Exploring Neural Networks Quantization via Layer-Wise Quantization Analysis
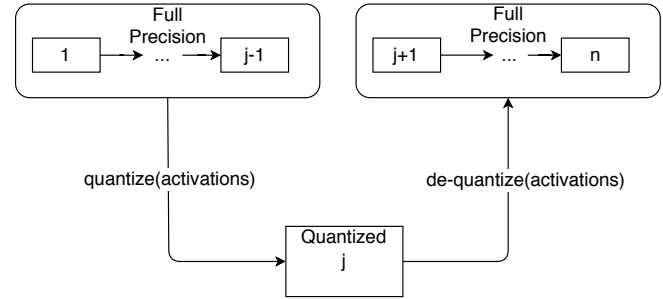
Shachar Gluska   Mark Grobman

Hailo Technologies

{shacharg, markg}@hailo.ai

Abstract

Quantization is an essential step in the efficient deployment of deep learning models and as such is an increasingly popular research topic. An important practical aspect that is not addressed in the current literature is how to analyze and fix fail cases where the use of quantization results in excessive degradation. In this paper, we present a simple analytic framework that breaks down overall degradation to its per layer contributions. We analyze many common networks and observe that a layer's contribution is determined by both intrinsic (local) factors – the distribution of the layer's weights and activations – and extrinsic (global) factors having to do with the the interaction with the rest of the layers. Layer-wise analysis of existing quantization schemes reveals local fail-cases of existing techniques which are not reflected when inspecting their overall performance. As an example, we consider ResNext26 on which SoTA post-training quantization methods perform poorly. We show that almost all of the degradation stems from a single layer. The same analysis also allows for local fixes – applying a common weight clipping heuristic only to this layer reduces degradation to a minimum while applying the same heuristic globally results in high degradation. More generally, layer-wise analysis allows for a more nuanced examination of how quantization affects the network, enabling the design of better performing schemes.

(a) Basic building block of layer-wise analysis
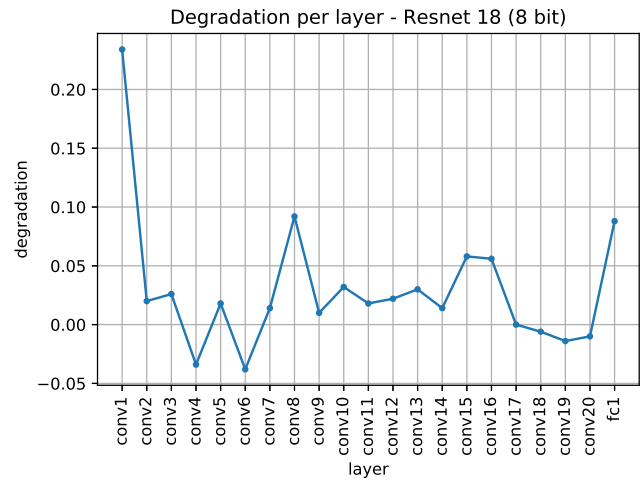
(b) ResNet18 on ImageNet-1K validation set

Figure 1. (a) The building block of our layer-wise analysis is the creation of a network with only a single quantized layer j. The input and output of the layer is quantized as well as the weights. The rest of the network is kept at full precision. (b) Layer-wise analysis applied to Resent-18 . This breakdown gives insight into which layers preform poorly under quantization

1. Introduction

Quantized neural networks are networks which have their weights and/or activations constrained to low-bit representation, usually 8-bit and below. Neural networks are resource intensive algorithms and quantization is an easy and efficient way to reduce both the computational and memory load of deployed networks. In recent years, quantization of deep neural networks has become ubiquitous: indeed, just 3 years ago the frontier for quantization was 8-bits for both activations and weights (Jacob et al., 2017), while state-of-the-art works operate at 4-bits and below (Banner et al., 2018; Uhlich et al., 2019; Fang et al., 2020; Jain et al., 2019; Stock et al., 2019; Dong et al., 2019a; Wang et al., 2018; Choukroun et al., 2019).

An existing gap in the literature is tools for quantization post-mortem. Once a quantization method has been applied it is evaluated solely on the basis of its degradation with respect to the full-precision network. We claim that an investigation into the sources of quantization degradation is valuable for (a) a better characterization of quantization methods in general and their fail cases in particular, (b) directing future research to address common fail-cases and (c) quick ad-hoc fixes of quantization problems in specific networks.

In this paper we take a step in addressing the above gap. We present ***layer-wise quantization analysis*** which is based on previously observed properties of quantization: The noise and degradation of individually quantized layers can be summed to get the overall noise and degradation of a fully quantized network (Zhou et al., 2017). Our method consists of iterating over the layers of a full precision network and at each step quantizing only a single layer and measuring the resulting noise at the output of the network. Given a quantization method and a network, we can quickly evaluate the layer-wise composition of the overall network degradation. The basic premise of this tool can be summarized by noting that for quantization the whole *is* the sum of its parts. The typical output of such an analysis is shown in Figure 1

In section 4 we explore the quantization of many neural networks through the prism of layer-wise analysis and observe some interesting phenomena – most of the degradation is usually caused by few layers. Interestingly, the common practice (Banner et al., 2018; Baskin et al., 2018; Esser et al., 2019; Yang et al., 2019; Dong et al., 2019b; Liu et al., 2020; Haroush et al., 2019; Nahshan et al., 2019) of refraining from quantizing the first and last layers doesn't seem warranted as these are seldom the worst offenders. The proposed method makes such analysis easy for a given quantized network and also significantly reduces the level of expertise required in order to achieve successful quantization as it highlights problematic layers on which the effort should to be focused.

