IMPQ: REDUCED COMPLEXITY NEURAL NETWORKS VIA GRANULAR PRECISION ASSIGNMENT

Sujan Kumar Gonugondla[*]

Amazon

Naresh R. Shanbhag

University of Illinois at Urbana-Champaign

ABSTRACT

The demand for the deployment of deep neural networks (DNN) on resource-constrained Edge platforms is ever increasing. Today's DNN accelerators support mixed-precision computations to enable reduction of computational and storage costs but require networks with precision at variable granularity, i.e., network, layer or kernel level. However, the problem of granular precision assignment is challenging due to an exponentially large search space and efficient methods for such precision assignment are lacking. To address this problem, we introduce the iterative mixed-precision quantization (IMPQ) framework to allocate precision at variable granularity. IMPQ employs a sensitivity metric to order the weight/activation groups in terms of the likelihood of misclassifying input samples due to its quantization noise. It iteratively reduces the precision of the weights and activations of a pretrained full-precision network starting with the least sensitive group. Compared to state-of-the-art methods, IMPQ reduces computational costs by 2×-to-2.5× for compact networks such as MobileNet-V1 on ImageNet with no accuracy loss. Our experiments reveal that kernel-wise granular precision assignment provides 1.7× higher compression than layer-wise assignment.

***Index Terms*—** mixed-precision, DNN, quantization

1. INTRODUCTION

In the past decade, we have seen deep neural networks (DNNs) achieve tremendous success in a wide variety of applications. Due to concerns in terms of privacy, reliability, and latency, there is an increasing demand for the deployment of DNNs in battery-powered mobile and edge devices. However, the large computation and storage costs of DNNs pose a challenge for such deployment. This challenge is currently being addressed by the development of a) specialized accelerators for DNNs, and b) low-complexity DNN. However, these approaches are often addressed independently and risk being incompatible with each other.

Reducing the DNN complexity has been an active area of research. Techniques such as network pruning [1] reduce complexity by removing redundant weights and computations. However, this results in an irregular network structure that is challenging to implement on DNN accelerators that favors regular data-flows. Designing compact network architectures from scratch has been effective in DNN complexity reduction, e.g., MobileNets [2], SqueezeNet [3], ShuffleNet [4], and ConDenseNet [5]. These networks employ specialized layers such as *depth-wise separable layers*, *grouped convolutions*, *point-wise convolutions*, among others and are designed to achieve high accuracy with far fewer computations and parameters.

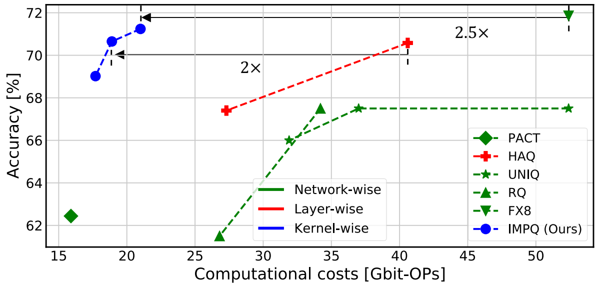


Fig. 1: Classification accuracy on ImageNet-1k using MobileNet-V1 vs. computational costs (Gbit-OPs). The proposed IMPQ method achieves a high accuracy with a low computational cost compared to the state-of-art fixed-point quantization methods (PACT [10], HAQ [17], UNIQ [18], and RQ [19]).

Recently, training methods were developed that use binary and ternary precision for weights and activations [6, 7]. These techniques applied to large networks such as VGG-16 and ResNet-18 result in minimal accuracy drop from their floating-point counterparts. However, ultra low-precision training applied to compact networks such as MobileNet-V1 results in a catastrophic accuracy drop [8] and therefore still remains a challenge.

Conventional quantization techniques [9, 10] use the same number of bits across all layers. However, the sensitivity of inference accuracy to quantization noise varies across layers [11], and across kernels within a layer [12, 13] leading to different per-layer/kernel precision requirements. This diversity across layers and kernels can be exploited by variable precision DNN accelerators such as [14, 15, 16]. In fact we find the DNN quantization with precision assigned at a kernel-level granularity aggressively reduces the computation complexity without minimal loss in accuracy (see Fig. 1).

