Is In-Domain Data Really Needed?
A Pilot Study on Cross-Domain Calibration for Network Quantization

Haichao Yu¹, Linjie Yang², Humphrey Shi¹
¹University of Illinois at Urbana-Champaign, ²ByteDance Inc.
haichao3@illinois.edu, linjie.yang@bytedance.com, shihonghui3@gmail.com

Abstract

Post-training quantization methods use a set of calibration data to compute quantization ranges for network parameters and activations. The calibration data usually comes from the training dataset which could be inaccessible due to sensitivity of the data. In this work, we want to study such a problem: can we use out-of-domain data to calibrate the trained networks without knowledge of the original dataset? Specifically, we go beyond the domain of natural images to include drastically different domains such as X-ray images, satellite images and ultrasound images. We find cross-domain calibration leads to surprisingly stable performance of quantized models on 10 tasks in different image domains with 13 different calibration datasets. We also find that the performance of quantized models is correlated with the similarity of the Gram matrices between the source and calibration domains, which can be used as a criterion to choose calibration set for better performance. We believe our research opens the door to borrow cross-domain knowledge for network quantization and compression.

1. Introduction

With the increasing popularity of deploying neural networks on edge devices, neural network quantization has become a widely studied topic [6, 32, 27, 43, 20, 42, 39, 15, 2, 41]. By quantizing its weights and activations to low-bit integers, a neural network can be stored with smaller size and executed at faster speed with less memory footprint and computational resources.

Most existing network quantization methods can be roughly divided into two groups. One is quantization-aware training (QAT), the other is post-training quantization (PQ). By inserting differentiable simulated quantization operations into the network during training, QAT methods allow training losses to back-propagate through the quantization operations. Although QAT methods can achieve satisfactory performance in most cases, it is time-consuming and data-hungry, which is unacceptable in some scenarios, e.g., when the time budget is restricted or training dataset is inaccessible. Compare with QAT, PQ has advantages on time and data efficiency. Given a pretrained full-precision model, PQ runs the model over a small set of calibration data to calculate value ranges of intermediate activations. These calibration data shares the same distribution as the training data on which the model is trained. After this, both network activations and weights can be quantized into low-precision integers for inference acceleration. Compared with QAT, this post-training quantization process is highly efficient (e.g., in a few minutes).

However, PQ methods still require in-domain calibration data, which may be inaccessible in some situations due to privacy or security reasons. For example, federated learning [16] trains models using distributed private user data that are not accessible by the developers. Without calibration data from the same domain as the training data, the traditional PQ methods will fail. Targeting at this problem, Cai et al. proposed a novel method termed ZeroQ that generates synthetic data for calibration [3]. Specifically, ZeroQ utilized BatchNorm [13] statistics from the pretrained model as supervision to optimize randomly initialized input such that the BatchNorm statistics of the input are similar to those from the pretrained model. ZeroQ showed its
effectiveness on various datasets including ImageNet [7] and MSCOCO [23]. Another method is to use Generative Adversarial Networks (GAN) to create fake calibration data [37]. The proposed framework named GDFQ is trained with BatchNorm statistics loss, Cross Entropy classification loss and knowledge distillation loss.

One drawback of these data-free quantization methods is their high time complexity. With common hardware setting (e.g., on single GPU), ZeroQ requires hundreds of back-propagation iterations to synthesize a single batch and GDFQ needs to be trained for hundreds of epochs. On the other hand, due to the optimization nature of these methods, hyper-parameters also need to be tuned for satisfactory results. If in-domain training data is inaccessible and synthetic data generation is time-consuming, a straightforward question to ask is “Can we use real-world images from another domain for calibration?”. As shown in Figure 1, we use calibration data from domain $B$ to calibration model trained on domain $A$, resulting in a quantized model. The model is further evaluated on a test set in domain $A$.

