Do All MobileNets Quantize Poorly? Gaining Insights into the Effect of Quantization on Depthwise Separable Convolutional Networks Through the Eyes of Multi-scale Distributional Dynamics

Stone Yun[1,2] and Alexander Wong[1,2]
[1]Vision and Image Processing Research Group, University of Waterloo
[2]Waterloo Artificial Intelligence Institute, Waterloo, Canada
{s22yun, a28wong}@uwaterloo.ca

Abstract

As the "Mobile AI" revolution continues to grow, so does the need to understand the behaviour of edge-deployed deep neural networks. In particular, MobileNets [9, 22] are the go-to family of deep convolutional neural networks (CNN) for mobile. However, they often have significant accuracy degradation under post-training quantization. While studies have introduced quantization-aware training and other methods to tackle this challenge, there is limited understanding into why MobileNets (and potentially depthwise-separable CNNs (DWSCNN) in general) quantize so poorly compared to other CNN architectures. Motivated to gain deeper insights into this phenomenon, we take a different strategy and study the multi-scale distributional dynamics of MobileNet-V1, a set of smaller DWSCNNs, and regular CNNs. Specifically, we investigate the impact of quantization on the weight and activation distributional dynamics as information propagates from layer to layer, as well as overall changes in distributional dynamics at the network level. This fine-grained analysis revealed significant dynamic range fluctuations and a "distributional mismatch" between channelwise and layerwise distributions in DWSCNNs that lead to increasing quantized degradation and distributional shift during information propagation. Furthermore, analysis of the activation quantization errors show that there is greater quantization error accumulation in DWSCNN compared to regular CNNs. The hope is that such insights can lead to innovative strategies for reducing such distributional dynamics changes and improve post-training quantization for mobile.

1. Introduction

The past decade has seen an enormous boom in deep learning research, particularly in the fields of NLP and computer vision. As a result, deep learning algorithms such as convolutional neural network (CNN) models have become more accessible than ever. Mobile devices have become a primary platform on which CNNs have rapidly proliferated. "AI on-the-edge," has driven an increasing demand for deploying fast, power-efficient CNNs that can maintain highly accurate performance while operating in a resource-constrained environment. Consequently, various avenues of research have looked at making CNNs efficient enough to "fit" on mobile devices. Methods such as efficient CNN architecture design [9, 22, 8, 2, 10, 32, 29], weight pruning [5, 16, 28, 17, 15], and quantization [12, 5, 20, 21] are all aimed at reducing the storage, computation, and memory requirements of CNN algorithms. Among these methods, depthwise separable convolutional networks (DWSCNN) such as in MobileNets [9, 22, 8] and fixed point integer/quantized inference have become ubiquitous tools for designing efficient "Mobile AI" algorithms.

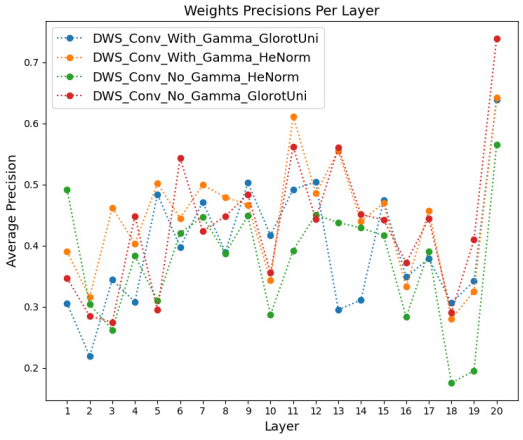

Figure 1. Layerwise average precison (see Eq. 2) of trained weights in the DWS-Convnets. Low average precision at depthwise-conv layers indicate a mismatch between each individual channel's dynamic range and the entire tensor's range. The resulting distributional dynamics lead to significantly higher accumulation of quantization errors. We see a similar pattern in the batchnorm-folded weights and activations of DWSCNNs.

As Mobile AI continues to proliferate, so does the need to better understand the behaviour of the models

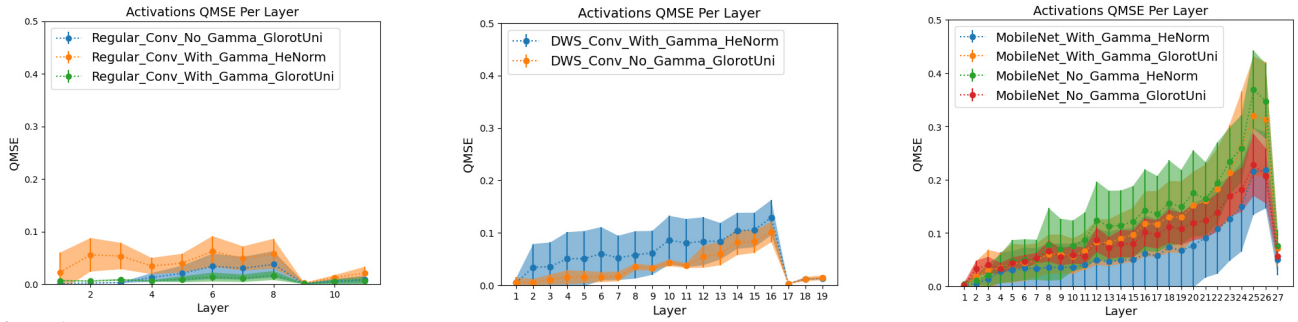

Figure 2. Comparing layerwise QMSE of Regular-ConvNets (left), DWS-ConvNets (center), and MobileNets-V1 (right). We are able to see how depthwise separable convolutional networks accumulate much more QMSE on average when traversing through a quantized DWSCNN. **Note**: Some outliers were removed from the Regular-ConvNet and DWS-ConvNet plots as they were on a much larger scale and obscure the rest of the results. See supplementary materials for diagrams of all layerwise QMSE results. The solid line represents the average values across 5 quantization trials and the shaded region is the standard deviation.

we deploy. Despite the success of MobileNets, there has been a well-documented phenomenon wherein simple post-training static quantization completely destroys their accuracy. While several works have designed solutions to address this, there is still limited data in the literature demonstrating *why* MobileNets quantize so poorly compared to other CNN architectures. Furthermore, we wonder if this problem is inherent to all DWSCNN architectures.

