Applying Post-Training Quantization to Vision Transformers

2025 동계 세미나 – 25.02.21

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
최재민

Outline

- Introduction
  - Quantization

- Papers
  - NoisyQuant: Noisy Bias-Enhanced Post-Training Activation Quantization for Vision Transformers [CVPR 2023]
  - ERQ: Error Reduction for Post-Training Quantization of Vision Transformers [ICML 2024]

Introduction

- Quantization
  - 일반적으로 performance와 model size는 비례하는 경향이 있음
    - Model size ↑ → inference time ↑, computational cost ↑
    - 메모리 용량이 제한적인 edge device 환경에서 한계가 존재함
  - Full-precision → Low-precision
    - Weight, activation을 16bit 이하의 low-precision으로 낮춘 후 연산을 수행하여 inference time을 줄이는 가속화 기법

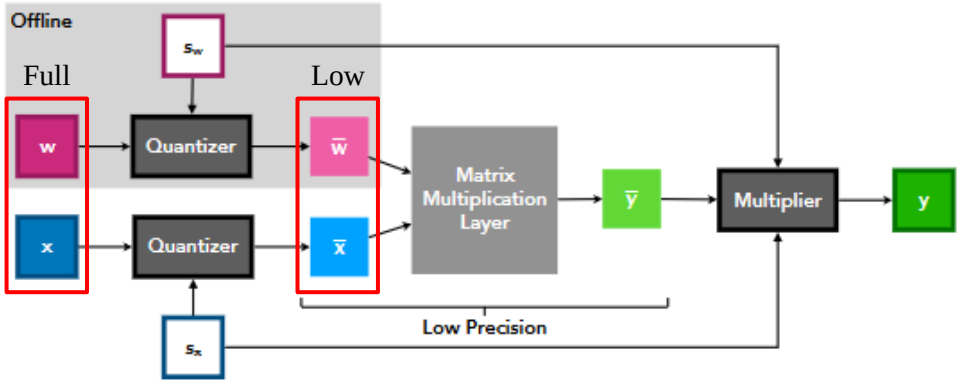


Fig 1. Quantization을 활용한 가속화 원리[1)]

- Quantization process : $\bar{x} = Q(x) = clamp\left(\left\lfloor \frac{x}{s} \right\rfloor + z, 0, 2^{bit} - 1\right), s = \frac{\max(x) - \min(x)}{2^{bit} - 1}, z = \left\lfloor -\frac{\min(x)}{s} \right\rceil$
- Dequantization process : $\hat{x} = s \cdot \bar{x}$
- Fully-Connected layer에서의 연산 : $f = WX + B$ → FP32 Matmul
- Quantized FC layer에서의 연산 : $\hat{f} = \hat{W}\hat{X} + B = (s_W \bar{W})(s_X \bar{X}) + B = s_W s_X (\bar{W}\bar{X}) + B$ → INT8 Matmul

Introduction

- Quantization

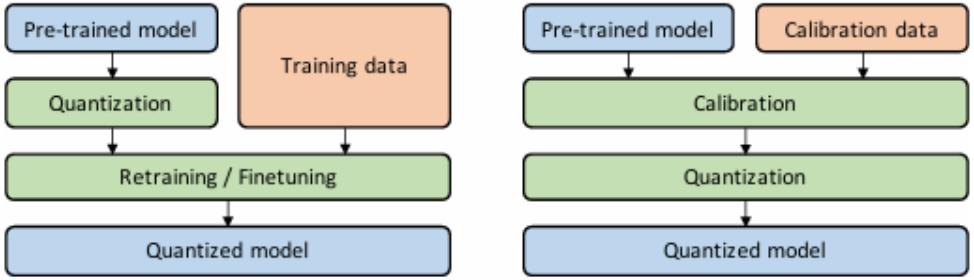


Fig 2. Comparison between QAT and PTQ

- Quantization-Aware Training (QAT)
  - Quantization 적용 후 pre-trained model의 train dataset으로 retraining/fine-tuning하는 방식
  - Retraining/fine-tuning 과정이 많은 시간을 필요로 함
  - PTQ에 비해 좋은 성능

- Post-Training Quantization (PTQ)
  - 소량의 데이터(calibration dataset)만으로 pre-trained model에서의 weight, activation 등의 파라미터들을 보정
  - Inference 과정에서 quantization 적용 → inference time ↓
  - 소량의 데이터만을 사용하기 때문에 적은 시간만이 필요함
  - QAT에 비해 낮은 성능

Introduction

- Linear quantizer
  - Ex) $Q(x) = \bar{x} = clamp\left(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^{bit} - 1\right), DQ(\bar{x}) = \hat{x} = \bar{x} \cdot s$
    - Uniform distribution과 같이 x의 분포가 고루 퍼져 있는 경우에 적합
    - CNN 기반 모델에서 주로 사용됨

- Non-linear quantizer
  - Ex) $Q(x) = \bar{x} = clamp\left(\left\lfloor -\log_2 \frac{x}{s} \right\rceil + z, 0, 2^{bit} - 1\right), DQ(\bar{x}) = 2^{-\bar{x}} \cdot s$
    - Power-law distribution과 같이 x가 작은 값에 쏠려 있는 경우에 적합
    - Transformer 기반 모델의 self-attention을 효율적으로 반영할 수 있어 ViT 기반 모델에서 주로 사용됨

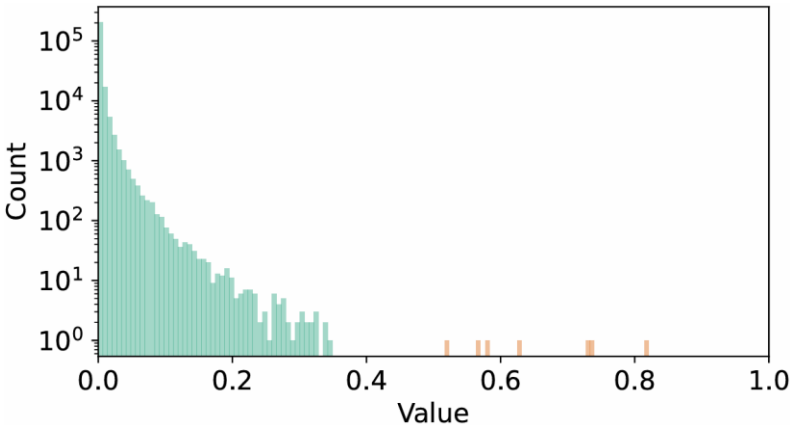


Fig 3. Histogram of the post-Softmax activations

NoisyQuant: Noisy Bias-Enhanced Post-Training Activation Quantization for Vision Transformers [CVPR 2023]

NoisyQuant[1)]

- Introduction
  - ViT 구조에 PTQ 방법론을 적용하는 기존의 연구들은 self-attention의 power-law distribution에 적합한 quantizer를 구하는 것
    - Ex) log 2, log$\sqrt{2}$ quantizer를 사용하여 특정 값에 몰려 있는 영역에 대해 더욱 촘촘한 양자화 값 설정
  - 본 논문에서는 power-law distribution을 기존의 quantizer에 적합하도록 변환하기 위해 noise를 첨가하는 방식을 최초로 제안함
    - Activation에 noise를 첨가한 후 quantization을 수행하면 quantization error를 줄일 수 있음을 발견