Granular precision assignment across layers is challenging due to the search space that increases exponentially with the number of layers/kernels. For example, an N granular precision assignment will have B^N possibilities, where B is the maximum assignable precision. Therefore, there are greater than 4 billion possibilities for VGG-16 if each layer has four possible precision assignments. To solve this HAQ and AutoQ [17, 12] uses a reinforcement learning (RL) agent to pick a precision assignment by iteratively evaluating the network on a hardware model. HAQ and AutoQ need to train an RL agent which is computationally expensive.

In this paper, we present the iterative mixed-precision quantization (IMPQ) to address the challenge of assigning bit precision in DNNs at any granularity. IMPQ employs an iterative process, where in each iteration IMPQ uses sensitivity estimates to assign appropriate precision to weight/activation group and fine-tune the network

[*] This work was performed by the first author at the University of Illinois at Urbana-Champaign prior to joining Amazon.

This work was supported by the Center for Brain-Inspired Computing (C-BRIC) and Artificial Intelligence Hardware (AIHW) funded by the Semiconductor Research Corporation (SRC) and the Defense Advanced Research Projects Agency (DARPA).

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

with this precision assignment. Though IMPQ can be used at any granularity, in the rest of the paper we will use kernel-wise precision assignment for weights and layer-wise for activations. We demonstrate the effectiveness of our approach with a compact network such as MobileNet-V1. The benefits of the granular precision assignment are discussed and demonstrated via experiments.

2. BACKGROUND - NOISE GAIN ANALYSIS

Assessing the sensitivity of weights and activations is a key step in complexity reduction techniques. Some works have used weight magnitudes [1] or hessian diagonals [13] to determine sensitivity, which are often not correlated to classification accuracy or computationally expensive.

One such technique that directly assesses the impact of quantization on the classification accuracy is *noise gain analysis* (NGA) [20, 11]. Given a floating-point baseline and a precision assignment, NGA provides an analytical upper bound on the mismatch probability, $p_m = P(\hat{Y}_{\text{fx}} \neq \hat{Y}_{\text{fl}})$, where $\hat{Y}_{\text{fl}}$ and $\hat{Y}_{\text{fx}}$ are the class labels predicted by the floating-point baseline and the fixed-point network, respectively. This bound on p_m is given by:

$$p_m \leq \sum_l \Delta^2_{A,l} E_{A,l} + \sum_k \Delta^2_{W,k} E_{W,k} \tag{1}$$

where $\Delta^2_{A,l} = 2^{-(B_{A,l}-1)}$, and $\Delta^2_{W,k} = 2^{-(B_{W,k}-1)}$ are the quantization step-size of the set of activations $\mathcal{A}_l$ and the set of weights $\mathcal{W}_k$, respectively. The noise gains $E_{A,l}$ and $E_{W,k}$ quantify the impact of quantization noise of $\mathcal{A}_l$ and $\mathcal{W}_k$ on the mismatch probability, respectively. Analytical expressions for the noise gains are given by:

$$E_{A,l} = \mathbb{E}\left[\sum_{i \neq \hat{Y}_{\text{fl}}} \frac{\sum_{a \in \mathcal{A}_l} \left|\frac{\partial(Z_i - Z_{\hat{Y}_{\text{fl}}})}{\partial a}\right|^2}{24|Z_i - Z_{\hat{Y}_{\text{fl}}}|^2}\right] \tag{2}$$

$$E_{W,k} = \mathbb{E}\left[\sum_{i \neq \hat{Y}_{\text{fl}}} \frac{\sum_{w \in \mathcal{W}_k} \left|\frac{\partial(Z_i - Z_{\hat{Y}_{\text{fl}}})}{\partial w}\right|^2}{24|Z_i - Z_{\hat{Y}_{\text{fl}}}|^2}\right] \tag{3}$$

where Z_i is the soft output of the DNN corresponding to the class label i. Empirically, the noise gains are estimated by taking gradients with respect to margins on a subset of the training samples during the standard back-propagation phase of training.

3. ITERATIVE MIXED-PRECISION QUANTIZATION (IMPQ)

IMPQ employs a four-step iterative process: a) evaluate sensitivity, b) pick weight/activation groups, c) reduce precision, and d) fine-tune network.

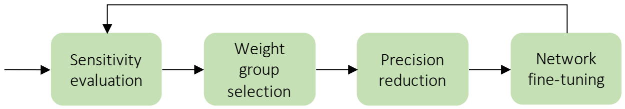


Fig. 2: The proposed iterative mixed-precision quantization (IMPQ) methodology.

