

APTQ: Attention-aware Post-Training Mixed-Precision Quantization for Large Language Models

Ziyi Guan^{1,2*}, Hantao Huang^{1*}, Yupeng Su¹, Hong Huang¹, Ngai Wong², Hao Yu¹
School of Microelectronics, Southern University of Science and Technology, Shen Zhen, China¹
Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong, China²

ABSTRACT

Large Language Models (LLMs) have greatly advanced the natural language processing paradigm. However, the high computational load and huge model sizes pose a grand challenge for deployment on edge devices. To this end, we propose APTQ (Attention-aware Post-Training Mixed-Precision Quantization) for LLMs, which considers not only the second-order information of each layer’s weights, but also, for the first time, the nonlinear effect of attention outputs on the entire model. We leverage the Hessian trace as a sensitivity metric for mixed-precision quantization, ensuring an informed precision reduction that retains model performance. Experiments show APTQ surpasses previous quantization methods, achieving an average of 4 bit width a 5.22 perplexity nearly equivalent to full precision in the C4 dataset. In addition, APTQ attains state-of-the-art zero-shot accuracy of 68.24% and 70.48% at an average bitwidth of 3.8 in LLaMa-7B and LLaMa-13B, respectively, demonstrating its effectiveness to produce high-quality quantized LLMs.

CCS CONCEPTS

• Computing methodologies → Natural language generation.

KEYWORDS

Large Language Models, quantization, mixed-precision quantization, attention-based quantization, Hessian matrix sensitivity

ACM Reference Format:

Ziyi Guan^{1,2*}, Hantao Huang^{1*}, Yupeng Su¹, Hong Huang¹, Ngai Wong², Hao Yu¹. 2024. APTQ: Attention-aware Post-Training Mixed-Precision Quantization for Large Language Models. In *61st ACM/IEEE Design Automation Conference (DAC '24)*, June 23–27, 2024, San Francisco, CA, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3649329.3658498>

1 INTRODUCTION

Large Language Models (LLMs), such as ChatGPT [14], OPT [19], LLaMA [17], etc., exhibit impressive performance across various tasks. However, deploying these models on edge devices is challenging due to their exorbitant computational demands and memory

footprints. Existing model compression solutions such as pruning [1] and neural architecture search [2] often require model re-training, which is extremely time-consuming and expensive for billion-parameter models. Recently, post-training quantization (PTQ) methods, such as GPTQ [6], have been proposed and achieved relatively high accuracy without retraining. However, GPTQ only considers the weight quantization strategy in the scope of a single layer as an optimization problem to minimize $\|WX - \hat{W}X\|_2^2$, with W , \hat{W} and X representing float weights, quantized weights and inputs, respectively. This simplification fails to consider the complex and nonlinear effects such as softmax in the attention computation, and leads to a sub-optimal solution.

To achieve lower bitwidths without sacrificing the accuracy on edge devices, this paper presents an Attention-aware Post-Training Mixed-Precision Quantization (APTQ) technique, which is designed to consider the quantization optimization problem within the scope of the attention block including the nonlinear softmax operation. Specifically, APTQ utilizes gradients derived from the attention output and develops a second-order Hessian optimization strategy to quantize the weights. By doing so, APTQ significantly reduces the quantization error in these crucial components, thereby preserving the model’s integrity throughout compression.

Furthermore, APTQ proposes a novel Hessian trace-based quantization sensitivity metric to implement mixed-precision quantization to further compress LLM models. This approach judiciously applies varying bitwidths across the model parameters to fit the limited memory size on edge devices with balanced size and accuracy. As a result, APTQ constitutes a mixed-precision 2/4-bit hybrid scheme with performance comparable to a uniform 4-bit representation. In particular, APTQ produces a compressed model close to its full-precision counterpart, and outperforming the GPTQ method especially in the realm of ultra-low-bit quantization scenarios. Through comprehensive experiments on the LLaMA-7B and LLaMA-13B models [17], the effectiveness of APTQ is validated on both perplexity and zero-shot performance, thus entailing a viable solution for the deployment of LLMs on edge devices.