To investigate this, we do our best to recreate MobileNets-V1 training procedure described in [9], with some modifications for the CIFAR-10 experiments. Furthermore, based on findings in [30], it is possible that choices such as the random weight initialization method and potentially even the use of BatchNorm with/without scaling (ie. the γ parameter described in [11]) could have significant impact on the trained layerwise distributions to be quantized. Thus, we perform a systematic, iterative series of experiments to isolate the impact of these factors on quantized MobileNets-V1 performance. This experiment is then repeated on a set of smaller DWSCNN architectures (that we will refer to as DWS-ConvNets) to investigate if this problem is inherent to depthwise-separable convolutional networks in general. For a baseline comparison, we also train a corresponding set of Regular-ConvNets. Our systematic ablation study is coupled with fine-grained, layerwise analysis similar to those described in [30, 29].

We find that significantly fluctuating dynamic ranges from layer-to-layer and a "distributional mismatch" between channelwise and layerwise distributions leads to increasing quantized degradation. Consequently, our fine-grained analysis shows that the quantized mean squared error (QMSE), quantized cross-entropy (QCE), and quantized KL-Divergence (QKL-Div) (defined in Sec. 4) of each layer's activations accumulate much more during forward propagation of quantized DWSCNNs and lead to noticeably larger degradation compared to regular CNNs. Thus, indicating that there is greater error propagation and shifting distributional dynamics as information propagates through each layer. Furthermore, DWSCNNs appear to have much larger variation in quantization behaviour depending on the method of random weight initialization used. These observed phenomena would explain why channelwise quantization [14] and methods such as [21, 18, 24] that decrease distributional mismatch can provide impressive improvements on MobileNets quantization.

Utilizing fine-grained analysis enables a detailed view of the multi-scale distributional dynamics of our CNN architectures. Thus, facilitating better understanding of how CNN design choices affect the final trained distributions of weights and activations and the complex interactions between each layer's feature mappings and quantization noise. Tracking the layerwise QMSE captures the spatial-channel structure of accumulating quantized activation errors and helps us understand how these errors propagate through the network. Meanwhile, QCE/QKL-Div quantifies the degree of distributional shift and change in distribution dynamics of a network's representations when quantization noise is introduced. Analyzed together, we can gain a deeper understanding of the complex system dynamics involved in CNN quantization. We hope that these insights can lead to further innovative strategies for reducing and compensating for such distributional dynamics changes and improve post-training quantization for mobile deployment.

2. Motivations and Related Work

2.1. Efficient CNN Architectures via Depthwise Separable Convolutions

Efficient CNN architecture design is now a well-established field with several works [9, 22, 32, 2, 10] proposing various design patterns for factorizing convolution layers and reducing the computational load of inference. A primary tool for reducing the number of parameters and multiply-accumulate (MAC) operations in a CNN is depthwise-separable (DWS) convolution. Architectures such as MobileNets [9, 22, 8], ShuffleNets [32] and FBNets [27] make heavy use of DWS convolution to achieve state-of-the-art accuracy for low-power/efficient

CNNs. Depthwise separable convolution factorizes regular or "dense" convolution into a $K \times K$ depthwise convolution (that is, each convolutional kernel of size K is only applied to a single input channel) for low-dimensional feature extraction and a "pointwise" convolution (a dense convolution with 1×1 kernels) for mixing channel information. As mentioned in [9], this leads to between 8 to 9 times reduction in MACs for a 3×3 DWS convolution compared to its regular/dense counterpart. Furthermore, with publicly available ImageNet-trained checkpoints of the MobileNets "model family," their efficient, generalized features can be leveraged for various application-specific tasks via transfer learning. In this way they can be used "off-the-shelf" for finetuning with much less training overhead compared to training accurate models from scratch. However, even though MobileNets can already run incredibly fast with floating point (fp32) operations on a mobile CPU, further storage and power reductions can be gained if they are converted for 8-bit fixed point integer (quint8) operation and run on low-power, parallelized hardware such as a digital signal processor (DSP). Consequently, creating robust models that maintain accuracy during quantized inference has been a growing area of research with particular interest in improving the robustness of compact models like MobileNets.

2.2. Fixed-point Quantization For Efficient Mobile Inference

In conjunction with efficient CNN architecture design, low bit-width (16 bits and below, though most commonly 8 bits), fixed point quantization has enabled highly parallelized processors such as DSPs to run fast, low-power inference entirely with integer arithmetic. These methods [12, 19, 4] project the neural network weights and activations of each layer onto a low-dimensional, discretized space while minimizing loss of information. However, the noise induced by quantization error has complex interactions with the weights and activations of each layer and its impact on CNN output behaviour can be difficult to quantify. Thus, it can often be hard to predict which CNN architectures will quantize well (ie. are "quantization friendly") and are suitable for deployment.

