Systematic Quantization of Vision Models based on MLPs

Anonymous submission

Abstract

Quantization is wildly taken as a model compression technique, which obtains efficient models by converting floating-point weights and activations in the neural network into lower-bit integers. Quantization has been proven to work well on convolutional neural networks and transformer-based models. Despite the decency of these models, recent works (Touvron et al. 2021; Fusco, Pascual, and Staar 2022; Ma et al. 2022) have shown that MLP-based models are able to achieve comparable results on various tasks ranging from computer vision, NLP to 3D point cloud, while achieving higher throughput due to the parallelism and network simplicity. However, as we show in the paper, directly applying quantization to MLP-based models will lead to significant accuracy degradation. Based on our analysis, two major issues account for the accuracy gap: 1) the range of activations in MLP-based models can be too large to quantize, and 2) specific components in the MLP-based models are sensitive to quantization. Consequently, we propose to 1) apply LayerNorm to control the quantization range of activations, 2) utilize bounded activation functions, 3) apply percentile quantization on activations, 4) use our improved module named multiple token-mixing MLPs, and 5) apply linear asymmetric quantizer for sensitive operations. Equipped with the abovementioned techniques, our Q-MLP models can achieve 79.68% accuracy on ImageNet with 8-bit uniform quantization (model size 30 MB) and 78.47% with 4-bit quantization (15 MB).

Introduction

The deployment of Neural Network (NN) models is often impossible due to application-specific constraints on latency, power consumption, and memory footprint. This prohibits the use of state-of-the-art models with excellent accuracy but large parameter size and FLOPS. Quantization has been proposed as one of several model compression methods to enable efficient inference. Generally, quantization converts floating-point NN models into integer-only or fixed-point models, which is efficient in memory consumption. Thanks to faster integer arithmetic compared to its floating-point counterparts, quantization can also reduce the computation when both weights and activations in the NN are quantized.

Since the success of traditional CNN networks (Tan and Le 2019; Howard et al. 2019) and transformer-based networks (Dosovitskiy et al. 2020; Liu et al. 2021b) on various tasks from computer vision to NLP, significant research efforts have been spent on finding better building blocks for NN architectures. Recent works (Tolstikhin et al. 2021; Touvron et al. 2021; Liu et al. 2021a; Fusco, Pascual, and Staar 2022; Ma et al. 2022) have proposed that NN models based on Multi-Layer-Perceptron (MLPs) can also achieve state-of-the-art performance on those tasks. In addition to accuracy, MLP-based models benefit from their intrinsic parallelism and model simplicity and can potentially achieve higher throughput compared to CNNs and transformers. As Table 1 shows, ResMLP has larger throughput than ViT and ResNets despite having more parameters and FLOPs.

Table 1: Comparing MLP-based models with transformers and CNNs. The throughput is measured on a single TITAN RTX 2080Ti (24GB) GPU with batch size fixed to 128. For reference, the accuracy included here is obtained by models trained solely on ImageNet with no extra data.

Model	**Params** ($\times 10^6$)	**Throughput** (img/sec)	**FLOPs** (G)	**Top-1** (%)
ResMLP-S24/16 (Touvron et al. 2021)	30	468	11.94	79.4
ResMLP-B24/16 (Touvron et al. 2021)	115	195	46.08	81.0
ViT-S/16 (Dosovitskiy et al. 2020)	22	451	8.48	78.1
ViT-B/16 (Dosovitskiy et al. 2020)	86	255	33.72	79.9
ResNet-50 (He et al. 2016)	26	466	7.76	77.7
ResNet-101 (He et al. 2016)	45	287	15.20	79.2

Specifically, each block in MLP-based models has the same parameter size and the same resolution of feature maps, whereas the blocks at the beginning of CNNs tend to have a much smaller parameter size and a larger resolution of feature maps than the subsequent blocks. This uniformity of building blocks makes MLP-based models easier to deploy and optimize on the hardware platforms compared to CNNs. Furthermore, uniform blocks are also friendly to uniform quantization. In contrast, when applying ultra-low bit quantization on CNNs, mixed-precision quantization (Wang et al. 2019; Dong et al. 2019, 2020) is often required to alleviate the accuracy degradation, for which the hardware support can be sub-optimal. Compared to transformers, MLP-based models are also more efficient since they can avoid intensive computation (Tolstikhin et al. 2021) by not explicitly applying the attention mechanism.