The main contributions of this paper are threefold:

- This is the first work to quantize LLMs by integrating the attention-based gradients with second-order Hessian optimization, leading to a nuanced update mechanism that enhances the precision throughout the quantization process.
- An innovative Hessian trace-driven mixed-precision quantization scheme is proposed that judiciously allocates high/low bitwidths across different layers based on their sensitivity, optimizing model performance while maintaining efficiency.
- Through extensive experimentation on the LLaMa models, APTQ not only achieves state-of-the-art (SOTA) results on

*Equal contribution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DAC '24, June 23–27, 2024, San Francisco, CA, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0601-1/24/06...\$15.00

<https://doi.org/10.1145/3649329.3658498>

the C4 dataset [15] but also attains near full-precision perplexity at an average quantization of 4 bits. In zero-shot tasks, APTQ also demonstrates superior performance compared to the SOTA approaches.

2 RELATED WORK

To deploy large models on edge devices, quantization is a versatile technique for reducing model size and computation. Quantization-Aware Training (QAT) is known to be effective by integrating the quantization process into the training process. A representative work is LLM-QAT [12], which proposes data-free distillation. However, this method introduces new trainable parameters, necessitates high-end GPU computational resources, and incurs a large time consumption. In contrast, Post-Training Quantization (PTQ) employs moderate resources to quantize pre-trained models without model retraining. Recent work, such as SpQR [3] and SqueezeLLM [8], compress most weights to 4 bits but maintain outlier weights at 16 bits, which complicates the inference process with both 4-bit and 16-bit inference.

SmoothQuant [18] introduces a per-channel scaling transformation that effectively smooths the magnitudes to address the challenge of quantizing activations. GPTQ [6] and OBQ [5] introduce an innovative weight quantization method based on approximate second-order information, ensuring high accuracy and efficiency in the quantization process. Our work shares the same ethos as GPTQ but additionally considers the softmax and matmul operations within the attention computation to formulate the quantization problem, resulting in improved accuracy.

Mixed-precision quantization offers a trade-off strategy for edge devices to maintain the accuracy with minimized model size. Existing works usually define some metrics to determine the quantization sensitivity of each layer. One representative work is HAWQ-V2 [4], which adopts Hessian trace for CNN layer sensitivity assessment and utilizes the Hutchinson algorithm to approximately estimate the Hessian trace. Our APTQ method also employs Hessian trace for sensitivity but adopts the Levenberg-Marquardt approximation [9] to directly calculate the Hessian trace with respect to the attention output, which is also an extension of GPTQ [6] by further considering the nonlinear operation (softmax) and matmul in the attention output. Another close related work is PB-LLM [16], which adopts a mixed 1-bit and fp-16 (half floating point) precision based on the Hessian values. Extreme low-bit quantization (1bit) is challenging for the accuracy. However, our APTQ method opts for a 2-bit and 4-bit mixed-precision quantization offering a better accuracy with the same model size comparing to PB-LLM. The effectiveness of this strategy is demonstrated in Section 4, where our method shows superior performance in terms of efficiency and model compression when compared to PB-LLM.

3 ALGORITHM

This section starts with the preliminaries to outline the evolution of quantization techniques from optimal brain quantization (OBQ) [5] to our proposed Hessian-attention-based quantization. We then propose an Attention-aware Post-Training Mixed-Precision Quantization, APTQ, to further compress the LLMs.

3.1 Preliminaries

General Quantization Framework. Quantization aims to reduce weight precision in neural networks, thus conserving computational resources. The general goal is to find a quantized weight matrix \hat{W} that approximates full precision output, minimizing the squared error. This process can be formally expressed as:

$$\operatorname{argmin}_{\hat{W}} \|WX - \hat{W}X\|_2^2. \quad (1)$$

In this equation, X represents the input to the layer, and \hat{W} denotes the quantized weight.

Optimal Brain Quantization (OBQ). Optimal Brain Quantization (OBQ) [5] is an innovative method that minimizes quantization errors by treating each neural network weight independently. The core of OBQ lies in iteratively quantizing each weight and adjusting the remaining unquantized weights to compensate for the quantization-induced errors. This approach is mathematically articulated as follows:

$$w_q = \operatorname{argmin}_{w_q} \frac{\operatorname{quant}(w_q) - w_q}{[H_F^{-1}]_{qq}}, \quad (2)$$

$$\delta_F = - \frac{w_q - \operatorname{quant}(w_q)}{[H_F^{-1}]_{qq}} \cdot (H_F^{-1})_{:,q}, \quad (3)$$

$$H_{-q}^{-1} = (H^{-1} - \frac{1}{[H^{-1}]_{qq}} H_{:,Q}^{-1} H_{Q,-}^{-1})_{-p}. \quad (4)$$

The Hessian matrix $H_F = 2X_F X_F^T$ guides the selection of the quantization candidate w_q from the full-precision weights F , and the update δ_F is calculated to minimize quantization error, as formalized in equations (2), (3) and (4) with $\operatorname{quant}(w)$ mapping weights to their nearest quantized values. Building upon OBQ, GPTQ [6] extends the principles by adopting the fixed order weights update strategy and Cholesky reformulation to speed up the computation.

