ITC 2/54

Information Technology and Control
Vol. 54 / No. 2/ 2025
pp. 682-694
DOI 10.5755/j01.itc.54.2.40573

ORPTQ: An Improved Large Model Quantization Method Based on Optimal Quantization Range

Received 2025/02/19 | Accepted after revision 2025/04/19

HOW TO CITE: Tian, S., Huang, K. (2025). ORPTQ: An Improved Large Model Quantization Method Based on Optimal Quantization Range. *Information Technology and Control*, 54(2), 682-694. https://doi.org/10.5755/j01.itc.54.2.40573

ORPTQ: An Improved Large Model Quantization Method Based on Optimal Quantization Range

Shicen Tian, Kejie Huang

College of Information Science and Electronic Engineering, ZheJiang University, 38 Zheda Road, Hangzhou, 310027, Zhejiang Province, P. R. China; e-mails: tianshicen@zju.edu.cn, huangkejie@zju.edu.cn

Corresponding author: huangkejie@zju.edu.cn

Quantization reduces model storage by representing model in low bits. It can help to improve the application capability of transformer-based large models and make them possible to be deployed on resource-limited systems such as PCs and mobile devices. The best weight-only quantization method currently is to use second-order information to fine-tune the weight step by step during the quantization process, compensating for the quantization errors that have occurred. The method can minimize the functional loss of weight due to quantization by adjusting the remaining elements through algebraic transformations in each step. However, the performance of this quantization method will deteriorate rapidly when the adjustment for weight deviates too far from the starting point, especially in low-bit quantization (e.g. 4 bits or fewer). To meet the mathematical prerequisite of this method in the quantization, this paper introduces two parameters α, β to adjust the quantization range based on the second-order method, and presents three approaches to seek their optimal values. The experimental results show that the performance of the proposed method significantly outperforms the original second-order method in low-bit quantization. The code of this paper is available on github.com/t-scen/ORPTQ.

KEYWORDS: Large Model Quantization; Optimal Quantization Range; Transformer; GPTQ; ORPTQ

1. Introduction

In recent years, transformer-based large models have shown outstanding performance in fields such as natural language processing and image processing [1, 3, 20]. With increasing model parameters, these large models can handle some highly complex tasks [5, 26]. Although the pre-training model technology can alleviate the requirements of model training resources, large models still face problems in heavy reasoning and storage requirements, which limit their practical applications.

Model compression, aimed at reducing model deployment and application resource requirements, mainly includes model quantization, model pruning, and knowledge distillation.

Model quantization is a low-bit precision technology that stores floating-point model parameters in integer or smaller byte data types. It can significantly reduce storage requirements and speed up model reasoning when combined with specific hardware. The main focus of model quantization research is to minimize quantization loss while reducing storage size, which can be divided into quantization-aware training (QAT) technology and post-training quantization (PTQ) technology.

QAT quantizes the model during training with complete training data, where all weights and quantization parameters are optimized together. In this way, the model parameters can better adapt to the information loss caused by quantization. The performance of this technology is generally higher than that of PTQ [17, 25]. However, QAT needs to insert quantization nodes into the original model and retrain the model. Except for the QAT technology applied to BERT currently, there is little research on conducting QAT on large language models due to the huge training overhead [2, 7, 31].

PTQ is a quantization technology applied to pre-training models. It can avoid the high cost of the model training process and is easy to implement, making it more popular at present. PTQ can be further divided into weight quantization and activation quantization.

For transformer-based pre-trained models, the weight layers are fixed, while the activation layers are computed during inference and depend on the model inputs and weight layers. Activation quantization aims to reduce the memory overhead required for storing activation values and improve inference efficiency. To achieve this, it is essential to understand the distribution of activation values. Dynamic quantization collects activation values from hidden layers during the model inference process and computes their distribution in real time. In contrast, static quantization uses a calibration dataset to infer activation value distributions prior to inference, which are then used to quantize activation layers during actual inference. Handling outliers is a key technique for reducing activation quantization errors. Typical approaches include per-channel scaling transformations (e.g. SmoothQuant [29]), channel clustering rearrangement (e.g. RPTQ [30]), and adaptive channel reorganization (e.g. QLLM [15]).

Weight quantization is the primary task of model quantization and forms the foundation for activation quantization. Early weight quantization methods mainly focused on handling of outliers.

GPT.int8() found that a small number of very important outliers will appear when the model scale becomes larger. By detecting these outliers and handling them separately, quantization performance can be improved [6].

For handling outliers, Outlier Suppression [28] found that the gamma values of LayerNorm amplify outliers in the output and cause significant quantization errors and shifted them to the next module. Outlier Suppression+ [27] explored a more accurate outlier suppression solution with channel-level shifting and scaling functions, which helps to adjust the range of different activation channels and reduce outliers. However, while handling outliers can improve quantization performance, it also leads to additional storage overhead and increases computational complexity. Moreover, it is difficult to determine when this technique has reached its limit.

Optimization-based quantization can avoid the above problems. The optimal quantization can be achieved by minimizing the evaluation function. There are two types of evaluation function: one based on similarity and the other on functionality.

The similarity-based evaluation function calculates the similarity between the original weights and the quantized weights, whereas cosine similarity is used in most cases. The higher the similarity, the better the quantization. The functional evaluation function, which evaluates quantization based on its inference performance, is more direct and effective, making it the main evaluation function in optimization-based weight quantization.

The best weight-only quantization method currently is an optimization-based algorithm with functional evaluation on second-order information [9]. It quantizes the vector values per element and minimizes the functional loss of the vector due to quantization by adjusting the remaining elements. The underlying mathematical framework of this method is based on a second-order Taylor expansion and an approximation using the Hessian matrix, which assumes that the adjustment range must be confined to a small neighborhood around the starting point. However, the optimization conditions upon which this method is based are sensitive to the starting point, particularly in low-bit quantization. When the starting point deviates, the quantization performance will deteriorate rapidly.