In summary, the main contributions of this paper are:

- **Quantization post-mortem** – We develop a simple and efficient way of analyzing quantization degradation by breaking it down to layer-wise contribution. Such analysis can be used for better characterization of quantization methods and gaining insights about the mechanisms by which degradation occurs. Another important aspect of our method is that it enables practitioners without quantization expertise to quickly debug and fix poor quantization by focusing their efforts on specific layers.

- **Improving efficacy of existing methods** – We show

that applying some of the existing techniques in a global manner results in worse performance compared to applying them locally. Our method gives an efficient way by which to identify layers that require special treatment and allows targeted application. In section 4.2 we show that we can out-perform existing SoTA techniques by taking this approach.

2. Previous Work

Quantization has a rich taxonomy but can roughly be divided into two major branches: *Post-training quantization* (PTQ), where the quantization takes place directly on the original full-precision model and *quantization-aware training* (QAT) where training is used to offset the adverse effects of quantization. Post-training methods often try to apply some heuristic to each layer in order to reduce the quantization noise at the output of the layer. Common methods include clipping (Zhao et al., 2019; Banner et al., 2018), channel equalization via inversely-proportional factorization (Meller et al., 2019; Nagel et al., 2019) and bias-compensation (Finkelstein et al., 2019; Banner et al., 2018; Nagel et al., 2019). A shortcoming of these methods is that they use a local optimization of layer noise in order to achieve global performance optimization. A notable exception is the work of Zhou et. al. (2017) which derived a measure of the layer sensitivity to quantization via noise measurement at the output of the network. Their work was the first to decompose overall quantization into layer-wise contributions. In contrast to our work, they use the decomposition in order to establish an analytic correspondence between output noise and degradation in classifiers. Heuristics for ranking layers based on sensitivity have also been used in the context of network compression where one looks at the optimal bit-assignment per layer. The layer bit-width is determined either explicitly by a measured metric (Zhou et al., 2017; Dong et al., 2019b; Cai et al., 2020) or implicitly via a constrained optimization process (Wang et al., 2018; Elthakeb et al., 2018) which is resource intensive as it involves re-training the network from scratch. Recently Uhlich et al. (2019), proposed a simpler method which does not require full re-training. These methods all rank layers based on their contribution of degradation, however, the ranking is also affected by how much can be gained from the compression of the layer. Compared to this methods, our ranking of layers is both more computationally cheap and more direct, not relying on any proxies. Recent works have also assumed individual layer contributions. Cai et al. (2020) employ a similar method to ours but using KL-divergence instead of noise and network performance. They use their method for mixed-precision quantization in a data-free settings whereas we use our method for analysis of quantization in any settings. Nagel et al. (2020) use a layer-local loss in their method and Hubara

et al. (2020) explicitly optimize over the sum of per-layer loss. Both of these methods are used for post-training optimization and not for analysis. A similar technique to ours was recently used Wu et al. (2020) to rank layers for exclusion when using partial quantization. This method therefore sidesteps the problem whereas our aim is to both improve full quantization performance and to explore degradation mechanisms. To the best of our knowledge, ours is the first work to use layer-wise quantization as an empirical analysis tool.

3. Methods

Here we present our method for layer-wise quantization analysis. We first show how to get the individual degradation of each layer. Then we show that, under practical settings, these contributions are mostly independent and thus can be summed to arrive at the overall network degradation when all layers are quantized. We discuss some limitations of our approach in section 4.3 and section 4.4.

3.1. Layer-Wise Quantization Analysis

Consider a neural network L with n layers. We topologically sort the layers and enumerate them $l_1, \ldots, l_n$. A layer is considered to consist of a typical operation (e.g. convolution, fully-connected, pooling) followed by a non-linearity (e.g. ReLU). The basic step of our method is the creation of a network with only a single layer, l_j, being quantized while the rest of the network is kept at full precision. The quantization of a single layers entails the following: the layer's weights are quantized as well as it's input and output activations. The output of the layer is then de-quantized and propagated onward. See Figure 1a for an illustration.

We use y for the full-precision network output and $\hat{y}_j$ for the output of the network with a quantized layer l_j. The full method consists of iterating over all layers and computing $\hat{y}_j$ for each layer. The evaluation is done over a dataset D. In this work we consider two metrics: the ℓ_2 norm of the quantization noise $||y - \hat{y}_j||_2^2$ and the performance degradation $H_D(y) - H_D(\hat{y}_j)$. Where $H_D(y)$ is the metric of the task performed by the network (e.g. top-1 accuracy for a classification network). The full method is described more formally in Algorithm 1.

Measuring the individual degradation of each layer is valuable in itself as it immediately shows which layers produce high degradation. It is not clear that we can say something meaningful about the overall network degradation without taking into account the interaction of quantized layers with each other, however, Zhou et al. (2017) previously established analytically that the overall noise in the output of a network can be approximated by summing together the output noises of individually quantized layers. The basic

---

Algorithm 1 Description of our method for measuring the individual contribution of each layer. For each layer we measure both the noise and degradation resulting from quantization of that layer.