3.2 Hessian-Attention-based Quantization

While GPTQ effectively minimizes layer-specific quantization errors, it overlooks the intricate nonlinearities in attention mechanisms, leading to suboptimality. APTQ, by contrast, embraces a holistic quantization strategy, factoring in the entire attention block and its nonlinear dynamics, which sharpens the precision of the quantized model, particularly in low-bitwidth scenarios.

As shown in Figure.1, we present the advanced architecture of APTQ, demonstrating its comprehensive quantization strategy. Unlike GPTQ, which primarily processes loss in the current layer, APTQ integrates a full-scope analysis of the attention mechanism, including the Q, K, V, O matrices, matmul and nonlinear activation layers such as softmax. This extensive approach not only focuses on the intricacies beyond simple weight matrix multiplication, but also significantly mitigates quantization errors, offering a robust solution in low-bitwidth quantization scenarios.

Objective Function. At a macroscopic level, our methodology employs a layer-wise quantization approach to address the quantization reconstruction problem for each layer’s weights. In the Transformer architecture, two main structural levels exist: the attention layers and the feed-forward layers. Specifically, in contrast to GPTQ, which treats each weight matrix as a linear layer and ignores the impact of other structures on the output, we treat all structures of the same layer as a whole, represented by the function

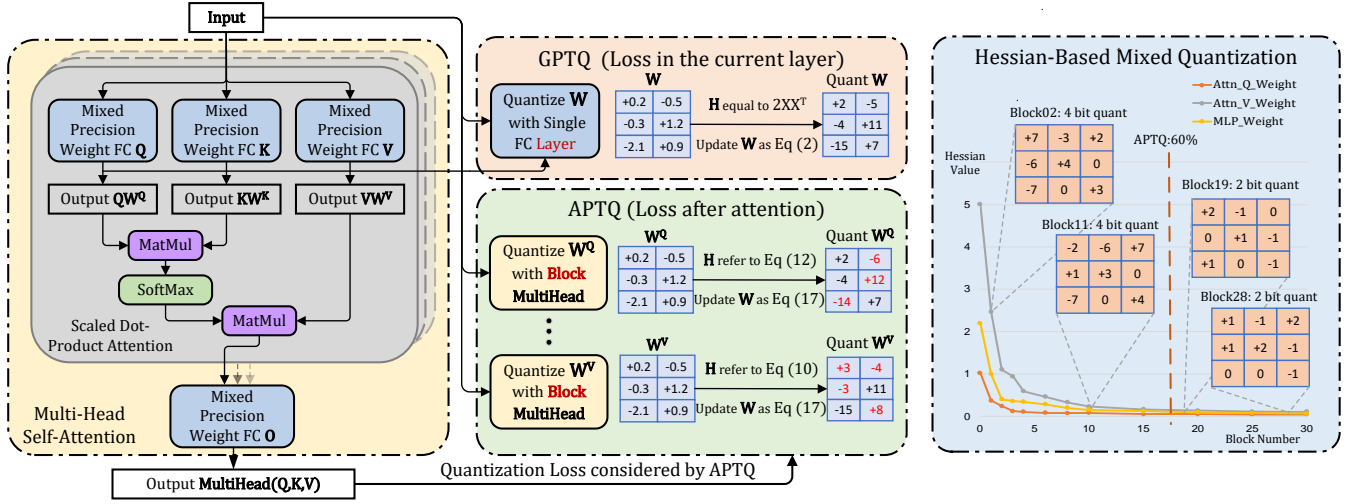


Figure 1: Overall architecture of APTQ (Attention-aware Post-Training Mixed-Precision Quantization): Unifying comprehensive transformer attention analysis with layer-specific Hessian trace quantization for enhanced model understanding.