In order to simultaneously achieve high accuracy and efficient inference, it is natural to explore quantization on MLP-

based models. However, directly applying quantization to MLP-based models will lead to high accuracy degradation. In this work, we first find that the range of activations in specific MLP-based models can become too large to quantize. Consequently, we propose to restrict the activation range with carefully designed normalization and activation layers. From our experiments, applying LayerNorm instead of the Affine operation, utilizing bounded activation functions, and applying percentile quantization for activations proved beneficial in reducing the activation range. Secondly, our analysis shows that specific operations are more sensitive than the others in MLP-based models. To tackle this issue, we propose a new component named multiple token-mixer, which can be both efficient and less sensitive to quantization. Furthermore, applying asymmetric linear quantizers onto or after sensitive operations helps improve accuracy, with a trivial overhead to support the mixture of symmetric and asymmetric quantizers. Our contributions can be summarized as follows:

- We are the first to analyze the causes of significant accuracy degradation when quantizing MLP-based models.
- We provide universal instructions for designing MLP-based models in order to make them quantization-friendly.
- Our proposed quantization methods can achieve 79.68% accuracy on ImageNet with 8-bit quantization (model size 30 MB), and our 4-bit quantized model has 78.47% accuracy with only 15 MB model size.

Related work

Quantization (Zhou et al. 2017a; Jacob et al. 2018; Zhang et al. 2018; Wang et al. 2019; Cai et al. 2020; Gholami et al. 2021) are common model compression techniques where low-bit precision is used for weights and activations to reduce model size without changing the original network architecture. Quantization can also potentially permit the use of low-precision matrix multiplication or convolution, making the inference process faster and more efficient.

Despite these advances, directly performing post-training quantization (PTQ) with uniform ultra-low bit-width still results in a significant accuracy degradation. As such, Quantization-aware training (QAT) is proposed to train the model to better adapt to quantization. Another promising direction is to use mixed-precision quantization (Zhou et al. 2017b; Wang et al. 2019; Yao et al. 2021), where some layers are kept at higher precision. Although mixed-precision quantization can be well supported on some existing hardware (such as FPGAs) (Huang et al. 2021; Dong et al. 2021), it can lead to a non-trivial overhead on many other hardware platforms (such as GPUs).

MLP-based Models have been recently proposed to perform various tasks, competing against previous convolutional neural networks (CNNs) and transformers. The MLP-Mixer (Tolstikhin et al. 2021) architecture, built entirely on multi-layer perceptrons (MLPs), has produced competitive results in vision tasks. Due to its simple and uniform structure, MLP-Mixer achieves high throughput and brings new possibilities to efficient-learning topics.

Another important characteristic of MLP-Mixer is that it separately uses a channel-mixing MLP to enable communications between different feature channels within each token and a token-mixing MLP to enable communications between different spatial locations across patches. This two-step process in each layer increases the interpretability of deep neural networks and enables further investigating and special designing of each part in later works. In a subsequent work (Touvron et al. 2021), the authors propose the architecture ResMLP, which simplifies the token-mixing module and the norm-layer, achieving a better efficiency-accuracy trade-off. Later, (Liu et al. 2021a; Yu et al. 2021, 2022) further pushes the limits of MLP-based models by finding better ways to improve token-mixing and channel-mixing simultaneously. A very recent work (Trockman and Kolter 2022) combines the merits of convolutions and this mixer-based communication separating technique and proposes ConvMixer, which outperforms not only CNNs but also vision transformers and MLP-Mixer variants. It should be noted that, although ConvMixer is not precisely composed of MLPs, it is intrinsically similar to MLP-based models rather than CNNs. Therefore, we still conduct a detailed analysis of it due to its mixer-based structure and state-of-the-art performance.

Methodology

In this section, we introduce a set of quantization techniques to combine the merit in MLP structure and the efficiency of quantization. We demonstrate that the MLP-based model provides inherent advantages for uniform quantization and can achieve a satisfying accuracy-efficiency trade-off when provided with appropriate techniques.

Quantization preliminaries

Quantization methods quantize weights and activations into integers with a scale factor S and a zero point Z_0. Uniformly quantizing activations or weights to k bit can be expressed as:

$$\begin{aligned} S &= \frac{r_{\max} - r_{\min}}{2^k - 1} \\ Z_0 &= \text{round}\left(2^{k-1} - 1 - \frac{r_{\max}}{S}\right) \\ q &= \text{round}\left(\frac{r}{S} + Z_0\right) \end{aligned} \tag{1}$$

where r is the real number and q is the quantized integer and can then be used to enable hardware integer arithmetic acceleration. In baseline methods, we use symmetric quantization, which means Z_0 equals 0, and the first bit of q serves as a sign bit, with the rest k-1 bits used to represent the integer. (More details in **Asymmetric quantization**)

There are usually two types of quantization methods: post-training quantization (PTQ) and quantization-aware training (QAT). For PTQ, we apply the above quantization directly in the inference stage to the pre-trained weights, and we use the quantized weights and activations to generate results. For QAT, we define the forward and backward pass for the above quantization operations and train quantized parameters together with the model parameters in order to get better quantization results. Both methods are useful and can be applied to different circumstances for deployment.

Figure 1: Token-mixing module using multiple token-mixing MLPs.

Restrict activation ranges

The magnitude of the activation range is highly related to the performance of the quantized models. Generally, a more extensive activation range loses more information with a given bitwidth quantization than a small activation range. In the experiments, we found that the activation ranges of some MLP-based models (e.g., ResMLP and ConvMixer) are unusually high, which leads to severe accuracy degradation in the quantized model. Therefore, it is of great importance to carefully deal with these activation ranges and use techniques to restrict them.