In this paper, we introduce a set of parameters to adjust the quantization range based on the second-order method, ensuring a better starting point. We propose three approaches to obtain the optimal values for these parameters. Experimental results show that the quantization performance in low-bit quantization such as 4 bits or less can be notably improved by adjusting the quantization range in comparison with the baseline. The main contributions of the paper are:

1. Analyzing the reason for the performance degradation of the current second-order quantization method in low-bit quantization.
2. Proposing an optimization-based solution to the performance degradation and attempting three different optimization algorithms to solve the problem.

The rest of the paper is organized as follows. Section 2 reviews the related work on the second-order quantization algorithm. Section 3 describes the proposed method in detail. Section 4 presents the experimental design and results. Finally, Section 5 concludes the paper.

2. Related Work

Treating model quantization as an optimization problem is essential to study quantization systematically. In quantization of NN networks such as ResNet and Inception, Yoni et al. took the mean square deviation of the quantized weight multiplied by a scaling factor and the original weight as a loss function and obtained the optimal scaling factor through a one-dimensional accurate search [4]. Adaround [19] added a parameterized rounding function in quantization, taking the L2 norm of the quantized weight layer output and the original weight layer output as the loss function, and using the gradient descent method to minimize the loss function to obtain the rounding parameter.

GPTQ is one of the first methods that successfully used optimization technology to quantize the transformer-based large models. The core of GPTQ is to utilize second-order information. The basis of GPTQ is OBS [12], which is a model compression framework proposed in 1993. The basic principle of OBS is to greedily modify the weight vector per element and use second-order information to adjust the unmodified elements to compensate for the task loss of the modified elements. Compared with OBS, GPTQ first gives up the greedy strategy and thus avoids the sorting operation. Secondly, the per-element modification is improved to a per-batch modification, and the matrix operation is introduced to improve the efficiency of the algorithm. Furthermore, it also improves the inversion of the Hessian matrix. Currently, GPTQ has become one of the best weight quantization methods for large model optimization.

On the basis of GPTQ, SpQR [8] finds some characteristics of the location distribution of sensitive weight values through statistics and carries out special processing for sensitive value groups and individual outliers, and uses different precision to quantize the weight separately. OWQ [13] uses the product of the root mean square error of the weight vector and the diagonal elements of the corresponding Hessian matrix to filter the sensitive columns in the weight matrix, and quantizes and stores the sensitive columns separately. APTQ [11] takes a further step on GPTQ using the output of each attention layer instead of the output of the linear layer to adjust the weight. VPTQ [16] is oriented towards very low-bit quantization scenarios. The per-column scale matrix operation of GPTQ is changed to a vector operation, and the quantization results are clustered and saved in a codebook. On the basis of VPTQ, CLAQ [24] uses three different column-level strategies to improve the quantization performance of low bits. In addition, AWQ [14] uses the activation value to find the important weight and multiply it by the parameter scale, and uses the search method to determine the best scale.

These works have extended the original GPTQ from different parts. However, they also suffer some inherent disadvantages of second-order methods, where performance deteriorates when the adjustment exceeds the range. The paper analyzes the reasons for the performance degradation of the second-order methods and proposes solution based on regulating parameters to address it. OmnitQuant [22] similarly used the regulating parameters, but those parameters were used for different purposes.

3. Methods

3.1 Background

Transformer-based models are usually stacked by multiple encoder and decoder layers. For every encoder or decoder layers, they are mainly composed of multi-head attention blocks, feedforward blocks, normalized blocks, and activation blocks, etc. The linear components in these blocks, which contain trainable parameters, are of great importance. Let us take a look into a typical encoder layer with matrix input and output .

$$\text{out}=\text{layer}_{\text{norm}(\text{out}_1+\text{feedforward}(\text{out}_1))}, \tag{1}$$

where

$\text{out}_1=\text{activation}(\text{layer\_norm}(\text{in}+\text{multi\_head}(\text{in})))$.

The multi-head attention blocks and feedforward blocks contain principal linear components of a transformer-based model. The multi-head attention blocks are concatenated up with multiply attention heads and each attention head is a function of linear components such as Query, Key and Value layer. The feedforward blocks themselves are basic linear components. The values of the learnable parameters W of the linear component should be the best for a pre-trained transformer-based model to infer on target tasks and be stored fixedly. When a W is replaced with a quantization value W_q, the quantization error occurs and spreads forward, leading to a loss in model precision. Our goal is to minimize the loss to achieve good quantization performance. Cosine similarity was early used to measure the difference between W and W_q. However, a higher cosine similarity does not guarantee less model precision loss. In contrast, measuring the difference between the outputs of the linear components corresponding to W and W_q is more reasonable. This is because the output of a layer is the only data passed to the next layer, and the parameter W of a linear layer only affects the output of that layer.

By minimizing the difference between the outputs of the linear components with respect to W and W_q, the model precision loss can be reduced. For each layer, an optimization problem can be formulated as follows.

$$W_q = \underset{W_q}{\operatorname{argmin}} \left\| in \cdot W - in \cdot W_q \right\|_2 , \tag{2}$$

where

in denotes the input of the linear component;

W denotes the weight of the linear component, which has been trained;

W_q denotes the quantization value of W, which is the optimizable parameter in the problem.

