Supplementary materials for "QPP: Real-Time Quantization Parameter Prediction for Deep Neural Networks"

Anonymous CVPR submission

Paper ID 8893

A. Standard Quantization

In this section we discuss standard quantization based on QPP. The following materials were not included in the paper to save space for the experiment analysis and explanation. Moreover, the D+S approach highlights the advantages of QPP stronger compared to standard quantization. More precisely, we show that QPP is also effective for standard quantization(it will be described in the next subsection A.1) to emphasize the generality of our approach. All results are obtained on the same sets of the datasets and the models (see Table A.2) with the same experiment setup (same places of QPPs).

As well as the D+S approach described in the paper, the standard quantization can use an asymmetric quantization scheme (what is also known as the quantization with offset) based on α-quantile statistics. So, QPP's prediction target for both quantization methods is the α-quantile. The difference is only in a value of α-quantile: D+S operates with values like 0.99 (1% density) or 0.97 (3% density) or even lower, and the standard uses the value larger than 0.999 (see Table A.2).

Most NN inference frameworks use the static standard quantization approach. That is why it is crucial to provide such experiments.

A.1. Definition

Feature maps. The standard approach to quantize activations requires clipping and round operations:

$$\tilde{x} = quant(x, Q^l) = \left\lfloor clip\left(\frac{x}{s^l},\ a^l,\ b^l\right)\right\rceil s^l, \qquad \text{(A.1)}$$

where $\lfloor\cdot\rceil$ denotes the round operation. We choose quantization bounds $Q^l = (a^l, b^l)$ as QPs at each l-th layer (a^l and b^l represent bottom and upper thresholds). The formula A.1 describes the so-called Fake quantization - the discretization of floating point variables used for emulation of integer arithmetic. Here for simplicity we are omitting the offset usually used. We assume equal quantization steps s^l with bit-width k of quantized values defined as follows:

$$s^l = \frac{b^l - a^l}{2^k - 1}. \qquad \text{(A.2)}$$

QPs Q^l values have a strong influence on the quality of the quantized model. Poorly selected they lead to an increase in the quantization error because either the quantization step is too large or the values of outliers change too much. Motivated by this observation, in our experiments we consider α-quantile value of tensor of activations as clipping values. Optimization is done via greedy search of α, which leads to best quality. Usually a^l and b^l are estimated by the minimum and the maximum of the activations correspondingly. Our formulation doesn't exclude this opportunity: when alpha tends to one, QPs Q^l becomes equal to min and max values respectively.

Weights. Our work focuses on predicting the quantization bounds of activations because they vary between samples in contrast to the weights quantization, which can be estimated off-line without a calibration set. In other words, the weights quantization is out of our research scope. So, for simplicity, we used the same quantization function A.1 to quantize weights as one for activations and the following quantization step:

$$s^l_w = \frac{\max(W^l) - \min(W^l)}{2^t - 1}. \qquad \text{(A.3)}$$

Bit-width. Going forward, we use 8 and 4 bits for weights and activations correspondingly ($t = 8, k = 4$). We discuss in detail the bit-width in Section A.3.

A.2. QP Estimation Stage

Dynamic. Let's settle why the dynamic quantization is not efficient (see Figure A.1). Firstly, one needs to run the 32-bit data through the CPU to obtain the QPs (min/max or α-quantiles). Secondly, we have to scan the data again to quantize data to int8 (or int4). Usually, the size of a tensor of feature map exceeds a CPU cache. Hence, the scanning requires interaction with RAM, which is an expensive operation in terms of both power and time consumption.

Table A.1. Inference time of different implementations of convolutions: float16 - full-precision convolution via BOLT; Int8 - quantized convolution via BOLT. Convolution size (KxKxF): 3x3xC. CPU: Kirin 980 (Cortex A76, 1 thread).

