Data-Free Quantization Through Weight Equalization and Bias Correction

2020-01-09 이승희

Introduction

- Motivation
  - Matrix multiplication이나 convolution operation들은 integer로 계산했을 때 더 빠르고, 전력 효율이 좋음 → **Deep learning service에서의 inference는 quantization이 필수적**
  - 하지만 FP32->INT8로 변환했을 때,
    - Quantization noise 때문에 정도에 따라 다르지만 **일반적으로 성능이 하락**함
    - 위의 성능저하를 줄이기 위해 가장 많이 쓰이는 방법은 quantization 후에 fine-tuning을 하는 것. 하지만 inference가 이루어지는 각 하드웨어에 맞는 방식으로 fine-tuning이 이루어져야 하는데, 이 fine-tuning이 cloud service나 edge device에서의 서비스에서는 큰 장애물임

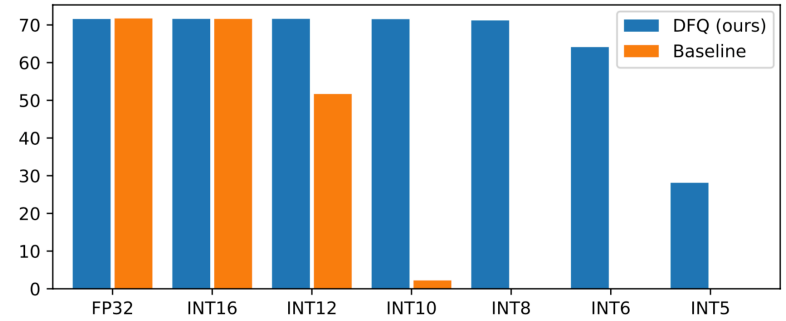

Figure 1. Fixed point inference for MobileNetV2 on ImageNet. The original model has significant drop in performance at 12-bit quantization whereas our model maintains close to FP32 performance even at 6-bit quantization.

Introduction

- Data-Free Quantization
  - Does not require data, fine-tuning, or hyperparameter tuning
  - Near-original model performance at FP32→INT8
  - Can be used as pre-processing for quantization-aware fine-tuning

Introduction

- Level of Prior Quantization Solutions
  - Level 1: 데이터나 backpropagation이 필요치 않음. 모델 정의와 웨이트 값만 있으면 변환이 가능하며 모든 모델에 적용 가능.
  - Level 2: 데이터는 필요하지만, backpropagation은 필요 없음. 데이터는 batch normalization를 re-calibrate하거나 layer-wise loss function을 계산해서 quantization performance를 높이기 위해 필요. 모든 모델에 적용 가능.
  - Level 3: 데이터와 backpropagation이 모두 필요함. Quantization 후에 fine-tuning이 반드시 필요하며, 종종 hyperparameter tuning까지도 필요함. 모든 모델에 적용 가능하지만, 아예 별도의 학습 파이프라인이 필요함.
  - Level 4: 데이터와 backpropagation이 모두 필요함. Quantization을 위해 네트워크 구조를 바꾸거나, 아니면 아예 처음에 학습할 때 부터 quantization을 염두에 둔 구조로 설계. 당연히 학습이 추가적으로 필요하며, 특정 모델에만 적용할 수 있음.

Proposed Method

- 착안점 1: Weight tensor channel ranges
  - MobileNetV2의 경우, per-tensor quantization보다 per-channel quantization에서 성능이 좋았으며, ImageNet validation set top-1 accuracy 기준 단 0.1%의 하락만이 있었음.
  - 실제로 어떤 레이어들에서는 weight distribution이 매우 다양함. 따라서 동일한 parameter로 tensor 전체를 quantize하는 것은 부적절.

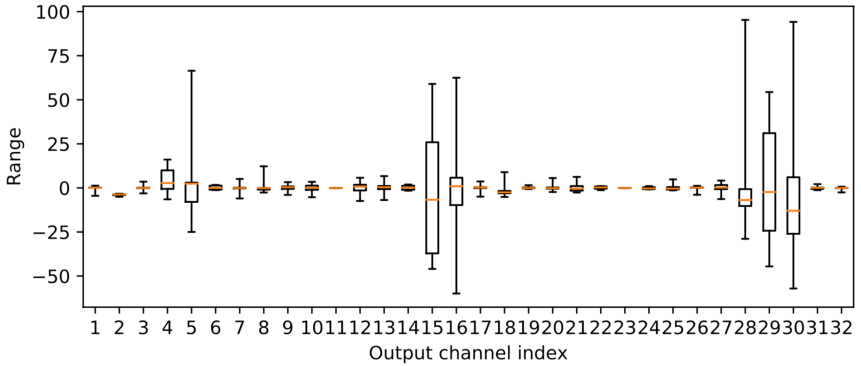

Figure 2. Per (output) channel weight ranges of the first depthwise-separable layer in MobileNetV2. In the boxplot the min and max value, the 2nd and 3rd quartile and the median are plotted for each channel. This layer exhibits strong differences between channel weight ranges.