Given the problem of quantization robustness, various research works [12, 13, 21] explore methods to increase the robustness of models to quantization noise. Methods such as quantization-aware training (QAT) [12] and trained quantization thresholds (TQT) [13] make use of simulated quantization and the straight-through estimator (STE) to train the network to adapt to quantization noise. Note that since these methods simulate quantization, the training is bit-width specific and a network must be retrained if one were to adapt a model for a different bit-width. Furthermore, quantization training can require hyperparameter tuning of its own, thus extending the design cycle. Oftentimes, the choice of how to optimize a model for quantization is based on available tools and trial-and-error. While many of the devised "quantization fixes" have shown remarkable results [21, 20, 13], they aren't necessarily guaranteed to transfer across applications. For example, image reconstruction and other continuous value prediction/regression tasks may have a much lower error tolerance than classification. Thus, we seek to use a systematic approach which can help delineate the various factors affecting quantization robustness. In this way, we can make more informed choices on how to improve the quantized behaviour of our CNN models.

[23, 18] perform a layerwise analysis of the signal-to-quantization-noise-ratio (SQNR) in a CNN. They use SQNR to estimate the amount of useful information passing from layer to layer in a quantized CNN and seek to maximize it. [23] analyzes layerwise SQNR to identify architectural choices that were hurting the quantized performance of MobileNets-V1. Similarly, we would like to use layerwise analysis to better understand the poor quantization of MobileNets and other DWSCNN models.

In a newer area of "robust quantization" works, the authors of [1, 24] seek to train models that are generally robust to quantization noise (using L_1 gradient regularization and a kurtosis-based weight regularizer (KURE) respectively) such that they can be easily quantized at varying bit-widths using simple post-training quantization methods. These kinds of models are trained to be robust to a broad range of perturbations that may be induced by various uniform quantization settings such that they can be smoothly deployed to quantized inference environments without further, potentially costly, quantization-based optimizations. Drawing on the ideas of [23, 1, 24] we hope to gain better insight on how different parametrizations of CNN can be constructed that tend towards learning quantization-robust solutions. In our work, we will focus on 8-bit uniform quantization described in [12] as it has become the most common method adopted in industry for mobile devices.

3. Experiments

Since we want to explore quantization results of both "official" MobileNets architecture and DWSCNNs in general, we run multiple trainings with the CIFAR-10 dataset. In this study, the detailed multi-scale distribution dynamics analysis on the variety of quantized network architectures was conducted on CIFAR-10 as its smaller size allows for a broader ablation study given resource constraints. Thus, we can quickly do a systematic comparison across multiple DWS convolution based architectures. Due to the much smaller 32×32 images of CIFAR-10, we could not use the exact same MobileNets-V1 architecture as reported in [9]. The main architectural differences are as follows:

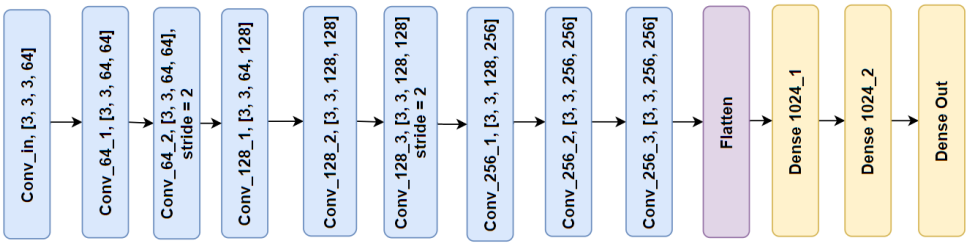

Figure 3. **Simple ToyNet Macroarchitecture**. For our ablation study we define a simple macroarchitecture (ie. input/output channels of each layer, stride, kernel size). We then ablate through a few hyperparameter settings that might affect the final layerwise distributions. Shape of weight tensors for regular convolution is in square brackets. For DWS-ConvNets, we replace regular convolution with DWS convolution while preserving input/output dimensions.

Network Architecture	FP32 Acc (%)	QUINT8 Acc (%)	QMSE	QCE	QKL-Div	Percent Acc Decrease
MobileNet No Gamma HeNorm	78.63	65.92 +/- 5.76	0.0266 +/- 0.011	1.08 +/- 0.45	0.76 +/- 0.43	16.16 +/- 7.32
MobileNet No Gamma GlorotUni	78.73	71.88 +/- 1.79	0.0186 +/- 0.0029	0.77 +/- 0.065	0.45 +/- 0.065	8.70 +/- 2.28
MobileNet With Gamma HeNorm	78.65	66.11 +/- 4.51	0.0278 +/- 0.0076	1.17 +/- 0.40	0.84 +/- 0.37	15.95 +/- 5.74
MobileNet With Gamma GlorotUni	78.46	65.87 +/- 2.36	0.0273 +/- 0.0046	1.05 +/- 0.13	0.72 +/- 0.13	16.05 +/- 3.01
DWS Conv No Gamma HeNorm	79.48	65.89 +/- 8.29	0.0236 +/- 0.012	1.03 +/- 0.33	0.60 +/- 0.33	17.10 +/- 10.43
DWS Conv No Gamma GlorotUni	78.63	68.93 +/- 2.90	0.0187 +/- 0.0036	0.88 +/- 0.11	0.48 +/- 0.11	12.34 +/- 3.69
DWS Conv With Gamma HeNorm	80.16	70.72 +/- 2.91	0.0187 +/- 0.0040	0.84 +/- 0.095	0.45 +/- 0.095	11.78 +/- 3.63
DWS Conv With Gamma GlorotUni	78.17	45.43 +/- 20.53	0.0592 +/- 0.037	2.51 +/- 1.34	2.04 +/- 1.34	41.88 +/- 26.26
Regular Conv No Gamma HeNorm	87.27	78.29 +/- 2.67	0.0170 +/- 0.0045	0.66 +/- 0.13	0.46 +/- 0.13	10.29 +/- 3.06
Regular Conv No Gamma GlorotUni	86.63	83.32 +/- 0.75	0.00764 +/- 0.0016	0.37 +/- 0.044	0.18 +/- 0.044	3.83 +/- 0.87
Regular Conv With Gamma HeNorm	88.01	80.26 +/- 0.77	0.0152 +/- 0.0013	0.55 +/- 0.022	0.38 +/- 0.022	8.80 +/- 0.88
Regular Conv With Gamma GlorotUni	86.58	81.15 +/- 2.17	0.0113 +/- 0.0044	0.46 +/- 0.11	0.26 +/- 0.11	6.27 +/- 2.51

Table 1. Detailed quantization results for CIFAR-10 networks. Quantization results reported as mean and standard deviation across five quantization trials. The output QMSE, QCE and QKL-Div of DWS convolution based networks are noticeably higher than regular CNNs.

