KLAWQ: KL-Aware Weight Quantization for Efficient Large Language Models on Edge Devices

Muhammad Haseeb Hamza Iqbal Dr Murtaza Taj

Abstract

Large Language Models (LLMs) exhibit remarkable capabilities but their substantial size and computational demands impede deployment on resource-constrained edge devices. Post-Training Quantization (PTQ) offers an efficient compression pathway, yet state of the art methods like GPTQ, while effective, primarily optimize for weight reconstruction and can suffer accuracy degradation, especially at very low bit-widths. Knowledge Distillation (KD) aims to preserve functional fidelity but is typically a separate, costly training process. We propose KLAWQ (KL-Aware Weight Quantization), a novel PTQ framework that enhances GPTQ by directly integrating a Kullback-Leibler (KL) divergence objective into its second-order, layer-wise quantization procedure. By making the quantization process explicitly aware of the full-precision teacher model's output distribution, KLAWQ seeks to preserve both structural weight fidelity and crucial functional characteristics. Experimental evaluation on the GPT-2 model at 8-bit precision demonstrates that KLAWQ can achieve improved perplexity compared to vanilla GPTQ through its KL-aware optimization. This approach offers a promising direction for developing more robust and accurate quantization techniques, facilitating the deployment of LLMs on edge platforms. Code:`https://github.com/ha405/Compression-Framework-for-EdgeAI`.

1. Introduction

Transformer-based Language Models have been at the core of recent advancements in artificial intelligence, and have demonstrated unprecedented capabilities across a range of natural language and vision tasks. However, their immense memory and computational requirements, due to billions of parameters, pose significant barriers to their deployment in resource-constrained environments. For instance, GPT-3's 175 billion parameters consume over 350 GB of memory in FP16 just to load (Brown et al., 2020). Edge computing demands real-time responsiveness, low latency, and energy efficiency, rendering the direct deployment of such large-scale AI models infeasible.

A standard approach to tackle this compute and memory overhead is model compression. Quantization methods excel at reducing model size and accelerating inference by converting the model to lower numerical precision (Zhu et al., 2024). Post training quantizaton (PTQ) efficiently converts a full-precision LLM to low-precision without retraining, saving memory and computational costs. Advanced PTQ techniques like GPTQ (Frantar et al., 2023) leverage second-order information (Hessian approximations) to minimize weight reconstruction errors and achieve better accuracy preservation than vanilla quantization. However, even state-of-art PTQ methods can encounter significant performance degradation when pushing to very low bit-widths (e.g., 4-bit or 3-bit).

Knowledge distillation (KD) presents an alternative philosophy for model compression, focusing explicitly on functional preservation (Hinton et al., 2015). KD transfers knowledge from a larger, high-performance ”teacher” model to a smaller ”student” model by training the student to mimic the teacher's output probability distributions using the Kullback-Leibler (KL) divergence loss (Xu & Zhang, 2024). This encourages the student to learn the decision boundaries of the teacher and yield better performance on downstream tasks despite its smaller size.

Empirical evidence suggests that KD achieves stable performance across varying conditions while quantization's delivers peak performance but at the cost of stability (Qin et al., 2024). Quantization focuses purely on weight fidelity, it ignores the model's functional behavior potentially discarding the decision boundaries. In this paper, we propose KLAWQ (KL-Aware Weight Quantization) to quantize a model to low-precisions while preserving functional-fidelity and stability. KLAWQ modifies the standard Hessian-weighted reconstruction objective by incorporating a KL divergence term. The explicit awareness of the original model's output distribution guides KLAWQ to yield low-precision representations and maintaining high fidelity to both the original weight structure and the model's functional behavior.

2. Methodology: KLAWQ

The KLAWQ framework extends the GPTQ (Frantar et al., 2023) algorithm by merging knowledge distillation with its layer-wise, second-order quantization process. Our objective is to guide weight quantization process to minimize reconstruction error and preserve the functional behavior of the full-precision teacher model by aligning their output distributions. We optionally allow for the inclusion of a cross-entropy (CE) term for direct task-specific fine-tuning.

