McQueen: Mixed Precision Quantization of Early Exit Networks

Utkarsh Saxena
saxenau@purdue.edu
Kaushik Roy
kaushik@purdue.edu

Department of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA

Abstract

Mixed precision quantization offers a promising way of obtaining the optimal trade-off between model complexity and accuracy. However, most quantization techniques do not support input adaptive execution of neural networks resulting in a fixed computational cost for all the instances in a dataset. On the other hand, early exit networks augment traditional architectures with multiple exit classifiers and spend varied computational effort depending on dataset instance complexity, reducing the computational cost. In this work, we propose McQueen, a mixed precision quantization technique for early exit networks. Specifically, we develop a Parametric Differentiable Quantizer (PDQ) which learns the quantizer precision, threshold, and scaling factor during training. Further, we propose a gradient masking technique that facilitates the joint optimization of exit and final classifiers to learn PDQ and network parameters. Extensive experiments on a variety of datasets demonstrate that our method can achieve significant reduction in Bit-Operations (BOPs) while maintaining the top-1 accuracy of the original floating-point model. Specifically, McQueen is able to reduce BOPs by 109x compared to floating-point baseline without accuracy degradation on ResNet-18 trained on ImageNet.

1 Introduction

The popularity of deep convolution neural networks (CNNs) can be attributed to their superhuman level performance in various computer vision and image processing tasks. Although CNNs can deliver remarkable performance, they often require millions of parameters and billions of floating-point operations (FLOPs), resulting in inefficient use of computational resources. To tackle this issue, researchers have proposed a range of methods, including network pruning [13], lightweight architecture design [15], dynamic execution [14, 18], low-rank compression [16] and quantization [4, 7]. Among these techniques, quantization gained popularity due to its effectiveness and simplicity. It targets efficiency by reducing the bit precision of weights and activations of the neural network, limiting them to a constrained set of values. Homogeneous quantization [4, 38] assigns the same precision to all the layers in a CNN while mixed precision quantization [33, 36] assigns different precisions, obtaining an improved tradeoff between CNN complexity and accuracy.

The majority of quantization techniques are limited to static CNNs, resulting in a fixed computational cost for all the test samples during inference. This leads to sub-optimal efficiency as the computational budget is over-provisioned for easy samples in the dataset [17].

To that effect, dynamic neural networks (DyNN) support input adaptive execution and expend varied computational effort depending on dataset instance complexity [10]. Among different DyNNs, early exit networks [11, 18, 25, 32, 39] have gained popularity due to their effectiveness in reducing the prediction time of CNNs. Early Exit networks achieve this by employing intermediate classifiers along the backbone CNN, which can quickly return predictions on easy samples, improving the latency and energy efficiency associated with CNN inference. However, the existing literature on early exit networks does not consider the impact of quantization and is only limited to full-precision networks [18, 32].

Unifying dynamic execution via early exit and mixed precision quantization can result in improved efficiency of CNNs. However, we observe that naive training of quantized early exit networks leads to a considerable drop in accuracy. To tackle this, we propose McQueen, a mixed precision quantization approach for early exit networks. McQueen unravels a paradigm of static precision dynamic depth networks where the layer-wise precision assignment is input independent (static), while the number of layers executed in a model is input dependent (dynamic). In summary, we make the following contributions, **(1)** We develop a Parametric Differential Quantizer (PDQ) which learns the optimal quantizer precision, threshold, and scaling factor during training using gradient descent. **(2)** We propose a gradient masking technique which masks gradients from exit classifiers to improve learning of the final classifier. **(3)** We evaluate McQueen on CIFAR-10 and Imagenet datasets and compare with the state-of-the art works on homogeneous and mixed precision quantization. **(4)** We implement our proposed technique on Bit-Fusion [26] accelerator which supports low-precision operations and evaluate the improvements achieved by McQueen.

2 Related works

Quantization. Quantizer design can be categorized into uniform quantization [4, 7, 8, 22, 38] and non-uniform quantization [20, 21, 34, 35, 37]. Among uniform quantization approaches, DoREFA-Net [38] developed a training methodology for CNNs with weights, activations, and gradients quantized. PACT [4] proposed a learned clipping function for quantized activations. LSQ [7] proposed a trainable step size for weights and activations quantization. More recently, N2UQ [22] developed a trainable quantizer where individual input thresholds are trained to better match the underlying data distribution. Among the literature in non-uniform quantization, LQ-Nets [37] is a seminal work that optimized non-uniform quantization levels based on a quantization error minimization algorithm.

Mixed Precision Quantization (MPQ). MPQ approaches can be categorized into search based, metric based, and optimization based. Search based techniques use neural architecture search [9, 33] for searching the quantization strategy. Metric based techniques propose a metric that acts as a proxy for deriving layer importance. Important layers are assigned higher precision compared to less important layers. HAWQ [6] uses trace of the Hessian matrix, Learned Layerwise Importance (LLI) [28] utilizes quantizer step size while OMPQ [24] uses layerwise orthogonality to assess layer importance. Finally, optimization based techniques leverage gradient descent to learn optimal weight and activation precision while minimizing the task loss [30, 31, 36]. More recently, DQ-Net [23] proposed dynamic precision networks where model precision is determined by the complexity of input sample.