- We do not downsample with stride = 2 at the first DWS-Conv block with output channels 128
- We do not downsample with stride = 2 at the first DWS-Conv block with output channels 1024
- Thus, the input tensor shape to Global Average Pooling layer is $4 \times 4 \times 1024$ rather than $1 \times 1 \times 1024$

In addition to directly comparing regular convolution vs. depthwise separable convolution, we also performed additional experiments with ResNet-34 using the same set of experiment conditions (ie. training hyperparameters, weight initializer choices, use of BatchNorm γ-scaling). For ResNet-34, the CIFAR-10 specific architectural modifications, quantization results, and layerwise plots can be found in supplementary materials. We refer readers to [9, 7] for original architecture details. We tried to preserve some of the overall topologies of MobileNets-V1 (eg. downsampling after the first convolution layer, downsampling when output channels increase etc.) while creating an architecture that could get reasonable results. We tried our best to recreate the training process of MobileNets. All experiments were conducted using Tensorflow 1.15. The optimizer, hyperparameters, data augmentations etc. are as follows:

- RMSProp. Momentum and decay = 0.9, epsilon = 1.0
- Training batch size = 128
- Logging and quantization batch size = 800
- Number of epochs = 200
- Initial learning rate 0.045. Scaled by 0.97 each epoch.
- Data augmentation: random rotation, width and height shift, zoom, random horizontal and vertical flip

While the macroarchitecture and the above listed hyperparameters were fixed, we also tried comparing the effect of weight initializaton method (Glorot Uniform [3] vs He Normal [6]) and use of BatchNorm scaling (ie. we compared applying BatchNorm with and without γ parameter). In [30], the authors show that BatchNorm and weight initialization methods can have a significant effect on quantization results and so we wanted to observe the effects they may have had on MobileNets-V1. Some hyperparameter choices were slightly adapted for CIFAR-10 such as number of training epochs, learning rate decay schedule etc. We decided to train for longer and decay the learning rate a little bit slower based on similar training setups for CIFAR-10 experiments such as in [31, 7]. For the DWS-ConvNets and Regular-ConvNets, we use the architecture described in Figure 3. Dropout is used in between all dense layers with a keep-probability of 0.5. Once trained, we follow the model analysis described in [30, 29] and observe the statistics of the trained weights, activations, and BatchNorm folded (BN-Fold) weights[1] of each layer (see Eq. 1). Thus,

[1]Where γ is batchnorm scaling parameter, w and w_{fold} are weights and BN-Fold weights respectively, $EMA(\sigma_B^2)$ is variance statistics of the given layer collected during training, ϵ is a small constant for numerical stability

Network Architecture	FP32 Acc (%)	QUINT8 Acc (%)	QMSE	QCE	QKL-Div	Percent Acc Decrease
MobileNet-V1-1.0-224	71.04	3.00 +/- 0.22	7.95E-4 +/- 8.7E-6	7.46 +/- 0.13	6.47 +/- 0.13	95.78 +/- 0.31
VGG-19	71.00	64.9 +/- 0.31	9.70E-5 +/- 4.1E-6	1.50 +/- 0.019	0.433 +/- 0.019	8.58 +/- 0.43

Table 2. Detailed quantization results for VGG-19 and MobileNet-V1-1.0-224 trained on ImageNet. Quantization results reported as mean and standard deviation across three quantization trials. While the scale of QMSE is different from CIFAR-10 due to the softmax distribution being computed over 1000 classes, we observe similar trends of greater accumulated error and distributional shift in MobileNet.

the logging and quantization batch size refers to the batch size used for computing the activation stats.

$$w_{fold} = \frac{\gamma w}{\sqrt{EMA(\sigma_B^2) + \epsilon}} \tag{1}$$

Additionally, we also define QMSE, QCE, and QKL-Div as the mean squared error, cross-entropy, and KL-Divergence between the fp32 model/hidden-layer outputs and the dequantized quint8 model/hidden-layer outputs respectively. QMSE quantifies the average distance/error whereas QCE and QKL-Div are able to capture differences in the output distribution shapes. Thus, by analyzing these hidden layer output and model output statistics, we can observe the accumulated quantization errors as well as the shifting distribution dynamics at different scales. Note, the QCE/QKL-Div of hidden layers is computed based on the distribution of activations aggregated over an entire batch whereas the QCE/QKL-Div of the model output is computed as the mean cross-entropy/KL-divergence between the softmax output of the fp32 model and quint8 model (ie. the same mean cross-entropy loss computation that would be performed for classification training). We use natural log for all entropy calculations so as to match and compare with the cross-entropy loss observed during training. Thus, for the layerwise quantization error analysis, QCE/QKL-Div illustrate the distributional dynamics of each layer's representations under quantization and how they might change during quantized information propagation through each layer. To compute QCE/QKL-Div, we collect histograms of each layer's activations for the quint8 and fp32 model. The approximate discrete distributions are then used for computing QCE and QKL-Div.

$$average\_precision = \frac{1}{K}\sum_{i=1}^{K}\frac{range_i}{range_{tensor}} \tag{2}$$

For layerwise statistics, we are primarily concerned with the range and average precision[2] of each layer (see Eq. 2, also defined in [21]). For activations, we perform percentile clipping to get the range. However, this percentile clipping slightly differs from true percentile clipping. Instead, we follow the method in Tensorflow Graph Transform tool [25] for min/max percentiles since we use it for creating quantized inference graphs. We use min/max percentile of 5% for both stats logging and activation quantization during inference.

