Supplementary Materials of McQueen: Mixed Precision Quantization of Early Exit Networks

BMVC 2023 Submission # 511

1 Appendix

1.1 Parametric Differentiable Quantizer (PDQ)

PDQ enables training of quantizer scaling factor α, quantizer threshold β and quantizer precision n during training. Given data to quantize x, the quantizer threshold β and scaling factor α, the quantized representation x_q is given by,

$$x_q = \alpha \cdot clip(\lfloor \frac{x}{\beta} \rceil, Q_n, Q_p) \tag{1}$$

where, $\lfloor \cdot \rceil$ is the round function, Q_n and Q_p are integer clipping bounds determined by the quantizer precision n. For signed x, $Q_p = \lfloor 2^{n-1} - 1 \rceil$ and $Q_n = \lfloor -2^{n-1} \rceil$; while for unsigned x, $Q_p = \lfloor 2^n - 1 \rceil$ and $Q_n = 0$. Gradient derivation for PDQ parameters is simplified when the quantizer is written as,

$$x_q = \begin{cases} \alpha Q_p & , \lfloor \frac{x}{\beta} \rceil \geq Q_p \\ \alpha \lfloor \frac{x}{\beta} \rceil & , Q_n < \lfloor \frac{x}{\beta} \rceil < Q_p \\ \alpha Q_n & , \lfloor \frac{x}{\beta} \rceil \leq Q_n \end{cases} \tag{2}$$

The gradient for scaling factor α is derived using,

$$\frac{\partial \mathcal{L}}{\partial \alpha} = \frac{\partial \mathcal{L}}{\partial x_q} \cdot \frac{\partial x_q}{\partial \alpha} \tag{3}$$

$\frac{\partial \mathcal{L}}{\partial x_q}$ is the layer weight or activation gradient obtained using Pytorch Autograd. Finally, $\frac{\partial x_q}{\partial \alpha}$ is obtained using eq 2 as follows,

$$\frac{\partial x_q}{\partial \alpha} = \begin{cases} Q_p & , \lfloor \frac{x}{\beta} \rceil \geq Q_p \\ \lfloor \frac{x}{\beta} \rceil & , Q_n < \lfloor \frac{x}{\beta} \rceil < Q_p \\ Q_n & , \lfloor \frac{x}{\beta} \rceil \leq Q_n \end{cases} \tag{4}$$

Similarly, gradient for quantizer threshold β is derived using,

$$\frac{\partial \mathcal{L}}{\partial \beta} = \frac{\partial \mathcal{L}}{\partial x_q} \cdot \frac{\partial x_q}{\partial \beta} \tag{5}$$

$\frac{\partial x_q}{\partial \alpha}$ is obtained using eq 2 as follows,

$$\frac{\partial x_q}{\partial \beta} = \begin{cases} 0 & , \lfloor \frac{x}{\beta} \rceil \geq Q_p \\ \frac{-\alpha x}{\beta^2} & , Q_n < \lfloor \frac{x}{\beta} \rceil < Q_p \\ 0 & , \lfloor \frac{x}{\beta} \rceil \leq Q_n \end{cases} \tag{6}$$

For signed data, gradients for precision n is derived using,

$$\frac{\partial \mathcal{L}}{\partial n} = \frac{\partial \mathcal{L}}{\partial x_q} \cdot \frac{\partial x_q}{\partial Q_p} \cdot \frac{\partial Q_p}{\partial n} + \frac{\partial \mathcal{L}}{\partial x_q} \cdot \frac{\partial x_q}{\partial Q_n} \cdot \frac{\partial Q_n}{\partial n} \tag{7}$$

$$= \frac{\partial \mathcal{L}}{\partial x_q} \cdot (2^{n-1} ln(2)) \cdot (\frac{\partial x_q}{\partial Q_p} - \frac{\partial x_q}{\partial Q_n}) \tag{8}$$

$$\tag{9}$$

$\frac{\partial x_q}{\partial Q_p}$ is derived using eq. 2 as follows,

$$\frac{\partial x_q}{\partial Q_p} = \begin{cases} \alpha & , \lfloor \frac{x}{\beta} \rceil \geq Q_p \\ 0 & , otherwise \end{cases} \tag{10}$$