To study this problem, we carry out a large-scale benchmark with 10 different tasks and 13 diversified image domains including nature images, low-resolution images, ultrasound images, satellite images, etc. We find that a simple treatment on the BatchNorm layers in the calibration procedure greatly improves performance of cross-domain calibration, almost bridging the gap between in-domain and out-of-domain calibration on a wide range of tasks. We also find that the performance of quantized models is correlated with the similarity of the Gram matrices between the source and calibration domains, which can be used as a criterion to choose similar image domains for better performance. Compared with synthetic data generation method ZeroQ, our approach achieves comparable or better performance on our large-scale benchmark. Although the study is mainly empirical, it reveals that cross-domain data can be used for post-training calibration just as effective as in-domain data, which could motivate the community to explore more in the direction of model quantization and compression with cross-domain knowledge.

2. Related Work

2.1. Quantization-Aware Training

Quantization-aware training (QAT) inserts simulated quantization operation into model forward and backward passes during training. First introduced by Courbariaux et al. [12], binary neural networks achieved seven times faster inference speed on GPU. Further in [32], Rastegari et al. proposed to use a binary tensor and a float scalar to approximate the weight or activation tensor. Parallel to binarization, multi-bit model quantization is also widely explored. DoReFa-Net [42] was proposed to quantize both network and training gradients. Scale-adjusted training [14] improve the model performance by scaling activation in the network to preserve proper scale of gradients. To bring flexibility to the model quantizer, some works also introduced learnable quantization schemes. In [39], LQ-Net was proposed to learn the quantization basis vectors. In [15], Jung et al. proposed a novel method to learn the quantization intervals.

2.2. Post-training Quantization

Different from QAT, post-training quantization (PQ) directly computes quantization parameters with a calibration set. In [17], Krishnamoorthi et al. introduced PQ with layer-wise and channel-wise quantization schemes. To improve quantization performance, Banner et al. proposed a novel per-channel PQ scheme that analytically computes clipping thresholds and bits allocated for each channel [2]. To mitigate the quantization error from the clipping operation, Zhao et al. introduced a technique named OCS to split the channels with extreme values [41].

Traditionally, PQ method requires a calibration dataset which is often sampled from training data to estimate the clipping ranges of model activations. However, in some scenarios, training data may be inaccessible due to data privacy issues. In this case, quantization could be conducted without a calibration set. Assuming features from a BatchNorm layer follow Gaussian distribution, Nagel et al. directly used BatchNorm statistics (i.e., running means and variances) to estimate activation ranges [29]. The drawback of this method is the estimated ranges may be inaccurate since the distribution of intermediate layers may be different from Gaussian distribution. DFC and ZeroQ proposed to synthesize calibration data by gradient descent under the supervision of BatchNorm statistics [10, 3]. Generative Adversarial Networks (GAN) is also utilized to generate synthetic calibration data [37], in which the generator generates calibration data and the discriminator is the quantized model. One problem with these optimization-based methods is the high time complexity. Hundreds of iterations or even hundreds of epochs of back-propagation are commonly required. In addition, much human efforts are needed for hyper-parameter tuning such as learning rate and optimizer selection. In contrast to these methods, we find that out-of-domain image datasets can serve as calibration data effectively for a wide range of tasks.

2.3. Batch Normalization and Domain Knowledge

To adapt a deep neural network to a different domain, Li et al. proposed Adaptive BatchNorm to update running statistics in target domain [22]. The central assumption is that domain specific information is encoded in BatchNorm layers. Sharing a similar spirit with this, Li et al. showed that style transfer can be conducted by matching the BatchNorm statistics between two images [21]. Inspired by
the effect of BatchNorm statistics, we design a BatchNorm updating method on out-of-domain calibration dataset to estimate ranges of activations, which effectively improves the performance of cross-domain calibration even when the training and the calibration domains have huge appearance differences.