Input: Dataset D, Evaluation function H, network L composed of n layers $l_1, l_2, \ldots, l_n$
Output: $\{noise_i | i \in \{1 \ldots n\}\}, \{perf_i | i \in \{1 \ldots n\}\}$
for $i = 1$ **to** n **do**
  $noise_i = \frac{1}{|D|} \sum\limits_{x \in D} ||y - \hat{y}_i||_2^2$
  $perf_i = H_D(y) - H_D(\hat{y}_i)$
end for

---

assumption underpinning their analysis is that (a) the quantization noises of each layer are independent, (b) quantization noises are small compared to activations and (c) quantization noise has zero mean. In practical settings, these assumptions hold well. Empirically, Zhou et al. (2017) show their results hold for noise when only the weights of the network are quantized, however, it is easy to see that their analytical analysis will hold also for the case when activations are also quantized and that the additive nature of noise should extend to degradation as well. To confirm this, we compare the degradation from the sum of individual layers contribution with the measured degradation when quantizing the whole network. The results are shown in Figure 2. We see that additivity holds well into the ∼10% degradation range which is much larger than the permissible degradation in practice. In BRECQ (Anonymous, 2021) it is suggested that the inter-layer dependence is not negligible. We note that they showed this when quantizing to 2-bits. Other recent works (Wang et al., 2020; Nagel et al., 2020; Hubara et al., 2020) show good results even when assuming inter-layer independence.

4. Experiments

In this section we experiment with layer-wise quantization and show its two main benefits – analysis of degradation errors and improving network quantization by applying layer-specific fixes. We first analyze the results of applying layer-wise analysis to a variety of ImageNet (Russakovsky et al., 2015) classifiers. We then show how the quantization of a given network can be improved using the layer-wise breakdown as a compass to pinpoint us to problematic layers.

Unless otherwise stated, we follow the post-training quantization scheme presented in (Jacob et al., 2017). Per-layer quantization is used for both weights and activations. Batch Normalization (Ioffe & Szegedy, 2015) layers are folded into the previous layers. While these are the settings of our experiments, our method is general and can be used both in channel-wise and in group-wise settings.
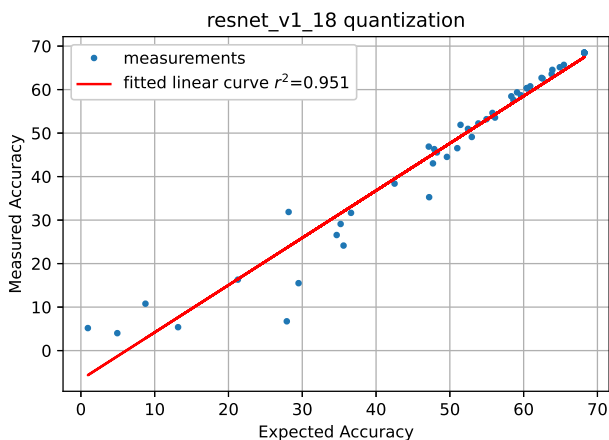
Figure 2. Expected accuracy vs the measured accuracy. Expected degradation is the sum of individual layer degradations. Measurements were done on ImageNet-1K validation set. Each data point is a different quantization of ResNet18. To induce large degradation we quantize a random subset of layers weights to 4 bits while the rest of the layers are quantized to 8 bits. Activations are always quantized to 8 bits. Moving more layers to 4 bits results in larger degradation. We see that the additivity assumption holds well even for very high degradation.

4.1. Analyzing ImageNet Classifiers

We apply our layer-wise analysis to several popular ImageNet classifiers: ResNet18 (He et al., 2015), MobileNet-V1-1.0 (Howard et al., 2017), Inception-V1 (Szegedy et al., 2014), ResNext26 (Xie et al., 2016), MobileNet-V2-1.0 and MobileNet-V2-1.4 (Sandler et al., 2018b). For each network, we analyze the degradation when quantizing each layer's weights between 3-8 bits. Activations are quantized to 8-bits. Degradation is measured over the full ImageNet validation set. The results are shown in Figure 3.

Looking at the results, we can make several observations. We see that some layers are very sensitive to quantization while others can be quantized to as little as 3-bits with little degradation. A natural question is whether a layer's sensitivity is determined by the network's architecture or is it the result of a specific training. When comparing MobileNet-V2-1.0 and MobileNet-V2-1.4 we see that they have almost identical profiles. These networks share the same architecture but differ in hyper-parameters and are the results of two different training sessions. This would seem to indicate that, under the same training scheme, the sensitivity of layers to quantization is strongly tied to network architecture. To examine whether the results will be similar even under different training settings, we apply Two-Step Equalization (Meller et al., 2019). Under equalization the network's output and architecture remain unchanged while the network parameters are changed. This simulates a different training scheme while controlling for the network's performance and architecture. As we can see in Figure 4, the equalized model has a different profile than the non-equalized baseline. This suggests that the training process also plays a role in the network's noise profile under quantization. The question of how much the quantization profile is affected by architecture vs training has important implications. Models in deployment are continuously trained and quantized. Knowing that repeated training session don't require changing quantization schemes can greatly simplify quantization flows, especially those that involve lengthy fine-tuning session – one should always focus on the layers that do the most harm. It is interesting to note that many works (Banner et al., 2018; Baskin et al., 2018; Esser et al., 2019; Yang et al., 2019; Dong et al., 2019b; Liu et al., 2020; Haroush et al., 2019; Nahshan et al., 2019) refrain from quantizing the first and last layers since it seems to be a commonly held belief that they are harder to quantize. We find that for classifiers the last layer is generally indifferent to quantization and with the exception of ResNet, the first layer is also not the most sensitive layer. Another salient feature is that the depthwise layers in the MobileNet architecture are more sensitive to quantization compared to standard conv2d layers, with the first depthwise layer being particularly finicky.

