

Fixed-point Quantization for Vision Transformer

1st Zhixin Li

NLPR&AiRiA, Institute of Automation,
Chinese Academy of Sciences
School of Artificial Intelligence,
University of Chinese Academy of Sciences
Beijing, China
zhixin.li@nlpr.ia.ac.cn

2nd Peisong Wang

NLPR&AiRiA, Institute of Automation,
Chinese Academy of Sciences
Beijing, China
peisong.wang@nlpr.ia.ac.cn

3rd Zhiyuan Wang *

Defense Innovation Institute (DII),
Artificial Intelligence Research Center (AIRC)
Beijing, China
alice_zyw@foxmail.com

4rd Jian Cheng

NLPR&AiRiA, Institute of Automation,
Chinese Academy of Sciences
Beijing, China
jcheng@nlpr.ia.ac.cn

Abstract—Recently, transformer-based models has shown promising results on miscellaneous computer vision tasks. However, its high computation cost makes it neither practical to deploy on mobile devices, nor economic to compute on servers. In this paper, we propose two effective quantization schemes for reducing the memory usage and computation consumption of vision transformers. First, we develop an approximation-based Post-training Quantization (PTQ) approach which optimizes for a set of quantization scaling factors that minimize quantization errors. Moreover, we introduce a learning-based Quantization-aware Training (QAT) approach that allows for model finetuning after inserting quantization operations to restore accuracy. Furthermore, we reveal the complementary effects of learning-based approach and approximation-based approach in QAT and propose an effective strategy for the initialization of quantization parameters. We evaluate our approaches on ImageNet for different vision transformer models. Our quantization algorithms outperform the previous state-of-art approaches for both post-training quantization and quantization-aware training benchmark. With weights and activations in vision transformer quantized to 8-bit integers, we obtain a $\times 4$ compression rate of model parameters with an accuracy drop of less than 0.2% for models of various scales.

Index Terms—quantization, compression, acceleration, transformer, fixed-point

I. INTRODUCTION

These years have witnessed the great success of Transformer-based [1] [2] [3] models in natural language processing (NLP) tasks. Recently some efforts have been made to transfer Transformer into computer vision (CV) domain and show promising results [4] [5] [6] [7]. However, the Transformer-based models suffer from large number of parameters and high computation amount, which will inevitably cause heavy memory usage and high latency during inference. To alleviate this problem, many methods are introduced to

compress Transformer, like pruning [8] [9] [10] [11], knowledge distillation [12] [13] [14], and quantization [15] [16] [17] [18] [19].

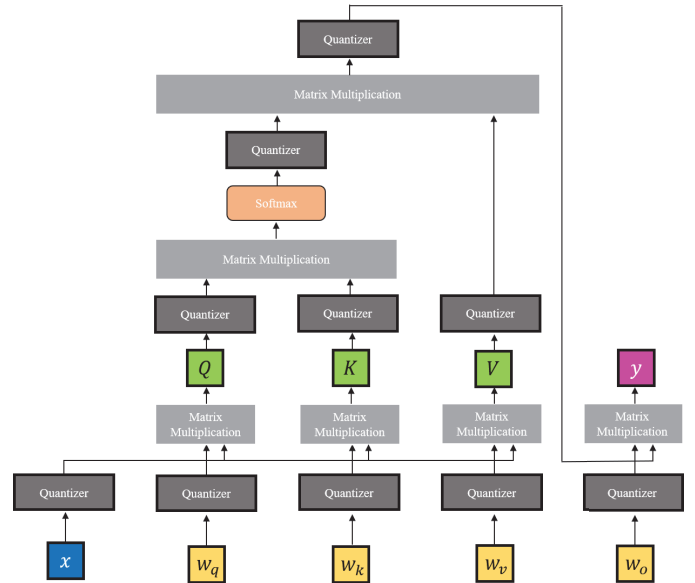


Fig. 1. Quantization pipeline of the Multi-Head Attention module.

Compared to other existing compression methods, quantization is more likely to achieve a better trade-off between the compression rate and performance. There have been some explorations about the quantization of BERT [1], an effective Transformer-based model that can handle various downstream NLP tasks after being pretrained on a large dataset and finetuned on specific tasks. [15] quantizes the word embedding layers and fully connected layers in BERT to 8 bits and reports no severe performance drop on the General Language Understanding Evaluation (GLUE) [20] dataset. But it doesn't quantize the multi-head attention layers which are the compu-

This work was supported in part by Jiangsu Key Research and Development Plan (No.BE2021012-2).