- Key contributions
  - 1. Noisy bias를 첨가한 후 quantization을 수행하면 quantization error를 줄일 수 있음을 이론적으로 증명하였음
  - 2. 증명한 사실을 토대로 NoisyQuant라는 PTQ 방법론을 제안하고, 최적의 quantizer를 탐색하는 기존의 연구와는 다른 가능성을 제시하였음

NoisyQuant[1)]

- Objective
  - Quantization의 주요 목표
    - floating-point value와 dequantized value의 차이를 줄이는 것
  - 본 논문의 목표 설정

$$D(X,N) = QE(X+N) - QE(X) = \|Q(X+N) - (X+N)\|_2^2 - \|Q(X) - X\|_2^2 \leq 0$$

    - 즉, quantization error가 감소하도록 하는 noise bias N을 찾는 것

- Theorem
  - Quantization bin = 2b (quantization value : {…, -3b, -b, b, 3b, …})
  - $N \sim U(-n, n),\ x \leq n \leq 2b - x$를 만족하는 noise bias N에 대하여 아래의 부등식이 성립

$$0 \leq x \leq n\left(1 - \sqrt{\frac{n}{3b}}\right)\text{에 대하여, } D(X,N) \leq 0\text{가 성립}$$

  - Proof)
    - 1. Quantization value : {…, -3b, -b, b, 3b, …}에 대하여 $0 \leq x \leq b$인 경우 $Q(x) = b, QE(x) = (b-x)^2$
    - 2. $N \sim U(-n, n)$에 대하여, $X + N \sim U(x-n,\ x+n)$이므로 아래와 같은 부등식이 성립

$$Q(X_i + N_i) = \begin{cases} b, & (x \leq n \leq 2b - x, N_i \in [-x, n]) \\ -b, & (x \leq n \leq 2b - x, N_i \in [-n, -x)) \end{cases}$$

NoisyQuant[1)]

- Theorem
  - Proof)
    - 1. Quantization value : {…, -3b, -b, b, 3b, …}에 대하여 $0 \le x \le b$인 경우 $Q(x) = b, QE(x) = (b-x)^2$
    - 2. $N \sim U(-n, n)$에 대하여, $X + N \sim U(x-n,\ x+n)$이므로 아래와 같은 부등식이 성립

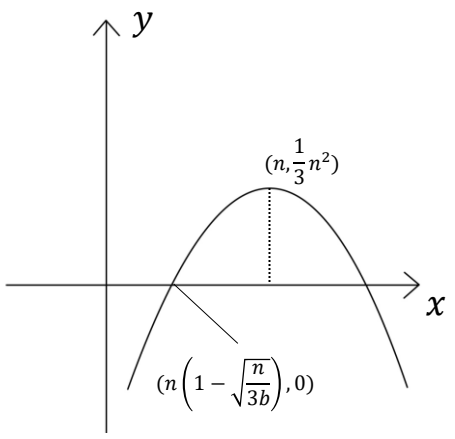
$$Q(X_i + N_i) = \begin{cases} b, & (x \le n \le 2b - x, N_i \in [-x, n]) \\ -b, & (x \le n \le 2b - x, N_i \in [-n, -x)) \end{cases}$$

    - 3. Weak Law of Large Numbers에 따라, 아래와 같은 수식이 성립

$$E_N[QE(X+N)] = \int_{x-n}^{x+n} QE(x+k) \cdot \frac{1}{2n} dk = \int_{x-n}^{0} (b+k)^2 \cdot \frac{1}{2n} dk + \int_{0}^{x+n} (b-k)^2 \cdot \frac{1}{2n} dk$$

$$= x^2 - \frac{b}{n}x^2 + \frac{1}{3}n^2 - nb + b^2$$

    - 4. $D(x,N) = QE(X+N) - QE(X) = E_N[QE(X+N)] - QE(X) = -\frac{b}{n}x^2 + 2bx + \frac{1}{3}n^2 - nb = -\frac{b}{n}(x-n)^2 + \frac{1}{3}n^2$
    - 5. $D(x,N) = -\frac{b}{n}(x-n)^2 + \frac{1}{3}n^2 = 0 \rightarrow x = n \pm \sqrt{\frac{n^3}{3b}}$
    - 6. $0 \le x \le b$에 대하여 $D(x,N) \le 0$ 의 해는 $0 \le x \le n\left(1 - \sqrt{\frac{n}{3b}}\right)$
    - 7. ∴ Quantization error를 줄일 수 있는 noise bias는 필연적으로 존재함



NoisyQuant[1)]

- Method
  - Stage 1. Noise addition
    - $0 \le x \le n\left(1-\sqrt{\frac{n}{3b}}\right)$를 만족하는 n을 설정하기 위해 x의 분포가 필요
    - Calibration data의 일부를 입력하여 x의 분포를 취득하고, $L(n)=\sum_{x\in X}[D(x,N)]$이 최소가 되도록 하는 n을 설정
    - Uniform distribution noise bias $N\sim U(-n,n)$을 activation X에 더해 $X+N$를 얻음

Stage 1. Noise addition

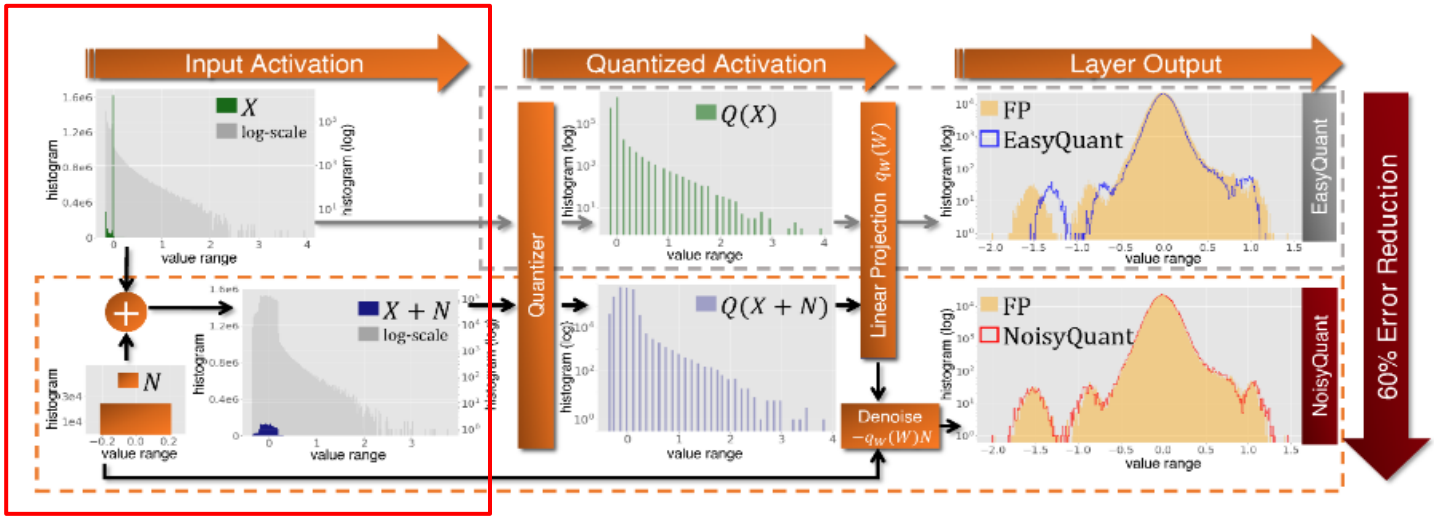


Fig 1. The overview of the NoisyQuant pipeline

NoisyQuant[1)]