4.2. Improving Quantization

In this section we show how layer-wise analysis can be leveraged to improve the quantization of a network. To that end, We choose ResNext26 (Xie et al., 2016) which has a >1% degradation even when using SoTA post-training techniques such as Equalization (Meller et al., 2019) and IBC (Finkelstein et al., 2019). The results of running the analysis on the network are shown in Figure 5. We see that almost all of the degradation stems from a single layer – conv4. Since 8-bit quantization usually gives good results, it is expected that layer exhibiting large degradation will be rare. We now examine the problematic layer. Since quantization noise is the result of the trade-off between representation range and precision, excess degradation is commonly due to large ranges of kernel values. In Figure 6 we plot the weights histogram of conv4 and indeed we see significant outliers. At this juncture, it is reasonable to try and use a standard weight clipping technique such as SAWB* (Choi et al., 2018) on conv4 which results in near lossless quantization. We emphasize that this result is possible due to the *local* application of clipping, as applying SAWB globally degrades the network performance even below baseline as shown in Table 1. This example illustrates a general principle: quantization techniques can be beneficial locally but degrade performance when used globally. Since PTQ

---

*The results we report are not sensitive to the type of clipping used provided it is aggressive enough to handle the outliers
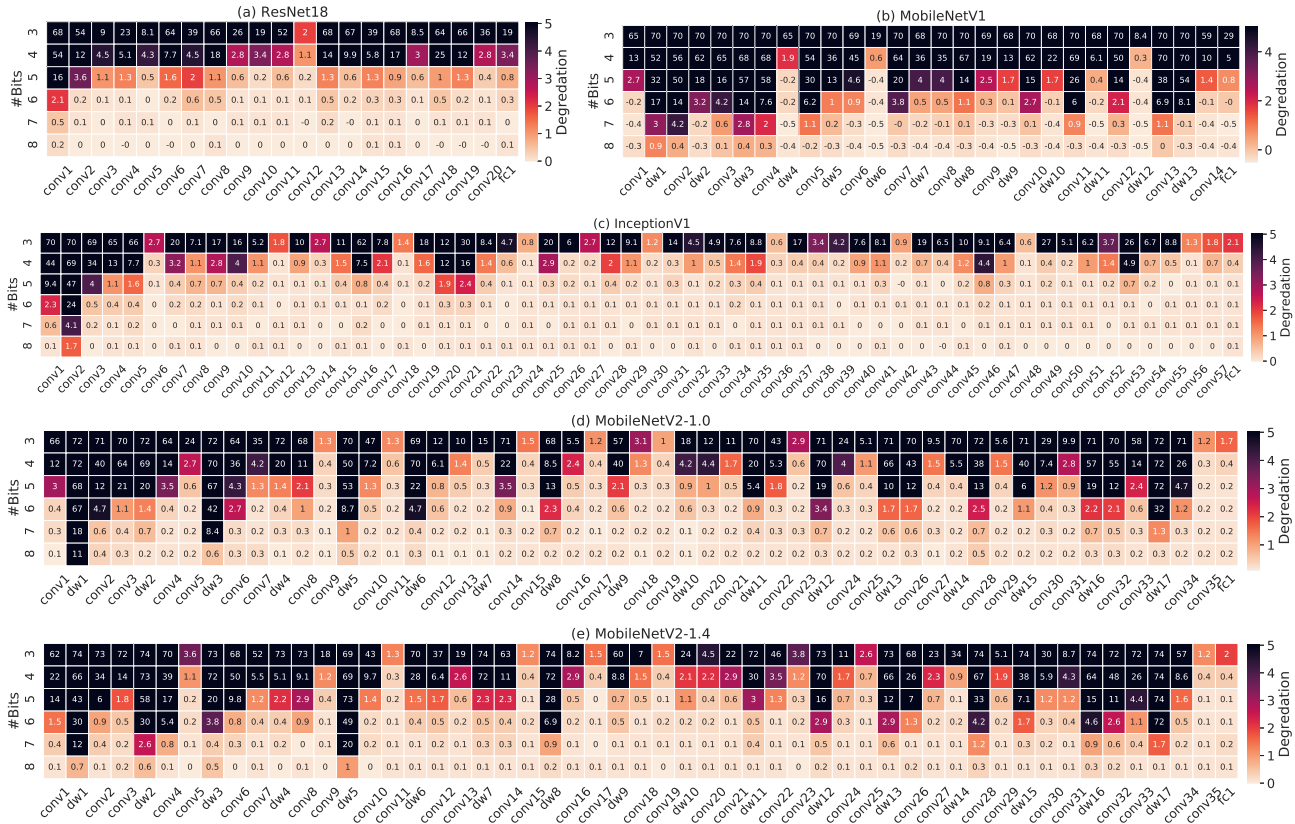
Figure 3. Layer-wise analysis of common networks. Each row is a single run of the tool with a different bit-width selected for the weights. Each column represents a layer of the network. The layers are topologically sorted. The annotation in each cell is the absolute top-1 degradation of the network when only that layer is quantized to a specific bit width. The color bar is saturated at degradation of 5% for better visibility.
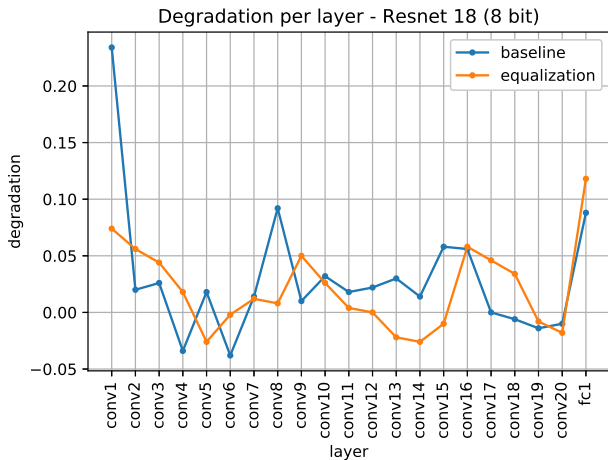