Early Exit Networks. Early Exit networks support input adaptive execution and allow inference of input sample to terminate early saving computational effort. Early exit was first proposed by [25] as a conditional deep learning network. Since then there have been

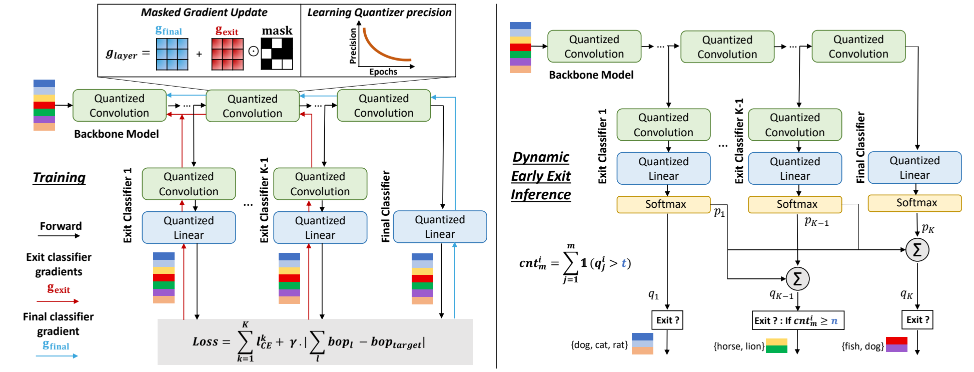

Figure 1: Overview of McQueen framework. (left) Training the quantized multi-exit model. (right) Dynamic Inference with early exits.

several works that focus on designing more advanced architectures amenable for early exit [3, 14, 29]. MSDNet [14] aims to provide prediction in case of insufficient computational resources. BranchyNet [29] augments AlexNet [27] with multiple classifiers and trains the augmented network from scratch. Authors in [3] stack multiple off-the-shelf CNNs to perform early exiting. Few works focus on designing the appropriate early exit policy [18, 39]. SDN [18] uses confidence of a prediction to perform early exits while PABEE [39] terminates inference after n consecutive same predictions. Further, Zero Time Waste (ZTW) [32] perform training of only exit classifiers and propose classifier cascading and ensembling to improve early exit accuracy. Alternatively, authors in [11] use meta-learning to derive optimal loss scaling parameters for optimizing multi-objective loss in early-exit architectures. The above mentioned works have been proposed for full-precision models and do not explore the interplay between early exit and quantization.

3 McQueen Framework

In this section, we describe the details of the proposed approach (Figure 1). We solve two challenges: 1) obtaining optimal layerwise precision assignments, and 2) achieving high accuracy with early exit classifiers. For the former, we develop a Parametric Differentiable Quantizer that learns the optimal precision values along with the quantizer scaling factor and threshold during training. For the latter, we develop gradient masking which selectively masks gradients from exit classifiers to achieve better learning of the final classifier.

3.1 Overview

Early Exit Network: The backbone model $\mathcal{F}_\theta$ with L layers comprises of a sequence of internal layers $\mathcal{F}_\theta^{(l)}$, for $l \in \{1,\dots,L\}$, with a final linear layer. $\mathcal{F}_\theta$ is converted to a K-exit network by attaching $K-1$ shallow exit classifiers at varying depths. Namely, let $\mathcal{G}_\phi^{(m,l)}$, for $m \in \{1,\dots,K-1\}$, be the m^{th} exit classifier attached to l-th hidden layer of the backbone network. In our framework, the exit classifier consists of a convolution layer, a pooling layer, and a linear layer. Given an input x, the output probabilities p_k from the k^{th} exit classifier is given by, $p_k = softmax(\mathcal{G}_{\phi}^{(m,l)}(\mathcal{F}_{\theta}^{(l)}(x)))$. Similarly, the output probability produced by the final classifier (K^{th} exit) is given by, $p_K = softmax(\mathcal{F}_{\theta}^{(L)}(x))$. Additionally, following the ZTW method [32], we present an ensemble version of the framework that utilizes a group of exit classifiers to determine the output probability. In particular, the probabilities from preceding exit classifiers are used to improve the accuracy of the current classifier. The probability of class i in the k^{th} ensemble is given by,

$$q_k^i = \frac{1}{Z_k} \cdot b_k^i \cdot \exp(\Sigma_{j=1}^k w_k^j * ln(p_j^i)) \tag{1}$$

Where bias b_k^i, weight w_k^j (for $j = 1,...,k$) are trainable parameters, p_j^i is the probability of i^{th} class at exit classifier j and Z_m is the normalization factor to ensure $\Sigma_i q_m^i = 1$.

Training the Multi-Exit Model: The K classifiers in the K-exit network are trained together to minimize the cumulative training loss. Several sophisticated techniques [11, 18] have been proposed to appropriately weigh loss from each classifier to obtain the training loss. We consider the most simplistic scenario where the loss of each classifier is given a unit weight. The training loss for the K-exit network is given by, $\mathcal{L}_{CE} = \Sigma_{k=1}^K l_{CE}^k$, where l_{CE}^k is the cross-entropy loss of the k^{th} classifier. Note that the gradients obtained from minimizing l_{CE}^k are used to update parameters in the exit classifier as well as the backbone model.