2.1. Foundation: GPTQ Review

GPTQ (Frantar et al., 2023) is a post-training quantization (PTQ) method that leverages second-order information to quantize pre-trained models with minimal accuracy loss. For a given layer with full-precision weights $\boldsymbol{W} \in \mathbb{R}^{D_{out} \times D_{in}}$ and quantized weights $\boldsymbol{Q}$, GPTQ aims to minimize the squared reconstruction error, weighted by the empirical input Hessian $\boldsymbol{H}$:

$$\mathcal{L}_{MSE}(\boldsymbol{Q}) = \operatorname{Tr}\left((\boldsymbol{W} - \boldsymbol{Q})^\top \boldsymbol{H}(\boldsymbol{W} - \boldsymbol{Q})\right). \tag{1}$$

The objective in Equation 1 is equivalent to minimizing the expected squared Euclidean distance between the outputs of the full-precision layer and the quantized layer, i.e., $\mathbb{E}_{\boldsymbol{x}\sim\text{data}}\|(\boldsymbol{W} - \boldsymbol{Q})\boldsymbol{x}\|_2^2$, where $\boldsymbol{x}$ represents an input activation vector.

The Hessian matrix $\boldsymbol{H} \in \mathbb{R}^{D_{in} \times D_{in}}$ is approximated by the average second moment of the input activations, collected from a calibration dataset comprising T samples $\{\boldsymbol{x}_t\}_{t=1}^T$:

$$\boldsymbol{H} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t \boldsymbol{x}_t^\top. \tag{2}$$

This matrix, $\boldsymbol{H}$, provides a second-order approximation of the loss landscape with respect to the layer weights. Its eigenvectors and corresponding eigenvalues delineate directions of varying sensitivity to weight perturbations; larger eigenvalues correspond to directions of higher input variance and thus greater sensitivity to quantization error.

Directly minimizing Equation 1 over all elements of $\boldsymbol{Q}$ is computationally intractable for large matrices. GPTQ does this by employing an iterative, column-wise quantization procedure. The weights $\boldsymbol{W}$ are processed column by column (or in small blocks of columns). Let $\boldsymbol{w}_i$ denote the i-th column of $\boldsymbol{W}$ and $\boldsymbol{q}_i$ its quantized counterpart. Upon quantizing $\boldsymbol{w}_i$ to $\boldsymbol{q}_i$, a quantization error $\boldsymbol{e}_i = \boldsymbol{w}_i - \boldsymbol{q}_i$ is introduced. To compensate for this error and its impact on the subsequent quantization of remaining columns, GPTQ utilizes the inverse of a damped Hessian matrix, $\boldsymbol{H'}^{-1}$. The update rule for any remaining unquantized column $\boldsymbol{w}_j$ (where $j > i$) is given by:

$$\boldsymbol{w}_j \leftarrow \boldsymbol{w}_j - \boldsymbol{e}_i \frac{(\boldsymbol{H'}^{-1})_{ij}}{(\boldsymbol{H'}^{-1})_{ii}}. \tag{3}$$