- Method
  - Stage 2. Quantization
    - $X + N$에 quantization을 적용, 기존의 power-law distribution에 비해 완화된 distribution을 취득
  - Stage 3. Denoise
    - $f_{N_q}(X) = q_W(W)q_A(X + N) + B'$에서 $B' = (B - q_W(W)N)$를 채택하여 noise addition에 의한 변형을 제거

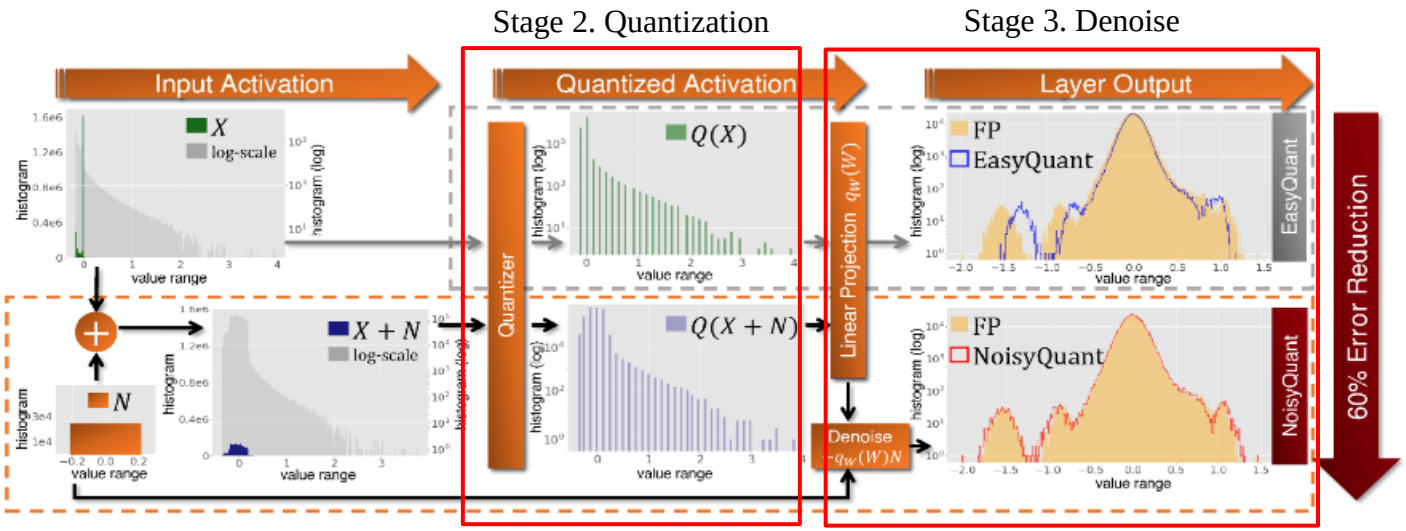


Fig 1. The overview of the NoisyQuant pipeline

NoisyQuant[1)]

- Empirical verification of theorem
  - $b = 1$로 설정, x와 n에 따른 $D(X, N)$

$$D(x, N) = -\frac{b}{n}(x - n)^2 + \frac{1}{3}n^2$$

- X의 모든 요소를 0.1로 설정하고 n과 $D(X, N)$ 사이의 관계를 측정한 결과, 이론적인 수식과 같은 경향성을 확인
  - $n = 1.4$일 때 $D(x, N)$가 최소이므로 $n = 1.4$를 선택
- $n = 1.4$를 선택한 후 x와 $D(X, N)$ 사이의 관계를 측정한 결과, 이론적인 수식과 같은 경향성을 확인
- 이는 이론적으로 분석한 결과가 실제로 타당함을 뒷받침함

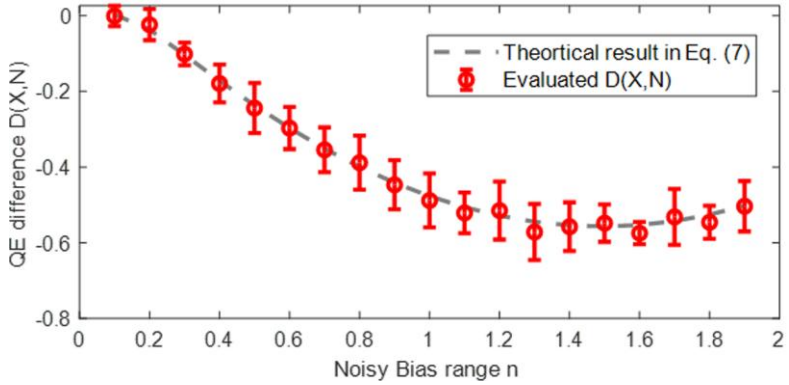


Fig 2. Graph of n vs $D(X, N)$ $(x = 0.1)$

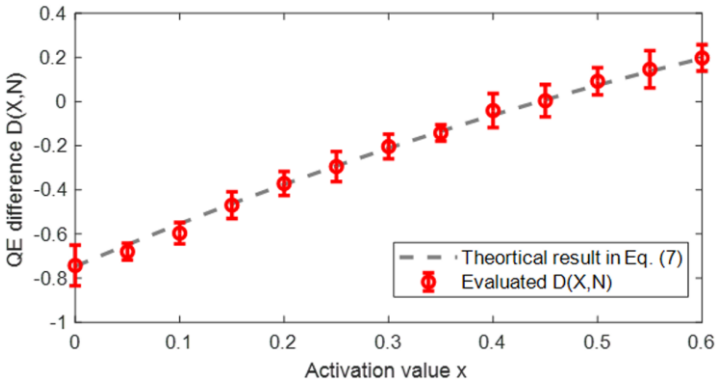


Fig 3. Graph of x vs $D(X, N)$ $(n = 1.4)$

NoisyQuant[1)]

- Empirical verification of Theorem
  - EasyQuant[2)]와의 비교
    - Fig 4 : EasyQuant에 비해 quantization error가 현저하게 감소한 모습 확인