Proposed Method

- 착안점 2: Biased quantization error
  - 흔히 quantization error에 bias가 없고, output 값에서 모두 상쇄된다고 가정함. 하지만 실제로는 weight의 quantization error에 bias가 있으며 output도 bias된 값이 나오게 됨.
  - 다음의 식을 따라 quantized layer output의 bias를 근사적으로 계산했을 때,

$$\mathbb{E}[\tilde{\mathbf{y}}_j - \mathbf{y}_j] \approx \frac{1}{N} \sum_n (\widetilde{\mathbf{W}}\mathbf{x}_n)_j - (\mathbf{W}\mathbf{x}_n)_j$$

MobileNetV2 기준 다음과 같은 분포(blue)를 보임. 즉, 실제로 편향이 존재하는 채널이 다수 존재.

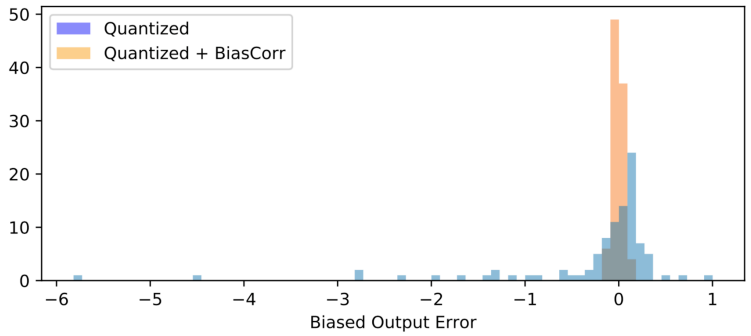

Figure 3. Per-channel biased output error introduced by weight quantization of the second depthwise-separable layer in MobileNetV2, before (blue) and after (orange) bias correction.

Proposed Method

- DFQ
  - 제시한 문제점들을 해결하기 위해 일반적인 quantization에 세 단계를 더함

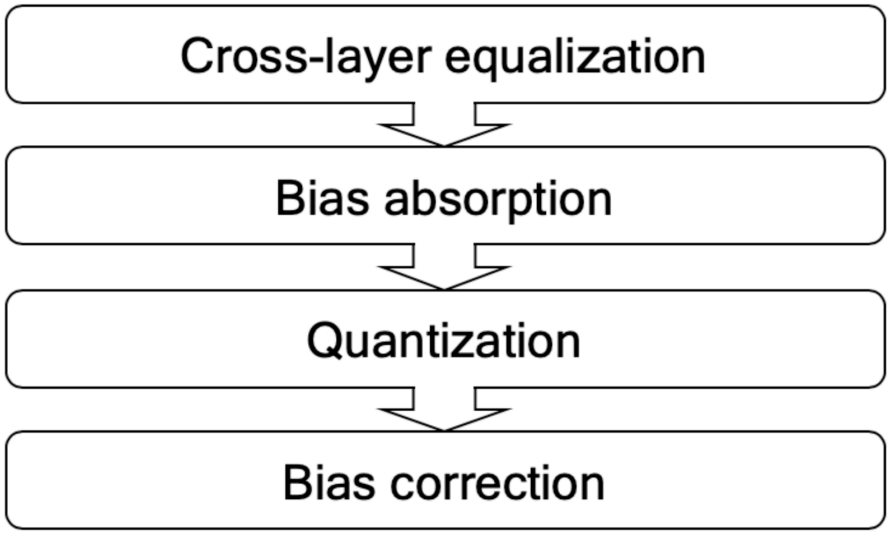

Figure 4. Flow diagram of the proposed DFQ algorithm.

