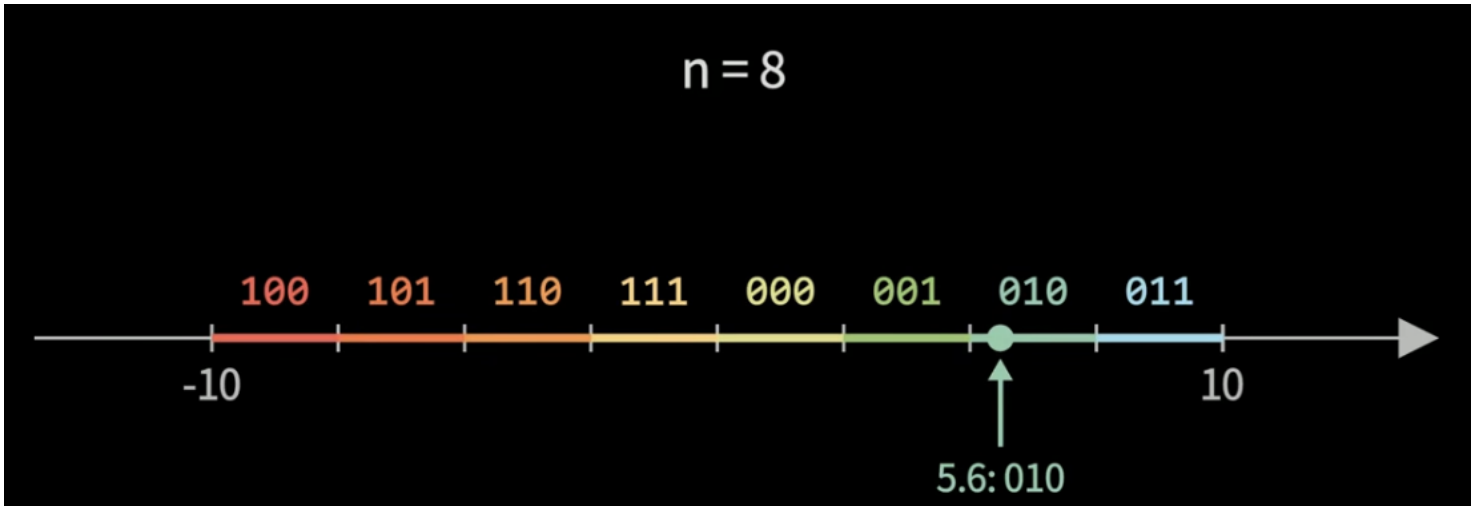
Review on TurboQuant:Online Vector Quantization with Near-optimal Distortion Rate

Department of Statistics, Seoul National University

June 5, 2026

Outline

Quantization



- Quantization encodes each coordinate of a real-valued vector using a b-bit string.

$$Q : \mathbb{R}^d \to \{0,1\}^{bd}, \qquad Q^{-1} : \{0,1\}^{bd} \to \mathbb{R}^d$$

- Dequantization reconstructs each coordinate using a predefined representative real value.
- Example:

$$(-9.9, 5.6) \xrightarrow{\text{3-bit quantization}} (100, 010) \xrightarrow{\text{dequantization}} (-10, 5)$$

LLM Quantization

- TurboQuant is an online quantization method for the key and value vectors generated during LLM inference.
- When stored in 16-bit precision, the KV cache for LLaMA-7B requires approximately 512 MB of memory to generate a sequence of 1,024 tokens in a single inference run.
- Applying b-bit quantization reduces the required memory to approximately $\frac{b}{16}$ of the original size.

