Quantization research from WACV 2024

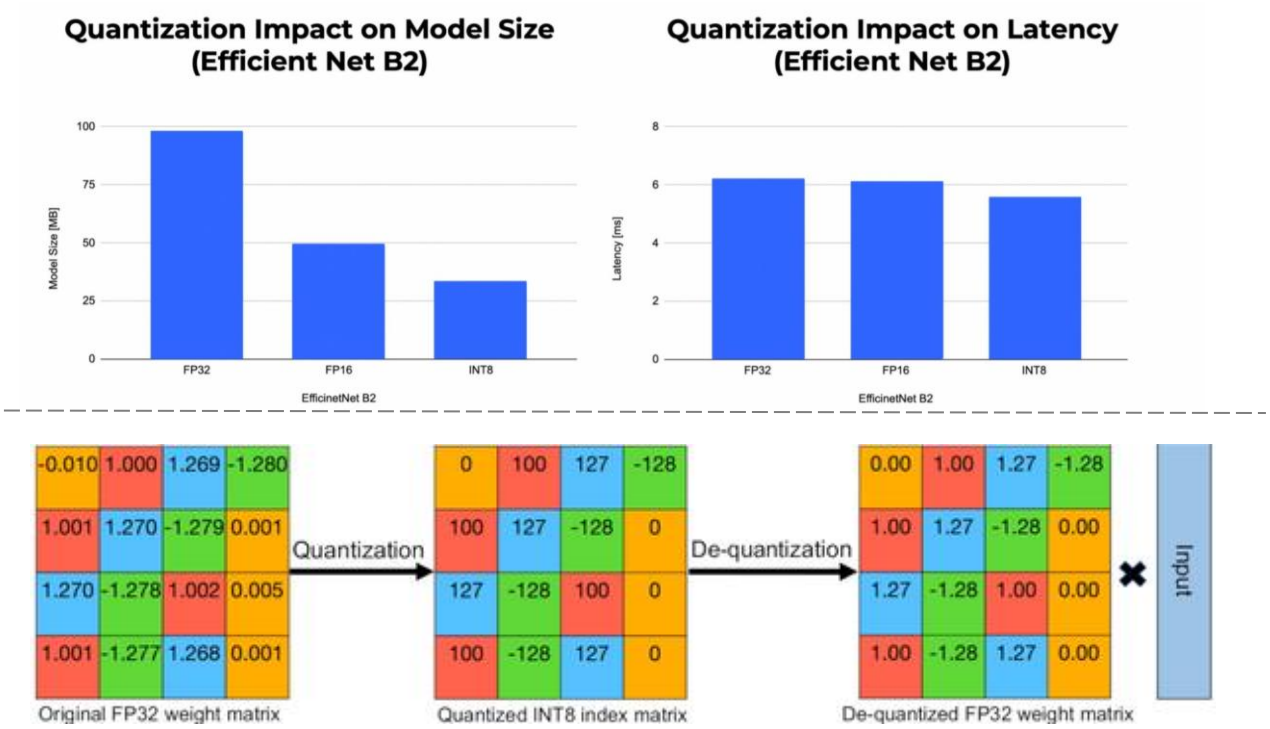
2024 winter seminar

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

Presented By
Beoungwoo Kang

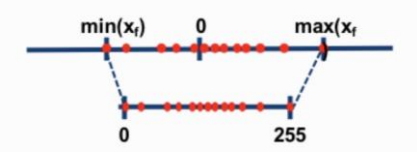
Background

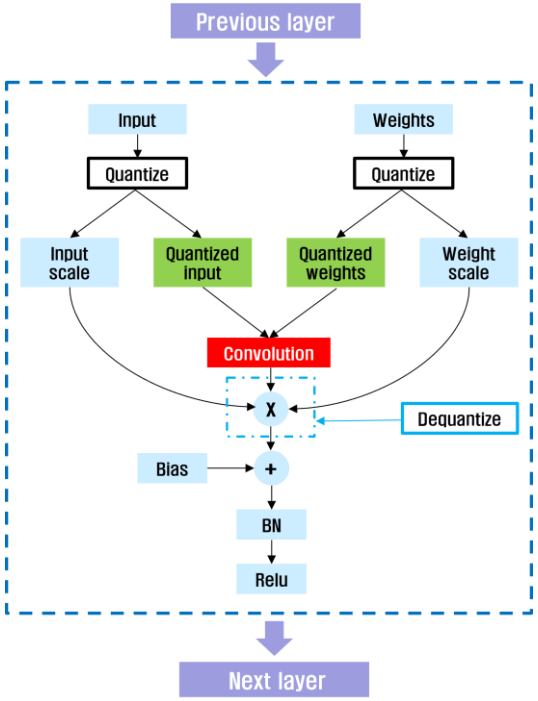
- Quantization
  - Neural network의 weight 및 activation을 lower bit-width의 integer 자료형으로 변환
    - Floating point (FP)의 matrix 곱 연산에 비해 빠른 연산 속도와 낮은 bandwidth를 가짐
    - 필연적으로 발생하는 information loss로 인해 낮은 precision 결과
    - Information loss를 최소화 하는 다양한 quantization 기법 연구 진행중