The error compensation mechansim of GPTQ relies on the inverse of the damped Hessian, $\boldsymbol{H'}^{-1}$. For each column $\boldsymbol{w}_i$ undergoing quantization, its corresponding diagonal element in the inverse Hessian, $(\boldsymbol{H'}^{-1})_{ii}$, defines a scaling factor $d_i = \sqrt{(\boldsymbol{H'}^{-1})_{ii}}$. This factor normalizes the quantization problem for $\boldsymbol{w}_i$ by transforming it into a coordinate system where the local loss curvature is approximately unity. The quantization of $\boldsymbol{w}_i$ to $\boldsymbol{q}_i$ is then executed within this transformed space as $\boldsymbol{q}_i = \text{Quantize}(\boldsymbol{w}_i/d_i) \cdot d_i$. Following this, the off-diagonal elements $(\boldsymbol{H'}^{-1})_{ij}$ govern the propagation of the quantization error $\boldsymbol{e}_i = \boldsymbol{w}_i - \boldsymbol{q}_i$ to the remaining unquantized columns $\boldsymbol{w}_j$ (for $j > i$), reflecting the inter-column coupling captured by the Hessian. This iterative, column-wise procedure, with error compensation, enables an efficient greedy approximation for minimizing the global objective (Equation 1). To ensure the numerical stability and invertibility of the Hessian $\boldsymbol{H}$, a damping term is applied prior to its inversion, forming a damped Hessian $\boldsymbol{H'} = \boldsymbol{H} + \lambda_d \boldsymbol{I}$, where λ_d is a small positive scalar.

2.2. KLAWQ: Augmented Optimization

KLAWQ augments the GPTQ objective by incorporating second-order curvature information derived from a knowledge distillation loss and, optionally, a cross-entropy task loss.

2.2.1. Loss Function Components

Let $\boldsymbol{p}_t = \text{softmax}(\boldsymbol{W}\boldsymbol{x}_t/\tau)$ be the teacher model's soft output probabilities for input $\boldsymbol{x}_t$, and $\boldsymbol{q}_t = \text{softmax}(\boldsymbol{Q}\boldsymbol{x}_t/\tau)$ be the student model's (with quantized weights $\boldsymbol{Q}$) soft outputs, where τ is the distillation temperature. The KLAWQ objective considers:

1. **Reconstruction Loss** ($\mathcal{L}_{MSE}$): As defined in Eq. 1.
2. **KL Divergence Loss** ($\mathcal{L}_{KL}$): To align student and teacher outputs:

$$\mathcal{L}_{KL}(\boldsymbol{Q}) = \sum_{t=1}^{T} \text{KL}(\boldsymbol{p}_t \| \boldsymbol{q}_t). \tag{4}$$

3. **Cross-Entropy Loss** ($\mathcal{L}_{CE}$, optional): If ground-truth labels $\boldsymbol{y}_t$ are available:

$$\mathcal{L}_{CE}(\boldsymbol{Q}) = \sum_{t=1}^{T} \text{CE}(\boldsymbol{y}_t, \boldsymbol{q}_t). \tag{5}$$

The total augmented loss $\mathcal{L}(\boldsymbol{Q})$ is a weighted sum:

$$\mathcal{L}(\boldsymbol{Q}) = \mathcal{L}_{MSE}(\boldsymbol{Q}) + \beta\mathcal{L}_{KL}(\boldsymbol{Q}) + \gamma\mathcal{L}_{CE}(\boldsymbol{Q}), \tag{6}$$

where $\beta \geq 0$ and $\gamma \geq 0$ are hyperparameters. Setting $\gamma = 0$ disables CE fine-tuning.

2.2.2. Second-Order Curvature Matrices

To incorporate $\mathcal{L}_{KL}$ and $\mathcal{L}_{CE}$ into the GPTQ framework, we approximate their second-order contributions in the input space, analogous to $\boldsymbol{H}$.

- **KL Hessian Approximation ($\boldsymbol{A}$):** The second-order curvature information from the KL divergence loss is incorporated via an approximated Hessian matrix $\boldsymbol{A} \in \mathbb{R}^{D_{in} \times D_{in}}$. This matrix is derived from the Hessian of the KL divergence, $\mathcal{L}_{KL}(\boldsymbol{Q}) = \sum_{t=1}^{T} \text{KL}(\boldsymbol{p}_t \| \boldsymbol{q}_t)$, with respect to the student's pre-softmax logits $\boldsymbol{z}_t^Q = \boldsymbol{Q}\boldsymbol{x}_t$. A standard formulation for the Hessian of losses involving softmax outputs, when projected back to the input space via $\boldsymbol{x}_t$, is given by:

$$\boldsymbol{A} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{\tau^2} \left(\text{diag}(\boldsymbol{q}_t) - \boldsymbol{q}_t \boldsymbol{q}_t^\top \right) \boldsymbol{x}_t \boldsymbol{x}_t^\top . \quad (7)$$

  Here, $\boldsymbol{q}_t = \text{softmax}(\boldsymbol{Q}\boldsymbol{x}_t/\tau)$ represents the student's soft output probabilities and τ is the distillation temperature.