Dynamic Early Exit Inference: During inference, forward propagation through the K-exit network is terminated when the exit policy is satisfied, saving the computational cost of executing subsequent layers. Inspired by SDN [18] and PABEE [39], we design our exit policy such that early exit occurs when n exit classifiers provide the same predictions with confidence greater than a predetermined threshold t. Formally, exit will occur at the k^{th} exit classifier, when the prediction counter $cnt_k^i \geq n$, where $cnt_k^i = \Sigma_{j=1}^k \mathbb{1}(p_j^i > t)$. The thresholds n and t are determined using a validation set after training the model. For the case of the ensemble, an early exit decision is made using the ensemble probability q_k^i(eq. 1) instead of p_k^i. The final classifier classifies the samples that did not exit early.

3.2 Parametric Differentiable Quantizer

Given data to quantize x, the quantizer threshold β and scaling factor α, the quantized representation x_q is given by,

$$x_q = \alpha \cdot clip(\lfloor \frac{x}{\beta} \rceil, Q_n, Q_p) \tag{2}$$

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 $Q_n = 0$. The backward pass through the quantizer is derived using straight through estimator (STE) [2]. This approximates the gradient through the round function by treating it as a pass through operation. It is given by,

$$\frac{\partial x_q}{\partial x} = \begin{cases} \frac{\alpha}{\beta} & , Q_n < \lfloor \frac{x}{\beta} \rceil < Q_p \\ 0 & , otherwise \end{cases} \tag{3}$$

The PDQ quantizer is a modification to LSQ [7] quantizer (where $\alpha = \beta$) with scaling factor and thresholds decoupled. PDQ enables learning of α, β and n during CNN training (more details in Appendix A.1). Learning α and β separately enables the quantizer to match distribution of full precision data x with higher fidelity. Note that although thresholds and scaling factor are decoupled, the quantizer is still a uniform quantizer. The scaling factor α and threshold β is learned by introducing the following gradient:

$$\frac{\partial x_q}{\partial \alpha} = \begin{cases} Q_p & , \lfloor \frac{x}{\beta} \rceil \geq Q_p \\ \lfloor \frac{x}{\beta} \rceil & , Q_n < \lfloor \frac{x}{\beta} \rceil < Q_p \\ Q_n & , \lfloor \frac{x}{\beta} \rceil \leq Q_n \end{cases} \quad \Bigg| \quad \frac{\partial x_q}{\partial \beta} = \begin{cases} 0 & , \lfloor \frac{x}{\beta} \rceil \geq Q_p \\ \frac{-\alpha \cdot x}{\beta^2} & , Q_n < \lfloor \frac{x}{\beta} \rceil < Q_p \\ 0 & , \lfloor \frac{x}{\beta} \rceil \leq Q_n \end{cases} \tag{4}$$

Lastly, the quantizer precision n is learned by obtaining gradients from the clipping bound Q_n and Q_p. For signed data x,

$$\frac{\partial x_q}{\partial n} = 2^{n-1} ln(2) \cdot \{\frac{\partial x_q}{\partial Q_p} - \frac{\partial x_q}{\partial Q_n}\} \tag{5}$$

$$\frac{\partial x_q}{\partial Q_p} = \begin{cases} \alpha & , \lfloor \frac{x}{\beta} \rceil \geq Q_p \\ 0 & , otherwise \end{cases} \quad \Bigg| \quad \frac{\partial x_q}{\partial Q_n} = \begin{cases} \alpha & , \lfloor \frac{x}{\beta} \rceil \leq Q_n \\ 0 & , otherwise \end{cases} \tag{6}$$

While for unsigned data x,

$$\frac{\partial x_q}{\partial n} = 2^n ln(2) \cdot \{\frac{\partial x_q}{\partial Q_p}\} \quad \Bigg| \quad \frac{\partial x_q}{\partial Q_p} = \begin{cases} \alpha & , \lfloor \frac{x}{\beta} \rceil \geq Q_p \\ 0 & , otherwise \end{cases} \tag{7}$$

3.2.1 Training with Quantization strategy

The primary objective of CNN quantization is to improve computational efficiency when CNN is deployed on mobile devices. This requires constraining the precision values n to enable efficiency. To that effect, we add a regularization term that attempts to reduce n. Our choice of regularization term is based on restricting bit-wise operations (BOPs) of the quantized model and is given by,

$$\mathcal{L}_{bop} = |\Sigma_l \text{bop}_l(n_w, n_a) - \text{bop}_{target}| \quad \Big| \quad \text{bop}_l = n_w \cdot n_a \cdot k_x \cdot k_y \cdot C \cdot K \cdot H \cdot W \tag{8}$$

where bop_{target} is the target computational cost provided by the user, n_w, n_a are precision of weights and activations in a layer, k_x, k_y are kernel width and height, K, C are output and input channels and H, W are output feature map height and width. The precision values are learned to minimize the training loss and the regularization loss. The training with PDQ is divided into two parts, 1) Quantization search and 2) Fine-tuning. Precision values are updated during quantization search and remain frozen during fine-tuning. Since n_w and n_a is a parameter that is updated during training, they can assume any floating point value (2.72-bit, for instance). The forward and backward pass through the quantizer works on the floating point value of precision. After appropriate quantization precision has been searched to meet the desired BOP target, the floating point precision values are rounded to the nearest integer, and the model is further fine-tuned.