For quantization, we perform 5 trials with randomly sampled training batches for calibrating activation ranges. This is to measure the variation in quantized performance caused by different calibration sets. For each trial, we record the activation statistics of each layer[3] to see how these distributions change and how they might correlate with quantization accuracy.

4. Results

In Table 1 we compare a few different testing metrics for the trained models. In terms of accuracy, we look at floating point accuracy, quantized accuracy, and relative/percent accuracy decrease (ie. change in accuracy over fp32 accuracy). As described in Sec. 3, we also observe the QMSE, QCE, and QKL-Div between the fp32 model output and quint8 model output. We see that on average, MobileNet-V1 and DWS-ConvNets experience much larger degradation in performance under quantization compared to the Regular CNNs. This is demonstrated by the much larger average percent accuracy decrease ($13.0 - 32.09$ vs. $6.12 - 9.22$), mean QMSE ($0.0186 - 0.0592$ vs. $0.00764 - 0.0170$), mean QCE ($0.77 - 2.51$ vs. $0.37 - 0.66$), and mean QKL-Div ($0.45 - 2.04$ vs. $0.18 - 0.46$). Furthermore, the depthwise-separable based CNNs seem to have much larger variation in quantization behaviour depending on random weight initialization method. In our ResNet-34 experiments, we observed results similar to the Regular-Conv networks (see supplementary materials). This is rather interesting since we had initially hypothesized that the residual learning block might enable learning of more compact distributions. Thus, reducing quantization noise/errors.

We now move onto a fine-grained analysis of the distributional dynamics of each network type. Coupled with the results observed in Table 1, we can gain a deeper understanding of how each layer's distributions interact with quantization and the resulting accumulation of quantization errors (QMSE) and distributional shifts (QCE/QKL-Div).

5. Discussion

To analyze the layerwise distributions of the different networks, we collected stats on the range and average pre-

[2]Where $range_i$ is range of $channel_i$ of convolution weights, $range_{tensor}$ is range of convolution weight tensor, K is number of channels/filters

[3]For activations, we can directly access range from the quantized graph

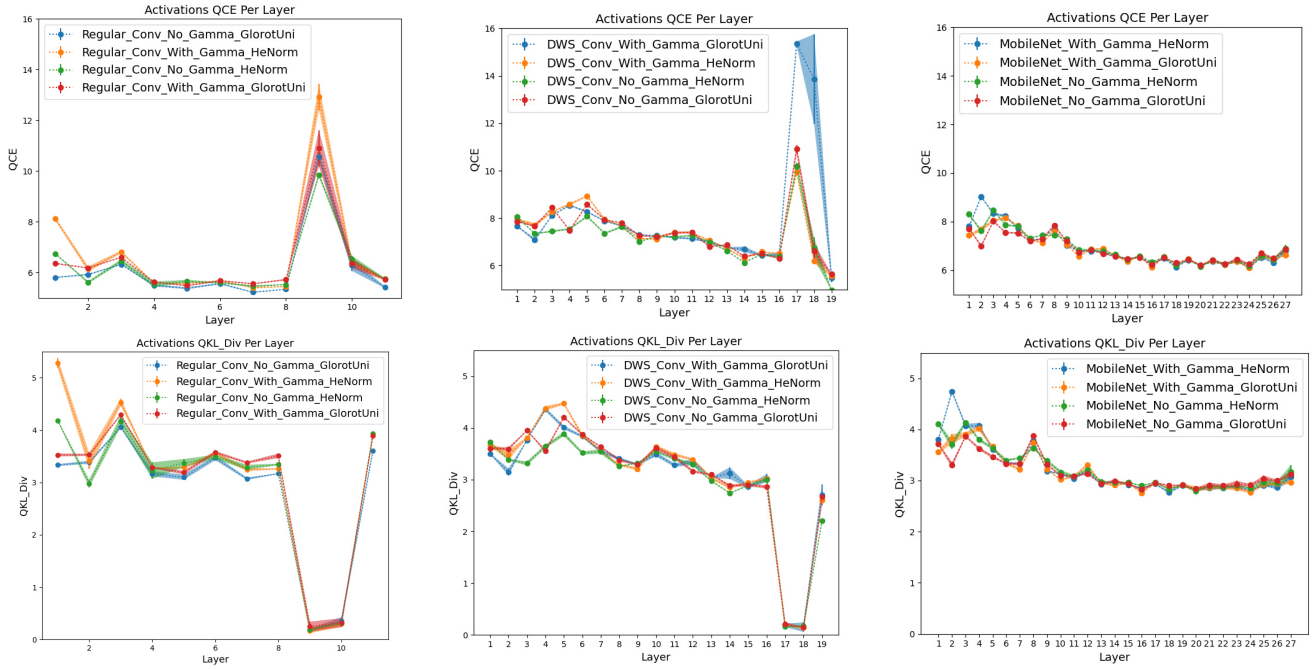

Figure 4. Layerwise QCE (top row) and QKL-Div (bottom row) for all trained networks. Regular CNN (left), DWS-ConvNet (center), and MobileNets-V1 (right). The drop in QKL-Div might explain why, despite significantly accumulatd QMSE, some networks are able to recover a relatively lower QMSE. The solid line represents the average values across 5 quantization trials and the shaded region is the standard devation.