TurboQuant: Objective

- TurboQuant considers two distortion measures:

$$D_{\text{mse}}(x) := \mathbb{E}_Q \left[\left\| x - Q^{-1}(Q(x)) \right\|_2^2 \right]$$

$$D_{\text{prod}}(x; q) := \mathbb{E}_Q \left[\left| \langle q, x \rangle - \langle q, Q^{-1}(Q(x)) \rangle \right|^2 \right]$$

- MSE-optimal quantization reduces the reconstruction error, but it does not guarantee that the inner-product estimator is unbiased.
- Therefore, TurboQuant:

MSE-optimal quantization $+$ inner product bias correction

Quantize Key/Value Vector: Algorithm

- Generate a random rotation matrix Π once and reuse it:

$$G_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \qquad G = UR \quad \text{(QR decomposition)}, \qquad \Pi = U$$

- For a unit-norm vector $x \in \mathbb{S}^d$, rotate the vector:

$$y = \Pi x \sim \text{Unif}(\mathbb{S}^d)$$

Then each coordinate has the same marginal distribution:

$$\frac{y_j + 1}{2} \overset{\text{i.i.d.}}{\sim} \text{Beta}\left(\frac{d-1}{2}, \frac{d-1}{2}\right), \qquad y_j \overset{\text{i.i.d.}}{\approx} \mathcal{N}\left(0, \frac{1}{d}\right)$$

- Encode each coordinate to nearest centroid $C = (c_1, \ldots c_{2^d}) \in \mathbb{R}^{2^d}$

$$c_{y_j} = \arg\min_{c_k \in C} |y_j - c_k|, \qquad Q_{\text{mse}}(x) = (c_{y_1}, \ldots, c_{y_d}) \in \mathbb{R}^d$$

- Decode rotate back:

$$Q_{\text{mse}}^{-1}\big(Q_{\text{mse}}(x)\big) = \widetilde{x} = \Pi^\top Q_{\text{mse}(x)}$$

Quantize Key/Value Vector: Theory

- The centroids are chosen to minimize the scalar quantization error:

$$C(f_Y, b) := \min_{c_1,\dots,c_{2^b}} \sum_{k=1}^{2^b} \int_{V_k} (t - c_k)^2 f_Y(t)\, dt$$

where $V_k = [\frac{c_k + c_{k+1}}{2}, \frac{c_{k-1} + c_k}{2}]$.

- Since Π is an orthogonal matrix, the MSE satisfies

$$\begin{aligned} D_{\text{mse}}(x) &:= \mathbb{E}_\Pi \left[\|x - \widetilde{x}\|_2^2 \right] \\ &= \mathbb{E}_\Pi \left[\left\| x - \Pi^\top Q_{\text{mse}}(x) \right\|_2^2 \right] = \mathbb{E}_\Pi \left[\|\Pi x - Q_{\text{mse}}(x)\|_2^2 \right] \\ &= \sum_{j=1}^{d} \mathbb{E}_\Pi \left[|y_j - c_{y_j}|^2 \right] = d \cdot C(f_Y, b) \le \frac{\sqrt{3}\pi}{2} \cdot 4^{-b} \end{aligned}$$

Inner Product Unbiasedness: Algorithm

- QJL:

$$S \in \mathbb{R}^d, \qquad s_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$$

$$Q_{\text{QJL}} : \mathbb{R}^d \to \{-1,+1\}, \qquad Q_{\text{QJL}}(x) := \text{sign}(Sx) \in \{-1,1\}$$

$$Q_{\text{QJL}}^{-1} : \{-1,+1\} \to \mathbb{R}^d, \qquad Q_{\text{QJL}}^{-1}(z) := \frac{\sqrt{\pi/2}}{d} S^\top z$$

- For any $q \in \mathbb{R}^d$, QJL is a 1-bit quantization method that preserves the inner product in expectation.

$$\mathbb{E}[< q, Q_{\text{QJL}}^{-1}(Q_{\text{QJL}}(x)) >] =< q, x >$$

- Apply QJL to the residual

$$r := x - Q_{\text{mse}}^{-1}\big(Q_{\text{mse}}(x)\big).$$

$$\tilde{x}_{\text{QJL}} = Q_{\text{QJL}}^{-1}(Q_{\text{QJL}}(r))$$

- Final TurboQuant result:

$$\tilde{x} = \tilde{x}_{\text{mse}} + \tilde{x}_{\text{QJL}}$$

Inner Product Unbiasedness: Theory

- The final reconstruction preserves the inner product in expectation:

$$\mathbb{E}\left[\langle y, \widetilde{x}_{\mathrm{QJL}}\rangle \,|\, \widetilde{x}_{\mathrm{mse}}\right] = \langle y, r\rangle$$

$$\mathbb{E}\left[\langle y, \widetilde{x}\rangle \,|\, \widetilde{x}_{\mathrm{mse}}\right] = \langle y, \widetilde{x}_{\mathrm{mse}}\rangle + \langle y, r\rangle = \langle y, x\rangle$$