Background

- Basic quantization techniques
  - Post training quantization (PTQ)
    - 학습이 완료된 neural network에 random한 input을 가하여 statics를 수집
    - Max calibration
      - Activation에서 가장 큰 값까지 quantization
      - Outlier에 취약함
    - Percentile calibration
      - Tensor의 histogram에서 일정 부분만 quantization
      - 주로 99.99% 를 사용함
    - Entropy calibration
      - KL divergence 기반 calibration
      - Information loss가 제일 적은 지점 까지 quantization

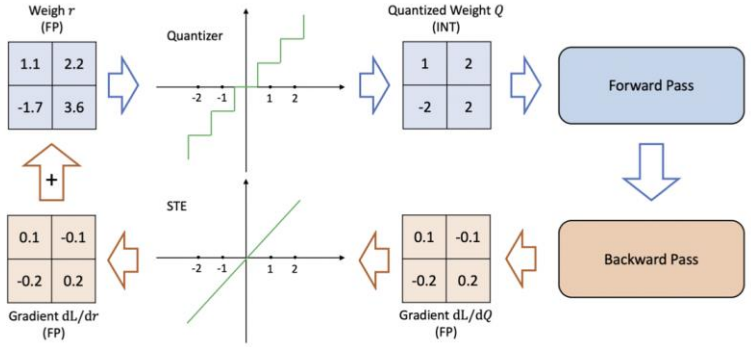


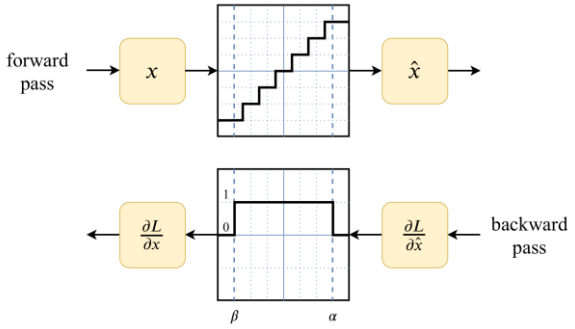


< Post training quantization >

Background

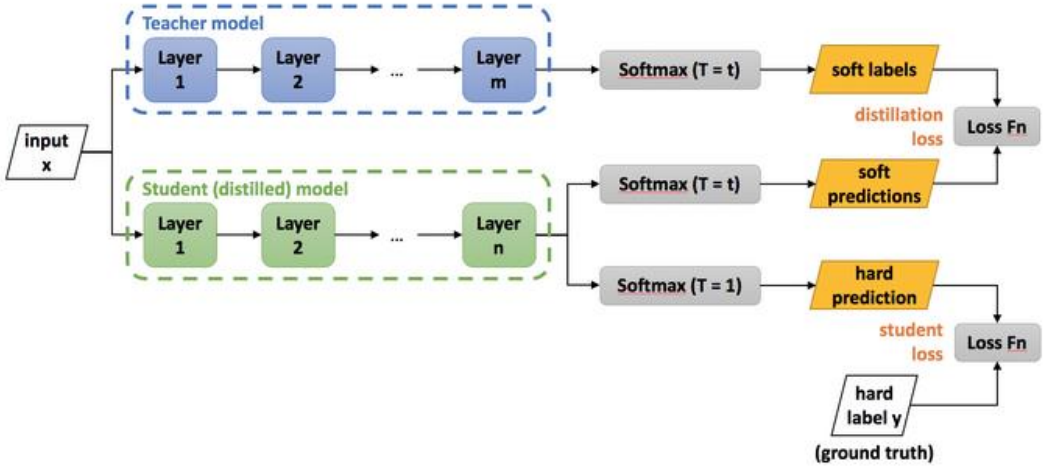
- Basic quantization techniques
  - Quantization aware training (QAT)
    - 학습과정에서 pseudo quantization을 적용 후 최적의 quantized parameter를 학습
      - Quantization 한 weight와 activation을 이용하여 forward pass
      - 타 딥러닝 모델과 동일하게 loss 계산 후 backward pass 진행
      - 일반적으로 STE를 이용하여 backpropagation 진행
    - Straight-Through Estimator (STE)
      - Neural network의 quantization operation의 경우 derivative가 0인 경우 존재
      - Backward pass에선 학습의 전이를 위해 STE operation을 사용