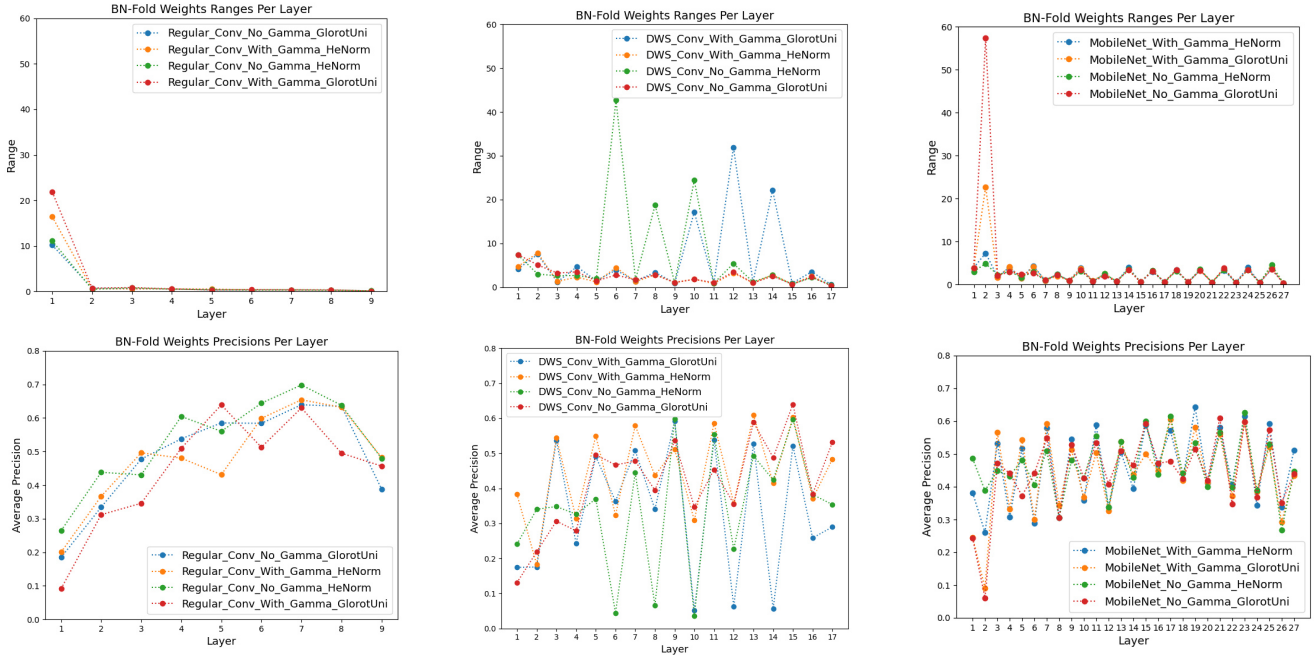

Figure 5. Layerwise BN-folded weights dynamic range (top row) and average precision (bottom row) for all trained networks. Regular CNN (left), DWSCNN (center), and MobileNets-V1 (right). The fluctuating range and average precision help explain the quantization degradation of DWSCNNs.

cisions for the weights, BN-Folded weights, and activations (see Figures 1, 5). For the sake of space, we have omitted most of the plots related to the weights distributions since it is the BN-Folded weights that will get quantized. However, comparing the distributions of weights before and after BN-Folding can reveal interesting insights on how each layer is being scaled and the resulting distributional shift. We also left out the layerwise activations plots as they follow similar trends to the BN-Folded weights. Detailed results and figures are included in supplementary materials. As men-

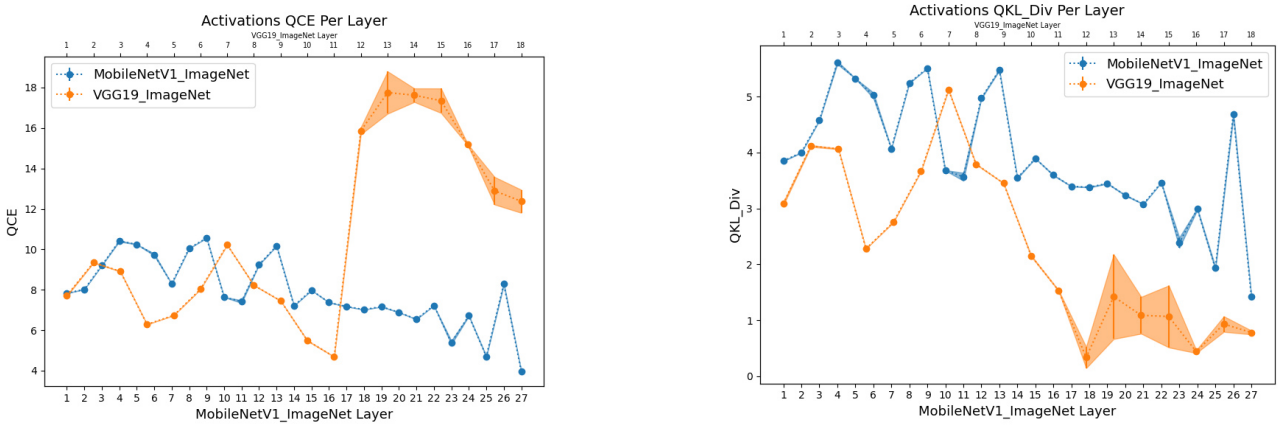

Figure 6. Layerwise quantization of VGG-19 and MobileNet-V1 on ImageNet. Layerwise QCE (left), Layerwise QKL-Div (right).