F standing for the attention output Multihead(Q, K, V). We aim to reformulate Equation (1) and minimize the new squared error equation as follows:

$$\operatorname{argmin}_{\hat{W}} \|F(W) - F(\hat{W})\|_2^2. \quad (5)$$

where W remains constant and \hat{W} is the quantized weights to be optimized. The Hessian matrix of this function is computed as:

$$H_{\hat{W}} = 2 \cdot (F'(\hat{W}) \cdot F'(\hat{W})^T + [F(W) - F(\hat{W})] \cdot F''(\hat{W})). \quad (6)$$

This is the general expression of Hessian matrix. To ensure $H_{\hat{W}}$ is positive definite and invertible, we only retain the first-order derivative portion as the expression for the Hessian matrix, which is widely known as the Levenberg-Marquardt approximation [9]:

$$H_{\hat{W}} = 2 \cdot [F'(\hat{W}) \cdot F'(\hat{W})^T]. \quad (7)$$

Derivatives for Different Quantization Layers. The current problem is transformed into finding the partial derivative of $F(\hat{W})$ with respect to the weights \hat{W} . The $F(\hat{W})$ function is different for the Feed-Forward layers and Attention layers. In the Feed-Forward layer, the main structure is a linear fully connected layer. The Hessian matrix is easily computed as $H_F = 2X_F X_F^T$, corresponding to the Hessian matrix form in the GPTQ method.

In the Attention layer, a multi-head mechanism is employed, where each attention head contains an Attention function:

$$F(W, X) = \text{MultiHead}(Q, K, V). \quad (8)$$

The quantized weight matrices lead to different derivatives. When quantizing the W^O matrix, consider W^Q, W^K, W^V as constants:

$$\frac{\partial F}{\partial W^O} = \text{Concat}(\text{head}_1, \dots, \text{head}_H)^T \frac{\partial F}{\partial X}. \quad (9)$$

When quantizing the W^V matrix, consider W^Q, W^K, W^O as constants:

$$\frac{\partial F}{\partial W^V} = M^T \frac{\partial F}{\partial X} (W^O)^T. \quad (10)$$

Here, M represents a matrix composed of H heads losing W_i^V :

$$M_h = \text{softmax}\left(\frac{QW_h^Q (W_h^K)^T K^T}{\sqrt{d_k}}\right)V, M = [M_1, \dots, M_H]. \quad (11)$$

When quantizing W^Q or W^K matrices, consider the remaining three terms as constants:

$$\frac{\partial F}{\partial W_h^Q} = \frac{1}{\sqrt{d_k}} Q^T \frac{\partial F}{\partial N} \mathbb{P}_h^T K W_h^K, \quad (12)$$

$$\frac{\partial F}{\partial W_h^K} = \frac{1}{\sqrt{d_k}} K^T \mathbb{P}_h \frac{\partial F}{\partial N} Q W_h^Q. \quad (13)$$

Here, W_h represents the weight matrix in the n -th attention head, and N and \mathbb{P}_h are given by:

$$N_h = \frac{QW_h^Q (W_h^K)^T K^T}{\sqrt{d_k}}, N = [N_1, \dots, N_H], \quad (14)$$

$$\mathbb{P}_h = (\dots, E_{n \times n}^h, \dots)_{n \times nH}. \quad (15)$$

After computing the gradients from equations (9), (10), (12) and (13), we can further get their second order gradients using equation (7) to obtain the corresponding Hessian matrix. Thus, referring to the optimization problem in equation (5), combining the quantization techniques in equations (2), (3), we derive the following formulas for updating weights in the context of attention mechanisms:

$$E = -\frac{w_q - \text{quant}(w_q)}{([H_{\hat{W}}^{-1}]_{qq})}, \quad (16)$$

$$\delta_F = E \cdot (H_{\hat{W}}^{-1})_{:,q}. \quad (17)$$

Here, E represents the quantization error, w_q refers to the quantized weights of the current group. δ_F refers to the corresponding optimal updates for the remaining float weights (not yet quantized weights of the current layer). This principle is uniformly applicable to the quantization of Q (query), K (key), V (value), and O (output) weight

matrices in attention mechanisms. By synthesizing these elements, we can effectively compute the second-order Hessian information relevant to the weights within the attention layers. This advanced computation aids in the update and optimization of weights, targeting the minimization of the original squared error as defined in equation (5). This approach facilitates the realization of quantized models with robust performance across different components of the attention mechanism. The comprehensive algorithm is detailed in Algorithm Box 1.

Algorithm 1 APTQ via Hessian-Attention-based Mixed-Precision Quantization

Input: Pre-trained model weights W , blocksize B , Hessian matrix H , quantization function quant , Layer names layerName , Ratio of 4-bit in 2/4 mixed-precision R .