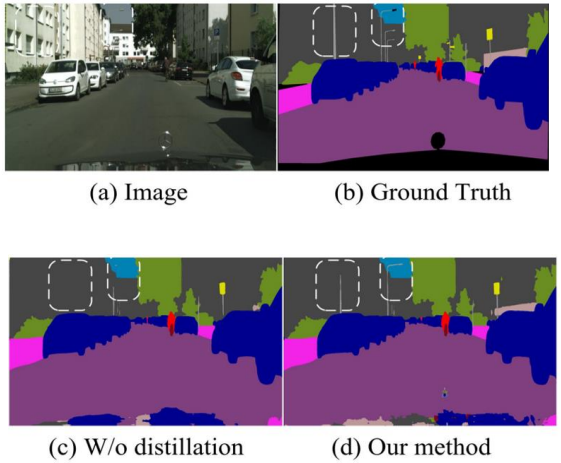




Background

- Basic quantization techniques
  - Knowledge distillation (KD)
    - 큰 네트워크 (teacher network)의 지식을 작은 네트워크 (student network) 에게 전달
      - Teacher model의 prediction을 soft label로 student model prediction간의 loss 설정
      - Student model은 hard label (실제 GT)와의 loss를 추가하여 학습
      - Soft label, hard label에 의한 loss 합이 낮아지도록 학습 진행





“ Edge Inference With Fully Differentiable Quantized Mixed Precision Neural Networks”, WACV 2024

Motivation

- Quantization aware training (QAT)
  - Ineffective QAT model when lower parameter size
    - Lower bit model 일수록 기존 FP32 model의 linear 한 분포 대비 더욱 step화
      - STE와 같은 projection 기법 backward pass 진행 시 비 효율적
    - Pre-trained weight를 이용하여 QAT를 진행하므로 2 step training이 되어 비 효율적
    - EfficientNet-Lite0, MobileNetV2와 같은 residual network에서 현재 SOTA model의 bit width distribution은 비 효율적
  - Contribution
    - QAT training strategy 제안
      - FP32 model pretrain 후 penalty를 이용하여 1 step training에 QAT training
    - STE가 아닌 최적의 gradient scaling 조합 제안
    - EfficientNet-Lite0와 MobileNetV2에서 가장 효율적인 bit width 제안

Method

- Quantization Training Dynamics
  - Three phase of mixed precision training method
    - 기존 대다수의 QAT training strategy는 pre-trained weight를 불러와서 QAT 진행
      - 2 step training 진행으로 시간 및 리소스 관점에서 비 효율적
    - Pre-training step 이후 size penalty를 linear하게 증가하여 점진적 bit width 학습 진행