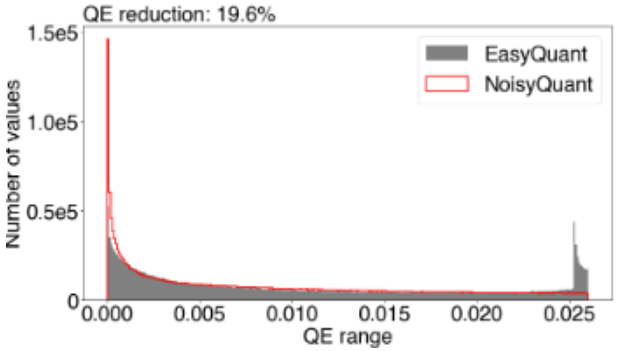


Fig 4. QE histogram of fc2 input activation

- Fig 5 : EasyQuant에 비해 floating-point value의 shape을 더욱 잘 나타내는 모습을 확인

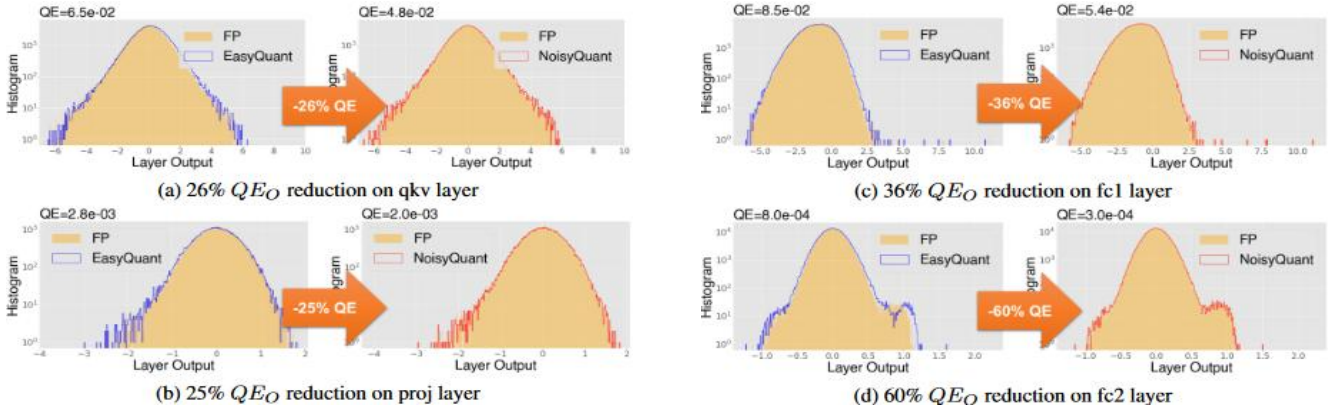


Fig 5. Output histogram in selected transformer layers

NoisyQuant[1)]

- Experiments
  - ViT 기반 모델 적용 실험
    - Table 1 ~ 2 : 8bits 환경에서 full-precision 대비 1% 이내의 오차, 6bits 환경에서 full-precision 대비 5% 이내의 오차 확인

Method	W/A	ViT-S	ViT-B	ViT-B*
Pretrained	32/32	81.39	84.54	86.00
Percentile [18]	6/6	67.74	77.63	77.60
Liu *et al.* [23]	6 MP	-	75.26	-
PTQ4ViT-Linear [43]	6/6	70.24	75.66	46.88
EasyQuant [36]	6/6	75.13	81.42	82.02
NoisyQuant-Linear	6/6	**76.86**±0.06	**81.90**±0.11	**83.00**±0.09
PTQ4ViT [43]	6/6	78.63	81.65	**83.34**
NoisyQuant-PTQ4ViT	6/6	**78.65**±0.07	**82.32**±0.09	83.22±0.10
Percentile [18]	8/8	78.77	80.12	82.53
Liu *et al.* [23]	8 MP	-	76.98	-
FQ-ViT [21]	8/8	-	83.31	-
PTQ4ViT-Linear [43]	8/8	80.46	83.89	85.35
EasyQuant [36]	8/8	80.75	83.89	85.53
NoisyQuant-Linear	8/8	**80.81**±0.01	**84.10**±0.03	**85.56**±0.01
PTQ4ViT [43]	8/8	81.00	**84.25**	85.82
NoisyQuant-PTQ4ViT	8/8	**81.15**±0.02	84.22±0.01	**85.86**±0.01

Table 1. 6~8bits에서의 실험 (ImageNet, ViT)

Method	W/A	Swin-T	Swin-S	Swin-B
Pretrained	32/32	81.39	83.23	85.27
Percentile [18]	6/6	77.75	80.41	81.90
PTQ4VIT-Linear [43]	6/6	78.45	81.74	83.35
EasyQuant [36]	6/6	79.51	82.45	84.30
NoisyQuant-Linear	6/6	**80.01**±0.06	**82.78**±0.04	**84.57**±0.04
PTQ4VIT [43]	6/6	80.47	82.38	84.01
NoisyQuant-PTQ4ViT	6/6	**80.51**±0.03	**82.86**±0.05	**84.68**±0.06
Percentile [18]	8/8	79.88	80.93	83.08
FQ-ViT [21]	8/8	80.51	82.71	-
PTQ4VIT [43]	8/8	80.96	82.75	84.79
EasyQuant [36]	8/8	80.95	83.00	85.10
NoisyQuant-Linear	8/8	**81.05**±0.03	**83.07**±0.03	**85.11**±0.04
PTQ4VIT [43]	8/8	81.24	83.10	85.14
NoisyQuant-PTQ4ViT	8/8	**81.25**±0.02	**83.13**±0.01	**85.20**±0.03

Table 2. 6~8bits에서의 실험 (ImageNet, Swin)

NoisyQuant[1)]

- Experiments
  - ViT 기반 모델 적용 실험
    - Table 3 : COCO dataset에 대하여 detection task에 적용 시 최고 성능 확인

Model	Method	W/A	mAP
DETR [4] (COCO2017)	Pretrained	32/32	42.0
	Percentile [18]	8/8	38.6
	Bit-Split [32]	8/8	40.6
	Liu *et al.* [23]	8/8	41.2
	EasyQuant [36]	8/8	41.1
	NoisyQuant-Linear	8/8	**41.4**±0.05

Table 3. 8bits 환경에서의 실험 (COCO, DETR)

    - Table 4 : fc2 layer에 적용되었을 때 가장 큰 성능 향상을 보임 (GELU의 power-law distribution에 좋은 성능을 보임)

Model	qkv noise	proj noise	fc1 noise	fc2 noise	Top-1 W6A6
DeiT-S [30]	✗	✗	✗	✗	75.27
	✓	✗	✗	✗	75.38
	✗	✓	✗	✗	75.45
	✗	✗	✓	✗	75.33
	✗	✗	✗	✓	76.21
	✓	✓	✓	✓	**76.37**
Swin-T [22]	✗	✗	✗	✗	79.51
	✓	✗	✗	✗	79.53
	✗	✓	✗	✗	79.56
	✗	✗	✓	✗	79.52
	✗	✗	✗	✓	79.80
	✓	✓	✓	✓	**80.01**

Table 4. NoisyQuant on Different layers (ImageNet, W6A6 DeiT, Swin)

ERQ: Error Reduction for Post-Training Quantization of Vision Transformers [ICML 2024]

ERQ[1)]

- Introduction
  - ViT 기반 모델의 power-law distribution과 무관하게 적용 가능한 PTQ 방법론이 필요함
  - Activation quantization과 weight quantization으로 인해 발생하는 error를 모두 고려할 필요가 있음
    - Activation/weight quantization이 동시에 고려하는 수식을 분석하는 것은 어려운 문제

- Key contributions
  - Activation quantization과 weight quantization을 모두 고려한 Error Reduction Quantization(ERQ)을 제시함
    - Activation quantization error reduction (Aqer)
      - Activation quantization으로 인해 발생하는 error를 수식적으로 분석
      - 이를 ridge regression problem으로 연결하여 error를 보정하는 방향으로 weight update하였음
    - Weight quantization error reduction (Wqer)
      - Aqer의 결과가 반영된 weight를 quantization하여 발생하는 error를 수식적으로 분석
      - Weight의 절반씩 update하며 점진적으로 error를 완화하는 메커니즘을 제안하였음