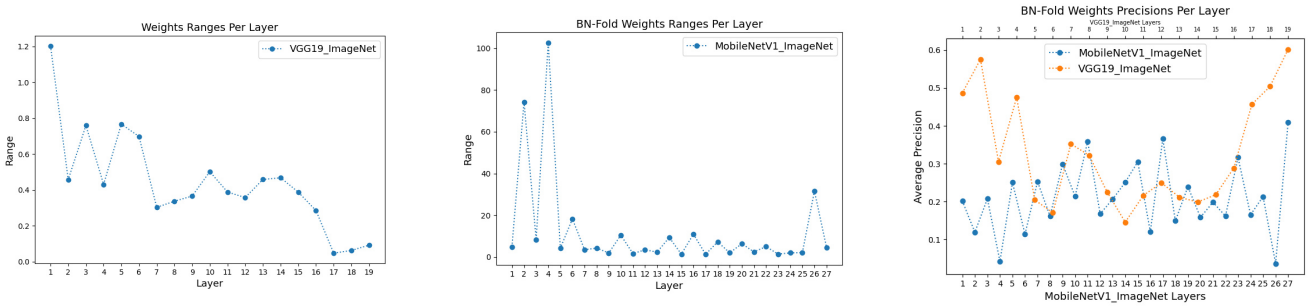

Figure 7. Layerwise weights/BN-Folded weights range of VGG-19 (left) and MobileNet-V1 (center) on ImageNet. Layerwise average precision of weights/BN-Folded weights of VGG-19 and MobileNet-V1 on ImageNet (right). Note the vastly different scales for weights ranges.

tioned in [29], the average precision gives a measure of how well the layerwise quantization encodings represent the information in an individual channel. If the precision is low, it is possible that the quantization noise may "wash out" the information of the given individual channel. Thus, range and average precision capture information about the distributions at both a layerwise and channelwise scale. Together, they give a picture of the magnitude of quantization noise, and the potential interactions of information in individual channels with the quantization noise.

We can see in the plots that the DWS convolution based architectures have significantly fluctuating ranges and average precisions. The sharply fluctuating plots illustrate the poorly behaved distributions of weights and activations that DWSCNNs tend to learn. Most notably, at depthwise convolution layers (even-numbered layers in the plots), the dynamic range peaks while average precision simultaneously drops. Suggesting that features extracted by the depthwise convolution will have significantly more quantization noise compared to their regular convolution counterparts. Low average precision suggests that the channelwise connection sparsity of depthwise-separable convolutions could be a detriment to quantized behaviour, possibly leading to learned distributions with low inter-channel correlation. Thus, causing tensorwise quantization to be a non-representative mapping of the weights and activations into discretized space.

Besides examining the layerwise distributions, we can directly look at our layerwise quantization noise statistics to better understand how the observed distributional dynamics manifest in the quantized behaviour. In Figures 2, 4 we can observe the interactions between each hidden layer's output activations and the noise induced by uniform 8-bit quantization. It is immediately apparent that QMSE accumulates significantly more in the DWS-ConvNets and MobileNets-v1.[4] While QMSE captures the 2D-structure of the accumulated quantization error, QCE and QKL-Div describe the distributional shift induced by quantization and that accumulation during information propagation. From an information theoretic perspective, we could interpret them together as the amount of information in the hidden representations and how well the 8-bit encoding is representing that information. Considering the DWS-ConvNet and Regular-ConvNet plots, the peaking QCE that coincides with a drop in QKL-Div could be interpreted as maximal information being transmitted by later layers (entropy of the distribution) and a fairly representative mapping of the hidden distribution from continuous fp32 space to discrete

[4] As noted in Figure 2, some outliers were removed from the QMSE plots. Full layerwise QMSE plots in supplementary material

quint8 space. This drop in QKL-Div might explain why despite relatively high peaks in QMSE, most of the DWS networks can "recover" and return to a lower QMSE at the output. The drop in QKL-Div would imply that the overall distribution of activations is preserved and consequently the relevant information is still passed onto the following layers. By contrast, the quantization noise statistics in MobileNet-v1 demonstrate a steadier level of distributional shift likely due to the deeper architecture. However, MobileNets still suffers from fluctuating layer dynamics and distributional mismatch leading to an overall larger accumulation of QMSE. Thus, it is not able to quantize as well as the Regular-ConvNets.

Based on these insights, we believe there are a few approaches that could be explored to improve quantization by either reducing accumulation of errors or improving the alignment of layerwise and channelwise distributions. Minimizing layerwise QMSE in addition to quantized output errors could be beneficial at lower bit-widths (eg. 4-bit, 2-bit quantization) where the backpropagation of STE's biased gradient estimate from the output layer can lead to increasingly inaccurate gradient estimates. The QMSE loss can act as a closer, more accurate gradient feedback, especially for earlier layers. With respect to minimizing misalignment of distributions, finetuning with weight normalization or a regularizer that better aligns the weight distributions could help reduce quantization distributional shifts and make the layer/tensorwise quantization mapping more representative of the information in each individual channel.

6. ImageNet Analysis

To study the distributional dynamics on more complex deep neural network architectures and on a much larger dataset, we conduct a similar analysis on VGG-19 and MobileNet-V1 trained on ImageNet.[5] We follow the same quantization procedures as described in Section 3. However, due to time and compute constraints, we limited the number of quantization trials conducted to 3. We also increase the size of the quantization/activation logging set to 2000. For layerwise quantization and activation stats, we iterate through 2000 images with batches of 50 images and compute the mean due to RAM constraints.