Similarly, $\frac{\partial x_q}{\partial Q_n}$ is derived using eq. 2 as follows,

$$\frac{\partial x_q}{\partial Q_n} = \begin{cases} \alpha & , \lfloor \frac{x}{\beta} \rceil \leq Q_n \\ 0 & , otherwise \end{cases} \tag{11}$$

For unsigned data, $Q_n = 0$, therefore gradients for precision n is derived using,

$$\frac{\partial \mathcal{L}}{\partial n} = \frac{\partial \mathcal{L}}{\partial x_q} \cdot \frac{\partial x_q}{\partial Q_p} \cdot \frac{\partial Q_p}{\partial n} \tag{12}$$

$$= \frac{\partial \mathcal{L}}{\partial x_q} \cdot (2^n ln(2)) \cdot \frac{\partial x_q}{\partial Q_p} \tag{13}$$

$$\tag{14}$$

$\frac{\partial x_q}{\partial Q_p}$ is given by eq. 10.

1.2 Gradient similarity visualizations

This section provides additional visualizations on gradient similarity with exits placed at different intervals of the backbone model. The visualizations are shown in Fig. 1, Fig. 2 and Fig. 3. Following observations can be made from the figures,

1. Gradient masking considerably improves the similarity between gradients from exit classifiers and gradients from final classifier.
2. Often the similarity is lower than 0 for the initial layers if backbone model. First layer receives the most diverged gradient.
3. Exits added to earlier layers of the model have much severe impact in gradient similarity than exits added at later layers.

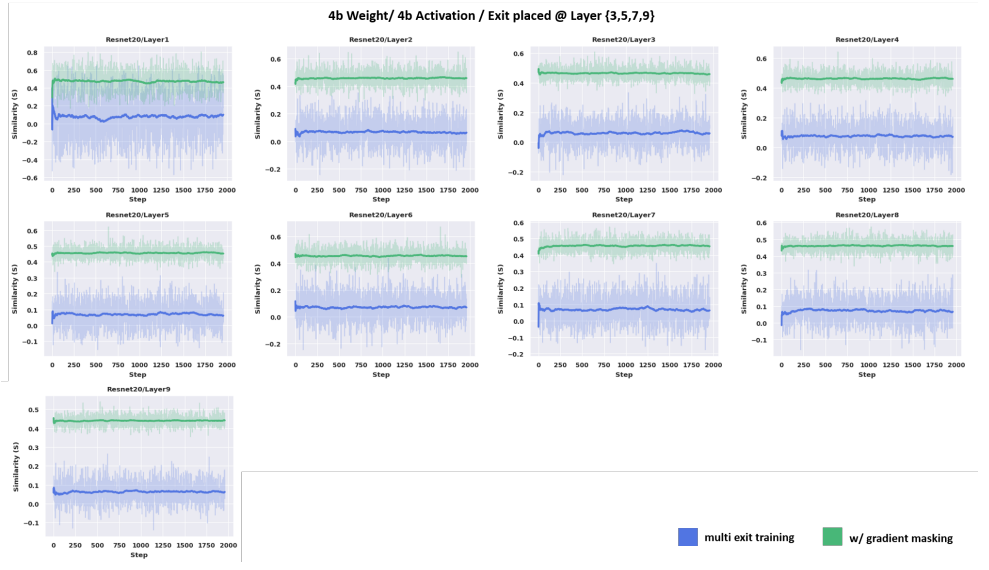

Figure 1: Gradient similarity for Resnet-20 with exits placed after layer 3,5,7,9. (Bold lines show the moving average)