ERQ[1)]

- Objective
  - Activation/weight quantization error에 의해 발생하는 loss를 줄이는 것
    - Weight quantization error = $\delta\mathbf{x} = \bar{\mathbf{x}} - \mathbf{x}\ \ (\mathbf{x} \in R^{D_{in}})$
    - Activation quantization error = $\delta\mathbf{W} = \bar{\mathbf{W}} - \mathbf{W}\ (\mathbf{W} \in R^{D_{out} \times D_{in}})$
  - 본 논문의 목표 설정 : $L^{MSE} = E\left[\left|\left|\mathbf{W}\mathbf{x} - \bar{\mathbf{W}}\bar{\mathbf{x}}\right|\right|_2^2\right] = E\left[\left|\left|\mathbf{W}\mathbf{x} - (\mathbf{W} + \delta\mathbf{W})(\mathbf{x} + \delta\mathbf{x})\right|\right|_2^2\right]$

- Method
  - $\delta\mathbf{W}$와 $\delta\mathbf{x}$의 영향을 동시에 고려하며 최적화하는 것은 어렵기 때문에 순차적인 최적화 방법을 제안함
    - Activation quantization → Error reduction (Aqer) → Weight quantization → Error reduction (Wqer)

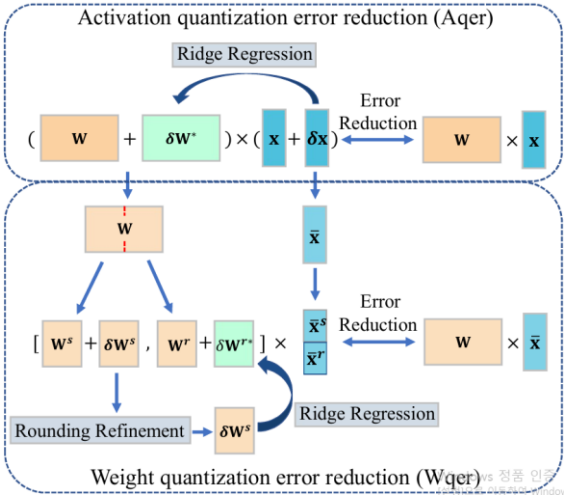


Fig 1. The overview of ERQ

ERQ[1)]

- Method
  - Activation quantization error reduction (Aqer)
    - Activation quantization만을 적용했을 때의 loss function : $L^{MSE} = E\left[\left|\left|\mathbf{W}\mathbf{x} - \mathbf{W}(\mathbf{x}+\delta\mathbf{x})\right|\right|_2^2\right]$
    - Activation quantization에 의한 발생한 error를 최소화하는 방향으로 weight update하기 위해 ridge regression 적용

$$E\left[\left|\left|\mathbf{W}\mathbf{x} - (\mathbf{W}+\delta\mathbf{W}^*)(\mathbf{x}+\delta\mathbf{x})\right|\right|_2^2\right] + \lambda_1\left|\left|\delta\mathbf{W}^*\right|\right|_2^2 = E\left[\left|\left|-\delta\mathbf{W}^*(\mathbf{x}+\delta\mathbf{x}) - \mathbf{W}\delta\mathbf{x}\right|\right|_2^2\right] + \lambda_1\left|\left|\delta\mathbf{W}^*\right|\right|_2^2$$
$$= E\left[\left|\left|\delta\mathbf{W}^*\bar{\mathbf{x}} + \mathbf{W}\delta\mathbf{x}\right|\right|_2^2\right] + \lambda_1\left|\left|\delta\mathbf{W}^*\right|\right|_2^2$$

$$\frac{\partial}{\partial\delta\mathbf{W}^*}\left\{E\left[\left|\left|\delta\mathbf{W}^*\bar{\mathbf{x}} + \mathbf{W}\delta\mathbf{x}\right|\right|_2^2\right] + \lambda_1\left|\left|\delta\mathbf{W}^*\right|\right|_2^2\right\} = E[2(\delta\mathbf{W}^*\bar{\mathbf{x}} + \mathbf{W}\delta\mathbf{x})\bar{\mathbf{x}}^T] + 2\lambda_1\delta\mathbf{W}^* = 0$$
$$\rightarrow \delta\mathbf{W}^* = -\mathbf{W}E[\bar{\mathbf{x}}\delta\mathbf{x}^T](E[\bar{\mathbf{x}}\bar{\mathbf{x}}^T] + \lambda_1\mathbf{I})^{-1}$$

    - Aqer : $\mathbf{W} \leftarrow \mathbf{W} + \delta\mathbf{W}^*$ (Error를 최소화하는 방향으로 weight update)

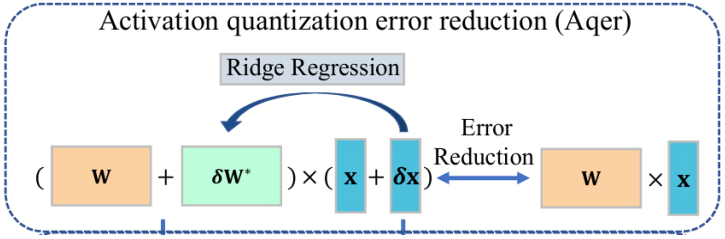


Fig 2. The overview of Aqer

ERQ[1)]

- Method
  - Weight quantization error reduction (Wqer)
    - Aqer의 적용으로 얻은 weight $\mathbf{W}$에 대하여 weight quantization 적용
    - Weight quantization 적용 시 loss function $L^{MSE} = E\left[\left|\left|\mathbf{W}\bar{\mathbf{x}} - (\mathbf{W}+\delta\mathbf{W})\bar{\mathbf{x}}\right|\right|_2^2\right] = \sum_i^{D_{out}} L_i^{MSE} = \sum_i^{D_{out}} E\left[\left|\left|\mathbf{W_{i,:}}\bar{\mathbf{x}} - (\mathbf{W_{i,:}}+\delta\mathbf{W_{i,:}})\bar{\mathbf{x}}\right|\right|_2^2\right]$
    - Step 1) Quantization을 weight의 절반에만 적용
      - $\mathbf{W} = [\mathbf{W^s}, \mathbf{W^r}]$ ($\mathbf{W^s}$: quantization 적용 대상, $\mathbf{W^r}$: remained full-precision weight) 에 대하여 아래와 같이 수식 전개

