Transformer Model Compression

Vineeth S
M. Tech Artificial Intelligence
SR. No. 16543

Background

- Transformer based architectures has revolutionized the domain of Natural Language, Speech and Image processing
- Edge deployed transformer models will provide a better experience to the user
- Large number of parameters and the huge computation cost inhibits the transformer models to be deployed on edge devices
- Model compression attacks this problem producing smaller and lite models
- Model compression types: Quantization, Pruning, and Knowledge distillation

Proposed Approach

- We explore model compression for transformer architectures by quantization
- Quantization reduces the memory footprint, and improves energy efficiency
- Quantization may also act as a regularizer when we are in low data regime
- We explore binarizing the weights during training (existing method for DNN)

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

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

- We propose a new method which can be used for quantization-aware training as well as post-training quantization

Technical Details

- Transformer architecture consists of embedding, linear and layer norm layers
- We focus on quantizing the linear layers and embedding layers
- Binarizing the decoder generator hurts the performance badly
- Quantization and Binarization operations are non-differentiable
- Use straight-through-estimator to approximate the derivative to identity function

$$\frac{\partial quantize}{\partial w} = \mathbb{1}, \frac{\partial binarize}{\partial w} = \mathbb{1}_{|W| \leq 1}$$

Contributions (Novelty)

- Method for quantization-aware training and post-training quantization
- Modified version of the function originally proposed for communication 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 $E[Q_s(v_i)] = v_i$.

Results & Conclusion

- We evaluate the baseline and quantized models on IWSLT dataset (BLEU Score)

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
Quantized - 8 Bit (All Linear)	**27.19**
Quantized - 4 Bit (All Linear)	**27.72**

- The proposed method can also be used for post-training quantization with minimal performance loss (< 1%) on pretrained BERT models on GLUE tasks.