Norm-layer design The choice of norm layer significantly impacts the activation range of features and, therefore, is crucial to the PTQ performance of MLP-based models. Different MLP-based models use different norm layers, which lead to very different activation ranges.

Some models use a simple Affine transformation (Equation 2) as the norm-layer, which only rescales and shifts the input in an element-wise manner. Though it is demonstrated to be a slightly simpler and more efficient layer than LayerNorm, we found it potentially leads to a huge activation range (more details in Section **Experimental results**) and incurs an accuracy drop in its quantized model.

$$\mathrm{Aff}_{\alpha,\beta}(x) = \mathrm{Diag}(\alpha)x + \beta \tag{2}$$

Therefore, we proposed to replace Affine transformation with LayerNorm or BatchNorm (both can be represented by Equation 3) in all the MLP-based models in order to restrict the activation range using channel/batch statistics.

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta \tag{3}$$

Activation layer design The choice of activation layer is another critical factor that affects the activation range in MLP models. Most MLP models use ReLU or GELU as activation layers, and they have been tested to have similar performance (Touvron et al. 2021). However, both GELU and ReLU are not the best choice for quantized MLP-based models since they are not bounded when activations are positive. We propose that the best activations for quantized MLP-based models are ones that are both bounded in negative input values and positive input values.

Parametrized clipping activation (PACT) (Equation 4), which we apply in our experiments, is one of the good choices for the activation layer of MLP-based models. It sets a learnable upper bound parameter to clip all the input values into the range of [0,α]. (More details in paper (Choi et al. 2018))

$$\begin{aligned} y = PACT(x) &= 0.5(|x| - |x - \alpha| + \alpha) \\ &= \begin{cases} 0, & x \in (-\infty, 0) \\ x, & x \in [0, \alpha) \\ \alpha, & x \in [\alpha, +\infty) \end{cases} \end{aligned} \tag{4}$$

In practice, this kind of activation layer can largely restrict the activation range of the MLP-based model and lead to a better performance in quantized MLP models.

Percentiles in activation range Though some MLP-based models have a very large activation range, it does not necessarily imply that they have a large mean value of activations. Instead, the large range may be caused by some extreme outliers in the outputs. When this happens, we can effectively recover the performance by using percentiles to clip the extreme activation values.

Tackle sensitive layers

Parameters and activations in different layers have relatively different sensitivity. Mixed-precision quantization methods allocate different bitwidth for different layers to overcome this problem. However, in the context of uniform quantization, we cannot tune a set of different bitwidths, so we proposed two alternative methods to tackle this issue.

Multiple token-mixing MLPs Usually, MLP-based models have two modules in each layer: the token-mixing module and the channel-mixing module. However, the two modules are different in parameter size and sensitivity. We use the Hessian trace analysis (Yao et al. 2020) to evaluate the sensitivity of the token-mixing MLPs and the channel-mixing MLPs in MLP-Mixer and present the results in Figure 3. We found that the average sensitivity of the parameters (indicated by the mean Hessian trace of the learnable parameters) in token-mixing MLPs is much higher than that in channel-mixing MLPs. We can also comprehend this from a different perspective: since the channel dimensions are usually 4-5 times the token dimensions, parameters in token-mixing MLPs are usually reused 3-4 times more than that of channel-mixing MLPs (because the same MLP applies to all the token/channel dimensions). Therefore, the parameters in token-mixing MLPs are intuitively more sensitive to changes.

The above analysis explains why many subsequent papers performed better in MLP-based models after redesigning the token-mixing MLPs. It also indicates that we should carefully deal with parameters in token-mixing MLPs since performance drop in post-training quantization is highly related to parameter sensitivity.

Consequently, we modify the structure of the original token-mixing MLPs to reduce their sensitivity. As shown in Figure 1, different from applying the same token-mixing MLP to each of the C different channels in the original MLP-Mixer layers, we divide the channels into several groups and apply different token-mixing MLPs to different channel groups. This approach reduces the reuse of parameters in token-mixing MLPs and increases the expressibility of

token MLPs. The experiments show that both the accuracy of MLP-Mixer and the accuracy of the post-training quantized MLP model increase after introducing the multiple token-mixing MLPs. Meanwhile, since the number of the parameters in channel-mixing MLPs is 30 times larger than that in token MLPs, using multiple token-mixing MLPs will not significantly increase the total parameter size of the model. Moreover, thanks to the merits of quantization, the model size of the quantized multiple token-mixing Mixer is still much smaller than the original MLP-Mixer.

Asymmetric quantization For MLP-based models, sensitivity imbalances are not only found in weights in different MLPs but also found between weights and activations. In the experiments, we find that activations are much more sensitive than weights in MLP-based models. This argument is derived from the fact that we can have a relatively good PTQ result with ultra-low bitwidth (3 or 4) weight quantization and 8-bit activation quantization, while we cannot get any acceptable PTQ results with an activation bit less than 8 (no matter how many bit the weights use).