One of the best approaches to this problem is the step-by-step optimization approach based on second-order information adopted by GPTQ. Let W_0 denote the weight value of a given linear component of a pre-trained model, which is the initial value of the parameter W. Let $L(W_0)$ denote the precision loss of the linear component with respect to W_0, and L(W) denote the precision loss of the linear component with respect to the parameter W, According to the functional Taylor series formula, the following Equation (3) is true when the value of W is close to W_0.

$$\text{Loss}=L(W)-L(W_0)=\frac{1}{2}\delta_w{}^T H\delta_w , \tag{3}$$

where

$\delta_W = W - W_0$, and H is the Hessian matrix respect to δ_W.

Solving Equation (2) is equivalent to minimizing the Loss in Equation (3), which is a convex optimization problem with a closed-form solution. To satisfy the prerequisites that the value of W must be close to W_0, the OBS algorithm sets the initial value of W to W_0 and modifies it gradually, making only minor changes each step. The GPTQ enhances the algorithm by replacing greedy-order quantization with an arbitrary order, introducing lazy batch updates, and employing a Cholesky reformulation, thus promoting the efficiency of the method to adapt for the transformer-based models. In theory, this method can achieve the ultimate performance of quantification.

However, it is essential to ensure that the prerequisites of Equation (3) are met in the quantization process. If the modified W in any step deviates too much from W_0, the GPTQ algorithm will not achieve the best quantization. Instead, it may get an even worse quantization than that of the standard quantization algorithm. When in low-bit quantization, such as 4 bits or below, the quantization errors become larger, and those deviations may occur commonly.

Based on the observation above, we introduce a parameterized quantization formula for W_0, By using optimization approaches, we make the quantized W as close as possible to W_0 to ensure quantization performance.

3.2 Method

The overview of our proposed approach is shown in Figure 1. For each linear layer of a transformer-based large model, we quantize the weight using a parameterized formula and obtain the output of that layer with quantization weight. By minimizing the loss of this output relative to the output of the same layer without quantization, we can obtain the optimal quantization parameters. Those optimal quantization parameters were used to quantize the weight of that linear layer in a stepwise way based on second-order information. The impact of parameters α and β on quantization performance is shown in the upper right corner of Figure 1. The parameters α and β are real values between [0, 1]. When multiplying them onto the min and max values of the weight respectively, the quantization range is reduced, resulting in higher quantization accuracy for each element in the new range. However, elements outside the new quantization range will be treated as outliers, resulting in lower quantization accuracy. This is a contradiction. We aim to achieve the best overall quantization performance by adjusting the parameters α and β. By introducing parameters α, β, we actually adjust the quantification range and suppress outliers in a weight.

The quantization formula of W_0 is defined in Equation (4) below.

$$W_0=\frac{W}{Scale}+zero \tag{4}$$

with

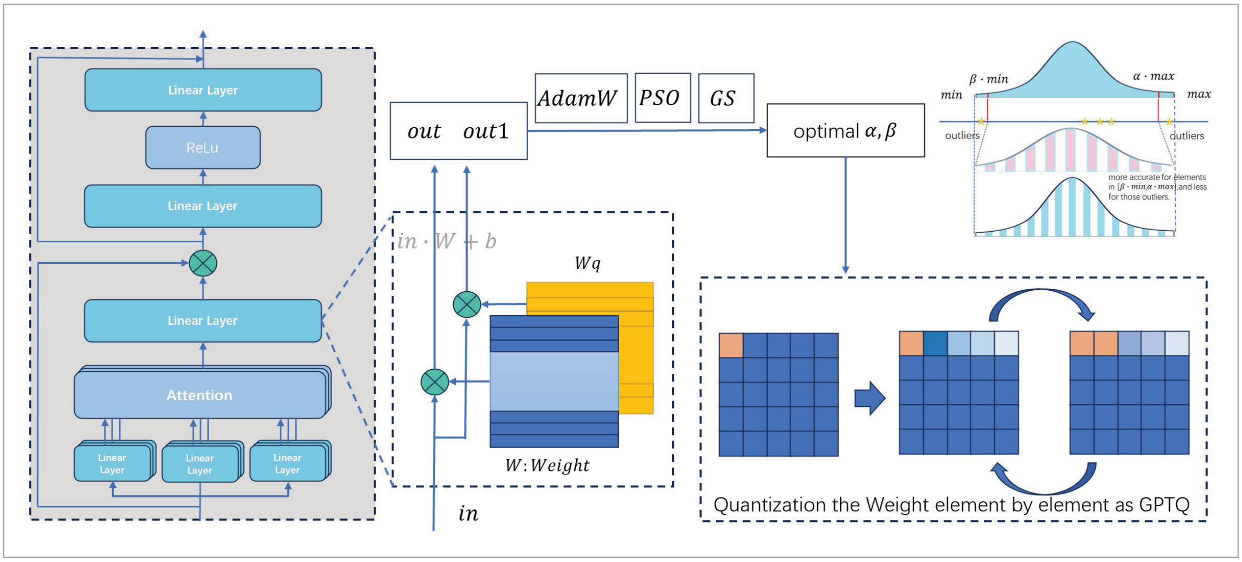
$$Scale=\frac{\alpha \cdot Max(W)-\beta \cdot Min(W)}{2^{wbits}-1},$$

$$zero=-\frac{Min(W)}{Scale}, \qquad \alpha,\beta \in (0,1)$$

As mentioned above, the best values of α, β are the values that minimize the deviation between W and W_0. To determine α, β, we define an optimization problem in the following.

Figure 1

The overview of the proposed approach.



$$\alpha,\beta = \arg\min_{\alpha,\beta} \|in \cdot W - in \cdot W_0\|_2 \qquad (5)$$

We present three approaches to solve the problem in Equation (5) and then obtain the best values of the parameters α and β: gradient-based optimization approach, PSO-based optimization approach, and Golden section linear search approach.