3.3 Gradient Masking

Training a multi-exit model involves finding the optimal parameter values which minimize the multi objective training loss. This means that gradient steps are taken to minimize the overall training loss which impacts the learning of final classifier. Accuracy of final classifier is important since all the samples which do not exit early need to be classified by it. We observe that multi-exit training often reduces accuracy of the final classifier. (Table 1). Adding

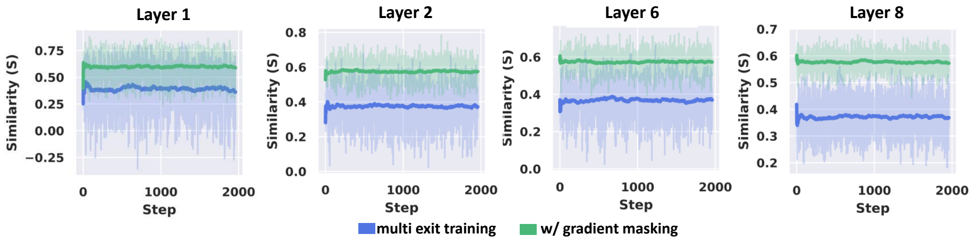

Figure 2: Gradient similarity for Resnet-20 (Bold lines show the moving average)

early exits motivates separability of class-wise features at early layers of the backbone model while final classifier demands class-wise separability at the final layer creating a conflict. To analyze this, we observed the gradients obtained by minimizing loss at the final classifier and those obtained by minimizing losses of exit classifiers. Let $\mathbf{g_{exit}} = \nabla_w \Sigma_{k=1}^{K-1} l_{CE}^k(w)$ be the gradient obtained from exit loss while $\mathbf{g_{final}} = \nabla_w l_{CE}^K(w)$ be the gradient obtained from final classifier loss. We evaluate the cosine similarity (S) between $\mathbf{g_{exit}}$ and $\mathbf{g_{final}}$ gradients across training. A high value of S (closer to 1) implies high alignment while a lower value implies increased conflict between gradients. Figure 2 shows the gradient similarity across training steps for different layers of ResNet-20 model [12]. We observe that the similarity between gradients obtained from multi-exit training is low, which manifests itself as accuracy degradation of the backbone model. Also, gradient similarity at the first layer is often less than zero, implying that multi-exit training prevents learning of the backbone classifier. More visualizations of gradient similarity are presented in Appendix A.2.

Based on these observations, we propose gradient masking to preserve high similarity between $\mathbf{g_{exit}}$ and $\mathbf{g_{final}}$. In particular, the overall layer gradient is given by, $\mathbf{g_{layer}} = \mathbf{g_{final}} + mask \odot \mathbf{g_{exit}}$. The mask is given as,

$$mask = \begin{cases} 1 & , sgn(\mathbf{g_{exit}}) = sgn(\mathbf{g_{final}}) \\ 0 & , otherwise \end{cases} \tag{9}$$

Table 1: Accuracy degradation of final classifier for 2bit model.

Method	Dataset	w/o Early Exit	w Early Exit
ResNet-20	CIFAR-10	89.27	88.19
ResNet-18	ImageNet	67.6	66.4

where sgn is the sign function. We apply $\mathbf{g_{exit}}$ for a particular weight element only when its sign matches with the sign of $\mathbf{g_{final}}$. This ensures that exit gradients are always aligned with final classifier gradients and do not conflict with learning of the final classifier. Figure 2 shows that similarity between gradients is greatly improved when gradient masking is applied. Gradient masking prioritizes learning of final classifier over exit classifiers since some gradient updates in $\mathbf{g_{exit}}$ are set to 0. This leads to higher accuracy of final classifier but at the cost of slightly lower accuracy of exit classifiers. However, we observe and show later in Section 4.1, that gradient masking has an overall improved effect on BOPS vs accuracy tradeoff in the presence of dynamic early exits.

3.4 Training

We divide the total training effort of the quantized multi-exit model into 4 stages. The backbone model is chosen to be off-the-shelf full precision pre-trained model while exit classifiers are randomly initialized. **First** (*Full precision fine-tuning*): the full precision multi-exit

Table 2: Accuracy with different positions of exit classifiers and impact of gradient masking.