Activation size	flloat16 conv, ms (%)	Inference time, ms (%)				Quantile
		Min/Max	Quant	int8 conv	Total	
56x56x64	3.23 (184%)	0.03 (2%)	0.19 (11%)	1.76 (100%)	1.98 (113%)	0.22 (12%)
112x112x64	13.53 (188%)	0.37 (5%)	0.75 (10%)	7.18 (100%)	8.30 (116%)	0.78 (11%)
224x224x64	62.17 (200%)	0.71 (2%)	2.98 (10%)	31.03 (100%)	34.72 (112%)	2.99 (10%)

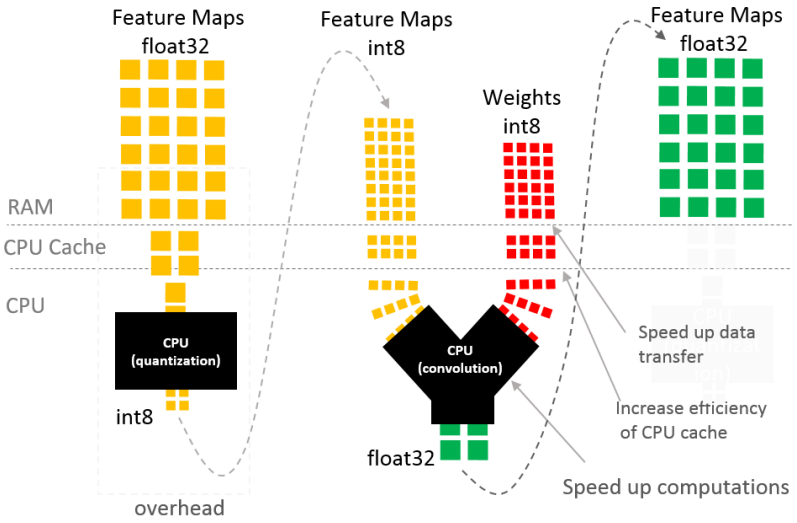

Figure A.1. Dynamic quantization scheme. The quantization requires running the 32-bit data through the CPU.

If the inference time of int8 convolution is taken as a unit, then the min/max estimation costs 2-5%, and α-quantile costs 10-12% (see Table A.1). In the case of min/max, it is possible to fuse its estimation to previous convolution (like it was done in the BOLT framework) and avoid this overhead. Unfortunately, this cannot be done for the α-quantile measuring.

The main disadvantage of dynamic scheme is the absence of an opportunity to fuse the quantization operation (10-11% overhead) with the previous convolution.

Thus, the cumulative overhead of α-quantile estimation (12%) and the quantization (11%) can reach 25%, making it not advisable.

Static. If all QPs are known before inference, we can fuse convolution and quantization operations and eliminate these overheads (see Figure A.2) but typically it leads to higher quality degradation compared to dynamic method (see Section A.3).

Static with QPP. QPP allows one to calculate parameters once (several times in the case of extra QPPs). Therefore, one can significantly reduce the time for QPs evaluating but almost completely preserve the quality compared to the dynamic approach. In next Section A.3 we show that.

A.3. Experiments

We examined QPP's performance for 4 tasks: **classification** (ResNet18 and ResNet34 models [6] on ImageNet [5] dataset), **segmentation** (HRNetV2-W18 [1] model [9, 10] on CityScapes dataset [4]), **facial landmark** (HRNetV2-W18 model on COFW [3], WFLW [11], 300W [7] datasets), and **super-resolution** (ESPCN [8] on Vimeo [12], Set5 [2] and Set14 [13] datasets). More precise experiment set up can be found in Table A.2.

The pipeline for every experiment is the same and can be described as follows. In our experiments we considered α-quantiles of activations as clipping values, i.e. QPs. The particular sub-optimal quantiles were chosen for each NN and for each dataset correspondingly as results of grid search. Such simple yet efficient approach allowed us to achieve a small quality drop in all experiments, and in case of classification allowed to achieve comparable results with ACIQ (w/o bit allocation and bias-correction)[1]: 66.6 % for ACIQ and 66.9 % for our dynamic method.