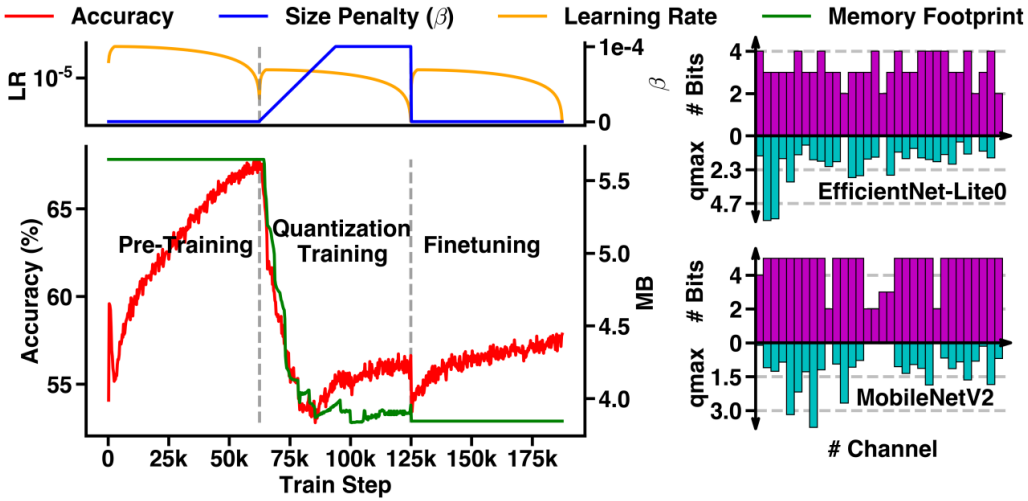


Figure 3. On the left illustrating the three phases of our mixed precision training method. The top left shows scheduled learning rate and size penalty (β) term meanwhile the bottom left shows the evolution of model accuracy and model size. On the right we show an example per-channel bit allocation in a weight kernel of an EfficientNet-Lite0 and MobileNetV2. Extreme quantization is correlated to low dynamic range (q_+). However, the contra does not necessarily hold.

Method

- Gradient scaling
  - Inverse Hyperbolic tangent based gradient scaling method
    - Lower bit model 일수록 step화 된 weight 분포를 보임
    - 일반적인 STE 대신 Arctanh 기반 gradient scaling method가 높은 성능을 보임

1. Position based gradient scaling (PBGS) [27]:
$$\text{scale} = 1 + \delta \cdot |x - \text{round}(x)|.$$

2. Element-wise gradient scaling (EWGS) [28]:
$$\text{scale} = 1 + \delta \cdot \text{sign}(g_x) \cdot (x - \text{round}(x)).$$

3. Modified absolute cosine (Acos) [31] gradient scaling:
$$\text{scale} = 1 + \delta \cdot \sin(\pi \cdot (x - \text{round}(x))).$$

4. Hyperbolic tangent (Tanh) gradient scaling:
$$\text{scale} = 1 + \delta \cdot \text{sign}(g_x) \cdot \tanh(\alpha \cdot (x - \text{round}(x))).$$

5. Inverse hyperbolic tangent (InvTanh) gradient scaling:
$$\text{scale} = 1 + \delta \cdot \text{sign}(g_x) \cdot \text{arctanh}(\alpha \cdot (x - \text{round}(x))).$$

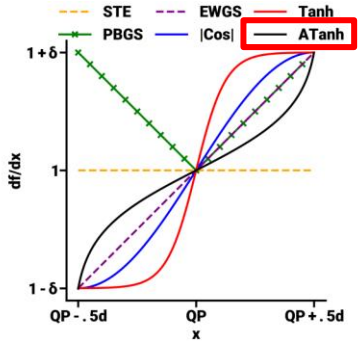


Figure 2. We illustrate different gradient scaling functions: straight-through-estimators (STE [2]), elementwise gradient scaling (EWGS [28]), position based gradient scaling (PBGS [27]), absolute cosine regularization (Acos [31]) as well as hyperbolic tangent function (Tanh) and its inverse (InvTanh). Note that QP denotes quantization point, d is the step size and δ is the magnitude control hyper parameter for gradient scaling.

Method

- Optimized bit width for residual network
  - Efficient bit width for EfficientNET-Lite0 and MobileNetV2
    - Early layer의 weight bit width를 크게 quantization 하는 것이 효율적
      - Residual connection 으로 인해, layer가 깊어져도 information이 잘 보존 됨
      - 다른 residual network에도 동일 경향성 존재

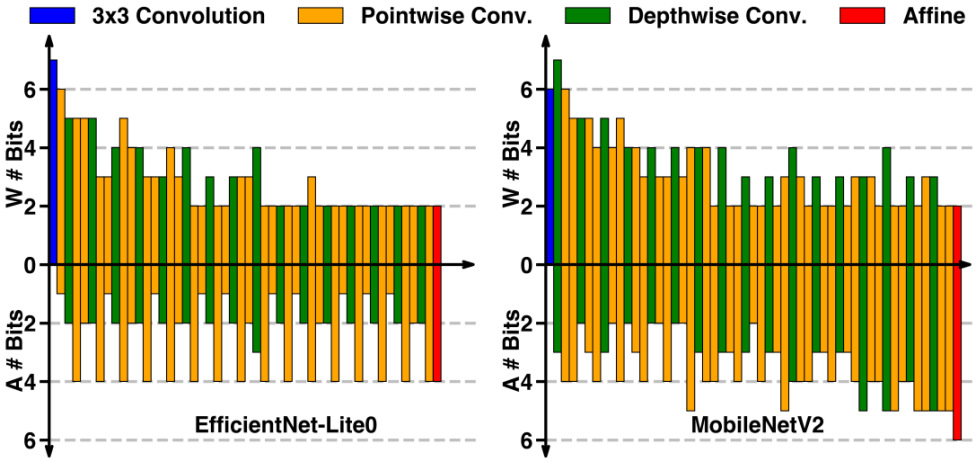


Figure 5. Internal bit allocation across layers of weights (up) and activations (down) for EfficientNet-Lite0 and MobileNetV2. Weights in the first layers have higher bit-widths for both models. Activations bitwidths for EfficientNet-Lite0 form a high precision path, e.g. activations which are residuals have higher precision. For both models the last affine layer has high precision.