- **CE Hessian Approximation ($\boldsymbol{B}$):** Similarly, for the CE loss:

$$\boldsymbol{B} = \frac{1}{T} \sum_{t=1}^{T} \left(\text{diag}(\boldsymbol{q}_t) - \boldsymbol{q}_t \boldsymbol{q}_t^\top \right) \boldsymbol{x}_t \boldsymbol{x}_t^\top . \quad (8)$$

Note that $\boldsymbol{A}$ and $\boldsymbol{B}$ share the same structural form if $\tau = 1$ for $\boldsymbol{A}$.

2.2.3. Combined Hessian and Normalization

The individual Hessian matrices ($\boldsymbol{H}$, $\boldsymbol{A}$, and optionally $\boldsymbol{B}$) are aggregated into a total effective Hessian, $\boldsymbol{H}_{tot}$. To ensure balanced contributions from each component, $\boldsymbol{A}$ and $\boldsymbol{B}$ undergo trace normalization with respect to $\boldsymbol{H}$. Defining $s_H = \text{Tr}(\boldsymbol{H})$, $s_A = \text{Tr}(\boldsymbol{A})$, and $s_B = \text{Tr}(\boldsymbol{B})$, the normalized matrices are $\boldsymbol{A}' = (s_H/s_A)\boldsymbol{A}$ and $\boldsymbol{B}' = (s_H/s_B)\boldsymbol{B}$, assuming non-zero s_A, s_B. The total Hessian is then a weighted sum:

$$\boldsymbol{H}_{tot} = \boldsymbol{H} + \beta \boldsymbol{A}' + \gamma \boldsymbol{B}', \quad (9)$$

where $\beta, \gamma \geq 0$ are hyperparameters. For numerical stability and to ensure invertibility, $\boldsymbol{H}_{tot}$ is subsequently damped by adding a scaled identity matrix:

$$\boldsymbol{H}'_{tot} = \boldsymbol{H}_{tot} + \lambda_d \boldsymbol{I}, \quad (10)$$

where $\boldsymbol{I}$ is the identity matrix and $\lambda_d > 0$ is a damping factor, proportional to $\text{Tr}(\boldsymbol{H}_{tot})$.

2.3. KLAWQ Quantization Procedure

With the damped and inverted total Hessian $(\boldsymbol{H}'_{tot})^{-1}$, KLAWQ performs column-wise quantization. For each column $\boldsymbol{w}_i$ of $\boldsymbol{W}$ (where $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{D_{in}}]$):

1. **Compute Scale Factor:** $d_i = \sqrt{((\boldsymbol{H}'_{tot})^{-1})_{ii}}$.
2. **Quantize Column:** $\boldsymbol{q}_i = \text{Quantize}(\boldsymbol{w}_i/d_i) \cdot d_i$. The function Quantize$(\cdot)$ performs quantization to the target bit-width and scheme (e.g., symmetric, asymmetric, group-wise).
3. **Compute Quantization Error:** $\boldsymbol{e}_i = \boldsymbol{w}_i - \boldsymbol{q}_i$.
4. **Propagate Error:** The error $\boldsymbol{e}_i$ is propagated to subsequent unquantized columns $\boldsymbol{w}_j$ ($j > i$):

$$\boldsymbol{w}_j \leftarrow \boldsymbol{w}_j - \boldsymbol{e}_i \frac{((\boldsymbol{H}'_{tot})^{-1})_{ij}}{((\boldsymbol{H}'_{tot})^{-1})_{ii}}. \quad (11)$$