IMPQ (see Fig. 2 and Algorithm 1) begins with a pretrained network where weights and activations are quantized with identical precision across all layers and kernels. The precision of this pretrained network is chosen to meet the state-of-the-art accuracy of a floating-point network, and therefore it is the maximum allowable precision in the final mixed-precision network. We use the term *weight group* or *activation group* to refer to the granular block having the same precision. Weight groups can be at a kernel-wise or layer-wise granularity, and activation groups are at a layer-wise granularity. Kernel in this work refers to a 3-dimensional filter that is convolved with the input feature maps to obtain one output feature map.

Sensitivity Evaluation: Sensitivity estimates the impact of perturbations of weights or activations on the classification accuracy. We use the method proposed in [20, 11] to evaluate the sensitivity to quantization noise. However, instead of comparing a fixed-point network to a floating-point network, we are interested in evaluating the impact of reducing precision for a specific weight group or an activation group in a quantized network. Therefore, we adapt (2) and (3) for a quantized network given the predicted class $\hat{Y}$ as follows:

$$M_{A,l} = 2^{-B_{A,l}} \mathbb{E}\left[\sum_{i \in S_{\hat{Y}}} \frac{\sum_{a \in \mathcal{A}_l} \left|\frac{\partial(Z_i - Z_{\hat{Y}})}{\partial a}\right|^2}{24|Z_i - Z_{\hat{Y}}|^2}\right] \tag{4}$$

$$M_{W,k} = 2^{-B_{W,k}} \mathbb{E}\left[\sum_{i \in S_{\hat{Y}}} \frac{\sum_{w \in \mathcal{W}_k} \left|\frac{\partial(Z_i - Z_{\hat{Y}})}{\partial w}\right|^2}{24|Z_i - Z_{\hat{Y}}|^2}\right] \tag{5}$$

where $M_{A,l}$ and $M_{W,k}$ are the sensitivities of activation group $\mathcal{A}_l$ and the weight group $\mathcal{W}_k$, respectively, $S_{\hat{Y}}$ is a set of class labels excluding $\hat{Y}$, and $B_{A,l}$ and $B_{W,k}$ are the precisions of l-th activation group and k-th weight group, respectively.

Weight/Activation Groups Selection: In the second step of IMPQ methodology, we use the sensitivity estimated via (4) and (5) to identify the weight/activation groups that have the least impact on mismatch probability. There are three possible ways to pick these weight/activation groups: a) thresholding the sensitivity, b) thresholding mismatch probability, and c) constrain the number of groups chosen.

Precision Reduction: We reduce the precision of the weight or activation groups that were chosen in the previous step. This may result in an accuracy drop that needs to be recovered.

Network Fine-tuning: Here we train the current network usually for a few epochs on the training data.

Each iteration of IMPQ takes a small step in reducing the overall precision requirements. The network at the end of each iteration is used to determine if the complexity is sufficiently reduced as per the user requirements. The specifics of the experiments and approximations used for practical implementation of this methodology are presented in the next section.

4. EXPERIMENTS

We demonstrate the effectiveness of IMPQ with experiments on the following image classification datasets: 1) CIFAR-10 [21], 2) SVHN [22], and 3) ImageNet-1k [23]. We choose ResNet-20 [24] and VGG-Small [25] for classifying the CIFAR-10 dataset, and MobileNet-V1 for classifying the ImageNet-1k dataset.

4.1. Implementation Details

In all our experiments, we train full-precision (FP) baseline networks from scratch. We use stochastic gradient descent with weight decay and the same hyperparameters, such as the learning rate, transformation, and augmentation for all out experiments [1].