Figure 4. ResNet18 (ImageNet-1K) layer-wise degradation under equalization. The network architecture and full precision output is identical to the baseline model. The results however differ.

methods are inherently heuristic, the standard practice of evaluating them only on the merit of global application is bad practice; moreover, compound quantization methods which apply a larger-range of heuristics locally can give much better results.

4.3. Quantization-Aware Training

Our method was designed to be used in a post-training quantization setting. Networks which undergo quantization-aware fine-tuning (Jacob et al., 2017) via straight-through estimation can't be used with our method due to the fact that the full-precision network is no longer the appropriate reference. It is, however, still of interest to examine the results. We continue with the same network as in the previous section. The optimization target of the fine-tune process is *the quantized network* and so it doesn't act to reduce the noise in the full-precision network but rather to compensate for it in the quantized network. This is illustrated in Figure 7 where we see that the fine-tuned network has almost the same noise profile as the original net-
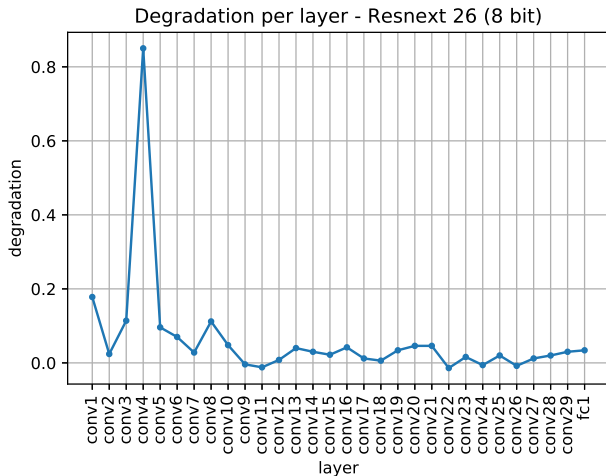
Figure 5. Layer-wise analysis of ResNext26 on ImageNet-1K validation set. Most of the degradation is caused by conv4
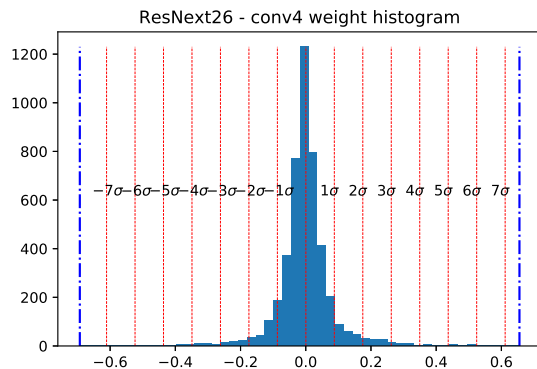
Table 1. Top-1 Accuracy for ResNext26 on ImageNet-1K validation set under different post-training methods for 8-bit quantization. The standard approaches still show high degradation. By using our layer-wise analysis we identify the problem and manage to get better results.

Method	Top-1 Accuracy
Full Precision	76.08
Naive Quantization	74.28
Equalization + IBC	75.00
SAWB globally	71.88
SAWB conv4$^\psi$	**75.91**

$^\psi$ Methods using our layer-wise breakdown

Figure 6. The weight distribution of conv4 in ResNext26. The quantization range is decided by the minimum and maximum values denoted by the dash-dot lines in the figure. We see there are outliers which drastically increase the dynamic range.

work, however, since the network was trained in the presence of noise, the performance of the fine-tuned network is *improved* when noise is introduced to the network by quantization.

The problem stems from the fact that the full-precision network and the optimization target of the training process are no longer one and the same. Our type of analysis only works well when the reference against which we compare the quantized network is the optimization target. Another way of expressing the above is that when the network is trained in the presence of a specific noise profile then the assumption of noise independence no longer holds. Thus, it is still possible to use our method to analyze fine-tuned models, but we must use quantization noise that is inde-

pendent of the network training. This is shown in Figure 8 when we take the full-precision model fine-tuned to 8-bit quantization and use 7-bits quantization instead. This above doesn't mean that layer-wise analysis is of no use when using quantization-aware training. The fact that the fine-tuning only compensates but does not reduce the noise, means that performing an analysis prior to lengthy training is still beneficial - problematic layers can be recognized and dealt with prior to training for improved performance.