Experiments

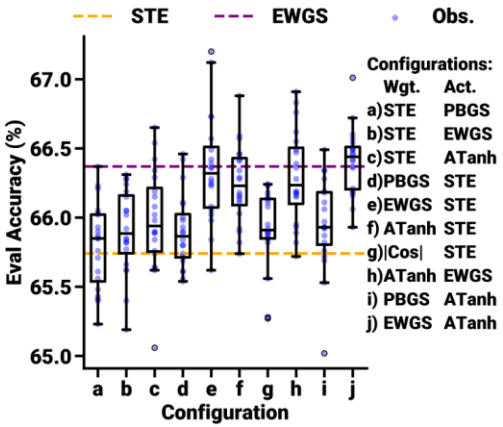
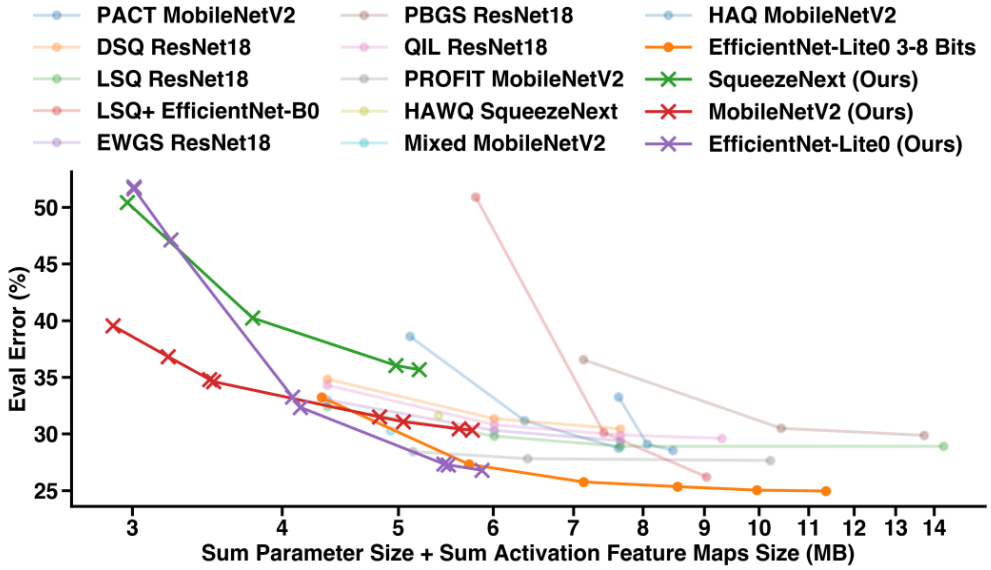


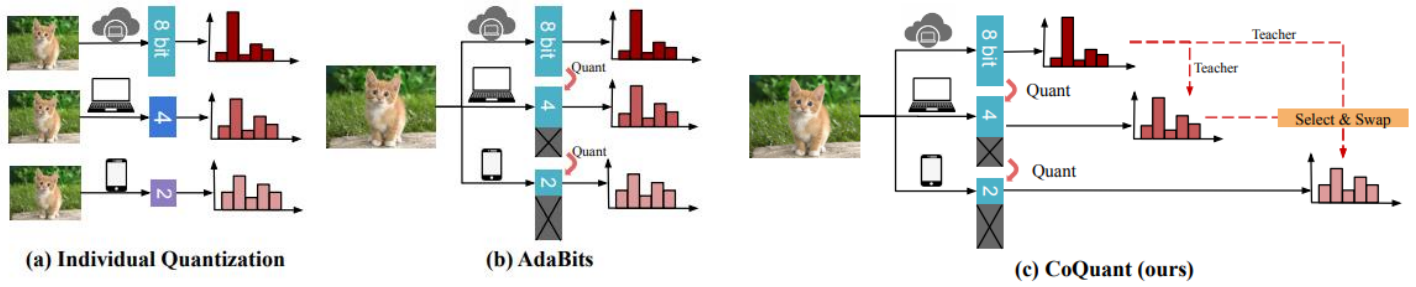
Figure 4. The performance of different mixed gradient scaling functions (different gradient scaling for weights and activations) on a homogeneous 3 bit EfficientNet-Lite0 on ImageNet.



