Improved Techniques for Quantizing Deep Networks with Adaptive Bit-Widths (Supplementary Material)

Ximeng Sun[1], Rameswar Panda[2,3], Chun-Fu (Richard) Chen[2,3], Naigang Wang[3], Bowen Pan[4], Aude Oliva[2,4], Rogerio Feris[2,3], Kate Saenko[1,2]
[1]Boston University, [2]MIT-IBM Watson AI Lab, [3]IBM Research, [4]MIT

1. Algorithm

We jointly train the network under all the quantization precisions. For each mini-batch, we apply our dynamic teacher selection strategy to choose a teacher from all the higher-precision candidates and swap in the teacher block to transfer knowledge to the lower precision. We gather losses from all precisions and then update the network. We summarize our proposed method **`CoQuant`** for quantizing deep networks with adaptive bit-widths in Algorithm 1.

Algorithm 1 **`CoQuant`**

Require:
A bit-width set $\mathcal{B}$ for Adaptive Deep Network Quantization.
A full-precision model $\mathcal{M}$.

for $t = 1, \cdots,$ epochs **do**
  Sample the current batch (x, y) from the dataset.
  $\mathcal{L} = 0$
  for $b \in \mathcal{B}$ **do**
    if $b = b_1$ **then**
      Feed-Forward: $y_b = \mathcal{M}_b(x)$
      $\mathcal{L}_b = \text{CE}(y_b, y)$
    else
      Choose the best teacher $\mathcal{M}_t$
      Feed-Forward, $y_b = \text{Swap}(\mathcal{M}_b, \mathcal{M}_t, p)(x)$
      $\mathcal{L}_b = \text{CE}(y_b, y) + \text{KL}(y_t, y_b)$
    end if
    $\mathcal{L} \leftarrow \mathcal{L} + \mathcal{L}_b$
  end for
  Back-propagation with the loss $\mathcal{L}$
end for

2. Dataset Details

We evaluate our method using both image classification (CIFAR10 [5] and ImageNet [7]) and video classification (ActivityNet [3] and Mini-Kinetics [1]) datasets. Below, we provide more details on the video classification datasets.

Dataset – Network	32-bit
CIFAR10 – ResNet18	95.3
CIFAR10 – MobileNet V2	94.1
ImageNet – ResNet18	69.9
ActivityNet – ResNet18	67.3
ActivityNet – ResNet50	69.8
Mini-Kinetics – ResNet18	66.3

Table 1: **Full-Precision Performance.**

ActivityNet. We use ActivityNet-v1.3, an untrimmed video datatset, with 10,024 videos for training, 4926 videos for validation and 5044 videos for testing and each video has an average duration of 117 seconds. It contains 200 different daily activities (*e.g.* drinking coffee, washing dishes, walking the dog etc.) and at least 100 untrimmed videos per class. In our paper, we train all models on the training set and test on the validation set since the test set labels are withheld by the authors. The dataset is publicly available to download at `http://activity-net.org/download.html`.

Mini-Kinetics. Kinetics-400 is a large-scale dataset containing 400 action classes and 240K training videos that are collected from YouTube. Since the full Kinetics dataset is quite large and the original version is no longer available from official site (about $\sim$15% videos are missing), we use the Mini-Kinetics dataset that contains 121K videos for training and 10K videos for testing, with each video lasting 6-10 seconds. We use official training/validation splits of Mini-Kinetics released by authors [6] in our experiments.

3. Implementation Details

In this section, we provide more implementation details regarding network architectures and initialization, full-precision performance and hyper-parameter selections. We will make our code publicly available after the acceptance.

Network Architectures and Initialization. We use the standard architectures for ImageNet, ActivityNet and Mini-Kinetics datasets. The input image is randomly cropped to 224×224 and randomly flipped horizontally. For CIFAR10

Dataset – Network	init p_1	λ	α_{init}	α_{lr}	α_{wd}	batch size	lr	lr scheduler
CIFAR10 – ResNet18	0.9	0.001	8	0.1	1e-4	128	0.1	MultiSteps (150, 225, 270)
CIFAR10 – MobileNet V2	0.5	0.001	8	0.01	4e-5	128	0.01	MultiSteps (82, 122)
ImageNet – ResNet18	0.7	0.001	8	0.1	6e-5	2880	0.01	MultiSteps (30, 60, 85, 95)
ActivityNet – ResNet18	0.5	0.001	4	0.01	5e-4	72	0.01	Cosine
ActivityNet – ResNet50	0.7	0.0001	4	0.1	5e-4	72	0.01	Cosine
Mini-Kinetics – ResNet18	0.7	0.001	2	0.1	1e-4	576	0.01	Cosine

Table 2: **Hyperparameters.**

dataset, we pad the input to 32x32 and adapt ResNet18 and MobileNet V2 as suggested [1]. We optimize 350 epochs for ResNet18 on CIFAR10 and 160 epochs for MobileNet V2 on CIFAR10. We follow [4] and initialize the network with the full precision model except CIFAR10 dataset since we found that low precision models in this setting do not converge.

Full-Precision Performance. In Table 1, we provide the 32-bit full-precision performance for 6 different {dataset, architecture} pairs that we use in the main paper to better show the performance of adaptive quantization network.