To better deal with these sensitive activations in the context of uniform quantization, we propose to use asymmetric quantization for activations and still use symmetric quantization for weights. The term asymmetric quantization means that the zero point could be any floating point value, and the activation range depends on the difference in max and min of the input value instead of the max of the absolute value. More concretely, the scaling factor can be expressed as Equation 5 for symmetric quantization:

$$S = \frac{\max(abs(r_{\max}), abs(r_{\min}))}{2^{k-1} - 1} \tag{5}$$

and Equation 6 for asymmetric quantization,

$$S = \frac{r_{\max} - r_{\min}}{2^{k} - 1} \tag{6}$$

where r refers to the real number inputs and k refers to the activation bitwidth. From Equation 5 and 6, we can see that asymmetric quantization potentially provides one more bit for activation quantization (if the max positive and negative values are in different orders of magnitude). Therefore, we can better deal with sensitive activations while staying within the scope of uniform quantization.

Experimental results

To evaluate our proposed quantization approaches for MLP-based models, we perform a series of experiments on ImageNet with MLP-Mixer-B/16, ResMLP-S24/16, and ConvMixer-768/32 (since they have similar model scales). Results show that some of these techniques benefit the model's accuracy, and others are even indispensable to avoid severe performance loss incurred by quantization. We should note that, although ConvMixer does not have the MLP in its structure, it absorbs the idea in those recently proposed MLP-based models that the multi-head attention module can be replaced by MLP or convolution for higher efficiency. By default, we use uniform quantization throughout the layers to take advantage of the simplicity of MLP-based models, with weights and activations quantized to 8-bit for post-training quantization, and weights to 3/4/8 bit and activations to 8-bit for quantization-aware training. In addition, we use channel-wise quantization for weights and exponential moving averages (EMA) with momentum to derive the quantization range for activation quantization.

Restrict activation ranges

In experiments, we find that some MLP-based models have unusually high activation ranges, which lead to large quantization intervals and result in accuracy degradation during quantization. Therefore, we first calculate the max and 99% percentile of the activation values throughout the layers and plot their activation statistics in Figure 2. These values are crucial for quantization results since they determine the activation range in standard quantization settings and percentile quantization settings, respectively. Results in the top two graphs in Figure 2 show that the original ResMLP and ConvMixer models have a relatively high activation range compared to traditional CNNs and Vision Transformers. The bottom two graphs in Figure 2 show that after using our proposed methods, such as replacing Affine with LayerNorm in ResMLP and using PACT as activation layers in ConvMixer, the activation range is well restricted. In conclusion, these graphs not only present the causes of large accuracy degradation in quantized MLP-based models but also validate the effectiveness of our proposed methods.

Norm-layer design As mentioned in Section **Methodology**, quantized MLP-based models may benefit from LayerNorm or BatchNorm. We demonstrate this by replacing the Affine function in ResMLP with LayerNorm. Results in Table 2 show that this approach achieves better classification results in PTQ experiments than the original ResMLP model. Moreover, instability issue occurs during QAT experiments for the original ResMLP, and quantization can only be done successfully for ResMLP with LayerNorm. The results suggest that LayerNorm helps restrict the activation range better than the Affine function, which then helps to derive a more accurate integer representation and leads to better accuracy.

Table 2: Affine vs. LayerNorm in ResMLP. Here, we abbreviate Weight Precision and Activation Precision as "Precision", Norm-layer as "Norm", Model Size as "Size" (in MB), Bit Operations as "BOPS" (in G), and Top-1 Accuracy as "Top-1". Note that BOPS is defined as FLOPS $\times$ activation bits $\times$ weight bits, and "WxAy" means weight with x-bit and activation with y-bit. Other tables use similar abbreviations in the rest of the paper.

Method	Precisioin	Norm	Size(MB)	BOPS(G)	Top-1
Baseline	W32A32	Affine	120	12226	79.38
		LN	120	12226	79.59
PTQ	W8A8	Affine	30	764	74.93
		LN	30	764	79.20
QAT	W8A8	Affine	30	764	-
		LN	30	764	79.44
QAT	W4A8	LN	15	382	78.35