- 1: Initialize quantized weight matrix $Q \leftarrow 0_{d_{\text{row}} \times d_{\text{col}}}$.
- 2: Initialize block quantization error matrix $E \leftarrow 0_{d_{\text{row}} \times B}$.
- 3: **Step 1: 4-bit Hessian-Attention-Based Quantization**
- 4: **for** $i = 0, B, 2B, \dots$ **do**
- 5: **for** $j = i, \dots, i + B - 1$ **do**
- 6: **if** “self_attn.k_proj” in layerName **then**
- 7: $H_W^K = 2 \left[\frac{\partial F}{\partial W^K} \cdot \frac{\partial F}{\partial W^K}^T \right]$ from Equation (13)
- 8: $Q_{:,j}^K \leftarrow \text{quant}(W_{:,j})$
- 9: $E_{:,j-i}^K \leftarrow (W_{:,j}^K - Q_{:,j}^K) / [H_W^{-1}]_{jj}^K$ based on Equation (16)
- 10: $W_{:,j:(i+B)}^K \leftarrow W_{:,j:(i+B)}^K - E_{:,j-i}^K \cdot (H_W^{-1})_{:,j:(i+B)}^K$ based on Equation (17)
- 11: For self_attn.Q, V, and O projection layers, similar updates are applied
- 12: Compute the average Hessian trace for each layer in block $i : (i + B)$.
- 13: **end if**
- 14: **end for**
- 15: **end for**
- 16: **Step 2: Hessian-trace-based Mixed-Precision Quantization**
- 17: Calculate Hessian trace values for each layer, and order them from highest to lowest, starting with the previously established 4-bit quantization.
- 18: Determine the layers for mixed-precision quantization based on the computed Hessian trace values and R .
- 19: **for** each selected layer **do**
- 20: Calibrate the bit allocation in line with each layer’s Hessian trace sensitivity and R .
- 21: Implement 2/4 bit mixed-precision quantization
- 22: **end for**

Output: The resulting quantized model weights Q are characterized by scale, zero-point, and quantization error.

3.3 Hessian-Trace-based Mixed-Precision Quantization

As mentioned in Section 2, the Hessian trace provides sensitivity information for implementing mixed-precision quantization. Figure 1 illustrates the APTQ method’s allocation of 4-bit and 2-bit quantizations, utilizing average Hessian trace values as a measure of layer sensitivity. This approach diverges from the GPTQ method, which concentrates solely on the matrix multiplication within the current layer, while APTQ provides a comprehensive assessment of each layer’s impact.

By computing the average trace of the Hessian matrix, the method determines the appropriate level of precision for the quantization of each layer. Layers with higher Hessian Trace values, which exert a greater influence on the network’s output, require higher bit precision to ensure the model’s accuracy. Utilizing this mixed-precision quantization scheme results in models with an average bit precision defined by the formula:

$$\text{average bits} = 4 \times R + 2 \times (1 - R), \quad (18)$$

where R denotes the proportion of weights quantized at 4 bits within the overall quantization process. This formula is a pivotal aspect of the APTQ methodology, facilitating a dynamic adjustment that is particularly advantageous for deploying large language models on edge devices. The adaptability of R allows the APTQ algorithm to allocate higher precision to layers with greater sensitivity, while applying more robust quantization to less sensitive layers. Consequently, this leads to a quantized model that achieves an optimal balance between performance and size to deploy on edge devices.

Algorithm 1 unfolds into two decisive steps aimed at enhancing model efficiency while preserving performance. Step 1 applies 4-bit quantization to the attention mechanism’s K (key) layer, guided by the Hessian matrix, H_W^K , that entails the second-order derivative crucial for this optimization, as formulated in Equation (13). This step adjusts the precision of the K layer’s weights, considering the broader implications for the model’s performance. The individual optimization of the K , Q , V , and O layers is informed by their respective Hessian matrices, ensuring that quantization is precisely targeted to maintain the balance between efficiency and accuracy. In essence, Hessian-Attention-based quantization strategically refines weight precision within attention layers to maintain model accuracy without unnecessary computational burden.

In the algorithm’s second phase, a mixed-precision quantization strategy is implemented, beginning with the calculation of Hessian trace values across the layers. These values are then ordered in a descending sequence, starting with the layers previously quantized at a 4-bit level. This ordering informs the selection of layers for subsequent mixed-precision quantization, which is performed in accordance with the computed Hessian trace values. This selective quantization process is designed to align closely with each layer’s functional impact on the overall model, ensuring a quantization scheme that is both effective and efficient.