3. Cross-domain Calibration for Post-training Quantization

3.1. Motivation

Given a pretrained full-precision model, post-training quantization (PQ) is widely used for inference acceleration. Most existing PQ methods require a set of calibration data to calculate the quantization parameters for weights and activations. Training data is normally used for calibration. However, in some situations, the training data is not available due to privacy or security issues. To solve this problem, Cai et al. [3] proposed to synthesize calibration data using BatchNorm statistics as supervision. Since different types of real images are vastly available, we would like to explore another option: use out-of-domain real images for calibration. We conduct a large-scale empirical study with drastically different datasets and tasks to investigate this setting.

3.2. Quantization Schemes

In all our experiments, we employ layer-wise uniform quantization for both network weights and activations [17]. To quantize a tensor \( x \) by \( k \)-bit, the quantized tensor \( x_q \) is calculated as

\[
x_{\text{int}} = \text{round}\left(\frac{x}{s}\right) + z,
\]

\[
x_q = \text{clip}\left(x_{\text{int}}, c_l, c_h\right),
\]

where \( s \) and \( z \) are scale and zero point parameters, \( c_l \) and \( c_h \) are lower and upper clipping thresholds. For network weight \( w \), we use a symmetric min-max quantization scheme, which is defined as

\[
s_w = \max\left(|w_{\text{min}}|, |w_{\text{max}}|\right),
\]

\[
z_w = 0,
\]

\[
c_l = -2^{k-1},
\]

\[
c_h = 2^{k-1} - 1,
\]

where \( w_{\text{min}} \) and \( w_{\text{max}} \) are minimum and maximum of \( w \). To quantize activation \( a \), we employ an affine histogram quantization scheme, which is also used in PyTorch\(^1\). To determine the quantization parameters, we have

\[
s_a = \frac{a_k - a_l}{2^k - 1},
\]

\[
z_a = a_l(2^k - 1),
\]

\[
c_l = 0,
\]

\[
c_h = 2^k - 1,
\]

where \( a_l \) and \( a_h \) are lower and upper clipping thresholds for \( a \). For each layer, a histogram is built to represent the activation distribution. We search for the optimal \( a_l \) and \( a_h \) values such that the quantization error is minimized with respect to \( a \). In all our experiments, we first fold BatchNorm layers to their preceding convolutional or linear layers before computing quantization parameters.

3.3. Datasets and Networks

Our experiments span across different image domains including natural images [19, 7, 8, 5, 23, 18], X-ray [36], ultrasound [1], and satellite images [4]. We also experiment with models in various computer vision tasks including image classification, semantic segmentation and object detection. The datasets and corresponding example images are illustrated in Figure 2. We train floating-point models on all the datasets from Figure 2a to Figure 2j, where each dataset has one or more associated models. These datasets and three additional datasets from Figure 2k to Figure 2m are used as calibration sets. When calibration and training data have different sizes, we resize the calibration data to the same size of the training data. One exception is when calibrating ResNet-18 on Cifar100, calibration samples are randomly cropped. In addition, when Agriculture-Vision data is used to calibrate models on other datasets, only RGB channels are used. When other datasets are used to calibrate Agriculture-Vision models, we use RGB channels to compute a grayscale channel to be used as NIR channel.

These models to be quantized are summarized in Table 1. The evaluation metrics are summarized as below:

1. All classification tasks except CelebA: top-1 accuracy.
2. CelebA: average top-1 accuracy on 40 attributes.
3. Pascal VOC 2007: mAP.
4. MSCOCO, Cityscapes and Agriculture-Vision: mIoU.
5. Ultrasound: DICE.

Specifically, DICE is a widely used evaluation metric for binary segmentation, which is defined as

\[
\text{DICE} = \frac{2|p \cap g|}{|p| + |g|},
\]

where \( p \) and \( g \) are binary prediction and ground truth tensors respectively.