4.4. Bias Shift

One of our underlying assumptions for noise additivity is that quantization noise has zero mean. Previous works (Finkelstein et al., 2019; Nagel et al., 2019) have shown that layers with small kernels (e.g. depthwise layers) receive a mean activation shift (Finkelstein et al., 2019) due to quantization. For these networks there is strong interaction between noises from adjacent layers, skewing the result of layer-wise analysis. For example, using layer-wise analysis on MobileNets (Howard et al., 2017; Sandler et al., 2018a) (Figure 3) which have many depthwise layers, shows that the linear additivity of quantization noise no longer holds. We note that we can still apply layer-wise analysis on MobileNets (section 4.1) for finding the individual degradation of layers, however, the sum of these individual contributions will under-estimate the overall network degradation.

5. Discussion and Future Directions

In this work, we presented a method for layer-wise analysis of post-training quantization methods. The use-case we presented in section 4.2 illustrates a general issue with advanced post-training quantization methods – they often
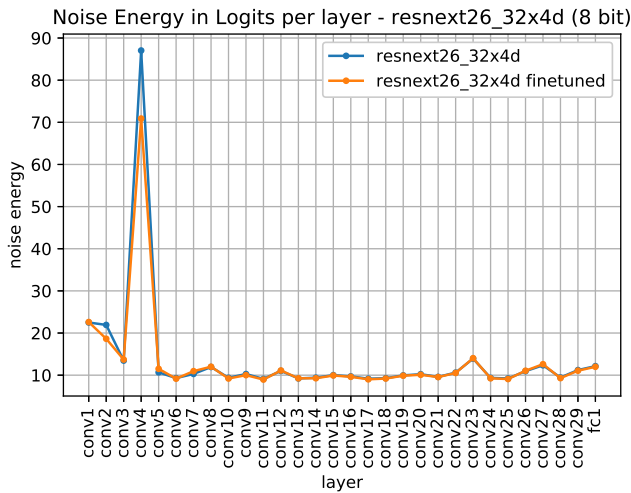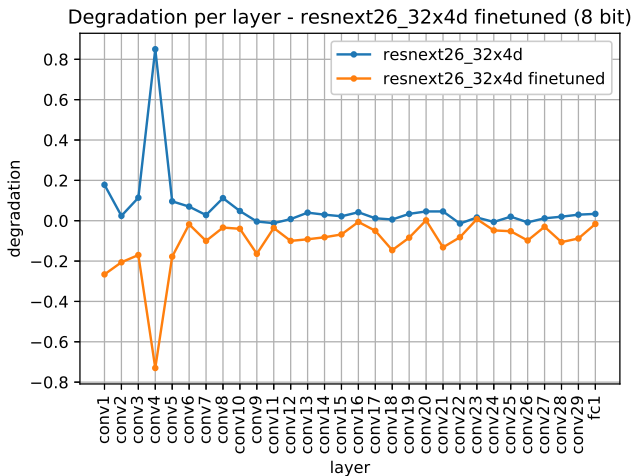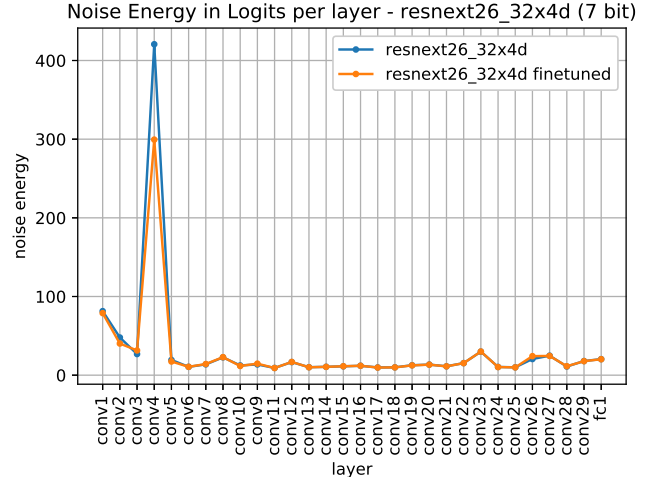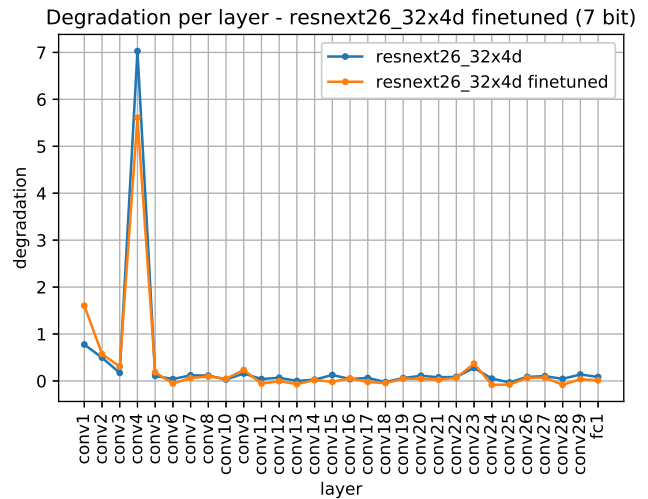
Figure 7. Layer-wise analysis of ResNext-26 on ImageNet-1K validation set before and after fine-tuning. The usual correlation between noise and degradation breaks down. (a) Noise energy has decreased in conv4 after fine-tuning, but is still high. The rest of the noise profile is similar. (b) The Fine-tuned network is showing improved accuracy when a layer is quantized. Using the full-precision network after fine-tuning as the reference network is no longer appropriate.