Note: Columns can be reordered based on descending d_i^2 values before quantization to prioritize more sensitive columns

2.4. Workflow Overview

1. **Calibration Pass:** A single forward pass through the full-precision teacher model on a calibration dataset $\{(\boldsymbol{x}_t, \boldsymbol{y}_t)\}$ to cache layer inputs $\boldsymbol{x}_t^{(l)}$ and teacher soft targets $\boldsymbol{p}_t$.
2. **Layer-wise Quantization Loop:** For each layer $l = 1, \ldots, L$:
   (a) Obtain student soft targets $\boldsymbol{q}_t$ by passing $\boldsymbol{x}_t^{(l)}$ through previously quantized layers $(1, \ldots, l-1)$ and the current layer l (using its full-precision weights $\boldsymbol{W}^{(l)}$ for this step).
   (b) Compute $\boldsymbol{H}^{(l)}$, $\boldsymbol{A}^{(l)}$, and optionally $\boldsymbol{B}^{(l)}$ using cached $\boldsymbol{x}_t^{(l)}$, $\boldsymbol{p}_t$, $\boldsymbol{q}_t$, and $\boldsymbol{y}_t$.
   (c) Form $\boldsymbol{H}_{tot}^{(l)}$, damp to $\boldsymbol{H}_{tot}^{'(l)}$, and compute its inverse.
   (d) Execute the column-wise quantization (Section 2.3) for layer l using $(\boldsymbol{H}_{tot}^{'(l)})^{-1}$.
3. **Hyperparameter Tuning:** $\beta, \gamma, \lambda, \tau$ are tuned on a validation set.

This integrated, layer-wise approach allows KLAWQ to implicitly optimize the combined objective (Eq. 6) within a PTQ framework.

3. Experimental Results

This section presents an empirical evaluation of the KLAWQ methodology. Experiments focus on 8-bit quantization of a pre-trained GPT2, with alienc4 as calibration dataset. (Merity et al., 2016) We evaluated our quantized model on WikiText-2 using perplexity (PPL) and loss metrics. The primary objective is to quantify the impact of KL-divergence awareness, as incorporated by KLAWQ, relative to established quantization baselines.

3.1. Experimental Setup

Model and Dataset: All quantization experiments were conducted on the GPT-2 model (Radford et al., 2019). For calibration of the quantization process, a subset of 1024 samples was drawn from the C4 dataset (specifically, the 'allenai/c4' collection) (Raffel et al., 2020). The benchmark for evaluating language modeling performance (perplexity and loss) was the WikiText-2 dataset (Merity et al., 2016).

Quantization Configuration: A symmetric 8-bit quantization scheme was uniformly applied across all comparative methods. Weight quantization employed a group size of 128, a common practice for structured quantization.

Baseline Methods: The performance of KLAWQ was benchmarked against two established baseline techniques:

1. **BitsAndBytes 8-bit (BNB-8bit)**: This represents a standard 8-bit Post-Training Quantization (PTQ) approach as implemented in the widely adopted BitsAndBytes library (Dettmers et al., 2022).

2. **GPTQ 8-bit (Vanilla)**: Our implementation of the standard GPTQ algorithm, corresponding to KLAWQ with the KL-divergence strength $\beta = 0$ and cross-entropy weight $\gamma = 0$. This baseline utilizes an identical 8-bit quantization configuration to KLAWQ, isolating the effect of the KL-awareness term.

KLAWQ Configuration: For KLAWQ experiments, we systematically varied the KL divergence strength hyperparameter β and the distillation temperature τ. The optional cross-entropy component ($\mathcal{L}_{CE}$) was not activated in the present study (i.e., $\gamma = 0$).

Evaluation Metrics: Model performance was primarily quantified using cross-entropy loss and perplexity (PPL) on the WikiText-2 validation set. Lower values for both metrics indicate superior language modeling performance.