[1] link to PyTorch implementation : https://github.com/gsujankumar/IMPQ

Algorithm 1 Proposed iterative mixed-precision quantization (IMPQ) methodology

Input: Neural network architecture $f(\mathbf{X}, \mathcal{W})$; floating point reference weights $\mathcal{W}_{k,f} \subset \mathcal{W}_f$; quantized weights $\mathcal{W}_k \subset \mathcal{W}$ are the k-th weight group with precision $B_{w,k}$; $\mathcal{A}_l \subset \mathcal{A}$ are the l-th activation group with precision $B_{a,l}$; training data and labels $\{(\mathbf{X}_j, \mathbf{y}_j)\}$ where j is index of the training batch; number of training batches N_{Batch}; number of groups picked for precision reduction N_{group}; and the number of iterations of IMPQ N_{iter}.
Output: Quantized network weights.
Initialize: $\mathcal{W}_f$,, $B_{W,k} = B_{W,start}$ and $B_{A,l} = B_{A,start}$

```
function QuantizedTrain
    $i \leftarrow i + 1$
    while not converged do
        $\mathbf{y}'_i \leftarrow f(\mathbf{X}_i, \mathcal{W})$                ▷ Forward propagation
        Evaluate $\mathcal{L}(\mathcal{W}, \mathbf{y}'_i, \mathbf{y}_i)$                ▷ Loss function
        $\mathcal{W}_f \leftarrow \mathcal{W}_f - \nabla\mathcal{L}(\mathcal{W}, \mathbf{y}'_i, \mathbf{y}_i)$                ▷ Weight update
        $\mathcal{W}_k \leftarrow$ Quantize($\mathcal{W}_{k,f}, B_{w,k}$)
        $i \leftarrow 0$ if ($i ==$ NBatch)    $i + 1$    otherwise
    end while
end function
QuantizedTrain()                ▷ Train initial network
for $k := 1$ to $N_{\text{iter}}$ do
    $\mathbf{z} \leftarrow f(\mathbf{X}, \mathcal{W})$                ▷ Forward propagation
    Evaluate sensitivity $M_{W,l}$ using (5)
    SI $\leftarrow$ SortIdx($M_{W,l}$)        ▷ Sorts weight groups based on $M_{W,l}$
    for $n := 1$ to $N_{\text{group}}$ do
        $r \leftarrow$ SI($n$)
        $B_{W,r} \leftarrow B_{W,r} - 1$                ▷ Reduce precision
    end for
    QuantizedTrain()
end for
Repeat for activations
```

Quantization: IMPQ works for both uniform and non-uniform quantization. Experiments in this paper use uniform quantization for both weights and activations. For each weight group we clip the weights to a range $[d, -d]$ before quantization, where d is a multiple of the second moment of $\mathcal{W}_k$, given by $d = 4\sqrt{\sum_{w\in\mathcal{W}_k} w^2 / |\mathcal{W}_k|}$.

Activations are unsigned, and are clipped and quantized between $[0, d]$. The clipping value $d = \max(\beta_i + 6\gamma_i)$, where β_i and γ_i are the shift and scale batch-norm parameters of the i-th feature map, respectively.

Approximations to sensitivity: Estimating the sensitivity as per (4) and (5) requires us to estimate the gradients of the weights with respect to each class probability, making this step complex. Therefore, we consider the gradients with respect to a 10 classes with the smallest margins, i.e., the set $S_{\hat{Y}}$ in (4) and (5) is a subset of class labels.

Evaluation metrics: We use the following metrics to evaluate the storage and computation costs of quantized networks:

1) *Effective weight/activation precisions* ($B_{w,\text{eff}}$ / $B_{a,\text{eff}}$): Average weight/activation precision of the network, defined as,

$$B_{w,\text{eff}} = \frac{\sum_k B_{w,k}|\mathcal{W}_k|}{\sum_k |\mathcal{W}_k|} \qquad B_{a,\text{eff}} = \frac{\sum_l B_{a,l}|\mathcal{A}_l|}{\sum_k |\mathcal{A}_l|} \tag{6}$$

2) *Computational costs* ($\mathcal{C}_C$): Average number of bit operations for the model where each multiply-accumulate operation in a M-dimensional dot product:

$$\mathcal{C}_C = \sum_k N_k\big(B_{w,k}B_{a,k} + (B_{w,k} + B_{a,k} + \log_2(M))\big) \tag{7}$$

where M is the dot product size, N_k, $B_{a,k}$ and $B_{w,k}$ are the total number of max operation, is the activation precision, and weight precision of that MAC operation associated with k-th weight group, respectively.

Table 1: Classification accuracy with weight-only quantization.