Table 2 shows the quantized output results. As expected, the quantized accuracy drop for MobileNet is catastrophic. However, the relative degradation of VGG-19 (8.58% relative accuracy decrease) stays fairly consistent with CIFAR-10 results. Upon comparing the range and average precision of the BN-Folded weights[6] of the two networks (see Figure 7) we see that the issue of mismatched distributions combined with large, spiking dynamic ranges is even more pronounced in MobileNet trained on ImageNet. As DWS convolution decouples the convolutional channels from each other, we believe the diversity of ImageNet data significantly aggravates the misalignment of channelwise distributions vs. layerwise distributions and as a result leads to a significant loss of information. While VGG-19 also has a decrease in average weight precision near the "middle" of the network, its dynamic ranges are about an order of magnitude smaller and the average precision is still generally better.

We see in Figure 6 that the higher layerwise QKL-Div for MobileNets indicate much greater quantized distributional shift. Due to differences in preprocessing,[7] the layerwise QMSE and layerwise activations data are on completely different scales (VGG-19 has much larger ranges). Thus, making it hard to compare these values.[8] However, similar to CIFAR-10 results, MobileNet trained on ImageNet had much larger output QMSE, QCE, and QKL-Div. Interestingly, this shows how large dynamic activations range do not automatically translate to poor quantization performance. Instead, accumulated errors, distributional shifts and the multi-scale distributional dynamics of each layer should be observed together as a whole to better understand the quantization dynamics at play.

7. Conclusion

We perform a systematic ablation study with fine-grained analysis to understand why DWSCNNs seem to quantize more poorly than regular CNNs on average. We find that this appears to be inherent to depthwise separable convolutions owing to the fact that depthwise convolutions tend to learn representations with greatly fluctuating dynamic ranges and significant intra-layer distributional mismatch. Analyzing CNN quantization through the lens of multi-scale distributional dynamics, we observe that depthwise convolution based CNNs suffer from larger accumulation of quantization errors and distributional shift. We believe these insights point to potential new ways to mitigate such distributional dynamics changes and improve robustness of post-training quantization for efficient depthwise-convolution based CNNs. As MobileNet-V2 also quantizes poorly, future work involves extending our analysis to the inverted bottleneck residual blocks described in [22]. Furthermore, we would also like to try setting certain layers of a quantized network to floating point operation and observing the correlation of layerwise QMSE/QCE/QKL-Div with quantized output behaviour.

[5] We leveraged ImageNet-trained fp32 checkpoints of VGG-19 and MobileNetV1-1.0-224 from [26].

[6] The VGG-19 checkpoint does not use BatchNorm. Thus we compared VGG-19 weights to MobileNet BN-Folded weights.

[7] MobileNet scales images to the range $[-1, 1]$ whereas VGG only zero-centers each RGB channel without scaling

[8] The activations and QMSE plots are included in supplementary materials

References

[1] Milad Alizadeh, Arash Behboodi, Mart van Baalen, Christos Louizos, Tijmen Blankevoort, and Max Welling. Gradient ℓ_1 regularization for quantization robustness, 2020.

[2] Amir Gholami, Kiseok Kwon, Bichen Wu, Zizheng Tai, Xiangyu Yue, Peter H. Jin, Sicheng Zhao, and Kurt Keutzer. Squeezenext: Hardware-aware neural network design. *CoRR*, abs/1803.10615, 2018.

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. JMLR Workshop and Conference Proceedings.

[4] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir D. Bourdev. Compressing deep convolutional networks using vector quantization. *CoRR*, abs/1412.6115, 2014.

[5] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations (ICLR)*, 2016.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[8] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. *CoRR*, abs/1905.02244, 2019.

[9] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[10] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[12] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *CoRR*, abs/1712.05877, 2017.

[13] Sambhav Jain, Albert Gural, Michael Wu, and Chris Dick. Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 112–128, 2020.

[14] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *CoRR*, abs/1806.08342, 2018.

[15] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *CoRR*, abs/1608.08710, 2016.

[16] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[17] Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l0 regularization. *CoRR*, abs/1712.01312, 2017.

[18] Eldad Meller, Alexander Finkelstein, Uri Almog, and Mark Grobman. Same, same but different: Recovering neural network quantization error through weight factorization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4486–4495. PMLR, 09–15 Jun 2019.

[19] Daisuke Miyashita, Edward H. Lee, and Boris Murmann. Convolutional neural networks using logarithmic data representation. *CoRR*, abs/1603.01025, 2016.

[20] Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization, 2020.

[21] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. *CoRR*, abs/1906.04721, 2019.

[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[23] T. Sheng, C. Feng, S. Zhuo, X. Zhang, L. Shen, and M. Aleksic. A quantization-friendly separable convolution for mobilenets. In *2018 1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2)*, pages 14–18, 2018.

[24] moran shkolnik, Brian Chmiel, Ron Banner, Gil Shomron, Yury Nahshan, Alex Bronstein, and Uri Weiser. Robust quantization: One model to rule them all. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5308–5317. Curran Associates, Inc., 2020.

[25] Tensorflow. Tensorflow graph transform tool github.

[26] Google Tensorflow. Tensorflow-slim image classification model library, May 2020.

[27] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[28] Huanrui Yang, Wei Wen, and Hai Li. Deephoyer: Learning sparser neural network with differentiable scale-invariant sparsity measures. In *International Conference on Learning Representations*, 2020.

[29] Stone Yun and Alexander Wong. Factorizenet: Progressive depth factorization for efficient network architecture exploration under quantization constraints, 2020.

[30] Stone Yun and Alexander Wong. Where should we begin? a low-level exploration of weight initialization impact on quantized behaviour of deep neural networks, 2020.

[31] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

[32] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *CoRR*, abs/1707.01083, 2017.