Proposed Method

- 1. Cross-layer range equalization
  - Scaling equivariance in activation

Positive scaling equivariance in ReLU: $f(sx) = sf(x) \qquad \forall s \geq 0$

…can be relaxed to any piece-wise linear activation functions

$$f(x) = \begin{cases} a_1 x + b_1 & \text{if } x \leq c_1 \\ a_2 x + b_2 & \text{if } c_1 < x \leq c_2 \\ & \vdots \\ a_n x + b_n & \text{if } c_{n-1} < x \end{cases}$$

Proposed Method

- 1. Cross-layer range equalization
  - Scaling equivariance in neural network

Parameterization:

$$\begin{aligned}\mathbf{y} &= f(\mathbf{W}^{(2)} f(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}) \\ &= f(\mathbf{W}^{(2)}\mathbf{S}\hat{f}(\mathbf{S}^{-1}\mathbf{W}^{(1)}\mathbf{x} + \mathbf{S}^{-1}\mathbf{b}^{(1)}) + \mathbf{b}^{(2)}) \\ &= f(\widehat{\mathbf{W}}^{(2)}\hat{f}(\widehat{\mathbf{W}}^{(1)}\mathbf{x} + \widehat{\mathbf{b}}^{(1)}) + \mathbf{b}^{(2)})\end{aligned}$$

$$\widehat{\mathbf{W}}^{(1)} = \mathbf{S}^{-1}\mathbf{W}^{(1)}$$

$$\widehat{\mathbf{W}}^{(2)} = \mathbf{W}^{(2)}\mathbf{S}$$

$$\widehat{\mathbf{b}}^{(1)} = \mathbf{S}^{-1}\mathbf{b}^{(1)}$$

Re-scaling:

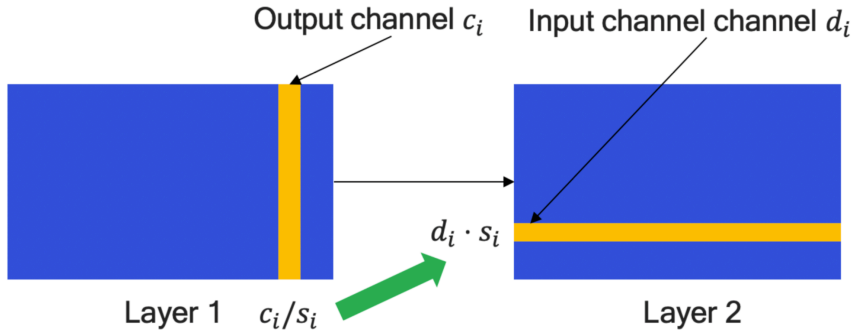

Figure 5. Illustration of the rescaling for a single channel. If scaling factor s_i scales c_i in layer 1; we can instead factor it out and multiply d_i in layer 2.

Proposed Method

- 1. Cross-layer range equalization

Precision of a channel (channel의 range over range of the whole tensor로 정의):

$$p_i^{(1)} = \frac{r_i^{(1)}}{R^{(1)}}$$

Want to find S such that total precision-per-channel is maximized:

$$\max_{\mathbf{S}} \sum_i p_i^{(1)} p_i^{(2)}$$

Necessary condition for maximization problem:

$$\arg\max_j \frac{1}{s_j} r_j^{(1)} = \arg\max_k s_k r_k^{(2)} \quad \Rightarrow \quad s_i = \frac{1}{r_i^{(1)}} \sqrt{r_i^{(1)} r_i^{(2)}} \quad \Rightarrow \quad \forall i : r_i^{(1)} = r_i^{(2)}$$

즉, 두 채널의 range가 최대한 동일한 값이어야 함

여러 layer에 대해서 동시에 수행할 때는, 인접한 layer pair에 대해서 수렴할 때까지 반복

Proposed Method

- 2. Bias absorption
  - $s_i < 1$인 경우에는 앞의 range-equalization 과정에서 bias $b_i^{(1)}$가 커짐

    → activation quantization의 range가 오히려 커질 수 있기 때문에, 이 range를 다음 layer로 흡수시켜줌
  - ReLU에 대해서 다음을 만족하는 non-negative vector c가 존재

$$r(\mathbf{W}\mathbf{x} + \mathbf{b} - \mathbf{c}) = r(\mathbf{W}\mathbf{x} + \mathbf{b}) - \mathbf{c}$$

Proposed Method

- 2. Bias absorption
  - Trivial solution 0을 제외하고, $r(\mathbf{W}\mathbf{x}+\mathbf{b}-\mathbf{c}) = r(\mathbf{W}\mathbf{x}+\mathbf{b}) - \mathbf{c}$ 를 만족하는 c에 대해 다음이 성립함:

$$
\begin{aligned}
\mathbf{y} &= \mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)} \\
&= \mathbf{W}^{(2)}(r(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{c} - \mathbf{c}) + \mathbf{b}^{(2)} \\
&= \mathbf{W}^{(2)}(r(\mathbf{W}^{(1)}\mathbf{x} + \hat{\mathbf{b}}^{(1)}) + \mathbf{c}) + \mathbf{b}^{(2)} \\
&= \mathbf{W}^{(2)}\hat{\mathbf{h}} + \hat{\mathbf{b}}^{(2)}
\end{aligned}
$$

$$
\hat{\mathbf{b}}^{(2)} = \mathbf{W}^{(2)}\mathbf{c} + \mathbf{b}^{(2)}, \hat{\mathbf{h}} = \mathbf{h} - \mathbf{c}, \text{ and } \hat{\mathbf{b}}^{(1)} = \mathbf{b}^{(1)} - \mathbf{c}
$$