3.2.1 Gradient-based Optimization Approach

An optimization problem is defined for each linear layer of the model as in Equation (5), where W_0 is calculated using the formula in Equation (4). To make α and β within the bound of (0,1) and simplify the calculations, α and β are set to the value of a sigmoid function, respectively.

A small calibration dataset containing 128 samples is used to generate input for each linear layer. For a pretrained model, the calibration samples are fed into the first layer and the input of each linear layer is obtained using a hook technique. The input of each layer is then used to calculate two outputs: one corresponding to the original W, and the other corresponding to W_0. The model is temporarily set to full precision and an AdamW optimizer is employed to minimize MSELoss between the two outputs by the gradient. To make the gradient of the function *round*(x) calculable, a straight-through estimator is used.

3.2.2 PSO-based Optimization Approach

Particle swarm optimization (PSO) is a stochastic optimization algorithm that belongs to the class of evolutionary algorithms. It seeks for the global optimal solution by population cooperation without much requirements for functional form, which can be applied to a wide range of optimization problems.

We used the PSO approach to find the optimal values of α and β within the bounds of (0,1) by two steps. In the first step, we seek the optimal value of α by fixing the value of β to 1. In the second step, we seek the optimal value of β by fixing the value of α to the optimal value obtained in the first step. In each step, a population of particles is randomly generated within the interval (0,1) uniformly. Then each particle is evaluated and updated according to the PSO formula iteratively. The evaluation function is defined as MSELoss between the two outputs similarly. We seek optimal values over the inputs provided by the calibration samples and use the averages as the final optimal values.

3.2.3 Golden-Section Linear Search Approach

The Golden-Section linear search algorithm is a one-dimensional search algorithm designed for unconstrained optimization problems. As mentioned above, one-dimensional search algorithm was used in NN quantization [17]. The core of the Golden-Section linear search algorithm is to reduce the search interval according to the Golden-Section ratio. The new search interval is then determined on the basis of a comparison of function values, and the search range is progressively narrowed until the termination conditions are met. This algorithm is applicable to any unimodal function, whether continuous or differentiable. If the objective function in this work is a unimodal function, this approach will theoretically be the most efficient.

4. Experiments

We conducted several experiments to verify the effectiveness of the α and β parameters and the optimization approaches. Since techniques on high-bit quantization such as 8 bits and higher are already very good, the experiments in this work mainly focus on low-bit quantization such as 4 bits, 3 bits and 2 bits.

4.1 Experimental Setup

Models.

We choose opt [32] and Llama-2 [23] models to carry out the experiment. Those two models are typical transformer-based large language models that are used extensively to evaluate quantization techniques.

Both OPT and Llama have multiple versions, varying in the number of parameters, but all versions consist of the same basic modules. For convenience, we used the OPT-6.7B and Llama2-7B versions in the experiments.

Evaluation.

We evaluate the quantization techniques on the two models by reporting the perplexity of the language generation experiments, as done in previous works [8, 9]. The perplexity scores are tested on Wiki-

Text2[18] and C4 [21]. 128 samples from C4 are also chosen as calibration data for the GPTQ method, as the original GPTQ does. In this paper, we also used those 128 samples as input data to obtain the optimal values of the parameters α and β. We use the lm-evaluation-harness [10] to evaluate the accuracy of the model, and four tasks such as ARC-Easy, ARC-Challenge, PIQA, WINOGRANDE are selected.

Baselines.

The original GPTQ method is chosen as the baseline to compare with the proposed technique. And the plain vector-wise quantization method is used as another baseline, which quantizes the weight asymmetrically.

Implementation details.

In the gradient-based optimization approach (AdamW-GPTQ), the learning rate of the AdamW optimizer is set to 0.01, and the training epoch is set to 3. The initial values of the parameters α and β are set to sigmoid(4).

In the PSO-based optimization approach (PSO-GPTQ), the velocity of a particle is determined by three parts: the original velocity of the particle, the distance between the current position of the particle and the best position of all particles. The factor of the first part is ω, which varies by iterations from an initial value to an end value. The initial value is set to 0.9 and the end value is set to 0.4 to let the importance of the original velocity of the particle decrease gradually. The factors of the second part and the third part are set to a random decimal in [0,1], divided by 5.0 to make them balanced. The iteration loops are set to 8.

In the Golden-Section linear search approach (GS-GPTQ), the interval gradually decreases from the initial interval by the golden ratio until the interval is less than the threshold or the maximum iteration times is reached. The threshold is set to 0.05 and the maximum iteration loops is set to 20.

4.2 Experimental Results

We first test the perplexity performance of the three approaches in low-bit quantization of 4 bits, 3 bits, and 2 bits, respectively, and compare them with the original GPTQ algorithm. This experiment is done to verify the improvement of the generation performance of proposed approaches in low-bit quantization. The above three approaches combined with the primary quantization algorithm, are denoted as “AdamW-GPTQ”, “PSO-GPTQ”, and “GS-GPTQ”, respectively. The results are shown in Table 1 below.

We test accuracy performance of the three approaches in low-bit quantization of 4 bits, 3 bits, and 2 bits, respectively, and compare them with the original GPTQ algorithm. This experiment is done to verify the improvement of the prediction performance of proposed approaches in low-bit quantization. The results are shown in Table 2 below.

Table 1

Perplexity scores of proposed approaches compared with original GTPQ.