3.2. Baseline Performance Analysis

The performance of the baseline 8-bit quantization methods on WikiText-2 is presented in Table 1. Standard BNB-8bit quantization yielded a cross-entropy loss of 9.0638 and a corresponding perplexity of 8636.98. In contrast, our vanilla GPTQ implementation (8-bit, $\beta = 0$) achieved a substantially lower loss of 8.9618 and a perplexity of 7799.30. This marked improvement underscores the inherent advantages of leveraging second-order information, as employed by GPTQ, for minimizing reconstruction error during quantization.

Table 1. Baseline 8-bit quantization performance of GPT-2 on the WikiText-2 validation set.

Method	Loss	PPL
BNB-8bit	9.0638	8636.98
GPTQ 8-bit (Vanilla, $\beta = 0$)	8.9618	7799.30

3.3. KLAWQ Efficacy with KL-Divergence Awareness

To assess the impact of the KL-divergence term in KLAWQ, extensive hyperparameter sweeps were performed, focusing on the interplay between the divergence strength β and the distillation temperature τ.

3.3.1. Hyperparameter Sensitivity Analysis

An initial hyperparameter exploration involved varying β while maintaining a fixed temperature $\tau = 0.5$, followed by a subsequent sweep of τ for the empirically determined optimal β from the first stage. These results are detailed in Table 2. This investigation revealed that a KLAWQ configuration with $\beta = 0.4$ and $\tau = 0.5$ achieved the lowest perplexity (7765.01) and loss (8.9574) within this specific set of experiments. This outcome represents a marginal improvement over the vanilla GPTQ baseline (PPL 7799.30), suggesting a benefit from the KL-aware objective.

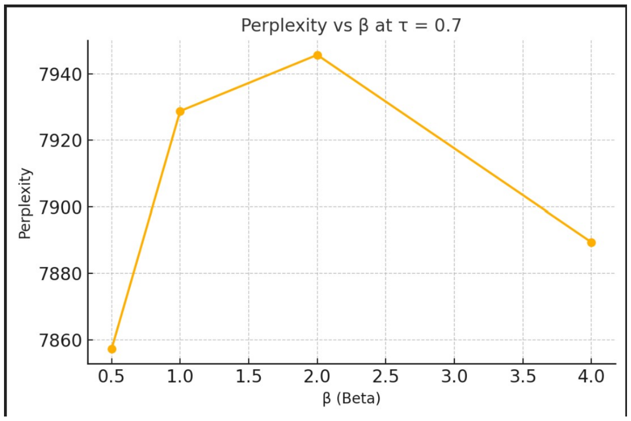
Table 2. KLAWQ performance on WikiText-2: Results from an initial sweep varying β (at $\tau = 0.5$) and subsequently τ (at optimal $\beta = 0.4$).

β	τ	Loss	PPL
0.2	0.5	8.9656	7829.30
0.3	0.5	8.9780	7926.92
0.4	**0.5**	**8.9574**	**7765.01**
0.6	0.5	8.9745	7899.04
0.4	0.2	8.9733	7889.56
0.4	0.3	8.9681	7848.91

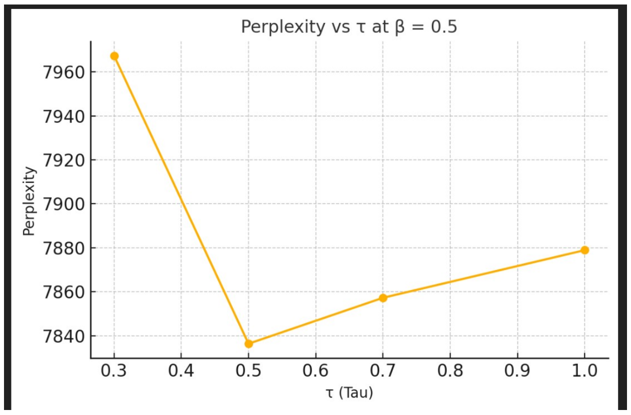
A more extensive grid search, presented in Table 3, further explored the (β, τ) hyperparameter space. This broader sweep did not identify configurations surpassing the previously established optimal PPL of 7765.01. Specifically, for a fixed $\tau = 0.7$, varying β produced perplexities in the range of 7857.26 to 7945.68. Similarly, with $\beta = 0.5$ fixed, variations in τ resulted in perplexities between 7836.36 and 7967.42.