Exit Positions	Gradient Masking	top-1 accuracy @ 4bW/4bA					
		Final classifier	Exit #1	Exit #2	Exit #3	Exit #4	Early Exit Accuracy
Exits @ Layer {3,5,7,9}	✗	90.5 ± 0.10	66.9 ± 0.26	72.6 ± 0.28	75.6 ± 0.57	83.4 ± 0.31	89.9 ± 0.09
	✓	91.1 ± 0.20	62.8 ± 1.95	70.9 ± 1.19	73.6 ± 0.45	83.1 ± 0.35	90.4 ± 0.21
Exits @ Layer {7,9,11,13}	✗	91.3 ± 0.14	72.6 ± 0.41	83.8 ± 0.47	87.0 ± 0.43	88.3 ± 0.38	91.0 ± 0.19
	✓	91.6 ± 0.14	73.2 ± 0.53	83.6 ± 0.11	86.7 ± 0.09	88.1 ± 0.25	91.3 ± 0.17
Exits @ Layer {11,13,15,17}	✗	92.2 ± 0.07	86.8 ± 0.08	88.5 ± 0.09	90.9 ± 0.08	91.7 ± 0.08	92.0 ± 0.07
	✓	92.2 ± 0.09	86.3 ± 0.14	88.1 ± 0.37	90.8 ± 0.17	91.5 ± 0.11	92.1 ± 0.16

model is fine-tuned to train randomly initialized exit classifiers. The fine-tuned model acts as an initialization point for the next stage. **Second** (*Quantization search*): this stage involves training the model while searching for optimal weight and activation precision. The precision for every layer is initialized to 8-bit before starting the quantization search. The model parameters along with PDQ parameters are trained to meet the target BOPs constraint. The total loss to be minimized is given by $\mathcal{L}_{total} = \mathcal{L}_{CE} + \gamma \cdot \mathcal{L}_{bop}$. γ is a hyperparameter to control the relative weight between the cross-entropy and regularization loss. **Third** (*Quantized Finetuning*): the quantizer precisions obtained after second stage are rounded to the nearest integer value and remain frozen for the remaining training effort. In this stage, the model is fine-tuned further to achieve high accuracy with the modified and frozen quantizer precisions. **Fourth** (*Training ensemble model*): the ensemble model is trained to achieve higher accuracy of classifiers by reusing predictions made by preceding classifiers (Sec.3.1). In this stage, the multi-exit model is frozen and only the weight and bias of the ensemble model are trained.

4 Experiments

We analyze the design choices of McQueen on CIFAR-10 [19] and compare the framework with state-of-the-art baselines on ImageNet [5]. For CIFAR-10 we use ResNet-20 model as the backbone while for ImageNet, we use ResNet-18 model as the backbone with exits placed after layers 9, 11, 13, and 15. The backbone models are initialized with a full precision pretrained model obtained from the TorchVision model zoo repository [1] while exit classifiers are randomly initialized. We present results with ensembling (named McQueen-ensemble) and without ensembling (named McQueen) on ImageNet. The hyperparameters for training the models are provided in Appendix A.3.

4.1 Results on CIFAR-10

We study the impact of positioning exit classifiers at various depths on accuracy. Columns 3-7 in Table 2 show the accuracies of the classifiers when each of them is evaluated on the entire test set. While column 8 in Table 2 indicates the classification accuracy with early exits which will be referred to as *early exit accuracy* (using the exit policy described in sec. 3.1, $n = 2$, $t = 0.9$). We consider three scenarios with exit classifiers attached at 1) early layers (layers 3,5,7,9), 2) middle layers (layers 7,9,11,13), and, 3) later layers (layers 11,13,15,17). Exits attached at early layers have fewer parameters to update and hence achieve a low classification accuracy. Interestingly, the positioning of exit classifiers impacts the accuracy of the final classifier, an artifact of varied gradient similarity between final and exit classifiers. Final classifier accuracy is highest when exits are added to later layers of the backbone model.

Further, we analyze the impact of gradient masking on individual and early exit accuracies. We observe that incorporating gradient masking improves the accuracy of the final classifier by 0.6% and 0.3% when exits are placed at early and middle layers respectively. For the case of exits placed at later layers, similarity between exit and final classifier gradients is high and hence improvements with gradient masking are not significant. The enhanced performance of the final classifier with gradient masking comes at the cost of reduced accuracy of exit classifiers. However, the early exit accuracy obtained with gradient masking still remains high as shown in Table 2. Further, we evaluate the early exit performance of the trained models at different confidence thresholds (t) at prediction counter threshold $n = 2$ leading to varied BOPs (Figure 3). Results on varying n are shown in Appendix A.4. We sweep the value of t which manifests as varied number of early exits impacting BOPs. Here, a lower threshold increases the number of early exits reducing BOPs. For iso-BOPs, higher early exit accuracy is obtained for models trained with gradient masking. Figure 3 demonstrates that gradient masking achieves better accuracy to BOPs tradeoff compared to conventional multi-exit training.

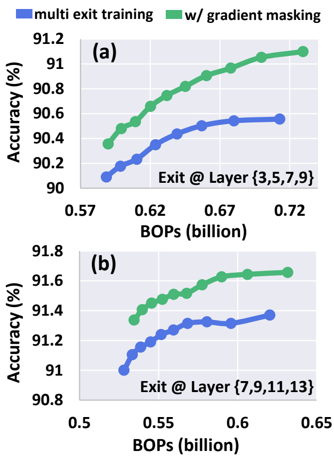

Figure 3: Accuracy vs BOPs.