\(^1\)https://pytorch.org/docs/master/torch.quantization.html


<table>
<thead>
<tr>
<th>Datasets</th>
<th>Tasks</th>
<th>Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSCOCO</td>
<td>S</td>
<td>Inception-V3 [34]</td>
</tr>
<tr>
<td>Cityscapes</td>
<td>S</td>
<td>FCN [28]</td>
</tr>
<tr>
<td>Pascal VOC 2007</td>
<td>D</td>
<td>MobileNetV2 SSD-Lite [33] [25]</td>
</tr>
<tr>
<td>Cifar100</td>
<td>C</td>
<td>ResNet-18</td>
</tr>
<tr>
<td>CelebA</td>
<td>C</td>
<td>ResNet-50</td>
</tr>
<tr>
<td>CUB-200-2011</td>
<td>C</td>
<td>MMAL-Net [40]</td>
</tr>
<tr>
<td>Ultrasound</td>
<td>S</td>
<td>U-net</td>
</tr>
<tr>
<td>NIH Chest X-ray</td>
<td>C</td>
<td>ResNet-50</td>
</tr>
</tbody>
</table>

3.4. Cross-domain Calibration

A Naive Approach  First we employ a naive cross-domain calibration approach, *i.e.*, directly using out-of-domain data to calibrate a model trained in another domain. The results of 8-bit quantization are plotted as blue bars in Figure 3. The in-domain calibration results are marked in dashed lines.

We can see that for all the datasets and models, a large number of out-of-domain calibration results are comparable to the in-domain baselines. Take ResNet-50 on ImageNet as an example, the gap of out-of-domain and in-domain calibration results is very small. This observation is surprising since most existing PQ methods assume in-domain calibration data is required to estimate activation ranges of intermediate layers. For some models such as MobileNetV2 SSD-Lite on Pascal VOC, ResNet-50 on CelebA, U-net on Ultrasound, and ResNet-50 on NIH Chest X-ray, there is still an accuracy gap between out-of-domain calibration results and the baseline in-domain results. Such discrepancy often occurs when the original and the calibration domain have a large gap, *e.g.* most natural image datasets are drastically different from ultrasound images. Motivated by this observation, we introduce an approach to reduce the representation gap between the original and the calibration domains for the target model, namely BatchNorm adjustment.
BatchNorm Adjustment Since BatchNorm was proposed to reduce Internal Covariate Shift [13], it has been widely applied to deep neural networks. For one neural network, different domains will generate different BatchNorm parameters. One interesting application is that BatchNorm parameters can be updated on a new domain for domain adaptation [22, 21]. As stated by these work, BatchNorm layers encode information that is specific to the domain that the model is trained on. Inspired by this, we propose BatchNorm adjustment to adapt the models to the out-of-domain calibration datasets. We describe our method as below:

1. Given a model $M$ to be quantized and an out-of-domain dataset $D$.
2. Reset running means and variances of all BatchNorm layers in $M$: $\mu \leftarrow 0$, $\sigma \leftarrow 1$.
3. Run $M$ on $D$ to accumulate new BatchNorm statistics.
4. Fold BatchNorm layers to their preceding convolutional or linear layers.
5. Calculate weight quantization parameters ($s_w$ and $z_w$ in Equation 2).
6. Run $M$ on $D$ again to calculate activation quantization parameters ($s_a$ and $z_a$ in Equation 3).

The results are plotted in orange bars in Figure 3. For most datasets and models, BatchNorm adjustment gets better performance than the baselines without BatchNorm adjustment. In most cases, the results are comparable to or even slightly better than the in-domain calibration results. On ResNet-18/50 on ImageNet, FCN on MSCOCO, ResNet-18 on CIFAR100, U-net on Ultrasound and ResNet-50 on NIH Chest X-ray, all out-of-domain calibration results are within 0.3% performance gap from the in-domain calibration results. On some tasks such as MMAL-Net on CUB-200-2011 and MSCI-Net-101 on Agriculture-Vision, some out-of-domain data still performs relatively lower than others. In such cases, randomly chosen calibration datasets cannot guarantee performance similar to in-domain calibration. In order to fix this issue, we investigate how domain discrepancy affects calibration and how to improve the calibration results.