Table 3. KLAWQ performance on WikiText-2: Results from an extended grid sweep over β and τ.

β	τ	Loss	PPL
0.5	0.7	8.9692	7857.26
1.0	0.7	8.9783	7928.76
2.0	0.7	8.9804	7945.68
4.0	0.7	8.9733	7889.40
0.5	0.3	8.9831	7967.42
0.5	0.5	8.9665	7836.36
0.5	1.0	8.9720	7878.93



(a) Perplexity vs. τ at fixed $\beta = 0.5$.



(b) Perplexity vs. β at fixed $\tau = 0.7$.

Figure 1. Hyperparameter sensitivity of KLAWQ on WikiText-2. (a) shows how perplexity changes with the distillation temperature τ at $\beta = 0.5$. (b) shows how perplexity changes with the KL-awareness strength β at $\tau = 0.7$. Both plots indicate non-monotonic behavior, suggesting the need for careful tuning to find optimal values.

4. Discussion

The empirical evaluation offers several insights into the performance of KLAWQ for 8-bit quantization of the GPT-2 model. The substantial performance gap between vanilla GPTQ and the standard BitsAndBytes 8-bit baseline highlights the role of second-order information in preserving model accuracy for post-training quantization. GPTQ's ability to minimize reconstruction error weighted by input sensitivity provides a strong foundation upon which more advanced techniques can build.

Our primary contribution, KLAWQ, introduces KL-divergence awareness into the established second-order GPTQ framework. The 8-bit precision results demonstrate that this addition can yield further, albeit modest, improvements in perplexity. The optimal KLAWQ configuration ($\beta = 0.4, \tau = 0.5$) achieved a perplexity of 7765.01, surpassing the vanilla GPTQ baseline (PPL 7799.30) also with a lower loss. This improvement suggests that explicitly guiding the quantization process to maintain functional fidelity with the teacher model, in addition to structural weight fidelity, leads to better overall language modeling performance. By incorporating the approximated KL Hessian ($\boldsymbol{A}$) into the total effective Hessian ($\boldsymbol{H}_{tot}$), KLAWQ directly alters the optimization landscape for weight quantization. This ensures that updates during the column-wise procedure (Eq. 11) are influenced not only by weight reconstruction error but also by the preservation of the teacher's output distribution. In this sense, the KL divergence term, weighted by β, acts as a **regularizer**, encouraging the quantized model to retain the teacher's output distribution characteristics, which can be particularly important for preserving nuanced linguistic behaviors.

However, the hyperparameter sensitivity observed for β and τ is a noteworthy aspect, as illustrated in Figure 1. Subfigure 1(a) shows perplexity variation with τ when β is fixed at 0.5. Subfigure 1(b) shows perplexity variation with β when τ is fixed at 0.7. Performance did not monotonically improve with increasing KL-divergence strength (β). Instead, a carefully calibrated balance appears necessary. This implies that an overly strong emphasis on mimicking the teacher's soft labels might conflict with the primary quantization objective of minimizing weight error, or that the specific approximation of the KL Hessian might become less effective at extreme β values. The distillation temperature τ also plays a role in shaping the teacher's soft targets; lower temperatures create sharper distributions, while higher temperatures soften them. The optimal $\tau = 0.5$ suggests that a moderately sharpened teacher distribution was most effective for guiding KLAWQ in this setting. Identifying these optimal, balanced values for β and τ currently requires empirical sweeps.A direction for future research is the development of more principled methods to efficiently determine these optimal hyperparameter configurations and reducing the extensive tuning required to maximize KLAWQ's performance.