Bits	Method	OPT-Wiki	OPT-C4	Llama-Wiki	Llama-C4
4	GPTQ	11.40	12.52	6.09	7.24
	AdamW-GPTQ	**11.18**	**12.42**	**6.06**	7.18
	PSO-GPTQ	11.38	12.50	6.12	**7.14**
	GS-GPTQ	11.55	12.60	9.04	15.62
3	GPTQ	15.05	16.21	10.17	10.59
	AdamW-GPTQ	14.84	15.46	8.08	9.95
	PSO-GPTQ	**13.51**	**14.61**	**8.05**	**8.90**
	GS-GPTQ	14.76	15.87	9.35	13.88
2	GPTQ	2867.96	204.46	3923.19	916.15
	AdamW-GPTQ	**105.88**	85.26	1586.72	295.98
	PSO-GPTQ	180.58	**60.46**	4178.87	251.48
	GS-GPTQ	163.06	69.67	**194.12**	**20.83**

Table 2

Accuracy scores of proposed approaches compared with original GTPQ.

Bits	Method	ARC-Easy		ARC-Challenge		PIQA		WINORANDE		Average
		Opt	Llama	Opt	Llama	Opt	Llama	Opt	Llama	
4	GPTQ	0.641	0.7471	0.2995	0.4113	0.7552	0.7617	0.6298	0.6827	0.616
	AdamW-GPTQ	0.6549	0.7172	0.2986	0.384	0.7601	0.7699	0.6551	0.778	**0.6272**
	PSO-GPTQ	0.6423	0.7437	0.3029	0.4104	0.7552	0.7758	0.6448	0.6914	0.6208
	GS-GPTQ	0.6406	0.6178	0.2944	0.3225	0.753	0.6779	0.6527	0.6806	0.5799
3	GPTQ	0.6098	0.6599	0.2782	0.343	0.7334	0.7231	0.6022	0.6219	0.5714
	AdamW-GPTQ	0.6233	0.6822	0.2892	0.3464	0.747	0.7427	0.6267	0.6504	0.5884
	PSO-GPTQ	0.6174	0.7155	0.2799	0.3797	0.7476	0.753	0.6227	0.6472	**0.5954**
	GS-GPTQ	0.6115	0.5774	0.2841	0.2875	0.7481	0.6703	0.6219	0.618	0.5524
2	GPTQ	0.303	0.2652	0.186	0.215	0.5582	0.5098	0.5012	0.4902	0.3786
	AdamW-GPTQ	0.3965	0.2559	0.1928	0.2201	0.611	0.5185	0.6083	0.4728	0.4095
	PSO-GPTQ	0.3775	0.2559	0.1928	0.2201	0.611	0.5185	0.6083	0.4728	0.4095
	GS-GPTQ	0.4205	0.4373	0.209	0.2159	0.6219	0.6284	0.5138	0.5351	**0.4477**

From Table 1 and Table 2, we can see that the proposed approaches almost exceed the baseline in all three types of quantization in both perplexity and accuracy. The best score outperforms the baseline in all three types of quantization. The perplexity of 4 bits, 3 bits and 2 bits gains average decreases with 1.21%, 13.36% and 95.18%, respectively, and the accuracy gains average increases with 1.82%, 4.20% and 18.25%, respectively.

We also test the performances of the three approaches without GPTQ in low-bit quantization and compare them with the primary quantization algorithm. The purpose of the experiment is to conduct ablation experiment. By comparing with the previous experiments, it can be seen the effectiveness of the GPTQ and the proposed method in this paper.

The formula of the primary quantization algorithm, which is denoted as "Plain", is shown below.

$$W_q=\frac{W}{Scale}+zero \qquad (6)$$

with

$$Scale=\frac{\alpha\cdot Max(W)-\beta\cdot Min(W)}{2^{wbits}-1},$$

$$zero=-\frac{Min(W)}{Scale}, \qquad \alpha,\beta\in(0,1)$$

The values of α and β are both set to 1 for the primary quantization algorithm. The three approaches combined with the primary quantization algorithm are denoted as "AdamW-Plain", "PSO-Plain" and "GS-Plain" respectively. The values of α and β for those approaches are determined by the optimization algorithm of their own.

We test the perplexity performance of the three approaches in low-bit quantization of 4 bits, 3 bits, and 2 bits, respectively in this setting to verify the generation performance of those approaches in low-bit quantization. The results are shown in Table 3 below.

We test accuracy performance of those approaches in low-bit quantization of 4 bits, 3 bits, and 2 bits, respectively to verify the of the prediction performance of those approaches. The results are shown in Table 4 below.

From Tables 3-4 we can see that the overall performances of the methods combined with the "Plain" quantization algorithm are lower than those combined with the "GPTQ" quantization algorithm. Meanwhile, in the "Plain" quantization settings, the proposed method also outperforms the base-line in all three types of quantization. The perplexity of 4 bits, 3 bits and 2 bits gains average decreases with 4.18%, 98.03% and 46.37%, respectively, and the accuracy gains average increases with 1.13%, 25.49% and 0.11%, respectively.

Table 3

Perplexity scores of approaches with the primary quantization algorithm.

Bits	Method	OPT-Wiki	OPT-C4	Llama-Wiki	Llama-C4
4	Plain	12.10	13.84	6.12	7.61
	AdamW-Plain	**11.57**	**13.10**	**5.97**	**7.37**
	PSO-Plain	11.62	13.23	5.99	7.41
	GS-Plain	11.91	13.37	10.66	16.71
3	Plain	5796.67	4669.35	524.91	413.01
	AdamW-Plain	**74.63**	**109.47**	**17.80**	**23.73**
	PSO-Plain	1918.43	1690.30	22.04	29.84
	GS-Plain	154.17	226.63	39.37	64.75
2	Plain	28366.31	13320.24	**17792.30**	29628.09
	AdamW-Plain	**7095.70**	**4668.29**	116479.00	131214.78
	PSO-Plain	18262.62	10168.82	32986.38	37461.42
	GS-Plain	9233.38	5330.62	18573.03	**17452.04**

Table 4

Accuracy scores of approaches with the primary quantization algorithm.