* Corresponding Author

tation overhead in some tasks using long sequences as input. [16] applies quantization to all layers in the encoder and uses a Hessian-based mixed-precision method to achieve low bits quantization of weights.

Vision Transformer (ViT) [4] [5] of Transformer in computer vision tasks, reporting results comparable to convolution neural networks (CNN) like ResNet [21] on classification datasets. [6], [7] extend vision transformers to various computer vision tasks like instance segmentation as well as other low-level tasks such as denoising and super-resolution and achieve State Of The Art (SOTA). Despite its effectiveness in computer vision, little effort has been made to the quantization of vision transformers. Considering the scale of ImageNet dataset is much bigger than GLUE, it will be more challenging to search for an optimal discrete solution after introducing quantization operations. In this work, we proposed fixed-point quantized Vision Transformer, which quantizes all layers in the DeiT model proposed by [5] to 8 bits, including the patch embedding layer, the transformer layers, and the final classification layer. The quantization pipeline of the Multi-Head Attention, which is the essential component of ViT, is illustrated as Fig.1. Post-training quantization takes no extra time for finetuning, thus is fast for deployment. Nevertheless, it suffers from a considerable performance gap compared to the quantization-aware training approach. Therefore quantization-aware training is favored in circumstances when abundant deploying time is given to pursue higher performance. We use an approximation-based approach for post-training quantization, while a learning-based approach for quantization-aware training. Furthermore we reveal the complementary effect of the approximation method and the learning-based method in improving the performance of the quantized model. We obtain a $\times 4$ compression rate of model parameters while maintaining the accuracy on ImageNet [22]. Experimental results on ImageNet [22] demonstrate the effectiveness of our method by outperforming previous state-of-art quantization approaches. For instance, we obtain an 77.79% top-1 accuracy using DeiT-Small model on ImageNet for 8-bit post-training quantization, surpassing the previous state-of-art PTQ approach by a margin of 0.3%.

II. RELATED WORK

A. Vision Transformers

The Visual Transformer (ViT) introduced by [4] directly inherits the Transformer architect from natural language processing by dividing input images into sequential patches. It presents promising results compared to CNN-based models. But it is pre-trained using a private image dataset (JFT-300M [23], 300 million images) and it requires massive computing resources to train. To address these problems, [5] uses miscellaneous training strategies to allow for effective training on public datasets like ImageNet and doesn't require a large number of computation resources.

B. Quantization

Study of Quantization has a long history for CNNs. It projects floating-point parameters in a network into discrete formats to reduce its storage size. Meanwhile, the floating-point matrix multiplication will be replaced with a fixed-point one if both weights and inputs are quantized, which provides remarkable acceleration of inference speed. Quantization methods can be classified into different categories according to their strategies of optimizing the quantization levels. There is approximation-based quantization [24] [25] [26] which aims at minimizing quantization error introduced by quantization operation, while loss-aware based quantization [27] [28] [29] directly optimizes the quantizer to minimize the task loss. Vector or product quantization [30] [31] clusters the full-precision weight vectors or the outputs of matrix multiplication to several quantization centers stored in a look-up codebook.

On the other hand, quantization methods can also be distinguished by their quantization pipelines, i.e., whether requiring retraining after quantization. Post-training Quantization (PTQ) [32] [33] [34] [35] optimize neural networks to be robust to quantization without a complete training but using a little percentage of data. [36] [37] [38] further improve the pipelines to allow for retraining without using any data at all. Quantization-aware Training (QAT) [39] [40] [41] generally outperforms PTQ in terms of performance at the cost of more training time and access to training data. Operations are inserted in the neural network computational graph that simulate the quantization noise introduced by the quantization procedural. Several recent papers enhance its performance by introducing learnable quantization parameters [42] [43] [44] [45] [46] [47]. These learning-based methods collect the gradient of the training loss w.r.t the quantization parameters and take a standard gradient descent optimization step.

Some work has been studied for the quantization of Transformer-based models. Fully-quantized Transformer [18] and Q8BERT [15] successfully applied 8-bit fixed-point quantization to BERT. Lower bits quantization is also investigated in [16] [17]. To avoid severe performance drop in low-bit weight quantization, Q-BERT [16] and GOBO [17] utilize mixed-precision quantization. Nevertheless, mixed-precision quantization can be unfriendly to hardware implementation. Although there is few work about the quantization of ViT, recently [19] propose a post-training quantization approach which takes into account the ranking orders of the attention score.