4.2 Results on ImageNet

Comparison with homogeneous quantization: For a fair comparison, we homogeneously quantize all the layers of multi-exit model to the same precision. Inputs to the model and weights, activations of linear layers are set to be 8bit while remaining layers are quantized to the precision given in Table 3(left). Since layer precisions are predetermined, the second training stage involving quantization search (sec 3.4) is skipped. Related works present results with varied floating-point baselines, therefore, we compare accuracy degradation from the floating-point baseline (delta). McQueen achieves considerably fewer BOPs with lesser degradation of classification accuracy (Table 3 left). Compared to the recent N2UQ [22] method, we achieve 4.76 billion lesser BOPs for the same accuracy degradation under 2bit quantization. McQueen-ensemble improves accuracy by 0.7% and 0.3% in 2bit and 3bit models without ensembling respectively. For 3bit model, McQueen-ensemble achieves 0.3% higher accuracy than the floating-point baseline.

Comparison with mixed precision quantization: We achieve lesser BOPs with lower accuracy degradation compared to related works as shown in Table 3 (right). For both 3bit and 4bit mixed precision multi-exit ResNet-18, we achieve improved delta with McQueen and McQueen-ensemble extends the improvements further. Our 4bit ResNet-18, achieves 0.7% higher than full precision baseline and 1.1% higher with ensembling. Compared with the most performant baseline LLI [28], we achieve 1.0% (0.5%) higher accuracy with (without) emsembling on 3bit model and 0.7%(0.3%) higher accuracy with (without) ensembling on 4bit model. Additionally, for the McQueen-ensemble model, we lower exit policy thresholds until the accuracy of the model matches that of LLI and correspondingly, we obtain 2.02 billion and 3.3 billion lesser BOPs. Compared to DQ-Net [23] which supports input adaptive execution, we achieve 0.2% and 0.4% higher accuracy at 4.03 billion and 10.19 billion lower

Table 3: Comparison of various methods with ResNet-18 trained on ImageNet. Homogeneous quantization (left) and mixed precision quantization (right).

Method (homogeneous)	Precision (W/A)	top-1	Delta	BOPs (billion)	FP top-1
DoReFa [38]	2/2	64.7	-5.0	14.36	69.7
PACT [illegible]	2/2	64.4	-5.8	14.36	70.2
LSQ [7]	2/2	67.6	-2.9	14.36	70.5
LQ-Net [37]	2/2	64.9	-5.4	14.36	70.3
DSQ [8]	2/2	65.2	-4.7	14.36	69.9
N2UQ [22]	2/2	69.4	-2.4	14.36	71.8
McQueen	**2/2**	**66.7**	**-3.0**	**9.38**	**69.7**
McQueen-ensemble	**2/2**	**67.4**	**-2.3**	**9.48**	**69.7**
DoReFa [38]	3/3	67.5	-2.2	22.84	69.7
PACT [illegible]	3/3	69.2	-1.0	22.84	70.2
LSQ [7]	3/3	70.2	-0.3	22.84	70.5
LQ-Net [37]	3/3	68.2	-2.1	22.84	70.3
DSQ [8]	3/3	68.7	-1.2	22.84	69.9
N2UQ [22]	3/3	71.9	0.1	22.84	71.8
McQueen	**3/3**	**69.7**	**0.0**	**17.0**	**69.7**
McQueen-ensemble	**3/3**	**70.0**	**0.3**	**17.0**	**69.7**

Method (mixed)	Precision (W/A)	top-1	Delta	BOPs (billion)	FP top-1
SPOS [9]	$3_{MP}/3_{MP}$	69.4	-1.5	21.92	70.9
DNAS [33]	$3_{MP}/3_{MP}$	68.7	-2.3	24.34	71.0
FracBits-SAT [36]	$3_{MP}/3_{MP}$	69.4	-0.8	22.93	70.2
LLI [28]	$3_{MP}/3_{MP}$	69.0	-0.6	23.02	69.6
DQ-Net [23]	$4_{MP}/4_{MP}$	69.8	0.0	27.18	69.8
McQueen	$\mathbf{3_{MP}/3_{MP}}$	**69.5**	**-0.2**	**22.64**	**69.7**
McQueen-ensemble	$\mathbf{3_{MP}/3_{MP}}$	**70.0**	**0.3**	**23.15**	**69.7**
McQueen-ensemble	$\mathbf{3_{MP}/3_{MP}}$	**69.0**	**-0.7**	**21.0**	**69.7**
SPOS [9]	$4_{MP}/4_{MP}$	70.5	-0.4	31.81	70.9
DNAS [33]	$4_{MP}/4_{MP}$	70.6	-0.4	35.17	71.0
FracBits-SAT [36]	$4_{MP}/4_{MP}$	70.6	0.4	34.7	70.2
LLI [28]	$4_{MP}/4_{MP}$	70.1	0.5	33.05	69.6
DQ-Net [23]	$5_{MP}/5_{MP}$	70.4	0.6	42.49	69.8
McQueen	$\mathbf{4_{MP}/4_{MP}}$	**70.4**	**0.7**	**32.18**	**69.7**
McQueen-ensemble	$\mathbf{4_{MP}/4_{MP}}$	**70.8**	**1.0**	**32.3**	**69.7**
McQueen-ensemble	$\mathbf{4_{MP}/4_{MP}}$	**70.1**	**0.4**	**29.7**	**69.7**

BOPs with McQueen-ensemble for 3bit and 4bit models respectively.

