LQ-Nets: Learned Quantization for Highly Accurate and Compact Deep Neural Networks
(Supplementary Material)

1 Statistics of Weights and Activations

In the main paper we presented the statistics of weights and activations in the ResNet-20 model quantized with “2/2” bits. Here we show the cases with “3/3” bit-widths in Fig. I.

Fig. I: Statistics of the weights (top row) and activations (bottom row) before (i.e., the floating-point values) and after quantization. The ResNet-20 model with “3/3” quantization is used. The orange diamonds indicate the eight quantization levels of our learned quantizers. Note that in the left figures for the floating-point values the histogram bins are of equal step size, whereas in the right figures each of the four bins contains all the values quantized to its corresponding quantization levels.

2 Detailed Hyper-Parameter and Other Setups

We presented here the detailed hyper-parameters and other training setups that are omitted in the main paper due to space limitation.
2.1 CIFAR-10 Experiments

Data augmentation: Following [4, 2], in the training stage we pad 4 pixels on each side of the original 32×32 images, and randomly crop a 32×32 sample or its horizontal flip. The original images are used at test time.

Hyper-parameters: For all the experiments on CIFAR-10, we train the models for up to 200 epochs and use a momentum of 0.9. For the ResNet-20 model, the learning rate starts at 0.1 and is divided by 10 at 82 and 123 epochs. Weight decay of $1e-4$ and batch size of 128 are adopted following the original paper. For VGG-Small, the learning rate starts at 0.02 and is divided by 10 at 80 and 160 epochs. Following [1], we set weight decay to $5e-4$ and batch size to 100.

2.2 ImageNet Experiments

Data augmentation: Our data augmentation strategy mostly follows [1]. During training, we first resize the shorter side of the images to 256, and then randomly sample 224×224 (227×227 for AlexNet) image crops with horizontal flipping applied at random. At test time, a single, centered crop of size 224×224 (227×227 for AlexNet) is used for each image. When training networks with bit-widths larger than “1/2”, we follow the augmentation strategy of the ResNet Torch implementation\(^1\). Specifically, we use the scale and aspect ratio augmentation from [5] and color augmentation proposed in [3].

Hyper-parameters: For all the experiments on ImageNet, following [2] we train the models for up to 120 epochs with a momentum of 0.9. For all the experiments with bit-widths larger than “1/2”, the batch size is 256 and the weight decay is $1e-4$. The learning rate starts at 0.1 and is divided by 10 at 30, 60, 85, 95, 105 epochs.

When comparing against HWGQ [1] on ResNet, AlexNet, VGG-Variant and GoogLeNet with bit-widths of “1/2”, we use the same hyper-parameters in HWGQ’s implementation. Specifically, the learning rate starts at 0.1 for ResNet and GoogLeNet, 0.01 for VGG-Variant, and 0.02 for AlexNet, respectively. Polynomial learning rate annealing with power of 1 is adopted instead of the multi-step annealing. The total training epoch is set to 64 for all experiments. The batch size is 128 for VGG-Variant and 256 for others. The weight decay is $5e-4$ for AlexNet and VGG-Variant, and $5e-5$ for ResNet and GoogLeNet.

References


\(^1\) https://github.com/facebook/fb.resnet.torch (accessed July 10, 2018)