We reduced the bit-width of activations to 4 bits to highlight the differences between the methods of QPs estimation. For example, for the considered models and the datasets 8-bit models show no accuracy drop.

[1] https://github.com/HRNet

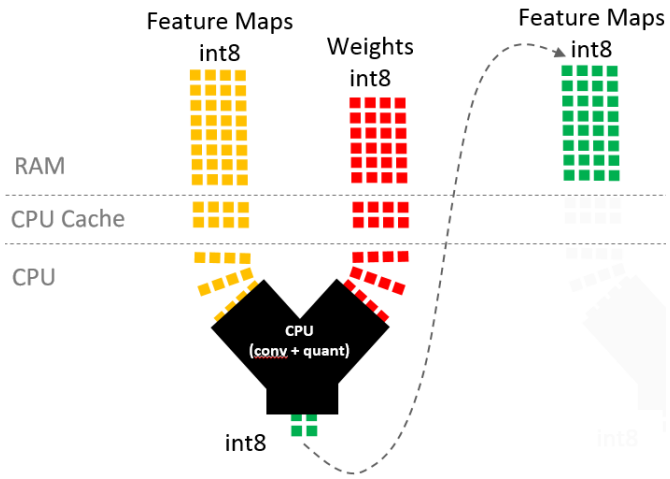

Figure A.2. Static quantization scheme. There is no overhead on a QP estimation and a quantization.

Table A.2. Experiments set up

Task	Models	Datasets	Quantized head	Quantized tail	QPPs number (QPPs positions)	QPP train samples	Optimal quantile
Classification	ResNet18, ResNet34	ImageNet	No	No	1 QPP (first ReLU)	30000	0.9997
Segmentation	HRNet	CityScapes	No	No	4 QPPs (first ReLU, Tr. Layer 1,2,3)	All	0.9993
Facial Landmark	HRNet	COFW, WFLW, 300W	No	No	4 QPPs (first ReLU, Tr. Layer 1,2,3)	All	0.9991 0.9991 0.9993
Super - Resolution	ESPCN	Vimeo, Set5, Set14	Yes	Yes	1 QPP (NN's input)	10000	0.9998

Table A.3. Performance of quantization schemes vs ground truth (weights: 8 bit, activations: 4 bit)

3*Scheme	Classification Acc Top 1, % / Acc Top 5, %		Segmentation mIoU, %	Facial Landmark NME, % / FR(0.1), %			Super-Resolution PSNR, dB		
	Imagenet		CityScapes	COFW	WFLW	300W	Vimeo	Set5	Set14
	ResNet18	ResNet34	HRNet	HRNet			ESPCN		
FP32	69.76 / 89.08	73.30 / 91.42	70.26	3.45 / 0.2	4.6 / 3.12	3.85 / 0.34	34.06	30.74	27.06
Dynamic	66.85 / 87.22	70.34 / 89.70	66.14	3.7 / 1.18	4.95 / 4.56	4.01 / 0.5	33.45	30.39	26.84
Static	66.65 / 87.14	70.27 / 89.53	66.23	3.77 / 0.79	4.99 / 5.00	4.15 / 1.00	33.31	30.25	26.75
QPP	66.90 / 87.23	70.30 / 89.69	66.21	3.69 / 0.59	4.96 / 4.88	4.12 / 0.5	33.42	30.36	26.84

We presented the results of our experiments on full quantization in two tables. In Table A.3 we demonstrate the scores of the same models computed for ground truth labels. In three out of four experiments we see that dynamic quantization is superior to static quantization (except the segmentation task). In all experiments one can also observe that the QPP-based scheme results are closer to dynamic quantization. The typical behavior of methods in relation to each other we illustrated by the example of ResNet18. Here we see that the QPP-based scheme lets us increase accuracy by 0.25% compared to the static approach and almost repeat the result of dynamic quantization. If we consider the top5 accuracy, we can see that the scores of dynamic quantization and QPP scheme differ by only 0.01%, which indicates

Table A.4. Performance of quantization schemes vs FP32 models (weights: 8 bit, activations: 4 bit)