4.3 Analysis

Contribution of EE: We analyze the impact of early exit in reduction of BOPs on top of reduction already achieved by quantization. Table 4 shows BOPs with early exit (EE) and without EE (samples exiting at final classifier) for ResNet-18 models at different precisions. We see that dynamic early exits contribute to a 17-24% reduction in BOPs without loss in accuracy. Additionally, we show percentage of samples exiting at each exit classifier. Recall that samples exiting early are determined by the exit policy (confidence threshold t and prediction counter n) which is chosen based on a validation set. The values of n and t are determined such that BOPs are reduced without any degradation in accuracy. Further efficiency may be obtained by choosing a more aggressive exit policy albeit at the cost of accuracy.

Table 4: Impact of early exit BOPs of ResNet-18 model.

Precision	**w/o EE (BOPs/top-1)**	**w EE (BOPs/top-1)**	**Improv. w/ EE**	**Samples exiting at each exit (%)**
2/2	11.3/66.8	9.3/66.7	17.7%	0 / 21.9 / 17.2 / 9.9 / 50.9
3/3	22.4/69.7	17.0/69.7	24.1%	29.4 / 20.2 / 9.4 / 9.4 / 31.6
$3_{MP}/3_{MP}$	27.8/69.6	22.6/69.5	18.7%	0 / 29.2 / 8.8 / 11.43 / 50.62
$4_{MP}/4_{MP}$	39.0/70.5	32.2/70.4	17.4%	0 / 20.9 / 16.1 / 9.9 / 53.1

Overheads with multi-exit architecture. Exit classifiers incur an additional parameter overhead. Table 5 lists down additional parameter overhead for ResNet-18 due to the presence of exit classifiers. For 3bit homogenous quantized model, exit classifiers amount to 53.9% storage overhead most of it coming from linear layers. Since linear layers have a minor contribution to total model BOPs, the BOP regularization term for linear layers is low causing the learned precision to be high (8bits in our simulations), which causes a significant storage overhead. One possible solution to reduce storage overhead would be to incorporate more regularization penalties which minimize the model storage size in addition to minimizing model BOPs.

Table 5: Storage overhead.

Precision (W/A)	Size backbone	Size multi-exit	Overhead (%)
3/3	4.48MB	6.89MB	53.9%
$4_{MP}/4_{MP}$	5.46MB	7.88MB	44.2%

Table 6: Peformance on Bit-Fusion

Method	Precision (W/A)	top-1 (%)	Latency (s)	Energy (J)
Dorefa	4/8	69.8	2.565	1.098
McQueen	$\mathbf{3_{MP}/4_{MP}}$	**69.8**	**1.605**	**0.71**

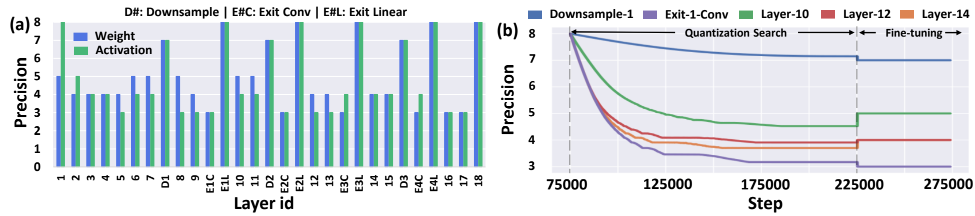

Figure 4: Layerwise precisions learned for 4bit ResNet-18 model. (a) Learned weight and activation precisions, and (b) Evolution of precisions across training steps.

Precision assignments: The weight and activation precision for 4bit mixed precision ResNet-18 model is shown in Figure 4(a). We observe that often activations are assigned lower precision than weights in the same layer. Figure 4(b), shows the evolution of precision for selected layers of the 4bit model during training. Starting from 8bit, the precision decreases heavily at the start due to the high regularization penalty and the decrement smooths down later into the training. Finally, the precision is rounded to the nearest integer value and remains frozen for the remaining training effort. Additional results are presented in Appendix A.5.

Hardware Performance: We conducted experiments to evaluate the hardware efficiency of McQueen. In Table 6 we evaluate our model on Bit-Fusion [26] accelerator which supports low precision operations. Bit-Fusion only supports 2,4,8,16 bit operations, therefore, we round the precision of our model to the nearest supported value after quantization search. Our multi-exit Resnet-18 model achieves higher accuracy than the baseline with much lower energy and latency on the entire ImageNet test set.

5 Conclusion

We have presented McQueen, which performs mixed precision quantization of early exit models. The overarching goal is to achieve a significant reduction in CNN computational cost while minimizing the degradation of CNN accuracy. We achieve this by combining parameter quantization with dynamic early exits. The layers in a CNN are quantized to low precision values while the number of layers executed dynamically depends on input sample complexity. We develop PDQ which automatically learns optimal weight and activation precision during training. Further, we propose gradient masking which achieves high accuracy with multi-exit training. McQueen achieves the lowest computational cost (BOPs) with lower degradation in accuracy compared to state-of-the-art baselines. Additionally, we implement the design on a hardware accelerator and evaluate the improvements achieved.