Dataset : CIFAR 10	Network : ResNet-20			
Method	$B_{w,\text{eff}}$	FP[†] Acc.	Acc. [%]	Change
BWN [26]	1	92.10	90.2	1.90
TWN [6]	Ternary	91.77	90.78	0.89
TTQ [7]	Ternary	91.77	91.13	0.64
ELQ [27]	Ternary	91.25	91.45	-0.20
ELQ [27]	1	91.25	91.15	0.10
DoReFa [9]	3	92.10	91.81	0.29
DoReFa [9]	2	92.10	91.41	0.69
LQ-Net* [25]	3	92.00	92.00	0
LQ-Net* [25]	2	92.00	91.80	0.20
IMPQ	1.74	92.10	92.00	0.10

Dataset : CIFAR 10	Network : VGG-Small			
Method	$B_{w,\text{eff}}$	FP[†] Acc.	Acc. [%]	Change
BWN [26]	1	93.18	91.77	1.45
TWN [6]	Ternary	93.18	92.56	0.62
LQ-Net* [25]	2	93.8	93.8	0
IMPQ	1.55	93.1	92.97	0.13

Dataset : SVHN	Network : VGG-Small			
Method	$B_{w,\text{eff}}$	FP[†] Acc.	Acc. [%]	Change
BWN [26]	1	97.54	96.22	-1.32
LQ-Net* [25]	2	97.54	97.62	-0.08
IMPQ	1.7	97.54	97.58	-0.04

Dataset : ImageNet	Network : MobileNet-V1			
Method	$B_{w,\text{eff}}$	FP[†] Acc.	Top-1 Acc. [%]	Change
DeepC [28, 17]	2	70.90	37.62	-33.28
DeepC [28, 17]	3	70.90	65.94	-4.96
DeepC [28, 17]	4	70.90	71.14	0.24
HAQ [17]	2	70.90	57.14	-13.76
HAQ [17]	3	70.90	67.66	-3.24
HAQ [17]	4	70.90	71.74	0.84
IMPQ	2	71.20	66.51	-4.71
IMPQ	3	71.20	68.3	-2.92
IMPQ	4	71.20	70.2	-2.02

[*] nonlinear quantization   [†] reported full precision baseline

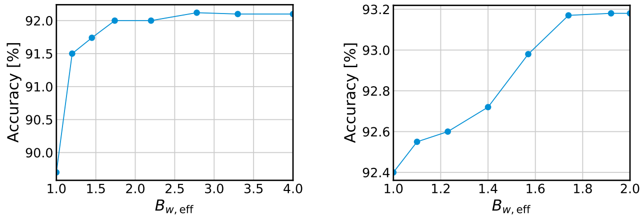
4.2. Weight-only Quantization

We first demonstrate the effectiveness of IMPQ with weight-only quantization. Unlike other techniques such as PACT [10], HAQ [17], and LQ-Nets [25], we quantize all the layers, including the first and the last fully connected layers.

Impact of Network Complexity: Figures 3(a) and 3(b) show how the accuracy and the effective precision change with multiple iterations of IMPQ on ResNet-20 and VGG-Small respectively. The precision of the initial pretrained network on ResNet-20 and VGG-Small is 4-b and 2-b, respectively.

At iso-accuracy, IMPQ reduces the effective precision by >42% with respect to LQ-Net on ResNet-20 (see Table 1). In contrast, over-parameterized networks such as VGG-Small (17× more parameters than the more compact ResNet-20) can operate with very few bits and may not require mixed-precision techniques. Similar results were observed on the SVHN dataset using the VGG-Small network architecture.

Similarly, unlike large networks such as VGG-16 that can operate with ternary precision with minimal loss in accuracy, MobileNet-V1 is sensitive to quantization. For example, we observe a significant accuracy drop on MobileNet-V1 for $B_{w,\text{eff}} \leq 3$ when using Deep Compression [28] and HAQ [17]. In contrast to HAQ and Deep Compression, IMPQ is effective even at a lower precision, e.g., the accuracy of Deep Compression and HAQ at $B_{w,\text{eff}} = 3$ is comparable



(a) (b)

Fig. 3: Classification accuracy on CIFAR-10 using (a) ResNet-20, and (b) VGG-Small as a function of the effective precision recorded at the end of each iteration.