The incremental nature of improvements which KLAWQ shows at 8-bit precision is worth noting. At 8-bits, which is less aggressive than 4-bit or 3-bit quantization, vanilla GPTQ might already reduce reconstruction errors effectively. This could mean that the extra guidance KLAWQ provides for functional similarity offers only a small additional benefit. KLAWQ's ability to preserve how the model works might be more impactful when quantizing to even lower bit-widths, where standard PTQ methods often struggle more. In those very low-bit situations, keeping the teacher model's decision-making process could be key.

Furthermore, the current study focused on a single model architecture (GPT-2) and dataset (WikiText-2). The generalizability of these findings, and the optimal hyperparameter ranges for KLAWQ, may vary across different model sizes, architectures, and data domains. The choice of calibration dataset for computing the Hessians could also influence outcomes, although WikiText-2 is a standard choice.

Although, the optional cross-entropy (CE) term was not activated in this study, its inclusion presents a good direction for future exploration. Directly optimizing for task-specific metrics via a CE Hessian, in conjunction with KL-awareness and reconstruction fidelity, could potentially yield even better performance when deploying models for specific downstream applications where supervised labels are available during the PTQ phase.

5. Conclusion

In this work, we introduced KLAWQ, a novel post-training quantization framework that augments the second-order optimization of GPTQ with Kullback-Leibler divergence awareness. By incorporating a KL Hessian term into the combined Hessian used for quantization, KLAWQ aims to preserve not only the structural fidelity of model weights but also the functional characteristics of the full-precision teacher model.

Our empirical evaluation on the GPT-2 model using the WikiText-2 dataset demonstrated that KLAWQ, at 8-bit precision, can achieve modest improvements in perplexity compared to a strong vanilla GPTQ baseline. We observed that the performance of KLAWQ is sensitive to the KL divergence strength (β) and distillation temperature (τ), underscoring the need for careful hyperparameter tuning.

While the gains at 8-bit were incremental, KLAWQ provides a theoretically grounded approach to enhance PTQ by explicitly considering functional preservation. Future research should focus on evaluating KLAWQ in more aggressive quantization regimes (e.g., 4-bit and below), where its benefits may be more pronounced. Exploring its efficacy across diverse model architectures and datasets, as well as studying the impact of the optional cross-entropy augmentation for task-specific quantization, offers a good research direction. KLAWQ represents a step towards developing more robust and accurate quantization techniques for deploying large language models efficiently on resource-constrained edge devices.

Contributions

M. Haseeb and H. Iqbal jointly conceived the initial idea and designed the KLAWQ methodology. M. Haseeb was primarily responsible for developing the theoretical formalism and conducting experiments. H. Iqbal implemented the algorithm in code. Both authors collaborated on the analysis of results and the writing of the manuscript.

Acknowledgements

The authors would like to express their gratitude to Saad Haroon for guidance on research report. We also thank Muzammil from the CVG Lab at LUMS for providing support to deploy and run models on Jetson.

References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Dettmers, T., Lewis, M., Shleifer, S., and Zettlemoyer, L. 8-bit optimizers via block-wise quantization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=shpkpVXzo3h.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Gptq: Accurate post-training quantization for generative pretrained transformers. In *Advances in Neural Information Processing Systems*, pp. 4567–4579, 2023.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. URL https://arxiv.org/abs/1609.07843.

Qin, T., Yang, W., and Liu, J. A comparative study of quantization and distillation for llm compression. In *International Conference on Learning Representations*, 2024.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (140):1–67, 2020.

Xu, L. and Zhang, P. On the role of kl divergence in knowledge distillation. *Journal of Machine Learning Research*, 25(1):1–15, 2024.

Zhu, X., Li, Y., and Wang, Z. Efficient quantization for large language models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1234–1243, 2024.