Acknowledgement

This work was supported in part by, Center for Brain Inspired Computing (C-BRIC), a DARPA sponsored JUMP center, Semiconductor Research Corporation (SRC), National Science Foundation, Intel, the DoD Vannevar Bush Fellowship, and the U.S. Department of Energy, Office of Science, for support of microelectronics research, under Grant DE-AC02-06CH11357.

References

[1] Models and pre-trained weights. URL https://pytorch.org/vision/stable/models.html.

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[3] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, pages 527–536. PMLR, 2017.

[4] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks, 2018.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] 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/CVF International Conference on Computer Vision*, pages 293–302, 2019.

[7] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. In *International Conference on Learning Representation (ICLR)*, 2020.

[8] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4852–4861, 2019.

[9] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 544–560. Springer, 2020.

[10] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7436–7456, 2021.

[11] Yizeng Han, Yifan Pu, Zihang Lai, Chaofei Wang, Shiji Song, Junfeng Cao, Wenhui Huang, Chao Deng, and Gao Huang. Learning to weight samples for dynamic early-exiting networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 362–378. Springer, 2022.

[12] 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)*, June 2016.

[13] Yang He, Yuhang Ding, Ping Liu, Linchao Zhu, Hanwang Zhang, and Yi Yang. Learning filter pruning criteria for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2009–2018, 2020.

[14] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017.

[15] 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.

[16] Yerlan Idelbayev and Miguel A Carreira-Perpinán. Low-rank compression of neural nets: Learning the rank of each layer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8049–8059, 2020.

[17] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

[18] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *International conference on machine learning*, pages 3301–3310. PMLR, 2019.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009.

[20] Yuhang Li, Xin Dong, and Wei Wang. Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In *International Conference on Learning Representations*, 2020.

[21] Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. *Advances in neural information processing systems*, 30, 2017.

[22] Zechun Liu, Kwang-Ting Cheng, Dong Huang, Eric P. Xing, and Zhiqiang Shen. Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4942–4952, June 2022.

[23] Zhenhua Liu, Yunhe Wang, Kai Han, Siwei Ma, and Wen Gao. Instance-aware dynamic neural network quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12434–12443, June 2022.

[24] Yuexiao Ma, Taisong Jin, Xiawu Zheng, Yan Wang, Huixia Li, Yongjian Wu, Guannan Jiang, Wei Zhang, and Rongrong Ji. Ompq: Orthogonal mixed precision quantization. *arXiv preprint arXiv:2109.07865*, 2021.

[25] Priyadarshini Panda, Abhronil Sengupta, and Kaushik Roy. Conditional deep learning for energy-efficient and enhanced pattern recognition. In *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 475–480. IEEE, 2016.

[26] Hardik Sharma, Jongse Park, Naveen Suda, Liangzhen Lai, Benson Chau, Vikas Chandra, Hadi Esmaeilzadeh, and Joon Kyung Kim. Bit fusion: Bit-level dynamically composable architecture for accelerating deep neural network. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pages 764–775, 2018. doi: 10.1109/ISCA.2018.00069.

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[28] Chen Tang, Kai Ouyang, Zhi Wang, Yifei Zhu, Wen Ji, Yaowei Wang, and Wenwu Zhu. Mixed-precision neural network quantization via learned layer-wise importance. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 259–275. Springer, 2022.

[29] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2464–2469. IEEE, 2016.

[30] Stefan Uhlich, Lukas Mauch, Fabien Cardinaux, Kazuki Yoshiyama, Javier Alonso Garcia, Stephen Tiedemann, Thomas Kemp, and Akira Nakamura. Mixed precision dnns: All you need is a good parametrization. *arXiv preprint arXiv:1905.11452*, 2019.

[31] Manoj Rohit Vemparala, Nael Fasfous, Lukas Frickenstein, Alexander Frickenstein, Anmol Singh, Driton Salihu, Christian Unger, Naveen-Shankar Nagaraja, and Walter Stechele. Hardware-aware mixed-precision neural networks using in-train quantization. In *British Machine Vision Conference (BMVC)*, 2021.

[32] Maciej Wołczyk, Bartosz Wójcik, Klaudia Bałazy, Igor T Podolak, Jacek Tabor, Marek Śmieja, and Tomasz Trzcinski. Zero time waste: recycling predictions in early exit neural networks. *Advances in Neural Information Processing Systems*, 34:2516–2528, 2021.

[33] Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer. Mixed precision quantization of convnets via differentiable neural architecture search. *arXiv preprint arXiv:1812.00090*, 2018.

[34] Kohei Yamamoto. Learnable companding quantization for accurate low-bit neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5029–5038, 2021.

[35] Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[36] Linjie Yang and Qing Jin. Fracbits: Mixed precision quantization via fractional bitwidths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10612–10620, 2021.

[37] 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)*, September 2018.

[38] 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, 2018.

[39] Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems*, 33:18330–18341, 2020.