$$
\begin{aligned}
L_i^{MSE} &= E\left[\left|\left|[\mathbf{W}_{i,:}^s, \mathbf{W}_{i,:}^r][\bar{\mathbf{x}}^s, \bar{\mathbf{x}}^r] - [\mathbf{W}_{i,:}^s + \delta\mathbf{W}_{i,:}^s, \mathbf{W}_{i,:}^r][\bar{\mathbf{x}}^s, \bar{\mathbf{x}}^r]\right|\right|_2^2\right] \\
&= E\left[\left|\left|\delta\mathbf{W}_{i,:}^s\bar{\mathbf{x}}^s\right|\right|_2^2\right] = \left(E\left[\delta\mathbf{W}_{i,:}^s\bar{\mathbf{x}}^s\right]\right)^2 + Var\left[\delta\mathbf{W}_{i,:}^s\bar{\mathbf{x}}^s\right] \\
&= \delta\mathbf{W}_{i,:}^s\boldsymbol{\mu}^s\boldsymbol{\mu}^{sT}\left(\delta\mathbf{W}_{i,:}^s\right)^T + \delta\mathbf{W}_{i,:}^s\boldsymbol{\Sigma}^s\left(\delta\mathbf{W}_{i,:}^s\right)^T = \delta\mathbf{W}_{i,:}^s\left(\boldsymbol{\mu}^s\boldsymbol{\mu}^{sT} + \boldsymbol{\Sigma}^s\right)\left(\delta\mathbf{W}_{i,:}^s\right)^T
\end{aligned}
$$

$$
G_{\delta\mathbf{W}_{i,:}^s} = \frac{\partial L_i^{MSE}}{\partial\delta\mathbf{W}_{i,:}^s} = \frac{\partial}{\partial\delta\mathbf{W}_{i,:}^s}\left\{\delta\mathbf{W}_{i,:}^s\left(\boldsymbol{\mu}^s\boldsymbol{\mu}^{sT} + \boldsymbol{\Sigma}^s\right)\left(\delta\mathbf{W}_{i,:}^s\right)^T\right\} = 2\delta\mathbf{W}_{i,:}^s\left(\boldsymbol{\mu}^s\boldsymbol{\mu}^{sT} + \boldsymbol{\Sigma}^s\right)
$$

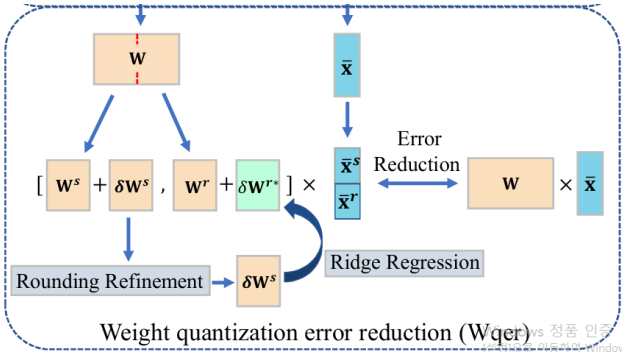


Fig 3. The overview of Wqer

ERQ[1)]

- Method
  - Weight quantization error reduction (Wqer)
    - Step 2) Loss function을 최소화하는 방향으로 rounding direction 설정 (Round refinement)

      $\delta W^s_{i,j} \cdot G_{\delta W^s_{i,j}} > 0$ 인 경우에 대해서만 round refinement 적용

$$\delta \mathbf{W}^s_{i,j} = \begin{cases} \delta \mathbf{W}^{s\downarrow}_{i,j} & (\delta \mathbf{W}^s_{i,j} = \delta \mathbf{W}^{s\uparrow}_{i,j}) \\ \delta \mathbf{W}^{s\uparrow}_{i,j} & (\delta \mathbf{W}^s_{i,j} = \delta \mathbf{W}^{s\downarrow}_{i,j}) \end{cases} \left(\delta \mathbf{W}^{s\downarrow}_{i,j} = \mathbf{W}^s_{i,j} - \mathbf{W}^{s\downarrow}_{i,j}, \delta \mathbf{W}^{s\uparrow}_{i,j} = \mathbf{W}^s_{i,j} - \mathbf{W}^{s\uparrow}_{i,j}\right)$$

    - Round refinement : $\bar{\mathbf{W}}^s_{i,:} \leftarrow \mathbf{W}^s_{i,:} + \delta \mathbf{W}^s_{i,:}$
    - Step 3) $\mathbf{W}^s$ quantization에 의해 발생한 error를 최소화하는 방향으로 $\mathbf{W}^r$를 update하기 위해 ridge regression 적용

$$E\left[\left|\left|\delta \mathbf{W}^s_{i,:}\bar{\mathbf{x}}^s + \delta \mathbf{W}^{r*}_{i,:}\bar{\mathbf{x}}^r\right|\right|_2^2\right] + \lambda_2 \left|\left|\delta \mathbf{W}^{r*}_{i,:}\right|\right|_2^2 \rightarrow \delta \mathbf{W}^{r*}_{i,:} = -\delta \mathbf{W}^s_{i,:} E\left[\bar{\mathbf{x}}^s \bar{\mathbf{x}}^{rT}\right]\left(E\left[\bar{\mathbf{x}}^r \bar{\mathbf{x}}^{rT}\right] + \lambda_2 \mathbf{I}\right)^{-1}$$

    - Wqer : $\mathbf{W}^r_{i,:} \leftarrow \mathbf{W}^r_{i,:} + \delta \mathbf{W}^{r*}_{i,:}$
    - Step 4) 모든 weight에 대해 적용될 때까지 혹은 특정 횟수만큼 step 1~3 반복

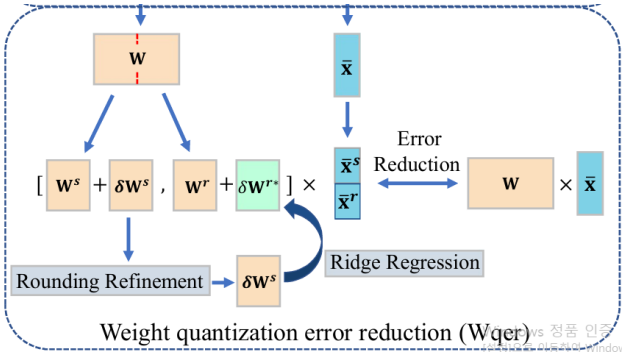


Fig 3. The overview of Wqer

ERQ[1)]

- Method
  - Weight quantization error reduction (Wqer)