Figure 8. layer-wise analysis of ResNext-26 before and after fine-tuning to 8-bit quantization. The network is quantized 7 bits when performing the analysis to make the quantization noise independent of the 8-bit noise used for fine-tuning. We see that the correlation between noise and degradation is restored.

have some hyperparameter (e.g. clipping value) which is selected based on greedy optimization. Values that are optimal for certain layers can be detrimental to others to the point where overall network performance is degraded. Our tool directly measures the contribution of the layer to the overall degradation, taking into account both the layer location in the network as well as the whole network architecture. Once problematic layers have been identified it is easy to apply existing methods in a more precise fashion. We further claim that the difference between global and lo-

cal application of quantization methods is symptomatic of a larger issue which is the one-dimensional evaluation of quantization methods. As shown in section 4.1 using layer-wise analysis gives a more nuanced evaluation of the performance of quantization methods, allowing disambiguation between architecture-specific effects and general qualities of the quantization method. We believe that it is important for future works to give a layer-wise breakdown of their method that will show exactly from which layers the improvement in performance is achieved. This will allow both better understanding and create a richer design space for compound quantization methods. Our method is a first step in the direction of establishing better analysis tools for quantization which we hope will serve as a framework for
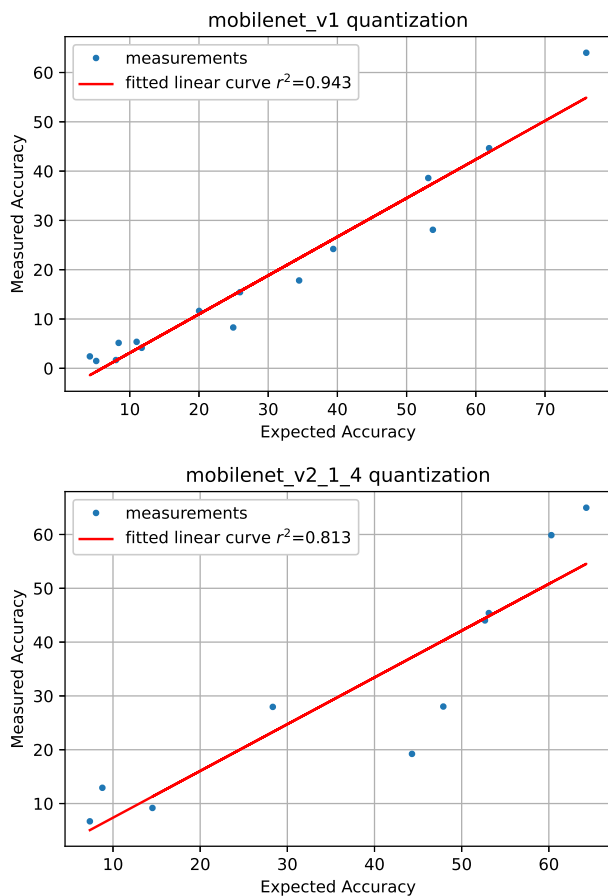
Figure 9. MobileNets expected degradation vs. measured. Correlation is lower due to bias shift. Measurements were done on ImageNet-1K validation set and were taken similarly to Figure 2

future works.

References

Anonymous. {BRECQ}: Pushing the limit of post-training quantization by block reconstruction. In *Submitted to International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=POWv6hDd9XH. under review. 3

Banner, R., Nahshan, Y., Hoffer, E., and Soudry, D. ACIQ: analytical clipping for integer quantization of neural networks. *CoRR*, abs/1810.05723, 2018. URL http://arxiv.org/abs/1810.05723. 1, 2, 4

Baskin, C., Liss, N., Chai, Y., Zheltonozhskii, E., Schwartz, E., Giryes, R., Mendelson, A., and Bronstein, A. M. NICE: noise injection and clamping estimation for neural network quantization. *CoRR*, abs/1810.00162, 2018. URL http://arxiv.org/abs/1810.00162. 2, 4

Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. Zeroq: A novel zero shot quantization framework. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. doi: 10.1109/cvpr42600.2020.01318. URL http://dx.doi.org/10.1109/cvpr42600.2020.01318. 2

Choi, J., Chuang, P. I., Wang, Z., Venkataramani, S., Srinivasan, V., and Gopalakrishnan, K. Bridging the accuracy gap for 2-bit quantized neural networks (QNN). *CoRR*, abs/1807.06964, 2018. URL http://arxiv.org/abs/1807.06964. 4

Choukroun, Y., Kravchik, E., Yang, F., and Kisilev, P. Low-bit quantization of neural networks for efficient inference. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct 2019. doi: 10.1109/iccvw.2019.00363. URL http://dx.doi.org/10.1109/ICCVW.2019.00363. 1