“ Improved Techniques for Quantizing Deep Networks With Adaptive Bit-Widths ”, WACV 2024

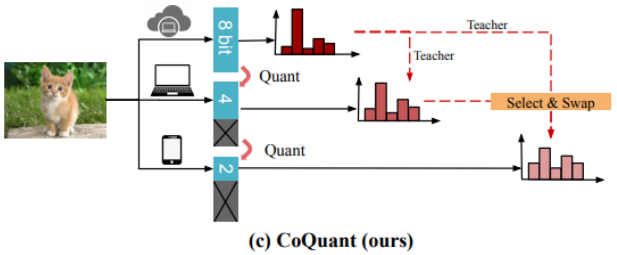
Motivation

- Quantization aware training with Knowledge distillation
  - Inefficient of individual quantization
    - Low bit model을 만들기 위해 FP32 model에서 individual quantization은 비 효율적
      - ∴ 매번 특정 bit width에 맞게 quantization 해야함
    - Knowledge distillation을 도입하여 1 step으로 여러 bit size model을 QAT 할 수 있으나, low bit model은 high precision model을 mimic 하기 힘듦
  - Contribution
    - Knowledge distillation과 결합한 QAT 방법론을 제안
      - ∴ 선택적으로 best teacher model을 판단 후 mimic
    - Best teacher model에 대해 student의 parameter block과 직접 swap하는 방법론 제안
    - 8, 6, 4, 2bit model의 성능 감소폭이 1 step training 기법들 중 가장 적음



(a) Individual Quantization (b) AdaBits (c) CoQuant (ours)

Method



(c) CoQuant (ours)

- Dynamic Teacher Selection and Swapping
  - Teacher Selection
    - 여러 bit width를 갖는 pre-trained model들을 학습 시작 시 이용
      - Lower bit model은 pseudo quantization 적용 후 학습 시작
    - Student model의 confidence를 entropy term $H(y_i)$ 을 이용하여 판단
      - Student model의 entropy가 높은 경우 해당 parameter block 교체 필요
    - Hyper parameter λ는 linearly increase
      - 학습 초기에 low bit model의 성능 수렴 목표

$$\underset{i \in \{b_1, b_2, \cdots, b_{k-1}\}}{\arg\min} \quad H(y_i) + \lambda ||\mathcal{M}_i - \mathcal{M}_{b_k}||$$

각 bit width model의 logit 값에 대한 entropy : Entropy가 높은 경우 confidence 낮아서 swap 필요

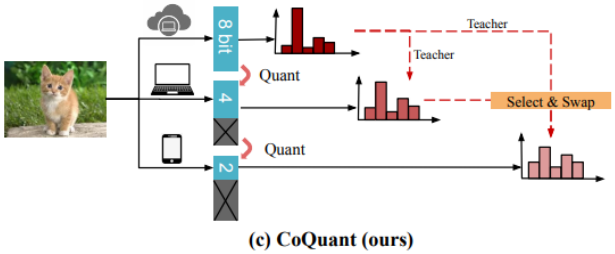
Linearly increase

각 bit width를 갖는 model간의 L1 distance

$$||\mathcal{M}_{b_i} - \mathcal{M}_{b_j}|| = \sum_{l=1}^{L} D(\widehat{W}_{b_i}^l, \widehat{W}_{b_j}^l)$$

Method

- Dynamic Teacher Selection and Swapping
  - Block swapping
    - Teacher selection 의 $\underset{i\in\{b_1,b_2,\cdots,b_{k-1}\}}{\arg\min} H(y_i)+\boxed{\lambda||\mathcal{M}_i-\mathcal{M}_{b_k}||}$ term을 줄이기 위해 swapping
      - L1 distance가 큰 teacher model block으로 swapping
    - $\beta_l = \text{Bernoulli}(p_l)$ 의 p_l factor를 linearly increase
      - 학습 시 early layer는 gradient 영향력이 낮기에 teacher parameter block 주되게 이용
      - $\beta_l = 0$: teacher network swapped / $\beta_l = 1$: maintain student network