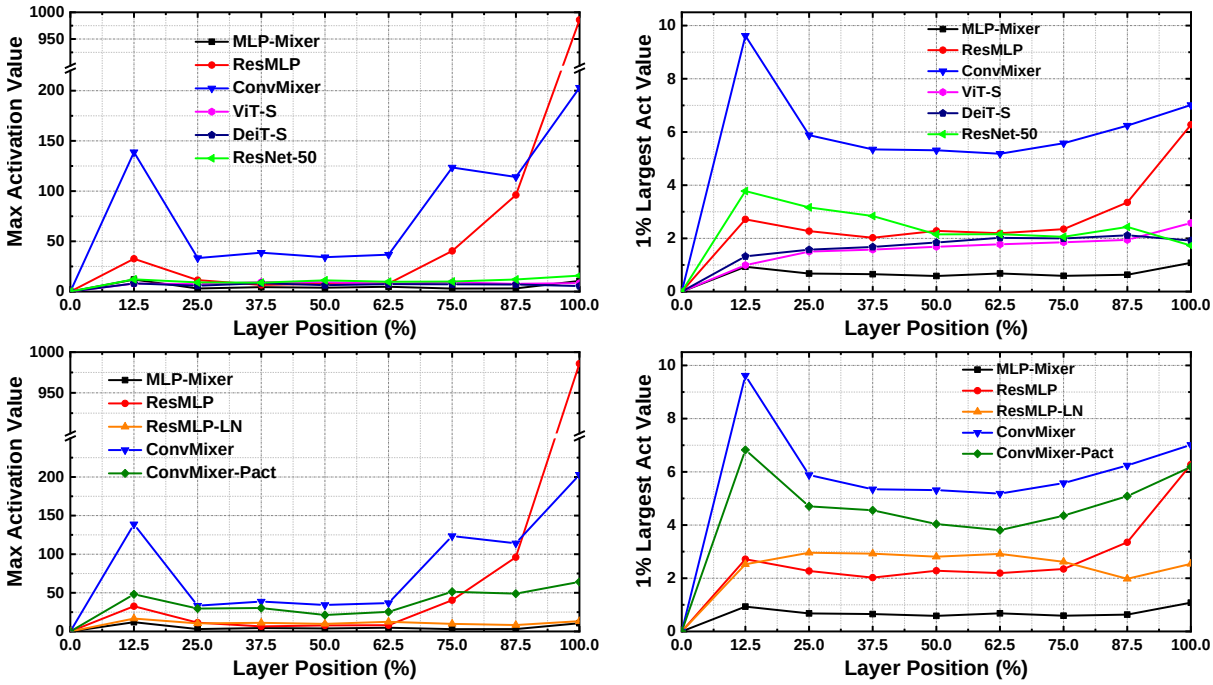

Figure 2: Graph of the max (left) and the 1% largest (right) activation values in vision models throughout the layers. Layer Position shows the relative position of a given layer in the whole model. For example, the output of the third layer in a 24-layer ResMLP model has a layer position of 12.5%. The top two graphs show the activation values in MLP, Vision Transformers, and CNNs. The bottom two graphs show activation values in MLPs and our proposed quantization-friendly MLP variants.

Activation layer design For models with an extremely large activation range, using LayerNorm or BatchNorm may not be sufficient for restricting the activation range. For example, ConvMixer, using BatchNorm as its norm layer, still suffers from quantization degradation due to its large activation range. However, results in Table 3 demonstrate that we can efficiently restrict the activation range by using activation layers with bounded outputs for both negative and positive input values. With the help of a narrower activation range, our PACT-ConvMixer achieves at least similar results in different PTQ settings and much better results in all settings of QAT. Here, we choose to use PACT activation in our quantized ConvMixer model since it has a learnable upper bound and is potentially more capable in restricting the range. Other bounded activation layers (for example, ReLU6) should work as well. It is important to mention that we use the asymmetric quantization settings for QAT comparison since QAT for ConvMixers cannot converge in symmetric settings. A potential reason is that QAT suffers more from sensitive activation ranges, and asymmetric quantization reduces activation sensitivity, which will be discussed in detail in **Tackle sensitive layers**.

Percentile An easier way to restrict the activation range is to use a percentile max value when calculating the quantization parameters. For example, using a 99% percentile option can help clip the 1% biggest activation values so that the activation range will no longer depend on those extreme outliers. Table 4 shows that percentile partly helps to recover the accuracy of the quantized models. Note that calculating percentiles in QAT make the training process much slower, so we only use activation percentiles in PTQ experiments.

Table 3: ReLU vs. PACT in ConvMixer

Method	Precisioin	Activation	Size(MB)	BOPS(G)	Top-1
Baseline	W32A32	ReLU	84	42762	80.16
		PACT	84	42762	80.22
PTQ	W8A8	ReLU	21	2672	57.81
		PACT	21	2672	68.91
QAT(asym)	W8A8	ReLU	21	2672	77.09
		PACT	21	2672	78.65
QAT(asym)	W4A8	ReLU	11	1336	75.88
		PACT	11	1336	77.89

Table 4: Percentile in ResMLP

Method	Precisioin	Percentile	Size(MB)	BOPS(G)	Top-1
Baseline	W32A32	×	120	12226	79.38
PTQ	W8A8	×	30	764	74.93
		✓	30	764	77.74

Tackle sensitive layers

Multiple token-mixing MLPs As mentioned in Section **Methodology**, we calculate the Hessian traces of each MLP

in MLP-Mixer in Figure 3. Results show that parameters in token-mixing MLPs are more sensitive than in channel-mixing MLPs. Therefore, token-mixing MLPs should be carefully designed in order to achieve better PTQ results.