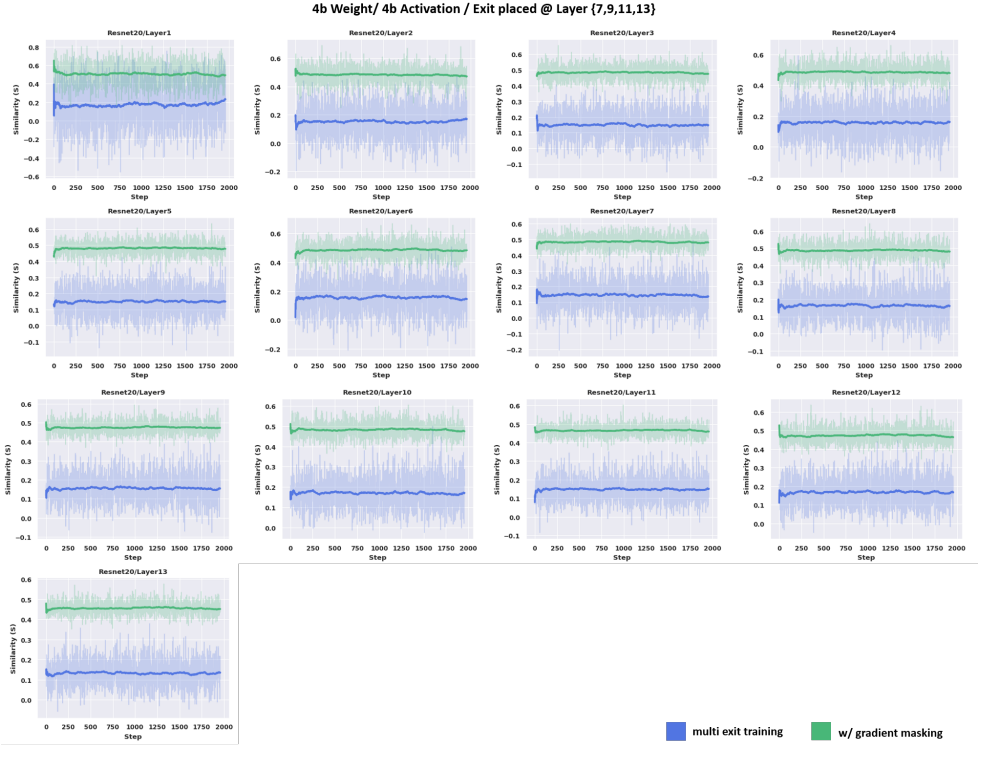

Figure 2: Gradient similarity for Resnet-20 with exits placed after layer 7,9,11,13 (Bold lines show the moving average)