Proposed Method

- 2. Bias absorption
  - 추가적인 데이터 없이 c를 찾기 위해서, pre-bias activation이 batch normalization scale(γ)과 shift (β)의 분포를 따른다고 가정함.
  - $\mathbf{c} = \max(\mathbf{0}, \boldsymbol{\beta} - 3\boldsymbol{\gamma})$로 설정하면 99.865%의 x들에 대해서 앞의 식이 성립하여, 첫번째 레이어의 bias를 두번째 레이어의 bias로 흡수시킬 수 있음.
  - (데이터가 있는 경우엔, pre-bias distribution을 emprical하게 구해서 c를 정하면 됨)

Proposed Method

- Result of cross-layer range equalization bias absorption

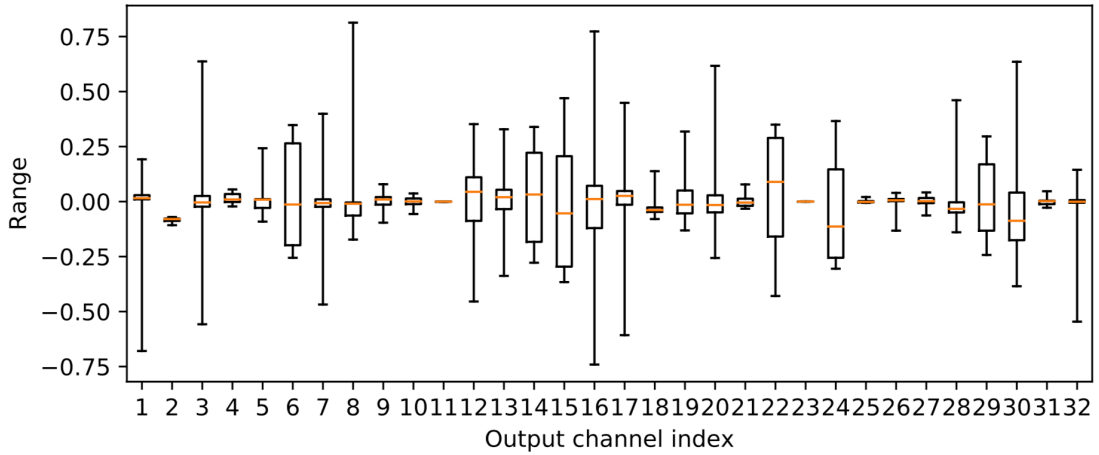

Figure 6. Per (output) channel weight ranges of the first depthwise-separable layer in MobileNetV2 after equalization. In the boxplot the min and max value, the 2nd and 3rd quartile and the median are plotted for each channel. Most channels in this layer are now within similar ranges.

Proposed Method

- 3. Quantization Bias Correction
  - 앞에서 확인한, quantization 후 output에 존재하는 bias를 제거하기 위해 다시 batch norm의 파라미터를 이용.
  - Weight W, quantization 후 weight를 $\widetilde{W}$라고 할 때, $\tilde{y} = \widetilde{W}x,\ \tilde{y} = y + \epsilon x$. Quantization error ϵ을 $\widetilde{W} - W$로 정의.

    → $$\begin{aligned}\mathbb{E}[\mathbf{y}] &= \mathbb{E}[\mathbf{y}] + \mathbb{E}[\boldsymbol{\epsilon}\mathbf{x}] - \mathbb{E}[\boldsymbol{\epsilon}\mathbf{x}] \\ &= \mathbb{E}[\widetilde{\mathbf{y}}] - \mathbb{E}[\boldsymbol{\epsilon}\mathbf{x}].\end{aligned}$$

    → Input의 평균값에 해당하는 마지막 항을 output에서 빼주면 됨

Proposed Method

- 3. Quantization Bias Correction
  - Then, how can we get the expectation of x?
    - Batch-norm을 사용하지 않거나, data를 사용할 수 있는 경우라면 quantization 전후의 activation을 비교해서 계산 가능
    - Batch-norm을 사용하는 경우
      - Assumption 1) Activation 전에 batch norm이 적용되어 activation 직전 output이 normally distributed 되어 있음.
      - Assumption 2) Activation function이 input을 특정 range [a,b]에 맞게 클립하는 clipped linear activation function이라고 가정함.