Hyper-parameters. In Table 2, we provide hyperparameters for 6 different {dataset, architecture} pairs that we use in the main paper. There are 3 separate groups of hyperparameters in our proposed **CoQuant**. (1) In dynamic teacher selection, we use λ in Eq. (4) to balance the prediction confidence and the model distance. We also define the initial value p_1, which is the probability of using the student block for the first layer. During the training, p_1 is linearly increased to 1 at the end. (2) We use a learnable uniform quantizer PACT [2] as our activation quantizer, whose performance depends on the initial value (α_{init}), learning rate (α_{lr}) and the weight decay (α_{wd}) of the clipping value α. (3) We also provide the batch size, learning rate and learning rate scheduler that we use to train the adaptive quantization network.

Code. Please refer to **"CoQuant_code.zip"** in supplementary material for our code submission. We will make the code publicly available after acceptance.

4. Ablation Study on λ in Eq. 4. (main paper)

We investigate effect of the hyperparameter λ on CIFAR10 (w/ ResNet18) and observe that $\lambda = 0$ obtains 98.3% while $\lambda = 1$ leads to 97.7% in $\Delta_{\mathcal{B}}$. We empirically set $\lambda = 0.001$ to balance the trade-off and achieves the best accuracy of 100.1% on CIFAR10 (same for other datasets).

5. Zero Shot Testing

In the main paper, we provide zero-shot testing in two scenarios: ResNet18 on CIFAR10 and ResNet50 on ActivityNet. In this section, we compare **CoQuant** with baselines in two other scenarios (see Table 3): ResNet18 on ActivityNet and ResNet18 on Mini-Kinetics. In both scenarios,

[1] https://github.com/kuangliu/pytorch-cifar

Methods	7-bit	5-bit	3-bit
ActivityNet – ResNet18			
Switchable BN	64.6	64.7	51.3
AdaBits (CVPR'20)	64.7	64.5	53.4
CoQuant (Ours)	65.5	65.1	59.3
Mini-Kinetics – ResNet18			
Switchable BN	64.2	64.1	48.7
AdaBits (CVPR'20)	64.1	63.7	44.9
CoQuant (Ours)	64.3	64.2	54.5

Table 3: **Zero Shot Testing with BN Calibration.** With ResNet18 on ActivityNet and ResNet18 on Mini-Kinetics, **CoQuant** achieves the best performance when evaluating with 7, 5, and 3 bits.

CoQuant outperforms Switchable BN and AdaBits [4] in all the 3 remaining quantization precisions.

6. Different Bit-width Combinations

In this section, we compare networks trained using different pre-defined bit-width sets to understand effect of individual bit-width on instant adaptation to different precisions. We evaluate each network with bit-widths from 2 to 8 bit. If the evaluated bit-width is missing from $\mathcal{B}$, we report its zero-shot testing result. From Table 4, we have the following observations. (1) If a network is trained under fewer bit-widths, the performance for the bit-width in $\mathcal{B}$ improves since it is easier for the network to handle fewer bit-widths during the training. (2) When a network is trained with $\mathcal{B} = \{8, 4\}$, 8-bit performance is much better than the one trained with $\mathcal{B} = \{8, 2\}$. It is because training with 4 bit is a closer task to training with 8 bit than training with 2 bit, which results in more positive knowledge transfer in the joint training. (3) When a network is trained with $\mathcal{B} = \{8, 6, 4, 2\}$, the middle-precisions achieve higher top-1 accuracy compared with the network trained with $\mathcal{B} = \{8, 2\}$, since the network is unable to keep the good performance for the middle precisions when the supervisions of all middle precisions are missing in $\mathcal{B}$. (4) 2 bit performance drops dramatically when the network is trained 8 and 4 bit which shows that it is very important to keep the lowest precision in $\mathcal{B}$ to achieve good performance for it during inference.

$\mathcal{B}$	$\{8, 6, 4, 2\}$	$\{8, 2\}$	$\{8, 4\}$
8-bit	95.2	95.2	95.6
7-bit	95.2	95.2	95.5
6-bit	95.4	95.2	95.5
5-bit	95.3	95.2	95.3
4-bit	95.1	94.5	95.5
3-bit	94.6	94.7	93.8
2-bit	94.1	94.5	73.2

Table 4: **Evaluation Results on 2 to 8 bits with different $\mathcal{B}$.** On CIFAR10, We train all-at-once quantization network with different pre-defined bit-width sets $\mathcal{B}$ and evaluate with 2 bit to 8 bit. We use zero-shot testing for all missing bit-widths.

Methods	8-bit	6-bit	4-bit	2-bit
DoReFa	65.2	65.0	64.5	59.7
Modified Weight Quantization	65.3	65.5	64.3	59.9

Table 5: **Comparison between Modified Weight Quantization and DoReFa.**.

7. Modified Weight Quantization

We compare our modified weight quantization with the original DoReFa [8] quantizer in Table 5 by performing individual quantization to a single precision (8-bit, 6-bit, 4-bit or 2-bit) on ActivityNet [3]. We did not observe bad effect on performance with our modified weight quantization. For all four precisions, the performance of modified weight quantization is $0.2\% - 0.5\%$ above or below the original DoReFa performance.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[2] 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.

[3] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[4] Qing Jin, Linjie Yang, and Zhenyu Liao. Adabits: Neural network quantization with adaptive bit-widths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2146–2156, 2020.

[5] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[6] Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and Rogerio Feris. Ar-net: Adaptive frame resolution for efficient action recognition. 2020.

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[8] 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.