4 EXPERIMENT

4.1 Experiment Setup

To evaluate APTQ’s performance, we focus on two primary metrics: perplexity and zero-shot performance. The LLaMa family [17] serves as the foundation for our experiments, owing to its efficacy and critical influence in recent model advancements. To maintain consistency and comparability, our benchmarking procedures against GPTQ adhere to identical experimental configurations. Our calibration dataset encompasses 128 segments, each containing 2048 tokens randomly sampled from the C4 dataset. All experiments deploy a group size of 128 and are executed on a single NVIDIA A100 GPU of 80GB memory. Our APTQ is applied directly

Table 1: Comparison of Perplexity of Quantized LLaMa Models on C4 and WikiText-2 Datasets.

Method	Avg bit	C4 ↓	WikiText-2 ↓
LLaMa-7B	16	5.22	5.68
GPTQ [6]	4.0	5.62	8.14
OWQ [10]	4.01	5.56	7.15
LLM-QAT [12]	4.0	7.40	10.90
PB-LLM-20% [16]	3.4	20.61	17.19
APTQ	4.0	5.23	6.45
APTQ-75%	3.5	5.54	6.54
APTQ-50%	3.0	6.24	6.76

to the pre-trained model (post-training quantization). The evaluation of zero-shot performance is conducted using the EleutherAI/lm-evaluation-harness [7]. Note that we use the format APTQ- R to represent the mixed precision (2/4-bit) setting, with R represents the percentage of 4-bit weights as discussed in Equation (18).

4.2 Evaluation of Perplexity performance

We assess the the performance of APTQ using the C4 [15] and WikiText-2 [13] benchmarks. We compare APTQ against three established PTQ methods: GPTQ [6], OWQ [10], and PB-LLM [16]. Notably, OWQ and PB-LLM extend upon GPTQ, with PB-LLM incorporating mixed-precision quantization. To ensure a balanced comparison, all methods are evaluated on a standardized platform. Moreover, we benchmark APTQ’s performance with the leading QAT approach, LLM-QAT. Table 1 reveals that APTQ, at an average 4 bit, closely matches the full-precision model and attains SOTA performance on the C4 dataset, showing only a 0.01-point increase in perplexity. Remarkably, even with average bit rates reduced to 3.5 and 3.0, APTQ’s perplexity remains comparable to that of GPTQ’s 4-bit model. This evidence of APTQ’s stability at low bit rates positions it as a potent tool for optimizing the quantization and deployment of large-scale language models like LLaMa-7B.

To substantiate the robustness and broad applicability of the Hessian trace-based mixed-precision quantization posited in our study, we conducted a comparative analysis of various 4-bit utilization levels of APTQ against other prevalent PTQ methods applied to the LLaMa-7B model on the C4 dataset. The APTQ model, quantized at an average of 4 bit, not only approaches the full-precision model’s perplexity but also outperforms all other PTQ approaches at a reduced precision of 3.5 bits. Impressively, configurations below 3 bits still surpass the 4-bit LLM-QAT baseline, underscoring APTQ’s efficacy. These results unequivocally demonstrate the superior performance of APTQ, leveraging Hessian trace-driven precision allocation to optimize quantization outcomes.

Figure 2 visually summarizes our findings. It presents the comparative perplexity results of the LLaMa-7B model using APTQ at various bit utilization ratios when benchmarked against other PTQ and QAT methods on the C4 dataset. As depicted in the figure, the APTQ model consistently maintains competitive performance, even at significantly reduced bit rates. This graphical representation reinforces the effectiveness of the Hessian trace-based mixed-precision approach we advocate in this study, illustrating its potential for resource-efficient large model deployment.

4.3 Evaluation of Zero-shot performance

In the evaluation of zero-shot performance, we extend our investigation to a suite of challenging zero-shot language tasks. These tasks,