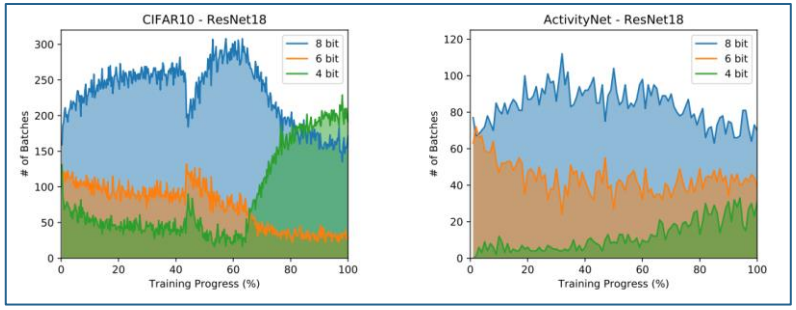
(c) CoQuant (ours)

$$A^{l+1} = \beta_l f(\widehat{A}_s^l, \widehat{W}_s^l) + (1-\beta_l) f(\widehat{A}_t^l, \widehat{W}_t^l)$$

$\beta_l = \text{Bernoulli}(p_l)$

Linearly increase
: Early layer는 gradient 영향력이 적기 때문에 teacher의 parameter를 주되게 사용

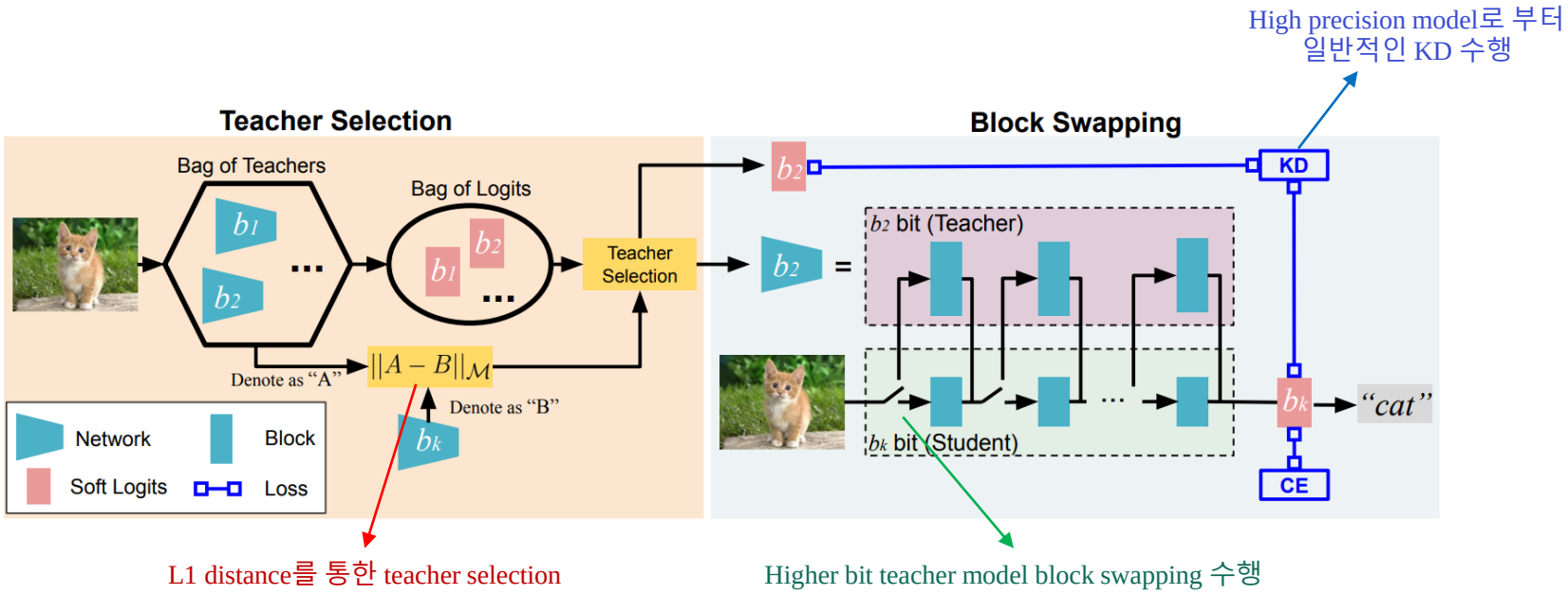
Activation, Weight
s : student, t : teacher