3.5. Influence of Domain Discrepancy on Calibration

In this section, we investigate the relationship between domain discrepancy and cross-domain calibration performance. Since Maximum Mean Discrepancy (MMD) was proposed to measure difference of sample mean in Reproducing Kernel Hilbert Space [9], it has been widely used as a domain discrepancy measure. As proved by Li et al. [21],
matching MMD with the second order polynomial kernel is equivalent to matching Gram matrices of feature maps. Similar to [21], we employ mean $L_2$ distance between Gram matrices of feature maps to measure the discrepancy between two image domains. Formally, domain discrepancy $D$ is defined as

$$D = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \| G_{ij}^A - G_{ij}^B \|_2^2, \quad (5)$$

where $G^A \in \mathbb{R}^{N \times N}$ and $G^B \in \mathbb{R}^{N \times N}$ are average Gram matrices of two domains $A$ and $B$, which are defined as

$$G^A = \frac{1}{|A|} \sum_{k=1}^{|A|} G^A_k \quad \text{and} \quad G^B = \frac{1}{|B|} \sum_{k=1}^{|B|} G^B_k, \quad (6)$$

where $|A|$ and $|B|$ are number of samples in domain $A$ and $B$ respectively. Ignoring the domain superscript for simplicity, each element $G_{k,ij}$ in $G^A_k$ or $G^B_k$ is

$$G_{k,ij} = \sum_{m=1}^{M} F_{k,im} F_{k,jm}, \quad (7)$$

where $F_k$ is the feature embedding of $k$-th sample, which is normalized by mean and standard deviation over all images in this domain, and $M$ is the embedding dimension.

In our experiments, we use the output of last conv layer in the fourth conv block conv_4_3 of VGG-16 to extract the feature maps.

Next we study how domain discrepancy is related to calibration performance. In Figure 4, we plot model performance vs domain discrepancy with the source training data into scatter points. We also calculate correlation coefficient between model performance and domain discrepancy shown in the title of each sub-figure. First, on most of the datasets, calibration performance is negatively correlated to domain discrepancy in red for better visualization. The out-of-domain datasets that have the smaller domain discrepancy with the in-domain data always achieve comparable performance with in-domain data. We marked the top-3 out-of-domain data with smallest domain discrepancy in red for better visualization. This observation can be useful for cross-domain calibration, when calibration/training data is sensitive and inaccessible but some high-level statistics such as Gram matrix are available. One can ask the data owner to provide a mean Gram matrix of the source dataset, which can be used to search in the pre-built candidate pool of cross-domain datasets for the out-of-domain calibration dataset with smallest domain discrepancy.

Using the above strategy, we compare our method with in-domain calibration and data synthesis approach Ze-
Table 2: 8-bit quantization results with different calibration datasets. The dataset abbreviations are defined in Figure 2. R-18, R-50 and IV3 are Resnet-18, Resnet-50 and InceptionV3 respectively. Cross-domain (MS) uses an average of gram matrices from multiple layers. Best results without using in-domain data are emphasized in bold.