Dong, Z., Yao, Z., Cai, Y., Arfeen, D., Gholami, A., Mahoney, M. W., and Keutzer, K. Hawq-v2: Hessian aware trace-weighted quantization of neural networks. 2019a. 1

Dong, Z., Yao, Z., Gholami, A., Mahoney, M. W., and Keutzer, K. HAWQ: hessian aware quantization of neural networks with mixed-precision. *CoRR*, abs/1905.03696, 2019b. URL http://arxiv.org/abs/1905.03696. 2, 4

Elthakeb, A. T., Pilligundla, P., Yazdanbakhsh, A., Kinzer, S., and Esmaeilzadeh, H. Releq: A reinforcement learning approach for deep quantization of neural networks. *CoRR*, abs/1811.01704, 2018. URL http://arxiv.org/abs/1811.01704. 2

Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned step size quantization. *CoRR*, abs/1902.08153, 2019. URL http://arxiv.org/abs/1902.08153. 2, 4

Fang, J., Shafiee, A., Abdel-Aziz, H., Thorsley, D., Georgiadis, G., and Hassoun, J. Near-lossless post-training quantization of deep neural networks via a piecewise linear approximation. 2020. 1

Finkelstein, A., Almog, U., and Grobman, M. Fighting quantization bias with bias. *CoRR*, abs/1906.03193, 2019. URL http://arxiv.org/abs/1906.03193. 2, 4, 6

Haroush, M., Hubara, I., Hoffer, E., and Soudry, D. The knowledge within: Methods for data-free model compression. 2019. 2, 4

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385. 4

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL http://arxiv.org/abs/1704.04861. 4, 6

Hubara, I., Nahshan, Y., Hanani, Y., Banner, R., and Soudry, D. Improving post training neural quantization: Layer-wise calibration and integer programming. 2020. 2, 3

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015. 3

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A. G., Adam, H., and Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *CoRR*, abs/1712.05877, 2017. URL http://arxiv.org/abs/1712.05877. 1, 3, 5

Jain, S. R., Gural, A., Wu, M., and Dick, C. H. Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks. 2019. 1

Liu, X., Ye, M., Zhou, D., and Liu, Q. Post-training quantization with multiple points: Mixed precision without mixed precision. 2020. 2, 4

Meller, E., Finkelstein, A., Almog, U., and Grobman, M. Same, same but different - recovering neural network quantization error through weight factorization. *CoRR*, abs/1902.01917, 2019. URL http://arxiv.org/abs/1902.01917. 2, 4

Nagel, M., van Baalen, M., Blankevoort, T., and Welling, M. Data-free quantization through weight equalization and bias correction. *CoRR*, abs/1906.04721, 2019. URL http://arxiv.org/abs/1906.04721. 2, 6

Nagel, M., Amjad, R. A., van Baalen, M., Louizos, C., and Blankevoort, T. Up or down? adaptive rounding for post-training quantization. 2020. 2, 3

Nahshan, Y., Chmiel, B., Baskin, C., Zheltonozhskii, E., Banner, R., Bronstein, A. M., and Mendelson, A. Loss aware post-training quantization. 2019. 2, 4

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. 3

Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., and Chen, L. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018a. 6

Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., and Chen, L. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018b. URL http://arxiv.org/abs/1801.04381. 4

Stock, P., Joulin, A., Gribonval, R., Graham, B., and Jégou, H. And the bit goes down: Revisiting the quantization of neural networks. *CoRR*, abs/1907.05686, 2019. URL http://arxiv.org/abs/1907.05686. 1

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842. 4

Uhlich, S., Mauch, L., Yoshiyama, K., Cardinaux, F., García, J. A., Tiedemann, S., Kemp, T., and Nakamura, A. Differentiable quantization of deep neural networks. *CoRR*, abs/1905.11452, 2019. URL http://arxiv.org/abs/1905.11452. 1, 2

Wang, K., Liu, Z., Lin, Y., Lin, J., and Han, S. HAQ: hardware-aware automated quantization. *CoRR*, abs/1811.08886, 2018. URL http://arxiv.org/abs/1811.08886. 1, 2

Wang, P., Chen, Q., He, X., and Cheng, J. Towards accurate post-training network quantization via bit-split and stitching. In *Proceedings of the 37nd International Conference on Machine Learning (ICML)*, pp. 243–252, July 2020. 3

Wu, H., Judd, P., Zhang, X., Isaev, M., and Micikevicius, P. Integer quantization for deep learning inference: Principles and empirical evaluation, 2020. 3

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated Residual Transformations for Deep Neural Networks. *arXiv e-prints*, art. arXiv:1611.05431, November 2016. 4

Yang, J., Shen, X., Xing, J., Tian, X., Li, H., Deng, B., Huang, J., and Hua, X. Quantization networks. 2019. 2, 4

Zhao, R., Hu, Y., Dotzel, J., Sa, C. D., and Zhang, Z. Improving neural network quantization without retraining using outlier channel splitting. *CoRR*,

abs/1901.09504, 2019. URL http://arxiv.org/abs/1901.09504. 2

Zhou, Y., Moosavi-Dezfooli, S.-M., Cheung, N.-M., and Frossard, P. Adaptive Quantization for Deep Neural Network. *arXiv e-prints*, art. arXiv:1712.01048, December 2017. 2, 3