III. APPROACH

In this section, we discuss our quantization scheme, including post-training quantization (PTQ) and quantization-aware training (QAT).

We quantize both weights and activations in ViT to allow for inference acceleration via fixed-point matrix multiplication. Let the full-precision weights in ViT be w , the corresponding quantized weight is denoted as $\hat{w} = Q(w)$ where Q is the quantization operation. Similarly, the activations x in the network are quantized to be $\hat{x} = Q(x)$. After introducing

the quantization operations, we retrain the model to restore model's accuracy on ImageNet [22]. Specifically, at each mini-batch iteration, we quantize the weights w and activations x in the model to 8-bit format \hat{w} and \hat{x} . Then we do the forward propagation with the quantized weights and activations. We can get the gradients of the loss with respect to the quantized weights $\frac{\partial \mathcal{L}}{\partial \hat{w}}$ through standard backward propagation. We use these gradients to update the latent weights, i.e. the full precision weights: $w^{t+1} = \text{Update}(w^t, \frac{\partial \mathcal{L}}{\partial \hat{w}^t}, \eta^t)$, with η^t being the learning rate at the t -th iteration.

In the following, we will first interpret how to quantize in Section III-A and what to quantize in Section III-B and III-C.

A. Quantization Method

As shown in (1) and (2), the quantizer takes the given tensor v as input, and outputs a scaled integer representation \bar{v} that allows for fixed-point calculation, which is afterwards scaled back to the float tensor \hat{v} with the same scaling factor. We use s for scaling factor, Q_P and Q_N for the maximum and minimum value of quantization levels, respectively.

$$\bar{v} = \lfloor \text{clip}(v/s, -Q_N, Q_P) \rfloor, \quad (1)$$

$$\hat{v} = \bar{v} \cdot s. \quad (2)$$

$\text{clip}(z, r_1, r_2)$ clamps z with values smaller than r_1 to r_1 and larger than r_2 to r_2 , while $\lfloor z \rfloor$ rounds z to the closest integer. Providing quantization bitwidth b , unsigned integers have $Q_N = 0$ and $Q_P = 2^b - 1$ and signed integers have $Q_N = 2^{b-1}$ and $Q_P = 2^{b-1} - 1$. For our case, we use signed integers to quantize all the weights and activations in ViT except for the attention score matrix, which is always positive and can be represented with unsigned integers.

At inference stage, with both weights and activations quantized to integer representations \bar{w} and \bar{x} , fixed-point matrix multiplication can be utilized to implement convolutional or fully connected layers, after which the outputs of these layers are scaled back using the same scaling factor by a relatively low-cost floating-point scalar-tensor multiplication.

For post-training quantization, inspired by [26], we propose an approximation-based method to solve the optimal scaling factor. Specifically, we formulate it as an optimization problem as (3).

$$s^* = \arg \min_s \|v - \hat{v}\|, \text{ where } \hat{v} = \bar{v} \cdot s \quad (3)$$

The optimal s is supposed to minimize the main square error (MSE) between the quantized tensors \hat{v} and the full precision ones v . Here, $\|\cdot\|$ is the l_2 norm of the given vectors. There is no closed-form solution for s , given that \bar{v} itself is dependent on s as showed in (1). A naive method would be searching the value of s using brute-force and selecting the one that minimizes quantization error. Here we adopt an iterative approach to solve this optimization problem, i.e., we fix s and compute \bar{v} , after which we use the updated \bar{v} to optimize s . These two steps are taken iteratively until s converges. Our approach only takes negligible time compared to the brute-force search method.

Previous researches suggest that Transformer is more sensitive to quantization in contrast with CNN. Thus PTQ is less likely to regain the performance due to its lack of retraining. In this case, Quantization-aware Training (QAT) can be used to bridge the gap between the quantized model and the full precision one.