Algorithm 1 Weight Quantization Error Reduction

Input: $\mathbf{W}$, $\bar{\mathbf{W}} = \varnothing$, $\{\bar{\mathbf{x}}_n\}_{n=1}^N$, maximum iteration T
for i in range(D_{out}): — Step 1
  $\mathbf{W}_{i,:} = \varnothing$, $\{\bar{\mathbf{x}}_n\}_{n=1}^N = \{\bar{\mathbf{x}}_n\}_{n=1}^N$
  while $|\mathbf{W}_{i,:}| > 0$
    Partition $\mathbf{W}_{i,:}$ into $\left[\mathbf{W}_{i,:}^s, \mathbf{W}_{i,:}^r\right]$
    Partition $\{\bar{\mathbf{x}}_n\}_{i=1}^N$ into $\left\{\left[\bar{\mathbf{x}}_n^s, \bar{\mathbf{x}}_n^r\right]\right\}_{i=1}^N$ — Step 2
    /* Rounding Refinement */
    Obtain $\hat{\boldsymbol{\mu}}^s\hat{\boldsymbol{\mu}}^{sT} + \hat{\boldsymbol{\Sigma}}^s$ from cache or calculate it with $\{\bar{\mathbf{x}}_n^s\}_{n=1}^N$, calculate $\delta\mathbf{W}_{i,:}^{s\downarrow}, \delta\mathbf{W}_{i,:}^{s\uparrow}$ with $\mathbf{W}_{i,:}^s$
    while $0 \leq$ T--:
      Calculate proxy $\mathcal{L}_{old}$ with $\delta\mathbf{W}_{i,:}^s$ by Eq. 12
      Calculate gradients $\boldsymbol{G}_{\delta\mathbf{W}_{i,:}^s}$ by Eq. 14
      Obtain $\mathcal{S}$ by Eq. 15
      Obtain adjusted $\delta\mathbf{W}_{i,:}^{'}$ by Eq. 13
      Calculate proxy $\mathcal{L}_{now}$ with $\delta\mathbf{W}_{i,:}^{'}$ by Eq. 12
      if $\mathcal{L}_{now} > \mathcal{L}_{old}$: break
      $\delta\mathbf{W}_{i,:}^s = \delta\mathbf{W}_{i,:}^{'}$
    $\bar{\mathbf{W}}_{i,:} \leftarrow \bar{\mathbf{W}}_{i,:} \cup (\mathbf{W}_{i,:}^s + \delta\mathbf{W}_{i,:}^s)$
    /* END Rounding Refinement */ — Step 3
    /* Ridge Regression */
    Calculate $\delta\mathbf{W}_{i,:}^{r*}$ by Eq. 17
    $\mathbf{W}_{i,:} \leftarrow \mathbf{W}_{i,:}^r + \delta\mathbf{W}_{i,:}^{r*}$
    /* END Ridge Regression */
    $\{\bar{\mathbf{x}}_n\}_{n=1}^N \leftarrow \{\bar{\mathbf{x}}_n^r\}_{n=1}^N$
  $\bar{\mathbf{W}} \leftarrow \bar{\mathbf{W}} \cup \bar{\mathbf{W}}_{i,:}$
Output: Quantized weight $\bar{\mathbf{W}}$

- Step 1) Quantization을 weight의 절반에만 적용
- Step 2) Loss function을 최소화하는 방향으로 rounding direction 설정 (Round refinement)
- Step 3) $\mathbf{W}^s$ quantization에 의해 발생한 error를 최소화하는 방향으로 $\mathbf{W}^r$를 update하기 위해 Ridge regression 적용
- Step 4) 모든 weight에 대해 적용될 때까지 혹은 특정 횟수만큼 step 1~3 반복

ERQ[1)]

- Experiments
  - Image Classification task

Method	W/A	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
Full-Precision	32/32	81.39	84.54	72.21	79.85	81.80	83.23	85.27
FQ-ViT (Lin et al., 2022)	6/6	4.26	0.10	58.66	45.51	64.63	66.50	52.09
PSAQ-ViT (Li et al., 2022b)	6/6	37.19	41.52	57.58	63.61	67.95	72.86	76.44
Ranking (Liu et al., 2021b)	6/6	-	75.26	-	74.58	77.02	-	-
PTQ4ViT (Yuan et al., 2022)	6/6	78.63	81.65	69.68	76.28	80.25	82.38	84.01
APQ-ViT (Ding et al., 2022)	6/6	79.10	82.21	70.49	77.76	80.42	82.67	84.18
NoisyQuant† (Liu et al., 2023b)	6/6	76.86	81.90	-	76.37	79.77	82.78	84.57
NoisyQuant‡ (Liu et al., 2023b)	6/6	78.65	82.32	-	77.43	80.70	82.86	84.68
GPTQ* (Frantar et al., 2022)	6/6	80.44	83.72	71.05	78.95	81.37	82.82	84.89
RepQ-ViT (Li et al., 2023)	6/6	80.43	83.62	70.76	78.90	81.27	82.79	84.57
EasyQuant (Wu et al., 2020)	6/6	75.13	81.42	-	75.27	79.47	82.45	84.30
Bit-shrinking (Lin et al., 2023)	6/6	80.44	83.16	-	78.51	80.47	82.44	-
BRECQ* (Li et al., 2021)	6/6	61.18	71.29	69.62	70.93	79.46	81.85	84.08
QDrop* (Wei et al., 2022)	6/6	68.57	74.38	69.98	76.57	80.66	82.53	84.31
PD-Quant* (Liu et al., 2023a)	6/6	71.38	63.14	70.74	77.63	79.32	82.33	84.38
ERQ (Ours)	6/6	**80.48**	**83.89**	**71.14**	**79.03**	**81.41**	**82.86**	**85.02**

Table 1. 6bit 환경에서의 실험 (ImageNet)

Method	Runtime	Top-1 Acc. (%)
BRECQ* (Li et al., 2021)	~48 minutes	32.89
QDrop* (Wei et al., 2022)	~80 minutes	35.79
PD-Quant* (Liu et al., 2023a)	~110 minutes	64.85
GPTQ* (Frantar et al., 2022)	~3 minutes	70.85
RepQ-ViT (Li et al., 2023)	~1 minute	69.03
ERQ (Ours)	~4 minutes	**72.56**

Table 3. 4bit 환경에서의 성능 비교 (ImageNet, W4/A4 DeiT-S)

Method	W/A	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
Full-Precision	32/32	81.39	84.54	72.21	79.85	81.80	83.23	85.27
FQ-ViT* (Lin et al., 2022)	3/4	0.10	0.10	0.10	0.10	0.10	0.10	0.10
PTQ4ViT* (Yuan et al., 2022)	3/4	0.10	0.10	0.20	0.15	0.59	0.64	0.53
GPTQ* (Frantar et al., 2022)	3/4	23.32	44.63	42.25	48.95	61.75	66.71	71.43
RepQ-ViT* (Li et al., 2023)	3/4	15.65	26.98	29.34	45.82	58.92	59.83	44.17
AdaRound* (Nagel et al., 2020b)	3/4	11.04	4.72	36.05	33.56	62.50	68.12	53.92
BRECQ* (Li et al., 2021)	3/4	4.97	1.25	29.23	18.58	40.49	66.93	53.38
QDrop* (Wei et al., 2022)	3/4	9.77	11.87	17.85	30.27	61.12	73.47	74.33
PD-Quant* (Liu et al., 2023a)	3/4	4.56	21.81	41.87	41.65	53.63	70.07	56.48
ERQ (Ours)	3/4	**45.68**	**53.88**	**44.09**	**57.63**	**70.33**	**75.08**	**75.78**
FQ-ViT (Lin et al., 2022)	4/4	0.10	0.10	0.10	0.10	0.10	0.10	0.10
PTQ4ViT (Yuan et al., 2022)	4/4	42.57	30.69	36.96	34.08	64.39	76.09	74.02
APQ-ViT (Ding et al., 2022)	4/4	47.95	41.41	47.94	43.55	67.48	77.15	76.48
GPTQ* (Frantar et al., 2022)	4/4	67.59	75.12	58.96	70.85	76.10	80.17	81.08
RepQ-ViT (Li et al., 2023)	4/4	65.05	68.48	57.43	69.03	75.61	79.45	78.32
AdaRound* (Nagel et al., 2020b)	4/4	63.09	70.51	55.65	69.24	75.20	76.05	78.12
BRECQ* (Li et al., 2021)	4/4	11.31	3.03	38.41	32.89	59.10	68.40	56.51
QDrop* (Wei et al., 2022)	4/4	17.77	21.72	31.65	35.79	65.47	78.92	80.49
PD-Quant* (Liu et al., 2023a)	4/4	32.64	34.86	58.50	64.85	60.06	77.04	75.84
ERQ (Ours)	4/4	**68.91**	**76.63**	**60.29**	**72.56**	**78.23**	**80.74**	**82.44**
FQ-ViT* (Lin et al., 2022)	5/5	0.10	0.10	0.10	0.10	0.10	0.10	0.10
PTQ4ViT* (Yuan et al., 2022)	5/5	72.74	72.32	65.00	70.26	72.65	80.90	81.87
GPTQ* (Frantar et al., 2022)	5/5	78.63	82.06	69.05	77.12	80.17	82.19	83.00
RepQ-ViT* (Li et al., 2023)	5/5	78.43	82.03	69.00	77.04	80.08	82.08	83.22
AdaRound* (Nagel et al., 2020b)	5/5	77.53	82.00	68.87	76.22	80.18	82.12	84.09
BRECQ* (Li et al., 2021)	5/5	47.35	43.51	62.12	63.15	75.61	80.66	82.31
QDrop* (Wei et al., 2022)	5/5	56.32	57.92	62.36	70.07	78.41	81.73	83.61
PD-Quant* (Liu et al., 2023a)	5/5	65.06	58.40	68.02	74.94	74.61	81.27	82.12
ERQ (Ours)	5/5	**78.83**	**82.81**	**69.42**	**77.58**	**80.65**	**82.44**	**84.50**