3*Scheme	Classification Acc Top 1, %		Segmentation mIoU, %	Facial Landmark NME, %			Super-Resolution PSNR, dB		
	Imagenet		CityScapes	COFW	WFLW	300W	Vimeo	Set5	Set14
	ResNet18	ResNet34	HRNet	HRNet			ESPCN		
Dynamic	83.616	84.716	80.52	1.52	2.14	1.78	43.48	41.60	40.71
Static	83.120	84.396	80.87	1.58	2.33	1.99	42.74	40.44	40.11
QPP	83.630	84.432	80.75	1.51	2.22	1.87	43.25	41.32	40.48

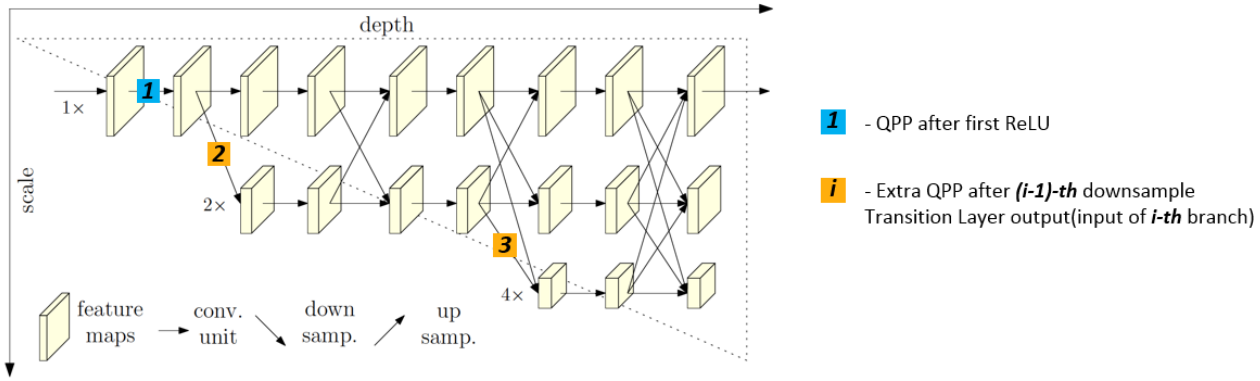

Figure A.3. QPPs positions in HRNet[2]

high stability of the quantized NN's output.

Additionally, we want to note that the paper addressed to improve the post-training quantization methods, which main purpose is to obtain the quantized NN which is able to produce the same outputs as the NN in full-precision format given the same inputs. Thus, we provide the scores computed between outputs of quantized NNs and outputs of full-precision NNs in Table A.4. Here we see the same picture as in the ground truth case. For the ResNet18 model, we have an increase of 0.5% in accuracy compared to the static approach. This also proves that the properly constructed QPPs can predict quantization parameters for various NNs and datasets well.

B. QPP Positions

In this section we discuss QPP positions in NN. First of all, we would like to note that, in most cases, the first convolutional layer of the NN is not quantized. Thus, there is no sense in placing QPP in front of it and estimating QPs from the NN's input. Therefore, in our experiments (except Super-Resolution models, where all layers were quantized) we placed the QPP after the first activation function and predicted subsequent QPs based on the tensor of activations (see Figure B.4). This approach can significantly improve the quality of predictions. [3]

Also, deep models require several QPPs to preserve quality of predictions. For segmentation and facial landmark tasks we used HRNet as a baseline model, which has big amount of layers. Therefore, we decided to place 4 QPPs in these models. As it was discussed above the first QPP is placed after the first ReLU. Other three QPPs are placed before first convolutions of every new branch (see Figure A.3).