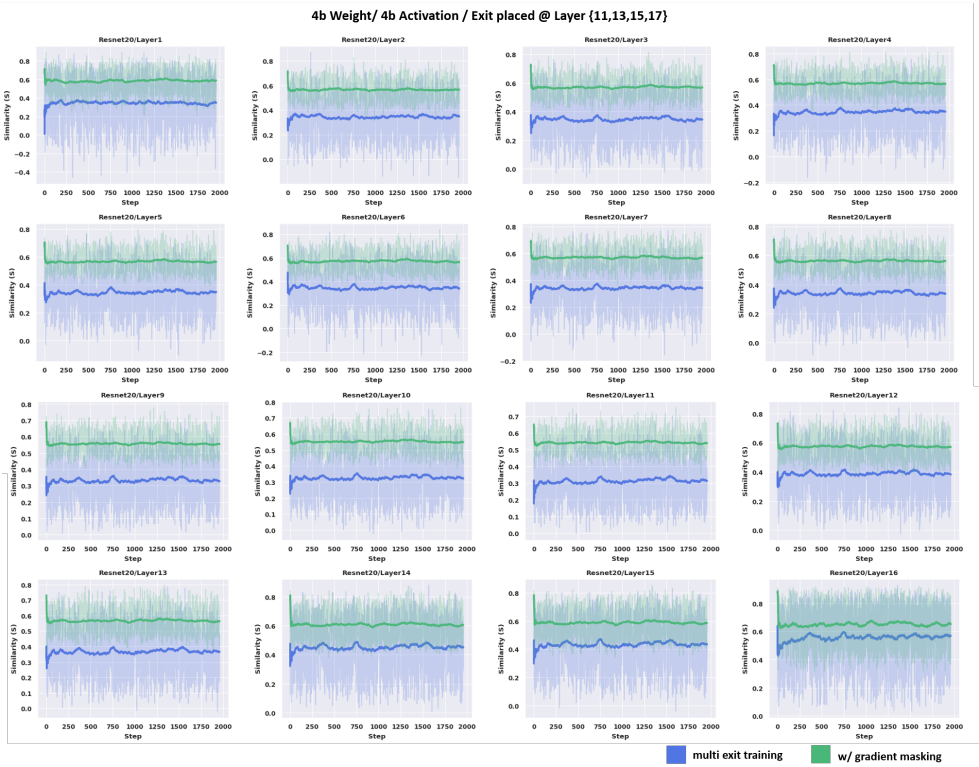

Figure 3: Gradient similarity for Resnet-20 with exits placed after layer 11,13,15,17 (Bold lines show the moving average)

Network (W/A)	ResNet18 trained on ImageNet 4/4				ResNet18 trained on ImageNet 3/3			
	Stage 1 (Full precision finetuning)	Stage 2 (Quantization Search)	Stage 3 (Quantized Finetuning)	Stage 4 (Training Ensemble Model)	Stage 1 (Full precision finetuning)	Stage 2 (Quantization Search)	Stage 3 (Quantized Finetuning)	Stage 4 (Training Ensemble Model)
Epoch	30	60	90	1	30	60	90	1
Batch size	512	512	512	512	512	512	512	512
Teacher	–	–	Resnet-101	–	–	–	Resnet-101	–
Optimizer	SGD w/ momentum	SGD w/ momentum	SGD w/ momentum	SGD w/ momentum	SGD w/ momentum	SGD w/ momentum	SGD w/ momentum	SGD w/ momentum
Initial lr	0.001	0.001	0.001	0.0001	0.001	0.001	0.001	0.0001
lr scheduler	Cosine	Cosine	Cosine	–	Cosine	Cosine	Cosine	–
Weight decay	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
Momentum	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
Random Crop	✓	✓	✓	✓	✓	✓	✓	✓
Random Flip	✓	✓	✓	✓	✓	✓	✓	✓
bop target	N/A	0	N/A	N/A	N/A	0	N/A	N/A
gamma	N/A	0.035	N/A	N/A	N/A	0.05	N/A	N/A

Table 1: Hyperparameters for Training ResNet18 on ImageNet

1.3 Hyperparameters for training

When comparing our results with the baselines in the main paper, we use the hyperparameters shown in Table 1. Generally, the hyperparameters are primarily same for all the training stages. We observed that lower learning rate provided better results when training the ensemble model during the fourth training stage. For stage 1,2,3 of training, cosine learning rate scheduling provided the best results.

1.4 Early Exit inference policy

We experiment with different confidence threshold values t and patience counter n and evaluate the inference performance (Fig. 4). We plot Acc v/s BOPs curve where BOPs are determined by number of samples exiting early. t and n impact number of samples exiting early affecting BOPs. We see that patience counter n of 1 provides higher accuracy at same BOPs.

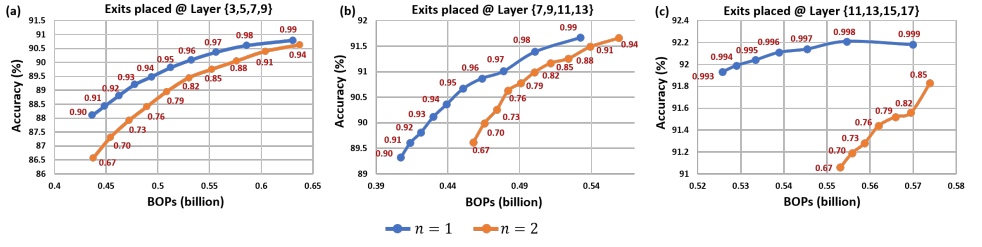

Figure 4: Accuracy v/s BOPs obtained for different confidence threshold t (shown in red) and patience counter n for 4-bit multi-exit ResNet-20 with exits placed at (a) Layer {3,5,7,9}, (b) Layer {7,9,11,13}, and (c) Layer {11,13,15,17}