학습 초기에 high precision model, 학습 후기에 low precision model이 swapping

Method

- Method overview



Experiment

Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_{\mathcal{B}} \uparrow$
Individual Quant.	95.1	95.4	95.0	94.1	100
DQ (32 bit)	10.7	10.4	10.2	10.8	11.1
DQ (8 bit)	95.1	93.8	29.9	8.3	59.7
DQ (32 bit) + BN Calib	95.1	94.9	93.0	9.6	76.9
DQ (8 bit) + BN Calib	95.1	94.3	93.1	31.0	82.5
Joint Training	34.3	44.7	56.0	26.7	42.6
Switchable BN	94.6	94.5	94.5	92.3	99.2
AdaBits	94.4	94.2	94.2	92.4	99.0
Any Precision	94.2	94.2	94.2	92.3	98.8
Bit-Mixer	94.7	94.6	94.5	91.9	99.0
CoQuant (Ours)	95.2	95.4	95.1	94.1	**100.1**

Table 1: **ResNet18 on CIFAR10.** **CoQuant** achieves best overall performance $\Delta_{\mathcal{B}}$ (higher is better) among all baselines.

Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_{\mathcal{B}} \uparrow$
Individual Quant.	94.2	93.8	93.6	89.0	100
DQ (32 bit)	10.0	. 9.9	10.0	10.1	10.8
DQ (8 bit)	94.2	93.0	80.3	10.9	74.3
DQ (32 bit) + BN Calib	91.4	91.3	89.8	38.2	83.3
DQ (8 bit) + BN Calib	94.2	94.1	93.7	77.6	96.9
Joint Training	14.2	15.3	29.2	46.1	28.6
Switchable BN	94.2	94.0	93.3	85.4	99.0
AdaBits	93.9	93.8	93.2	86.2	99.3
Any Precision	93.4	93.3	93.1	86.3	98.8
Bit-Mixer	94.1	94.0	92.9	86.7	99.3
CoQuant (Ours)	94.1	94.2	94.0	87.5	**99.8**

Table 2: **MobileNet V2 on CIFAR10.** **CoQuant** achieves best overall performance $\Delta_{\mathcal{B}}$ among all compared methods.

Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_{\mathcal{B}} \uparrow$
Individual Quant.	69.1	68.8	68.1	60.1	100
DQ (32 bit)	0.1	0.1	0.1	0.1	0.2
DQ (8 bit)	69.1	13.9	0.1	0.1	30.1
DQ (32 bit) + BN Calib	69.3	68.8	52.3	0.2	69.3
DQ (8 bit) + BN Calib	69.1	68.1	39.3	0.1	65.1
Joint Training	8.4	11.4	33.3	1.5	20.0
Switchable BN	67.9	67.7	66.5	54.0	96.0
AdaBits	67.9	67.7	66.5	54.1	96.1
Any Precision	67.4	67.3	66.7	54.0	95.8
Bit-Mixer	67.1	67.1	66.5	54.7	95.8
CoQuant (Ours)	67.9	67.6	66.6	57.1	**97.3**

Table 3: **ResNet18 on ImageNet.** **CoQuant** achieves the best overall performance $\Delta_{\mathcal{B}}$ among all compared methods.

Methods	8-bit	6-bit	4-bit	2-bit	$\Delta_{\mathcal{B}} \uparrow$
Individual Quant.	66.2	66.5	66.6	62.4	100
DQ (32 bit)	0.4	0.5	0.5	0.5	0.7
DQ (8 bit)	66.2	62.1	1.9	0.5	49.3
DQ (32 bit) + BN Calib	65.9	65.8	54.4	0.5	70.3
DQ (8 bit) + BN Calib	66.2	65.4	54.5	0.6	70.3
Joint Training	0.5	0.5	0.5	0.7	0.9
Switchable BN	64.7	64.8	63.4	43.9	90.2
AdaBits	64.1	64.3	64.2	48.3	91.8
Any Precision	64.6	63.2	61.7	36.2	85.9
Bit-Mixer	65.8	66.1	64.7	53.4	**95.5**
CoQuant (Ours)	64.6	64.8	64.4	55.5	95.2

Table 4: **ResNet18 on Mini-Kinetics.** **CoQuant** achieves the second best overall performance $\Delta_{\mathcal{B}}$ among all baselines.

$\Delta_{\mathcal{B}}$: Individual quantization 대비 8, 6, 4, 2 bit model의 performance의 평균

감사합니다