The MSE-based method can be applied to quantization-aware training directly, however, potential harm could be brought to performance by the deviation of distribution of weights and activations after finetuning, implying the scaling factors computed before finetuning to be sub-optimal. Taking this into consideration, we follow the procedural of LSQ [44], a learning-based algorithm, to conduct the quantization-aware training due to its effectiveness in convolution networks. As is done in [44], we update the quantization scaling factor using a scaled gradient backward propagating from the training loss.

$$\partial \hat{v} / \partial s = \begin{cases} -v/s + \lfloor v/s \rfloor & \text{if } -Q_N < \frac{v}{s} < Q_P \\ -Q_N & \text{if } v/s \leq -Q_N \\ Q_P & \text{if } v/s \geq Q_P \end{cases} \quad (4)$$

$$s^{t+1} = \text{Update}(s^t, g \frac{\partial \mathcal{L}}{\partial \hat{v}} \frac{\partial \hat{v}}{\partial s^t}), \text{ where } g = 1/\sqrt{N_F Q_P} \quad (5)$$

g in (5) is a gradient scaling scalar that makes sure the converging speed of s is approximately equal with model parameters. Most of the operations in (1) and (2) are differentiable and can backward propagate normally except the rounding function, for its gradient is zero across almost the whole axis. Hence, we use the straight through estimator (STE) [48] to approximate its gradient function, which allows for the gradient flow from the loss to penetrate the quantization function.

Despite LSQ [44] performs well in CNNs, a problem emerges in our experiments on ViTs that the quantized model is hard to converge during the retraining stage. By analyzing the training process we attribute this to the ill initialization of the quantization scale factors, as is similar to the conclusions from [47]. The initialization values of scaling factors are far from the final converged values, as shown in Fig. 2, and it takes many iterations for the scaling factors to eventually converge to the optimal solution, during which period the model parameters would optimize in the wrong solution space.

To alleviate this phenomenon, we conduct an additional optimization procedure using the approximation-based approach mentioned above before learning the scaling factors, i.e., we initialize the scaling factor using the optimal value that minimize the mean square error (MSE) between the quantized tensors \hat{v} and the full precision ones v , as (3). It differs from the initialization method used in [47], for we use the MSE-based approach instead of Gaussian statistics to initialize model weights. After modifying the initialization scheme, we obtain a considerable performance boost, and the initialization values of the scaling factors become much closer to the optimal values, as is seen in Fig. 2. It implies that the approximation-based approach and the learning-based approach could work complementarily in QAT. Initializing

quantization parameters using approximation-based method speeds up convergence while learnable quantization parameters promise better performance of the converged model. The details are further discussed in Section IV.

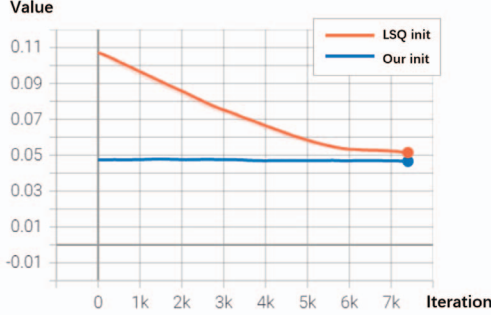


Fig. 2. Scaling factor learning curve of a typical fully-connected layer in ViT.

B. Quantize Multi-Head Attention

the l -th Transformer block in ViT can be formulated as

$$\begin{aligned} Y_l &= \text{LayerNorm}(X_l + \text{MHA}(X_l)) \\ X_{l+1} &= \text{LayerNorm}(Y_l + \text{MLP}(Y_l)), \end{aligned}$$

$X_l \in \mathbb{R}^{n \times d}$ and $Y_l \in \mathbb{R}^{n \times d}$ are inputs of the l -th MHA layer and MLP layer, respectively. The LayerNorm stands for the normalization technique proposed by [49].

The major difference between Transformer-based networks and conventional neural networks is the Multi-Head Attention (MHA) module, which is the computation overhead when high resolution images are given as model inputs. Here, we quantize all matrix multiplications in MHA including the linear projection and the self-attention operation, as is illustrated in Fig. 1.

For the l -th Transformer layer, its input $X_l \in \mathbb{R}^{n \times d}$ is the activations from the previous layer where n and d are the number of patches and the embedding dimension, respectively. $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$ are the projection weights for Q, K, V matrixes, while the latter are used to compute the self-attention of the input patches. The scaled attention scores are computed as

$$A = \text{Softmax}(QK^\top / \sqrt{d}) \quad (6)$$

And then output of a MHA is computed by

$$Y_l = \text{MHA}(X_l) = AVW_o = \text{Softmax}(QK^\top / \sqrt{d})VW_o \quad (7)$$

For weight quantization, we quantize the linear projection matrixes W_q, W_k, W_v, W_o . For activation quantization, we quantize X_l, Q, K, V, A, V . With weights and activations quantized to 8-bit integers, integer multiplication can be performed to speed up inference.