Bits	Method	ARC-Easy		ARC-Challenge		PIQA		WINORANDE		Average
		Opt	Llama	Opt	Llama	Opt	Llama	Opt	Llama	
4	Plain	0.6553	0.7403	0.291	0.413	0.759	0.7671	0.6417	0.6827	0.6188
	AdamW-Plain	0.6515	0.7521	0.3097	0.4258	0.7557	0.7753	0.6472	0.689	0.6258
	PSO-Plain	0.6406	0.75	0.3003	0.4232	0.7541	0.7758	0.6433	0.6764	0.6205
	GS-Plain	0.6431	0.7096	0.3063	0.3712	0.7557	0.7476	0.6361	0.6361	0.6007
3	Plain	0.2567	0.3476	0.2116	0.2031	0.5277	0.58	0.5099	0.5201	0.3946
	AdamW-Plain	0.487	0.5661	0.2005	0.2611	0.6415	0.6703	0.5209	0.5675	0.4894
	PSO-Plain	0.3022	0.5118	0.1954	0.2628	0.549	0.6714	0.4783	0.562	0.4416
	GS-Plain	0.4188	0.6242	0.1962	0.302	0.6115	0.7138	0.513	0.5817	0.4952
2	Plain	0.2639	0.2601	0.2167	0.2227	0.512	0.5223	0.4933	0.5154	0.3758
	AdamW-Plain	0.2576	0.2521	0.2142	0.2201	0.5196	0.5272	0.4996	0.498	0.3736
	PSO-Plain	0.2622	0.2639	0.2065	0.2184	0.5131	0.5174	0.4886	0.4783	0.3686
	GS-Plain	0.2618	0.2555	0.2065	0.2227	0.5223	0.5316	0.4941	0.5162	0.3763

By comparing with Tables 1-2, it can be seen that the GPTQ-based quantization methods generally perform much better than the primary quantization methods, which indicates that GPTQ has significant advantages over primary quantization algorithms. By introducing the adjustment parameters α and β, the performance of GPTQ is further improved. Figure 2 shows the performance scores of Plain, GPTQ, AdamW-GPTQ and the best score of the three optimization approaches such as AdamW-GPTQ, PSO-GPTQ and RS-GPTQ. It can be seen that the performance improvements from primary quantization approach to GPTQ, and from GPTQ to AdamW-GPTQ. And the fewer quantization bits, the more significant the improvement.

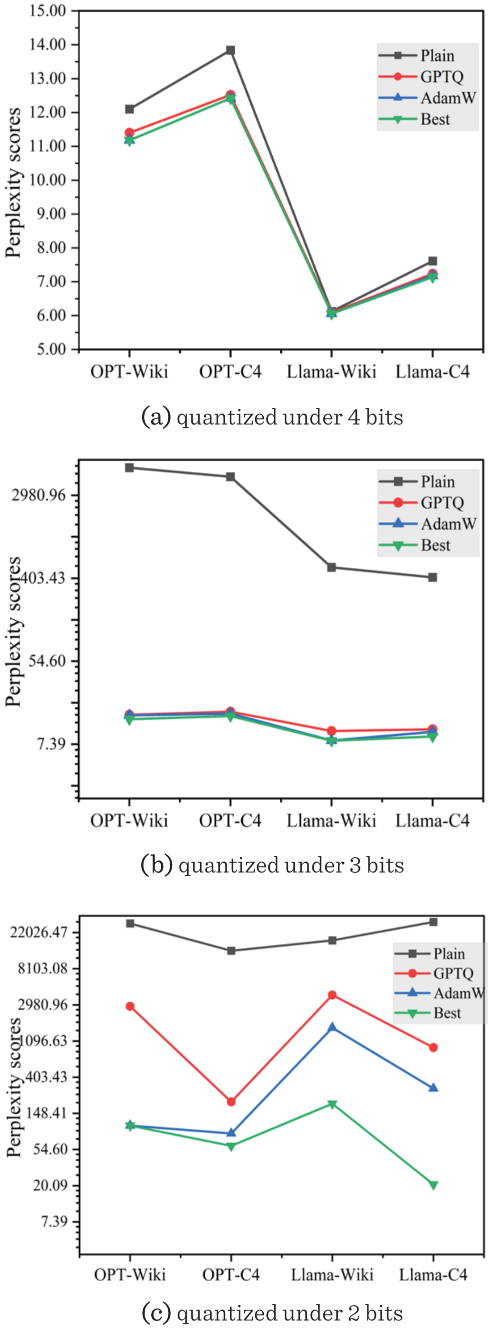
The experimental results show that the second-order information-based quantization method has many advantages over those primary asymmetric quantization methods. The proposed approaches can improve the performance of both second-order quantization methods and primary asymmetric quantization methods. In the case of combining with the second-order quantization method, the degree of performance improvement increases as the number of quantization bits decreases, which shows the advantage of the combination of the optimal quantization range and the GPTQ. As being analyzed above, although second-order quantization is an optimization-based quantization method, its quantization performance will be poor when the quantized weight value deviates too much from the starting point. The results confirm the aforementioned viewpoint that the second-order quantization method cannot reach the optimal state for the prerequisites of the method are not met. By introducing parameters α and β to adjust the quantization range, the proposed approaches work together with the second-order quantization method to make the optimization process with a better starting point at each step. This complementarity improves the performance.