Proposed Method

- 3. Quantization Bias Correction
  - 앞에서와 마찬가지로 batch norm parameter scale (γ)과 shift (β)로부터 pre-activation의 mean, std 가정 가능
  - Clipped normal distribution: range [a,b]로 clip된 variabl들이 따르는 분포 (μ, σ^2)

    → a, b, μ, σ의 관계는 closed-form solution이 존재함
  - ReLU를 예로 들면, 다음의 관계가 있음 (c는 채널)

$$\begin{aligned}\mathbb{E}[\mathbf{x}_c] &= \mathbb{E}\left[\mathrm{ReLU}\left(\mathbf{x}_c^{pre}\right)\right] \\ &= \boldsymbol{\gamma}_c \mathcal{N}\left(\frac{-\boldsymbol{\beta}_c}{\boldsymbol{\gamma}_c}\right) + \boldsymbol{\beta}_c\left[1 - \Phi\left(\frac{-\boldsymbol{\beta}_c}{\boldsymbol{\gamma}_c}\right)\right]\end{aligned}$$

Experiments

- Ablation study
  - A new state-of-the-art for level 1 quantization!

Model	FP32	INT8
Original model	71.72%	0.12%
Replace ReLU6	71.70%	0.11%
+ equalization	71.70%	69.91%
+ absorbing bias	71.57%	**70.92%**
Per channel quantization	71.72%	70.65%

Table 1. Top1 ImageNet validation results for MobileNetV2, evaluated at full precision and 8-bit integer quantized. Per-channel quantization is our own implementation of [16] applied post-training.

Model	FP32	INT8
Original Model	71.72%	0.12%
Bias Corr	71.72%	**52.02%**
Clip @ 15	67.06%	2.55%
+ Bias Corr	**71.15%**	**70.43%**
Rescaling + Bias Absorption	71.57%	70.92%
+ Bias Corr	71.57%	**71.19%**

Table 2. Top1 ImageNet validation results for MobileNetV2, evaluated at full precision and 8-bit integer quantized. Bold results show the best result for each column in each cell.

Experiments

- Comparison to other works

	∼D	∼BP	∼AC	MobileNetV2		MobileNetV1		ResNet18		
				FP32	INT8	FP32	INT8	FP32	INT8	INT6
DFQ (ours)	✓	✓	✓	71.7%	**71.2%**	70.8%	**70.5%**	69.7%	**69.7%**	66.3%
Per-layer [18]	✓	✓	✓	71.9%	0.1%	70.9%	0.1%	69.7%	69.2%*	63.8%*
Per-channel [18]	✓	✓	✓	71.9%	69.7%	70.9%	70.3%	69.7%	69.6%*	**67.5%***
QT [16] ˆ	✗	✗	✓	71.9%	70.9%	70.9%	70.0%	-	**70.3%**†	67.3%†
SR+DR†	✗	✗	✓	-	-	-	**71.3%**	-	68.2%	59.3%
QMN [31]	✗	✗	✗	-	-	70.8%	68.0%	-	-	-
RQ [21]	✗	✗	✗	-	-	-	70.4%	-	69.9%	**68.6%**

Table 5. Top1 ImageNet validation results for different models and quantization approaches. The top half compares level 1 approaches (∼D: data free, ∼BP: backpropagation-free, ∼AC: Architecture change free) whereas in the second half we also compare to higher level approaches in literature. Results with * indicates our own implementation since results are not provided, ˆ results provided by [18] and † results from table 2 in [21].

Experiments

- In other tasks..

Model	FP32	INT8
Original model	72.94	41.40
DFQ (ours)	72.45	**72.33**
Per-channel quantization	72.94	71.44

Table 3. DeeplabV3+ (MobileNetV2 backend) on Pascal VOC segmentation challenge. Mean intersection over union (mIOU) evaluated at full precision and 8-bit integer quantized. Per-channel quantization is our own implementation of [16] applied post-training.

Model	FP32	INT8
Original model	68.47	10.63
DFQ (ours)	68.56	**67.91**
Per-channel quantization	68.47	67.52

Table 4. MobileNetV2 SSD-lite on Pascal VOC object detection challange. Mean average precision (mAP) evaluated at full precision and 8-bit integer quantized. Per-channel quantization is our own implementation of [16] applied post-training.