C. Quantize MLP

The MLP layer in ViT is two fully connected layers stacked together with a nonlinear activation function, which is, in this

case, the Gaussian Error Linear Units (GELU) function. It can be formulated as

$$\text{MLP}(Y_l) = \text{GELU}(Y_l W^1 + b^1) W^2 + b^2. \quad (8)$$

$W^1 \in \mathbb{R}^{d \times d_{mlp}}, b^1 \in \mathbb{R}^{d_{mlp}}$ and $W^2 \in \mathbb{R}^{d_{mlp} \times d}, b^2 \in \mathbb{R}^d$ respectively, where d_{mlp} is the hidden embedding dimension in MLP.

We quantize the weights of the two linear layers W_1, W_2 as well as their activations.

As is done in previous work, we don't quantize the GELU nonlinear activation function, the softmax operation as well as Layer Norm, for these operations require high precision representations, and quantizing these layers would result in great degradation of performance.

IV. EXPERIMENT

In this section, we evaluate the performance of our quantization approach on ImageNet [22], for both post-training quantization and quantization-aware training. We compare our approach with previous work [19] [50] [51] using reported results in [19].

A. Implementation Details

Model structures. The original ViT [4] requires custom datasets that are not available to the public, thus we use DeiT proposed by [5] as our full precision baseline. We apply quantization to both DeiT-Small and DeiT-Tiny, whose structures are illustrated as Table I.

Datasets. Same as our float baseline DeiT [5], we evaluate our quantized model on ImageNet [22], a public dataset for visual classification, containing 1.2 million training images and 50K validation images with labels of 1,000 categories.

Settings. For float baseline, we follow the hyper-parameter settings of the original DeiT [5]. And for post-training quantization, we sample 512 images from the trainset to optimize quantization parameters. As for quantization-aware training, we initialize the model parameters from the pretrained full precision model and finetune it on the original trainset after inserting quantization operations. The learning rate in the finetuning stage is carefully tuned for different model structures and different quantization schemes, of which the optimal value is around $5e-6$. We don't warm up the learning rate and we also discard the learning rate lower bound used in the DeiT training, because experiments show these settings benefit the performance of the quantized models. Any other hyper-parameters are kept unchanged. We finetune the quantized models for 3 epochs, while the full precision model are trained for 300 epochs.

B. Results and Analysis

Post-training quantization. The experimental results for post-training quantization are shown in Table II. Our method outperforms previous state-of-art RAQ [19] by 0.33% for

TABLE I
DEiT MODEL CONFIGURATION

Model	embedding dimension	heads	layers	params	training resolution
DeiT-Tiny	192	3	12	5M	224
DeiT-Small	384	6	12	22M	224

TABLE II
TOP-1 ACCURACY ON IMAGENET FOR POST-TRAINING QUANTIZATION

Model	Method	W-bit	A-bit	Size (MB)	Top-1
DeiT-Tiny	Baseline	32	32	20	74.57
	Ours	8	8	5	74.05
DeiT-Small	Baseline	32	32	88	79.8
	Percentile [50]	8	8	22	73.98
	EasyQuant [51]	8	8	22	76.59
	RAQ [19]	8	8	22	77.47
	Ours	8	8	22	77.80

DeiT-Small, indicating the effectiveness of our method. For DeiT-Tiny, our approach achieve a $4\times$ compression of model size while keeping the accuracy drop within 0.5%.

Quantization-aware training. The results are shown in Table III. The LSQ method refers to the learning-based method proposed by [44], which is currently the state-of-art QAT approach in CNNs. The original authors don't evaluate their method on DeiT so we use our own implementations of LSQ to report the accuracy. It can be seen in the results that the original LSQ [44] fails to reach a persuasive accuracy. After introducing the proposed MSE initialization in the LSQ procedural, we observe a performance boost of 0.3% and achieve 8-bit quantization with negligible performance degradation, i. e., -0.09% and -0.16% for DeiT-Tiny and DeiT-Small, respectively. This indicates that approximation approaches and learning-based approaches can work complementarily in the quantization-aware training. Specifically, approximation approaches establish basic estimation for the quantization parameters while the learning-based approaches offer dynamic corrections during the finetuning stage.

