TRANSFORMER MODEL COMPRESSION

Vineeth S

M.Tech Artificial Intelligence, SR No. 16543

ABSTRACT

In this work we explore model compression for transformer architectures, which would lead to reduced storage, memory footprint and compute power requirements. We show that transformer models can be compressed with no loss of or improved performance on the IWSLT English-German translation task. We specifically explore quantization aware training of the linear layers and demonstrate the performance for 8 bits, 4 bits, 2 bits and 1 bit quantization. We find that the linear layers of the attention network to be highly resilient to quantization and can be compressed aggressively.

Index Terms— transformers, model compression

1. INTRODUCTION

Transformer architectures and their extensions such as BERT, GPT etc, has revolutionized the world of Natural Language, Speech and Image processing[1][2][3]. The large number of parameters and the computation cost inhibits the transformer models to be deployed on edge devices such as smartphones. In this work, we explore the model compression for transformer architectures by quantization. Quantization not only reduces the memory footprint, but also improves energy efficiency. [4] has shown that 8 bit quantized model uses 4x lesser memory and 18x lesser energy. Linear layers constitute the majority of transformer architecture. Hence, we focus on compressing the linear layer of transformers.

2. TECHNICAL DETAILS

Transformer architecture consists of embedding, linear and layer normalization layers. We focus on quantizing the linear layers and embedding layers. We choose not to quantize the decoder generator linear layer of size 512×58790, which when quantized hurts the performance badly. The quantization and binarization operations are non-differentiable, hence we use straight-through-estimator (STE) proposed in [5] to approximate the derivative to identity function.

2.1. Binarized Training

Inspired by [6] and [7], we first explore training the transformer architecture by binarizing the weights during training.

The binarization function is given by $B(\mathbf{v})$

$$B(v_i) = sign(v_i)$$

2.2. Quantized Training

We propose a novel method for quantization-aware training. We use a modified version of the quantization function proposed in [8] originally proposed for gradient compression.

The quantization function is given by $Q_s(\mathbf{v})$, where s is a tunable parameter, corresponding to number of quantization levels. Let $0 \leq l < s$ be an integer such that, ${}^{|v_i|}/_{||v||_2} \in [{}^{l}/_{s}, {}^{l+1}/_{s}]$.

For $\mathbf{v} \neq \mathbf{0}$,

$$Q_s(v_i) = ||v||_2 \cdot sign(v_i) \cdot \xi_i(\mathbf{v}, s)$$

$$\xi_i(\mathbf{v}, s) = \begin{cases} {}^{l}/_{s}, & \text{with prob } 1 - p(\frac{|v_i|}{||v||_2}, s) \\ {}^{l+1}/_{s}, & \text{otherwise} \end{cases}$$

where $p(a, s) = as - l$ for $a \in [0, 1]$. For $\mathbf{v} = \mathbf{0}$, we define $Q_s(\mathbf{v}) = \mathbf{0}$. We can also note that $E[Q_s(v_i)] = v_i$.

3. RESULTS

We evaluate the baseline models and proposed quantization methods on IWSLT dataset. We use Bilingual Evaluation Understudy (BLEU) Score as our evaluation metric.

Model	BLEU Score
Base line	27.9
Binary Quantization (All Linear)	13.2
Binary Quantization (Attention Linear)	**26.87**
Quantized - 8 Bit (Attention Linear)	**29.83**
Quantized - 4 Bit (Attention Linear)	**29.76**
Quantized - 2 Bit (Attention Linear)	**28.72**
Quantized - 1 Bit (Attention Linear)	**24.32**
Quantized - 8 Bit (Attention + Embedding)	21.26

In the binary and 1 bit quantization, we have to modify the attention score equation by replacing $\sqrt{d}$ by $25d$ in order to prevent loss function from plateauing at a high value. The proposed method can also be used for post-training quantization with minimal performance loss ($< 1\%$) on pretrained BERT models. (Results are not shown due to lack of space).

4. CONTRIBUTIONS

We implement binary quantization and the proposed quantization method from scratch.

5. RESOURCES

Baseline Transformer Code: https://github.com/gordicaleksa/pytorch-original-transformer

Dataset: IWSLT CS Toronto http://www.cs.toronto.edu/~pekhimenko/tbd/datasets.html

Toolkits: PyTorch, Spacy, NLTK, Torchtext

6. 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," 2017.

[2] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5884–5888.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020.

[4] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 10–14.

[5] Yoshua Bengio, Nicholas Léonard, and Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013.

[6] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1," 2016.

[7] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David, "Binaryconnect: Training deep neural networks with binary weights during propagations," 2016.

[8] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," 2017.