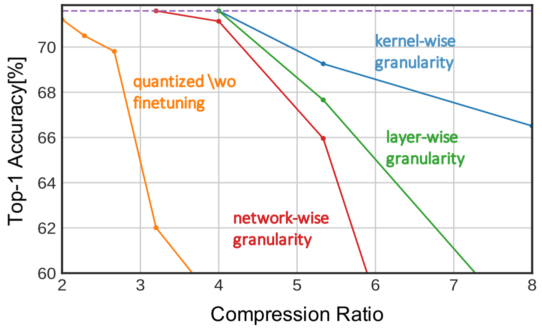


Fig. 4: Classification accuracy on ImageNet-1k using MobileNet-V1 vs. compression ratio (CR) and the granularity of precision assignment. The accuracy of the floating-point network is with no retraining. The compression ratio is calculated with respect to a fixed point network with 16-b weights.

to the accuracy of IMPQ at $B_{w,\text{eff}} = 2$. Thus IMPQ leads to a 33% better compression on MobileNet-V1 for weight-only quantization.

Impact of Granularity: We hypothesize that a granular assignment of precision would lead to better network compression on the whole. We study this impact of granularity using the compression ratio (CR) as a metric, defined as CR $= {}^{B}/_{B_{w,\text{eff}}}$, where B is the precision of a baseline network we are comparing with. The compression ratio quantifies the reduction in model size for storage with respect to a baseline network. We applied IMPQ with: 1) layer-wise, 2) kernel-wise, and 3) network-wise precision allocation. We find that kernel-wise precision allocation gave the best CR followed by layer-wise precision allocation, thus supporting our hypothesis (see Fig. 4).

4.3. Weight and Activation Quantization

For the simultaneous quantization of both weights and activations, we first apply IMPQ with weight-only quantization, and then we extend to activation quantization. Note that applying the activation quantization first is also possible. Activations are quantized layer-wise so that the dot-product computations of the networks can be mapped to fixed-point hardware.

Layer-wise Trends: Figure 5 shows the layer-wise effective weight and activation precision of MobileNet-V1 quantized using IMPQ. We observe the following trends: a) fully-connected layers that constitute 25% of the parameters are aggressively quantized, b) layers closer to the input image are the most sensitive and hence quantized less, and c) point-wise layers that have more parameters have fewer bits than the depth-wise layers. In general, we find that the layers with more parameters and farther from the input are less sensitive, and hence quantized more heavily.

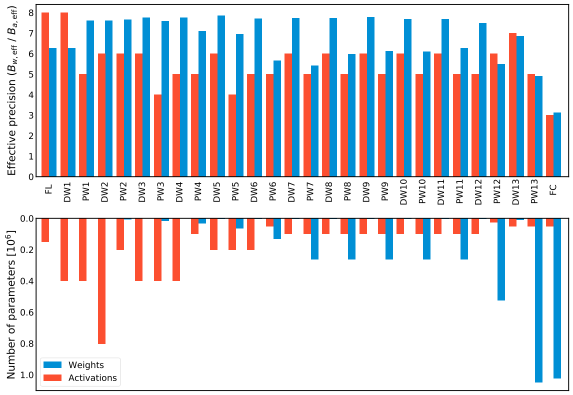


Fig. 5: Effective weight and activation precision across different layers of MobileNet-V1 used for classifying the ImageNet-1k dataset after the application of IMPQ. Here DWk and PWk refer k-th depth-wise and point-wise layers, respectively.

Table 2: Classification accuracy on ImageNet-1k using MobileNet-V1 with both weights and activations quantized.

Method	$B_{w,\text{eff}}$	$B_{a,\text{eff}}$	Top-1 Acc. [%]	$\mathcal{C}_C$ [Gbit-OPs]
PACT [10, 17]	6	6	71.22	34.2
PACT [10, 17]	5	5	67.00	26.8
PACT [10, 17]	4	4	62.44	15.9
HAQ [17]	6	6	70.90	-
HAQ [17]	5	5	70.58	-
HAQ [17]	4	4	67.40	-
UNIQ [18]	8	8	67.50	52.4
UNIQ [18]	5	8	67.50	37.0
UNIQ [18]	4	8	66.00	31.9
RQ [19]	6	6	67.50	34.2
RQ [19]	5	5	61.50	26.8
DBQ* [8]	3	8	70.92	21.8
FP Baseline	32	32	71.84	-
FX8 Baseline	8	8	71.86	52.4
IMPQ	6	6	71.24	21.0
IMPQ	5	5	70.65	18.9
IMPQ	4	5.8	69.02	17.7

* nonlinear quantization

Comparisons with the State-of-the-Art: Table 2 summarizes the accuracy and effective precision of both weights and activations compared to other recent works. The techniques that use identical precision across the network, such as UNIQ, RQ, and PACT, result in significant accuracy drop with precision reduction. For example, PACT observes $> 7\%$ accuracy drop going from a 6-b quantization to a 4-b quantization. This accuracy drop with layer-wise quantization techniques such as HAQ is 3.5%, while IMPQ's kernel-wise quantization results in an accuracy drop of 2.22%.