C. Ablation Study

In this section, we evaluate the effects of the quantization-aware training strategies proposed in the former sections, including the approximation-based approach using MSE, the learning-based approach inspired by [44], and last but not least, their combined effects in the process of QAT. The

TABLE III
TOP-1 ACCURACY ON IMAGENET FOR QUANTIZATION-AWARE TRAINING

Model	Method	W-bit	A-bit	Size (MB)	Top-1
DeiT-Tiny	Baseline	32	32	20	74.57
	LSQ [44]	8	8	5	74.16
	Ours	8	8	5	74.48
DeiT-Small	Baseline	32	32	88	79.8
	LSQ	8	8	22	79.20
	Ours	8	8	22	79.64

results of these ablations experiments are shown in Table IV. The MSE label stands for our proposed method of only using MSE to optimize the quantization parameters, i. e. ,

TABLE IV
ABLATION STUDY OF THE PROPOSED QAT SCHEME

Model	MSE	Learned	Top-1
DeiT-Tiny	-	-	74.57
	✓	×	74.36
	×	✓	74.16
	✓	✓	74.48
DeiT-Small	-	-	79.8
	✓	×	79.55
	×	✓	79.20
	✓	✓	79.64

the quantization parameters are kept fixed when we finetune the model parameters. The Learned label refers to that we use gradient descent directly to update the quantization parameters without introducing any other optimization procedural. With both labels checked, the quantization parameters are initialized by the MSE approach and then back propagated to learn the optimal solution. It can be seen in the results that the learning-based approach fails to reach a satisfying accuracy. MSE performs slightly better but still has a gap of 0.25% from the full precision baseline for DeiT-Small. After combining the two approaches, higher accuracy are achieved without introducing extra computation, indicating the effectiveness of the proposed approach.

V. CONCLUSION

In this paper, we apply both post-training quantization and quantization-aware training to vision transformer. For post-training quantization, we propose an approximation-based approach to estimate the optimal quantization parameters that minimize the quantization error measured by mean square error. Post-training quantization saves the trouble of finetuning and access to the original data, thus is friendly for deployment. Nevertheless, it suffers from a considerable performance gap compared to the quantization-aware training approach. Thus we also propose a learning-based quantization-aware training approach that enables end-to-end training for both the model parameters and quantization scaling factors. Moreover, we observe a convergence problem in this approach, and combine it with the approximation-based approach to obtain a higher accuracy. The effectiveness of our methods is verified on ImageNet for various vision transformer models. We surpass the previous state-of-art methods by a large margin in post-training quantization as well as quantization-aware training.

REFERENCES

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.
- [3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