From the perspective of the optimization approaches, as in the "GPTQ" settings, we can see that the performance of the AdamW-GPTQ approach exceeds the baseline in all three types of quantization. The PSO-GPTQ approach performs well in most cases, especially in 3-bit quantization, where it gets the best score. However, in some cases, its performance is worse than the baseline. GS-GPTQ approach also exceeds the baseline in some cases, while its performance is more unstable. In the "Plain" settings, AdamW-Plain gets the best score in most cases of the three types of quantization. AdamW-Plain and PSO-Plain perform better than baseline in all cases. GS-Plain performs better than baseline in most cases. However, in some cases, it performs worse than the baseline. These results show that the objective function in Equation (5) is not a unimodal one but a much more complicated one. The gradient-based optimization approach and the PSO-based optimization approach are more suitable for that problem.

From the quantization performance of the three optimization approaches, it can be seen that some good perplexity scores are obtained by GS-GPTQ. It is not

Figure 2

Comparison of the performance of Plain, GPTQ, AdamW-GPTQ, and the Best.
*(b) and (c) use a logarithmic scale on the y-axis (ln scale). The perplexity scores for "Best" in the figures are derived from the best results of the three solution methods described above.



(a) quantized under 4 bits

(b) quantized under 3 bits

(c) quantized under 2 bits

clear why other approaches are not able to obtain those good scores. But the most likely scenario is that all three approaches only obtain relative optimal values, not global optimal values. Although the proposed approaches achieve better performance than the baseline, there is still a requirement to find better optimization methods in the future. The experimental results also reflect some overfitting about the optimized value that is not positively correlated with the quantization performance, which needs further study.

As a weight-only quantization approach, the proposed method can be easily integrated into other quantization technologies without negative impacts.

5. Conclusion

Second-order information-based quantization algorithm is one of the best weight-only quantization algorithms, which has been widely concerned and many algorithms have been developed on it. In response to the problem that the algorithm may experience a decrease in quantization performance due to not meeting optimization conditions, this paper introduces a set of parameters α and β to adjust the quantization range in the GPTQ algorithm and proposes three approaches to obtain the optimal values of α and β. The experimental results show that the optimal quantization range can further improve the quantization performance on the basis of the original GPTQ algorithm.

There are still some issues need to be further addressed related to the paper. Specifically, how to check whether the quantized value is within the neighborhood during the step-by-step quantization process; Is there a method to best handle those quantized values that deviate from the neighborhood to maintain the second-order optimization condition. In addition, there is still much room for improvement in optimization algorithms. Optimization-based quantitative approaches are indisputable to have excellent development prospects, and these problems are expected to be addressed in forthcoming research.

Acknowledgement

Not applicable.

Funding Statement

This research was funded by Vivo Information Technology Joint Research Center, grant number K20213641(H.); and by National Innovative Training Program for Undergraduates, grant number 202410335120(T.).

Author Contributions

The authors confirm contribution to the paper as follows: Conceptualization, Kejie Huang and Shicen Tian; methodology, Shicen Tian; software, Shicen Tian; writing—original draft preparation, Shicen Tian; writing—review and editing, Kejie Huang; supervision, Kejie Huang; project administration, Kejie Huang; funding acquisition, Kejie Huang and Shicen Tian. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials

The data that support the findings of this study are openly available in [huggingace] at [https://huggingface.co/datasets/allenai/c4] and [https://huggingface.co/datasets/mindchain/wikitext2].

Ethics Approval

Not applicable.

Conflicts of Interest

The authors declare no conflicts of interest to report regarding the present study.

References

1. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W. Pre-Trained Image Processing Transformer. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, 12299-12310. https://doi.org/10.1109/CVPR46437.2021.01212
2. Chen, J., Bai, S., Huang, T., Wang, M., Tian, G., Liu, Y. Data-Free Quantization Via Mixed-Precision Compensation Without Fine-Tuning. Pattern Recognition, 2023, 143, Article 109780. https://doi.org/10.1016/j.patcog.2023.109780

3. Chen, Z., Xu, L., Zheng, H., Chen, L., Tolba, A., Zhao, L., Yu, K., Feng, H. Evolution and Prospects of Foundation Models: From Large Language Models to Large Multimodal Models. Computers, Materials & Continua, 2024, 80(2). https://doi.org/10.32604/cmc.2024.052618

4. Choukroun, Y., Kravchik, E., Yang, F., Kisilev, P. Low-Bit Quantization of Neural Networks for Efficient Inference. 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019, 3009-3018. https://doi.org/10.1109/ICCVW.2019.00363

5. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al. PaLM: Scaling Language Modeling with Pathways. Journal of Machine Learning Research, 2023, 24(240), 1-113.

6. Dettmers, T., Lewis, M., Belkada, Y., Zettlemoyer, L. GPT3.Int8(): 8-Bit Matrix Multiplication for Transformers at Scale. Advances in Neural Information Processing Systems, 2022, 35, 30318-30332.

7. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L. QLoRA: Efficient Fine-Tuning of Quantized LLMs. Advances in Neural Information Processing Systems, 2024, 36.

8. Dettmers, T., Svirschevski, R., Egiazarian, V., Kuznedelev, D., Frantar, E., Ashkboos, S., Borzunov, A., Hoefler, T., Alistarh, D. SPQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression. 12th International Conference on Learning Representations (ICLR), 2024.

9. Frantar, E., Ashkboos, S., Hoefler, T., Alistarh, D. OPTQ: Accurate Post-Training Quantization for Generative Pre-Trained Transformers. 11th International Conference on Learning Representations (ICLR), 2023.

10. Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., Zou, A. A Framework for Few-Shot Language Model Evaluation. Zenodo, 2024, Article 12608602.