5. CONCLUSIONS

This paper presents the IMPQ methodology to obtain mixed-precision networks with precision assignment at arbitrary granularity. IMPQ was validated with experiments on ResNet-20, VGG-Small for classifying the CIFAR-10 dataset, VGG-Small for classifying the SVHN dataset, and MobileNet-V1 for classifying the ImageNet-1k dataset. It was found to be most effective on compact networks and at lower precision. Next steps include extending IMPQ to DNN training.

6. REFERENCES

[1] Song Han, Jeff Pool, John Tran, and William Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 1135–1143.

[2] 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," *arXiv preprint arXiv:1704.04861*, 2017.

[3] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[4] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6848–6856.

[5] Gao Huang, Shichen Liu, Laurens Van der Maaten, and Kilian Q Weinberger, "Condensenet: An efficient densenet using learned group convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2752–2761.

[6] Fengfu Li, Bo Zhang, and Bin Liu, "Ternary weight networks," *arXiv preprint arXiv:1605.04711*, 2016.

[7] Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally, "Trained ternary quantization," *arXiv preprint arXiv:1612.01064*, 2016.

[8] Hassan Dbouk, Hetul Sanghvi, Mahesh Mehendale, and Naresh Shanbhag, "Dbq: A differentiable branch quantizer for lightweight deep neural networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 90–106.

[9] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou, "DoReFa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2016.

[10] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan, "PACT: Parameterized clipping activation for quantized neural networks," *arXiv preprint arXiv:1805.06085*, 2018.

[11] Charbel Sakr and Naresh Shanbhag, "An analytical method to determine minimum per-layer precision of deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1090–1094.

[12] Qian Lou, Feng Guo, Lantao Liu, Minje Kim, and Lei Jiang, "AutoQ: Automated kernel-wise neural network quantization," *arXiv preprint arXiv:1902.05690*, 2019.

[13] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer, "HAWQ: Hessian aware quantization of neural networks with mixed-precision," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 293–302.

[14] Bert Moons, Roel Uytterhoeven, Wim Dehaene, and Marian Verhelst, "ENVISION: A 0.26-to-10 TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm FDSOI," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2017, pp. 246–247.

[15] Hongyang Jia, Yinqi Tang, Hossein Valavi, Jintao Zhang, and Naveen Verma, "A microprocessor implemented in 65nm CMOS with configurable and bit-scalable accelerator for programmable in-memory computing," *arXiv preprint arXiv:1811.04047*, 2018.

[16] Jinmook Lee, Changhyeon Kim, Sanghoon Kang, Dongjoo Shin, Sangyeob Kim, and Hoi-Jun Yoo, "UNPU: A 50.6 TOPS/W unified deep neural network accelerator with 1b-to-16b fully-variable weight bit-precision," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2018, pp. 218–220.

[17] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han, "HAQ: Hardware-aware automated quantization with mixed precision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8612–8620.

[18] Chaim Baskin, Eli Schwartz, Evgenii Zheltonozhskii, Natan Liss, Raja Giryes, Alex M Bronstein, and Avi Mendelson, "Uniq: Uniform noise injection for non-uniform quantization of neural networks," *arXiv preprint arXiv:1804.10969*, 2018.

[19] Christos Louizos, Matthias Reisser, Tijmen Blankevoort, Efstratios Gavves, and Max Welling, "Relaxed quantization for discretized neural networks," *arXiv preprint arXiv:1810.01875*, 2018.

[20] Charbel Sakr, Yongjune Kim, and Naresh Shanbhag, "Analytical guarantees on numerical precision of deep neural networks," in *International Conference on Machine Learning*, 2017, pp. 3007–3016.

[21] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," Tech. Rep., Citeseer, 2009.

[22] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[25] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua, "LQ-nets: Learned quantization for highly accurate and compact deep neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 365–382.

[26] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 3123–3131.

[27] Aojun Zhou, Anbang Yao, Kuan Wang, and Yurong Chen, "Explicit loss-error-aware quantization for low-bit deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9426–9435.

[28] Song Han, Huizi Mao, and William J Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.