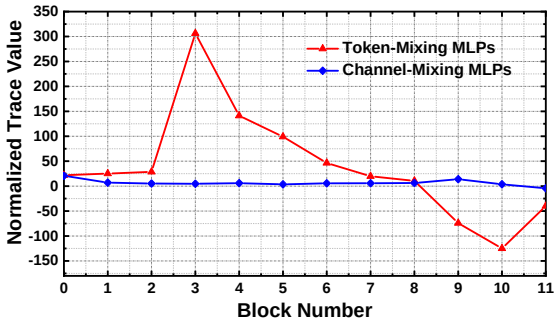

Figure 3: Mean Hessian traces of token-mixing MLP and channel-mixing MLP in layers of MLP-Mixer. Note that the Hessian trace values are normalized according to their parameter size.

In Table 5 we show that the full precision accuracy, PTQ, and QAT results all obtain a remarkable improvement after introducing the multiple token-mixing MLP into the original MLP model. It indicates that reducing the sensitivity of specific parameters is crucial for obtaining high-performance quantized MLP-based models.

Table 5: Token-mixing in MLP-Mixer

Method	Precisioin	Token-mixing	Size(MB)	BOPS(G)	Top-1
Baseline	W32A32	Single	240	25825	76.64
		Multiple	261	25825	78.35
PTQ	W8A8	Single	60	1614	74.46
		Multiple	65	1614	75.34
QAT	W4A8	Single	30	807	75.82
		Multiple	33	807	77.66
QAT	W3A8	Multiple	25	605	76.85

Asymmetric quantization As described in Section **Methodology**, asymmetric quantization can help ease the sensitivity of activation layers by adding an extra bitwidth implicitly. Table 6 shows that asymmetric quantization is helpful in quantized MLP-based models and especially important in the quantization of ConvMixer. We also find that ConvMixer can only use QAT in the asymmetric quantization mode, and the training would be very likely to diverge otherwise. These results imply that using asymmetric quantization to reduce sensitivity not only provides better performance but also helps stabilize QAT.

Ablation study

Here, we take ConvMixer as an example and combine all the techniques mentioned above to provide a thorough ablation study to illustrate the effectiveness of our methods. As the results shown in Table 7, bounded activation layer, activation percentiles, and asymmetric quantization not only improve the quantization performance separately, but they can boost the performance with any of the combinations. Incorporating all of the aforementioned methods gives us the best results for ConvMixer with an 8-bit PTQ of 76.78% ImageNet classification accuracy.

Table 6: Symmetric vs. Asymmetric in PTQ experiments

Model	Precisioin	Sym/Asym	Size(MB)	BOPS(G)	Top-1
Q-MLP-Mixer	W8A8	Sym	60	1614	74.46
		Asym	60	1614	76.20
Q-ResMLP	W8A8	Sym	30	764	74.93
		Asym	30	764	78.28
Q-ConvMixer	W8A8	Sym	21	2672	57.81
		Asym	21	2672	76.21

Table 7: Ablation Study

Model	BatchNorm	Percentile	Asymmetric	PACT	Top-1
ConvMixer	-	-	-	-	80.16
Q-ConvMixer	✓	×	×	×	57.81
	✓	✓	×	×	69.36
	✓	×	✓	×	76.21
	✓	✓	✓	×	76.35
	✓	×	×	✓	68.91
	✓	✓	×	✓	73.88
	✓	×	✓	✓	76.06
	✓	✓	✓	✓	76.78

Best quantized models

Combining all of the abovementioned methods, we derive the best results for quantized MLP-Mixer, ResMLP, ConvMixer in Table 8 and 9 with boldface. Since we are the first to investigate quantization aspects of MLP models, there are few previous works to compare with. Therefore, we apply the open-sourced CNN-targeted quantization method HAWQ-v3 (Yao et al. 2021), which supports both PTQ and QAT, to MLP-based models for comparison. Results in Table 8 and 9 show that our quantization method works much better on MLP-based models, indicating the importance of considering the MLP models' particular structure. Although all three models gain much better accuracy after applying our proposed methods, ResMLP distinctly outperforms the other two MLP-based models in experiments. It implies that ResMLP variants are potentially more efficient and quantization friendly. Meanwhile, though MLP-Mixer's performance and computation-accuracy trade-off are slightly behind the other two models, it is the easiest to quantize among the three MLP-based models. QAT on MLP-Mixer can be conducted smoothly, while instability issue occurs in ResMLP and ConvMixer unless we redesign the norm-layer and activation layer according to Section **Methodology**.

To compare our best quantized MLP-based models with quantized CNNs and transformer-based networks, we highlight the best quantized MLP-based models with different

Table 8: Comparison of the post-training quantization performance of MLP-based models with CNNs and transformer-based models using different quantization methods.