11. Guan, Z., Huang, H., Su, Y., Huang, H., Wong, N., Yu, H. APTQ: Attention-Aware Post-Training Mixed-Precision Quantization for Large Language Models. Proceedings of the 61st ACM/IEEE Design Automation Conference, 2024, 1-6. https://doi.org/10.1145/3649329.3658498

12. Hassibi, B., Stork, D. G., Wolff, G. J. Optimal Brain Surgeon and General Network Pruning. IEEE International Conference on Neural Networks, 1993, 293-299. https://doi.org/10.1109/ICNN.1993.298572

13. Lee, C., Jin, J., Kim, T., Kim, H., Park, E. OWQ: Outlier-Aware Weight Quantization for Efficient Fine-Tuning and Inference of Large Language Models. Proceedings of the AAAI Conference on Artificial Intelligence, 2024, 38, 13355-13364. https://doi.org/10.1609/aaai.v38i12.29237

14. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., Han, S. AWQ: Activation-Aware Weight Quantization for On-Device LLM Compression and Acceleration. Proceedings of Machine Learning and Systems, 2024, 6, 87-100.

15. Liu, J., Gong, R., Wei, X., Dong, Z., Cai, J., Zhuang, B. QLLM: Accurate and Efficient Low-Bit Width Quantization for Large Language Models. 12th International Conference on Learning Representations (ICLR), 2024.

16. Liu, Y., Wen, J., Wang, Y., Ye, S., Zhang, L.L., Cao, T., Li, C., Yang, M. VPTQ: Extreme Low-Bit Vector Post-Training Quantization for Large Language Models. arXiv Preprint, 2024, arXiv:2409.17066. https://doi.org/10.18653/v1/2024.emnlp-main.467

17. Liu, Z., Oguz, B., Zhao, C., Chang, E., Stock, P., Mehdad, Y., Shi, Y., Krishnamoorthi, R., Chandra, V. LLM-QAT: Data-Free Quantization Aware Training for Large Language Models. Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), 2024, 467-484. https://doi.org/10.18653/v1/2024.findings-acl.26

18. Merity, S., Xiong, C., Bradbury, J., Socher, R. Pointer Sentinel Mixture Models. 5th International Conference on Learning Representations (ICLR), 2017. https://doi.org/10.48550/arXiv.1609.07843

19. Nagel, M., Amjad, R.A., Van Baalen, M., Louizos, C., Blankevoort, T. Up or Down? Adaptive Rounding for Post-Training Quantization. International Conference on Machine Learning (ICML), 2020, 7197-7206. https://doi.org/10.48550/arXiv.2004.10568

20. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I. Learning Transferable Visual Models from Natural Language Supervision. International Conference on Machine Learning (ICML), 2021, 8748-8763. https://doi.org/10.48550/arXiv.2103.00020

21. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research,

2020, Vol. 21(140), 1-67. https://doi.org/10.48550/arXiv.1910.10683

22. Shao, W., Chen, M., Zhang, Z., Xu, P., Zhao, L., Li, Z., Zhang, K., Gao, P., Qiao, Y., Luo, P. OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models. 12th International Conference on Learning Representations (ICLR), 2024. https://doi.org/10.48550/arXiv.2308.13137
23. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. LLaMA2: Open Foundation and Fine-Tuned Chat Models. arXiv Preprint, 2023. https://doi.org/10.48550/arXiv.2307.09288
24. Wang, H., Liu, B., Shao, H., Xiao, B., Zeng, K., Wan, G., Qian, Y. CLAQ: Pushing the Limits of Low-Bit Post-Training Quantization for LLMs. arXiv Preprint, 2024. https://doi.org/10.48550/arXiv.2405.17233
25. Wang, Z., Wang, C., Xu, X., Zhou, J., Lu, J. QuantFormer: Learning Extremely Low-Precision Vision Transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(7), 8813-8826. https://doi.org/10.1109/TPAMI.2022.3229313
26. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q.V., Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Advances in Neural Information Processing Systems (NeurIPS), 2022, 35, 24824-24837. https://doi.org/10.48550/arXiv.2201.11903
27. Wei, X., Zhang, Y., Li, Y., Zhang, X., Gong, R., Guo, J., Liu, X. Outlier Suppression+: Accurate Quantization of Large Language Models by Equivalent and Optimal Shifting and Scaling. 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2023, 1648-1665. https://doi.org/10.18653/v1/2023.emnlp-main.102
28. Wei, X., Zhang, Y., Zhang, X., Gong, R., Zhang, S., Zhang, Q., Yu, F., Liu, X. Outlier Suppression: Pushing the Limit of Low-Bit Transformer Language Models. Advances in Neural Information Processing Systems (NeurIPS), 2022, 35, 17402-17414. https://doi.org/10.48550/arXiv.2209.13325
29. Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., Han, S. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. International Conference on Machine Learning (ICML), 2023, 38087-38099. https://doi.org/10.48550/arXiv.2211.10438
30. Yuan, Z., Niu, L., Liu, J., Liu, W., Wang, X., Shang, Y., Sun, G., Wu, Q., Wu, J., Wu, B. RPTQ: Reorder-Based Post-Training Quantization for Large Language Models. arXiv Preprint, 2023. https://doi.org/10.48550/arXiv.2304.01089
31. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., et al. OPT: Open Pre-Trained Transformer Language Models. arXiv Preprint, 2022. https://doi.org/10.48550/arXiv.2205.01068
32. Zhao, C., Hua, T., Shen, Y., Lou, Q., Jin, H. Automatic Mixed-Precision Quantization Search of BERT. International Joint Conference on Artificial Intelligence (IJCAI), 2021, 3427-3433. https://doi.org/10.24963/ijcai.2021/472