<table>
<thead>
<tr>
<th>Calib Methods \ Datasets</th>
<th>IN R18</th>
<th>IN R50</th>
<th>IN IV3</th>
<th>CO</th>
<th>CS</th>
<th>VO</th>
<th>CI</th>
<th>CE</th>
<th>CB</th>
<th>AG</th>
<th>US</th>
<th>NI</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP32</td>
<td>69.76</td>
<td>76.13</td>
<td>77.46</td>
<td>60.47</td>
<td>79.10</td>
<td>0.686</td>
<td>76.50</td>
<td>91.74</td>
<td>89.63</td>
<td>54.63</td>
<td>0.730</td>
<td>0.773</td>
</tr>
<tr>
<td>In-domain</td>
<td>69.39</td>
<td>75.73</td>
<td>76.88</td>
<td>60.17</td>
<td>79.09</td>
<td>0.685</td>
<td>76.30</td>
<td>91.72</td>
<td>89.36</td>
<td>53.93</td>
<td>0.738</td>
<td>0.758</td>
</tr>
<tr>
<td>ZeroQ</td>
<td>69.36</td>
<td>75.56</td>
<td>76.89</td>
<td>60.17</td>
<td>78.99</td>
<td>0.681</td>
<td>76.01</td>
<td>91.59</td>
<td>87.90</td>
<td>53.34</td>
<td>0.739</td>
<td>0.718</td>
</tr>
<tr>
<td>ZeroQ-real</td>
<td>69.29</td>
<td>75.57</td>
<td>76.89</td>
<td>60.17</td>
<td>78.87</td>
<td>0.681</td>
<td>75.87</td>
<td>91.62</td>
<td>89.29</td>
<td>53.29</td>
<td>0.738</td>
<td>0.675</td>
</tr>
<tr>
<td>Cross-domain</td>
<td>69.47</td>
<td>75.70</td>
<td>76.93</td>
<td>60.12</td>
<td>79.00</td>
<td>0.685</td>
<td>76.36</td>
<td>91.50</td>
<td>89.19</td>
<td>53.85</td>
<td>0.733</td>
<td>0.761</td>
</tr>
<tr>
<td>Cross-domain (MS)</td>
<td>69.47</td>
<td>75.70</td>
<td>76.93</td>
<td>60.08</td>
<td>79.01</td>
<td>0.684</td>
<td>76.37</td>
<td>91.50</td>
<td>89.19</td>
<td>53.38</td>
<td>0.732</td>
<td>0.761</td>
</tr>
</tbody>
</table>

Figure 5: Visualization of Gram matrices of different datasets. The feature extraction model is VGG-16 with BatchNorm trained on ImageNet. In Column (b) and (c), BatchNorm adjustment is applied before calculating gram matrices. In Column (d), VGG-16 is used to synthesize the ZeroQ data. Best viewed in color.

In addition, we also compare with a variant of ZeroQ that we term as ZeroQ-real, short for ZeroQ with real images. Different from ZeroQ that initializes the synthesized data with random Gaussian values, ZeroQ-real uses real images for initialization. In all the experiments of ZeroQ-real, ImageNet data is used for initialization. We show performance of two variants of our method with one or multiple feature maps in VGG-16 to compute domain discrepancy, shown as Cross-domain and Cross-domain (MS) respectively. When multiple feature maps are used, each feature map will generate a Gram matrix and the domain discrepancy is defined as the average of $L_2$ distances of all Gram matrices. As shown in Table 2, our method achieves comparable or even better results than both ZeroQ and ZeroQ-real. Using one or multiple feature maps to compute domain discrepancy achieves similar performance across different tasks. Our method using selected cross-domain calibration data is on-par with in-domain data, proving the effectiveness of using cross-domain calibration data for post-training quantization.

3.6. Visualization of Activation Ranges

In this section, we visualize the activation ranges (clipping thresholds) calculated on different out-of-domain datasets. In Figure 6, we show three examples: Ultrasound to calibrate ResNet-18 on ImageNet, NIH Chest X-ray to calibrate FCN on MSCOCO, and ImageNet to calibrate ResNet-50 on NIH Chest X-ray. In each sub-figure, we show the lower and upper clipping thresholds for each layer. Compared with the naive cross-domain calibration method, the proposed BatchNorm adjustment makes a closer estimation of activation ranges to those from the in-domain data. Specifically, for ResNet-50 on NIH Chest X-ray, Batch-
In this work, we explored cross-domain calibration for post-training quantization. To study this problem, we conducted a large-scale study that spans across various tasks, datasets and neural networks. We find that a simple BatchNorm adjustment strategy can effectively improve the performance of quantized models by a large margin, almost bridging the gap between cross-domain calibration and in-domain calibration. In addition, we find that performance with cross-domain calibration is correlated with Gram matrix similarity between the source and the calibration domains. Therefore, Gram matrix similarity can be used as a criterion to select calibration dataset from a candidate pool to further improve performance. We believe our work will motivate future research on utilizing cross-domain knowledge for network quantization and compression.
References


the European conference on computer vision (ECCV), pages 722–737, 2018.