- [4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [5] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020.
- [6] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," *arXiv preprint arXiv:2012.00364*, 2020.
- [7] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.
- [8] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?," *arXiv preprint arXiv:1905.10650*, 2019.
- [9] J. McCarley, R. Chakravarti, and A. Sil, "Structured pruning of a bert-based question answering model," *arXiv preprint arXiv:1910.06360*, 2019.
- [10] A. Fan, E. Grave, and A. Joulin, "Reducing transformer depth on demand with structured dropout," *arXiv preprint arXiv:1909.11556*, 2019.
- [11] Z. Wang, J. Wohlwend, and T. Lei, "Structured pruning of large language models," *arXiv preprint arXiv:1910.04732*, 2019.
- [12] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.
- [13] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.
- [14] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for bert model compression," *arXiv preprint arXiv:1908.09355*, 2019.
- [15] O. Zafrir, G. Boudoukh, P. Izsak, and M. Wasserblat, "Q8bert: Quantized 8bit bert," *arXiv preprint arXiv:1910.06188*, 2019.
- [16] S. Shen, Z. Dong, J. Ye, L. Ma, Z. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer, "Q-bert: Hessian based ultra low precision quantization of bert," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 8815–8821, 2020.
- [17] A. H. Zadeh, I. Edo, O. M. Awad, and A. Moshovos, "Gobo: Quantizing attention-based nlp models for low latency and energy efficient inference," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 811–824, IEEE, 2020.
- [18] G. Prato, E. Charlaix, and M. Rezagholizadeh, "Fully quantized transformer for improved translation," *arXiv e-prints*, 2019.
- [19] Z. Liu, Y. Wang, K. Han, S. Ma, and W. Gao, "Post-training quantization for vision transformer," *arXiv preprint arXiv:2106.14156*, 2021.
- [20] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.
- [21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
- [23] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.
- [24] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*, pp. 525–542, Springer, 2016.
- [25] F. Li, B. Zhang, and B. Liu, "Ternary weight networks," *arXiv preprint arXiv:1605.04711*, 2016.
- [26] P. Wang, Q. Hu, Y. Zhang, C. Zhang, Y. Liu, and J. Cheng, "Two-step quantization for low-bit neural networks," in *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pp. 4376–4384, 2018.
- [27] C. Leng, Z. Dou, H. Li, S. Zhu, and R. Jin, "Extremely low bit neural network: Squeeze the last bit out with admm," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.
- [28] L. Hou and J. T. Kwok, "Loss-aware weight quantization of deep networks," *arXiv preprint arXiv:1802.08635*, 2018.
- [29] L. Hou, Q. Yao, and J. T. Kwok, "Loss-aware binarization of deep networks," *arXiv preprint arXiv:1611.01600*, 2016.
- [30] R. Gray, "Vector quantization," *IEEE Assp Magazine*, vol. 1, no. 2, pp. 4–29, 1984.
- [31] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4820–4828, 2016.
- [32] R. Banner, Y. Nahshan, and D. Soudry, "Post training 4-bit quantization of convolutional networks for rapid-deployment," in *Advances in Neural Information Processing Systems*, pp. 7948–7956, 2019.
- [33] R. Zhao, Y. Hu, J. Dotzel, C. De Sa, and Z. Zhang, "Improving neural network quantization without retraining using outlier channel splitting," in *International Conference on Machine Learning*, pp. 7543–7552, 2019.
- [34] Y. Choukroun, E. Kravchik, F. Yang, and P. Kisilev, "Low-bit quantization of neural networks for efficient inference," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3009–3018, IEEE, 2019.
- [35] P. Wang, Q. Chen, X. He, and J. Cheng, "Towards accurate post-training network quantization via bit-split and stitching," in *International Conference on Machine Learning*, pp. 9847–9856, PMLR, 2020.
- [36] M. Nagel, M. v. Baalen, T. Blankevoort, and M. Welling, "Data-free quantization through weight equalization and bias correction," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1325–1334, 2019.
- [37] Y. Cai, Z. Yao, Z. Dong, A. Gholami, M. W. Mahoney, and K. Keutzer, "Zeroq: A novel zero shot quantization framework," *CoRR*, vol. abs/2001.00281, 2020.
- [38] M. Alizadeh, A. Behboodi, M. van Baalen, C. Louizos, T. Blankevoort, and M. Welling, "Gradient l1 regularization for quantization robustness," *arXiv e-prints*, 2020.
- [39] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 1737–1746, 2015.
- [40] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [41] C. Louizos, M. Reisser, T. Blankevoort, E. Gavves, and M. Welling, "Relaxed quantization for discretized neural networks," in *International Conference on Learning Representations (ICLR)*, 2019.
- [42] S. Jung, C. Son, S. Lee, J. Son, J.-J. Han, Y. Kwak, S. J. Hwang, and C. Choi, "Learning to quantize deep networks by optimizing quantization intervals with task loss," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4350–4359, 2019.
- [43] S. R. Jain, A. Gural, M. Wu, and C. H. Dick, "Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks," *arXiv preprint arXiv:1903.08066*, vol. 2, no. 3, p. 7, 2019.
- [44] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," *arXiv preprint arXiv:1902.08153*, 2019.
- [45] X. Zhao, Y. Wang, X. Cai, C. Liu, and L. Zhang, "Linear symmetric quantization of neural networks for low-precision integer hardware," in *International Conference on Learning Representations*, 2019.
- [46] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "Pact: Parameterized clipping activation for quantized neural networks," *arXiv preprint arXiv:1805.06085*, 2018.
- [47] Y. Bhalgat, J. Lee, M. Nagel, T. Blankevoort, and N. Kwak, "LSQ+: improving low-bit quantization through learnable offsets and better initialization," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pp. 2978–2985, IEEE, 2020.
- [48] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.
- [49] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [50] R. Li, Y. Wang, F. Liang, H. Qin, J. Yan, and R. Fan, "Fully quantized network for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2810–2819, 2019.
- [51] D. Wu, Q. Tang, Y. Zhao, M. Zhang, Y. Fu, and D. Zhang, "Easyquant: Post-training quantization via scale optimization," *arXiv preprint arXiv:2006.16669*, 2020.