Category	Model	Method	Precisioin	Size(MB)	BOPS(G)	Top-1
MLP-based Networks	Q-MLP-Mixer	HAWQ-V3 (Yao et al. 2021)	W8A8	60	1614	74.40
		Ours	W8A8	65	1614	**77.75**
	Q-ResMLP	HAWQ-V3 (Yao et al. 2021)	W8A8	30	764	76.69
		Ours	W8A8	30	764	**79.43**
	Q-ConvMixer	HAWQ-V3 (Yao et al. 2021)	W8A8	21	2672	72.54
		Ours	W8A8	21	2672	**76.78**
Transformer Networks	Q-DeiT-S	EasyQuant (Wu et al. 2020)	W8A8	22	543	76.59
		Bit-Split (Wang et al. 2020)	W8A8	22	543	77.06
	Q-DeiT-B	EasyQuant (Wu et al. 2020)	W8A8	86	2158	79.36
		Bit-Split (Wang et al. 2020)	W8A8	86	2158	79.42
	Q-ViT-B	Percentile (Li et al. 2019)	W8A8	86	2158	74.10
		PTQ-ViT (Liu et al. 2021c)	W8A8	86	2158	76.98
Convolutional Networks	Q-ResNet50	Bit-Split (Wang et al. 2020)	W8A8	26	496	75.96
		ZeroQ (Cai et al. 2020)	W8A8	26	496	77.67

Table 9: Comparison of the quantization-aware training performance of MLP-based models with CNNs using different quantization methods (we did not find QAT results of transformers in previous works). Here, 4/8 in HAWQ-V3 means mixed precision with 4 and 8 bits. Note that Q-ResMLP and Q-ConvMixer results using HAWQ-V3 are derived after applying our norm-layer and activation-layer design to the original model.

Category	Model	Method	Precisioin	Size(MB)	BOPS(G)	Top-1
MLP-based Networks	Q-MLP-Mixer	HAWQ-V3 (Yao et al. 2021)	W8A8	60	1614	76.28
		Ours	W8A8	65	1614	**78.17**
		Ours	W4A8	33	807	**77.94**
	Q-ResMLP	HAWQ-V3 (Yao et al. 2021)	W8A8	30	764	77.36
		Ours	W8A8	30	764	**79.68**
		Ours	W4A8	15	382	**78.47**
	Q-ConvMixer	HAWQ-V3 (Yao et al. 2021)	W8A8	21	2672	75.88
		Ours	W8A8	21	2672	**78.65**
		Ours	W4A8	11	1336	**77.89**
Convolutional Networks	Q-ResNet50	Integer Only (Jacob et al. 2018)	W8A8	26	496	74.90
		RVQuant (Park, Yoo, and Vajda 2018)	W8A8	26	496	75.67
		HAWQ-V3 (Yao et al. 2021)	W8A8	26	496	77.58
		HAWQ-V3 (Yao et al. 2021)	W4/8A4/8	19	308	75.39
		LQ-Nets (Zhang et al. 2018)	W4A32	13	992	76.40

precision settings in Table 8 and 9. Results show that Q-ResMLP outperforms other quantized models with similar model scales and can even achieve comparable performance with some much larger models.

Implementation details

We primarily evaluate our proposed and existing models on the ImageNet-1k validation set. Specifically, we add our Q-MLP-Mixer, Q-ResMLP, Q-ConvMixer models into the timm framework (Wightman 2019), and then train new models and implement the Quantization-Aware Traning (QAT) under the default settings in (Tolstikhin et al. 2021; Touvron et al. 2021; Trockman and Kolter 2022) except changing the initial learning rate to 2×10^{-5} during QAT. The training for new models, such as multiple token-mixing MLPs and the ResMLP with LayerNorm, usually takes 4-5 days on eight TITAN RTX 2080Ti (24GB) GPUs, and the QAT experiments usually take 1-2 days.

Conclusions

In this work, we analyze the quantization of state-of-the-art MLP-based vision models. Two major problems concluded are: 1) MLP-based models suffer from large quantization ranges of activations 2) specific components of MLP-based models are sensitive to quantization. To alleviate the degradation caused by activations, we propose to apply LayerNorm to control the quantization range, and we also take advantage of bounded activation functions and percentile quantization on activations. In order to tackle the sensitivity, we propose to apply our multiple token-mixing MLPs and use linear asymmetric quantizers for the sensitive operations in MLP-based models. With these practical techniques, our Q-MLP models can achieve 79.43% top-1 accuracy on ImageNet with 8-bit post-training quantization (30 MB model size). For quantization-aware training, our Q-MLP models can achieve 79.68% accuracy using 8-bit (30 MB) and 78.47% accuracy using 4-bit quantization (15 MB).

References

Cai, Y.; Yao, Z.; Dong, Z.; Gholami, A.; Mahoney, M. W.; and Keutzer, K. 2020. ZeroQ: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13169–13178.

Choi, J.; Wang, Z.; Venkataramani, S.; Chuang, P. I.-J.; Srinivasan, V.; and Gopalakrishnan, K. 2018. PACT: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*.