- The inner-product distortion D_{prod} is controlled by the residual MSE:

$$D_{\mathrm{prod}}(x) \leq \frac{\pi}{2d}\|q\|_2^2 \mathbb{E}\left[\|x - \widetilde{x}_{\mathrm{mse}}\|_2^2\right]$$

$$D_{\mathrm{prod}}(x) \leq \frac{\sqrt{3}\pi^2\|q\|_2^2}{d} \cdot 4^{-b}$$

Why Near Optimal?

- Near optimality is established by comparing the upper and lower bounds on the distortion rate.
- TurboQuant upper bounds:

$$D_{\text{mse}}^{\text{TQ}}(x) \leq \frac{\sqrt{3}\pi}{2} \cdot 4^{-b}, \qquad \forall x \in \mathbb{S}^d$$

$$D_{\text{prod}}^{\text{TQ}}(x, q) \leq \frac{\sqrt{3}\pi^2 \|q\|_2^2}{d} \cdot 4^{-b}, \qquad \forall x \in \mathbb{S}^d,\ \forall q \in \mathbb{R}^d$$

- Using Yao's minimax principle and the Shannon lower bound, for any randomized b-bit quantizer Q, there exists a hard input $x \in \mathbb{S}^d$ such that

$$D_{\text{mse}}^{Q}(x) \geq 4^{-b}$$

and there exists a query vector $q \in \mathbb{S}^d$ such that

$$D_{\text{prod}}^{Q}(x, q) \geq \frac{1}{d} \cdot 4^{-b}$$

Outline

Appendix: Shannon Lower Bound

Shannon Lower Bound for MSE Distortion

Let $X \in \mathbb{R}^d$ be a random vector with finite differential entropy $h(X)$. Define the distortion-rate function

$$D(p_X, B) := \inf_{\substack{P_{\widehat{X}|X}: \\ I(X;\widehat{X}) \le B}} \mathbb{E}\left[\|X - \widehat{X}\|_2^2\right].$$

Then,

$$D(p_X, B) \ge \frac{d}{2\pi e} \cdot 2^{\frac{2}{d}(h(X)-B)}.$$

Corollary: SLB for a Random Point on the Hypersphere

Let

$$X \sim \mathrm{Unif}\left(\mathbb{S}^d\right)$$

and let $D(B)$ denote the optimal MSE distortion achievable with a total bit budget B. Then,

$$D(B) \ge 2^{-2B/d}$$

Appendix: Yao's minimax principle

Yao's Minimax Principle

Let $\mathcal{A}$ be a set of deterministic algorithms, $\mathcal{X}$ a set of inputs, and $\ell(A, x)$ the loss of $A \in \mathcal{A}$ on $x \in \mathcal{X}$. For finite $\mathcal{A}$ and $\mathcal{X}$,

$$\min_{\rho \in \Delta(\mathcal{A})} \max_{x \in \mathcal{X}} \mathbb{E}_{A \sim \rho} \left[\ell(A, x)\right] = \max_{\mu \in \Delta(\mathcal{X})} \min_{A \in \mathcal{A}} \mathbb{E}_{x \sim \mu} \left[\ell(A, x)\right].$$

In particular, for any input distribution μ,

$$\min_{\rho \in \Delta(\mathcal{A})} \max_{x \in \mathcal{X}} \mathbb{E}_{A \sim \rho} \left[\ell(A, x)\right] \geq \min_{A \in \mathcal{A}} \mathbb{E}_{x \sim \mu} \left[\ell(A, x)\right].$$