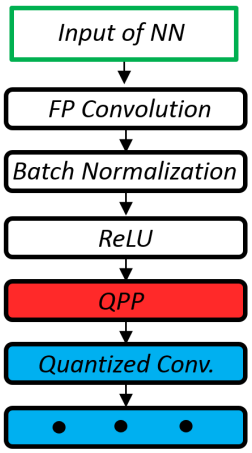

Figure B.4. QPP after first ReLU

C. Feature Selection

We would like to note that the regressor can't have high-dimensional feature space as it strongly affects computational complexity. So one needs to reach the highest pos-

[3]Original picture of HRNet: https://jingdongwang2017.github.io/Projects/HRNet/

Table B.5. Power consumption: float16 - full-precision convolution via BOLT; Int8 - quantized convolution via BOLT; D+S convolution - 1% density. Activation size (HxWxC): 56x56x64. Convolution size (KxKxF): 3x3x64. CPU: Kirin 980 (Cortex A76, 1 thread).

Metric	float16 Conv	Min/Max	Quantile	Dynamic Int8 Conv Total	Dynamic D(int8)+S(float16) Conv
mAh	0.603	0.001	0.020	0.295	0.355
%	100%	0.1%	3%	49%	59%

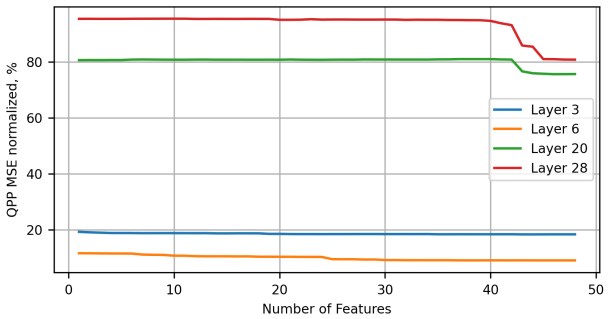

Figure C.5. QPP MSE vs Number of Features (ResNet34)

sible quality of predictions while keeping the feature space small. The solution lies in the proper feature selection procedure. To start with, we observed that α-quantiles below 0.9-quantile are fairly stable from sample to sample and therefore have a big correlation, which leads to the poor performance of linear regressor. This is due to the type of distribution and the large size of activations. Moreover, when extra QPP is used, the input distribution of this QPP is slightly changed by quantization. Therefore, linearly dependent features can lead to unpredictable results for these QPPs.

To estimate linear dependence between features and QPs we performed F-test. We choose the 44 different α-quantiles, mean, max, min, std, absolute max values of input tensor as baseline feauture space. Then we selected k best features and trained regressors on them. The results for different k are summarized in Figure C.5. The Y-axis represents the MSE of predictions on train subset normalized to the MSE of QPs obtained by the static quantization with averaging. One can see that the increase in the number of features has almost no effect on the quality of predictions. That is why in our experiments we used the feature space consisting of 3 α-quantiles, mean and max values of input tensor.

D. Power Consumption

In this paper we discussed inference time and quality of the quantized model as the main characteristics of quantization algorithm. But power consumption is also an essential parameter for industry application. We computed it on ARM CPU Kirin 980 (Cortex A76, 1 thread) and presented results in Table B.5. The following quantization approaches were considered: 1) FP16 convolution via BOLT Framework, 2) Int8 convolution via BOLT Framework, 3) Dense + Sparse convolution (our implementation). One can see that calculation of min/max or quantile of tensor of activations requires low power (0.1% and 3% respectively). Therefore, using a dynamic approach instead of a static brings almost no additional energy costs. In addition, D+S with a density of 1% requires 59% of the power consumption of the FP16 model, which is 10% more power than for dynamic quantization of int8. From our point of view, this is an acceptable level of costs, since D+S allows one to achieve almost FP quality of the quantized model and save 41% of energy.

References

[1] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7950–7958. Curran Associates, Inc., 2019. 2

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie line Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference*, pages 135.1–135.10. BMVA Press, 2012. 2

[3] X.P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *ICCV*, 2013. 2

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 2

[6] 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. 2

[7] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: database and results. *Image and vision computing*, 47:3–18, 3 2016. 2

[8] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[9] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 2

[10] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019. 2

[11] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, 2018. 2

[12] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, Feb 2019. 2

[13] Roman Zeyde, Michael Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, 2010. 2