KL- and CE-Aware GPTQ Quantization

Haseeb-26100253, Hamza-26100130

April 2025

Abstract

We propose an improved GPTQ quantization method that incorporates both KL-distillation and optional cross-entropy fine-tuning. Each step is designed for clarity and practical deployment, with explicit scaling and normalization to balance reconstruction error, teacher mimicking, and supervised accuracy.

1. Motivation and Overview

Quantization reduces model size and inference cost but risks degrading accuracy. Standard GPTQ minimizes a weighted mean-square error (MSE) under the empirical input Hessian. We extend this by:

- Injecting a second-order KL-distillation curvature so the quantized model mimics the teacher's soft outputs.
- Optionally adding a supervised cross-entropy (CE) term for true-label alignment.
- Normalizing all curvature contributions so a single scale parameter (β, γ) suffices.

2. Step-by-Step Procedure

Each numbered section below corresponds to a phase in our pipeline. Brief explanations are provided for clarity.

2.1 Empirical Hessian Computation

What: Compute the average second-moment of layer inputs.

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

Why: $\boldsymbol{H}$ captures the local quadratic geometry of the MSE loss in weight space, weighting directions by input variance.

2.2 Hessian Damping and Inversion

What: Stabilize and invert $\boldsymbol{H}$:

$$\boldsymbol{H}' = \boldsymbol{H} + \lambda \frac{\mathrm{tr}(\boldsymbol{H})}{N}\boldsymbol{I}, \quad \boldsymbol{H}'^{-1} = (\boldsymbol{H}')^{-1}.$$

Why: A damping term λ ensures invertibility and controls numerical conditioning for the subsequent column updates.

2.3 Vanilla GPTQ Objective

What: Define the baseline quantization loss:

$$L_{\mathrm{MSE}}(\boldsymbol{Q}) = \big((\boldsymbol{W} - \boldsymbol{Q})^\top \boldsymbol{H} (\boldsymbol{W} - \boldsymbol{Q})\big).$$

Why: Minimizing this preserves weighted reconstruction fidelity under the input distribution.

2.4 Column-Wise Quantization Loop

For each column $i = 1 \dots N$:

1. *Compute scale* $d_i = \sqrt{(\boldsymbol{H}'^{-1})_{ii}}$.
   Purpose: Scale transforms each weight into a unit-curvature coordinate.
2. *Quantize* $\boldsymbol{q}_i = d_i(\boldsymbol{w}_i / d_i)$.
   Purpose: Rounds to nearest integer grid in scaled space, minimizing local quadratic loss.
3. *Error* $\boldsymbol{e}_i = \boldsymbol{w}_i - \boldsymbol{q}_i$.
   Why track it: Error propagates to future columns.
4. *Propagate* to $j > i$: $\boldsymbol{w}_j \leftarrow \boldsymbol{w}_j - \frac{(\boldsymbol{H}'^{-1})_{ij}}{(\boldsymbol{H}'^{-1})_{ii}} \boldsymbol{e}_i$.
   Purpose: Corrects remaining weights for the quantization error along mutual curvature directions.

2.5 Column Ordering (Optional)

What: Sort columns by descending $(\boldsymbol{H}'^{-1})_{ii}$ before quantization.
Why: Prioritizes columns with higher sensitivity (curvature), improving overall fidelity.

2.6 Average Per-Column Loss Reporting

What: Compute

$$\mathrm{Loss}_{\mathrm{avg}} = \frac{1}{N} \sum_i \frac{\|\boldsymbol{w}_i^{\mathrm{before}} - \boldsymbol{q}_i\|^2}{2(\boldsymbol{H}'^{-1})_{ii}}.$$

Why: Provides a normalized metric for tuning and comparison across layers.

3. KL- and CE-Augmentation

We now add two curvature terms for distillation and optional supervised fine-tuning.

3.1 Loss Function

Define model outputs for input $\boldsymbol{x}_t$:

$$p_t = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{x}_t/\tau), \quad q_t = \mathrm{softmax}(\boldsymbol{Q}\boldsymbol{x}_t/\tau).$$

Augmented loss:

$$L(\boldsymbol{Q}) = L_{\mathrm{MSE}}(\boldsymbol{Q}) + \beta \sum_t \mathrm{KL}(p_t \| q_t) + \gamma \sum_t \mathrm{CE}(y_t, q_t),$$

Note: $\gamma = 0$ turns off supervised fine-tuning; $\beta = 0$ recovers pure CE-based quantization.

3.2 Global Second-Order Curvature

What: Approximate second-order contributions in input space:

$$A = \frac{1}{T} \sum_t \frac{1}{\tau} \sum_k q_{t,k}(1 - q_{t,k})\, \boldsymbol{x}_t \boldsymbol{x}_t^\top, B = \frac{1}{T} \sum_t [\mathrm{diag}(q_t) - q_t q_t^\top]\, \boldsymbol{x}_t \boldsymbol{x}_t^\top.$$

Normalization: Let $s_H = \mathrm{tr}(H), s_A = \mathrm{tr}(A), s_B = \mathrm{tr}(B)$. Set $A' = (s_H/s_A)A,\ B' = (s_H/s_B)B$.

3.3 Combined Hessian and Quantization

What: Form and damp combined curvature:

$$\boldsymbol{H}_{\text{tot}} = \boldsymbol{H} + \beta A' + \gamma B', \quad \boldsymbol{H}'_{\text{tot}} = \boldsymbol{H}_{\text{tot}} + \lambda \frac{\text{tr}(\boldsymbol{H}_{\text{tot}})}{N} \boldsymbol{I}.$$

Invert via Cholesky and apply the column-loop from Section 2.4 using $\boldsymbol{H}'_{\text{tot}}$ in place of $\boldsymbol{H}$.

4. Complexity and Practical Notes

- H, A, B: cost $O(TN^2)$ each, dominates calibration pass.
- Cholesky: $O(N^3)$. Column loop: $O(N^2 M)$.
- Tune $\beta, \gamma, \lambda, \tau$ on a held-out set.
- Choose CE-based fine-tuning ($\gamma > 0$) if labels are available; else set $\gamma = 0$.
- Column ordering can be enabled selectively for critical layers.

5. Workflow Overview

This high-level summary shows how the pieces fit together:

1. **Calibration Pass:** One forward-only pass through the full-precision (teacher) network to cache all layer inputs $x_t^{(\ell)}$ and compute teacher soft targets p_t. No quantization is applied yet.
2. **Layer Loop ($\ell = 1$ to L):** For each layer:
   - Obtain student soft targets q_t by passing x_t through layers $1 \ldots \ell - 1$ (quantized) and ℓL (full-precision).
   - Build input-space Hessians $H^{(\ell)}$, KL Hessian $A^{(\ell)}$, and (optional) CE Hessian $B^{(\ell)}$ using cached $x_t^{(\ell)}$ and (p_t, q_t).
   - Trace-normalize and combine into $H_{\text{tot}}^{(\ell)}$, damp and invert via Cholesky.
   - Run the GPTQ column-wise quantization loop on layer ℓ using $H_{\text{tot}}^{'(\ell)}$.
3. **Completion:** After processing all layers, the student is fully quantized, having implicitly optimized the combined MSE+KL(+CE) surrogate in a single pass.
4. **Tuning:** Balance $\beta, \gamma, \lambda, \tau$ on validation data to trade off reconstruction, distillation fidelity, and supervised accuracy.