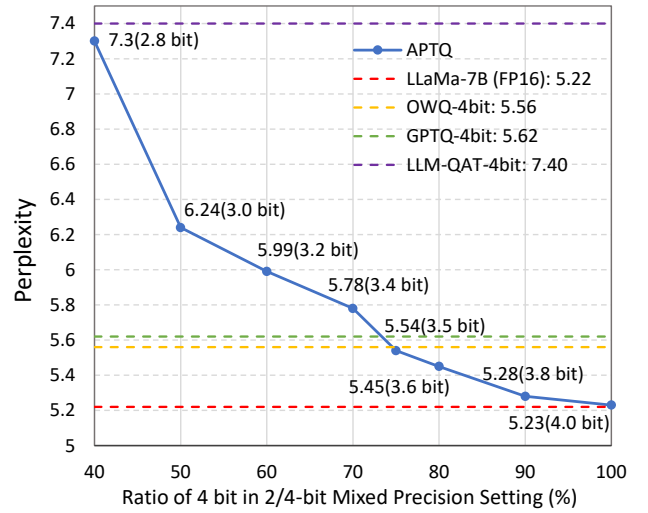


Figure 2: Comparative perplexity results of LLaMa-7B using APTQ at various 4-bit ratio against others on C4 dataset which span Predictive Question Answering (PIQA), Hellaswag, ARC-Easy (Arc-E), ARC-Challenge (Arc-C), and WinoGrande, serve as a benchmark for common sense reasoning in machine comprehension. We compare the proposed APTQ method on LLaMa-7B and LLaMa-13B with other advanced quantization techniques including round-to-nearest (RTN), SmoothQuant [18], FPQ [11], LLM-QAT [12], and GPTQ [6].

As depicted in Table 2, we benchmark the APTQ framework against current SOTA PTQ methodologies applied to the LLaMa-7B model. Our findings illustrate that APTQ, when configured to 3.8 bits, sustains a remarkably minimal deviation in accuracy, with a diminutive average accuracy drop of only 0.32 points from the full-precision model. Even when the APTQ is optimized down to an average of 3.6 or 3.5 bits, it still consistently outperforms the majority of 4-bit PTQ models. These findings demonstrate that APTQ excels in zero-shot tasks with minimal bit usage, highlighting its effectiveness in deploying large-scale language models in environments with limited computational resources. This underscores APTQ’s advantage in resource-efficient performance.

4.4 Ablation Study

Furthermore, we present an ablation study to validate the superiority of APTQ over manual block-wise quantization schemes. Given that quantization is performed on a layer-wise basis, the most intuitive mixed-precision quantization strategy is to uniformly quantize all layers within each block. Here, we compare this conventional approach with APTQ on the LLaMa-7B model tested on the C4 dataset, with perplexity as the evaluation metric. The results in Table 3 reveal APTQ’s efficacy over manual block-wise quantization for LLaMa-7B on C4, reflected in its consistently lower PPL across various quantization ratios.

5 CONCLUSION

This paper presented an Attention-aware Post-Training Mixed-Precision Quantization (APTQ) algorithm for quantizing large language models to mixed precisions. APTQ is a promising post-training quantization strategy by utilizing the second-order information of each layer’s weights with consideration of the nonlinear

Table 2: Zero-shot accuracy of quantized LLaMa models on common sense reasoning tasks.

Model	LLaMa-7B							LLaMa-13B					
	Avg bit	PIQA	Hellaswag	Arc-E	Arc-C	WinoGrande	$\overline{Acc}\% \uparrow$	PIQA	Hellaswag	Arc-E	Arc-C	WinoGrande	$\overline{Acc}\% \uparrow$
FP16	16	79.2	76.2	72.8	44.7	69.9	68.56	80.3	79.0	74.8	47.9	72.7	70.94
RTN [12]	4.0	77.3	72.7	68.8	43.1	66.9	65.76	79.1	76.8	72.6	46.5	70.5	69.10
SmoothQuant [18]	4.0	76.4	68.1	67.3	39.6	66.0	63.48	77.9	74.2	76.3	45.5	69.7	68.72
FPQ [11]	4.0	77.8	75.0	72.4	41.7	69.0	66.60	79.4	77.7	72.8	47.3	71.5	69.74
LLM-QAT [12]	4.0	78.3	74.0	70.0	41.7	69.0	66.60	79.4	77.7	72.8	47.3	71.5	69.74
GPTQ [6]	4.0	76.0	69.4	66.9	43.0	66.7	64.40	79.8	77.7	73.2	45.9	72.6	69.84
PB-LLM 30% [16]	4.1	78	74.3	69.0	42.3	69.7	66.66	-	-	-	-	-	-
PB-LLM 10% [16]	2.7	67.8	68.1	58.7	39.6	67.4	60.32	-	-	-	-	-	-
APTQ	4.0	78.6	75.7	72.4	44.4	69.3	68.08	79.9	78.8	73.9	47.0	72.1	70.34
APTQ-90%	3.8	78.8	75.9	73.6	43.5	69.4	68.24	79.4	78.8	73.8	47.8	72.6	70.48
APTQ-80%	3.6	78.0	75.3	70.2	43.7	69.5	67.34	79.5	78.2	72.8	46.5	72.6	69.92
APTQ-75%	3.5	77.5	74.5	68.7	44.2	70.2	67.02	79.3	77.6	71.8	46.1	73.2	69.60
APTQ-70%	3.4	77.6	73.4	66.9	41.3	68.9	65.62	78.3	77.5	71.4	46.3	72.5	69.20
APTQ-60%	3.2	76.8	72.1	63.1	39.3	69.5	64.16	78.6	74.2	69.5	44.2	69.5	67.20
APTQ-50%	3.0	74.5	68.3	57.9	36.4	65.3	60.48	74.4	71.2	64.1	41.0	68.0	63.74