1.5 Precision Assignment

We plot the evolution of weight and activation precision during training, shown in Fig. 5 and Fig. 6. Starting from 8-bit, the layer precisions decrease heavily during initial iterations and then the decrement slows down. Finally, the precisions are rounded to the nearest integer for subsequent training stages. We observe that linear layer precisions remain minutely perturbed. This is because the contribution to BOPs from linear layers is significantly low and consequently, the regularization penalty to reduce precision is low.

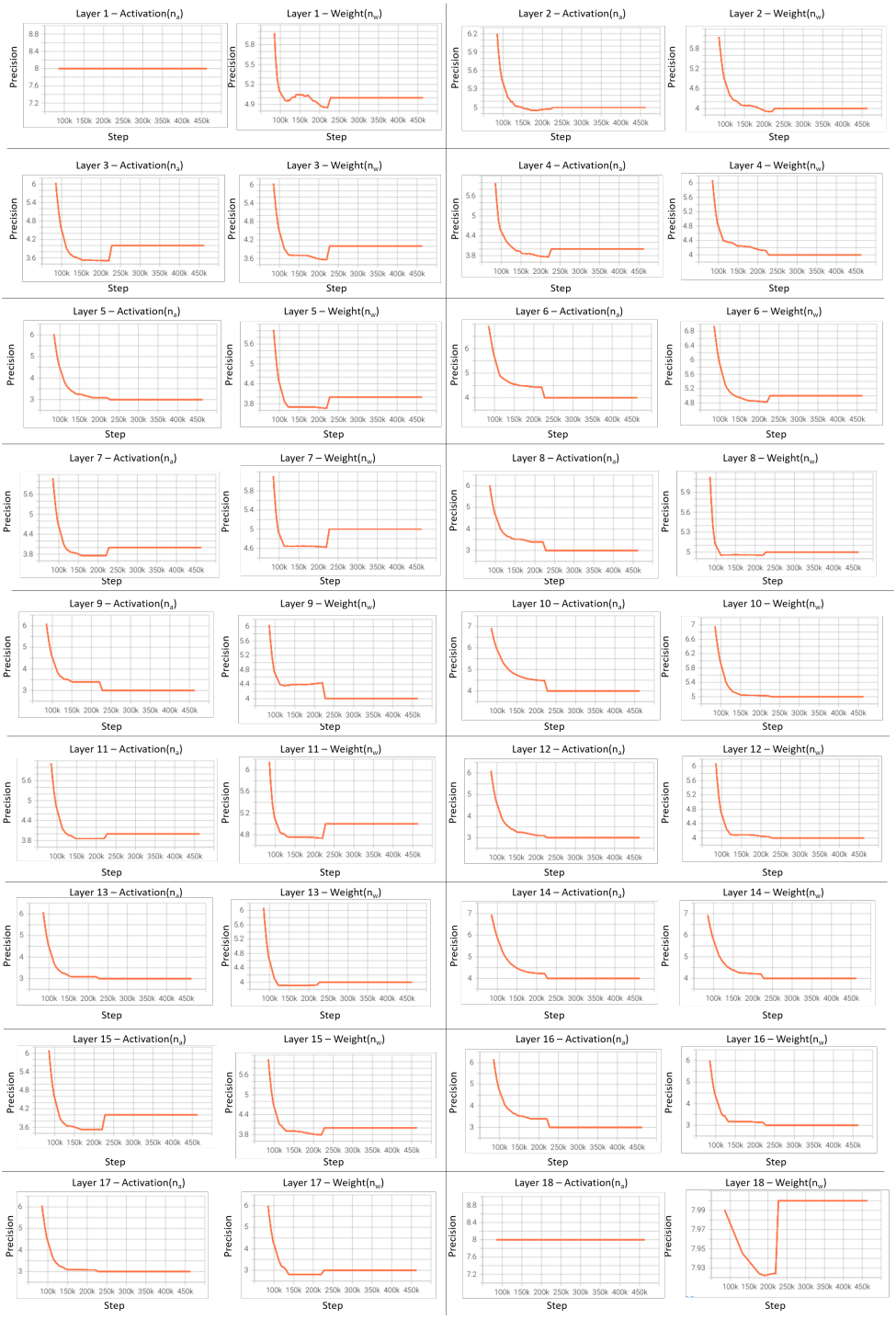

Figure 5: Precision evolution of 4-bit ResNet-18 model layers during training.

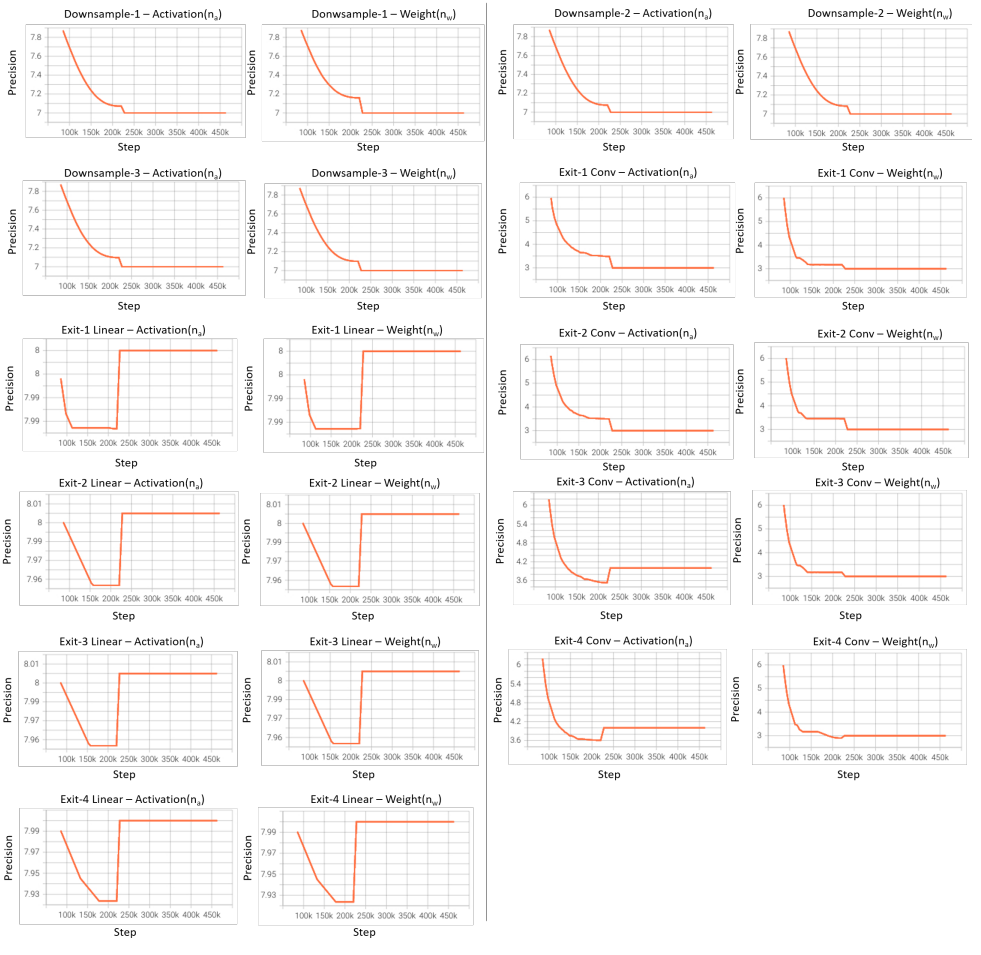

Figure 6: Precision evolution of 4-bit ResNet-18 model exit and downsample layers during training.