LLMEasyQuant

An easy to use package for LLM quantization

Dong Liu* Chuyue Zhang*
*denotes equal contribution

Workshop on
CS 638 Large Language Models in Industry
University of Wisconsin-Madison
1 May 2024

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

LLMEasyQuant

- **Challenges:** Nowadays, packages like TensorRT and Quanto have many underlying structures and self-invoking internal functions, which are not conducive to developers' personalized development and learning for deployment.
- **Our solution:** LLMEasyQuant: We aim to develop a package for easy quantization deployment that is user-friendly and suitable for beginners' learning.

Quantization

What is Quantization?

Quantization is the process of mapping a large set of input values to a smaller set of output values, often integers. It is a key technique in digital signal processing where continuous signals are mapped to discrete digital values.

Quantization Formula

The quantization process can be described by:

$$Q(x) = \text{clamp}\left(\left\lfloor \frac{x - \min(X)}{\Delta} + 0.5 \right\rfloor + z, -128, 127\right)$$

where:

- x is a floating-point value from the dataset X,
- Δ (scale) is calculated as $\frac{\max(X) - \min(X)}{255}$,
- z (zero point) is -128 or calculated to shift the scale,
- clamp function ensures values are kept within the $[-128, 127]$ range.

LLMEasyQuant Functions

- absmax: scale $= \frac{127}{\max(|X|)}$
- zeropoint: Shifting the tensor values based on a computed zero-point
- smoothquant: A smoothing technique to the quantization process.Xiao et al. [2024]
- symquant: Symmetric scaling based on the absolute maximum value.yao2022zeroquant
- zeroquant: Adjust the numeric range of input data so that zero values in the original data can be represented exactly in the quantized format. Yao et al. [2022]
- simquant: A quantization technique by Scale and Zero Point Calculation. Hooper et al. [2024]

Simulation Quantization - `sim_quantize`

Quantization Process

- Calculate range values:

$$\text{vals}_{\text{min}} = \min(X_{\text{channel}}), \quad \text{vals}_{\text{max}} = \max(X_{\text{channel}})$$

- Compute scale s and zero point z:

$$s = \frac{2^{\text{bits}} - 1}{\text{vals}_{\text{max}} - \text{vals}_{\text{min}}}, \quad z = -\text{vals}_{\text{min}} \cdot s$$

- Apply quantization:

$$X_{\text{quant}} = \text{clamp}(\lfloor X \cdot s + z + 0.5 \rfloor, 0, 2^{\text{bits}} - 1)$$

Dequantization Process

$$X_{\text{dequant}} = \frac{X_{\text{quant}} - z}{s}$$

Symmetric 8-bit and Layer-by-Layer Quantization

Symmetric 8-bit Quantization

Given a tensor X, quantize to 8-bit integers:

$$
\begin{aligned}
s &= \frac{\max(|X|)}{127.5}, \\
X_{\text{quant}} &= \text{clamp}\left(\text{round}\left(\frac{X}{s}\right), -128, 127\right), \\
X_{\text{dequant}} &= X_{\text{quant}} \cdot s.
\end{aligned}
$$

Symmetric 8-bit and Layer-by-Layer Quantization

Layer-by-Layer ZeroQuant Function

Applies quantization per layer in a model:

- Encode input and compute unquantized outputs.
- For each layer i:
  - Freeze all but layer i.
  - Update i-th layer's parameters by quantization.
- Compute loss and update using:

$$\text{loss} = \text{MSE}(\text{teacher\_outputs}[i], \text{quantized\_outputs}[i])$$

- Optimize with gradient descent.

SmoothQuant

Algorithm

Given a tensor X and an activity scale act_scales, the smooth quantization is computed as follows:

- Compute weight scales weight_scales as the maximum absolute value per feature:

$$\text{weight\_scales} = \max(|X|)_{\text{column-wise}}$$

- Calculate scales s using:

$$s = \left(\frac{\text{act\_scales}^{\alpha}}{\text{weight\_scales}^{1-\alpha}} \right)$$

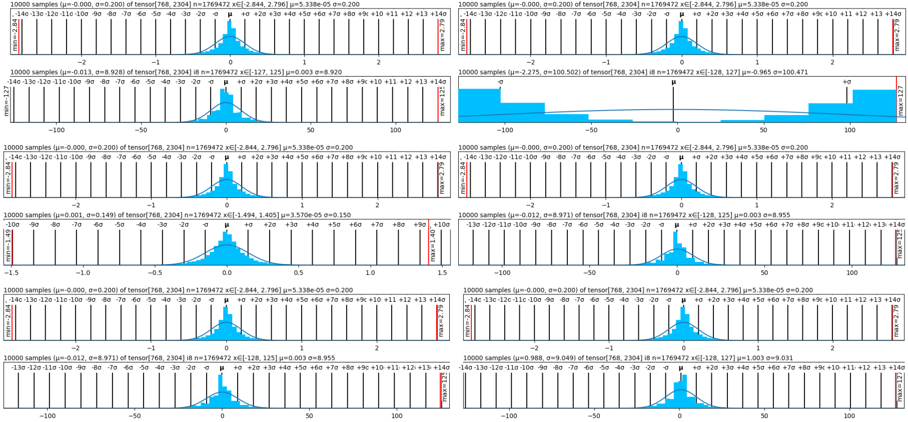
- Apply smoothing:

$$\text{smoothed\_X} = X \cdot s$$

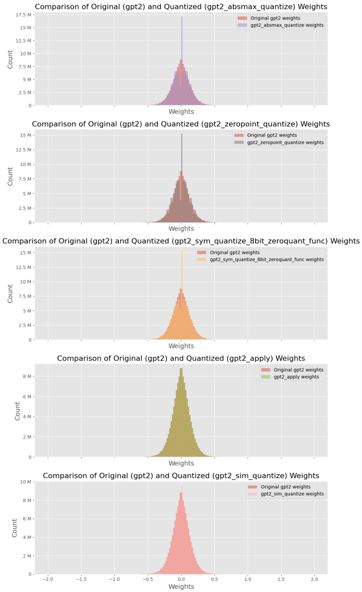
- Dequantize:

$$\text{dequantized\_X} = \frac{\text{smoothed\_X}}{s}$$

Quantized Numerical Representation



Weights Comparison



Perplexity Analysis

Models	Perplexity (ppl)
GPT-2	4.01
GPT-2 INT8	6.83
GPT-2 AbsMax Quantize	9.32
GPT-2 ZeroPoint Quantize	8.93
GPT-2 Smooth Quant Apply	6.31
GPT-2 Sim Quantize	7.16
GPT-2 Sym Quantize 8bit	7.01
GPT-2 Sym Quantize 8bit ZeroQuant Func	7.37

Table: Perplexity of Quantization Models

References I

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization, 2024.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models, 2024.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers, 2022.