Table 3: Ablation Study: Comparison of APTQ and Manual Block-wise Quantization on LLaMa-7B’s C4 Perplexity

Method	Ratio of 4-bit	Avg bit	Perplexity ↓
Manual Block-wise	75%	3.5	5.84
APTQ-75%	75%	3.5	5.54
Manual Block-wise	50%	3.0	7.04
APTQ-50%	50%	3.0	6.24

effect of attention outputs. Furthermore, the Hessian trace is developed as a sensitivity measurement to further achieve mixed 2/4-bit precision. For LLM LLaMa-7B, APTQ surpasses previous quantization methods, achieving an average of 4 bits with a 5.22 perplexity, nearly equivalent to full precision in the C4 dataset. Furthermore, under the zero-shot LLM setting, APTQ achieves the state-of-the-art results 68.24% and 70.48% accuracy at an average bitwidth of 3.8 for LLaMa-7B and LLaMa-13B, respectively, indicating that APTQ can achieve a deeply quantized solution for large language models without sacrificing accuracy.

6 ACKNOWLEDGEMENT

This work was supported by Shenzhen Science and Technology Program (Grant No. KQTD20200820113051096), Science and Technology Innovation Committee Foundation of Shenzhen (Grant No. JCYJ20220818100217038), and by the Theme-based Research Scheme (TRS) project T45-701/22-R, Hong Kong SAR.

REFERENCES

- [1] Miguel A Carreira-Perpinán and Yerlan Idelbayev. 2018. “learning-compression” algorithms for neural net pruning. In *IEEE CVPR*. 8532–8541.
- [2] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. 2021. Progressive darts: Bridging the optimization gap for nas in the wild. *IJCV* 129 (2021), 638–655.
- [3] Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefer, and Dan Alistarh. 2023. SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression. *arXiv preprint arXiv:2306.03078* (2023).
- [4] Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Hawq-v2: Hessian aware trace-weighted quantization of neural networks. *NIPS* 33 (2020), 18518–18529.
- [5] Elias Frantar and Dan Alistarh. 2022. Optimal brain compression: A framework for accurate post-training quantization and pruning. *NeurIPS* 35 (2022), 4475–4488.
- [6] Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. 2023. GPTQ: Accurate Post-training Compression for Generative Pretrained Transformers. *ICLR* (2023).
- [7] Leo Gao, Jonathan Tow, Stella Biderman, Charles Lovering, Jason Phang, Anish Thite, Fazz, Niklas Muennighoff, and et al. 2022. *EleutherAI/lm-evaluation-harness: v0.3.0*. <https://doi.org/10.5281/zenodo.7413426>
- [8] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. 2023. SqueezeLLM: Dense-and-Sparse Quantization. *arXiv preprint arXiv:2306.07629* (2023).
- [9] Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. *NeurIPS* 2 (1989).
- [10] Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. 2023. OWQ: Lessons learned from activation outliers for weight quantization in large language models. *arXiv preprint arXiv:2306.02272* (2023).
- [11] Shih-yang Liu, Zechun Liu, Xijie Huang, Pingcheng Dong, and Kwang-Ting Cheng. 2023. LLM-FP4: 4-Bit Floating-Point Quantized Transformers. *arXiv preprint arXiv:2310.16836* (2023).
- [12] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023. LLM-QAT: Data-Free Quantization Aware Training for Large Language Models. *arXiv preprint arXiv:2305.17888* (2023).
- [13] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843* (2016).
- [14] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *NeurIPS* 35 (2022), 27730–27744.
- [15] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR* 21, 1 (2020), 5485–5551.
- [16] Yuzhang Shang, Zhihang Yuan, Qiang Wu, and Zhen Dong. 2023. PB-LLM: Partially Binarized Large Language Models. *arXiv preprint arXiv:2310.00034* (2023).
- [17] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
- [18] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *ICML*. 38087–38099.
- [19] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).