uniform, schemes can be used such as the ones based on KMeans or loss aware schemes. We plan on starting with uniform quantization and given time available experiment with non-uniform schemes.

Baselines and evaluation

The resulting compressed model will be first and foremost compared to TinyBERT and BERT. We will evaluate our models on two general criteria, performance on standardized tasks and memory and computational ressources required for training and inference at deployment.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[5] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. *CoRR*, abs/1909.10351, 2019.

[6] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of bert's attention. *CoRR*, abs/1906.04341, 2019.

[7] Manish Gupta and Puneet Agrawal. Compression of deep learning models for text: A survey, 2020.

[8] Sharan Narang, Eric Undersander, and Gregory F. Diamos. Block-sparse recurrent neural networks. *CoRR*, abs/1711.02782, 2017.

[9] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822, 2018.

[10] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. *CoRR*, abs/1808.05326, 2018.

Appendix

Architecture	Number of parameters
BERT	340M
GPT-2	1.5B
MegatronLM	8.3B
T5	11B
T-NLG	17B
GShard	600B

Table 1: PLMs and their sizes[7]

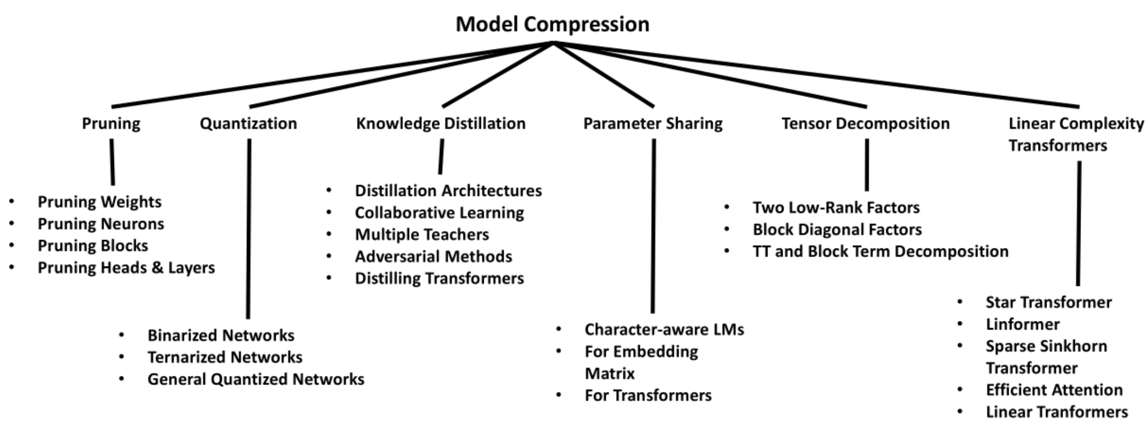

Figure 1: Model Compression Taxonomy[7]