Dong, Z.; Gao, Y.; Huang, Q.; Wawrzynek, J.; So, H. K.; and Keutzer, K. 2021. Hao: Hardware-aware neural architecture optimization for efficient inference. In *2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 50–59. IEEE.

Dong, Z.; Yao, Z.; Arfeen, D.; Gholami, A.; Mahoney, M. W.; and Keutzer, K. 2020. HAWQ-V2: Hessian Aware trace-Weighted Quantization of Neural Networks. *Advances in neural information processing systems*.

Dong, Z.; Yao, Z.; Gholami, A.; Mahoney, M. W.; and Keutzer, K. 2019. HAWQ: Hessian AWare Quantization of Neural Networks With Mixed-Precision. In *The IEEE International Conference on Computer Vision (ICCV)*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fusco, F.; Pascual, D.; and Staar, P. 2022. pNLP-Mixer: an Efficient all-MLP Architecture for Language. *arXiv preprint arXiv:2202.04350*.

Gholami, A.; Kim, S.; Dong, Z.; Yao, Z.; Mahoney, M. W.; and Keutzer, K. 2021. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. 2019. Searching for MobileNetV3. In *Proceedings of the IEEE International Conference on Computer Vision*, 1314–1324.

Huang, Q.; Wang, D.; Dong, Z.; Gao, Y.; Cai, Y.; Li, T.; Wu, B.; Keutzer, K.; and Wawrzynek, J. 2021. Codenet: Efficient deployment of input-adaptive object detection on embedded fpgas. In *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 206–216.

Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; and Kalenichenko, D. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2704–2713.

Li, R.; Wang, Y.; Liang, F.; Qin, H.; Yan, J.; and Fan, R. 2019. Fully Quantized Network for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2810–2819.

Liu, H.; Dai, Z.; So, D.; and Le, Q. 2021a. Pay attention to MLPs. *Advances in Neural Information Processing Systems*, 34.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Liu, Z.; Wang, Y.; Han, K.; Zhang, W.; Ma, S.; and Gao, W. 2021c. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34: 28092–28103.

Ma, X.; Qin, C.; You, H.; Ran, H.; and Fu, Y. 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv preprint arXiv:2202.07123*.

Park, E.; Yoo, S.; and Vajda, P. 2018. Value-aware quantization for training and inference of neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 580–595.

Tan, M.; and Le, Q. V. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34.

Touvron, H.; Bojanowski, P.; Caron, M.; Cord, M.; El-Nouby, A.; Grave, E.; Izacard, G.; Joulin, A.; Synnaeve, G.; Verbeek, J.; et al. 2021. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*.

Trockman, A.; and Kolter, J. Z. 2022. Patches Are All You Need? *arXiv preprint arXiv:2201.09792*.

Wang, K.; Liu, Z.; Lin, Y.; Lin, J.; and Han, S. 2019. HAQ: Hardware-Aware Automated Quantization. *In Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wang, P.; Chen, Q.; He, X.; and Cheng, J. 2020. Towards accurate post-training network quantization via bit-split and stitching. In *International Conference on Machine Learning*, 9847–9856. PMLR.

Wightman, R. 2019. PyTorch Image Models. https://github.com/rwightman/pytorch-image-models.

Wu, D.; Tang, Q.; Zhao, Y.; Zhang, M.; Fu, Y.; and Zhang, D. 2020. EasyQuant: Post-training quantization via scale optimization. *arXiv preprint arXiv:2006.16669*.

Yao, Z.; Dong, Z.; Zheng, Z.; Gholami, A.; Yu, J.; Tan, E.; Wang, L.; Huang, Q.; Wang, Y.; Mahoney, M.; et al. 2021. Hawq-v3: Dyadic neural network quantization. In *International Conference on Machine Learning*, 11875–11886. PMLR.

Yao, Z.; Gholami, A.; Keutzer, K.; and Mahoney, M. W. 2020. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, 581–590. IEEE.

Yu, T.; Li, X.; Cai, Y.; Sun, M.; and Li, P. 2021. Rethinking token-mixing mlp for mlp-based vision backbone. *arXiv preprint arXiv:2106.14882*.

Yu, T.; Li, X.; Cai, Y.; Sun, M.; and Li, P. 2022. S2-mlp: Spatial-shift mlp architecture for vision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 297–306.

Zhang, D.; Yang, J.; Ye, D.; and Hua, G. 2018. LQ-Nets: Learned Quantization for Highly Accurate and Compact Deep Neural Networks. In *The European Conference on Computer Vision (ECCV)*.

Zhou, A.; Yao, A.; Guo, Y.; Xu, L.; and Chen, Y. 2017a. Incremental network quantization: Towards lossless CNNs with low-precision weights. *International Conference on Learning Representations*.

Zhou, Y.; Moosavi-Dezfooli, S.-M.; Cheung, N.-M.; and Frossard, P. 2017b. Adaptive quantization for deep neural network. *arXiv preprint arXiv:1712.01048*.