Table 2. 3~5bit 환경에서의 실험 (ImageNet)

ERQ[1)]

- Experiments
  - Object detection & Segmentation task
    - Table 4 : ViT 기반 PTQ 방법론들 중 대부분 최고 성능을 달성

Method	W/A	Mask R-CNN				Cascade Mask R-CNN			
		w. Swin-T		w. Swin-S		w. Swin-T		w. Swin-S	
		AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
Full-Precision	32/32	46.0	41.6	48.5	43.3	50.4	43.7	51.9	45.0
PTQ4ViT (Yuan et al., 2022)	4/4	6.9	7.0	26.7	26.6	14.7	13.5	0.5	0.5
APQ-ViT (Ding et al., 2022)	4/4	23.7	22.6	**44.7**	40.1	27.2	24.4	47.7	41.1
GPTQ* (Frantar et al., 2022)	4/4	36.3	36.3	42.9	40.2	47.1	41.5	49.2	43.2
RepQ-ViT (Li et al., 2023)	4/4	36.1	36.0	$44.2_{42.7}$†	$40.2_{40.1}$†	47.0	41.4	49.3	43.1
AdaRound* (Nagel et al., 2020a)	4/4	16.3	19.8	22.3	22.5	34.6	33.4	35.8	34.5
BRECQ* (Li et al., 2021)	4/4	25.2	27.3	32.4	32.9	40.4	35.9	41.5	37.2
QDrop* (Wei et al., 2022)	4/4	10.4	11.3	39.7	37.8	17.9	16.2	20.1	17.4
PD-Quant* (Liu et al., 2023a)	4/4	15.7	16.1	30.2	28.4	34.5	30.1	38.6	34.1
ERQ (Ours)	4/4	**36.8**	**36.6**	43.4	**40.7**	**47.9**	**42.1**	**50.0**	**43.6**

Table 4. 4bits 환경에서의 실험 (COCO, R-CNN)

- Table 5 : Weight quantization error에 강건한 모습을 확인 (detection task에서 activation의 bit-width에 영향을 많이 받음)

Model	Method	W/A	AP(box)	AP(mask)
Mask R-CNN (Swin-T)	Full-Precision	32/32	46.0	41.6
	ERQ	4/4	36.8	36.6
	ERQ	4/8	41.0	39.2
Mask R-CNN (Swin-S)	Full-Precision	32/32	48.5	43.3
	ERQ	4/4	43.4	40.7
	ERQ	4/8	46.1	42.2
Cascade R-CNN (Swin-T)	Full-Precision	32/32	50.4	43.7
	ERQ	4/4	47.9	42.2
	ERQ	4/8	49.5	43.3
Cascade R-CNN (Swin-S)	Full-Precision	32/32	51.9	45.0
	ERQ	4/4	50.0	43.6
	ERQ	4/8	51.3	44.5

Table 5. 4/4bits, 4/8bits 환경에서의 실험 (COCO)

ERQ[1)]

- Experiments
  - Ablation studies

Aqer	Wqer		Top-1 Acc. (%)
	Rounding	Ridge	
Baseline			68.41
✓			71.45 (+3.04)
	✓		69.24 (+0.83)
		✓	70.06 (+1.65)
	✓	✓	70.49 (+2.08)
✓	✓		71.83 (+3.42)
✓		✓	72.01 (+3.60)
✓	✓	✓	**72.56 (+4.15)**

Table 6. Components of ERQ (32 images)

Model	Image Numbers	Top-1 Acc. (%)
DeiT-S (W4/A4)	4	71.58
	8	71.87
	16	72.54
	32	72.56
	64	72.94
	128	73.19
	256	73.51
	512	73.68
	1024	73.69

Table 7. Performance of different image numbers

Model	Latency (ms)	Throughput (img/s)
ViT-S	184 104 (1.77x)	5.43 9.62 (1.77x)
ViT-B	746 443 (1.68x)	1.34 2.26 (1.68x)
DeiT-T	54 31 (1.74x)	18.51 32.26 (1.74x)
DeiT-S	163 106 (1.54x)	6.13 9.43 (1.54x)
DeiT-B	745 376 (1.98x)	1.34 2.66 (1.98x)
Swin-S	337 217 (1.55x)	2.97 4.61 (1.55x)
Swin-B	683 461 (1.48x)	1.46 2.17 (1.48x)

Table 8. Latency and throughput of W8A8 ViTs

- Table 6 : Aqer, Wqer 모두 유의미한 성능 향상을 보이며, 모두 적용되었을 때 baseline 대비 4.15%의 정확도 향상
- Table 7 : Calibration data의 개수에 증가함에 따라 유의미한 성능 향상을 확인
- Table 8 : 224×224 image를 입력하여 평가한 결과 일반적으로 1.5배~2배의 속도 향상을 확인
  - ONNX를 사용하여 Intel i5-10210U CPU 환경에서 수행

Conclusion

- NoisyQuant[1)]
  - Key contribution
    - Power-law distribution에 noise를 첨가하여 quantizer에 맞게 변형하는 방법론을 제안하였음
  - Limitations
    - Noisy bias의 형태가 uniform distribution이어야만 한다는 근거가 부족함
      - Gaussian distribution은?
    - 최적의 noisy bias parameter를 실험을 통해 직접 찾아야 함
      - Hyper parameter searching에 시간이 많이 소요됨
    - 5bits 이하의 실험이 존재하지 않음

- ERQ[2)]
  - Key contribution
    - Activation/weight quantization에 의해 발생하는 loss를 최적화하는 two-step PTQ 방법론을 제안하였음
  - Limitations
    - Wqer에서의 매 iteration마다 quantized weight element에 적용되는 rounding refinement에 의해 gradient의 분포가 변형될 수 있음을 무시하고 있음

감사합니다